



Open Source Used In [talos] CUA: virustotal-engine-go main-git- f168d08-20

Cisco Systems, Inc.

www.cisco.com

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

Text Part Number: 78EE117C99-2772723722

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

In your requests please include the following reference number 78EE117C99-2772723722

Contents

1.1 lz4 1.9.3

1.1.1 Available under license

1.2 zlib 1.3.1

1.2.1 Available under license

1.3 zstd 1.3.1

1.3.1 Available under license

1.4 librdkafka 2.10.0-r0

1.4.1 Notifications

1.4.2 Available under license

1.5 openssl 3.0.15

1.5.1 Available under license

1.6 curl 8.14.1-r2

1.6.1 Available under license

1.7 go.opentelemetry.io/otel/trace 1.38.0

1.7.1 Available under license

1.8 skalibs 2.14.4.0-r0

1.8.1 Available under license

1.9 goautoneg 0.0.0-20191010083416-a7dc8b61c822

1.9.1 Available under license

1.10 libmd 1.1.0-r0

1.10.1 Available under license

1.11 pax-utils 1.3.8-r1

1.11.1 Available under license

1.12 libcap 2.76-r0

1.12.1 Available under license

1.13 perks 1.0.1

1.13.1 Available under license

1.14 github.com/confluentinc/confluent-kafka-go v2 2.12.0

1.14.1 Available under license

1.15 [zlib](#) 1.3.1-r2

1.15.1 Available under license

1.16 [scanelf](#) 1.3.8-r1

1.16.1 Available under license

1.17 [alpine-base](#) 3.22.2-r0

1.17.1 Available under license

1.18 [otel](#) 1.38.0

1.18.1 Available under license

1.19 [alpine-keys](#) 2.5-r0

1.19.1 Available under license

1.20 [otel-trace](#) 1.38.0

1.20.1 Available under license

1.21 [logrus](#) 1.9.3

1.21.1 Available under license

1.22 go.opentelemetry.io/auto/sdk 1.2.1

1.22.1 Available under license

1.23 [protobuf](#) 1.36.10

1.23.1 Available under license

1.24 [utmps](#) 0.1.3.1-r0

1.24.1 Available under license

1.25 [ca-certificates](#) 20250911-r0

1.25.1 Available under license

1.26 [libssl3](#) 3.5.4-r0

1.26.1 Available under license

1.27 [musl](#) 1.2.5-r10

1.27.1 Available under license

1.28 [protobuf-go](#) 1.36.10

1.28.1 Available under license

1.29 [iputils-clockdiff](#) 20240905-r0

1.29.1 Available under license

1.30 [iputils-arping](#) 20240905-r0

1.30.1 Available under license

1.31 [apk-tools](#) 2.14.9-r3

1.31.1 Available under license

1.32 [kelseyhightower-envconfig](#) 1.4.0

1.32.1 Available under license

1.33 go.yaml.in/yaml/v2 2.4.3

1.33.1 Available under license

1.34 otel-metric 1.38.0

1.34.1 Available under license

1.35 openssl 3.5.4-r0

1.35.1 Available under license

1.36 auto-sdk 1.2.1

1.36.1 Available under license

1.37 bash 5.2.37-r0

1.37.1 Available under license

1.38 go.yaml.in-yaml 2.4.3

1.38.1 Available under license

1.39 pam 1.7.0-r4

1.39.1 Available under license

1.40 readline 8.2.13-r1

1.40.1 Available under license

1.41 ca-certificates-bundle 20250911-r0

1.41.1 Available under license

1.42 busybox 1.37.0-r19

1.42.1 Available under license

1.43 shadow 4.17.3-r0

1.43.1 Available under license

1.44 libncursesw 6.5_p20250503-r0

1.44.1 Available under license

1.45 ncurses 6.5_p20250503-r0

1.45.1 Available under license

1.46 utmps-lib 0.1.3.1-r0

1.46.1 Available under license

1.47 iputils-ping 20240905-r0

1.47.1 Available under license

1.48 go-elasticsearch 7.17.10

1.48.1 Available under license

1.49 libcrypto3 3.5.4-r0

1.49.1 Available under license

1.50 alpine-baselayout-data 3.7.0-r0

1.50.1 Available under license

1.51 iputils-tracepath 20240905-r0

1.51.1 Available under license

1.52 iputils 20240905-r0

1.52.1 Available under license

1.53 alpine-baselayout 3.7.0-r0

1.53.1 Available under license

1.54 alpine-release 3.22.2-r0

1.54.1 Available under license

1.55 libbsd 0.12.2-r0

1.55.1 Available under license

1.56 confluentinc-confluent-kafka-go 2.12.0

1.56.1 Available under license

1.57 elastic/go-elasticsearch 7.17.10

1.57.1 Available under license

1.58 ssl-client 1.37.0-r19

1.58.1 Available under license

1.59 libapk2 2.14.9-r3

1.59.1 Available under license

1.60 clientmodel 0.6.2

1.60.1 Available under license

1.61 skalibs-libs 2.14.4.0-r0

1.61.1 Available under license

1.62 musl-utils 1.2.5-r10

1.62.1 Available under license

1.63 github.com/go-logr/stdr 1.2.2

1.63.1 Available under license

1.64 sirupsen-logrus 1.9.3

1.64.1 Available under license

1.65 stdr 1.2.2

1.65.1 Available under license

1.66 cJSON 1.7.19-r0

1.66.1 Notifications

1.66.2 Available under license

1.67 logr 1.4.3

1.67.1 Available under license

1.68 prometheus-client-model 0.6.2

1.68.1 Available under license

1.69 prometheus-client 1.23.2

1.69.1 Available under license

1.70 xxhash 2.3.0

1.70.1 Available under license

1.71 procfs 0.19.2

- 1.71.1 Available under license
- 1.72 stdlib go1.25.4**
 - 1.72.1 Available under license
- 1.73 golang 1.25.4**
 - 1.73.1 Available under license
- 1.74 x-sys 0.38.0**
 - 1.74.1 Available under license
- 1.75 common 0.67.3**
 - 1.75.1 Available under license
- 1.76 prometheus-common 0.67.3**
 - 1.76.1 Available under license
- 1.77 github.com/elastic/elastic-transport-go/v8 8.8.0**
 - 1.77.1 Available under license
- 1.78 elastic-transport-go 8.8.0**
 - 1.78.1 Available under license

1.1 lz4 1.9.3

1.1.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

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

Preamble

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

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

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it

if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

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

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

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

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

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

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

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

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

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

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

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

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

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

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

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

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in

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

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

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

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

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

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

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

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

itself accompanies the executable.

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

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

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

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

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

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

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

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

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

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

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

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

of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,
THERE IS NO WARRANTY

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

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

THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

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

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU

General Public License as published by

the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

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

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

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

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

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

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

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

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

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

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this
is what you want to do, use the GNU Lesser General
Public License instead of this License.
Copyright (c) 2014, lpsantil
All rights reserved.

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

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

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

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

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

This repository uses 2 different licenses :

- all files in the `lib` directory use a BSD 2-Clause license
- all other files use a GPLv2 license, unless explicitly stated otherwise

Relevant license is reminded at the top of each source file, and with presence of COPYING or LICENSE file in associated directories.

This model is selected to emphasize that files in the `lib` directory are designed to be included into 3rd party applications, while all other files, in `programs`, `tests` or `examples`, receive more limited attention and support for such scenario.

LZ4 Library

Copyright (c) 2011-2016, Yann Collet

All rights reserved.

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

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

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

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

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

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

Upstream-Name: liblz4

Upstream-Contact: Yann Collet <Cyan4973@github.com>

Source: <https://github.com/lz4/lz4>

Files: *

Copyright: (C) 2011-2020 Yann Collet

License: GPL-2+

The full text of license: <https://github.com/lz4/lz4/blob/dev/lib/LICENSE>

1.2 zlib 1.3.1

1.2.1 Available under license :

Copyright notice:

(C) 1995-2022 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler

jloup@gzip.org madler@alumni.caltech.edu

/* zlib.h -- interface of the 'zlib' general purpose compression library
version 1.3.1, January 22nd, 2024

Copyright (C) 1995-2024 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied

warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

The data format used by the zlib library is described by RFCs (Request for Comments) 1950 to 1952 in the files <http://tools.ietf.org/html/rfc1950> (zlib format), rfc1951 (deflate format) and rfc1952 (gzip format).

*/

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
/* zlib.c -- zlib functions common to reading and writing gzip files
* Copyright (C) 2004-2024 Mark Adler
* For conditions of distribution and use, see copyright notice in zlib.h
*/
```

1.3 zstd 1.3.1

1.3.1 Available under license :

BSD-style without advertising clause

1.4 librdkafka 2.10.0-r0

1.4.1 Notifications :

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)

This product includes cryptographic software written by Eric Young (ey@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

1.4.2 Available under license :

For the files wingetopt.c wingetopt.h downloaded from <https://github.com/alex85k/wingetopt>

```
/*
* Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>
*
* Permission to use, copy, modify, and distribute this software for any
* purpose with or without fee is hereby granted, provided that the above
* copyright notice and this permission notice appear in all copies.
*
* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Sponsored in part by the Defense Advanced Research Projects
* Agency (DARPA) and Air Force Research Laboratory,
* Air Force
* Materiel Command, USAF, under agreement number F39502-99-1-0512.
*/
/*-
* Copyright (c) 2000 The NetBSD Foundation, Inc.
* All rights reserved.
*
* This code is derived from software contributed to The NetBSD Foundation
* by Dieter Baron and Thomas Klausner.
```

*
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
 * ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
 * TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
 * AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
 * BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 */
 src/rdxxhash.[ch] src/lz4*.[ch]: git@github.com:lz4/lz4.git 5ff839680134437dbf4678f3d0c7b371d84f4964

LZ4 Library
 Copyright (c) 2011-2020, Yann Collet
 All rights reserved.

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

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

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

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

parts of src/rdfnv1a.c: http://www.isthe.com/chongo/src/fnv/hash_32a.c

Please do not copyright this code. This code is in the public domain.

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

By:

chongo <Landon Curt Noll> /oo/
<http://www.isthe.com/chongo/>

Share and Enjoy! :-)

regex.c and regex.h from <https://github.com/ccxvii/minilibs> sha 875c33568b5a4aa4fb3dd0c52ea98f7f0e5ca684

"

These libraries are in the public domain (or the equivalent where that is not possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.

"

FindZstd.cmake: [git@github.com:facebook/folly](https://github.com/facebook/folly).git 87f1a403b49552dae75ae94c8610dd5979913477

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

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

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

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

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

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

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

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

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for

reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed

to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if

You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

For src/crc32c.c copied (with modifications) from

<http://stackoverflow.com/a/17646775/1821055>

/* crc32c.c -- compute CRC-32C using the Intel crc32 instruction

* Copyright (C) 2013 Mark Adler

* Version 1.1 1 Aug 2013 Mark Adler

*/

/*

This software is provided 'as-is', without any express or implied
warranty. In no event will the author be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original
software.
3. This notice may not be removed or altered from any source distribution.

Mark Adler
madler@alumni.caltech.edu

*/

For sys/queue.h:

* Copyright (c) 1991, 1993
* The Regents of the University of California. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 4. Neither the name of the University nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* @(#)queue.h 8.5 (Berkeley) 8/20/94
* \$FreeBSD\$

For cJSON.c and cJSON.h:

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

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

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

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

LICENSE

librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2022, Magnus Edenhill
2023, Confluent Inc.
All rights reserved.

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

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

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE

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

LICENSE.cjson

For cJSON.c and cJSON.h:

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

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

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

LICENSE.crc32c

For src/crc32c.c copied (with modifications) from
<http://stackoverflow.com/a/17646775/1821055>

/* crc32c.c -- compute CRC-32C using the Intel crc32 instruction
* Copyright (C) 2013 Mark Adler
* Version 1.1 1 Aug 2013 Mark Adler
*/

/*
This software is provided 'as-is',
without any express or implied
warranty. In no event will the author be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Mark Adler

madler@alumni.caltech.edu

*/

LICENSE.fnv1a

parts of src/rdfnv1a.c: http://www.isthe.com/chongo/src/fnv/hash_32a.c

Please

do not copyright this code. This code is in the public domain.

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

By:

chongo <Landon Curt Noll> ^oo^
<http://www.isthe.com/chongo/>

Share and Enjoy! :-)

LICENSE.hdrhistogram

This license covers src/rdhdrhistogram.c which is a C port of
Coda Hale's Golang HdrHistogram <https://github.com/codahale/hdrhistogram>
at revision 3a0bb77429bd3a61596f5e8a3172445844342120

The MIT License
(MIT)

Copyright (c) 2014 Coda Hale

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

The above copyright notice and this permission notice shall be included in

all copies or substantial portions of the Software.

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

LICENSE.lz4

src/rdxxhash.[ch] src/lz4*.ch: git@github.com:lz4/lz4.git 5ff839680134437dbf4678f3d0c7b371d84f4964

LZ4 Library

Copyright (c) 2011-2020, Yann Collet

All rights reserved.

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

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

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

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

LICENSE.murmur2

parts of src/rdmurmur2.c: git@github.com:abandoned/murmur2.git

MurMurHash2 Library

```
//-----  
// MurmurHash2 was written by Austin Appleby, and is placed in the public  
// domain. The author hereby disclaims copyright to this source code.
```

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

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

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

LICENSE.nanopb

For

files in src/nanopb :

<https://github.com/nanopb/nanopb/blob/8ef41e0ebd45daaf19459a011f67e66224b247cd/LICENSE.txt>

Copyright (c) 2011 Petteri Aimonen <jpa at nanopb.mail.kapsi.fi>

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may
not be removed or altered from any source
distribution.

LICENSE.opentelemetry

For files in src/opentelemetry: <https://github.com/open-telemetry/opentelemetry-proto/blob/81a296f9dba23e32d77f46d58c8ea4244a2157a6/LICENSE>

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes
of this definition,

"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

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

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

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

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated

by a

copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

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

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

LICENSE.pycrc

The following license applies to the files rdcrc32.c and rdcrc32.h which
have been generated by the pycrc tool.

=====

Copyright (c) 2006-2012, Thomas Pircher <tehpeh@gmx.net>

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

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

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

LICENSE.queue

For

sys/queue.h:

- * Copyright (c) 1991, 1993
- * The Regents of the University of California. All rights reserved.
- *
- * Redistribution and use in source and binary forms, with or without
- * modification, are permitted provided that the following conditions
- * are met:
- * 1. Redistributions of source code must retain the above copyright
- * notice, this list of conditions and the following disclaimer.
- * 2. Redistributions in binary form must reproduce the above copyright
- * notice, this list of conditions and the following disclaimer in the
- * documentation and/or other materials provided with the distribution.
- * 4. Neither the name of the University nor the names of its contributors
- * may be used to endorse or promote products derived from this software
- * without specific prior written permission.
- *
- * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
- * ANY EXPRESS OR
- * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
- * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
- * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
- * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
- * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
- * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
- * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
- * SUCH DAMAGE.
- *
- * @(#)queue.h 8.5 (Berkeley) 8/20/94
- * \$FreeBSD\$

LICENSE.regexp

regex.c and regex.h from <https://github.com/ccxvii/minilibs> sha 875c33568b5a4aa4fb3dd0c52ea98f7f0e5ca684

"

These libraries are in the public domain (or the equivalent where that is not possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.

"

LICENSE.snappy

```
-----  
#####  
# LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h #  
# originally retrieved from http://github.com/andikleen/snappy-c #  
# git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219 #  
#####
```

The snappy-c code is under the same license as the original snappy source

Copyright 2011 Intel Corporation All Rights Reserved.

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

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

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

- * Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

LICENSE.tinythread

From <https://github.com/tinythread/tinythread/README.txt> c57166cd510ffb5022dd5f127489b131b61441b9

License

Copyright (c) 2012 Marcus Geelnard
2013-2014 Evan Nemerson

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

LICENSE.wingetopt

For the files wingetopt.c wingetopt.h downloaded from <https://github.com/alex85k/wingetopt>

/*

* Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>

*

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

*

* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR

ANY DAMAGES

* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

*

* Sponsored in part by the Defense Advanced Research Projects
* Agency (DARPA) and Air Force Research Laboratory, Air Force
* Materiel Command, USAF, under agreement number F39502-99-1-0512.

*/

/*_

* Copyright (c) 2000 The NetBSD Foundation, Inc.

* All rights reserved.

*

* This code is derived from software contributed to The NetBSD Foundation
* by Dieter Baron and Thomas Klausner.

*

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

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

this list of conditions and the following disclaimer in the

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

*

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

*/

#####

LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h

originally retrieved from <http://github.com/andikleen/snappy-c>

git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219

#####

The snappy-c code is under the same license as the original snappy source

Copyright 2011 Intel Corporation All Rights Reserved.

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

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

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

- * Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2022, Magnus Edenhill
2023, Confluent Inc.
All rights reserved.

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

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

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

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

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

Upstream-Name: librdkafka

Source: <https://github.com/confluentinc/librdkafka>

License: BSD-2-clause

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

.

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

.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR

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

Files: *

Copyright: 2012-2022, Magnus Edenhill; 2023, Confluent Inc.

License: BSD-2-clause

Files: src/rdcrc32.c src/rdcrc32.h

Copyright: 2006-2012, Thomas Pircher <tehpeh@gmx.net>

License: MIT

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

.

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

.

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

Files: src/snappy.c src/snappy.h src/snappy_compat.h

Copyright: 2005, Google Inc.

2011, Intel Corporation

License: BSD-3-clause

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

.

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

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

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

.

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

Files: src/rdxxhash.h src/rdxxhash.c

Copyright: 2012-2014, Yann Collet

License: BSD-2-clause

Files: src/regex.h src/regex.c

Copyright: n/a

License: public-domain

License: public-domain

The files tagged with this license contain the following paragraphs:

.

These libraries are in the public domain (or the equivalent where that is not possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.

Files: debian/*

Copyright: 2013 Faidon Liambotis <paravoid@debian.org>

License: BSD-2-clause

BSD-style without advertising clause

Copyright (c) 2016-2019, Magnus Edenhill

All rights reserved.

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

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

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

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

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

For files in src/nanopb :

<https://github.com/nanopb/nanopb/blob/8ef41e0ebd45daaf19459a011f67e66224b247cd/LICENSE.txt>

Copyright (c) 2011 Petteri Aimonen <jpa at nanopb.mail.kapsi.fi>

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and

redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source

distribution.

parts of src/rdmurmur2.c: `git@github.com:abandoned/murmur2.git`

MurMurHash2 Library

//-----

// MurmurHash2 was written by Austin Appleby, and is placed in the public
// domain. The author hereby disclaims copyright to this source code.

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

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

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

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

Upstream-Name: librdkafka

Source: <https://github.com/confluentinc/librdkafka>

License: BSD-2-clause

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

.

1. Redistributions of source code must retain the above copyright notice,

this list of conditions and the following disclaimer.

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

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

Files: *

Copyright: 2012-2022, Magnus Edenhill; 2023 Confluent Inc.

License: BSD-2-clause

Files: src/rdcrc32.c src/rdcrc32.h

Copyright: 2006-2012, Thomas Pircher <tehpeh@gmx.net>

License: MIT

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

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

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

Files: src/snappy.c src/snappy.h src/snappy_compat.h

Copyright: 2005, Google Inc.

2011, Intel Corporation

License: BSD-3-clause

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

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

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

Files: debian/*

Copyright: 2013 Faidon Liambotis <paravoid@debian.org>

License: BSD-2-clause

This license covers src/rdhdrhistogram.c which is a C port of Coda Hale's Golang HdrHistogram <https://github.com/codahale/hdrhistogram> at revision 3a0bb77429bd3a61596f5e8a3172445844342120

The MIT License (MIT)

Copyright (c) 2014 Coda Hale

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

The above copyright notice and this permission notice shall be included in

all copies or substantial portions of the Software.

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

From <https://github.com/tinychread/tinychread/README.txt> c57166cd510ffb5022dd5f127489b131b61441b9

License

Copyright (c) 2012 Marcus Geelnard
2013-2014 Evan Nemerson

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

The following license applies to the files rdcrc32.c and rdcrc32.h which have been generated by the pycrc tool.

=====

Copyright (c) 2006-2012, Thomas Pircher <tehpeh@gmx.net>

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

furnished to do so, subject to the following conditions:

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

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

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.5 openssl 3.0.15

1.5.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

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

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

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

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

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

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

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

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

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

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be

distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

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

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

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

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use

in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

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

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

- a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,
- b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,
- c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

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

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

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

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

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

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

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

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

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

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) 19yy <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

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

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

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

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

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

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

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

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (a program to direct compilers to make passes at assemblers) written by James Hacker.

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

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the

computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
 - a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
 - b) use the modified Package only within your corporation or organization.
 - c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
 - d) make other distribution arrangements with the Copyright Holder.
4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:
 - a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.
 - b) accompany the distribution with the machine-readable source of the Package with your modifications.
 - c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names,

and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

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

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

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

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

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

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

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

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

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.6 curl 8.14.1-r2

1.6.1 Available under license :

Copyright (c) <year> <owner>.

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

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

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

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

COPYRIGHT AND PERMISSION NOTICE

Copyright (C) Daniel Stenberg, <daniel@haxx.se>, and many contributors, see the THANKS file.

All rights reserved.

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

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

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this

Software without prior written authorization of the copyright holder.

BSD-4-Clause (University of California-Specific)

Copyright [various years] The Regents of the University of California. All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

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

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

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2025, Daniel Stenberg, <daniel@haxx.se>, and many contributors, see the THANKS file.

All rights reserved.

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

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

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

1.7 go.opentelemetry.io/otel/trace 1.38.0

1.7.1 Available under license :

Apache License

Version 2.0, January 2004

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

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

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

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

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

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Copyright 2009 The Go Authors.

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

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

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

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

1.8 skalibs 2.14.4.0-r0

1.8.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

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

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

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

Preamble

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

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

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

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

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

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

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

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

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

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

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

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the

notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

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

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

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

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

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

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

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

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

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

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

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

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

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

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt

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

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

6.

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

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

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

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made

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

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

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

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

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

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

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

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

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

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

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

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

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

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

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

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

The hypothetical commands

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

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

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

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

```
Ty Coon, President of Vice
```

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

```
# Automatically generated by apkgbuild-cpan, template 4
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkgbuild-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
```

```

builddir="$srcdir/${_pkgreal-$pkgver}"

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

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

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

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

# Contributor: Maxim Karasev <begs@disroot.org>
# Maintainer: mio <miyopan@e.email>
pkgname=bsd-games
pkgver=3.3
pkgrel=2
pkgdesc="traditional text mode games from BSD"
url="https://bsd-games.sourceforge.io/"
arch="all"
license="BSD-3-Clause"
install="$pkgname.post-install"
makedepends="ncurses-dev coreutils" # configure causes busybox expr error
subpackages="$pkgname-doc"
source="https://sourceforge.net/projects/bsd-games/files/bsd-games-$pkgver.tar.gz
0001-bsd-games-scores-dir.patch
"

options="!check" # no tests

build() {
    # some GNU autoconf options are ignored, but it works
    ./configure \
        --build=$CBUILD \
        --host=$CHOST \
        --prefix=/usr \

```

```

--sysconfdir=/etc \
--mandir=/usr/share/man \
--sourcedir=/usr/share/$pkgname
make
}

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

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

MIT License

```

Copyright (c) 2023 Leaf Corcoran

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

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

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

Maintainer: Natanael Copa <ncopa@alpinelinux.org>

pkgname=libart-lgpl

pkgver=2.3.21

pkgrel=8

pkgdesc="A library for high-performance 2D graphics"

url="https://www.levien.com/libart/"

arch="all"

options="!check" # No test suite.

```

license="LGPL-2.0-or-later"
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart_lgpl/2.3/libart_lgpl-$pkgver.tar.bz2"
"

builddir="$srcdir"/libart_lgpl-$pkgver

prepare() {
    update_config_sub
    default_prepare
}

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

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

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"
mini_sendmail - accept email on behalf of real sendmail

```

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

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

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

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

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

Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>

Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>

pkgname=spdx-licenses

pkgver=3.26.0

pkgrel=0

pkgdesc="Various data formats for the SPDX License List"

url="https://spdx.org/"

arch="noarch"

options="!check" # no test suite

license="CC-BY-3.0"

source="license-list-data-\$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v\$pkgver.tar.gz"

builddir="\$srcdir/license-list-data-\$pkgver"

subpackages="\$pkgname-list"

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

for type in \$_types; do

subpackages="\$subpackages \$pkgname-\$type:_subpkg"

done

package() {

mkdir -p "\$pkgdir"

}

_subpkg() {

local type=\${subpkgname/\$pkgname-/}

pkgdesc="\$pkgdesc (\$type)"

install_if="\$pkgname=\$pkgver-r\$pkgrel"

mkdir -p "\$subpkgdir"/usr/share/spdx

cp -r "\$builddir"/\$type "\$subpkgdir"/usr/share/spdx/

}

list() {

pkgdesc="\$pkgdesc (licence list)"

mkdir -p "\$subpkgdir"/usr/share/spdx

local i; for i in "\$builddir"/text/*.txt;

do

local license=\${i##*/}

echo \${license%.*} >> "\$subpkgdir"/usr/share/spdx/license.lst

done

}

sha512sums="

67e618d4642dfe4f366935d9514d6f7941d3711e77c49bc1b340ac572d36015981b8f10c72a9f80f1d83e132ae7e5093c

```

0155e0c0e786e4777ee16d1cb7b885a license-list-data-3.26.0.tar.gz
"
# Contributor: Patrycja Rosa <alpine@ptrcnull.me>
# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
cdefs.h
queue.h
tree.h
"
builddir="$srcdir"
options="!check" # just headers

package() {
    mkdir -p "$pkgdir"
    install -Dm644 -t "$pkgdir"/usr/include/sys \
        cdefs.h queue.h tree.h
}
sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178 cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750 queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3 tree.h
"

Original Copyright (c) 1985 by Supoj Sutanthavibul
Parts Copyright (c) 1989-2012 by Brian V. Smith
Parts Copyright (c) 1991 by Paul King
Other Copyrights may be found in various files

```

Main Xfig copyright notice:

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

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

Parts Copyright (C) 1993 by Alan Richardson (mppa3@uk.ac.sussex.syma)

The text rotation code in w_rottext.c was written by Alan Richardson.

The above copyright notice holds for this work as well.

Parts Copyright (c) 1994 by Anthony Dekker

The Kohonen neural network code for color optimization was written by Anthony Dekker.

Parts Copyright (c) 1995 by C. Blanc and C. Schlick

The X-Spline code was written Carole Blanc (blanc@labri.u-bordeaux.fr) and Christophe Schlick (schlick@labri.u-bordeaux.fr) starting from an initial implementation done by C. Feuille, S. Grosbois, L. Maziere and L. Minihot as a student practice (Universite Bordeaux, France). For additional information about X-splines, see:

"X-Splines: A Spline Model Designed for the End User" by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95

<http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/sig1.html>

Contributor: Dhruvin Gandhi <contact@dhruvin>

Maintainer: Patrycja Rosa <alpine@ptrcnull.me>

pkgname=py3-license-expression

_pkgname=license_expression

pkgver=30.4.1

pkgrel=0

pkgdesc="Library to parse, compare, simplify and normalize license expressions"

url="https://github.com/nexB/license-expression"

arch="noarch"

license="Apache-2.0"

depends="python3 py3-boolean.py"

makedepends="py3-setuptools py3-setuptools_scm py3-gpep517 py3-wheel"

checkdepends="py3-pytest py3-pytest-xdist"

subpackages="\$pkgname-pyc"

source="https://files.pythonhosted.org/packages/source/l/_pkgname/_pkgname-\$pkgver.tar.gz"

builddir="\$srcdir/\$pkgname-\$pkgver"

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

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

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

sha512sums="

```

114c630f098939757e49187e35b4e54883266a2ee9ca283724992420ef4c860f9606a8342da0ab74d93627c10805f312
fda3f699cbc7c4fbd47a71e220440fb0
    license_expression-30.4.1.tar.gz
"

/* $NetBSD: queue.h,v 1.70 2015/11/02 15:21:23 christos Exp $ */

/*
 * Copyright (c) 1991, 1993
 * The Regents of the University of California. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
 * BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * @(#)queue.h 8.5 (Berkeley) 8/20/94
 */

#ifndef _SYS_QUEUE_H_
#define _SYS_QUEUE_H_

/*
 * This file defines five types of data structures: singly-linked lists,
 * lists, simple queues, tail queues, and circular queues.
 *
 * A singly-linked list is headed by a single forward pointer. The
 * elements are singly linked for minimum space and

```

pointer manipulation

- * overhead at the expense of $O(n)$ removal for arbitrary elements. New
- * elements can be added to the list after an existing element or at the
- * head of the list. Elements being removed from the head of the list
- * should use the explicit macro for this purpose for optimum
- * efficiency. A singly-linked list may only be traversed in the forward
- * direction. Singly-linked lists are ideal for applications with large
- * datasets and few or no removals or for implementing a LIFO queue.

*

- * A list is headed by a single forward pointer (or an array of forward
- * pointers for a hash table header). The elements are doubly linked
- * so that an arbitrary element can be removed without a need to
- * traverse the list. New elements can be added to the list before
- * or after an existing element or at the head of the list. A list
- * may only be traversed in the forward direction.

*

- * A simple queue is headed by a pair of pointers, one the head of the
- * list and the

other to the tail of the list. The elements are singly

- * linked to save space, so elements can only be removed from the
- * head of the list. New elements can be added to the list after
- * an existing element, at the head of the list, or at the end of the
- * list. A simple queue may only be traversed in the forward direction.

*

- * A tail queue is headed by a pair of pointers, one to the head of the
- * list and the other to the tail of the list. The elements are doubly
- * linked so that an arbitrary element can be removed without a need to
- * traverse the list. New elements can be added to the list before or
- * after an existing element, at the head of the list, or at the end of
- * the list. A tail queue may be traversed in either direction.

*

- * A circle queue is headed by a pair of pointers, one to the head of the
- * list and the other to the tail of the list. The elements are doubly
- * linked so that an arbitrary element can be removed without a need to
- * traverse the list. New elements

can be added to the list before or after

- * an existing element, at the head of the list, or at the end of the list.
- * A circle queue may be traversed in either direction, but has a more
- * complex end of list detection.

*

- * For details on the use of these macros, see the queue(3) manual page.

*/

/*

- * Include the definition of NULL only on NetBSD because sys/null.h
- * is not available elsewhere. This conditional makes the header
- * portable and it can simply be dropped verbatim into any system.
- * The caveat is that on other systems some other header

```

* must provide NULL before the macros can be used.
*/
#ifdef __NetBSD__
#include <sys/null.h>
#endif

#if defined(QUEUEDEBUG)
# if defined(_KERNEL)
#  define QUEUEDEBUG_ABORT(...) panic(__VA_ARGS__)
# else
#  include <err.h>
#  define QUEUEDEBUG_ABORT(...) err(1, __VA_ARGS__)
# endif
#endif

/*
* Singly-linked List definitions.
*/
#define SLIST_HEAD(name, type) \
struct name { \
    struct type *slh_first; /* first \
    element */ \
}

#define SLIST_HEAD_INITIALIZER(head) \
{ NULL }

#define SLIST_ENTRY(type) \
struct { \
    struct type *sle_next; /* next element */ \
}

/*
* Singly-linked List access methods.
*/
#define SLIST_FIRST(head) ((head)->slh_first)
#define SLIST_END(head) NULL
#define SLIST_EMPTY(head) ((head)->slh_first == NULL)
#define SLIST_NEXT(elm, field) ((elm)->field.sle_next)

#define SLIST_FOREACH(var, head, field) \
for((var) = (head)->slh_first; \
    (var) != SLIST_END(head); \
    (var) = (var)->field.sle_next)

#define SLIST_FOREACH_SAFE(var, head, field, tvar) \
for ((var) = SLIST_FIRST((head)); \
    (var) != SLIST_END(head) && \

```

```

        ((tvar) = SLIST_NEXT((var), field), 1); \
        (var) = (tvar))

/*
 * Singly-linked List functions.
 */
#define SLIST_INIT(head) do { \
    (head)->slh_first = SLIST_END(head); \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_AFTER(slistelm, elm, field) do { \
    (elm)->field.sle_next = (slistelm)->field.sle_next; \
    (slistelm)->field.sle_next \
    = (elm); \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_HEAD(head, elm, field) do { \
    (elm)->field.sle_next = (head)->slh_first; \
    (head)->slh_first = (elm); \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_AFTER(slistelm, field) do { \
    (slistelm)->field.sle_next = \
        SLIST_NEXT(SLIST_NEXT((slistelm), field), field); \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_HEAD(head, field) do { \
    (head)->slh_first = (head)->slh_first->field.sle_next; \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE(head, elm, type, field) do { \
    if ((head)->slh_first == (elm)) { \
        SLIST_REMOVE_HEAD((head), field); \
    } \
    else { \
        struct type *curelm = (head)->slh_first; \
        while(curelm->field.sle_next != (elm)) \
            curelm = curelm->field.sle_next; \
        curelm->field.sle_next = \
            curelm->field.sle_next->field.sle_next; \
    } \
} while (/*CONSTCOND*/0)

/*
 * List definitions.
 */
#define LIST_HEAD(name, type) \

```

```

struct name {      \
    struct type *lh_first; /* first element */ \
}

#define LIST_HEAD_INITIALIZER(head) \
    { NULL }

#define LIST_ENTRY(type) \
struct {      \
    struct type *le_next; /* next element */ \
    struct type **le_prev; /* address of previous next element */ \
}

/*
 * List access methods.
 */
#define LIST_FIRST(head) ((head)->lh_first)
#define LIST_END(head) NULL
#define LIST_EMPTY(head) ((head)->lh_first == LIST_END(head))
#define LIST_NEXT(elm, field) ((elm)->field.le_next)

#define LIST_FOREACH(var, head, field) \
    for ((var) = ((head)->lh_first); \
         (var) != LIST_END(head); \
         (var) = ((var)->field.le_next))

#define LIST_FOREACH_SAFE(var, head, field, tvar) \
    for ((var) = LIST_FIRST((head)); \
         (var) != LIST_END(head) && \
         ((tvar) = LIST_NEXT((var), field), 1); \
         (var) = (tvar))

#define LIST_MOVE(head1, head2) do { \
    LIST_INIT((head2)); \
    if (!LIST_EMPTY((head1))) \
    { \
        (head2)->lh_first = (head1)->lh_first; \
        LIST_INIT((head1)); \
    } \
} while (/*CONSTCOND*/0)

/*
 * List functions.
 */
#ifdef QUEUEDEBBUG
#define QUEUEDEBBUG_LIST_INSERT_HEAD(head, elm, field) \
    if ((head)->lh_first && \
        (head)->lh_first->field.le_prev != &(head)->lh_first) \

```

```

    QUEUEDEBUG_ABORT("LIST_INSERT_HEAD %p %s:%d", (head), \
        __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_OP(elm, field) \
    if ((elm)->field.le_next && \
        (elm)->field.le_next->field.le_prev != \
        &(elm)->field.le_next) \
        QUEUEDEBUG_ABORT("LIST_* forw %p %s:%d", (elm), \
            __FILE__, __LINE__); \
    if (*(elm)->field.le_prev != (elm)) \
        QUEUEDEBUG_ABORT("LIST_* back %p %s:%d", (elm), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field) \
    (elm)->field.le_next = (void *)1L; \
    (elm)->field.le_prev = (void *)1L;
#else
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_LIST_OP(elm,
    field)
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field)
#endif

#define LIST_INIT(head) do { \
    (head)->lh_first = LIST_END(head); \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_AFTER(listelm, elm, field) do { \
    QUEUEDEBUG_LIST_OP((listelm), field) \
    if (((elm)->field.le_next = (listelm)->field.le_next) != \
        LIST_END(head)) \
        (listelm)->field.le_next->field.le_prev = \
        &(elm)->field.le_next; \
    (listelm)->field.le_next = (elm); \
    (elm)->field.le_prev = &(listelm)->field.le_next; \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_BEFORE(listelm, elm, field) do { \
    QUEUEDEBUG_LIST_OP((listelm), field) \
    (elm)->field.le_prev = (listelm)->field.le_prev; \
    (elm)->field.le_next = (listelm); \
    *(listelm)->field.le_prev = (elm); \
    (listelm)->field.le_prev = &(elm)->field.le_next; \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBUG_LIST_INSERT_HEAD((head), (elm), field) \
    if (((elm)->field.le_next \
        = (head)->lh_first) != LIST_END(head)) \
        (head)->lh_first->field.le_prev = &(elm)->field.le_next; \
} while (/*CONSTCOND*/0)

```

```

(head)->lh_first = (elm); \
(elm)->field.le_prev = &(head)->lh_first; \
} while (/*CONSTCOND*/0)

#define LIST_REMOVE(elm, field) do { \
    QUEUEDEBUG_LIST_OP((elm), field) \
    if ((elm)->field.le_next != NULL) \
        (elm)->field.le_next->field.le_prev = \
            (elm)->field.le_prev; \
    *(elm)->field.le_prev = (elm)->field.le_next; \
    QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

#define LIST_REPLACE(elm, elm2, field) do { \
    if (((elm2)->field.le_next = (elm)->field.le_next) != NULL) \
        (elm2)->field.le_next->field.le_prev = \
            &(elm2)->field.le_next; \
    (elm2)->field.le_prev = (elm)->field.le_prev; \
    *(elm2)->field.le_prev = (elm2); \
    QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

/*
 * Simple queue definitions.
 */
#define SIMPLEQ_HEAD(name, type) \
struct name { \
    struct \
    type *sqh_first; /* first element */ \
    struct type **sqh_last; /* addr of last next element */ \
}

#define SIMPLEQ_HEAD_INITIALIZER(head) \
{ NULL, &(head).sqh_first }

#define SIMPLEQ_ENTRY(type) \
struct { \
    struct type *sqe_next; /* next element */ \
}

/*
 * Simple queue access methods.
 */
#define SIMPLEQ_FIRST(head) ((head)->sqh_first)
#define SIMPLEQ_END(head) NULL
#define SIMPLEQ_EMPTY(head) ((head)->sqh_first == SIMPLEQ_END(head))
#define SIMPLEQ_NEXT(elm, field) ((elm)->field.sqe_next)

```



```

#define SIMPLEQ_FOREACH(var, head, field) \
for ((var) = ((head)->sqh_first); \
    (var) != SIMPLEQ_END(head); \
    (var) = ((var)->field.sqe_next))

#define SIMPLEQ_FOREACH_SAFE(var, head, field, next) \
for ((var) = ((head)->sqh_first); \
    (var) != SIMPLEQ_END(head) && \
    ((next = ((var)->field.sqe_next)), 1); \
    (var) = (next))

/*
 * Simple queue functions.
 */
#define SIMPLEQ_INIT(head) do { \
    (head)->sqh_first = NULL; \
    (head)->sqh_last \
    = &(head)->sqh_first; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_HEAD(head, elm, field) do { \
    if (((elm)->field.sqe_next = (head)->sqh_first) == NULL) \
        (head)->sqh_last = &(elm)->field.sqe_next; \
    (head)->sqh_first = (elm); \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_TAIL(head, elm, field) do { \
    (elm)->field.sqe_next = NULL; \
    *(head)->sqh_last = (elm); \
    (head)->sqh_last = &(elm)->field.sqe_next; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_AFTER(head, listelm, elm, field) do { \
    if (((elm)->field.sqe_next = (listelm)->field.sqe_next) == NULL) \
        (head)->sqh_last = &(elm)->field.sqe_next; \
    (listelm)->field.sqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_HEAD(head, field) do { \
    if (((head)->sqh_first = (head)->sqh_first->field.sqe_next) == NULL) \
        (head)->sqh_last = &(head)->sqh_first; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_AFTER(head, elm, field) do { \
    if \
        (((elm)->field.sqe_next = (elm)->field.sqe_next->field.sqe_next) \
        == NULL) \
        (head)->sqh_last = &(elm)->field.sqe_next; \
} while (/*CONSTCOND*/0)

```

```

} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE(head, elm, type, field) do { \
    if ((head)->sqh_first == (elm)) { \
        SIMPLEQ_REMOVE_HEAD((head), field); \
    } else { \
        struct type *curelm = (head)->sqh_first; \
        while (curelm->field.sqe_next != (elm)) \
            curelm = curelm->field.sqe_next; \
        if ((curelm->field.sqe_next = \
            curelm->field.sqe_next->field.sqe_next) == NULL) \
            (head)->sqh_last = &(curelm->field.sqe_next); \
    } \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_CONCAT(head1, head2) do { \
    if (!SIMPLEQ_EMPTY((head2))) { \
        *(head1)->sqh_last = (head2)->sqh_first; \
        (head1)->sqh_last = (head2)->sqh_last; \
        SIMPLEQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_LAST(head, type, field) \
    (SIMPLEQ_EMPTY((head)) ? \
    NULL : \
    ((struct type \
    *) (void *) \
    ((char *)((head)->sqh_last) - offsetof(struct type, field))))

/*
 * Tail queue definitions.
 */

#define _TAILQ_HEAD(name, type, qual) \
    struct name { \
        qual type *tqh_first; /* first element */ \
        qual type *qual *tqh_last; /* addr of last next element */ \
    }

#define TAILQ_HEAD(name, type) _TAILQ_HEAD(name, struct type,)

#define TAILQ_HEAD_INITIALIZER(head) \
    { TAILQ_END(head), &(head).tqh_first }

#define _TAILQ_ENTRY(type, qual) \
    struct { \
        qual type *tqe_next; /* next element */ \
        qual type *qual *tqe_prev; /* address of previous next element */ \
    }

```

```

#define TAILQ_ENTRY(type) _TAILQ_ENTRY(struct type,)

/*
 * Tail queue access methods.
 */

#define TAILQ_FIRST(head) ((head)->tqh_first)
#define TAILQ_END(head) (NULL)
#define TAILQ_NEXT(elm, field) ((elm)->field.tqe_next)
#define TAILQ_LAST(head, headname) \
    (((struct headname *) (void *) ((head)->tqh_last))->tqh_last)
#define TAILQ_PREV(elm, headname, field) \
    (((struct headname *) (void *) ((elm)->field.tqe_prev))->tqh_last)
#define TAILQ_EMPTY(head) (TAILQ_FIRST(head) == TAILQ_END(head))

#define TAILQ_FOREACH(var, head, field) \
    for ((var) = ((head)->tqh_first); \
         (var) != TAILQ_END(head); \
         (var) = ((var)->field.tqe_next))

#define TAILQ_FOREACH_SAFE(var, head, field, next) \
    for ((var) = ((head)->tqh_first); \
         (var) != TAILQ_END(head) && \
         ((next) = TAILQ_NEXT(var, field), 1); (var) = (next))

#define TAILQ_FOREACH_REVERSE(var, head, headname, field) \
    for ((var) = TAILQ_LAST((head), headname); \
         (var) != TAILQ_END(head); \
         (var) = TAILQ_PREV((var), headname, field))

#define TAILQ_FOREACH_REVERSE_SAFE(var, head, headname, field, prev) \
    for ((var) = TAILQ_LAST((head), headname); \
         (var) != TAILQ_END(head) && \
         ((prev) = TAILQ_PREV((var), headname, field), 1); (var) = (prev))

/*
 * Tail queue functions.
 */

#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, \
    elm, field) \
    if ((head)->tqh_first && \
        (head)->tqh_first->field.tqe_prev != &(head)->tqh_first) \
        QUEUEDEBUG_ABORT("TAILQ_INSERT_HEAD %p %s:%d", (head), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field) \
    if (*(head)->tqh_last != NULL) \

```

```

    QUEUEDEBUG_ABORT("TAILQ_INSERT_TAIL %p %s:%d", (head), \
        __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_OP(elm, field) \
    if ((elm)->field.tqe_next && \
        (elm)->field.tqe_next->field.tqe_prev != \
        &(elm)->field.tqe_next) \
        QUEUEDEBUG_ABORT("TAILQ_* forw %p %s:%d", (elm), \
            __FILE__, __LINE__); \
    if (*(elm)->field.tqe_prev != (elm)) \
        QUEUEDEBUG_ABORT("TAILQ_* back %p %s:%d", (elm), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field) \
    if ((elm)->field.tqe_next == NULL && \
        (head)->tqh_last != &(elm)->field.tqe_next) \
        QUEUEDEBUG_ABORT("TAILQ_PREREMOVE head %p elm %p %s:%d", \
            (head), (elm), __FILE__, \
            __LINE__);
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field) \
    (elm)->field.tqe_next = (void *)1L; \
    (elm)->field.tqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field)
#define QUEUEDEBUG_TAILQ_OP(elm, field)
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field)
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field)
#endif

#define TAILQ_INIT(head) do { \
    (head)->tqh_first = TAILQ_END(head); \
    (head)->tqh_last = &(head)->tqh_first; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBUG_TAILQ_INSERT_HEAD((head), (elm), field) \
    if (((elm)->field.tqe_next = (head)->tqh_first) != TAILQ_END(head)) \
        (head)->tqh_first->field.tqe_prev = \
            &(elm)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm)->field.tqe_next; \
    (head)->tqh_first = (elm); \
    (elm)->field.tqe_prev = &(head)->tqh_first; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_TAIL(head, \
    elm, field) do { \
    QUEUEDEBUG_TAILQ_INSERT_TAIL((head), (elm), field) \
    (elm)->field.tqe_next = TAILQ_END(head); \

```

```

(elm)->field.tqe_prev = (head)->tqh_last; \
*(head)->tqh_last = (elm); \
(head)->tqh_last = &(elm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_AFTER(head, listelm, elm, field) do { \
    QUEUEDEBBUG_TAILQ_OP((listelm), field) \
    if (((elm)->field.tqe_next = (listelm)->field.tqe_next) != \
        TAILQ_END(head)) \
        (elm)->field.tqe_next->field.tqe_prev = \
            &(elm)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm)->field.tqe_next; \
        (listelm)->field.tqe_next = (elm); \
        (elm)->field.tqe_prev = &(listelm)->field.tqe_next; \
    } while (/*CONSTCOND*/0)

#define TAILQ_INSERT_BEFORE(listelm, elm, field) do { \
    QUEUEDEBBUG_TAILQ_OP((listelm), field) \
    (elm)->field.tqe_prev = (listelm)->field.tqe_prev; \
    (elm)->field.tqe_next = (listelm); \
    *(listelm)->field.tqe_prev = (elm); \
    (listelm)->field.tqe_prev \
    = &(elm)->field.tqe_next; \
    } while (/*CONSTCOND*/0)

#define TAILQ_REMOVE(head, elm, field) do { \
    QUEUEDEBBUG_TAILQ_PREREMOVE((head), (elm), field) \
    QUEUEDEBBUG_TAILQ_OP((elm), field) \
    if (((elm)->field.tqe_next != TAILQ_END(head)) \
        (elm)->field.tqe_next->field.tqe_prev = \
            (elm)->field.tqe_prev; \
    else \
        (head)->tqh_last = (elm)->field.tqe_prev; \
        *(elm)->field.tqe_prev = (elm)->field.tqe_next; \
        QUEUEDEBBUG_TAILQ_POSTREMOVE((elm), field); \
    } while (/*CONSTCOND*/0)

#define TAILQ_REPLACE(head, elm, elm2, field) do { \
    if (((elm2)->field.tqe_next = (elm)->field.tqe_next) != \
        TAILQ_END(head)) \
        (elm2)->field.tqe_next->field.tqe_prev = \
            &(elm2)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm2)->field.tqe_next; \
        (elm2)->field.tqe_prev = (elm)->field.tqe_prev; \
        *(elm2)->field.tqe_prev = (elm2); \
    QUEUEDEBBUG_TAILQ_POSTREMOVE((elm),

```

```

    field); \
} while (/*CONSTCOND*/0)

#define TAILQ_CONCAT(head1, head2, field) do { \
    if (!TAILQ_EMPTY(head2)) { \
        *(head1)->tqh_last = (head2)->tqh_first; \
        (head2)->tqh_first->field.tqe_prev = (head1)->tqh_last; \
        (head1)->tqh_last = (head2)->tqh_last; \
        TAILQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)

/*
 * Singly-linked Tail queue declarations.
 */
#define STAILQ_HEAD(name, type) \
struct name { \
    struct type *stqh_first; /* first element */ \
    struct type **stqh_last; /* addr of last next element */ \
}

#define STAILQ_HEAD_INITIALIZER(head) \
{ NULL, &(head).stqh_first }

#define STAILQ_ENTRY(type) \
struct { \
    struct type *stqe_next; /* next element */ \
}

/*
 * Singly-linked Tail queue access methods.
 */
#define STAILQ_FIRST(head) ((head)->stqh_first)
#define STAILQ_END(head) NULL
#define STAILQ_NEXT(elm, field) ((elm)->field.stqe_next)
#define STAILQ_EMPTY(head) (STAILQ_FIRST(head) \
    == STAILQ_END(head))

/*
 * Singly-linked Tail queue functions.
 */
#define STAILQ_INIT(head) do { \
    (head)->stqh_first = NULL; \
    (head)->stqh_last = &(head)->stqh_first; \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_HEAD(head, elm, field) do { \
    if (((elm)->field.stqe_next = (head)->stqh_first) == NULL) \

```

```

(head)->stqh_last = &(elm)->field.stqe_next; \
(head)->stqh_first = (elm); \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_TAIL(head, elm, field) do { \
    (elm)->field.stqe_next = NULL; \
    *(head)->stqh_last = (elm); \
    (head)->stqh_last = &(elm)->field.stqe_next; \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_AFTER(head, listelm, elm, field) do { \
    if (((elm)->field.stqe_next = (listelm)->field.stqe_next) == NULL)\
        (head)->stqh_last = &(elm)->field.stqe_next; \
    (listelm)->field.stqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define STAILQ_REMOVE_HEAD(head, field) do { \
    if (((head)->stqh_first = (head)->stqh_first->field.stqe_next) \
        == NULL) \
        (head)->stqh_last = &(head)->stqh_first; \
} while (/*CONSTCOND*/0)

#define STAILQ_REMOVE(head, elm, type, field) do { \
    if ((head)->stqh_first == (elm)) { \
        STAILQ_REMOVE_HEAD((head), field); \
    } else { \
        struct type *curelm = (head)->stqh_first; \
        while (curelm->field.stqe_next != (elm)) \
            curelm = curelm->field.stqe_next; \
        if ((curelm->field.stqe_next = \
            curelm->field.stqe_next->field.stqe_next) == NULL) \
            (head)->stqh_last = &(curelm)->field.stqe_next; \
    } \
} while (/*CONSTCOND*/0)

#define STAILQ_FOREACH(var, head, field) \
    for ((var) = ((head)->stqh_first); \
        (var); \
        (var) = ((var)->field.stqe_next))

#define STAILQ_FOREACH_SAFE(var, head, field, tvar) \
    for ((var) = STAILQ_FIRST((head)); \
        (var) && ((tvar) = STAILQ_NEXT((var), field), 1); \
        (var) = (tvar))

#define STAILQ_CONCAT(head1, head2) do { \
    if (!STAILQ_EMPTY((head2))) { \
        *(head1)->stqh_last

```

```

= (head2)->stqh_first; \
(head1)->stqh_last = (head2)->stqh_last; \
STAILQ_INIT((head2)); \
} \
} while (/*CONSTCOND*/0)

#define STAILQ_LAST(head, type, field) \
(STAILQ_EMPTY((head)) ? \
NULL : \
((struct type *) (void *) \
((char *)((head)->stqh_last) - offsetof(struct type, field))))

#ifndef _KERNEL
/*
 * Circular queue definitions. Do not use. We still keep the macros
 * for compatibility but because of pointer aliasing issues their use
 * is discouraged!
 */

/*
 * __launder_type(): We use this ugly hack to work around the the compiler
 * noticing that two types may not alias each other and elide tests in code.
 * We hit this in the CIRCLEQ macros when comparing 'struct name *' and
 * 'struct type *' (see CIRCLEQ_HEAD()). Modern compilers (such as GCC
 * 4.8) declare these comparisons as always false, causing the code to
 * not run as designed.
 *
 * This hack is only to be used for comparisons and thus can be fully const.
 * Do not use for assignment.
 *
 * If we ever choose to change the ABI of the CIRCLEQ macros, we could fix
 * this by changing the head/tail sentinel values, but see the note above
 * this one.
 */
static __inline const void * __launder_type(const void *);
static __inline const void *
__launder_type(const void *__x)
{
    __asm __volatile("" : "+r" (__x));
    return __x;
}

#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field) \
if ((head)->cqh_first != CIRCLEQ_ENDC(head) && \
(head)->cqh_first->field.cqe_prev != CIRCLEQ_ENDC(head)) \
QUEUEDEBUG_ABORT("CIRCLEQ head forw %p %s:%d", (head), \

```



```

    __FILE__, __LINE__); \
if ((head)->cqh_last != CIRCLEQ_ENDC(head) && \
    (head)->cqh_last->field.cqe_next != CIRCLEQ_ENDC(head)) \
    QUEUEDEBDEBUG_ABORT("CIRCLEQ head back %p %s:%d", (head), \
        __FILE__, __LINE__);
#define QUEUEDEBDEBUG_CIRCLEQ_ELM(head, elm, field) \
if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) { \
    if ((head)->cqh_last != (elm)) \
        QUEUEDEBDEBUG_ABORT("CIRCLEQ \
elm last %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} else { \
    if ((elm)->field.cqe_next->field.cqe_prev != (elm)) \
        QUEUEDEBDEBUG_ABORT("CIRCLEQ elm forw %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} \
if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) { \
    if ((head)->cqh_first != (elm)) \
        QUEUEDEBDEBUG_ABORT("CIRCLEQ elm first %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} else { \
    if ((elm)->field.cqe_prev->field.cqe_next != (elm)) \
        QUEUEDEBDEBUG_ABORT("CIRCLEQ elm prev %p %s:%d", \
            (elm), __FILE__, __LINE__); \
}
#define QUEUEDEBDEBUG_CIRCLEQ_POSTREMOVE(elm, field) \
(elm)->field.cqe_next = (void *)1L; \
(elm)->field.cqe_prev = (void *)1L;
#else
#define QUEUEDEBDEBUG_CIRCLEQ_HEAD(head, field)
#define QUEUEDEBDEBUG_CIRCLEQ_ELM(head, elm, field)
#define QUEUEDEBDEBUG_CIRCLEQ_POSTREMOVE(elm, field)
#endif

#define CIRCLEQ_HEAD(name, type) \
struct name { \
    struct type *cqh_first; /* \
first element */ \
    struct type *cqh_last; /* last element */ \
}

#define CIRCLEQ_HEAD_INITIALIZER(head) \
{ CIRCLEQ_END(&head), CIRCLEQ_END(&head) }

#define CIRCLEQ_ENTRY(type) \
struct { \
    struct type *cqe_next; /* next element */ \
    struct type *cqe_prev; /* previous element */ \
}

```

```

}

/*
 * Circular queue functions.
 */

#define CIRCLEQ_INIT(head) do { \
    (head)->cqh_first = CIRCLEQ_END(head); \
    (head)->cqh_last = CIRCLEQ_END(head); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_AFTER(head, listelm, elm, field) do { \
    QUEUEDEBDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBDEBUG_CIRCLEQ_ELM((head), (listelm), field) \
    (elm)->field.cqe_next = (listelm)->field.cqe_next; \
    (elm)->field.cqe_prev = (listelm); \
    if ((listelm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm); \
    else \
        (listelm)->field.cqe_next->field.cqe_prev = (elm); \
    (listelm)->field.cqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_BEFORE(head, \
    listelm, elm, field) do { \
    QUEUEDEBDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBDEBUG_CIRCLEQ_ELM((head), (listelm), field) \
    (elm)->field.cqe_next = (listelm); \
    (elm)->field.cqe_prev = (listelm)->field.cqe_prev; \
    if ((listelm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm); \
    else \
        (listelm)->field.cqe_prev->field.cqe_next = (elm); \
    (listelm)->field.cqe_prev = (elm); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBDEBUG_CIRCLEQ_HEAD((head), field) \
    (elm)->field.cqe_next = (head)->cqh_first; \
    (elm)->field.cqe_prev = CIRCLEQ_END(head); \
    if ((head)->cqh_last == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm); \
    else \
        (head)->cqh_first->field.cqe_prev = (elm); \
    (head)->cqh_first = (elm); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_TAIL(head, elm, field) do { \
    QUEUEDEBDEBUG_CIRCLEQ_HEAD((head), field) \

```

```

(elm)->field.cqe_next
= CIRCLEQ_END(head); \
(elm)->field.cqe_prev = (head)->cqh_last; \
if ((head)->cqh_first == CIRCLEQ_ENDC(head)) \
    (head)->cqh_first = (elm); \
else \
    (head)->cqh_last->field.cqe_next = (elm); \
    (head)->cqh_last = (elm); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_REMOVE(head, elm, field) do { \
    QUEUEDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBUG_CIRCLEQ_ELM((head), (elm), field) \
    if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm)->field.cqe_prev; \
    else \
        (elm)->field.cqe_next->field.cqe_prev = \
            (elm)->field.cqe_prev; \
    if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm)->field.cqe_next; \
    else \
        (elm)->field.cqe_prev->field.cqe_next = \
            (elm)->field.cqe_next; \
    QUEUEDEBUG_CIRCLEQ_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_FOREACH(var, head, field) \
    for ((var) = ((head)->cqh_first); \
        (var) != CIRCLEQ_ENDC(head); \
        (var) \
        = ((var)->field.cqe_next))

#define CIRCLEQ_FOREACH_REVERSE(var, head, field) \
    for ((var) = ((head)->cqh_last); \
        (var) != CIRCLEQ_ENDC(head); \
        (var) = ((var)->field.cqe_prev))

/*
 * Circular queue access methods.
 */

#define CIRCLEQ_FIRST(head) ((head)->cqh_first)
#define CIRCLEQ_LAST(head) ((head)->cqh_last)
/* For comparisons */
#define CIRCLEQ_ENDC(head) (__launter_type(head))
/* For assignments */
#define CIRCLEQ_END(head) ((void *)(head))
#define CIRCLEQ_NEXT(elm, field) ((elm)->field.cqe_next)
#define CIRCLEQ_PREV(elm, field) ((elm)->field.cqe_prev)

```

```

#define CIRCLEQ_EMPTY(head) \
    (CIRCLEQ_FIRST(head) == CIRCLEQ_ENDC(head))

#define CIRCLEQ_LOOP_NEXT(head, elm, field) \
    (((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
     ? ((head)->cqh_first) \
     : (elm->field.cqe_next))
#define CIRCLEQ_LOOP_PREV(head, elm, field) \
    (((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
     ? ((head)->cqh_last) \
     : \
     (elm->field.cqe_prev))
#endif /* !_KERNEL */

#endif /* !_SYS_QUEUE_H_ */
# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver

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

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

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

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237

```

```

b316bd503ae7dceed6ca284ccd5a74d
flake8-copyright-0.2.4.tar.gz
"

#ifndef _SYS_CDEFS_H_
#define _SYS_CDEFS_H_

#warning usage of non-standard #include <sys/cdefs.h> is deprecated

#undef __P
#undef __PMT

#define __P(args) args
#define __PMT(args) args

#define __CONCAT(x,y) x ## y
#define __STRING(x) #x

#ifdef __cplusplus
# define __BEGIN_DECLS extern "C" {
# define __END_DECLS }
#else
# define __BEGIN_DECLS
# define __END_DECLS
#endif

#if defined(__GNUC__) && !defined(__cplusplus)
# define __THROW __attribute__((__nothrow__))
# define __NTH(fct) __attribute__((__nothrow__)) fct
#else
# define __THROW
# define __NTH(fct) fct
#endif

#endif /* _SYS_CDEFS_H_ */
# Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"
url="https://github.com/onur/cargo-license"
arch="all"
license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

prepare() {
default_prepare

```

```

cargo fetch --target="$CTARGET" --locked
}

build() {
    cargo auditable build --release --frozen
}

check() {
    cargo test --frozen
}

package() {
    install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5 cargo-license-0.6.1.tar.gz
"

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

```

```

#ifndef _SYS_TREE_H_
#define _SYS_TREE_H_

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

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

/*
 * This file defines data structures for different types of trees:
 * splay trees and red-black trees.
 *
 * A splay tree is a self-organizing
 * data structure. Every operation
 * on the tree causes a splay to happen. The splay moves the requested
 * node to the root of the tree and partly rebalances it.
 *
 * This has the benefit that request locality causes faster lookups as
 * the requested nodes move to the top of the tree. On the other hand,
 * every lookup causes memory writes.
 *
 * The Balance Theorem bounds the total access time for m operations
 * and n inserts on an initially empty tree as  $O((m + n) \lg n)$ . The
 * amortized cost for a sequence of m accesses to a splay tree is  $O(\lg n)$ ;
 *
 * A red-black tree is a binary search tree with the node color as an
 * extra attribute. It fulfills a set of conditions:
 * - every search path from the root to a leaf consists of the
 *   same number of black nodes,
 * - each red node (except for the root) has a black parent,
 * - each leaf node is black.
 *
 * Every operation on a red-black tree is bounded as  $O(\lg n)$ .
 * The maximum height of a red-black tree is  $2 \lg (n+1)$ .
 */

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

```

```

}

#define SPLAY_INITIALIZER(root) \
{ NULL }

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

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

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

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

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

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

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

#define SPLAY_ASSEMBLE(head, node, left, right, field) do { \
    SPLAY_RIGHT(left, field) = SPLAY_LEFT((head)->sph_root, field); \

```



```

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

```

```

/* Generates prototypes and inline functions */

```

```

#define

```

```

    SPLAY_PROTOTYPE(name, type, field, cmp) \
void name##_SPLAY(struct name *, struct type *); \
void name##_SPLAY_MINMAX(struct name *, int); \
struct type *name##_SPLAY_INSERT(struct name *, struct type *); \
struct type *name##_SPLAY_REMOVE(struct name *, struct type *); \
\

```

```

/* Finds the node with the same key as elm */ \

```

```

static __inline struct type * \
name##_SPLAY_FIND(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) \
        return(NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) \
        return (head->sph_root); \
    return (NULL); \
} \
\

```

```

static __inline _sys_tree_h_unused struct type * \
name##_SPLAY_NEXT(struct name *head, struct type *elm) \
{ \
    name##_SPLAY(head, elm); \
    if (SPLAY_RIGHT(elm, field) != NULL) { \
        elm = SPLAY_RIGHT(elm, field); \
        while (SPLAY_LEFT(elm, field) != NULL) { \
            elm = SPLAY_LEFT(elm, field); \
        } \
    } \
    else \
        elm = NULL; \
    return (elm); \
} \
\

```

```

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

```

```

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

```

```

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

```

```

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

#define SPLAY_NEGINF -1
#define SPLAY_INF 1

#define SPLAY_INSERT(name, x, y) name##_SPLAY_INSERT(x, y)
#define SPLAY_REMOVE(name, x, y) name##_SPLAY_REMOVE(x, y)
#define SPLAY_FIND(name, x, y) name##_SPLAY_FIND(x, y)
#define SPLAY_NEXT(name, x, y) name##_SPLAY_NEXT(x, y)
#define SPLAY_MIN(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_NEGINF))
#define SPLAY_MAX(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_INF))

#define SPLAY_FOREACH(x, name, head) \
    for ((x) = SPLAY_MIN(name, head); \
        (x) != NULL; \
        (x) = SPLAY_NEXT(name, head, x))

/* Macros that define a red-black tree */
#define RB_HEAD(name, type) \
    struct name {

```

```

struct type *rbh_root; /* root of the tree */ \
}

#define RB_INITIALIZER(root) \
{
    NULL }

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

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

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

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

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

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

#define RB_ROTATE_LEFT(head, elm, tmp, field) do { \
    (tmp) = RB_RIGHT(elm, field); \
    if ((RB_RIGHT(elm, field) = RB_LEFT(tmp, field)) != NULL) { \
        RB_PARENT(RB_LEFT(tmp, field), field) = (elm); \

```

```

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

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

/* Generates prototypes and inline functions */
#define RB_PROTOTYPE(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp,)
#define RB_PROTOTYPE_STATIC(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_PROTOTYPE_INTERNAL(name, type, field, cmp, attr) \
    attr void name##_RB_INSERT_COLOR(struct name *, struct \
    type *); \
    attr void name##_RB_REMOVE_COLOR(struct name *, struct type *, struct type *); \
    attr struct type *name##_RB_REMOVE(struct name *, struct type *); \
    attr struct type *name##_RB_INSERT(struct name *, struct type *); \

```

```

attr struct type *name##_RB_FIND(struct name *, struct type *); \
attr struct type *name##_RB_NFIND(struct name *, struct type *); \
attr struct type *name##_RB_NEXT(struct type *); \
attr struct type *name##_RB_PREV(struct type *); \
attr struct type *name##_RB_MINMAX(struct name *, int); \
\

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

```

```

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

```



```

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

```

```

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

```

```

parent = RB_PARENT(elm, field); \
color = RB_COLOR(elm, field); \
if (child) \
    RB_PARENT(child, field) = parent; \
if (parent) { \
    if (RB_LEFT(parent, field) == elm) \
        RB_LEFT(parent, field) = child; \
    else \
        RB_RIGHT(parent,
field) = child; \
    RB_AUGMENT(parent); \
} else \
    RB_ROOT(head) = child; \
color: \
if (color == RB_BLACK) \
    name##_RB_REMOVE_COLOR(head, parent, child); \
return (old); \
} \
\
/* Inserts a node into the RB tree */ \
attr struct type * \
name##_RB_INSERT(struct name *head, struct type *elm) \
{ \
    struct type *tmp; \
    struct type *parent = NULL; \
    int comp = 0; \
    tmp = RB_ROOT(head); \
    while (tmp) { \
        parent = tmp; \
        comp = (cmp)(elm, parent); \
        if (comp < 0) \
            tmp = RB_LEFT(tmp, field); \
        else if (comp > 0) \
            tmp = RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    RB_SET(elm, parent, field); \
    if (parent != NULL) { \
        if (comp < 0) \
            RB_LEFT(parent, field) = elm; \
        else \
            RB_RIGHT(parent, field) = elm; \
        RB_AUGMENT(parent); \
    } else \
        RB_ROOT(head) = elm; \
    name##_RB_INSERT_COLOR(head,

```

```

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

```

```

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

```

```

if (val < 0)    \
    tmp = RB_LEFT(tmp, field); \
else          \
    tmp = RB_RIGHT(tmp, field); \
}             \
return (parent); \
}

#define RB_NEGINF -1
#define RB_INF 1

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

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

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

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

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

#define RB_FOREACH_REVERSE_FROM(x, name, y) \
for ((x) = (y); \
((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_REVERSE_SAFE(x, name,

```

```

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

#endif /* _SYS_TREE_H_ */
pkgname=linux-headers
pkgver=6.14.2 # Follow the latest Linux stable
_kernver=${pkgver%.*}
pkgrel=0
pkgdesc="Linux system headers"
url="https://kernel.org/"
arch="all" # Headers are architecture specific
license="GPL-2.0-only"
makedepends="perl"
options="!check !dbg !strip !tracedeps !archcheck"
source="https://kernel.org/pub/linux/kernel/v6.x/linux-${_kernver}.tar.xz
revert-broken-uapi.patch
0003-remove-inclusion-of-sysinfo.h-in-kernel.h.patch

```

1.9 goautoneg 0.0.0-20191010083416-a7dc8b61c822

1.9.1 Available under license :

Copyright (c) 2011, Open Knowledge Foundation Ltd.
All rights reserved.

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

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

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

Neither the name of the Open Knowledge Foundation Ltd. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

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

1.10 libmd 1.1.0-r0

1.10.1 Available under license :

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

Files:

*

Copyright:

Copyright 2009, 2011, 2016 Guillem Jover <guillem@hadrons.org>

License: BSD-3-clause

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

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

.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

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

Files:

include/sha2.h

src/sha2.c

Copyright:

Copyright 2000-2001, Aaron D. Gifford

All rights reserved.

License: BSD-3-clause-Aaron-D-Gifford

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

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

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

Files:

include/rmd160.h

src/rmd160.c

Copyright:

Copyright 2001 Markus Friedl. All rights reserved.

License: BSD-2-clause

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

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

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

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

Files:

src/md2.c

Copyright:

Copyright (c) 2001 The NetBSD Foundation, Inc.

All rights reserved.

.

This code is derived from software contributed to The NetBSD Foundation
by Andrew Brown.

License: BSD-2-clause-NetBSD

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

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

.

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

Files:

man/rmd160.3

man/sha1.3

man/sha2.3

Copyright:

Copyright 1997, 2003, 2004 Todd C. Miller <Todd.Miller@courtesan.com>

License: ISC

Permission to use, copy, modify, and distribute this software for any

purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

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

Files:

man/mdX.3

src/helper.c

Copyright:

Poul-Henning Kamp <phk@login.dkuug.dk>

License: Beerware

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

<phk@login.dkuug.dk> wrote this file. As long as you retain
this notice you

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

Files:

include/md4.h

src/md4.c

Copyright:

Colin Plumb

Todd C. Miller

License: public-domain-md4

This code implements the MD4 message-digest algorithm.

The algorithm is due to Ron Rivest. This code was
written by Colin Plumb in 1993, no copyright is claimed.

This code is in the public domain; do with it what you wish.

Todd C. Miller modified the MD5 code to do MD4 based on RFC 1186.

Files:

include/md5.h

src/md5.c

Copyright:

Colin Plumb

License: public-domain-md5

This code implements the MD5 message-digest algorithm.

The algorithm is due to Ron Rivest. This code was
written by Colin Plumb in 1993, no copyright is claimed.

This code is in the public domain; do with it what you wish.

Files:

include/sha1.h
src/sha1.c
Copyright:
Steve Reid <steve@edmweb.com>
License: public-domain-sha1
100% Public
Domain

1.11 pax-utils 1.3.8-r1

1.11.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

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

Preamble

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

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

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

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the

source code. And you must show them these terms so they know their rights.

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

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

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

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

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

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

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

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you

conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

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

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

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

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

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

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

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

collective works based on the Program.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES

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

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1.12 libcap 2.76-r0

1.12.1 Available under license :

```
/* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only */
```

Unless otherwise **explicitly** stated, the following text describes the licensed conditions under which the contents of this libcap/cap release may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

BSD 3-clause:

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

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

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

GPL v2.0:

ALTERNATIVELY, this product may be distributed under the terms of the GNU General Public License (v2.0 - see below), in which case the provisions of the GNU GPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential conflict between the GNU GPL and the restrictions contained in a BSD-style copyright.)

Full text of gpl-2.0.txt:

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

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

Preamble

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

This
General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

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

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

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

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

Also, for each author's protection and ours, we want to make certain

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

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

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

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

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

Activities

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

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

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

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

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

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

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

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

3. You may copy and distribute the Program (or a work based on it,

under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

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

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

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

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

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

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

5. You are not required to accept this License, since you have not

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

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

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

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

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

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

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

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

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

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software

Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

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

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL

ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

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

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

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

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

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

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

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

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

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

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

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

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

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

/* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only */

Unless otherwise **explicitly** stated, the following text describes the licensed conditions under which the contents of this libcap/psx release may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

BSD 3-clause:

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

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce all prior and current

copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.

3.

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

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

GPL v2.0:

ALTERNATIVELY, this product may be distributed under the terms of the GNU General Public License (v2.0 - see below), in which case the provisions of the GNU GPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential conflict between the GNU GPL and the restrictions contained in a BSD-style copyright.)

Full text of gpl-2.0.txt:

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

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

Preamble

The licenses for most software are designed to take away your

freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This

General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

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

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

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

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

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

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

The precise terms and conditions for copying, distribution and

modification follow.

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

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

Activities

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

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

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

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

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

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

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

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

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

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

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

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

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

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

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

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

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

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

You are not responsible for enforcing compliance by third parties to this License.

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

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

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

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

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

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will

be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

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

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software

Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

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

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL

ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest

possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

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

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

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

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

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

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

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

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

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

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

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
`Gnomovision' (which
```

makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

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

/* SPDX-License-Identifier: BSD-3-Clause OR GPL-2.0-only */

Unless otherwise *explicitly* stated, the following text describes the licensed conditions under which the contents of this libcap release may be used and distributed.

The licensed conditions are one or the other of these two Licenses:

- BSD 3-clause
- GPL v2.0

BSD 3-clause:

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

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

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

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

GPL v2.0:

ALTERNATIVELY, this product may be distributed under the terms of the GNU General Public License (v2.0 - see below), in which case the provisions of the GNU GPL are required INSTEAD OF the above restrictions.

Full text of gpl-2.0.txt:

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

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

Preamble

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

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

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

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

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

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

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

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

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

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

Activities other than copying, distribution and modification are not

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

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

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

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

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you

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

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

In addition,

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

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

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

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

If

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

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

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

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

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

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

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

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

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

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

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

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

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

preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

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

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING

ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

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

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

This program is distributed in the hope that it will be useful,

but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

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

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

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

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

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

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

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

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

This General Public

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

/* SPDX-License-Identifier: BSD-3-Clause OR LGPL-2.0-or-later */

Unless otherwise *explicitly* stated the following text describes the licensed conditions under which the contents of this module release may be distributed:

Redistribution and use in source and binary forms of this module, with

or without modification, are permitted provided that the following conditions are met:

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

ALTERNATIVELY, this product may be distributed under the terms of the GNU Library General Public License, in which case the provisions of the GNU LGPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential conflict between the GNU LGPL and the restrictions contained in a BSD-style copyright.)

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

1.13 perks 1.0.1

1.13.1 Available under license :

Copyright (C) 2013 Blake Mizerany

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

permit persons to whom the Software is furnished to do so, subject to the following conditions:

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

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

1.14 github.com/confluentinc/confluent-kafka-go/v2 2.12.0

1.14.1 Available under license :

LICENSE

librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2022, Magnus Edenhill
2023, Confluent Inc.
All rights reserved.

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

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

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.cjson

For cJSON.c and cJSON.h:

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

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

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

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

LICENSE.crc32c

For src/crc32c.c copied (with modifications) from
<http://stackoverflow.com/a/17646775/1821055>

/* crc32c.c -- compute CRC-32C using the Intel crc32 instruction
* Copyright (C) 2013 Mark Adler
* Version 1.1 1 Aug 2013 Mark Adler
*/

/*
This software is provided 'as-is',
without any express or implied
warranty. In no event will the author be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,

including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Mark Adler
madler@alumni.caltech.edu

*/

LICENSE.fnv1a

parts of src/rdfnv1a.c: http://www.isthe.com/chongo/src/fnv/hash_32a.c

Please
do not copyright this code. This code is in the public domain.

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

By:
chongo <Landon Curt Noll> ^oo^
<http://www.isthe.com/chongo/>

Share and Enjoy! :-)

LICENSE.hdrhistogram

This license covers src/rdhdrhistogram.c which is a C port of
Coda Hale's Golang HdrHistogram <https://github.com/codahale/hdrhistogram>
at revision 3a0bb77429bd3a61596f5e8a3172445844342120

The MIT License

(MIT)

Copyright (c) 2014 Coda Hale

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

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

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

LICENSE.lz4

src/rdxxhash.[ch] src/lz4*.[ch]: git@github.com:lz4/lz4.git 5ff839680134437dbf4678f3d0c7b371d84f4964

LZ4 Library

Copyright (c) 2011-2020, Yann Collet

All rights reserved.

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

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

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

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

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

LICENSE.murmur2

parts of src/rdmurmur2.c: git@github.com:abandoned/murmur2.git

MurMurHash2 Library

//-----
// MurmurHash2 was written by Austin Appleby, and is placed in the public
// domain. The author hereby disclaims copyright to this source code.

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

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

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

LICENSE.nanopb

For
files in src/nanopb :
<https://github.com/nanopb/nanopb/blob/8ef41e0ebd45daaf19459a011f67e66224b247cd/LICENSE.txt>

Copyright (c) 2011 Petteri Aimonen <jpa at nanopb.mail.kapsi.fi>

This software is provided 'as-is', without any express or
implied warranty. In no event will the authors be held liable
for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

LICENSE.opentelemetry

For files in src/opentelemetry: <https://github.com/open-telemetry/opentelemetry-proto/blob/81a296f9dba23e32d77f46d58c8ea4244a2157a6/LICENSE>

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

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

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

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

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

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

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

LICENSE.pycrc

The following license applies to the files rdcrc32.c and rdcrc32.h which have been generated by the pycrc tool.

=====

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

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

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

LICENSE.queue

For

sys/queue.h:

- * Copyright (c) 1991, 1993
- * The Regents of the University of California. All rights reserved.
- *
- * Redistribution and use in source and binary forms, with or without
- * modification, are permitted provided that the following conditions
- * are met:
- * 1. Redistributions of source code must retain the above copyright
- * notice, this list of conditions and the following disclaimer.
- * 2. Redistributions in binary form must reproduce the above copyright
- * notice, this list of conditions and the following disclaimer in the
- * documentation and/or other materials provided with the distribution.
- * 4. Neither the name of the University nor the names of its contributors
- * may be used to endorse or promote products derived from this software
- * without specific prior written permission.
- *
- * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
- * ANY EXPRESS OR
- * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
- * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
- * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

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

*

* @(#)queue.h 8.5 (Berkeley) 8/20/94

* \$FreeBSD\$

LICENSE.regexp

regexp.c and regexp.h from <https://github.com/ccxvii/minilibs> sha 875c33568b5a4aa4fb3dd0c52ea98f7f0e5ca684

"

These libraries are in the public domain (or the equivalent where that is not possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.

"

LICENSE.snappy

LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h #
originally retrieved from <http://github.com/andikleen/snappy-c> #
git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219 #
#####

The snappy-c code is under the same license as the original snappy source

Copyright 2011 Intel Corporation All Rights Reserved.

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

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

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

* Neither the name of Intel Corporation nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

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

LICENSE.tinythread

From <https://github.com/tinythread/tinythread/README.txt> c57166cd510ffb5022dd5f127489b131b61441b9

License

Copyright (c) 2012 Marcus Geelnard
2013-2014 Evan Nemerson

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

LICENSE.wingetopt

For the files wingetopt.c wingetopt.h downloaded from <https://github.com/alex85k/wingetopt>

```

/*
* Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>
*
* Permission to use, copy, modify, and distribute this software for any
* purpose with or without fee is hereby granted, provided that the above
* copyright notice and this permission notice appear in all copies.
*
* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR
* ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Sponsored in part by the Defense Advanced Research Projects
* Agency (DARPA) and Air Force Research Laboratory, Air Force
* Materiel Command, USAF, under agreement number F39502-99-1-0512.
*/
/*_
* Copyright (c) 2000 The NetBSD Foundation, Inc.
* All rights reserved.
*
* This code is derived from software contributed to The NetBSD Foundation
* by Dieter Baron and Thomas Klausner.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice,
* this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
* ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
* TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE

```

* POSSIBILITY OF SUCH DAMAGE.

*/

Apache License

Version 2.0, January 2004

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

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

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

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations,
or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

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

that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work

or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed

with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.15 zlib 1.3.1-r2

1.15.1 Available under license :

Copyright notice:

(C) 1995-2022 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly	Mark Adler
jloup@gzip.org	madler@alumni.caltech.edu

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.16 scanelf 1.3.8-r1

1.16.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

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

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

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

Preamble

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1.17 alpine-base 3.22.2-r0

1.17.1 Available under license :

MIT License

Copyright 2019-2021 Caleb Porzio and contributors

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

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

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

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT

1.18 otel 1.38.0

1.18.1 Available under license :

Apache License

Version 2.0, January 2004

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

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

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

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

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

designated in writing by the copyright owner as "Not a Contribution."

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright 2009 The Go Authors.

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

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

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

1.19 alpine-keys 2.5-r0

1.19.1 Available under license :

MIT License

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

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

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

Maintainer: Natanael Copa <ncopa@alpinelinux.org>

pkgname=alpine-keys

pkgver=2.5

pkgrel=0

pkgdesc="Public keys for Alpine Linux packages"

url="https://alpinelinux.org"

we install arch specific keys to /etc so we cannot do arch=noarch

arch="all"

license="MIT"

replaces="alpine-base"

1.20 otel-trace 1.38.0

1.20.1 Available under license :

Apache License

Version 2.0, January 2004

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

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

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

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

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

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright 2009 The Go Authors.

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

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

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

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

1.21 logrus 1.9.3

1.21.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Simon Eskildsen

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

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

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

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

1.22 go.opentelemetry.io/auto/sdk 1.2.1

1.22.1 Available under license :

MIT License

Copyright (c) 2019 Montgomery Edwards and Faye Amacker

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

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

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

Apache License

Version 2.0, January 2004

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

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

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

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

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

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

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

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The MIT License (MIT)

Copyright (c) 2014 Cenk Alt

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

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

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

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015, Dave Cheney <dave@cheney.net>

All rights reserved.

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

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

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

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

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

MIT License

Copyright (c) 2019-present Faye Amacker

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

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

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

Apache License

Version 2.0, January 2004

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

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

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

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

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

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

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where

such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form

of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted

for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or

claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to
in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Copyright 2011-2016 Canonical Ltd.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Data model artifacts for Prometheus.

Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

MIT License

Copyright (c) 2016 json-iterator

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

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

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

Copyright (c) 2011, Open Knowledge Foundation Ltd.

All rights reserved.

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

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

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

Neither the name of the Open Knowledge Foundation Ltd. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

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

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014 gRPC authors.

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2016 Caleb Spare

MIT License

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

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

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

THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

procfs provides functions to retrieve system, kernel and process metrics from the pseudo-filesystem proc.

Copyright 2014-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

This project is covered by two different licenses: MIT and Apache.

MIT License

The following files were ported to Go from C files of libyaml, and thus
are still covered by their original MIT license, with the additional
copyright starting in 2011 when the project was ported over:

apic.go emitterc.go parserc.go readerc.go scannerc.go
writerc.go yamlh.go yamlprivateh.go

Copyright (c) 2006-2010 Kirill Simonov

Copyright (c) 2006-2011 Kirill Simonov

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

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

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

Apache License

All the remaining project files are covered by the Apache license:

Copyright (c) 2011-2019 Canonical Ltd

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
Copyright (c) 2017-2021 Uber Technologies, Inc.

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

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

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

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

of such entity.

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

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

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

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

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

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

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright 2009 The Go Authors.

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

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

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

Common libraries shared by Prometheus Go components.

Copyright 2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

Copyright (C) 2014-2019, Matt Butcher and Matt Farina

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

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

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2017 Nathan Sweet

Copyright (c) 2018, 2019 Cloudflare

Copyright (c) 2019 Authors of Cilium

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

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

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

Copyright (c) 2020 The Go Authors. All rights reserved.

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

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

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

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

/* This is auto-generated file. See bpf_doc.py for details. */

/* Forward declarations of BPF structs */

```
struct bpf_fib_lookup;
struct bpf_sk_lookup;
struct bpf_perf_event_data;
struct bpf_perf_event_value;
struct bpf_pidns_info;
struct bpf_redir_neigh;
struct bpf_sock;
struct bpf_sock_addr;
struct bpf_sock_ops;
struct bpf_sock_tuple;
struct bpf_spin_lock;
struct bpf_sysctl;
struct bpf_tcp_sock;
struct bpf_tunnel_key;
struct bpf_xfrm_state;
struct linux_binprm;
struct pt_regs;
struct sk_reuseport_md;
struct sockaddr;
struct tcphdr;
struct seq_file;
struct tcp6_sock;
struct tcp_sock;
struct tcp_timewait_sock;
struct tcp_request_sock;
struct udp6_sock;
struct unix_sock;
struct task_struct;
struct cgroup;
struct __sk_buff;
struct sk_msg_md;
struct xdp_md;
struct path;
struct btf_ptr;
struct inode;
struct socket;
```

```

struct file;
struct bpf_timer;
struct mptcp_sock;
struct bpf_dynptr;
struct iphdr;
struct ipv6hdr;

/*
 * bpf_map_lookup_elem
 *
 * Perform a lookup in *map* for an entry associated
 * to *key*.
 *
 * Returns
 * Map value associated to *key*, or **NULL** if no entry was
 * found.
 */
static void *(* const bpf_map_lookup_elem)(void *map, const void *key) = (void *) 1;

/*
 * bpf_map_update_elem
 *
 * Add or update the value of the entry associated to *key* in
 * *map* with *value*. *flags* is one of:
 *
 * **BPF_NOEXIST**
 * The entry for *key* must not exist in the map.
 * **BPF_EXIST**
 * The entry for *key* must already exist in the map.
 * **BPF_ANY**
 * No condition on the existence of the entry for *key*.
 *
 * Flag value **BPF_NOEXIST** cannot be used for maps of types
 * **BPF_MAP_TYPE_ARRAY** or **BPF_MAP_TYPE_PERCPU_ARRAY** (all
 * elements always exist), the helper would return an error.
 *
 * Returns
 * 0 on success, or a negative error in case of failure.
 */
static long (* const bpf_map_update_elem)(void *map, const void *key, const void *value, __u64 flags) = (void *)
2;

/*
 * bpf_map_delete_elem
 *
 * Delete entry with *key* from
 * *map*.
 *

```

```

* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_map_delete_elem)(void *map, const void *key) = (void *) 3;

/*
* bpf_probe_read
*
* For tracing programs, safely attempt to read *size* bytes from
* kernel space address *unsafe_ptr* and store the data in *dst*.
*
* Generally, use **bpf_probe_read_user**\ () or
* **bpf_probe_read_kernel**\ () instead.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_probe_read)(void *dst, __u32 size, const void *unsafe_ptr) = (void *) 4;

/*
* bpf_ktime_get_ns
*
* Return the time elapsed since system boot, in nanoseconds.
* Does not include time the system was suspended.
* See: **clock_gettime**\ (**CLOCK_MONOTONIC**)
*
* Returns
* Current *ktime*.
*/
static __u64 (* const bpf_ktime_get_ns)(void) = (void *) 5;

/*
* bpf_trace_printk
*
* This helper is a "printk()-like" facility for debugging. It
* prints a message
* defined by format *fmt* (of size *fmt_size*)
* to file */sys/kernel/tracing/trace* from TraceFS, if
* available. It can take up to three additional **u64**
* arguments (as an eBPF helpers, the total number of arguments is
* limited to five).
*
* Each time the helper is called, it appends a line to the trace.
* Lines are discarded while */sys/kernel/tracing/trace* is
* open, use */sys/kernel/tracing/trace_pipe* to avoid this.
* The format of the trace is customizable, and the exact output
* one will get depends on the options set in
* */sys/kernel/tracing/trace_options* (see also the

```

```

* *README* file under the same directory). However, it usually
* defaults to something like:
*
* ::
*
* telnet-470 [001] .N.. 419421.045894: 0x00000001: <formatted msg>
*
* In the above:
*
* * ``telnet`` is the name of the current task.
* * ``470`` is the PID of the current task.
* * ``001`` is the CPU number on which the task is
*   running.
* * In
*   ``.N..``, each character refers to a set of
*   options (whether irqs are enabled, scheduling
*   options, whether hard/softirqs are running, level of
*   preempt_disabled respectively). **N** means that
*   **TIF_NEED_RESCHED** and **PREEMPT_NEED_RESCHED**
*   are set.
* * ``419421.045894`` is a timestamp.
* * ``0x00000001`` is a fake value used by BPF for the
*   instruction pointer register.
* * ``<formatted msg>`` is the message formatted with
*   *fmt*.
*
* The conversion specifiers supported by *fmt* are similar, but
* more limited than for printk(). They are **%d**, **%i**,
* **%u**, **%x**, **%ld**, **%li**, **%lu**, **%lx**, **%lld**,
* **%lli**, **%llu**, **%llx**, **%p**, **%s**. No modifier (size
* of field, padding with zeroes, etc.) is available, and the
* helper will return **-EINVAL** (but print nothing) if it
* encounters an unknown specifier.
*
* Also, note that **bpf_trace_printk**() is slow, and should
* only be used
* for debugging purposes. For this reason, a notice
* block (spanning several lines) is printed to kernel logs and
* states that the helper should not be used "for production use"
* the first time this helper is used (or more precisely, when
* **trace_printk**() buffers are allocated). For passing values
* to user space, perf events should be preferred.
*
* Returns
* The number of bytes written to the buffer, or a negative error
* in case of failure.
*/
static long (* const bpf_trace_printk)(const char *fmt, __u32 fmt_size, ...) = (void *) 6;

```

```

/*
 * bpf_get_prandom_u32
 *
 * Get a pseudo-random number.
 *
 * From a security point of view, this helper uses its own
 * pseudo-random internal state, and cannot be used to infer the
 * seed of other random functions in the kernel. However, it is
 * essential to note that the generator used by the helper is not
 * cryptographically secure.
 *
 * Returns
 * A random 32-bit unsigned value.
 */
static __u32 (* const bpf_get_prandom_u32)(void)
= (void *) 7;

/*
 * bpf_get_smp_processor_id
 *
 * Get the SMP (symmetric multiprocessing) processor id. Note that
 * all programs run with migration disabled, which means that the
 * SMP processor id is stable during all the execution of the
 * program.
 *
 * Returns
 * The SMP id of the processor running the program.
 */
static __u32 (* const bpf_get_smp_processor_id)(void) = (void *) 8;

/*
 * bpf_skb_store_bytes
 *
 * Store *len* bytes from address *from* into the packet
 * associated to *skb*, at *offset*. *flags* are a combination of
 * **BPF_F_RECOMPUTE_CSUM** (automatically recompute the
 * checksum for the packet after storing the bytes) and
 * **BPF_F_INVALIDATE_HASH** (set *skb*->hash**, *skb*->
 * **->swhash** and *skb*->l4hash** to 0).
 *
 * A call to this helper is susceptible to change the underlying
 * packet buffer. Therefore, at load time, all checks on pointers
 * previously done by the verifier are invalidated and must
 * be
 * performed again, if the helper is used in combination with
 * direct packet access.
 *

```

```

* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_store_bytes)(struct __sk_buff *skb, __u32 offset, const void *from, __u32 len, __u64
flags) = (void *) 9;

/*
* bpf_l3_csum_replace
*
* Recompute the layer 3 (e.g. IP) checksum for the packet
* associated to *skb*. Computation is incremental, so the helper
* must know the former value of the header field that was
* modified (*from*), the new value of this field (*to*), and the
* number of bytes (2 or 4) for this field, stored in *size*.
* Alternatively, it is possible to store the difference between
* the previous and the new values of the header field in *to*, by
* setting *from* and *size* to 0. For both methods, *offset*
* indicates the location of the IP checksum within the packet.
*
* This helper works in combination with **bpf_csum_diff**(),
* which does not update
the checksum in-place, but offers more
* flexibility and can handle sizes larger than 2 or 4 for the
* checksum to update.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_l3_csum_replace)(struct __sk_buff *skb, __u32 offset, __u64 from, __u64 to, __u64 size) =
(void *) 10;

/*
* bpf_l4_csum_replace
*
* Recompute the layer 4 (e.g. TCP, UDP or ICMP) checksum for the
* packet associated to *skb*. Computation is incremental, so the
* helper must know the former value of the header field that was
* modified (*from*), the new value of this field (*to*), and the
* number of bytes (2 or 4) for this field, stored on the lowest
* four bits
of *flags*. Alternatively, it is possible to store
* the difference between the previous and the new values of the

```


- * header field in `*to*`, by setting `*from*` and the four lowest
- * bits of `*flags*` to 0. For both methods, `*offset*` indicates the
- * location of the IP checksum within the packet. In addition to
- * the size of the field, `*flags*` can be added (bitwise OR) actual
- * flags. With `**BPF_F_MARK_MANGLED_0**`, a null checksum is left
- * untouched (unless `**BPF_F_MARK_ENFORCE**` is added as well), and
- * for updates resulting in a null checksum the value is set to
- * `**CSUM_MANGLED_0**` instead. Flag `**BPF_F_PSEUDO_HDR**` indicates
- * the checksum is to be computed against a pseudo-header.
- *
- * This helper works in combination with `**bpf_csum_diff**\()`,
- * which does not update the checksum in-place, but offers more
- * flexibility and can handle sizes larger than 2 or 4 for the
- * checksum to update.
- *
- * A call to this helper is susceptible to change the underlying
- * packet buffer. Therefore, at load time, all checks on pointers
- * previously done by the verifier are invalidated and must be
- * performed again, if the helper is used in combination with
- * direct packet access.
- *
- * Returns
- * 0 on success, or a negative error in case of failure.
- */

```
static long (* const bpf_l4_csum_replace)(struct __sk_buff *skb, __u32 offset, __u64 from, __u64 to, __u64 flags)
= (void *) 11;
```

/*

- * `bpf_tail_call`
- *
- * This special helper is used to trigger a "tail call", or in
- * other words, to jump into another eBPF program. The same stack
- * frame is used (but values on stack and in registers for the
- * caller are not accessible to the callee). This mechanism allows
- * for program chaining, either for raising the maximum number of
- * available eBPF instructions, or to execute given programs in
- * conditional blocks. For security reasons, there is an upper
- * limit to the number of successive tail calls that can be
- * performed.
- *
- * Upon call
- of this helper, the program attempts to jump into a
- * program referenced at index `*index*` in `*prog_array_map*`, a
- * special map of type `**BPF_MAP_TYPE_PROG_ARRAY**`, and passes
- * `*ctx*`, a pointer to the context.
- *
- * If the call succeeds, the kernel immediately runs the first
- * instruction of the new program. This is not a function call,

```

* and it never returns to the previous program. If the call
* fails, then the helper has no effect, and the caller continues
* to run its subsequent instructions. A call can fail if the
* destination program for the jump does not exist (i.e. *index*
* is superior to the number of entries in *prog_array_map*), or
* if the maximum number of tail calls has been reached for this
* chain of programs. This limit is defined in the kernel by the
* macro **MAX_TAIL_CALL_CNT** (not accessible to user space),
* which is currently set to 33.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const
bpf_tail_call)(void *ctx, void *prog_array_map, __u32 index) = (void *) 12;

/*
* bpf_clone_redirect
*
* Clone and redirect the packet associated to *skb* to another
* net device of index *ifindex*. Both ingress and egress
* interfaces can be used for redirection. The **BPF_F_INGRESS**
* value in *flags* is used to make the distinction (ingress path
* is selected if the flag is present, egress path otherwise).
* This is the only flag supported for now.
*
* In comparison with **bpf_redirect**() helper,
* **bpf_clone_redirect**() has the associated cost of
* duplicating the packet buffer, but this can be executed out of
* the eBPF program. Conversely, **bpf_redirect**() is more
* efficient, but it is handled through an action code where the
* redirection happens only after the eBPF program has returned.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously
done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure. Positive
* error indicates a potential drop or congestion in the target
* device. The particular positive error codes are not defined.
*/
static long (* const bpf_clone_redirect)(struct __sk_buff *skb, __u32 ifindex, __u64 flags) = (void *) 13;

/*

```

```

* bpf_get_current_pid_tgid
*
* Get the current pid and tgid.
*
* Returns
* A 64-bit integer containing the current tgid and pid, and
* created as such:
* *current_task* \ **->tgid << 32 \|**
* *current_task* \ **->pid**.
*/
static __u64 (* const bpf_get_current_pid_tgid)(void) = (void *) 14;

/*
* bpf_get_current_uid_gid
*
* Get the current uid and gid.
*
* Returns
* A 64-bit integer containing the current GID and UID, and
* created as such: *current_gid* **<< 32 \|** *current_uid*.
*/
static __u64 (* const
bpf_get_current_uid_gid)(void) = (void *) 15;

/*
* bpf_get_current_comm
*
* Copy the **comm** attribute of the current task into *buf* of
* *size_of_buf*. The **comm** attribute contains the name of
* the executable (excluding the path) for the current task. The
* *size_of_buf* must be strictly positive. On success, the
* helper makes sure that the *buf* is NUL-terminated. On failure,
* it is filled with zeroes.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_get_current_comm)(void *buf, __u32 size_of_buf) = (void *) 16;

/*
* bpf_get_cgroup_classid
*
* Retrieve the classid for the current task, i.e. for the net_cls
* cgroup to which *skb* belongs.
*
* This helper can be used on TC egress path, but not on ingress.
*
* The net_cls cgroup provides an interface to tag network packets

```

- * based on a user-provided identifier for all traffic coming from
- * the tasks belonging to the related cgroup. See also the related
- * kernel documentation, available from the Linux sources in file
- * `*Documentation/admin-guide/cgroup-v1/net_cls.rst*`.
- *
- * The Linux kernel has two versions for cgroups: there are
- * cgroups v1 and cgroups v2. Both are available to users, who can
- * use a mixture of them, but note that the `net_cls` cgroup is for
- * cgroup v1 only. This makes it incompatible with BPF programs
- * run on cgroups, which is a cgroup-v2-only feature (a socket can
- * only hold data for one version of cgroups at a time).
- *
- * This helper is only available if the kernel was compiled with
- * the `**CONFIG_CGROUP_NET_CLASSID**` configuration option set to
- * `"**y**"` or to `"**m**"`.
- *
- * Returns
- * The classid, or 0 for the default unconfigured classid.
- */

```
static __u32 (* const bpf_get_cgroup_classid)(struct __sk_buff *skb) = (void *) 17;
```



```
/*
 * bpf_skb_vlan_push
 *
 * Push a *vlan_tci* (VLAN tag control information) of protocol
 * *vlan_proto* to the packet associated to *skb*, then update
 * the checksum. Note that if *vlan_proto* is different from
 * **ETH_P_8021Q** and **ETH_P_8021AD**, it is considered to
 * be **ETH_P_8021Q**.
 *
 * A call to this helper is susceptible to change the underlying
 * packet buffer. Therefore, at load time, all checks on pointers
 * previously done by the verifier are invalidated and must be
 * performed again, if the helper is used in combination with
 * direct packet access.
 *
 * Returns
 * 0 on success, or a negative error in case of failure.
 */
```

```
static long (* const bpf_skb_vlan_push)(struct __sk_buff *skb, __be16 vlan_proto, __u16 vlan_tci) = (void *) 18;
```



```
/*
 * bpf_skb_vlan_pop
 *
 * Pop a VLAN header from the packet associated to *skb*.
 *
 * A call to this helper is susceptible to change the underlying
```

```

* packet buffer. Therefore, at load time, all checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_vlan_pop)(struct __sk_buff *skb) = (void *) 19;

/*
* bpf_skb_get_tunnel_key
*
* Get tunnel metadata. This helper takes a pointer *key* to an
* empty **struct bpf_tunnel_key** of **size**, that will be
* filled with tunnel metadata for the packet associated to *skb*.
* The *flags* can be set to **BPF_F_TUNINFO_IPV6**, which
* indicates that the tunnel is based on IPv6 protocol instead of
* IPv4.
*
* The **struct bpf_tunnel_key** is an object that generalizes the
* principal parameters used by various tunneling protocols into a
* single struct. This way, it can be used to easily make a
* decision based on the contents of the encapsulation header,
* "summarized" in this struct. In particular, it holds the IP
* address of the remote end (IPv4 or IPv6, depending on the case)
* in *key* \ **->remote_ipv4** or *key* \ **->remote_ipv6**. Also,
* this struct
* exposes the *key* \ **->tunnel_id**, which is
* generally mapped to a VNI (Virtual Network Identifier), making
* it programmable together with the **bpf_skb_set_tunnel_key** \
* () helper.
*
* Let's imagine that the following code is part of a program
* attached to the TC ingress interface, on one end of a GRE
* tunnel, and is supposed to filter out all messages coming from
* remote ends with IPv4 address other than 10.0.0.1:
*
* ::
*
* int ret;
* struct bpf_tunnel_key key = {};
*
* ret = bpf_skb_get_tunnel_key(skb, &key, sizeof(key), 0);
* if (ret < 0)
*     return TC_ACT_SHOT; // drop packet
*
* if (key.remote_ipv4 != 0x0a000001)

```

```

* return TC_ACT_SHOT; // drop packet
*
* return TC_ACT_OK; // accept packet
*
* This interface can also be used with all encapsulation devices
* that can operate in "collect metadata" mode: instead of having
* one network device per specific configuration, the "collect
* metadata" mode only requires a single
device where the
* configuration can be extracted from this helper.
*
* This can be used together with various tunnels such as VXLAN,
* Geneve, GRE or IP in IP (IPIP).
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_get_tunnel_key)(struct __sk_buff *skb, struct bpf_tunnel_key *key, __u32 size, __u64
flags) = (void *) 20;

/*
* bpf_skb_set_tunnel_key
*
* Populate tunnel metadata for packet associated to *skb.* The
* tunnel metadata is set to the contents of *key*, of *size*. The
* *flags* can be set to a combination of the following values:
*
* **BPF_F_TUNINFO_IPV6**
* Indicate that the tunnel is based on IPv6 protocol
* instead of IPv4.
* **BPF_F_ZERO_CSUM_TX**
* For IPv4 packets, add a flag to tunnel metadata
* indicating that checksum computation should be skipped
* and checksum set to zeroes.
* **BPF_F_DONT_FRAGMENT**
* Add a flag to tunnel metadata indicating that the
* packet should
not be fragmented.
* **BPF_F_SEQ_NUMBER**
* Add a flag to tunnel metadata indicating that a
* sequence number should be added to tunnel header before
* sending the packet. This flag was added for GRE
* encapsulation, but might be used with other protocols
* as well in the future.
* **BPF_F_NO_TUNNEL_KEY**
* Add a flag to tunnel metadata indicating that no tunnel
* key should be set in the resulting tunnel header.
*

```

```

* Here is a typical usage on the transmit path:
*
* ::
*
* struct bpf_tunnel_key key;
*     populate key ...
* bpf_skb_set_tunnel_key(skb, &key, sizeof(key), 0);
* bpf_clone_redirect(skb, vxlan_dev_ifindex, 0);
*
* See also the description of the bpf_skb_get_tunnel_key()
* helper for additional information.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/

static long (* const bpf_skb_set_tunnel_key)(struct __sk_buff *skb, struct bpf_tunnel_key *key, __u32 size, __u64
flags) = (void *) 21;

/*
* bpf_perf_event_read
*
* Read the value of a perf event counter. This helper relies on a
* *map* of type BPF_MAP_TYPE_PERF_EVENT_ARRAY. The nature of
* the perf event counter is selected when *map* is updated with
* perf event file descriptors. The *map* is an array whose size
* is the number of available CPUs, and each cell contains a value
* relative to one CPU. The value to retrieve is indicated by
* *flags*, that contains the index of the CPU to look up, masked
* with BPF_F_INDEX_MASK. Alternatively, *flags* can be set to
* BPF_F_CURRENT_CPU to indicate that the value for the
* current CPU should be retrieved.
*
* Note that before Linux 4.13, only hardware perf event can be
* retrieved.
*
* Also, be aware that the newer helper
* bpf_perf_event_read_value() is recommended over
* bpf_perf_event_read() in general. The latter has some ABI
* quirks where error and counter value are used as a return code
* (which is wrong
to do since ranges may overlap). This issue is
* fixed with bpf_perf_event_read_value(), which at the same
* time provides more features over the bpf_perf_event_read\
* () interface. Please refer to the description of
* bpf_perf_event_read_value() for details.
*
* Returns
* The value of the perf event counter read from the map, or a

```

```

* negative error code in case of failure.
*/
static __u64 (* const bpf_perf_event_read)(void *map, __u64 flags) = (void *) 22;

/*
* bpf_redirect
*
* Redirect the packet to another net device of index *ifindex*.
* This helper is somewhat similar to bpf_clone_redirect\
* (), except that the packet is not cloned, which provides
* increased performance.
*
* Except for XDP, both ingress and egress interfaces can be used
* for redirection. The BPF_F_INGRESS value in *flags* is used
* to make the distinction (ingress path is selected if the flag
* is present, egress path otherwise). Currently, XDP only
*
* supports redirection to the egress interface, and accepts no
* flag at all.
*
* The same effect can also be attained with the more generic
* bpf_redirect_map() , which uses a BPF map to store the
* redirect target instead of providing it directly to the helper.
*
* Returns
* For XDP, the helper returns XDP_REDIRECT on success or
* XDP_ABORTED on error. For other program types, the values
* are TC_ACT_REDIRECT on success or TC_ACT_SHOT on
* error.
*/
static long (* const bpf_redirect)(__u32 ifindex, __u64 flags) = (void *) 23;

/*
* bpf_get_route_realm
*
* Retrieve the realm or the route, that is to say the
* tcclassid field of the destination for the *skb*. The
* identifier retrieved is a user-provided tag, similar to the
* one used with the net_cls cgroup (see description for
* bpf_get_cgroup_classid() helper), but here this tag is
* held by a route (a destination entry), not by a task.
*
* Retrieving this identifier
* works with the clsact TC egress hook
* (see also tc-bpf(8)), or alternatively on conventional
* classful egress qdiscs, but not on TC ingress path. In case of
* clsact TC egress hook, this has the advantage that, internally,
* the destination entry has not been dropped yet in the transmit

```



```

* path. Therefore, the destination entry does not need to be
* artificially held via **netif_keep_dst**\() for a classful
* qdisc until the *skb* is freed.
*
* This helper is available only if the kernel was compiled with
* **CONFIG_IP_ROUTE_CLASSID** configuration option.
*
* Returns
* The realm of the route for the packet associated to *skb*, or 0
* if none was found.
*/
static __u32 (* const bpf_get_route_realm)(struct __sk_buff *skb) = (void *) 24;

/*
* bpf_perf_event_output
*
* Write raw *data* blob into a special BPF perf event held by
* *map* of type **BPF_MAP_TYPE_PERF_EVENT_ARRAY**. This perf
* event must have the following attributes: **PERF_SAMPLE_RAW**
* as **sample_type**, **PERF_TYPE_SOFTWARE** as **type**, and
* **PERF_COUNT_SW_BPF_OUTPUT** as **config**.
*
* The *flags* are used to indicate the index in *map* for which
* the value must be put, masked with **BPF_F_INDEX_MASK**.
* Alternatively, *flags* can be set to **BPF_F_CURRENT_CPU**
* to indicate that the index of the current CPU core should be
* used.
*
* The value to write, of *size*, is passed through eBPF stack and
* pointed by *data*.
*
* The context of the program *ctx* needs also be passed to the
* helper.
*
* On user space, a program willing to read the values needs to
* call **perf_event_open**\() on the perf event (either for
* one or for all CPUs) and to store the file descriptor into the
* *map*. This must be done before the eBPF program can send data
* into it. An example is available in file
* *samples/bpf/trace_output_user.c* in the Linux kernel source
* tree (the eBPF program counterpart is in
* *samples/bpf/trace_output_kern.c*).
*
* **bpf_perf_event_output**\() achieves better performance
* than **bpf_trace_printk**\() for sharing data with user
* space, and is much better suitable for streaming data from eBPF
* programs.
*

```

```

* Note that this helper is not restricted to tracing use cases
* and can be used with programs attached to TC or XDP as well,
* where it allows for passing data to user space listeners. Data
* can be:
*
* * Only custom structs,
* * Only the packet payload, or
* * A combination of both.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_perf_event_output)(void *ctx, void *map, __u64 flags, void *data, __u64 size) = (void *)
25;

/*
* bpf_skb_load_bytes
*
* This helper was provided as an easy way to load data from a
* packet. It can be used to load *len* bytes from *offset* from
* the packet associated to *skb*, into the buffer pointed by
* *to*.
*
* Since Linux 4.7, usage of this helper
has mostly been replaced
* by "direct packet access", enabling packet data to be
* manipulated with *skb->data* and *skb->data_end*
* pointing respectively to the first byte of packet data and to
* the byte after the last byte of packet data. However, it
* remains useful if one wishes to read large quantities of data
* at once from a packet into the eBPF stack.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_load_bytes)(const void *skb, __u32 offset, void *to, __u32 len) = (void *) 26;

/*
* bpf_get_stackid
*
* Walk a user or a kernel stack and return its id. To achieve
* this, the helper needs *ctx*, which is a pointer to the context
* on which the tracing program is executed, and a pointer to a
* *map* of type **BPF_MAP_TYPE_STACK_TRACE**.
*
* The last argument, *flags*, holds the number of stack frames to
* skip (from 0 to 255), masked with
* **BPF_F_SKIP_FIELD_MASK**. The next bits can

```

be used to set

- * a combination of the following flags:
- *
- * ****BPF_F_USER_STACK****
- * Collect a user space stack instead of a kernel stack.
- * ****BPF_F_FAST_STACK_CMP****
- * Compare stacks by hash only.
- * ****BPF_F_REUSE_STACKID****
- * If two different stacks hash into the same **stackid**,
- * discard the old one.
- *
- * The stack id retrieved is a 32 bit long integer handle which
- * can be further combined with other data (including other stack
- * ids) and used as a key into maps. This can be useful for
- * generating a variety of graphs (such as flame graphs or off-cpu
- * graphs).
- *
- * For walking a stack, this helper is an improvement over
- * ****bpf_probe_read****() , which can be used with unrolled loops
- * but is not efficient and consumes a lot of eBPF instructions.
- * Instead, ****bpf_get_stackid****() can collect up to
- * ****PERF_MAX_STACK_DEPTH**** both kernel and user frames. Note that
- * this limit can be controlled with the ****sysctl**** program, and
- * that it should

be manually increased in order to profile long

- * user stacks (such as stacks for Java programs). To do so, use:
- *
- * ::
- *
- * # sysctl kernel.perf_event_max_stack=<new value>
- *

* Returns

- * The positive or null stack id on success, or a negative error
- * in case of failure.

*/

```
static long (* const bpf_get_stackid)(void *ctx, void *map, __u64 flags) = (void *) 27;
```

/*

- * bpf_csum_diff
- *
- * Compute a checksum difference, from the raw buffer pointed by
- * **from**, of length **from_size** (that must be a multiple of 4),
- * towards the raw buffer pointed by **to**, of size **to_size**
- * (same remark). An optional **seed** can be added to the value
- * (this can be cascaded, the seed may come from a previous call
- * to the helper).
- *
- * This is flexible enough to be used in several ways:

```

*
* * With *from_size* == 0, *to_size* > 0 and *seed* set to
* checksum, it can be used when pushing new data.
* * With *from_size* > 0, *to_size* == 0 and *seed* set to
* checksum,
it can be used when removing data from a packet.
* * With *from_size* > 0, *to_size* > 0 and *seed* set to 0, it
* can be used to compute a diff. Note that *from_size* and
* *to_size* do not need to be equal.
*
* This helper can be used in combination with
* **bpf_l3_csum_replace**() and **bpf_l4_csum_replace**(), to
* which one can feed in the difference computed with
* **bpf_csum_diff**().
*
* Returns
* The checksum result, or a negative error code in case of
* failure.
*/
static __s64 (* const bpf_csum_diff)(__be32 *from, __u32 from_size, __be32 *to, __u32 to_size, __wsum seed) =
(void *) 28;

/*
* bpf_skb_get_tunnel_opt
*
* Retrieve tunnel options metadata for the packet associated to
* *skb*, and store the raw tunnel option data to the buffer *opt*
* of *size*.
*
* This helper can be used with encapsulation devices that can
* operate in "collect metadata" mode (please refer to the related
* note in the description of **bpf_skb_get_tunnel_key**\
() for
* more details). A particular example where this can be used is
* in combination with the Geneve encapsulation protocol, where it
* allows for pushing (with **bpf_skb_get_tunnel_opt**() helper)
* and retrieving arbitrary TLVs (Type-Length-Value headers) from
* the eBPF program. This allows for full customization of these
* headers.
*
* Returns
* The size of the option data retrieved.
*/
static long (* const bpf_skb_get_tunnel_opt)(struct __sk_buff *skb, void *opt, __u32 size) = (void *) 29;

/*
* bpf_skb_set_tunnel_opt
*

```

```

* Set tunnel options metadata for the packet associated to *skb*
* to the option data contained in the raw buffer *opt* of *size*.
*
* See also the description of the bpf_skb_get_tunnel_opt()
* helper for additional information.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_set_tunnel_opt)(struct __sk_buff *skb, void *opt, __u32 size) = (void *) 30;

/*
*
* bpf_skb_change_proto
*
* Change the protocol of the *skb* to *proto*. Currently
* supported are transition from IPv4 to IPv6, and from IPv6 to
* IPv4. The helper takes care of the groundwork for the
* transition, including resizing the socket buffer. The eBPF
* program is expected to fill the new headers, if any, via
* skb_store_bytes() and to recompute the checksums with
* bpf_l3_csum_replace() and bpf_l4_csum_replace
* (). The main case for this helper is to perform NAT64
* operations out of an eBPF program.
*
* Internally, the GSO type is marked as dodgy so that headers are
* checked and segments are recalculated by the GSO/GRO engine.
* The size for GSO target is adapted as well.
*
* All values for *flags* are reserved for future usage, and must
* be left at zero.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously done by the verifier are
* invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_change_proto)(struct __sk_buff *skb, __be16 proto, __u64 flags) = (void *) 31;

/*
* bpf_skb_change_type
*
* Change the packet type for the packet associated to *skb*. This

```

```

* comes down to setting *skb* \ **->pkt_type** to *type*, except
* the eBPF program does not have a write access to *skb*\
* **->pkt_type** beside this helper. Using a helper here allows
* for graceful handling of errors.
*
* The major use case is to change incoming *skb*s to
* **PACKET_HOST** in a programmatic way instead of having to
* recirculate via **redirect**\ (... , **BPF_F_INGRESS**), for
* example.
*
* Note that *type* only allows certain values. At this time, they
* are:
*
* **PACKET_HOST**
*   Packet is for us.
* **PACKET_BROADCAST**
*   Send packet to all.
* **PACKET_MULTICAST**
*   Send packet to group.
* **PACKET_OTHERHOST**
*   Send packet to someone else.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_change_type)(struct __sk_buff *skb, __u32 type) = (void *) 32;

/*
* bpf_skb_under_cgroup
*
* Check whether *skb* is a descendant of the cgroup2 held by
* *map* of type **BPF_MAP_TYPE_CGROUP_ARRAY**, at *index*.
*
* Returns
* The return value depends on the result of the test, and can be:
*
* * 0, if the *skb* failed the cgroup2 descendant test.
* * 1, if the *skb* succeeded the cgroup2 descendant test.
* * A negative error code, if an error occurred.
*/
static long (* const bpf_skb_under_cgroup)(struct __sk_buff *skb, void *map, __u32 index) = (void *) 33;

/*
* bpf_get_hash_recalc
*
* Retrieve the hash of the packet, *skb* \ **->hash**. If it is
* not set, in particular if the hash was cleared due to mangling,
* recompute this hash. Later accesses

```

```

to the hash can be done
* directly with *skb* \ **->hash**.
*
* Calling **bpf_set_hash_invalid**\ (), changing a packet
* prototype with **bpf_skb_change_proto**\ (), or calling
* **bpf_skb_store_bytes**\ () with the
* **BPF_F_INVALIDATE_HASH** are actions susceptible to clear
* the hash and to trigger a new computation for the next call to
* **bpf_get_hash_recalc**\ ().
*
* Returns
* The 32-bit hash.
*/
static __u32 (* const bpf_get_hash_recalc)(struct __sk_buff *skb) = (void *) 34;

/*
* bpf_get_current_task
*
* Get the current task.
*
* Returns
* A pointer to the current task struct.
*/
static __u64 (* const bpf_get_current_task)(void) = (void *) 35;

/*
* bpf_probe_write_user
*
* Attempt in a safe way to write *len* bytes from the buffer
* *src* to *dst* in memory. It only works for threads that are in
* user context, and *dst* must be a valid user space address.
*
* This helper should not be used to implement any kind of
* security mechanism
* because of TOC-TOU attacks, but rather to
* debug, divert, and manipulate execution of semi-cooperative
* processes.
*
* Keep in mind that this feature is meant for experiments, and it
* has a risk of crashing the system and running programs.
* Therefore, when an eBPF program using this helper is attached,
* a warning including PID and process name is printed to kernel
* logs.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_probe_write_user)(void *dst, const void *src, __u32 len) = (void *) 36;

```

```

/*
 * bpf_current_task_under_cgroup
 *
 * Check whether the probe is being run in the context of a given
 * subset of the cgroup2 hierarchy. The cgroup2 to test is held by
 * *map* of type **BPF_MAP_TYPE_CGROUP_ARRAY**, at *index*.
 *
 * Returns
 * The return value depends on the result of the test, and can be:
 *
 * * 1, if current task belongs to the cgroup2.
 * * 0, if current task does not belong to the cgroup2.
 * * A negative error code, if an error occurred.
 */
static long (* const bpf_current_task_under_cgroup)(void *map, __u32 index) = (void *) 37;

/*
 * bpf_skb_change_tail
 *
 * Resize (trim or grow) the packet associated to *skb* to the
 * new *len*. The *flags* are reserved for future usage, and must
 * be left at zero.
 *
 * The basic idea is that the helper performs the needed work to
 * change the size of the packet, then the eBPF program rewrites
 * the rest via helpers like **bpf_skb_store_bytes**(),
 * **bpf_l3_csum_replace**(), **bpf_l3_csum_replace**()
 * and others. This helper is a slow path utility intended for
 * replies with control messages. And because it is targeted for
 * slow path, the helper itself can afford to be slow: it
 * implicitly linearizes, unclones and drops offloads from the
 * *skb*.
 *
 * A call to this helper is susceptible to change the underlying
 * packet buffer. Therefore, at load time, all checks on pointers
 * previously done
 * by the verifier are invalidated and must be
 * performed again, if the helper is used in combination with
 * direct packet access.
 *
 * Returns
 * 0 on success, or a negative error in case of failure.
 */
static long (* const bpf_skb_change_tail)(struct __sk_buff *skb, __u32 len, __u64 flags) = (void *) 38;

/*
 * bpf_skb_pull_data

```



```

*
* Pull in non-linear data in case the *skb* is non-linear and not
* all of *len* are part of the linear section. Make *len* bytes
* from *skb* readable and writable. If a zero value is passed for
* *len*, then all bytes in the linear part of *skb* will be made
* readable and writable.
*
* This helper is only needed for reading and writing with direct
* packet access.
*
* For direct packet access, testing that offsets to access
* are within packet boundaries (test on *skb* \ **->data_end**) is
* susceptible to fail if offsets are invalid, or if the requested
* data is in non-linear parts of the *skb*. On failure the
* program
can just bail out, or in the case of a non-linear
* buffer, use a helper to make the data available. The
* **bpf_skb_load_bytes**() helper is a first solution to access
* the data. Another one consists in using **bpf_skb_pull_data**
* to pull in once the non-linear parts, then retesting and
* eventually access the data.
*
* At the same time, this also makes sure the *skb* is uncloned,
* which is a necessary condition for direct write. As this needs
* to be an invariant for the write part only, the verifier
* detects writes and adds a prologue that is calling
* **bpf_skb_pull_data**() to effectively unclone the *skb* from
* the very beginning in case it is indeed cloned.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
*
Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_pull_data)(struct __sk_buff *skb, __u32 len) = (void *) 39;

/*
* bpf_csum_update
*
* Add the checksum *csum* into *skb* \ **->csum** in case the
* driver has supplied a checksum for the entire packet into that
* field. Return an error otherwise. This helper is intended to be
* used in combination with **bpf_csum_diff**(), in particular

```

```

* when the checksum needs to be updated after data has been
* written into the packet through direct packet access.
*
* Returns
* The checksum on success, or a negative error code in case of
* failure.
*/
static __s64 (* const bpf_csum_update)(struct __sk_buff *skb, __wsum csum) = (void *) 40;

/*
* bpf_set_hash_invalid
*
* Invalidate the current *skb* **->hash**. It can be used after
* mangling on headers through direct packet access, in order to
* indicate that the hash is outdated and to trigger a
* recalculation the next
time the kernel tries to access this
* hash or when the **bpf_get_hash_recalc**() helper is called.
*
* Returns
* void.
*/
static void (* const bpf_set_hash_invalid)(struct __sk_buff *skb) = (void *) 41;

/*
* bpf_get_numa_node_id
*
* Return the id of the current NUMA node. The primary use case
* for this helper is the selection of sockets for the local NUMA
* node, when the program is attached to sockets using the
* **SO_ATTACH_REUSEPORT_EBPF** option (see also **socket(7)**),
* but the helper is also available to other eBPF program types,
* similarly to **bpf_get_smp_processor_id**().
*
* Returns
* The id of current NUMA node.
*/
static long (* const bpf_get_numa_node_id)(void) = (void *) 42;

/*
* bpf_skb_change_head
*
* Grows headroom of packet associated to *skb* and adjusts the
* offset of the MAC header accordingly, adding *len* bytes of
* space. It automatically extends and reallocates memory as
* required.
*
* This helper can be used

```

```

on a layer 3 *skb* to push a MAC header
* for redirection into a layer 2 device.
*
* All values for *flags* are reserved for future usage, and must
* be left at zero.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_change_head)(struct __sk_buff *skb, __u32 len, __u64 flags) = (void *) 43;

/*
* bpf_xdp_adjust_head
*
* Adjust (move) *xdp_md* **->data** by *delta* bytes. Note that
* it is possible to use a negative value for *delta*. This helper
* can be used to prepare the packet for pushing or popping
* headers.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all
checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_xdp_adjust_head)(struct xdp_md *xdp_md, int delta) = (void *) 44;

/*
* bpf_probe_read_str
*
* Copy a NUL terminated string from an unsafe kernel address
* *unsafe_ptr* to *dst*. See **bpf_probe_read_kernel_str**\ () for
* more details.
*
* Generally, use **bpf_probe_read_user_str**\ () or
* **bpf_probe_read_kernel_str**\ () instead.
*
* Returns

```

```

* On success, the strictly positive length of the string,
* including the trailing NUL character. On error, a negative
* value.
*/
static long (* const bpf_probe_read_str)(void *dst, __u32 size, const void *unsafe_ptr) = (void *) 45;

/*
* bpf_get_socket_cookie
*
* If the **struct sk_buff** pointed by *skb* has a known socket,
* retrieve the cookie
* (generated by the kernel) of this socket.
* If no cookie has been set yet, generate a new cookie. Once
* generated, the socket cookie remains stable for the life of the
* socket. This helper can be useful for monitoring per socket
* networking traffic statistics as it provides a global socket
* identifier that can be assumed unique.
*
* Returns
* A 8-byte long unique number on success, or 0 if the socket
* field is missing inside *skb*.
*/
static __u64 (* const bpf_get_socket_cookie)(void *ctx) = (void *) 46;

/*
* bpf_get_socket_uid
*
* Get the owner UID of the socket associated to *skb*.
*
* Returns
* The owner UID of the socket associated to *skb*. If the socket
* is **NULL**, or if it is not a full socket (i.e. if it is a
* time-wait or a request socket instead), **overflowuid** value
* is returned (note that **overflowuid** might also be the actual
* UID value for the socket).
*/
static __u32 (* const bpf_get_socket_uid)(struct __sk_buff *skb) =
(void *) 47;

/*
* bpf_set_hash
*
* Set the full hash for *skb* (set the field *skb*->hash**)
* to value *hash*.
*
* Returns
* 0
*/

```

```

static long (* const bpf_set_hash)(struct __sk_buff *skb, __u32 hash) = (void *) 48;

/*
 * bpf_setsockopt
 *
 * Emulate a call to **setsockopt()** on the socket associated to
 * *bpf_socket*, which must be a full socket. The *level* at
 * which the option resides and the name *optname* of the option
 * must be specified, see **setsockopt(2)** for more information.
 * The option value of length *optlen* is pointed by *optval*.
 *
 * *bpf_socket* should be one of the following:
 *
 * * **struct bpf_sock_ops** for **BPF_PROG_TYPE_SOCKET_OPS**.
 * * **struct bpf_sock_addr** for **BPF_CGROUP_INET4_CONNECT**,
 *   **BPF_CGROUP_INET6_CONNECT** and **BPF_CGROUP_UNIX_CONNECT**.
 *
 * This helper actually implements a subset of **setsockopt()**.
 * It supports the following *level*s:
 *
 * * **SOL_SOCKET**, which supports the following *optname*s:
 *   * **SO_RCVBUF**, **SO_SNDBUF**, **SO_MAX_PACING_RATE**,
 *   * **SO_PRIORITY**, **SO_RCVLOWAT**, **SO_MARK**,
 *   * **SO_BINDTODEVICE**, **SO_KEEPAIVE**, **SO_REUSEADDR**,
 *   * **SO_REUSEPORT**, **SO_BINDTOIFINDEX**, **SO_TXREHASH**.
 * * **IPPROTO_TCP**, which supports the following *optname*s:
 *   * **TCP_CONGESTION**, **TCP_BPF_IW**,
 *   * **TCP_BPF_SNDCWND_CLAMP**, **TCP_SAVE_SYN**,
 *   * **TCP_KEEPIDL**E, **TCP_KEEPI**NVL, **TCP_KEEPCNT**,
 *   * **TCP_SYNCNT**, **TCP_USER_TIMEOUT**, **TCP_NOTSENT_LOWAT**,
 *   * **TCP_NODELAY**, **TCP_MAXSEG**, **TCP_WINDOW_CLAMP**,
 *   * **TCP_THIN_LINEAR_TIMEOUTS**, **TCP_BPF_DELACK_MAX**,
 *   * **TCP_BPF_RTO_MIN**.
 * * **IPPROTO_IP**, which supports *optname* **IP_TOS**.
 * * **IPPROTO_IPV6**, which supports the following *optname*s:
 *   * **IPV6_TCLASS**, **IPV6_AUTOFLOWLABEL**.
 *
 * Returns
 *
 * 0 on success, or a negative error in case of failure.
 */
static long (* const bpf_setsockopt)(void *bpf_socket, int level,
int optname, void *optval, int optlen) = (void *) 49;

/*
 * bpf_skb_adjust_room
 *
 * Grow or shrink the room for data in the packet associated to

```

* *skb* by *len_diff*, and according to the selected *mode*.

*

* By default, the helper will reset any offloaded checksum

* indicator of the *skb* to `CHECKSUM_NONE`. This can be avoided

* by the following flag:

*

* **BPF_F_ADJ_ROOM_NO_CSUM_RESET**: Do not reset offloaded

* checksum data of the *skb* to `CHECKSUM_NONE`.

*

* There are two supported modes at this time:

*

* **BPF_ADJ_ROOM_MAC**: Adjust room at the mac layer

* (room space is added or removed between the layer 2 and

* layer 3 headers).

*

* **BPF_ADJ_ROOM_NET**: Adjust room at the network layer

* (room space is added or removed between the layer 3 and

* layer 4 headers).

*

* The following flags are supported at this time:

*

* **BPF_F_ADJ_ROOM_FIXED_GSO**: Do not adjust *gso_size*.

* Adjusting *mss* in this way is not allowed for datagrams.

*

* **BPF_F_ADJ_ROOM_ENCAP_L3_IPV4**,

* **BPF_F_ADJ_ROOM_ENCAP_L3_IPV6**:

* Any new space is reserved to hold a tunnel header.

* Configure *skb* offsets and other fields accordingly.

*

* **BPF_F_ADJ_ROOM_ENCAP_L4_GRE**,

* **BPF_F_ADJ_ROOM_ENCAP_L4_UDP**:

* Use with `ENCAP_L3` flags to further specify the tunnel type.

*

* **BPF_F_ADJ_ROOM_ENCAP_L2** *(len)*:

* Use with `ENCAP_L3/L4` flags to further specify the tunnel

* type; *len* is the length of the inner MAC header.

*

* **BPF_F_ADJ_ROOM_ENCAP_L2_ETH**:

* Use with `BPF_F_ADJ_ROOM_ENCAP_L2` flag to further specify the

* L2 type as Ethernet.

*

* **BPF_F_ADJ_ROOM_DECAP_L3_IPV4**,

* **BPF_F_ADJ_ROOM_DECAP_L3_IPV6**:

* Indicate the new IP header version after decapsulating the outer

* IP header. Used when the inner and outer IP versions are different.

*

* A call to this helper is susceptible to change the underlying

* packet buffer. Therefore, at load time, all

```

checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_adjust_room)(struct __sk_buff *skb, __s32 len_diff, __u32 mode, __u64 flags) = (void *) 50;

/*
* bpf_redirect_map
*
* Redirect the packet to the endpoint referenced by *map* at
* index *key*. Depending on its type, this *map* can contain
* references to net devices (for forwarding packets through other
* ports), or to CPUs (for redirecting XDP frames to another CPU;
* but this is only implemented for native XDP (with driver
* support) as of this writing).
*
* The lower two bits of *flags* are used as the return code if
* the map lookup fails. This is so that the return value can be
* one of the XDP program return codes up to **XDP_TX**, as chosen
* by the caller. The higher bits of *flags*
can be set to
* BPF_F_BROADCAST or BPF_F_EXCLUDE_INGRESS as defined below.
*
* With BPF_F_BROADCAST the packet will be broadcasted to all the
* interfaces in the map, with BPF_F_EXCLUDE_INGRESS the ingress
* interface will be excluded when do broadcasting.
*
* See also **bpf_redirect**(), which only supports redirecting
* to an ifindex, but doesn't require a map to do so.
*
* Returns
* **XDP_REDIRECT** on success, or the value of the two lower bits
* of the *flags* argument on error.
*/
static long (* const bpf_redirect_map)(void *map, __u64 key, __u64 flags) = (void *) 51;

/*
* bpf_sk_redirect_map
*
* Redirect the packet to the socket referenced by *map* (of type
* **BPF_MAP_TYPE_SOCKMAP**) at index *key*. Both ingress and
* egress interfaces can be used for redirection. The
* **BPF_F_INGRESS** value in *flags* is used to make the

```

```

* distinction (ingress path is selected if the flag is present,
* egress path otherwise). This is the only flag supported
for now.
*
* Returns
* **SK_PASS** on success, or **SK_DROP** on error.
*/
static long (* const bpf_sk_redirect_map)(struct __sk_buff *skb, void *map, __u32 key, __u64 flags) = (void *) 52;

/*
* bpf_sock_map_update
*
* Add an entry to, or update a *map* referencing sockets. The
* *skops* is used as a new value for the entry associated to
* *key*. *flags* is one of:
*
* **BPF_NOEXIST**
* The entry for *key* must not exist in the map.
* **BPF_EXIST**
* The entry for *key* must already exist in the map.
* **BPF_ANY**
* No condition on the existence of the entry for *key*.
*
* If the *map* has eBPF programs (parser and verdict), those will
* be inherited by the socket being added. If the socket is
* already attached to eBPF programs, this results in an error.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_sock_map_update)(struct bpf_sock_ops *skops, void *map, void *key, __u64 flags) = (void *)
*)
53;

/*
* bpf_xdp_adjust_meta
*
* Adjust the address pointed by *xdp_md* **->data_meta** by
* *delta* (which can be positive or negative). Note that this
* operation modifies the address stored in *xdp_md* **->data**,
* so the latter must be loaded only after the helper has been
* called.
*
* The use of *xdp_md* **->data_meta** is optional and programs
* are not required to use it. The rationale is that when the
* packet is processed with XDP (e.g. as DoS filter), it is
* possible to push further meta data along with it before passing
* to the stack, and to give the guarantee that an ingress eBPF

```


- * program attached as a TC classifier on the same device can pick
- * this up for further post-processing. Since TC works with socket
- * buffers, it remains possible to set from XDP the **mark** or
- * **priority** pointers, or other pointers for the socket buffer.
- * Having this scratch space generic and programmable allows for
- * more flexibility as the user is
- free to store whatever meta
- * data they need.
- *
- * A call to this helper is susceptible to change the underlying
- * packet buffer. Therefore, at load time, all checks on pointers
- * previously done by the verifier are invalidated and must be
- * performed again, if the helper is used in combination with
- * direct packet access.
- *
- * Returns
- * 0 on success, or a negative error in case of failure.
- */

```
static long (* const bpf_xdp_adjust_meta)(struct xdp_md *xdp_md, int delta) = (void *) 54;
```

/*

* bpf_perf_event_read_value

*

- * Read the value of a perf event counter, and store it into **buf**
- * of size **buf_size**. This helper relies on a **map** of type
- * **BPF_MAP_TYPE_PERF_EVENT_ARRAY**. The nature of the perf event
- * counter is selected when **map** is updated with perf event file
- * descriptors. The **map** is an array whose size is the number of
- * available CPUs, and each cell contains a value relative to one
- * CPU. The value to retrieve is indicated by **flags**, that
- * contains the index of the CPU to look up, masked with
- * **BPF_F_INDEX_MASK**. Alternatively, **flags** can be set to
- * **BPF_F_CURRENT_CPU** to indicate that the value for the
- * current CPU should be retrieved.
- *
- * This helper behaves in a way close to
- * **bpf_perf_event_read**() helper, save that instead of
- * just returning the value observed, it fills the **buf**
- * structure. This allows for additional data to be retrieved: in
- * particular, the enabled and running times (in **buf**
- * **->enabled** and **buf** **->running**, respectively) are
- * copied. In general, **bpf_perf_event_read_value**() is
- * recommended over **bpf_perf_event_read**(), which has some
- * ABI issues and provides fewer functionalities.
- *
- * These values are interesting, because hardware PMU (Performance
- * Monitoring Unit) counters are limited resources. When there are
- * more PMU based perf events opened than available counters,

- * kernel will multiplex these events so each event gets certain
- * percentage (but not all) of the PMU time. In case that
- * multiplexing happens, the number of samples or counter value
- * will not reflect the case compared to when no multiplexing
- * occurs. This makes comparison between different runs difficult.
- * Typically, the counter value should be normalized before
- * comparing to other experiments. The usual normalization is done
- * as follows.
- *
* ::
*
- *
$$\text{normalized\_counter} = \text{counter} * t\_enabled / t\_running$$
- *
* Where `t_enabled` is the time enabled for event and `t_running` is
- * the time running for event since last normalization. The
- * enabled and running times are accumulated since the perf event
- * open. To achieve scaling factor between two invocations of an
- * eBPF program, users can use CPU id as the key (which is
- * typical for perf array usage model) to remember the previous
- * value and do the calculation inside the eBPF program.
- *
* Returns
- * 0 on success, or a negative error in case of failure.
- */

```
static long (* const bpf_perf_event_read_value)(void *map, __u64 flags, struct bpf_perf_event_value *buf, __u32
buf_size) = (void *) 55;
```

/*

- * `bpf_perf_prog_read_value`
- *
* For an eBPF program attached to a perf event, retrieve the
- * value of the event counter associated to `*ctx*` and store it in
- * the structure pointed by `*buf*` and of size `*buf_size*`. Enabled
- * and running times are also stored in the structure (see
- * description of helper `**bpf_perf_event_read_value**\()` for
- * more details).
- *
* Returns
- * 0 on success, or a negative error in case of failure.
- */

```
static long (* const bpf_perf_prog_read_value)(struct bpf_perf_event_data *ctx, struct bpf_perf_event_value *buf,
__u32 buf_size) = (void *) 56;
```

/*

- * `bpf_getsockopt`
- *
* Emulate a call to `**getsockopt()` on the socket associated to

```

* *bpf_socket*, which must be a full socket. The *level* at
* which the option resides and the name *optname* of the option
* must be specified, see getsockopt(2)
for more information.
* The retrieved value is stored in the structure pointed by
* *opval* and of length *optlen*.
*
* *bpf_socket* should be one of the following:
*
* struct bpf_sock_ops for BPF_PROG_TYPE_SOCKET_OPS.
* struct bpf_sock_addr for BPF_CGROUP_INET4_CONNECT,
* BPF_CGROUP_INET6_CONNECT and BPF_CGROUP_UNIX_CONNECT.
*
* This helper actually implements a subset of getsockopt().
* It supports the same set of *optname*'s that is supported by
* the bpf_setsockopt() helper. The exceptions are
* TCP_BPF_* is bpf_setsockopt() only and
* TCP_SAVED_SYN is bpf_getsockopt() only.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_getsockopt)(void *bpf_socket, int level, int optname, void *optval, int optlen) = (void *) 57;

/*
* bpf_override_return
*
* Used for error injection, this helper uses kprobes to override
* the return value of the probed
* function, and to set it to *rc*.
* The first argument is the context *regs* on which the kprobe
* works.
*
* This helper works by setting the PC (program counter)
* to an override function which is run in place of the original
* probed function. This means the probed function is not run at
* all. The replacement function just returns with the required
* value.
*
* This helper has security implications, and thus is subject to
* restrictions. It is only available if the kernel was compiled
* with the CONFIG_BPF_KPROBE_OVERRIDE configuration
* option, and in this case it only works on functions tagged with
* ALLOW_ERROR_INJECTION in the kernel code.
*
* Also, the helper is only available for the architectures having
* the CONFIG_FUNCTION_ERROR_INJECTION option. As of this writing,
* x86 architecture is the only one to support this feature.

```

```

*
* Returns
* 0
*/
static long (* const bpf_override_return)(struct pt_regs *regs, __u64 rc) = (void
*) 58;

/*
* bpf_sock_ops_cb_flags_set
*
* Attempt to set the value of the **bpf_sock_ops_cb_flags** field
* for the full TCP socket associated to *bpf_sock_ops* to
* *argval*.
*
* The primary use of this field is to determine if there should
* be calls to eBPF programs of type
* **BPF_PROG_TYPE_SOCKET_OPS** at various points in the TCP
* code. A program of the same type can change its value, per
* connection and as necessary, when the connection is
* established. This field is directly accessible for reading, but
* this helper must be used for updates in order to return an
* error if an eBPF program tries to set a callback that is not
* supported in the current kernel.
*
* *argval* is a flag array which can combine these flags:
*
* * **BPF_SOCKET_OPS_RTO_CB_FLAG** (retransmission time out)
* * **BPF_SOCKET_OPS_RETRANS_CB_FLAG** (retransmission)
* * **BPF_SOCKET_OPS_STATE_CB_FLAG** (TCP state change)
* * **BPF_SOCKET_OPS_RTT_CB_FLAG** (every RTT)
*
* Therefore,
* this function can be used to clear a callback flag by
* setting the appropriate bit to zero. e.g. to disable the RTO
* callback:
*
* **bpf_sock_ops_cb_flags_set(bpf_sock,**
* **bpf_sock->bpf_sock_ops_cb_flags & ~BPF_SOCKET_OPS_RTO_CB_FLAG)**
*
* Here are some examples of where one could call such eBPF
* program:
*
* * When RTO fires.
* * When a packet is retransmitted.
* * When the connection terminates.
* * When a packet is sent.
* * When a packet is received.
*

```

```

* Returns
* Code **EINVAL** if the socket is not a full TCP socket;
* otherwise, a positive number containing the bits that could not
* be set is returned (which comes down to 0 if all bits were set
* as required).
*/
static long (* const bpf_sock_ops_cb_flags_set)(struct bpf_sock_ops *bpf_sock, int argval) = (void *) 59;

/*
* bpf_msg_redirect_map
*
* This helper is used in programs implementing policies at the
* socket level. If the message *msg* is allowed to pass (i.e. if
* the verdict eBPF program returns **SK_PASS**), redirect it to
* the socket referenced by *map* (of type
* **BPF_MAP_TYPE_SOCKMAP**) at index *key*. Both ingress and
* egress interfaces can be used for redirection. The
* **BPF_F_INGRESS** value in *flags* is used to make the
* distinction (ingress path is selected if the flag is present,
* egress path otherwise). This is the only flag supported for now.
*
* Returns
* **SK_PASS** on success, or **SK_DROP** on error.
*/
static long (* const bpf_msg_redirect_map)(struct sk_msg_md *msg, void *map, __u32 key, __u64 flags) = (void *)
60;

/*
* bpf_msg_apply_bytes
*
* For socket policies, apply the verdict of the eBPF program to
* the next *bytes* (number of bytes) of message *msg*.
*
* For example, this helper can be used in the following cases:
*
* * A single **sendmsg**() or **sendfile**() system call
* contains multiple logical messages that the eBPF program is
* supposed to read and for which
it should apply a verdict.
* * An eBPF program only cares to read the first *bytes* of a
* *msg*. If the message has a large payload, then setting up
* and calling the eBPF program repeatedly for all bytes, even
* though the verdict is already known, would create unnecessary
* overhead.
*
* When called from within an eBPF program, the helper sets a
* counter internal to the BPF infrastructure, that is used to
* apply the last verdict to the next *bytes*. If *bytes* is

```

```

* smaller than the current data being processed from a
* **sendmsg**\ () or **sendfile**\ () system call, the first
* *bytes* will be sent and the eBPF program will be re-run with
* the pointer for start of data pointing to byte number *bytes*
* **+ 1**. If *bytes* is larger than the current data being
* processed, then the eBPF verdict will be applied to multiple
* **sendmsg**\ () or **sendfile**\ () calls until *bytes* are
* consumed.
*
* Note that if a socket closes with the
internal counter holding
* a non-zero value, this is not a problem because data is not
* being buffered for *bytes* and is sent as it is received.
*
* Returns
* 0
*/
static long (* const bpf_msg_apply_bytes)(struct sk_msg_md *msg, __u32 bytes) = (void *) 61;

/*
* bpf_msg_cork_bytes
*
* For socket policies, prevent the execution of the verdict eBPF
* program for message *msg* until *bytes* (byte number) have been
* accumulated.
*
* This can be used when one needs a specific number of bytes
* before a verdict can be assigned, even if the data spans
* multiple **sendmsg**\ () or **sendfile**\ () calls. The extreme
* case would be a user calling **sendmsg**\ () repeatedly with
* 1-byte long message segments. Obviously, this is bad for
* performance, but it is still valid. If the eBPF program needs
* *bytes* bytes to validate a header, this helper can be used to
* prevent the eBPF program to be called again until *bytes* have
* been accumulated.
*
* Returns
* 0
*/
static long (* const bpf_msg_cork_bytes)(struct sk_msg_md *msg, __u32 bytes) = (void *) 62;

/*
* bpf_msg_pull_data
*
* For socket policies, pull in non-linear data from user space
* for *msg* and set pointers *msg* **->data** and *msg*\
* **->data_end** to *start* and *end* bytes offsets into *msg*,
* respectively.

```

```

*
* If a program of type BPF_PROG_TYPE_SK_MSG is run on a
* *msg* it can only parse data that the (**data**, **data_end**)
* pointers have already consumed. For **sendmsg**\() hooks this
* is likely the first scatterlist element. But for calls relying
* on the **sendpage** handler (e.g. **sendfile**\() this will
* be the range (**0**, **0**) because the data is shared with
* user space and by default the objective is to avoid allowing
* user space to modify data while (or after) eBPF verdict is
* being decided. This helper can be used to pull in data and to
* set the start and end pointer to given values. Data will be
* copied if necessary (i.e. if data was not linear and if start
* and end pointers do not point to the same chunk).
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* All values for *flags* are reserved for future usage, and must
* be left at zero.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_msg_pull_data)(struct sk_msg_md *msg, __u32 start, __u32 end, __u64 flags) = (void *)
63;

/*
* bpf_bind
*
* Bind the socket associated to *ctx* to the address pointed by
* *addr*, of length *addr_len*. This allows for making outgoing
* connection from the desired IP address, which can be useful for
* example when all processes inside a cgroup should use one
* single IP address on a host
that has multiple IP configured.
*
* This helper works for IPv4 and IPv6, TCP and UDP sockets. The
* domain (*addr*\ **->sa_family**) must be **AF_INET** (or
**AF_INET6**). It's advised to pass zero port (**sin_port**
or **sin6_port**) which triggers IP_BIND_ADDRESS_NO_PORT-like
behavior and lets the kernel efficiently pick up an unused
port as long as 4-tuple is unique. Passing non-zero port might
lead to degraded performance.
*
* Returns

```

```

* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_bind)(struct bpf_sock_addr *ctx, struct sockaddr *addr, int addr_len) = (void *) 64;

/*
* bpf_xdp_adjust_tail
*
* Adjust (move) *xdp_md*\ **->data_end** by *delta* bytes. It is
* possible to both shrink and grow the packet tail.
* Shrink done via *delta* being a negative integer.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously
done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_xdp_adjust_tail)(struct xdp_md *xdp_md, int delta) = (void *) 65;

/*
* bpf_skb_get_xfrm_state
*
* Retrieve the XFRM state (IP transform framework, see also
* **ip-xfrm(8)**) at *index* in XFRM "security path" for *skb*.
*
* The retrieved value is stored in the **struct bpf_xfrm_state**
* pointed by *xfrm_state* and of length *size*.
*
* All values for *flags* are reserved for future usage, and must
* be left at zero.
*
* This helper is available only if the kernel was compiled with
* **CONFIG_XFRM** configuration option.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_get_xfrm_state)(struct __sk_buff *skb, __u32 index, struct bpf_xfrm_state *xfrm_state,
__u32 size, __u64
flags) = (void *) 66;

/*
* bpf_get_stack
*

```



```

* Return a user or a kernel stack in bpf program provided buffer.
* To achieve this, the helper needs *ctx*, which is a pointer
* to the context on which the tracing program is executed.
* To store the stacktrace, the bpf program provides *buf* with
* a nonnegative *size*.
*
* The last argument, *flags*, holds the number of stack frames to
* skip (from 0 to 255), masked with
* **BPF_F_SKIP_FIELD_MASK**. The next bits can be used to set
* the following flags:
*
* **BPF_F_USER_STACK**
* Collect a user space stack instead of a kernel stack.
* **BPF_F_USER_BUILD_ID**
* Collect (build_id, file_offset) instead of ips for user
* stack, only valid if **BPF_F_USER_STACK** is also
* specified.
*
* *file_offset* is an offset relative to the beginning
* of the executable or shared object file backing the vma
* which the *ip* falls in. It is *not* an offset relative
* to that object's base address. Accordingly,
it must be
* adjusted by adding (sh_addr - sh_offset), where
* sh_{addr,offset} correspond to the executable section
* containing *file_offset* in the object, for comparisons
* to symbols' st_value to be valid.
*
* **bpf_get_stack**() can collect up to
* **PERF_MAX_STACK_DEPTH** both kernel and user frames, subject
* to sufficient large buffer size. Note that
* this limit can be controlled with the **sysctl** program, and
* that it should be manually increased in order to profile long
* user stacks (such as stacks for Java programs). To do so, use:
*
* ::
*
* # sysctl kernel.perf_event_max_stack=<new value>
*
* Returns
* The non-negative copied *buf* length equal to or less than
* *size* on success, or a negative error in case of failure.
*/
static long (* const bpf_get_stack)(void *ctx, void *buf, __u32 size, __u64 flags) = (void *) 67;

/*
* bpf_skb_load_bytes_relative
*

```

```

* This helper is similar to bpf_skb_load_bytes() in that
* it provides an easy way to load len bytes from offset
* from the packet associated to skb, into the buffer pointed
* by to. The difference to bpf_skb_load_bytes() is that
* a fifth argument start_header exists in order to select a
* base offset to start from. start_header can be one of:
*
* BPF_HDR_START_MAC
* Base offset to load data from is skb's mac header.
* BPF_HDR_START_NET
* Base offset to load data from is skb's network header.
*
* In general, "direct packet access" is the preferred method to
* access packet data, however, this helper is in particular useful
* in socket filters where skb->data does not always point
* to the start of the mac header and where "direct packet access"
* is not available.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_load_bytes_relative)(const void *skb, __u32 offset, void *to, __u32 len, __u32
start_header) = (void *) 68;

/*
* bpf_fib_lookup
*
* Do FIB lookup in kernel tables using parameters in params.
* If lookup is successful and result shows packet is to be
* forwarded, the neighbor tables are searched for the nexthop.
* If successful (ie., FIB lookup shows forwarding and nexthop
* is resolved), the nexthop address is returned in ipv4_dst
* or ipv6_dst based on family, smac is set to mac address of
* egress device, dmac is set to nexthop mac address, rt_metric
* is set to metric from route (IPv4/IPv6 only), and ifindex
* is set to the device index of the nexthop from the FIB lookup.
*
* plen argument is the size of the passed in struct.
* flags argument can be a combination of one or more of the
* following values:
*
* BPF_FIB_LOOKUP_DIRECT
* Do a direct table lookup vs full lookup using FIB
* rules.
* BPF_FIB_LOOKUP_TBID
* Used with BPF_FIB_LOOKUP_DIRECT.
* Use the routing table ID present in params->tbid
* for the fib lookup.

```

```

* **BPF_FIB_LOOKUP_OUTPUT**
*   Perform lookup from an egress perspective (default is
*   ingress).
* **BPF_FIB_LOOKUP_SKIP_NEIGH**
*   Skip the neighbour table lookup. *params*->dmac
*   and *params*->smac will not be set as output. A common
*   use case is to call **bpf_redirect_neigh**() after
*   doing **bpf_fib_lookup**().
* **BPF_FIB_LOOKUP_SRC**
*   Derive and set source IP addr in *params*->ipv{4,6}_src
*   for the nexthop. If the src addr cannot be derived,
*   **BPF_FIB_LKUP_RET_NO_SRC_ADDR** is returned. In this
*   case, *params*->dmac and *params*->smac are not set either.
*
* *ctx* is either **struct xdp_md** for XDP programs or
* **struct sk_buff** tc cls_act programs.
*
* Returns
* * * < 0 if any input argument is invalid
* * * 0 on success (packet is forwarded, nexthop neighbor exists)
* * * > 0 one of **BPF_FIB_LKUP_RET_** codes explaining why the
*   packet is not forwarded or needs assist from full stack
*
* If lookup fails with BPF_FIB_LKUP_RET_FRAG_NEEDED,
* then the MTU
* was exceeded and output params->mtu_result contains the MTU.
*/
static long (* const bpf_fib_lookup)(void *ctx, struct bpf_fib_lookup *params, int plen, __u32 flags) = (void *) 69;

/*
* bpf_sock_hash_update
*
* Add an entry to, or update a sockhash *map* referencing sockets.
* The *skops* is used as a new value for the entry associated to
* *key*. *flags* is one of:
*
* **BPF_NOEXIST**
*   The entry for *key* must not exist in the map.
* **BPF_EXIST**
*   The entry for *key* must already exist in the map.
* **BPF_ANY**
*   No condition on the existence of the entry for *key*.
*
* If the *map* has eBPF programs (parser and verdict), those will
* be inherited by the socket being added. If the socket is
* already attached to eBPF programs, this results in an error.
*
* Returns

```

```

* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_sock_hash_update)(struct bpf_sock_ops *skops, void *map, void
*key, __u64 flags) = (void *) 70;

/*
* bpf_msg_redirect_hash
*
* This helper is used in programs implementing policies at the
* socket level. If the message *msg* is allowed to pass (i.e. if
* the verdict eBPF program returns **SK_PASS**), redirect it to
* the socket referenced by *map* (of type
* **BPF_MAP_TYPE_SOCKHASH**) using hash *key*. Both ingress and
* egress interfaces can be used for redirection. The
* **BPF_F_INGRESS** value in *flags* is used to make the
* distinction (ingress path is selected if the flag is present,
* egress path otherwise). This is the only flag supported for now.
*
* Returns
* **SK_PASS** on success, or **SK_DROP** on error.
*/
static long (* const bpf_msg_redirect_hash)(struct sk_msg_md *msg, void *map, void *key, __u64 flags) = (void *)
71;

/*
* bpf_sk_redirect_hash
*
* This helper is used in programs implementing policies at the
* skb socket level. If the sk_buff *skb* is allowed to pass (i.e.
* if the verdict eBPF
program returns **SK_PASS**), redirect it
* to the socket referenced by *map* (of type
* **BPF_MAP_TYPE_SOCKHASH**) using hash *key*. Both ingress and
* egress interfaces can be used for redirection. The
* **BPF_F_INGRESS** value in *flags* is used to make the
* distinction (ingress path is selected if the flag is present,
* egress otherwise). This is the only flag supported for now.
*
* Returns
* **SK_PASS** on success, or **SK_DROP** on error.
*/
static long (* const bpf_sk_redirect_hash)(struct __sk_buff *skb, void *map, void *key, __u64 flags) = (void *) 72;

/*
* bpf_lwt_push_encap
*
* Encapsulate the packet associated to *skb* within a Layer 3
* protocol header. This header is provided in the buffer at

```

- * address **hdr**, with **len** its size in bytes. **type** indicates
- * the protocol of the header and can be one of:
- *
- * ****BPF_LWT_ENCAP_SEG6****
- * IPv6 encapsulation with Segment Routing Header
- * (****struct ipv6_sr_hdr****). **hdr** only contains the SRH,
- * the IPv6 header is computed by the kernel.
- * ****BPF_LWT_ENCAP_SEG6_INLINE****
- * Only works if **skb** contains an IPv6 packet. Insert a
- * Segment Routing Header (****struct ipv6_sr_hdr****) inside
- * the IPv6 header.
- * ****BPF_LWT_ENCAP_IP****
- * IP encapsulation (GRE/GUE/IPIP/etc). The outer header
- * must be IPv4 or IPv6, followed by zero or more
- * additional headers, up to ****LWT_BPF_MAX_HEADROOM****
- * total bytes in all prepended headers. Please note that
- * if ****skb_is_gso**** (**skb**) is true, no more than two
- * headers can be prepended, and the inner header, if
- * present, should be either GRE or UDP/GUE.
- *
- * ****BPF_LWT_ENCAP_SEG6**** \ **types* can be called by BPF programs
- * of type ****BPF_PROG_TYPE_LWT_IN****; ****BPF_LWT_ENCAP_IP**** type can
- * be called by bpf programs of types ****BPF_PROG_TYPE_LWT_IN**** and
- * ****BPF_PROG_TYPE_LWT_XMIT****.
- *
- * A call to this helper is susceptible to change the underlying
- * packet buffer. Therefore, at load time, all checks on
- pointers
- * previously done by the verifier are invalidated and must be
- * performed again, if the helper is used in combination with
- * direct packet access.
- *
- * Returns
- * 0 on success, or a negative error in case of failure.
- */

```
static long (* const bpf_lwt_push_encap)(struct __sk_buff *skb, __u32 type, void *hdr, __u32 len) = (void *) 73;
```

/*

- * **bpf_lwt_seg6_store_bytes**
- *
- * Store **len** bytes from address **from** into the packet
- * associated to **skb**, at **offset**. Only the flags, tag and TLVs
- * inside the outermost IPv6 Segment Routing Header can be
- * modified through this helper.
- *
- * A call to this helper is susceptible to change the underlying
- * packet buffer. Therefore, at load time, all checks on pointers
- * previously done by the verifier are invalidated and must be

```

* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_lwt_seg6_store_bytes)(struct
__sk_buff *skb, __u32 offset, const void *from, __u32 len) = (void *) 74;

/*
* bpf_lwt_seg6_adjust_srh
*
* Adjust the size allocated to TLVs in the outermost IPv6
* Segment Routing Header contained in the packet associated to
* *skb*, at position *offset* by *delta* bytes. Only offsets
* after the segments are accepted. *delta* can be as well
* positive (growing) as negative (shrinking).
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_lwt_seg6_adjust_srh)(struct __sk_buff *skb, __u32 offset, __s32 delta) = (void *) 75;

/*
* bpf_lwt_seg6_action
*
* Apply an IPv6 Segment Routing action of type *action* to the
* packet associated to *skb*. Each action takes a parameter
* contained at address *param*, and of length *param_len* bytes.
* *action* can be one of:
*
* **SEG6_LOCAL_ACTION_END_X**
* End.X action: Endpoint with Layer-3 cross-connect.
* Type of *param*: **struct in6_addr**.
* **SEG6_LOCAL_ACTION_END_T**
* End.T action: Endpoint with specific IPv6 table lookup.
* Type of *param*: **int**.
* **SEG6_LOCAL_ACTION_END_B6**
* End.B6 action: Endpoint bound to an SRv6 policy.
* Type of *param*: **struct ipv6_sr_hdr**.
* **SEG6_LOCAL_ACTION_END_B6_ENCAP**
* End.B6.Encap action: Endpoint bound to an SRv6

```

```

* encapsulation policy.
* Type of *param*: **struct ipv6_sr_hdr**.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_lwt_seg6_action)(struct __sk_buff *skb, __u32 action, void *param, __u32 param_len) =
(void *) 76;

/*
* bpf_rc_repeat
*
* This helper is used in programs implementing IR decoding, to
* report a successfully decoded repeat key message. This delays
* the generation of a key up event for previously generated
* key down event.
*
* Some IR protocols like NEC have a special IR message for
* repeating last button, for when a button is held down.
*
* The *ctx* should point to the lirc sample as passed into
* the program.
*
* This helper is only available if the kernel was compiled with
* the **CONFIG_BPF_LIRC_MODE2** configuration option set to
* **y**.
*
* Returns
* 0
*/
static long (* const bpf_rc_repeat)(void *ctx) = (void *) 77;

/*
* bpf_rc_keydown
*
* This helper is used in programs implementing IR decoding, to
* report a successfully decoded key
press with *scancode*,
* *toggle* value in the given *protocol*. The scancode will be
* translated to a keycode using the rc keymap, and reported as
* an input key down event. After a period a key up event is
* generated. This period can be extended by calling either

```

```

* **bpf_rc_keydown**\ () again with the same values, or calling
* **bpf_rc_repeat**\ ().
*
* Some protocols include a toggle bit, in case the button was
* released and pressed again between consecutive scancodes.
*
* The *ctx* should point to the lirc sample as passed into
* the program.
*
* The *protocol* is the decoded protocol number (see
* **enum rc_proto** for some predefined values).
*
* This helper is only available if the kernel was compiled with
* the **CONFIG_BPF_LIRC_MODE2** configuration option set to
* "y".
*
* Returns
* 0
*/
static long (* const bpf_rc_keydown)(void *ctx, __u32 protocol, __u64 scancode, __u32 toggle) = (void *) 78;

/*
* bpf_skb_cgroup_id
*
* Return
* the cgroup v2 id of the socket associated with the *skb*.
* This is roughly similar to the **bpf_get_cgroup_classid**\ ()
* helper for cgroup v1 by providing a tag resp. identifier that
* can be matched on or used for map lookups e.g. to implement
* policy. The cgroup v2 id of a given path in the hierarchy is
* exposed in user space through the f_handle API in order to get
* to the same 64-bit id.
*
* This helper can be used on TC egress path, but not on ingress,
* and is available only if the kernel was compiled with the
* **CONFIG_SOCK_CGROUP_DATA** configuration option.
*
* Returns
* The id is returned or 0 in case the id could not be retrieved.
*/
static __u64 (* const bpf_skb_cgroup_id)(struct __sk_buff *skb) = (void *) 79;

/*
* bpf_get_current_cgroup_id
*
* Get the current cgroup id based on the cgroup within which
* the current task is running.
*

```



```

* Returns
* A 64-bit integer containing the current cgroup id based
* on the cgroup within which
  the current task is running.
*/
static __u64 (* const bpf_get_current_cgroup_id)(void) = (void *) 80;

/*
* bpf_get_local_storage
*
* Get the pointer to the local storage area.
* The type and the size of the local storage is defined
* by the *map* argument.
* The *flags* meaning is specific for each map type,
* and has to be 0 for cgroup local storage.
*
* Depending on the BPF program type, a local storage area
* can be shared between multiple instances of the BPF program,
* running simultaneously.
*
* A user should care about the synchronization by himself.
* For example, by using the **BPF_ATOMIC** instructions to alter
* the shared data.
*
* Returns
* A pointer to the local storage area.
*/
static void *(* const bpf_get_local_storage)(void *map, __u64 flags) = (void *) 81;

/*
* bpf_sk_select_reuseport
*
* Select a **SO_REUSEPORT** socket from a
* **BPF_MAP_TYPE_REUSEPORT_SOCKARRAY** *map*.
* It checks the selected socket is matching the incoming
* request in the socket buffer.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_sk_select_reuseport)(struct sk_reuseport_md *reuse, void *map, void *key, __u64 flags) =
(void *) 82;

/*
* bpf_skb_ancestor_cgroup_id
*
* Return id of cgroup v2 that is ancestor of cgroup associated
* with the *skb* at the *ancestor_level*. The root cgroup is at

```

```

* *ancestor_level* zero and each step down the hierarchy
* increments the level. If *ancestor_level* == level of cgroup
* associated with *skb*, then return value will be same as that
* of bpf_skb_cgroup_id ().
*
* The helper is useful to implement policies based on cgroups
* that are upper in hierarchy than immediate cgroup associated
* with *skb*.
*
* The format of returned id and helper limitations are same as in
* bpf_skb_cgroup_id ().
*
* Returns
* The id is returned or 0 in case the id could not be retrieved.
*/
static __u64 (* const bpf_skb_ancestor_cgroup_id)(struct
__sk_buff *skb, int ancestor_level) = (void *) 83;

/*
* bpf_sk_lookup_tcp
*
* Look for TCP socket matching *tuple*, optionally in a child
* network namespace *netns*. The return value must be checked,
* and if non-NULL, released via bpf_sk_release ().
*
* The *ctx* should point to the context of the program, such as
* the skb or socket (depending on the hook in use). This is used
* to determine the base network namespace for the lookup.
*
* *tuple_size* must be one of:
*
* sizeof (*tuple* -> ipv4)
* Look for an IPv4 socket.
* sizeof (*tuple* -> ipv6)
* Look for an IPv6 socket.
*
* If the *netns* is a negative signed 32-bit integer, then the
* socket lookup table in the netns associated with the *ctx*
* will be used. For the TC hooks, this is the netns of the device
* in the skb. For socket hooks, this is the netns of the socket.
* If *netns* is any other signed 32-bit value greater than or
* equal to zero then it specifies the ID of the netns relative to
* the netns associated with the *ctx*. *netns* values beyond the
* range of 32-bit integers are reserved for future use.
*
* All values for *flags* are reserved for future usage, and must
* be left at zero.
*

```

```

* This helper is available only if the kernel was compiled with
* **CONFIG_NET** configuration option.
*
* Returns
* Pointer to **struct bpf_sock**, or **NULL** in case of failure.
* For sockets with reuseport option, the **struct bpf_sock**
* result is from *reuse*\ **->socks**\ [] using the hash of the
* tuple.
*/
static struct bpf_sock *(* const bpf_sk_lookup_tcp)(void *ctx, struct bpf_sock_tuple *tuple, __u32 tuple_size,
__u64 netns, __u64 flags) = (void *) 84;

/*
* bpf_sk_lookup_udp
*
* Look for UDP socket matching *tuple*, optionally in a child
* network namespace *netns*. The return value must be checked,
* and if non-**NULL**, released via **bpf_sk_release**\ ().
*
* The
* *ctx* should point to the context of the program, such as
* the skb or socket (depending on the hook in use). This is used
* to determine the base network namespace for the lookup.
*
* *tuple_size* must be one of:
*
* **sizeof**\ (*tuple*\ **->ipv4**)
* Look for an IPv4 socket.
* **sizeof**\ (*tuple*\ **->ipv6**)
* Look for an IPv6 socket.
*
* If the *netns* is a negative signed 32-bit integer, then the
* socket lookup table in the netns associated with the *ctx*
* will be used. For the TC hooks, this is the netns of the device
* in the skb. For socket hooks, this is the netns of the socket.
* If *netns* is any other signed 32-bit value greater than or
* equal to zero then it specifies the ID of the netns relative to
* the netns associated with the *ctx*. *netns* values beyond the
* range of 32-bit integers are reserved for future use.
*
* All values for *flags* are reserved for future usage, and must
* be left at zero.
*
* This helper is available
* only if the kernel was compiled with
* **CONFIG_NET** configuration option.
*
* Returns

```

```

* Pointer to **struct bpf_sock**, or **NULL** in case of failure.
* For sockets with reuseport option, the **struct bpf_sock**
* result is from *reuse*\ **->socks**\ [] using the hash of the
* tuple.
*/
static struct bpf_sock>(* const bpf_sk_lookup_udp)(void *ctx, struct bpf_sock_tuple *tuple, __u32 tuple_size,
__u64 netns, __u64 flags) = (void *) 85;

/*
* bpf_sk_release
*
* Release the reference held by *sock*. *sock* must be a
* non-**NULL** pointer that was returned from
* **bpf_sk_lookup_xxx**\ ().
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_sk_release)(void *sock) = (void *) 86;

/*
* bpf_map_push_elem
*
* Push an element *value* in *map*. *flags* is one of:
*
* **BPF_EXIST**
* If the queue/stack is full, the oldest element is
* removed to make room for this.
*
* Returns
* 0 on success, or
a negative error in case of failure.
*/
static long (* const bpf_map_push_elem)(void *map, const void *value, __u64 flags) = (void *) 87;

/*
* bpf_map_pop_elem
*
* Pop an element from *map*.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_map_pop_elem)(void *map, void *value) = (void *) 88;

/*
* bpf_map_peek_elem
*

```

```

* Get an element from *map* without removing it.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_map_peek_elem)(void *map, void *value) = (void *) 89;

/*
* bpf_msg_push_data
*
* For socket policies, insert *len* bytes into *msg* at offset
* *start*.
*
* If a program of type **BPF_PROG_TYPE_SK_MSG** is run on a
* *msg* it may want to insert metadata or options into the *msg*.
* This can later be read and used by any of the lower layer BPF
* hooks.
*
* This helper may fail if under memory pressure (a malloc
* fails) in these cases BPF programs
will get an appropriate
* error and BPF programs will need to handle them.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_msg_push_data)(struct sk_msg_md *msg, __u32 start, __u32 len, __u64 flags) = (void *)
90;

/*
* bpf_msg_pop_data
*
* Will remove *len* bytes from a *msg* starting at byte *start*.
* This may result in **ENOMEM** errors under certain situations if
* an allocation and copy are required due to a full ring buffer.
* However, the helper will try to avoid doing the allocation
* if possible. Other errors can occur if input parameters are
* invalid either due to *start* byte not being valid part of *msg*
* payload and/or *pop* value being too large.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_msg_pop_data)(struct sk_msg_md *msg, __u32 start, __u32 len, __u64 flags) = (void *) 91;

/*
* bpf_rc_pointer_rel
*

```

```

* This helper is used in programs implementing
IR decoding, to
* report a successfully decoded pointer movement.
*
* The *ctx* should point to the lirc sample as passed into
the program.
*
* This helper is only available if the kernel was compiled with
the CONFIG_BPF_LIRC_MODE2 configuration option set to
Y.
*
* Returns
* 0
*/
static long (* const bpf_rc_pointer_rel)(void *ctx, __s32 rel_x, __s32 rel_y) = (void *) 92;

/*
* bpf_spin_lock
*
* Acquire a spinlock represented by the pointer *lock*, which is
stored as part of a value of a map. Taking the lock allows to
safely update the rest of the fields in that value. The
spinlock can (and must) later be released with a call to
bpf_spin_unlock(\ *lock\ ).
*
* Spinlocks in BPF programs come with a number of restrictions
and constraints:
*
* bpf_spin_lock objects are only allowed inside maps of
types BPF_MAP_TYPE_HASH and BPF_MAP_TYPE_ARRAY (this
list could be extended in the future).
* BTF description of the map is mandatory.
* The BPF program can take ONE lock at a time, since taking two
or more could cause dead locks.
* Only one struct bpf_spin_lock is allowed per map element.
* When the lock is taken, calls (either BPF to BPF or helpers)
are not allowed.
* The BPF_LD_ABS and BPF_LD_IND instructions are not
allowed inside a spinlock-ed region.
* The BPF program MUST call bpf_spin_unlock() to release
the lock, on all execution paths, before it returns.
* The BPF program can access struct bpf_spin_lock only via
the bpf_spin_lock() and bpf_spin_unlock()
helpers. Loading or storing data into the struct
bpf_spin_lock *lock field of a map is not allowed.
* To use the bpf_spin_lock() helper, the BTF description
of the map value must be a struct and have struct
bpf_spin_lock *anyname field at the top level.

```

```

* Nested
lock inside another struct is not allowed.
* * The **struct bpf_spin_lock** *lock* field in a map value must
* be aligned on a multiple of 4 bytes in that value.
* * Syscall with command **BPF_MAP_LOOKUP_ELEM** does not copy
* the **bpf_spin_lock** field to user space.
* * Syscall with command **BPF_MAP_UPDATE_ELEM**, or update from
* a BPF program, do not update the **bpf_spin_lock** field.
* * **bpf_spin_lock** cannot be on the stack or inside a
* networking packet (it can only be inside of a map values).
* * **bpf_spin_lock** is available to root only.
* * Tracing programs and socket filter programs cannot use
* **bpf_spin_lock**\ () due to insufficient preemption checks
* (but this may change in the future).
* * **bpf_spin_lock** is not allowed in inner maps of map-in-map.
*
* Returns
* 0
*/
static long (* const bpf_spin_lock)(struct bpf_spin_lock *lock) = (void *) 93;

/*
* bpf_spin_unlock
*
* Release the *lock* previously locked
by a call to
* **bpf_spin_lock**\ (\ *lock*\ ).
*
* Returns
* 0
*/
static long (* const bpf_spin_unlock)(struct bpf_spin_lock *lock) = (void *) 94;

/*
* bpf_sk_fullsock
*
* This helper gets a **struct bpf_sock** pointer such
* that all the fields in this **bpf_sock** can be accessed.
*
* Returns
* A **struct bpf_sock** pointer on success, or **NULL** in
* case of failure.
*/
static struct bpf_sock *(* const bpf_sk_fullsock)(struct bpf_sock *sk) = (void *) 95;

/*
* bpf_tcp_sock
*

```

```

* This helper gets a struct bpf_tcp_sock pointer from a
* struct bpf_sock pointer.
*
* Returns
* A struct bpf_tcp_sock pointer on success, or NULL in
* case of failure.
*/
static struct bpf_tcp_sock *(* const bpf_tcp_sock)(struct bpf_sock *sk) = (void *) 96;

/*
* bpf_skb_ecn_set_ce
*
* Set ECN (Explicit Congestion Notification) field of IP header
* to CE (Congestion Encountered) if current value is ECT
* (ECN Capable Transport). Otherwise, do nothing.
Works with IPv6
* and IPv4.
*
* Returns
* 1 if the CE flag is set (either by the current helper call
* or because it was already present), 0 if it is not set.
*/
static long (* const bpf_skb_ecn_set_ce)(struct __sk_buff *skb) = (void *) 97;

/*
* bpf_get_listener_sock
*
* Return a struct bpf_sock pointer in TCP_LISTEN state.
* bpf_sk_release() is unnecessary and not allowed.
*
* Returns
* A struct bpf_sock pointer on success, or NULL in
* case of failure.
*/
static struct bpf_sock *(* const bpf_get_listener_sock)(struct bpf_sock *sk) = (void *) 98;

/*
* bpf_skc_lookup_tcp
*
* Look for TCP socket matching tuple, optionally in a child
* network namespace netns. The return value must be checked,
* and if non-NULL, released via bpf_sk_release().
*
* This function is identical to bpf_sk_lookup_tcp(), except
* that it also returns timewait or request sockets. Use
* bpf_sk_fullsock() or bpf_tcp_sock
* () to access the
* full structure.

```



```

*
* This helper is available only if the kernel was compiled with
* **CONFIG_NET** configuration option.
*
* Returns
* Pointer to **struct bpf_sock**, or **NULL** in case of failure.
* For sockets with reuseport option, the **struct bpf_sock**
* result is from *reuse* \ **->socks** \ [] using the hash of the
* tuple.
*/
static struct bpf_sock *(* const bpf_skc_lookup_tcp)(void *ctx, struct bpf_sock_tuple *tuple, __u32 tuple_size,
__u64 netns, __u64 flags) = (void *) 99;

/*
* bpf_tcp_check_syncookie
*
* Check whether *iph* and *th* contain a valid SYN cookie ACK for
* the listening socket in *sk*.
*
* *iph* points to the start of the IPv4 or IPv6 header, while
* *iph_len* contains **sizeof** \ (**struct iphdr**) or
* **sizeof** \ (**struct ipv6hdr**).
*
* *th* points to the start of the TCP header, while *th_len*
* contains the length of the TCP header (at least
* **sizeof** \ (**struct tcphdr**)).
*
* Returns
* 0 if *iph* and *th* are a valid SYN cookie ACK, or a negative
* error otherwise.
*/
static long (* const bpf_tcp_check_syncookie)(void *sk, void *iph, __u32 iph_len, struct tcphdr *th, __u32 th_len)
= (void *) 100;

/*
* bpf_sysctl_get_name
*
* Get name of sysctl in /proc/sys/ and copy it into provided by
* program buffer *buf* of size *buf_len*.
*
* The buffer is always NUL terminated, unless it's zero-sized.
*
* If *flags* is zero, full name (e.g. "net/ipv4/tcp_mem") is
* copied. Use **BPF_F_SYSCTL_BASE_NAME** flag to copy base name
* only (e.g. "tcp_mem").
*
* Returns
* Number of character copied (not including the trailing NUL).

```

```

*
* **_E2BIG** if the buffer wasn't big enough (*buf* will contain
* truncated name in this case).
*/
static long (* const bpf_sysctl_get_name)(struct bpf_sysctl *ctx, char *buf, unsigned long buf_len, __u64 flags) =
(void *) 101;

/*
* bpf_sysctl_get_current_value
*
* Get current value of sysctl as it is presented in
/proc/sys
* (incl. newline, etc), and copy it as a string into provided
* by program buffer *buf* of size *buf_len*.
*
* The whole value is copied, no matter what file position user
* space issued e.g. sys_read at.
*
* The buffer is always NUL terminated, unless it's zero-sized.
*
* Returns
* Number of character copied (not including the trailing NUL).
*
* **_E2BIG** if the buffer wasn't big enough (*buf* will contain
* truncated name in this case).
*
* **_EINVAL** if current value was unavailable, e.g. because
* sysctl is uninitialized and read returns -EIO for it.
*/
static long (* const bpf_sysctl_get_current_value)(struct bpf_sysctl *ctx, char *buf, unsigned long buf_len) = (void
*) 102;

/*
* bpf_sysctl_get_new_value
*
* Get new value being written by user space to sysctl (before
* the actual write happens) and copy it as a string into
* provided by program buffer *buf* of size *buf_len*.
*
* User space may write new value at file position >
0.
*
* The buffer is always NUL terminated, unless it's zero-sized.
*
* Returns
* Number of character copied (not including the trailing NUL).
*
* **_E2BIG** if the buffer wasn't big enough (*buf* will contain

```

```

* truncated name in this case).
*
* **EINVAL** if sysctl is being read.
*/
static long (* const bpf_sysctl_get_new_value)(struct bpf_sysctl *ctx, char *buf, unsigned long buf_len) = (void *)
103;

/*
* bpf_sysctl_set_new_value
*
* Override new value being written by user space to sysctl with
* value provided by program in buffer *buf* of size *buf_len*.
*
* *buf* should contain a string in same form as provided by user
* space on sysctl write.
*
* User space may write new value at file position > 0. To override
* the whole sysctl value file position should be set to zero.
*
* Returns
* 0 on success.
*
* **E2BIG** if the *buf_len* is too big.
*
* **EINVAL** if sysctl is being read.
*/
static long (* const bpf_sysctl_set_new_value)(struct
bpf_sysctl *ctx, const char *buf, unsigned long buf_len) = (void *) 104;

/*
* bpf_strtol
*
* Convert the initial part of the string from buffer *buf* of
* size *buf_len* to a long integer according to the given base
* and save the result in *res*.
*
* The string may begin with an arbitrary amount of white space
* (as determined by **isspace**\ (3)) followed by a single
* optional '**_**' sign.
*
* Five least significant bits of *flags* encode base, other bits
* are currently unused.
*
* Base must be either 8, 10, 16 or 0 to detect it automatically
* similar to user space **strtol**\ (3).
*
* Returns
* Number of characters consumed on success. Must be positive but

```

```

* no more than *buf_len*.
*
* **EINVAL** if no valid digits were found or unsupported base
* was provided.
*
* **ERANGE** if resulting value was out of range.
*/
static long (* const bpf_strtol)(const char *buf, unsigned long buf_len, __u64 flags, long *res) = (void *) 105;

/*
*
* bpf_strtoul
*
* Convert the initial part of the string from buffer *buf* of
* size *buf_len* to an unsigned long integer according to the
* given base and save the result in *res*.
*
* The string may begin with an arbitrary amount of white space
* (as determined by **isspace**\ (3)).
*
* Five least significant bits of *flags* encode base, other bits
* are currently unused.
*
* Base must be either 8, 10, 16 or 0 to detect it automatically
* similar to user space **strtoul**\ (3).
*
* Returns
* Number of characters consumed on success. Must be positive but
* no more than *buf_len*.
*
* **EINVAL** if no valid digits were found or unsupported base
* was provided.
*
* **ERANGE** if resulting value was out of range.
*/
static long (* const bpf_strtoul)(const char *buf, unsigned long buf_len, __u64 flags, unsigned long *res) = (void *)
106;

/*
* bpf_sk_storage_get
*
* Get a bpf-local-storage from a *sk*.
*
* Logically, it could be thought of getting
the value from
* a *map* with *sk* as the **key**. From this
* perspective, the usage is not much different from
* **bpf_map_lookup_elem**\ (*map*, **&sk*) except this

```

```

* helper enforces the key must be a full socket and the map must
* be a BPF_MAP_TYPE_SK_STORAGE also.
*
* Underneath, the value is stored locally at *sk* instead of
* the *map*. The *map* is used as the bpf-local-storage
* "type". The bpf-local-storage "type" (i.e. the *map*) is
* searched against all bpf-local-storages residing at *sk*.
*
* *sk* is a kernel struct sock pointer for LSM program.
* *sk* is a struct bpf_sock pointer for other program types.
*
* An optional *flags* (BPF_SK_STORAGE_GET_F_CREATE) can be
* used such that a new bpf-local-storage will be
* created if one does not exist. *value* can be used
* together with BPF_SK_STORAGE_GET_F_CREATE to specify
* the initial value of a bpf-local-storage. If *value* is
* NULL, the new bpf-local-storage
will be zero initialized.
*
* Returns
* A bpf-local-storage pointer is returned on success.
*
* NULL if not found or there was an error in adding
* a new bpf-local-storage.
*/
static void (* const bpf_sk_storage_get)(void *map, void *sk, void *value, __u64 flags) = (void *) 107;

/*
* bpf_sk_storage_delete
*
* Delete a bpf-local-storage from a *sk*.
*
* Returns
* 0 on success.
*
* -ENOENT if the bpf-local-storage cannot be found.
* -EINVAL if sk is not a fullsock (e.g. a request_sock).
*/
static long (* const bpf_sk_storage_delete)(void *map, void *sk) = (void *) 108;

/*
* bpf_send_signal
*
* Send signal *sig* to the process of the current task.
* The signal may be delivered to any of this process's threads.
*
* Returns
* 0 on success or successfully queued.

```

```

*
* ** -EBUSY** if work queue under nmi is full.
*
* ** -EINVAL** if *sig* is invalid.
*
* ** -EPERM** if no permission to send the *sig*.
*
* ** -EAGAIN** if bpf
program can try again.
*/
static long (* const bpf_send_signal)(__u32 sig) = (void *) 109;

/*
* bpf_tcp_gen_syncookie
*
* Try to issue a SYN cookie for the packet with corresponding
* IP/TCP headers, *iph* and *th*, on the listening socket in *sk*.
*
* *iph* points to the start of the IPv4 or IPv6 header, while
* *iph_len* contains **sizeof**\ (**struct iphdr**) or
* **sizeof**\ (**struct ipv6hdr**).
*
* *th* points to the start of the TCP header, while *th_len*
* contains the length of the TCP header with options (at least
* **sizeof**\ (**struct tcphdr**)).
*
* Returns
* On success, lower 32 bits hold the generated SYN cookie in
* followed by 16 bits which hold the MSS value for that cookie,
* and the top 16 bits are unused.
*
* On failure, the returned value is one of the following:
*
* ** -EINVAL** SYN cookie cannot be issued due to error
*
* ** -ENOENT** SYN cookie should not be issued (no SYN flood)
*
* ** -EOPNOTSUPP** kernel configuration
does not enable SYN cookies
*
* ** -EPROTONOSUPPORT** IP packet version is not 4 or 6
*/
static __s64 (* const bpf_tcp_gen_syncookie)(void *sk, void *iph, __u32 iph_len, struct tcphdr *th, __u32 th_len) =
(void *) 110;

/*
* bpf_skb_output
*

```

```

* Write raw *data* blob into a special BPF perf event held by
* *map* of type **BPF_MAP_TYPE_PERF_EVENT_ARRAY**. This perf
* event must have the following attributes: **PERF_SAMPLE_RAW**
* as **sample_type**, **PERF_TYPE_SOFTWARE** as **type**, and
* **PERF_COUNT_SW_BPF_OUTPUT** as **config**.
*
* The *flags* are used to indicate the index in *map* for which
* the value must be put, masked with **BPF_F_INDEX_MASK**.
* Alternatively, *flags* can be set to **BPF_F_CURRENT_CPU**
* to indicate that the index of the current CPU core should be
* used.
*
* The value to write, of *size*, is passed through eBPF stack and
* pointed by *data*.
*
* *ctx* is a pointer to in-kernel struct sk_buff.
*
* This helper is similar to
**bpf_perf_event_output**() but
* restricted to raw_tracepoint bpf programs.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_output)(void *ctx, void *map, __u64 flags, void *data, __u64 size) = (void *) 111;

/*
* bpf_probe_read_user
*
* Safely attempt to read *size* bytes from user space address
* *unsafe_ptr* and store the data in *dst*.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_probe_read_user)(void *dst, __u32 size, const void *unsafe_ptr) = (void *) 112;

/*
* bpf_probe_read_kernel
*
* Safely attempt to read *size* bytes from kernel space address
* *unsafe_ptr* and store the data in *dst*.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_probe_read_kernel)(void *dst, __u32 size, const void *unsafe_ptr) = (void *) 113;

```

```

/*
 * bpf_probe_read_user_str
 *
 * Copy a NUL terminated string from an unsafe
 * user address
 * *unsafe_ptr* to *dst*. The *size* should include the
 * terminating NUL byte. In case the string length is smaller than
 * *size*, the target is not padded with further NUL bytes. If the
 * string length is larger than *size*, just *size*-1 bytes are
 * copied and the last byte is set to NUL.
 *
 * On success, returns the number of bytes that were written,
 * including the terminal NUL. This makes this helper useful in
 * tracing programs for reading strings, and more importantly to
 * get its length at runtime. See the following snippet:
 *
 * ::
 *
 * SEC("kprobe/sys_open")
 * void bpf_sys_open(struct pt_regs *ctx)
 * {
 *     char buf[PATHLEN]; // PATHLEN is defined to 256
 *     int res = bpf_probe_read_user_str(buf, sizeof(buf),
 *                                     ctx->di);
 *
 *     // Consume buf, for example push it to
 *     // userspace via bpf_perf_event_output(); we
 *     // can use res (the string length) as event
 *     // size,
 *     after checking its boundaries.
 * }
 *
 * In comparison, using **bpf_probe_read_user**\ () helper here
 * instead to read the string would require to estimate the length
 * at compile time, and would often result in copying more memory
 * than necessary.
 *
 * Another useful use case is when parsing individual process
 * arguments or individual environment variables navigating
 * *current*\ **->mm->arg_start** and *current*\
 * **->mm->env_start** using this helper and the return value,
 * one can quickly iterate at the right offset of the memory area.
 *
 * Returns
 * On success, the strictly positive length of the output string,
 * including the trailing NUL character. On error, a negative
 * value.
 */

```



```
static long (* const bpf_probe_read_user_str)(void *dst, __u32 size, const void *unsafe_ptr) = (void *) 114;
```

```
/*
```

```
* bpf_probe_read_kernel_str
```

```
*
```

```
* Copy a NUL terminated string from an unsafe kernel address *unsafe_ptr*
```

```
* to *dst*. Same semantics as with **bpf_probe_read_user_str**\
```

```
() apply.
```

```
*
```

```
* Returns
```

```
* On success, the strictly positive length of the string, including
```

```
* the trailing NUL character. On error, a negative value.
```

```
*/
```

```
static long (* const bpf_probe_read_kernel_str)(void *dst, __u32 size, const void *unsafe_ptr) = (void *) 115;
```

```
/*
```

```
* bpf_tcp_send_ack
```

```
*
```

```
* Send out a tcp-ack. *tp* is the in-kernel struct **tcp_sock**.
```

```
* *rcv_nxt* is the ack_seq to be sent out.
```

```
*
```

```
* Returns
```

```
* 0 on success, or a negative error in case of failure.
```

```
*/
```

```
static long (* const bpf_tcp_send_ack)(void *tp, __u32 rcv_nxt) = (void *) 116;
```

```
/*
```

```
* bpf_send_signal_thread
```

```
*
```

```
* Send signal *sig* to the thread corresponding to the current task.
```

```
*
```

```
* Returns
```

```
* 0 on success or successfully queued.
```

```
*
```

```
* **-EBUSY** if work queue under nmi is full.
```

```
*
```

```
* **-EINVAL** if *sig* is invalid.
```

```
*
```

```
* **-EPERM** if no permission to send the *sig*.
```

```
*
```

```
* **-EAGAIN** if bpf program can try again.
```

```
*/
```

```
static long (* const bpf_send_signal_thread)(__u32 sig) = (void *) 117;
```

```
/*
```

```
* bpf_jiffies64
```

```
*
```

```

* Obtain the 64bit jiffies
*
* Returns
* The 64 bit jiffies
*/
static __u64 (* const bpf_jiffies64)(void) = (void *) 118;

/*
* bpf_read_branch_records
*
* For an eBPF program attached to a perf event, retrieve the
* branch records (**struct perf_branch_entry**) associated to *ctx*
* and store it in the buffer pointed by *buf* up to size
* *size* bytes.
*
* Returns
* On success, number of bytes written to *buf*. On error, a
* negative value.
*
* The *flags* can be set to **BPF_F_GET_BRANCH_RECORDS_SIZE** to
* instead return the number of bytes required to store all the
* branch entries. If this flag is set, *buf* may be NULL.
*
* **EINVAL** if arguments invalid or **size** not a multiple
* of **sizeof**\ (**struct perf_branch_entry**\ ).
*
* **ENOENT** if architecture does not support branch records.
*/
static long (* const bpf_read_branch_records)(struct bpf_perf_event_data *ctx, void *buf, __u32 size, __u64
flags) = (void *) 119;

/*
* bpf_get_ns_current_pid_tgid
*
* Returns 0 on success, values for *pid* and *tgid* as seen from the current
* *namespace* will be returned in *nsdata*.
*
* Returns
* 0 on success, or one of the following in case of failure:
*
* **EINVAL** if dev and inum supplied don't match dev_t and inode number
* with nsfs of current task, or if dev conversion to dev_t lost high bits.
*
* **ENOENT** if pidns does not exists for the current task.
*/
static long (* const bpf_get_ns_current_pid_tgid)(__u64 dev, __u64 ino, struct bpf_pidns_info *nsdata, __u32 size)
= (void *) 120;

```

```

/*
 * bpf_xdp_output
 *
 * Write raw *data* blob into a special BPF perf event held by
 * *map* of type **BPF_MAP_TYPE_PERF_EVENT_ARRAY**. This perf
 * event must have the following attributes: **PERF_SAMPLE_RAW**
 * as **sample_type**, **PERF_TYPE_SOFTWARE** as **type**, and
 * **PERF_COUNT_SW_BPF_OUTPUT** as **config**.
 *
 * The *flags* are used to indicate the index
 * in *map* for which
 * the value must be put, masked with **BPF_F_INDEX_MASK**.
 * Alternatively, *flags* can be set to **BPF_F_CURRENT_CPU**
 * to indicate that the index of the current CPU core should be
 * used.
 *
 * The value to write, of *size*, is passed through eBPF stack and
 * pointed by *data*.
 *
 * *ctx* is a pointer to in-kernel struct xdp_buff.
 *
 * This helper is similar to **bpf_perf_event_output**\ () but
 * restricted to raw_tracepoint bpf programs.
 *
 * Returns
 * 0 on success, or a negative error in case of failure.
 */
static long (* const bpf_xdp_output)(void *ctx, void *map, __u64 flags, void *data, __u64 size) = (void *) 121;

/*
 * bpf_get_netns_cookie
 *
 * Retrieve the cookie (generated by the kernel) of the network
 * namespace the input *ctx* is associated with. The network
 * namespace cookie remains stable for its lifetime and provides
 * a global identifier that can be assumed unique. If *ctx* is
 * NULL, then the helper returns the cookie
 * for the initial
 * network namespace. The cookie itself is very similar to that
 * of **bpf_get_socket_cookie**\ () helper, but for network
 * namespaces instead of sockets.
 *
 * Returns
 * A 8-byte long opaque number.
 */
static __u64 (* const bpf_get_netns_cookie)(void *ctx) = (void *) 122;

/*

```

```

* bpf_get_current_ancestor_cgroup_id
*
* Return id of cgroup v2 that is ancestor of the cgroup associated
* with the current task at the *ancestor_level*. The root cgroup
* is at *ancestor_level* zero and each step down the hierarchy
* increments the level. If *ancestor_level* == level of cgroup
* associated with the current task, then return value will be the
* same as that of bpf_get_current_cgroup_id() .
*
* The helper is useful to implement policies based on cgroups
* that are upper in hierarchy than immediate cgroup associated
* with the current task.
*
* The format of returned id and helper limitations are same as in
* bpf_get_current_cgroup_id() .
*
* Returns
* The id is returned or 0 in case the id could not be retrieved.
*/
static __u64 (* const bpf_get_current_ancestor_cgroup_id)(int ancestor_level) = (void *) 123;

/*
* bpf_sk_assign
*
* Helper is overloaded depending on BPF program type. This
* description applies to BPF_PROG_TYPE_SCHED_CLS and
* BPF_PROG_TYPE_SCHED_ACT programs.
*
* Assign the *sk* to the *skb*. When combined with appropriate
* routing configuration to receive the packet towards the socket,
* will cause *skb* to be delivered to the specified socket.
* Subsequent redirection of *skb* via bpf_redirect (),
* bpf_clone_redirect () or other methods outside of BPF may
* interfere with successful delivery to the socket.
*
* This operation is only valid from TC ingress path.
*
* The *flags* argument must be zero.
*
* Returns
* 0 on success, or a negative error in case of failure:
*
* -EINVAL if specified *flags* are not supported.
*
* -ENOENT
if the socket is unavailable for assignment.
*

```

```

* **ENETUNREACH** if the socket is unreachable (wrong netns).
*
* **EOPNOTSUPP** if the operation is not supported, for example
* a call from outside of TC ingress.
*/
static long (* const bpf_sk_assign)(void *ctx, void *sk, __u64 flags) = (void *) 124;

/*
* bpf_ktime_get_boot_ns
*
* Return the time elapsed since system boot, in nanoseconds.
* Does include the time the system was suspended.
* See: **clock_gettime**\ (**CLOCK_BOOTTIME**)
*
* Returns
* Current *ktime*.
*/
static __u64 (* const bpf_ktime_get_boot_ns)(void) = (void *) 125;

/*
* bpf_seq_printf
*
* **bpf_seq_printf**\ () uses seq_file **seq_printf**\ () to print
* out the format string.
* The *m* represents the seq_file. The *fmt* and *fmt_size* are for
* the format string itself. The *data* and *data_len* are format string
* arguments. The *data* are a **u64** array and corresponding format string
* values are stored in the array. For strings
* and pointers where pointees
* are accessed, only the pointer values are stored in the *data* array.
* The *data_len* is the size of *data* in bytes - must be a multiple of 8.
*
* Formats **%s**, **%p{i,I}{4,6}** requires to read kernel memory.
* Reading kernel memory may fail due to either invalid address or
* valid address but requiring a major memory fault. If reading kernel memory
* fails, the string for **%s** will be an empty string, and the ip
* address for **%p{i,I}{4,6}** will be 0. Not returning error to
* bpf program is consistent with what **bpf_trace_printk**\ () does for now.
*
* Returns
* 0 on success, or a negative error in case of failure:
*
* **EBUSY** if per-CPU memory copy buffer is busy, can try again
* by returning 1 from bpf program.
*
* **EINVAL** if arguments are invalid, or if *fmt* is invalid/unsupported.
*
* **E2BIG** if *fmt* contains too many format specifiers.

```

```

*
* **_EOVERFLOW** if an overflow happened: The same
object will be tried again.
*/
static long (* const bpf_seq_printf)(struct seq_file *m, const char *fmt, __u32 fmt_size, const void *data, __u32
data_len) = (void *) 126;

/*
* bpf_seq_write
*
* **bpf_seq_write**\ () uses seq_file **seq_write**\ () to write the data.
* The *m* represents the seq_file. The *data* and *len* represent the
* data to write in bytes.
*
* Returns
* 0 on success, or a negative error in case of failure:
*
* **_EOVERFLOW** if an overflow happened: The same object will be tried again.
*/
static long (* const bpf_seq_write)(struct seq_file *m, const void *data, __u32 len) = (void *) 127;

/*
* bpf_sk_cgroup_id
*
* Return the cgroup v2 id of the socket *sk*.
*
* *sk* must be a non-**NULL** pointer to a socket, e.g. one
* returned from **bpf_sk_lookup_xxx**\ (),
* **bpf_sk_fullsock**\ (), etc. The format of returned id is
* same as in **bpf_skb_cgroup_id**\ ().
*
* This helper is available only if the kernel was compiled with
* the
**CONFIG_SOCK_CGROUP_DATA** configuration option.
*
* Returns
* The id is returned or 0 in case the id could not be retrieved.
*/
static __u64 (* const bpf_sk_cgroup_id)(void *sk) = (void *) 128;

/*
* bpf_sk_ancestor_cgroup_id
*
* Return id of cgroup v2 that is ancestor of cgroup associated
* with the *sk* at the *ancestor_level*. The root cgroup is at
* *ancestor_level* zero and each step down the hierarchy
* increments the level. If *ancestor_level* == level of cgroup
* associated with *sk*, then return value will be same as that

```

```

* of bpf_sk_cgroup_id().
```

* The helper is useful to implement policies based on cgroups
 * that are upper in hierarchy than immediate cgroup associated
 * with **sk**.

* The format of returned id and helper limitations are same as in
 * **bpf_sk_cgroup_id**().

* Returns
 * The id is returned or 0 in case the id could not be retrieved.

```

*/
static __u64 (* const bpf_sk_ancestor_cgroup_id)(void *sk, int ancestor_level)
= (void *) 129;

/*
* bpf_ringbuf_output
*
* Copy size bytes from data into a ring buffer ringbuf.
* If BPF_RB_NO_WAKEUP is specified in flags, no notification
* of new data availability is sent.
* If BPF_RB_FORCE_WAKEUP is specified in flags, notification
* of new data availability is sent unconditionally.
* If 0 is specified in flags, an adaptive notification
* of new data availability is sent.
*
* An adaptive notification is a notification sent whenever the user-space
* process has caught up and consumed all available payloads. In case the user-space
* process is still processing a previous payload, then no notification is needed
* as it will process the newly added payload automatically.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_ringbuf_output)(void *ringbuf, void *data, __u64 size, __u64 flags) = (void *) 130;

/*
* bpf_ringbuf_reserve
*
* Reserve size bytes of payload
  in a ring buffer ringbuf.
* flags must be 0.
*
* Returns
* Valid pointer with size bytes of memory available; NULL,
* otherwise.
*/
static void (* const bpf_ringbuf_reserve)(void *ringbuf, __u64 size, __u64 flags) = (void *) 131;

```

```

/*
 * bpf_ringbuf_submit
 *
 * Submit reserved ring buffer sample, pointed to by *data*.
 * If **BPF_RB_NO_WAKEUP** is specified in *flags*, no notification
 * of new data availability is sent.
 * If **BPF_RB_FORCE_WAKEUP** is specified in *flags*, notification
 * of new data availability is sent unconditionally.
 * If **0** is specified in *flags*, an adaptive notification
 * of new data availability is sent.
 *
 * See 'bpf_ringbuf_output()' for the definition of adaptive notification.
 *
 * Returns
 * Nothing. Always succeeds.
 */
static void (* const bpf_ringbuf_submit)(void *data, __u64 flags) = (void *) 132;

/*
 * bpf_ringbuf_discard
 *
 * Discard reserved ring buffer sample, pointed to by *data*.
 * If **BPF_RB_NO_WAKEUP**
is specified in *flags*, no notification
 * of new data availability is sent.
 * If **BPF_RB_FORCE_WAKEUP** is specified in *flags*, notification
 * of new data availability is sent unconditionally.
 * If **0** is specified in *flags*, an adaptive notification
 * of new data availability is sent.
 *
 * See 'bpf_ringbuf_output()' for the definition of adaptive notification.
 *
 * Returns
 * Nothing. Always succeeds.
 */
static void (* const bpf_ringbuf_discard)(void *data, __u64 flags) = (void *) 133;

/*
 * bpf_ringbuf_query
 *
 * Query various characteristics of provided ring buffer. What
 * exactly is queried is determined by *flags*:
 *
 * * **BPF_RB_AVAIL_DATA**: Amount of data not yet consumed.
 * * **BPF_RB_RING_SIZE**: The size of ring buffer.
 * * **BPF_RB_CONS_POS**: Consumer position (can wrap around).
 * * **BPF_RB_PROD_POS**: Producer(s) position (can wrap around).

```



```

*
* Data returned is just a momentary snapshot of actual values
* and could be inaccurate,
so this facility should be used to
* power heuristics and for reporting, not to make 100% correct
* calculation.
*
* Returns
* Requested value, or 0, if *flags* are not recognized.
*/
static __u64 (* const bpf_ringbuf_query)(void *ringbuf, __u64 flags) = (void *) 134;

/*
* bpf_csum_level
*
* Change the skbs checksum level by one layer up or down, or
* reset it entirely to none in order to have the stack perform
* checksum validation. The level is applicable to the following
* protocols: TCP, UDP, GRE, SCTP, FCOE. For example, a decap of
* | ETH | IP | UDP | GUE | IP | TCP | into | ETH | IP | TCP |
* through **bpf_skb_adjust_room**() helper with passing in
* **BPF_F_ADJ_ROOM_NO_CSUM_RESET** flag would require one call
* to **bpf_csum_level**() with **BPF_CSUM_LEVEL_DEC** since
* the UDP header is removed. Similarly, an encap of the latter
* into the former could be accompanied by a helper call to
* **bpf_csum_level**() with **BPF_CSUM_LEVEL_INC**
if the
* skb is still intended to be processed in higher layers of the
* stack instead of just egressing at tc.
*
* There are three supported level settings at this time:
*
* **BPF_CSUM_LEVEL_INC**: Increases skb->csum_level for skbs
* with CHECKSUM_UNNECESSARY.
* **BPF_CSUM_LEVEL_DEC**: Decreases skb->csum_level for skbs
* with CHECKSUM_UNNECESSARY.
* **BPF_CSUM_LEVEL_RESET**: Resets skb->csum_level to 0 and
* sets CHECKSUM_NONE to force checksum validation by the stack.
* **BPF_CSUM_LEVEL_QUERY**: No-op, returns the current
* skb->csum_level.
*
* Returns
* 0 on success, or a negative error in case of failure. In the
* case of **BPF_CSUM_LEVEL_QUERY**, the current skb->csum_level
* is returned or the error code -EACCES in case the skb is not
* subject to CHECKSUM_UNNECESSARY.
*/
static long (* const bpf_csum_level)(struct __sk_buff *skb, __u64 level) = (void *) 135;

```

```

/*
 * bpf_skc_to_tcp6_sock
 *
 * Dynamically cast a *sk* pointer to
 * a *tcp6_sock* pointer.
 *
 * Returns
 * *sk* if casting is valid, or **NULL** otherwise.
 */
static struct tcp6_sock *(* const bpf_skc_to_tcp6_sock)(void *sk) = (void *) 136;

/*
 * bpf_skc_to_tcp_sock
 *
 * Dynamically cast a *sk* pointer to a *tcp_sock* pointer.
 *
 * Returns
 * *sk* if casting is valid, or **NULL** otherwise.
 */
static struct tcp_sock *(* const bpf_skc_to_tcp_sock)(void *sk) = (void *) 137;

/*
 * bpf_skc_to_tcp_timewait_sock
 *
 * Dynamically cast a *sk* pointer to a *tcp_timewait_sock* pointer.
 *
 * Returns
 * *sk* if casting is valid, or **NULL** otherwise.
 */
static struct tcp_timewait_sock *(* const bpf_skc_to_tcp_timewait_sock)(void *sk) = (void *) 138;

/*
 * bpf_skc_to_tcp_request_sock
 *
 * Dynamically cast a *sk* pointer to a *tcp_request_sock* pointer.
 *
 * Returns
 * *sk* if casting is valid, or **NULL** otherwise.
 */
static struct tcp_request_sock *(* const bpf_skc_to_tcp_request_sock)(void *sk) = (void *) 139;

/*
 * bpf_skc_to_udp6_sock
 *
 * Dynamically cast a *sk* pointer to a *udp6_sock* pointer.
 *
 * Returns

```

```

* *sk* if casting is valid, or **NULL** otherwise.
*/
static struct udp6_sock *(* const bpf_skc_to_udp6_sock)(void *sk) = (void *) 140;

/*
* bpf_get_task_stack
*
* Return a user or a kernel stack in bpf program provided buffer.
* Note: the user stack will only be populated if the *task* is
* the current task; all other tasks will return -EOPNOTSUPP.
* To achieve this, the helper needs *task*, which is a valid
* pointer to **struct task_struct**. To store the stacktrace, the
* bpf program provides *buf* with a nonnegative *size*.
*
* The last argument, *flags*, holds the number of stack frames to
* skip (from 0 to 255), masked with
* **BPF_F_SKIP_FIELD_MASK**. The next bits can be used to set
* the following flags:
*
* **BPF_F_USER_STACK**
* Collect a user space stack instead of a kernel stack.
* The *task* must be the current task.
* **BPF_F_USER_BUILD_ID**
* Collect
* buildid+offset instead of ips for user stack,
* only valid if **BPF_F_USER_STACK** is also specified.
*
* **bpf_get_task_stack**\ () can collect up to
* **PERF_MAX_STACK_DEPTH** both kernel and user frames, subject
* to sufficient large buffer size. Note that
* this limit can be controlled with the **sysctl** program, and
* that it should be manually increased in order to profile long
* user stacks (such as stacks for Java programs). To do so, use:
*
* ::
*
* # sysctl kernel.perf_event_max_stack=<new value>
*
* Returns
* The non-negative copied *buf* length equal to or less than
* *size* on success, or a negative error in case of failure.
*/
static long (* const bpf_get_task_stack)(struct task_struct *task, void *buf, __u32 size, __u64 flags) = (void *) 141;

/*
* bpf_load_hdr_opt
*
* Load header option. Support reading a particular TCP header

```

- * option for bpf program (**BPF_PROG_TYPE SOCK_OPS**).
- *
- * If *flags* is 0, it will search the option from the
- * *skops* \ **->skb_data**. The comment in **struct bpf_sock_ops**
- * has details on what skb_data contains under different
- * *skops* \ **->op**.
- *
- * The first byte of the *searchby_res* specifies the
- * kind that it wants to search.
- *
- * If the searching kind is an experimental kind
- * (i.e. 253 or 254 according to RFC6994). It also
- * needs to specify the "magic" which is either
- * 2 bytes or 4 bytes. It then also needs to
- * specify the size of the magic by using
- * the 2nd byte which is "kind-length" of a TCP
- * header option and the "kind-length" also
- * includes the first 2 bytes "kind" and "kind-length"
- * itself as a normal TCP header option also does.
- *
- * For example, to search experimental kind 254 with
- * 2 byte magic 0xeb9f, the searchby_res should be
- * [254, 4, 0xeb, 0x9f, 0, 0, ... 0].
- *
- * To search for the standard window scale option (3),
- * the *searchby_res* should be [3, 0, 0, ... 0].
- * Note, kind-length must be 0 for regular
- option.
- *
- * Searching for No-Op (0) and End-of-Option-List (1) are
- * not supported.
- *
- * *len* must be at least 2 bytes which is the minimal size
- * of a header option.
- *
- * Supported flags:
- *
- * **BPF_LOAD_HDR_OPT_TCP_SYN** to search from the
- * saved_syn packet or the just-received syn packet.
- *
- *
- * Returns
- * > 0 when found, the header option is copied to *searchby_res*.
- * The return value is the total length copied. On failure, a
- * negative error code is returned:
- *
- * **-EINVAL** if a parameter is invalid.

```

*
* **-ENOMSG** if the option is not found.
*
* **-ENOENT** if no syn packet is available when
* **BPF_LOAD_HDR_OPT_TCP_SYN** is used.
*
* **-ENOSPC** if there is not enough space. Only *len* number of
* bytes are copied.
*
* **-EFAULT** on failure to parse the header options in the
* packet.
*
* **-EPERM** if the helper cannot be used under the current
* *skops*\ **->op**.
*/
static long (* const bpf_load_hdr_opt)(struct bpf_sock_ops
*skops, void *searchby_res, __u32 len, __u64 flags) = (void *) 142;

/*
* bpf_store_hdr_opt
*
* Store header option. The data will be copied
* from buffer *from* with length *len* to the TCP header.
*
* The buffer *from* should have the whole option that
* includes the kind, kind-length, and the actual
* option data. The *len* must be at least kind-length
* long. The kind-length does not have to be 4 byte
* aligned. The kernel will take care of the padding
* and setting the 4 bytes aligned value to th->doff.
*
* This helper will check for duplicated option
* by searching the same option in the outgoing skb.
*
* This helper can only be called during
* **BPF_SOCK_OPS_WRITE_HDR_OPT_CB**.
*
*
* Returns
* 0 on success, or negative error in case of failure:
*
* **-EINVAL** If param is invalid.
*
* **-ENOSPC** if there is not enough space in the header.
* Nothing has been written
*
* **-EEXIST** if the option already exists.
*

```

```

* **EFAULT**
on failure to parse the existing header options.
*
* **EPERM** if the helper cannot be used under the current
* *skops* \ **->op**.
*/
static long (* const bpf_store_hdr_opt)(struct bpf_sock_ops *skops, const void *from, __u32 len, __u64 flags) =
(void *) 143;

/*
* bpf_reserve_hdr_opt
*
* Reserve *len* bytes for the bpf header option. The
* space will be used by **bpf_store_hdr_opt**() later in
* **BPF_SOCKET_OPS_WRITE_HDR_OPT_CB**.
*
* If **bpf_reserve_hdr_opt**() is called multiple times,
* the total number of bytes will be reserved.
*
* This helper can only be called during
* **BPF_SOCKET_OPS_HDR_OPT_LEN_CB**.
*
*
* Returns
* 0 on success, or negative error in case of failure:
*
* **EINVAL** if a parameter is invalid.
*
* **ENOSPC** if there is not enough space in the header.
*
* **EPERM** if the helper cannot be used under the current
* *skops* \ **->op**.
*/
static long (* const bpf_reserve_hdr_opt)(struct bpf_sock_ops *skops,
__u32 len, __u64 flags) = (void *) 144;

/*
* bpf_inode_storage_get
*
* Get a bpf_local_storage from an *inode*.
*
* Logically, it could be thought of as getting the value from
* a *map* with *inode* as the **key**. From this
* perspective, the usage is not much different from
* **bpf_map_lookup_elem**(*map*, **&inode*) except this
* helper enforces the key must be an inode and the map must also
* be a **BPF_MAP_TYPE_INODE_STORAGE**.
*

```

```

* Underneath, the value is stored locally at *inode* instead of
* the *map*. The *map* is used as the bpf-local-storage
* "type". The bpf-local-storage "type" (i.e. the *map*) is
* searched against all bpf_local_storage residing at *inode*.
*
* An optional *flags* (**BPF_LOCAL_STORAGE_GET_F_CREATE**) can be
* used such that a new bpf_local_storage will be
* created if one does not exist. *value* can be used
* together with **BPF_LOCAL_STORAGE_GET_F_CREATE** to specify
* the initial value of a bpf_local_storage.
  If *value* is
* **NULL**, the new bpf_local_storage will be zero initialized.
*
* Returns
* A bpf_local_storage pointer is returned on success.
*
* **NULL** if not found or there was an error in adding
* a new bpf_local_storage.
*/
static void (* const bpf_inode_storage_get)(void *map, void *inode, void *value, __u64 flags) = (void *) 145;

/*
* bpf_inode_storage_delete
*
* Delete a bpf_local_storage from an *inode*.
*
* Returns
* 0 on success.
*
* **_ENOENT** if the bpf_local_storage cannot be found.
*/
static int (* const bpf_inode_storage_delete)(void *map, void *inode) = (void *) 146;

/*
* bpf_d_path
*
* Return full path for given **struct path** object, which
* needs to be the kernel BTF *path* object. The path is
* returned in the provided buffer *buf* of size *sz* and
* is zero terminated.
*
* Returns
* On success, the strictly positive length of the string,
* including the trailing NUL character. On error, a negative
* value.
*/
static long (* const bpf_d_path)(struct path *path, char *buf, __u32 sz) = (void *) 147;

```

```

/*
 * bpf_copy_from_user
 *
 * Read *size* bytes from user space address *user_ptr* and store
 * the data in *dst*. This is a wrapper of **copy_from_user**\ ().
 *
 * Returns
 * 0 on success, or a negative error in case of failure.
 */
static long (* const bpf_copy_from_user)(void *dst, __u32 size, const void *user_ptr) = (void *) 148;

/*
 * bpf_snprintf_btf
 *
 * Use BTF to store a string representation of *ptr*->ptr in *str*,
 * using *ptr*->type_id. This value should specify the type
 * that *ptr*->ptr points to. LLVM __builtin_btf_type_id(type, 1)
 * can be used to look up vmlinux BTF type ids. Traversing the
 * data structure using BTF, the type information and values are
 * stored in the first *str_size* - 1 bytes of *str*. Safe copy of
 * the pointer data is carried out to avoid kernel crashes during
 * operation. Smaller types can use string space on the stack;
 * larger
 * programs can use map data to store the string
 * representation.
 *
 * The string can be subsequently shared with userspace via
 * bpf_perf_event_output() or ring buffer interfaces.
 * bpf_trace_printk() is to be avoided as it places too small
 * a limit on string size to be useful.
 *
 * *flags* is a combination of
 *
 * **BTF_F_COMPACT**
 * no formatting around type information
 * **BTF_F_NONAME**
 * no struct/union member names/types
 * **BTF_F_PTR_RAW**
 * show raw (unobfuscated) pointer values;
 * equivalent to printk specifier %px.
 * **BTF_F_ZERO**
 * show zero-valued struct/union members; they
 * are not displayed by default
 *
 * Returns
 * The number of bytes that were written (or would have been

```



```

* written if output had to be truncated due to string size),
* or a negative error in cases of failure.
*/
static long (* const bpf_snprintf_btf)(char *str, __u32 str_size, struct btf_ptr *ptr, __u32 btf_ptr_size, __u64 flags) =
(void *) 149;

/*
* bpf_seq_printf_btf
*
* Use BTF to write to seq_write a string representation of
* *ptr*->ptr, using *ptr*->type_id as per bpf_snprintf_btf().
* *flags* are identical to those used for bpf_snprintf_btf.
*
* Returns
* 0 on success or a negative error in case of failure.
*/
static long (* const bpf_seq_printf_btf)(struct seq_file *m, struct btf_ptr *ptr, __u32 ptr_size, __u64 flags) = (void
*) 150;

/*
* bpf_skb_cgroup_classid
*
* See bpf_get_cgroup_classid() for the main description.
* This helper differs from bpf_get_cgroup_classid() in that
* the cgroup v1 net_cls class is retrieved only from the *skb*'s
* associated socket instead of the current process.
*
* Returns
* The id is returned or 0 in case the id could not be retrieved.
*/
static __u64 (* const bpf_skb_cgroup_classid)(struct __sk_buff *skb) = (void *) 151;

/*
* bpf_redirect_neigh
*
* Redirect the packet to another net device of index *ifindex*
* and fill in L2 addresses from neighboring
* subsystem. This helper
* is somewhat similar to bpf_redirect(), except that it
* populates L2 addresses as well, meaning, internally, the helper
* relies on the neighbor lookup for the L2 address of the nexthop.
*
* The helper will perform a FIB lookup based on the skb's
* networking header to get the address of the next hop, unless
* this is supplied by the caller in the *params* argument. The
* *plen* argument indicates the len of *params* and should be set
* to 0 if *params* is NULL.
*

```

```

* The *flags* argument is reserved and must be 0. The helper is
* currently only supported for tc BPF program types, and enabled
* for IPv4 and IPv6 protocols.
*
* Returns
* The helper returns TC_ACT_REDIRECT on success or
* TC_ACT_SHOT on error.
*/
static long (* const bpf_redirect_neigh)(__u32 ifindex, struct bpf_redir_neigh *params, int plen, __u64 flags) =
(void *) 152;

/*
* bpf_per_cpu_ptr
*
* Take a pointer to a percpu ksym, *percpu_ptr*,
and return a
* pointer to the percpu kernel variable on *cpu*. A ksym is an
* extern variable decorated with '__ksym'. For ksym, there is a
* global var (either static or global) defined of the same name
* in the kernel. The ksym is percpu if the global var is percpu.
* The returned pointer points to the global percpu var on *cpu*.
*
* bpf_per_cpu_ptr() has the same semantic as per_cpu_ptr() in the
* kernel, except that bpf_per_cpu_ptr() may return NULL. This
* happens if *cpu* is larger than nr_cpu_ids. The caller of
* bpf_per_cpu_ptr() must check the returned value.
*
* Returns
* A pointer pointing to the kernel percpu variable on *cpu*, or
* NULL, if *cpu* is invalid.
*/
static void *(* const bpf_per_cpu_ptr)(const void *percpu_ptr, __u32 cpu) = (void *) 153;

/*
* bpf_this_cpu_ptr
*
* Take a pointer to a percpu ksym, *percpu_ptr*, and return a
* pointer to the percpu kernel variable on this cpu. See the
* description of 'ksym' in bpf_per_cpu_ptr\
().
*
* bpf_this_cpu_ptr() has the same semantic as this_cpu_ptr() in
* the kernel. Different from bpf_per_cpu_ptr\ (), it would
* never return NULL.
*
* Returns
* A pointer pointing to the kernel percpu variable on this cpu.
*/

```

```

static void (* const bpf_this_cpu_ptr)(const void *percpu_ptr) = (void *) 154;

/*
 * bpf_redirect_peer
 *
 * Redirect the packet to another net device of index *ifindex*.
 * This helper is somewhat similar to **bpf_redirect**(), except
 * that the redirection happens to the *ifindex* peer device and
 * the netns switch takes place from ingress to ingress without
 * going through the CPU's backlog queue.
 *
 * The *flags* argument is reserved and must be 0. The helper is
 * currently only supported for tc BPF program types at the
 * ingress hook and for veth and netkit target device types. The
 * peer device must reside in a different network namespace.
 *
 * Returns
 * The helper returns **TC_ACT_REDIRECT** on success or
 *
 * **TC_ACT_SHOT** on error.
 */
static long (* const bpf_redirect_peer)(__u32 ifindex, __u64 flags) = (void *) 155;

/*
 * bpf_task_storage_get
 *
 * Get a bpf_local_storage from the *task*.
 *
 * Logically, it could be thought of as getting the value from
 * a *map* with *task* as the **key**. From this
 * perspective, the usage is not much different from
 * **bpf_map_lookup_elem**(*map*, **&task*) except this
 * helper enforces the key must be a task_struct and the map must also
 * be a **BPF_MAP_TYPE_TASK_STORAGE**.
 *
 * Underneath, the value is stored locally at *task* instead of
 * the *map*. The *map* is used as the bpf-local-storage
 * "type". The bpf-local-storage "type" (i.e. the *map*) is
 * searched against all bpf_local_storage residing at *task*.
 *
 * An optional *flags* (**BPF_LOCAL_STORAGE_GET_F_CREATE**) can be
 * used such that a new bpf_local_storage will be
 * created if one does not exist. *value* can be used
 * together with **BPF_LOCAL_STORAGE_GET_F_CREATE**
 * to specify
 * the initial value of a bpf_local_storage. If *value* is
 * **NULL**, the new bpf_local_storage will be zero initialized.
 *

```

```

* Returns
* A bpf_local_storage pointer is returned on success.
*
* **NULL** if not found or there was an error in adding
* a new bpf_local_storage.
*/
static void *(* const bpf_task_storage_get)(void *map, struct task_struct *task, void *value, __u64 flags) = (void *)
156;

/*
* bpf_task_storage_delete
*
* Delete a bpf_local_storage from a *task*.
*
* Returns
* 0 on success.
*
* **ENOENT** if the bpf_local_storage cannot be found.
*/
static long (* const bpf_task_storage_delete)(void *map, struct task_struct *task) = (void *) 157;

/*
* bpf_get_current_task_btf
*
* Return a BTF pointer to the "current" task.
* This pointer can also be used in helpers that accept an
* *ARG_PTR_TO_BTF_ID* of type *task_struct*.
*
* Returns
* Pointer to the current task.
*/
static struct task_struct *(* const
bpf_get_current_task_btf)(void) = (void *) 158;

/*
* bpf_bprm_opts_set
*
* Set or clear certain options on *bprm*:
*
* **BPF_F_BPRM_SECUREEXEC** Set the secureexec bit
* which sets the **AT_SECURE** auxv for glibc. The bit
* is cleared if the flag is not specified.
*
* Returns
* **EINVAL** if invalid *flags* are passed, zero otherwise.
*/
static long (* const bpf_bprm_opts_set)(struct linux_binprm *bprm, __u64 flags) = (void *) 159;

```

```

/*
 * bpf_ktime_get_coarse_ns
 *
 * Return a coarse-grained version of the time elapsed since
 * system boot, in nanoseconds. Does not include time the system
 * was suspended.
 *
 * See: **clock_gettime**\ (**CLOCK_MONOTONIC_COARSE**)
 *
 * Returns
 * Current *ktime*.
 */
static __u64 (* const bpf_ktime_get_coarse_ns)(void) = (void *) 160;

/*
 * bpf_ima_inode_hash
 *
 * Returns the stored IMA hash of the *inode* (if it's available).
 * If the hash is larger than *size*, then only *size*
 * bytes will be copied to *dst*
 *
 * Returns
 * The **hash_algo** is returned on success,
 * **EOPNOTSUPP** if IMA is disabled or **-EINVAL** if
 * invalid arguments are passed.
 */
static long (* const bpf_ima_inode_hash)(struct inode *inode, void *dst, __u32 size) = (void *) 161;

/*
 * bpf_sock_from_file
 *
 * If the given file represents a socket, returns the associated
 * socket.
 *
 * Returns
 * A pointer to a struct socket on success or NULL if the file is
 * not a socket.
 */
static struct socket *(* const bpf_sock_from_file)(struct file *file) = (void *) 162;

/*
 * bpf_check_mtu
 *
 * Check packet size against exceeding MTU of net device (based
 * on *ifindex*). This helper will likely be used in combination
 * with helpers that adjust/change the packet size.
 *
 * The argument *len_diff* can be used for querying with a planned

```

- * size change. This allows to check MTU prior to changing packet
- * ctx. Providing a `*len_diff*` adjustment that is larger than the
- * actual packet size (resulting in negative packet size) will in
- * principle not exceed the MTU, which is why it is not considered
- * a failure. Other BPF helpers are needed for performing the
- * planned size change; therefore the responsibility for catching
- * a negative packet size belongs in those helpers.
- *
- * Specifying `*ifindex*` zero means the MTU check is performed
- * against the current net device. This is practical if this isn't
- * used prior to redirect.
- *
- * On input `*mtu_len*` must be a valid pointer, else verifier will
- * reject BPF program. If the value `*mtu_len*` is initialized to
- * zero then the ctx packet size is use. When value `*mtu_len*` is
- * provided as input this specify the L3 length that the MTU check
- * is done against. Remember XDP and TC length operate at L2, but
- * this value is L3 as this correlate to MTU and IP-header `tot_len`
- * values which are L3 (similar behavior as `bpf_fib_lookup`).
- *
- * The Linux kernel route table can configure MTUs on a more
- * specific per route level,
- which is not provided by this helper.
- * For route level MTU checks use the `**bpf_fib_lookup**\()`
- * helper.
- *
- * `*ctx*` is either `**struct xdp_md**` for XDP programs or
- * `**struct sk_buff**` for tc cls_act programs.
- *
- * The `*flags*` argument can be a combination of one or more of the
- * following values:
- *
- * `**BPF_MTU_CHK_SEGS**`
- * This flag will only works for `*ctx*` `**struct sk_buff**`.
- * If packet context contains extra packet segment buffers
- * (often knows as GSO skb), then MTU check is harder to
- * check at this point, because in transmit path it is
- * possible for the skb packet to get re-segmented
- * (depending on net device features). This could still be
- * a MTU violation, so this flag enables performing MTU
- * check against segments, with a different violation
- * return code to tell it apart. Check cannot use `len_diff`.
- *
- * On return `*mtu_len*` pointer contains the MTU value of the net
- * device. Remember the net device configured MTU is the
- L3 size,
- * which is returned here and XDP and TC length operate at L2.

```

* Helper take this into account for you, but remember when using
* MTU value in your BPF-code.
*
*
* Returns
* * 0 on success, and populate MTU value in *mtu_len* pointer.
*
* * < 0 if any input argument is invalid (*mtu_len* not updated)
*
* MTU violations return positive values, but also populate MTU
* value in *mtu_len* pointer, as this can be needed for
* implementing PMTU handing:
*
* ***BPF_MTU_CHK_RET_FRAG_NEEDED**
* ***BPF_MTU_CHK_RET_SEGS_TOOBIG**
*/
static long (* const bpf_check_mtu)(void *ctx, __u32 ifindex, __u32 *mtu_len, __s32 len_diff, __u64 flags) = (void
*) 163;

/*
* bpf_for_each_map_elem
*
* For each element in **map**, call **callback_fn** function with
* **map**, **callback_ctx** and other map-specific parameters.
* The **callback_fn** should be a static function and
* the **callback_ctx** should be a pointer to the stack.
* The **flags** is used to control
certain aspects of the helper.
* Currently, the **flags** must be 0.
*
* The following are a list of supported map types and their
* respective expected callback signatures:
*
* BPF_MAP_TYPE_HASH, BPF_MAP_TYPE_PERCPU_HASH,
* BPF_MAP_TYPE_LRU_HASH, BPF_MAP_TYPE_LRU_PERCPU_HASH,
* BPF_MAP_TYPE_ARRAY, BPF_MAP_TYPE_PERCPU_ARRAY
*
* long (*callback_fn)(struct bpf_map *map, const void *key, void *value, void *ctx);
*
* For per_cpu maps, the map_value is the value on the cpu where the
* bpf_prog is running.
*
* If **callback_fn** return 0, the helper will continue to the next
* element. If return value is 1, the helper will skip the rest of
* elements and return. Other return values are not used now.
*
*
* Returns

```

```

* The number of traversed map elements for success, -EINVAL for
* invalid flags.
*/
static long (* const bpf_for_each_map_elem)(void *map, void *callback_fn, void *callback_ctx, __u64 flags) =
(void *) 164;

/*
* bpf_snprintf
*
* Outputs a string into the str buffer of size str_size
* based on a format string stored in a read-only map pointed by
* fmt.
*
* Each format specifier in fmt corresponds to one u64 element
* in the data array. For strings and pointers where pointees
* are accessed, only the pointer values are stored in the data
* array. The data_len is the size of data in bytes - must be
* a multiple of 8.
*
* Formats %s and %p{i,I}{4,6} require to read kernel
* memory. Reading kernel memory may fail due to either invalid
* address or valid address but requiring a major memory fault. If
* reading kernel memory fails, the string for %s will be an
* empty string, and the ip address for %p{i,I}{4,6} will be 0.
* Not returning error to bpf program is consistent with what
* bpf_trace_printk() does for now.
*
* Returns
* The strictly positive length of the formatted string, including
* the trailing zero character. If
the return value is greater than
* str_size, str contains a truncated string, guaranteed to
* be zero-terminated except when str_size is 0.
*
* Or -EBUSY if the per-CPU memory copy buffer is busy.
*/
static long (* const bpf_snprintf)(char *str, __u32 str_size, const char *fmt, __u64 *data, __u32 data_len) = (void *)
165;

/*
* bpf_sys_bpf
*
* Execute bpf syscall with given arguments.
*
* Returns
* A syscall result.
*/

```



```

static long (* const bpf_sys_bpf)(__u32 cmd, void *attr, __u32 attr_size) = (void *) 166;

/*
 * bpf_btf_find_by_name_kind
 *
 * Find BTF type with given name and kind in vmlinux BTF or in module's BTFs.
 *
 * Returns
 * Returns btf_id and btf_obj_fd in lower and upper 32 bits.
 */
static long (* const bpf_btf_find_by_name_kind)(char *name, int name_sz, __u32 kind, int flags) = (void *) 167;

/*
 * bpf_sys_close
 *
 * Execute close syscall for given FD.
 *
 * Returns
 * A syscall result.
 */
static long (* const bpf_sys_close)(__u32
fd) = (void *) 168;

/*
 * bpf_timer_init
 *
 * Initialize the timer.
 * First 4 bits of *flags* specify clockid.
 * Only CLOCK_MONOTONIC, CLOCK_REALTIME, CLOCK_BOOTTIME are allowed.
 * All other bits of *flags* are reserved.
 * The verifier will reject the program if *timer* is not from
 * the same *map*.
 *
 * Returns
 * 0 on success.
 * **EBUSY** if *timer* is already initialized.
 * **EINVAL** if invalid *flags* are passed.
 * **EPERM** if *timer* is in a map that doesn't have any user references.
 * The user space should either hold a file descriptor to a map with timers
 * or pin such map in bpffs. When map is unpinned or file descriptor is
 * closed all timers in the map will be cancelled and freed.
 */
static long (* const bpf_timer_init)(struct bpf_timer *timer, void *map, __u64 flags) = (void *) 169;

/*
 * bpf_timer_set_callback
 *
 * Configure the timer to call *callback_fn* static function.

```

```

*
* Returns
* 0 on success.
* **EINVAL** if *timer* was not
    initialized with bpf_timer_init() earlier.
* **EPERM** if *timer* is in a map that doesn't have any user references.
* The user space should either hold a file descriptor to a map with timers
* or pin such map in bpffs. When map is unpinned or file descriptor is
* closed all timers in the map will be cancelled and freed.
*/
static long (* const bpf_timer_set_callback)(struct bpf_timer *timer, void *callback_fn) = (void *) 170;

/*
* bpf_timer_start
*
* Set timer expiration N nanoseconds from the current time. The
* configured callback will be invoked in soft irq context on some cpu
* and will not repeat unless another bpf_timer_start() is made.
* In such case the next invocation can migrate to a different cpu.
* Since struct bpf_timer is a field inside map element the map
* owns the timer. The bpf_timer_set_callback() will increment refcnt
* of BPF program to make sure that callback_fn code stays valid.
* When user space reference to a map reaches zero all
    timers
* in a map are cancelled and corresponding program's refcnts are
* decremented. This is done to make sure that Ctrl-C of a user
* process doesn't leave any timers running. If map is pinned in
* bpffs the callback_fn can re-arm itself indefinitely.
* bpf_map_update/delete_elem() helpers and user space sys_bpf commands
* cancel and free the timer in the given map element.
* The map can contain timers that invoke callback_fn-s from different
* programs. The same callback_fn can serve different timers from
* different maps if key/value layout matches across maps.
* Every bpf_timer_set_callback() can have different callback_fn.
*
* *flags* can be one of:
*
* **BPF_F_TIMER_ABS**
* Start the timer in absolute expire value instead of the
* default relative one.
* **BPF_F_TIMER_CPU_PIN**
* Timer will be pinned to the CPU of the caller.
*
* Returns
* 0 on success.
* **EINVAL** if *timer* was not initialized with bpf_timer_init() earlier
* or

```

```

invalid *flags* are passed.
*/
static long (* const bpf_timer_start)(struct bpf_timer *timer, __u64 nsecs, __u64 flags) = (void *) 171;

/*
 * bpf_timer_cancel
 *
 * Cancel the timer and wait for callback_fn to finish if it was running.
 *
 * Returns
 * 0 if the timer was not active.
 * 1 if the timer was active.
 * ** -EINVAL ** if *timer* was not initialized with bpf_timer_init() earlier.
 * ** -EDEADLK ** if callback_fn tried to call bpf_timer_cancel() on its
 * own timer which would have led to a deadlock otherwise.
 */
static long (* const bpf_timer_cancel)(struct bpf_timer *timer) = (void *) 172;

/*
 * bpf_get_func_ip
 *
 * Get address of the traced function (for tracing and kprobe programs).
 *
 * When called for kprobe program attached as uprobe it returns
 * probe address for both entry and return uprobe.
 *
 * Returns
 * Address of the traced function for kprobe.
 * 0 for kprobes placed within the function (not at the entry).
 * Address of the probe for uprobe
 * and return uprobe.
 */
static __u64 (* const bpf_get_func_ip)(void *ctx) = (void *) 173;

/*
 * bpf_get_attach_cookie
 *
 * Get bpf_cookie value provided (optionally) during the program
 * attachment. It might be different for each individual
 * attachment, even if BPF program itself is the same.
 * Expects BPF program context *ctx* as a first argument.
 *
 * Supported for the following program types:
 * - kprobe/uprobe;
 * - tracepoint;
 * - perf_event.
 */

```

```

* Returns
* Value specified by user at BPF link creation/attachment time
* or 0, if it was not specified.
*/
static __u64 (* const bpf_get_attach_cookie)(void *ctx) = (void *) 174;

/*
* bpf_task_pt_regs
*
* Get the struct pt_regs associated with **task**.
*
* Returns
* A pointer to struct pt_regs.
*/
static long (* const bpf_task_pt_regs)(struct task_struct *task) = (void *) 175;

/*
* bpf_get_branch_snapshot
*
* Get branch trace from hardware engines like Intel LBR. The
* hardware engine is stopped
* shortly after the helper is
* called. Therefore, the user need to filter branch entries
* based on the actual use case. To capture branch trace
* before the trigger point of the BPF program, the helper
* should be called at the beginning of the BPF program.
*
* The data is stored as struct perf_branch_entry into output
* buffer *entries*. *size* is the size of *entries* in bytes.
* *flags* is reserved for now and must be zero.
*
*
* Returns
* On success, number of bytes written to *buf*. On error, a
* negative value.
*
* **EINVAL** if *flags* is not zero.
*
* **ENOENT** if architecture does not support branch records.
*/
static long (* const bpf_get_branch_snapshot)(void *entries, __u32 size, __u64 flags) = (void *) 176;

/*
* bpf_trace_vprintk
*
* Behaves like **bpf_trace_printk**\ () helper, but takes an array of u64
* to format and can handle more format args as a result.
*

```

```

* Arguments are to be used as in bpf_seq_printf() helper.
*
*
Returns
* The number of bytes written to the buffer, or a negative error
* in case of failure.
*/
static long (* const bpf_trace_vprintk)(const char *fmt, __u32 fmt_size, const void *data, __u32 data_len) = (void *) 177;

/*
* bpf_skc_to_unix_sock
*
* Dynamically cast a *sk pointer to a *unix_sock pointer.
*
Returns
* *sk if casting is valid, or NULL otherwise.
*/
static struct unix_sock *(* const bpf_skc_to_unix_sock)(void *sk) = (void *) 178;

/*
* bpf_kallsyms_lookup_name
*
* Get the address of a kernel symbol, returned in *res. *res is
* set to 0 if the symbol is not found.
*
Returns
* On success, zero. On error, a negative value.
*
* -EINVAL if *flags is not zero.
*
* -EINVAL if string *name is not the same size as *name_sz.
*
* -ENOENT if symbol is not found.
*
* -EPERM if caller does not have permission to obtain kernel address.
*/
static long (* const bpf_kallsyms_lookup_name)(const char *name, int name_sz, int
flags, __u64 *res) = (void *) 179;

/*
* bpf_find_vma
*
* Find vma of *task that contains *addr, call *callback_fn
* function with *task, *vma, and *callback_ctx.
* The *callback_fn should be a static function and
* the *callback_ctx should be a pointer to the stack.
* The *flags is used to control certain aspects of the helper.

```

```

* Currently, the *flags* must be 0.
*
* The expected callback signature is
*
* long (*callback_fn)(struct task_struct *task, struct vm_area_struct *vma, void *callback_ctx);
*
*
* Returns
* 0 on success.
* **ENOENT** if *task->mm* is NULL, or no vma contains *addr*.
* **EBUSY** if failed to try lock mmap_lock.
* **EINVAL** for invalid **flags**.
*/
static long (* const bpf_find_vma)(struct task_struct *task, __u64 addr, void *callback_fn, void *callback_ctx,
__u64 flags) = (void *) 180;

/*
* bpf_loop
*
* For **nr_loops**, call **callback_fn** function
* with **callback_ctx** as the context parameter.
* The
**callback_fn** should be a static function and
* the **callback_ctx** should be a pointer to the stack.
* The **flags** is used to control certain aspects of the helper.
* Currently, the **flags** must be 0. Currently, nr_loops is
* limited to 1 << 23 (~8 million) loops.
*
* long (*callback_fn)(u32 index, void *ctx);
*
* where **index** is the current index in the loop. The index
* is zero-indexed.
*
* If **callback_fn** returns 0, the helper will continue to the next
* loop. If return value is 1, the helper will skip the rest of
* the loops and return. Other return values are not used now,
* and will be rejected by the verifier.
*
*
* Returns
* The number of loops performed, **EINVAL** for invalid **flags**,
* **E2BIG** if **nr_loops** exceeds the maximum number of loops.
*/
static long (* const bpf_loop)(__u32 nr_loops, void *callback_fn, void *callback_ctx, __u64 flags) = (void *) 181;

/*
* bpf_strncmp
*

```

```

* Do strcmp() between **s1** and
**s2**. **s1** doesn't need
* to be null-terminated and **s1_sz** is the maximum storage
* size of **s1**. **s2** must be a read-only string.
*
* Returns
* An integer less than, equal to, or greater than zero
* if the first **s1_sz** bytes of **s1** is found to be
* less than, to match, or be greater than **s2**.
*/
static long (* const bpf_strcmp)(const char *s1, __u32 s1_sz, const char *s2) = (void *) 182;

/*
* bpf_get_func_arg
*
* Get **n**-th argument register (zero based) of the traced function (for tracing programs)
* returned in **value**.
*
*
* Returns
* 0 on success.
* **EINVAL** if n >= argument register count of traced function.
*/
static long (* const bpf_get_func_arg)(void *ctx, __u32 n, __u64 *value) = (void *) 183;

/*
* bpf_get_func_ret
*
* Get return value of the traced function (for tracing programs)
* in **value**.
*
*
* Returns
* 0 on success.
* **EOPNOTSUPP** for tracing programs other than BPF_TRACE_FEXIT or BPF_MODIFY_RETURN.
*/
static long (* const bpf_get_func_ret)(void *ctx, __u64 *value) = (void *) 184;

/*
* bpf_get_func_arg_cnt
*
* Get number of registers of the traced function (for tracing programs) where
* function arguments are stored in these registers.
*
*
* Returns
* The number of argument registers of the traced function.
*/

```

```

static long (* const bpf_get_func_arg_cnt)(void *ctx) = (void *) 185;

/*
 * bpf_get_retval
 *
 * Get the BPF program's return value that will be returned to the upper layers.
 *
 * This helper is currently supported by cgroup programs and only by the hooks
 * where BPF program's return value is returned to the userspace via errno.
 *
 * Returns
 * The BPF program's return value.
 */
static int (* const bpf_get_retval)(void) = (void *) 186;

/*
 * bpf_set_retval
 *
 * Set the BPF program's return value that will be returned to the upper layers.
 *
 * This helper is currently supported by cgroup programs and only by the hooks
 * where BPF program's
 * return value is returned to the userspace via errno.
 *
 * Note that there is the following corner case where the program exports an error
 * via bpf_set_retval but signals success via 'return 1':
 *
 * bpf_set_retval(-EPERM);
 * return 1;
 *
 * In this case, the BPF program's return value will use helper's -EPERM. This
 * still holds true for cgroup/bind{4,6} which supports extra 'return 3' success case.
 *
 * Returns
 * 0 on success, or a negative error in case of failure.
 */
static int (* const bpf_set_retval)(int retval) = (void *) 187;

/*
 * bpf_xdp_get_buff_len
 *
 * Get the total size of a given xdp buff (linear and paged area)
 *
 * Returns
 * The total size of a given xdp buffer.
 */
static __u64 (* const bpf_xdp_get_buff_len)(struct xdp_md *xdp_md) = (void *) 188;

```



```

/*
 * bpf_xdp_load_bytes
 *
 * This helper is provided as an easy way to load data from a
 * xdp buffer. It can be used to load *len* bytes from *offset* from
 * the frame associated to *xdp_md*,
 * into the buffer pointed by
 * *buf*.
 *
 * Returns
 * 0 on success, or a negative error in case of failure.
 */
static long (* const bpf_xdp_load_bytes)(struct xdp_md *xdp_md, __u32 offset, void *buf, __u32 len) = (void *)
189;

/*
 * bpf_xdp_store_bytes
 *
 * Store *len* bytes from buffer *buf* into the frame
 * associated to *xdp_md*, at *offset*.
 *
 * Returns
 * 0 on success, or a negative error in case of failure.
 */
static long (* const bpf_xdp_store_bytes)(struct xdp_md *xdp_md, __u32 offset, void *buf, __u32 len) = (void *)
190;

/*
 * bpf_copy_from_user_task
 *
 * Read *size* bytes from user space address *user_ptr* in *tsk*'s
 * address space, and stores the data in *dst*. *flags* is not
 * used yet and is provided for future extensibility. This helper
 * can only be used by sleepable programs.
 *
 * Returns
 * 0 on success, or a negative error in case of failure. On error
 * *dst* buffer is zeroed out.
 */
static long (* const bpf_copy_from_user_task)(void *dst,
__u32 size, const void *user_ptr, struct task_struct *tsk, __u64 flags) = (void *) 191;

/*
 * bpf_skb_set_timestamp
 *
 * Change the __sk_buff->timestamp_type to *timestamp_type*
 * and set *timestamp* to the __sk_buff->timestamp together.

```

```

*
* If there is no need to change the __sk_buff->tstamp_type,
* the tstamp value can be directly written to __sk_buff->tstamp
* instead.
*
* BPF_SKB_TSTAMP_DELIVERY_MONO is the only tstamp that
* will be kept during bpf_redirect_*. A non zero
* *tstamp* must be used with the BPF_SKB_TSTAMP_DELIVERY_MONO
* *tstamp_type*.
*
* A BPF_SKB_TSTAMP_UNSPEC *tstamp_type* can only be used
* with a zero *tstamp*.
*
* Only IPv4 and IPv6 skb->protocol are supported.
*
* This function is most useful when it needs to set a
* mono delivery time to __sk_buff->tstamp and then
* bpf_redirect_() to the egress of an iface. For example,
* changing the (rcv) timestamp in __sk_buff->tstamp at
* ingress to a mono delivery time and then bpf_redirect_()
* to sch_fq@phy-dev.
*
* Returns
* 0 on success.
* **EINVAL** for invalid input
* **EOPNOTSUPP** for unsupported protocol
*/
static long (* const bpf_skb_set_tstamp)(struct __sk_buff *skb, __u64 tstamp, __u32 tstamp_type) = (void *) 192;

/*
* bpf_ima_file_hash
*
* Returns a calculated IMA hash of the *file*.
* If the hash is larger than *size*, then only *size*
* bytes will be copied to *dst*
*
* Returns
* The **hash_algo** is returned on success,
* **EOPNOTSUPP** if the hash calculation failed or **EINVAL** if
* invalid arguments are passed.
*/
static long (* const bpf_ima_file_hash)(struct file *file, void *dst, __u32 size) = (void *) 193;

/*
* bpf_kptr_xchg
*
* Exchange kptr at pointer *map_value* with *ptr*, and return the
* old value. *ptr* can be NULL, otherwise it must be a referenced

```

```

* pointer which will be released when this helper is called.
*
* Returns
* The old value of kptr (which can be NULL). The returned pointer
* if
not NULL, is a reference which must be released using its
* corresponding release function, or moved into a BPF map before
* program exit.
*/
static void (* const bpf_kptr_xchg)(void *map_value, void *ptr) = (void *) 194;

/*
* bpf_map_lookup_percpu_elem
*
* Perform a lookup in *percpu map* for an entry associated to
* *key* on *cpu*.
*
* Returns
* Map value associated to *key* on *cpu*, or **NULL** if no entry
* was found or *cpu* is invalid.
*/
static void (* const bpf_map_lookup_percpu_elem)(void *map, const void *key, __u32 cpu) = (void *) 195;

/*
* bpf_skc_to_mptcp_sock
*
* Dynamically cast a *sk* pointer to a *mptcp_sock* pointer.
*
* Returns
* *sk* if casting is valid, or **NULL** otherwise.
*/
static struct mptcp_sock (* const bpf_skc_to_mptcp_sock)(void *sk) = (void *) 196;

/*
* bpf_dynptr_from_mem
*
* Get a dynptr to local memory *data*.
*
* *data* must be a ptr to a map value.
* The maximum *size* supported is DYNPTR_MAX_SIZE.
* *flags*
is currently unused.
*
* Returns
* 0 on success, -E2BIG if the size exceeds DYNPTR_MAX_SIZE,
* -EINVAL if flags is not 0.
*/
static long (* const bpf_dynptr_from_mem)(void *data, __u32 size, __u64 flags, struct bpf_dynptr *ptr) = (void *)

```

197;

```
/*
 * bpf_ringbuf_reserve_dynptr
 *
 * Reserve *size* bytes of payload in a ring buffer *ringbuf*
 * through the dynptr interface. *flags* must be 0.
 *
 * Please note that a corresponding bpf_ringbuf_submit_dynptr or
 * bpf_ringbuf_discard_dynptr must be called on *ptr*, even if the
 * reservation fails. This is enforced by the verifier.
 *
 * Returns
 * 0 on success, or a negative error in case of failure.
 */
static long (* const bpf_ringbuf_reserve_dynptr)(void *ringbuf, __u32 size, __u64 flags, struct bpf_dynptr *ptr) =
(void *) 198;

/*
 * bpf_ringbuf_submit_dynptr
 *
 * Submit reserved ring buffer sample, pointed to by *data*,
 * through the dynptr interface. This is a no-op if the dynptr is
 * invalid/null.
 *
 *
 * For more information on *flags*, please see
 * 'bpf_ringbuf_submit'.
 *
 * Returns
 * Nothing. Always succeeds.
 */
static void (* const bpf_ringbuf_submit_dynptr)(struct bpf_dynptr *ptr, __u64 flags) = (void *) 199;

/*
 * bpf_ringbuf_discard_dynptr
 *
 * Discard reserved ring buffer sample through the dynptr
 * interface. This is a no-op if the dynptr is invalid/null.
 *
 * For more information on *flags*, please see
 * 'bpf_ringbuf_discard'.
 *
 * Returns
 * Nothing. Always succeeds.
 */
static void (* const bpf_ringbuf_discard_dynptr)(struct bpf_dynptr *ptr, __u64 flags) = (void *) 200;
```

```

/*
 * bpf_dynptr_read
 *
 * Read *len* bytes from *src* into *dst*, starting from *offset*
 * into *src*.
 * *flags* is currently unused.
 *
 * Returns
 * 0 on success, -E2BIG if *offset* + *len* exceeds the length
 * of *src*'s data, -EINVAL if *src* is an invalid dynptr or if
 * *flags* is not 0.
 */
static long (* const bpf_dynptr_read)(void *dst, __u32 len, const struct bpf_dynptr
*src, __u32 offset, __u64 flags) = (void *) 201;

/*
 * bpf_dynptr_write
 *
 * Write *len* bytes from *src* into *dst*, starting from *offset*
 * into *dst*.
 *
 * *flags* must be 0 except for skb-type dynptrs.
 *
 * For skb-type dynptrs:
 *   * All data slices of the dynptr are automatically
 *     invalidated after **bpf_dynptr_write**(). This is
 *     because writing may pull the skb and change the
 *     underlying packet buffer.
 *
 *   * For *flags*, please see the flags accepted by
 *     **bpf_skb_store_bytes**().
 *
 * Returns
 * 0 on success, -E2BIG if *offset* + *len* exceeds the length
 * of *dst*'s data, -EINVAL if *dst* is an invalid dynptr or if *dst*
 * is a read-only dynptr or if *flags* is not correct. For skb-type dynptrs,
 * other errors correspond to errors returned by **bpf_skb_store_bytes**().
 */
static long (* const bpf_dynptr_write)(const struct bpf_dynptr *dst, __u32 offset, void *src, __u32 len, __u64 flags)
= (void *)
202;

/*
 * bpf_dynptr_data
 *
 * Get a pointer to the underlying dynptr data.
 *
 * *len* must be a statically known value. The returned data slice

```

```

* is invalidated whenever the dynptr is invalidated.
*
* skb and xdp type dynptrs may not use bpf_dynptr_data. They should
* instead use bpf_dynptr_slice and bpf_dynptr_slice_rdwr.
*
* Returns
* Pointer to the underlying dynptr data, NULL if the dynptr is
* read-only, if the dynptr is invalid, or if the offset and length
* is out of bounds.
*/
static void *(* const bpf_dynptr_data)(const struct bpf_dynptr *ptr, __u32 offset, __u32 len) = (void *) 203;

/*
* bpf_tcp_raw_gen_syncookie_ipv4
*
* Try to issue a SYN cookie for the packet with corresponding
* IPv4/TCP headers, *iph* and *th*, without depending on a
* listening socket.
*
* *iph* points to the IPv4 header.
*
* *th* points to the start of the TCP header, while *th_len*
* contains the length of the TCP header (at least
* **sizeof**\ (**struct tcphdr**)).
*
* Returns
* On success, lower 32 bits hold the generated SYN cookie in
* followed by 16 bits which hold the MSS value for that cookie,
* and the top 16 bits are unused.
*
* On failure, the returned value is one of the following:
*
* **EINVAL** if *th_len* is invalid.
*/
static __s64 (* const bpf_tcp_raw_gen_syncookie_ipv4)(struct iphdr *iph, struct tcphdr *th, __u32 th_len) = (void *) 204;

/*
* bpf_tcp_raw_gen_syncookie_ipv6
*
* Try to issue a SYN cookie for the packet with corresponding
* IPv6/TCP headers, *iph* and *th*, without depending on a
* listening socket.
*
* *iph* points to the IPv6 header.
*
* *th* points to the start of the TCP header, while *th_len*
* contains the length of the TCP header (at least

```

```

* **sizeof**\ (**struct tcphdr**).
*
* Returns
* On success, lower 32 bits hold the generated SYN cookie in
* followed by 16 bits which hold the MSS value for that cookie,
* and the top 16 bits are unused.
*
* On failure,
the returned value is one of the following:
*
* **-EINVAL** if *th_len* is invalid.
*
* **-EPROTONOSUPPORT** if CONFIG_IPV6 is not builtin.
*/
static __s64 (* const bpf_tcp_raw_gen_syncookie_ipv6)(struct ipv6hdr *iph, struct tcphdr *th, __u32 th_len) =
(void *) 205;

/*
* bpf_tcp_raw_check_syncookie_ipv4
*
* Check whether *iph* and *th* contain a valid SYN cookie ACK
* without depending on a listening socket.
*
* *iph* points to the IPv4 header.
*
* *th* points to the TCP header.
*
* Returns
* 0 if *iph* and *th* are a valid SYN cookie ACK.
*
* On failure, the returned value is one of the following:
*
* **-EACCES** if the SYN cookie is not valid.
*/
static long (* const bpf_tcp_raw_check_syncookie_ipv4)(struct iphdr *iph, struct tcphdr *th) = (void *) 206;

/*
* bpf_tcp_raw_check_syncookie_ipv6
*
* Check whether *iph* and *th* contain a valid SYN cookie ACK
* without depending on a listening socket.
*
* *iph* points to the IPv6 header.
*
* *th*
points to the TCP header.
*
* Returns

```

```

* 0 if *iph* and *th* are a valid SYN cookie ACK.
*
* On failure, the returned value is one of the following:
*
* **EACCES** if the SYN cookie is not valid.
*
* **-EPROTONOSUPPORT** if CONFIG_IPV6 is not builtin.
*/
static long (* const bpf_tcp_raw_check_syncookie_ipv6)(struct ipv6hdr *iph, struct tcphdr *th) = (void *) 207;

/*
* bpf_ktime_get_tai_ns
*
* A nonsettable system-wide clock derived from wall-clock time but
* ignoring leap seconds. This clock does not experience
* discontinuities and backwards jumps caused by NTP inserting leap
* seconds as CLOCK_REALTIME does.
*
* See: **clock_gettime**\ (**CLOCK_TAI**)
*
* Returns
* Current *ktime*.
*/
static __u64 (* const bpf_ktime_get_tai_ns)(void) = (void *) 208;

/*
* bpf_user_ringbuf_drain
*
* Drain samples from the specified user ring buffer, and invoke
* the provided callback for each such sample:
*
* long (*callback_fn)(const struct bpf_dynptr *dynptr,
* void *ctx);
*
* If **callback_fn** returns 0, the helper will continue to try
* and drain the next sample, up to a maximum of
* BPF_MAX_USER_RINGBUF_SAMPLES samples. If the return value is 1,
* the helper will skip the rest of the samples and return. Other
* return values are not used now, and will be rejected by the
* verifier.
*
* Returns
* The number of drained samples if no error was encountered while
* draining samples, or 0 if no samples were present in the ring
* buffer. If a user-space producer was epoll-waiting on this map,
* and at least one sample was drained, they will receive an event
* notification notifying them of available space in the ring
* buffer. If the BPF_RB_NO_WAKEUP flag is passed to this

```



```

* function, no wakeup notification will be sent. If the
* BPF_RB_FORCE_WAKEUP flag is passed, a wakeup notification will
* be sent even if no sample was drained.
*
* On failure, the returned value is one of the following:
*
* ** -EBUSY**
if the ring buffer is contended, and another calling
* context was concurrently draining the ring buffer.
*
* ** -EINVAL** if user-space is not properly tracking the ring
* buffer due to the producer position not being aligned to 8
* bytes, a sample not being aligned to 8 bytes, or the producer
* position not matching the advertised length of a sample.
*
* ** -E2BIG** if user-space has tried to publish a sample which is
* larger than the size of the ring buffer, or which cannot fit
* within a struct bpf_dynptr.
*/
static long (* const bpf_user_ringbuf_drain)(void *map, void *callback_fn, void *ctx, __u64 flags) = (void *) 209;

/*
* bpf_cgrp_storage_get
*
* Get a bpf_local_storage from the *cgroup*.
*
* Logically, it could be thought of as getting the value from
* a *map* with *cgroup* as the **key**. From this
* perspective, the usage is not much different from
* **bpf_map_lookup_elem**(*map*, **&*** *cgroup*) except this
* helper enforces the
key must be a cgroup struct and the map must also
* be a **BPF_MAP_TYPE_CGRP_STORAGE**.
*
* In reality, the local-storage value is embedded directly inside of the
* *cgroup* object itself, rather than being located in the
* **BPF_MAP_TYPE_CGRP_STORAGE** map. When the local-storage value is
* queried for some *map* on a *cgroup* object, the kernel will perform an
* O(n) iteration over all of the live local-storage values for that
* *cgroup* object until the local-storage value for the *map* is found.
*
* An optional *flags* (**BPF_LOCAL_STORAGE_GET_F_CREATE**) can be
* used such that a new bpf_local_storage will be
* created if one does not exist. *value* can be used
* together with **BPF_LOCAL_STORAGE_GET_F_CREATE** to specify
* the initial value of a bpf_local_storage. If *value* is
* **NULL**, the new bpf_local_storage will be zero initialized.
*

```

```

* Returns
* A bpf_local_storage pointer is returned on success.
*
* **NULL** if not found or there was
an error in adding
* a new bpf_local_storage.
*/
static void (* const bpf_cgrp_storage_get)(void *map, struct cgroup *cgroup, void *value, __u64 flags) = (void *)
210;

/*
* bpf_cgrp_storage_delete
*
* Delete a bpf_local_storage from a *cgroup*.
*
* Returns
* 0 on success.
*
* **-ENOENT** if the bpf_local_storage cannot be found.
*/
static long (* const bpf_cgrp_storage_delete)(void *map, struct cgroup *cgroup) = (void *) 211;
ISC License

```

Copyright (c) 2012-2016 Dave Collins <dave@davec.name>

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

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

Copyright (c) 2014, Evan Phoenix
All rights reserved.

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

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the Evan Phoenix nor the names of its contributors

may be used to endorse or promote products derived from this software without specific prior written permission.

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

/* SPDX-License-Identifier: (LGPL-2.1 OR BSD-2-Clause) */

```
#ifndef __BPF_TRACING_H__
#define __BPF_TRACING_H__
```

```
#include "bpf_helpers.h"
```

/* Scan the ARCH passed in from ARCH env variable (see Makefile) */

```
#if defined(__TARGET_ARCH_x86)
#define bpf_target_x86
#define bpf_target_defined
#elif defined(__TARGET_ARCH_s390)
#define bpf_target_s390
#define bpf_target_defined
#elif defined(__TARGET_ARCH_arm)
#define bpf_target_arm
#define bpf_target_defined
#elif defined(__TARGET_ARCH_arm64)
#define bpf_target_arm64
#define bpf_target_defined
#elif defined(__TARGET_ARCH_mips)
#define bpf_target_mips
#define bpf_target_defined
#elif defined(__TARGET_ARCH_powerpc)
#define bpf_target_powerpc
#define bpf_target_defined
#elif defined(__TARGET_ARCH_sparc)
#define bpf_target_sparc
#define bpf_target_defined
#elif defined(__TARGET_ARCH_riscv)
#define bpf_target_riscv
#define bpf_target_defined
#elif defined(__TARGET_ARCH_arc)
#define bpf_target_arc
#define bpf_target_defined
```

```

#elif defined(__TARGET_ARCH_loongarch)
#define
bpf_target_loongarch
#define bpf_target_defined
#else

/* Fall back to what the compiler says */
#if defined(__x86_64__)
#define bpf_target_x86
#define bpf_target_defined
#elif defined(__s390__)
#define bpf_target_s390
#define bpf_target_defined
#elif defined(__arm__)
#define bpf_target_arm
#define bpf_target_defined
#elif defined(__aarch64__)
#define bpf_target_arm64
#define bpf_target_defined
#elif defined(__mips__)
#define bpf_target_mips
#define bpf_target_defined
#elif defined(__powerpc__)
#define bpf_target_powerpc
#define bpf_target_defined
#elif defined(__sparc__)
#define bpf_target_sparc
#define bpf_target_defined
#elif defined(__riscv) && __riscv_xlen == 64
#define bpf_target_riscv
#define bpf_target_defined
#elif defined(__arc__)
#define bpf_target_arc
#define bpf_target_defined
#elif defined(__loongarch__)
#define bpf_target_loongarch
#define bpf_target_defined
#endif /* no compiler target */

#endif

#ifndef __BPF_TARGET_MISSING
#define
__BPF_TARGET_MISSING "GCC error \"Must specify a BPF target arch via __TARGET_ARCH_xxx\""
#endif

#if defined(bpf_target_x86)

```

```

/*
 * https://en.wikipedia.org/wiki/X86_calling_conventions#System_V_AMD64_ABI
 */

#if defined(__KERNEL__) || defined(__VMLINUX_H__)

#define __PT_PARM1_REG di
#define __PT_PARM2_REG si
#define __PT_PARM3_REG dx
#define __PT_PARM4_REG cx
#define __PT_PARM5_REG r8
#define __PT_PARM6_REG r9
/*
 * Syscall uses r10 for PARM4. See arch/x86/entry/entry_64.S:entry_SYSCALL_64
 * comments in Linux sources. And refer to syscall(2) manpage.
 */
#define __PT_PARM1_SYSCALL_REG __PT_PARM1_REG
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG r10
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG __PT_PARM6_REG

#define __PT_RET_REG sp
#define __PT_FP_REG bp
#define __PT_RC_REG ax
#define __PT_SP_REG sp
#define __PT_IP_REG ip

#else

#ifdef __i386__

/*
 * i386 kernel is built with -mregparm=3 */
#define __PT_PARM1_REG eax
#define __PT_PARM2_REG edx
#define __PT_PARM3_REG ecx
/* i386 syscall ABI is very different, refer to syscall(2) manpage */
#define __PT_PARM1_SYSCALL_REG ebx
#define __PT_PARM2_SYSCALL_REG ecx
#define __PT_PARM3_SYSCALL_REG edx
#define __PT_PARM4_SYSCALL_REG esi
#define __PT_PARM5_SYSCALL_REG edi
#define __PT_PARM6_SYSCALL_REG ebp

#define __PT_RET_REG esp
#define __PT_FP_REG ebp

```

```

#define __PT_RC_REG eax
#define __PT_SP_REG esp
#define __PT_IP_REG eip

#else /* __i386__ */

#define __PT_PARM1_REG rdi
#define __PT_PARM2_REG rsi
#define __PT_PARM3_REG rdx
#define __PT_PARM4_REG rcx
#define __PT_PARM5_REG r8
#define __PT_PARM6_REG r9

#define __PT_PARM1_SYSCALL_REG __PT_PARM1_REG
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG r10
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG __PT_PARM6_REG

#define __PT_RET_REG rsp
#define
__PT_FP_REG rbp
#define __PT_RC_REG rax
#define __PT_SP_REG rsp
#define __PT_IP_REG rip

#endif /* __i386__ */

#endif /* __KERNEL__ || __VMLINUX_H__ */

#elif defined(bpf_target_s390)

/*
 * https://github.com/IBM/s390x-abi/releases/download/v1.6/lzsabi_s390x.pdf
 */

struct pt_regs__s390 {
    unsigned long orig_gpr2;
};

/* s390 provides user_pt_regs instead of struct pt_regs to userspace */
#define __PT_REGS_CAST(x) ((const user_pt_regs*)(x))
#define __PT_PARM1_REG gprs[2]
#define __PT_PARM2_REG gprs[3]
#define __PT_PARM3_REG gprs[4]
#define __PT_PARM4_REG gprs[5]
#define __PT_PARM5_REG gprs[6]

```

```

#define __PT_PARM1_SYSCALL_REG orig_gpr2
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG gprs[7]
#define PT_REGS_PARM1_SYSCALL(x) PT_REGS_PARM1_CORE_SYSCALL(x)
#define PT_REGS_PARM1_CORE_SYSCALL(x) \
    BPF_CORE_READ((const
struct pt_regs____s390 *) (x), __PT_PARM1_SYSCALL_REG)

#define __PT_RET_REG gprs[14]
#define __PT_FP_REG gprs[11] /* Works only with CONFIG_FRAME_POINTER */
#define __PT_RC_REG gprs[2]
#define __PT_SP_REG gprs[15]
#define __PT_IP_REG psw.addr

#elif defined(bpf_target_arm)

/*
 * https://github.com/ARM-software/abi-aa/blob/main/aapcs32/aapcs32.rst#machine-registers
 */

#define __PT_PARM1_REG uregs[0]
#define __PT_PARM2_REG uregs[1]
#define __PT_PARM3_REG uregs[2]
#define __PT_PARM4_REG uregs[3]

#define __PT_PARM1_SYSCALL_REG __PT_PARM1_REG
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG uregs[4]
#define __PT_PARM6_SYSCALL_REG uregs[5]
#define __PT_PARM7_SYSCALL_REG uregs[6]

#define __PT_RET_REG uregs[14]
#define __PT_FP_REG uregs[11] /* Works only with CONFIG_FRAME_POINTER */
#define __PT_RC_REG uregs[0]
#define __PT_SP_REG uregs[13]
#define __PT_IP_REG uregs[12]

#elif
defined(bpf_target_arm64)

/*
 * https://github.com/ARM-software/abi-aa/blob/main/aapcs64/aapcs64.rst#machine-registers

```

```

*/

struct pt_regs__arm64 {
    unsigned long orig_x0;
};

/* arm64 provides struct user_pt_regs instead of struct pt_regs to userspace */
#define __PT_REGS_CAST(x) ((const struct user_pt_regs *) (x))
#define __PT_PARM1_REG regs[0]
#define __PT_PARM2_REG regs[1]
#define __PT_PARM3_REG regs[2]
#define __PT_PARM4_REG regs[3]
#define __PT_PARM5_REG regs[4]
#define __PT_PARM6_REG regs[5]
#define __PT_PARM7_REG regs[6]
#define __PT_PARM8_REG regs[7]

#define __PT_PARM1_SYSCALL_REG orig_x0
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG __PT_PARM6_REG
#define PT_REGS_PARM1_SYSCALL(x) PT_REGS_PARM1_CORE_SYSCALL(x)
#define PT_REGS_PARM1_CORE_SYSCALL(x) \
    BPF_CORE_READ((const struct pt_regs__arm64 *) (x), \
        __PT_PARM1_SYSCALL_REG)

#define __PT_RET_REG regs[30]
#define __PT_FP_REG regs[29] /* Works only with CONFIG_FRAME_POINTER */
#define __PT_RC_REG regs[0]
#define __PT_SP_REG sp
#define __PT_IP_REG pc

#ifdef bpf_target_mips

/*
 * N64 ABI is assumed right now.
 * https://en.wikipedia.org/wiki/MIPS_architecture#Calling_conventions
 */

#define __PT_PARM1_REG regs[4]
#define __PT_PARM2_REG regs[5]
#define __PT_PARM3_REG regs[6]
#define __PT_PARM4_REG regs[7]
#define __PT_PARM5_REG regs[8]
#define __PT_PARM6_REG regs[9]
#define __PT_PARM7_REG regs[10]

```



```

#define __PT_PARM8_REG regs[11]

#define __PT_PARM1_SYSCALL_REG __PT_PARM1_REG
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG /* only N32/N64 */
#define __PT_PARM6_SYSCALL_REG __PT_PARM6_REG /* only N32/N64 */

#define __PT_RET_REG regs[31]
#define __PT_FP_REG regs[30] /* Works only with CONFIG_FRAME_POINTER
*/
#define __PT_RC_REG regs[2]
#define __PT_SP_REG regs[29]
#define __PT_IP_REG cp0_epc

#elif defined(bpf_target_powerpc)

/*
* http://refspecs.linux-foundation.org/elf/elfspec\_ppc.pdf (page 3-14,
* section "Function Calling Sequence")
*/

#define __PT_PARM1_REG gpr[3]
#define __PT_PARM2_REG gpr[4]
#define __PT_PARM3_REG gpr[5]
#define __PT_PARM4_REG gpr[6]
#define __PT_PARM5_REG gpr[7]
#define __PT_PARM6_REG gpr[8]
#define __PT_PARM7_REG gpr[9]
#define __PT_PARM8_REG gpr[10]

/* powerpc does not select ARCH_HAS_SYSCALL_WRAPPER. */
#define PT_REGS_SYSCALL_REGS(ctx) ctx
#define __PT_PARM1_SYSCALL_REG orig_gpr3
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG __PT_PARM6_REG
#if !defined(__arch64__)
#define __PT_PARM7_SYSCALL_REG __PT_PARM7_REG /* only powerpc (not powerpc64) */
#endif

#define __PT_RET_REG regs[31]
#define
__PT_FP_REG __unsupported__
#define __PT_RC_REG gpr[3]

```

```

#define __PT_SP_REG sp
#define __PT_IP_REG nip

#ifdef bpf_target_sparc

/*
 * https://en.wikipedia.org/wiki/Calling_convention#SPARC
 */

#define __PT_PARM1_REG u_regs[UREG_I0]
#define __PT_PARM2_REG u_regs[UREG_I1]
#define __PT_PARM3_REG u_regs[UREG_I2]
#define __PT_PARM4_REG u_regs[UREG_I3]
#define __PT_PARM5_REG u_regs[UREG_I4]
#define __PT_PARM6_REG u_regs[UREG_I5]

#define __PT_PARM1_SYSCALL_REG __PT_PARM1_REG
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG __PT_PARM6_REG

#define __PT_RET_REG u_regs[UREG_I7]
#define __PT_FP_REG __unsupported__
#define __PT_RC_REG u_regs[UREG_I0]
#define __PT_SP_REG u_regs[UREG_FP]
/* Should this also be a bpf_target check for the sparc case? */
#ifdef __arch64__
#define __PT_IP_REG tpc
#else
#define
__PT_IP_REG pc
#endif

#ifdef bpf_target_riscv

/*
 * https://github.com/riscv-non-isa/riscv-elf-psabi-doc/blob/master/riscv-cc.adoc#risc-v-calling-conventions
 */

/* riscv provides struct user_regs_struct instead of struct pt_regs to userspace */
#define __PT_REGS_CAST(x) ((const struct user_regs_struct*)(x))
#define __PT_PARM1_REG a0
#define __PT_PARM2_REG a1
#define __PT_PARM3_REG a2
#define __PT_PARM4_REG a3
#define __PT_PARM5_REG a4

```

```

#define __PT_PARM6_REG a5
#define __PT_PARM7_REG a6
#define __PT_PARM8_REG a7

#define __PT_PARM1_SYSCALL_REG __PT_PARM1_REG
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG __PT_PARM6_REG

#define __PT_RET_REG ra
#define __PT_FP_REG s0
#define __PT_RC_REG a0
#define __PT_SP_REG sp
#define __PT_IP_REG pc

#elif defined(bpf_target_arc)

/*
 * Section "Function Calling Sequence" (page
 * 24):
 * https://raw.githubusercontent.com/foss-for-synopsys-dwc-arc-processors/toolchain/files/ARCV2\_ABI.pdf
 */

/* arc provides struct user_regs_struct instead of struct pt_regs to userspace */
#define __PT_REGS_CAST(x) ((const struct user_regs_struct *)(x))
#define __PT_PARM1_REG scratch.r0
#define __PT_PARM2_REG scratch.r1
#define __PT_PARM3_REG scratch.r2
#define __PT_PARM4_REG scratch.r3
#define __PT_PARM5_REG scratch.r4
#define __PT_PARM6_REG scratch.r5
#define __PT_PARM7_REG scratch.r6
#define __PT_PARM8_REG scratch.r7

/* arc does not select ARCH_HAS_SYSCALL_WRAPPER. */
#define PT_REGS_SYSCALL_REGS(ctx) ctx
#define __PT_PARM1_SYSCALL_REG __PT_PARM1_REG
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG __PT_PARM6_REG

#define __PT_RET_REG scratch.blink
#define __PT_FP_REG scratch.fp
#define __PT_RC_REG scratch.r0

```

```

#define
__PT_SP_REG scratch.sp
#define __PT_IP_REG scratch.ret

#elif defined(bpf_target_loongarch)

/*
 * https://docs.kernel.org/loongarch/introduction.html
 * https://loongson.github.io/LoongArch-Documentation/LoongArch-ELF-ABI-EN.html
 */

/* loongarch provides struct user_pt_regs instead of struct pt_regs to userspace */
#define __PT_REGS_CAST(x) ((const struct user_pt_regs *) (x))
#define __PT_PARM1_REG regs[4]
#define __PT_PARM2_REG regs[5]
#define __PT_PARM3_REG regs[6]
#define __PT_PARM4_REG regs[7]
#define __PT_PARM5_REG regs[8]
#define __PT_PARM6_REG regs[9]
#define __PT_PARM7_REG regs[10]
#define __PT_PARM8_REG regs[11]

/* loongarch does not select ARCH_HAS_SYSCALL_WRAPPER. */
#define PT_REGS_SYSCALL_REGS(ctx) ctx
#define __PT_PARM1_SYSCALL_REG __PT_PARM1_REG
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG
__PT_PARM6_REG

#define __PT_RET_REG regs[1]
#define __PT_FP_REG regs[22]
#define __PT_RC_REG regs[4]
#define __PT_SP_REG regs[3]
#define __PT_IP_REG csr_era

#endif

#if defined(bpf_target_defined)

struct pt_regs;

/* allow some architectures to override `struct pt_regs` */
#ifndef __PT_REGS_CAST
#define __PT_REGS_CAST(x) (x)
#endif

```

```

/*
 * Different architectures support different number of arguments passed
 * through registers. i386 supports just 3, some arches support up to 8.
 */
#ifndef __PT_PARM4_REG
#define __PT_PARM4_REG __unsupported__
#endif
#ifndef __PT_PARM5_REG
#define __PT_PARM5_REG __unsupported__
#endif
#ifndef __PT_PARM6_REG
#define __PT_PARM6_REG __unsupported__
#endif
#ifndef __PT_PARM7_REG
#define __PT_PARM7_REG __unsupported__
#endif
#ifndef __PT_PARM8_REG
#define __PT_PARM8_REG __unsupported__
#endif
/*
 * Similarly, syscall-specific conventions might differ between function call
 * conventions within each architecture. All supported architectures pass
 * either
 * 6 or 7 syscall arguments in registers.
 *
 * See syscall(2) manpage for succinct table with information on each arch.
 */
#ifndef __PT_PARM7_SYSCALL_REG
#define __PT_PARM7_SYSCALL_REG __unsupported__
#endif

#define PT_REGS_PARM1(x) (__PT_REGS_CAST(x)->__PT_PARM1_REG)
#define PT_REGS_PARM2(x) (__PT_REGS_CAST(x)->__PT_PARM2_REG)
#define PT_REGS_PARM3(x) (__PT_REGS_CAST(x)->__PT_PARM3_REG)
#define PT_REGS_PARM4(x) (__PT_REGS_CAST(x)->__PT_PARM4_REG)
#define PT_REGS_PARM5(x) (__PT_REGS_CAST(x)->__PT_PARM5_REG)
#define PT_REGS_PARM6(x) (__PT_REGS_CAST(x)->__PT_PARM6_REG)
#define PT_REGS_PARM7(x) (__PT_REGS_CAST(x)->__PT_PARM7_REG)
#define PT_REGS_PARM8(x) (__PT_REGS_CAST(x)->__PT_PARM8_REG)
#define PT_REGS_RET(x) (__PT_REGS_CAST(x)->__PT_RET_REG)
#define PT_REGS_FP(x) (__PT_REGS_CAST(x)->__PT_FP_REG)
#define PT_REGS_RC(x) (__PT_REGS_CAST(x)->__PT_RC_REG)
#define PT_REGS_SP(x) (__PT_REGS_CAST(x)->__PT_SP_REG)
#define PT_REGS_IP(x) (__PT_REGS_CAST(x)->__PT_IP_REG)

#define PT_REGS_PARM1_CORE(x)

```

```

BPF_CORE_READ(__PT_REGS_CAST(x), __PT_PARM1_REG)
#define PT_REGS_PARM2_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_PARM2_REG)
#define PT_REGS_PARM3_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_PARM3_REG)
#define PT_REGS_PARM4_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_PARM4_REG)
#define PT_REGS_PARM5_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_PARM5_REG)
#define PT_REGS_PARM6_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_PARM6_REG)
#define PT_REGS_PARM7_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_PARM7_REG)
#define PT_REGS_PARM8_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_PARM8_REG)
#define PT_REGS_RET_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_RET_REG)
#define PT_REGS_FP_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_FP_REG)
#define PT_REGS_RC_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_RC_REG)
#define PT_REGS_SP_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_SP_REG)
#define PT_REGS_IP_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_IP_REG)

#if defined(bpf_target_powerpc)

#define
BPF_KPROBE_READ_RET_IP(ip, ctx) ({ (ip) = (ctx)->link; })
#define BPF_KRETPROBE_READ_RET_IP BPF_KPROBE_READ_RET_IP

#elif defined(bpf_target_sparc)

#define BPF_KPROBE_READ_RET_IP(ip, ctx) ({ (ip) = PT_REGS_RET(ctx); })
#define BPF_KRETPROBE_READ_RET_IP BPF_KPROBE_READ_RET_IP

#else

#define BPF_KPROBE_READ_RET_IP(ip, ctx) \
({ bpf_probe_read_kernel(&(ip), sizeof(ip), (void *)PT_REGS_RET(ctx)); })
#define BPF_KRETPROBE_READ_RET_IP(ip, ctx) \
({ bpf_probe_read_kernel(&(ip), sizeof(ip), (void *) (PT_REGS_FP(ctx) + sizeof(ip))); })

#endif

#ifndef PT_REGS_PARM1_SYSCALL
#define PT_REGS_PARM1_SYSCALL(x) (__PT_REGS_CAST(x)->__PT_PARM1_SYSCALL_REG)
#define PT_REGS_PARM1_CORE_SYSCALL(x) BPF_CORE_READ(__PT_REGS_CAST(x), \
__PT_PARM1_SYSCALL_REG)
#endif
#ifndef PT_REGS_PARM2_SYSCALL
#define PT_REGS_PARM2_SYSCALL(x) (__PT_REGS_CAST(x)->__PT_PARM2_SYSCALL_REG)
#define PT_REGS_PARM2_CORE_SYSCALL(x) BPF_CORE_READ(__PT_REGS_CAST(x), \
__PT_PARM2_SYSCALL_REG)
#endif
#ifndef
PT_REGS_PARM3_SYSCALL
#define PT_REGS_PARM3_SYSCALL(x) (__PT_REGS_CAST(x)->__PT_PARM3_SYSCALL_REG)
#define PT_REGS_PARM3_CORE_SYSCALL(x) BPF_CORE_READ(__PT_REGS_CAST(x),

```

```

__PT_PARM3_SYSCALL_REG)
#endif
#ifndef PT_REGS_PARM4_SYSCALL
#define PT_REGS_PARM4_SYSCALL(x) (__PT_REGS_CAST(x)->__PT_PARM4_SYSCALL_REG)
#define PT_REGS_PARM4_CORE_SYSCALL(x) BPF_CORE_READ(__PT_REGS_CAST(x),
__PT_PARM4_SYSCALL_REG)
#endif
#ifndef PT_REGS_PARM5_SYSCALL
#define PT_REGS_PARM5_SYSCALL(x) (__PT_REGS_CAST(x)->__PT_PARM5_SYSCALL_REG)
#define PT_REGS_PARM5_CORE_SYSCALL(x) BPF_CORE_READ(__PT_REGS_CAST(x),
__PT_PARM5_SYSCALL_REG)
#endif
#ifndef PT_REGS_PARM6_SYSCALL
#define PT_REGS_PARM6_SYSCALL(x) (__PT_REGS_CAST(x)->__PT_PARM6_SYSCALL_REG)
#define PT_REGS_PARM6_CORE_SYSCALL(x) BPF_CORE_READ(__PT_REGS_CAST(x),
__PT_PARM6_SYSCALL_REG)
#endif
#ifndef PT_REGS_PARM7_SYSCALL
#define PT_REGS_PARM7_SYSCALL(x) (__PT_REGS_CAST(x)->__PT_PARM7_SYSCALL_REG)
#define PT_REGS_PARM7_CORE_SYSCALL(x) BPF_CORE_READ(__PT_REGS_CAST(x),
__PT_PARM7_SYSCALL_REG)
#endif

#else /* defined(bpf_target_defined) */

#define PT_REGS_PARM1(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM2(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM3(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM4(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM5(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM6(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM7(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM8(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_RET(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_FP(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_RC(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_SP(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_IP(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })

#define PT_REGS_PARM1_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l;
})
#define PT_REGS_PARM2_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM3_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM4_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM5_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM6_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM7_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM8_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })

```

```

#define PT_REGS_RET_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_FP_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_RC_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_SP_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_IP_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })

#define BPF_KPROBE_READ_RET_IP(ip, ctx) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define BPF_KRETPROBE_READ_RET_IP(ip, ctx) ({ _Pragma(__BPF_TARGET_MISSING);
0l; })

#define PT_REGS_PARM1_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM2_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM3_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM4_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM5_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM6_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM7_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })

#define PT_REGS_PARM1_CORE_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM2_CORE_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM3_CORE_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM4_CORE_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM5_CORE_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM6_CORE_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define
PT_REGS_PARM7_CORE_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })

#endif /* defined(bpf_target_defined) */

/*
 * When invoked from a syscall handler kprobe, returns a pointer to a
 * struct pt_regs containing syscall arguments and suitable for passing to
 * PT_REGS_PARMn_SYSCALL() and PT_REGS_PARMn_CORE_SYSCALL().
 */
#ifndef PT_REGS_SYSCALL_REGS
/* By default, assume that the arch selects ARCH_HAS_SYSCALL_WRAPPER. */
#define PT_REGS_SYSCALL_REGS(ctx) ((struct pt_regs *)PT_REGS_PARM1(ctx))
#endif

#ifndef __bpf_concat
#define __bpf_concat(a, b) a ## b
#endif
#ifndef __bpf_apply
#define __bpf_apply(fn, n) __bpf_concat(fn, n)
#endif
#ifndef __bpf_nth
#define __bpf_nth(_, _1, _2, _3, _4, _5, _6, _7, _8, _9, _a, _b, _c, N, ...) N
#endif

```



```

#ifndef __bpf_narg
#define __bpf_narg(...) __bpf_nth(, ##__VA_ARGS__, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0)
#endif

#define __bpf_ctx_cast0()      ctx
#define __bpf_ctx_cast1(x)     __bpf_ctx_cast0(), (void *)ctx[0]
#define
__bpf_ctx_cast2(x, args...) __bpf_ctx_cast1(args), (void *)ctx[1]
#define __bpf_ctx_cast3(x, args...) __bpf_ctx_cast2(args), (void *)ctx[2]
#define __bpf_ctx_cast4(x, args...) __bpf_ctx_cast3(args), (void *)ctx[3]
#define __bpf_ctx_cast5(x, args...) __bpf_ctx_cast4(args), (void *)ctx[4]
#define __bpf_ctx_cast6(x, args...) __bpf_ctx_cast5(args), (void *)ctx[5]
#define __bpf_ctx_cast7(x, args...) __bpf_ctx_cast6(args), (void *)ctx[6]
#define __bpf_ctx_cast8(x, args...) __bpf_ctx_cast7(args), (void *)ctx[7]
#define __bpf_ctx_cast9(x, args...) __bpf_ctx_cast8(args), (void *)ctx[8]
#define __bpf_ctx_cast10(x, args...) __bpf_ctx_cast9(args), (void *)ctx[9]
#define __bpf_ctx_cast11(x, args...) __bpf_ctx_cast10(args), (void *)ctx[10]
#define __bpf_ctx_cast12(x, args...) __bpf_ctx_cast11(args), (void *)ctx[11]
#define __bpf_ctx_cast(args...) __bpf_apply(__bpf_ctx_cast, __bpf_narg(args))(args)

/*
 * BPF_PROG is a convenience wrapper for generic
 * tp_btf/fentry/fexit and
 * similar kinds of BPF programs, that accept input arguments as a single
 * pointer to untyped u64 array, where each u64 can actually be a typed
 * pointer or integer of different size. Instead of requiring user to write
 * manual casts and work with array elements by index, BPF_PROG macro
 * allows user to declare a list of named and typed input arguments in the
 * same syntax as for normal C function. All the casting is hidden and
 * performed transparently, while user code can just assume working with
 * function arguments of specified type and name.
 *
 * Original raw context argument is preserved as well as 'ctx' argument.
 * This is useful when using BPF helpers that expect original context
 * as one of the parameters (e.g., for bpf_perf_event_output()).
 */
#define BPF_PROG(name, args...) \
    name(unsigned long long *ctx); \
    static __always_inline typeof(name(0)) \
    ____##name(unsigned long long *ctx, ##args); \
    typeof(name(0)) \
    name(unsigned long long *ctx) \
    { \
        _Pragma("GCC diagnostic push") \
        _Pragma("GCC diagnostic ignored \"-Wint-conversion\"") \
        return ____##name(__bpf_ctx_cast(args)); \
        _Pragma("GCC diagnostic pop") \
    }

```

```

static __always_inline typeof(name(0))      \
____##name(unsigned long long *ctx, ##args)

#ifndef __bpf_nth2
#define __bpf_nth2(, _1, _2, _3, _4, _5, _6, _7, _8, _9, _10, _11, _12, _13, \
    _14, _15, _16, _17, _18, _19, _20, _21, _22, _23, _24, N, ...) N
#endif
#ifndef __bpf_narg2
#define __bpf_narg2(...) \
    __bpf_nth2(, ##__VA_ARGS__, 12, 12, 11, 11, 10, 10, 9, 9, 8, 8, 7, 7, \
        6, 6, 5, 5, 4, 4, 3, 3, 2, 2, 1, 1, 0)
#endif

#define __bpf_treg_cnt(t) \
    __builtin_choose_expr(sizeof(t) == 1, 1, \
    __builtin_choose_expr(sizeof(t) == 2, 1, \
    __builtin_choose_expr(sizeof(t) == 4, 1, \
    __builtin_choose_expr(sizeof(t) == 8, 1, \
    __builtin_choose_expr(sizeof(t) == 16, 2, \
        (void)0))))

#define
    __bpf_reg_cnt0() (0)
#define __bpf_reg_cnt1(t, x) (__bpf_reg_cnt0() + __bpf_treg_cnt(t))
#define __bpf_reg_cnt2(t, x, args...) (__bpf_reg_cnt1(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt3(t, x, args...) (__bpf_reg_cnt2(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt4(t, x, args...) (__bpf_reg_cnt3(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt5(t, x, args...) (__bpf_reg_cnt4(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt6(t, x, args...) (__bpf_reg_cnt5(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt7(t, x, args...) (__bpf_reg_cnt6(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt8(t, x, args...) (__bpf_reg_cnt7(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt9(t, x, args...) (__bpf_reg_cnt8(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt10(t, x, args...) (__bpf_reg_cnt9(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt11(t, x, args...) (__bpf_reg_cnt10(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt12(t, x, args...) (__bpf_reg_cnt11(args)
    + __bpf_treg_cnt(t))
#define __bpf_reg_cnt(args...) __bpf_apply(__bpf_reg_cnt, __bpf_narg2(args))(args)

#define __bpf_union_arg(t, x, n) \
    __builtin_choose_expr(sizeof(t) == 1, ({ union { __u8 z[1]; t x; } __t = { .z = {ctx[n]}}; __t.x; }), \
    __builtin_choose_expr(sizeof(t) == 2, ({ union { __u16 z[1]; t x; } __t = { .z = {ctx[n]}}; __t.x; }), \
    __builtin_choose_expr(sizeof(t) == 4, ({ union { __u32 z[1]; t x; } __t = { .z = {ctx[n]}}; __t.x; }), \
    __builtin_choose_expr(sizeof(t) == 8, ({ union { __u64 z[1]; t x; } __t = { .z = {ctx[n]}}; __t.x; }), \
    __builtin_choose_expr(sizeof(t) == 16, ({ union { __u64 z[2]; t x; } __t = { .z = {ctx[n], ctx[n + 1]}}; __t.x; }), \
        (void)0))))

#define __bpf_ctx_arg0(n, args...)
#define __bpf_ctx_arg1(n, t, x) , __bpf_union_arg(t, x, n - __bpf_reg_cnt1(t, x))

```

```

#define __bpf_ctx_arg2(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt2(t, x, args)) __bpf_ctx_arg1(n,
args)
#define __bpf_ctx_arg3(n,
t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt3(t, x, args)) __bpf_ctx_arg2(n, args)
#define __bpf_ctx_arg4(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt4(t, x, args)) __bpf_ctx_arg3(n,
args)
#define __bpf_ctx_arg5(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt5(t, x, args)) __bpf_ctx_arg4(n,
args)
#define __bpf_ctx_arg6(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt6(t, x, args)) __bpf_ctx_arg5(n,
args)
#define __bpf_ctx_arg7(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt7(t, x, args)) __bpf_ctx_arg6(n,
args)
#define __bpf_ctx_arg8(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt8(t, x, args)) __bpf_ctx_arg7(n,
args)
#define __bpf_ctx_arg9(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt9(t, x, args)) __bpf_ctx_arg8(n,
args)
#define __bpf_ctx_arg10(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt10(t, x, args))
__bpf_ctx_arg9(n, args)
#define __bpf_ctx_arg11(n,
t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt11(t, x, args)) __bpf_ctx_arg10(n, args)
#define __bpf_ctx_arg12(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt12(t, x, args))
__bpf_ctx_arg11(n, args)
#define __bpf_ctx_arg(args...) __bpf_apply(__bpf_ctx_arg, __bpf_narg2(args))(__bpf_reg_cnt(args), args)

#define __bpf_ctx_decl0()
#define __bpf_ctx_decl1(t, x) , t x
#define __bpf_ctx_decl2(t, x, args...) , t x __bpf_ctx_decl1(args)
#define __bpf_ctx_decl3(t, x, args...) , t x __bpf_ctx_decl2(args)
#define __bpf_ctx_decl4(t, x, args...) , t x __bpf_ctx_decl3(args)
#define __bpf_ctx_decl5(t, x, args...) , t x __bpf_ctx_decl4(args)
#define __bpf_ctx_decl6(t, x, args...) , t x __bpf_ctx_decl5(args)
#define __bpf_ctx_decl7(t, x, args...) , t x __bpf_ctx_decl6(args)
#define __bpf_ctx_decl8(t, x, args...) , t x __bpf_ctx_decl7(args)
#define __bpf_ctx_decl9(t, x, args...) , t x __bpf_ctx_decl8(args)
#define __bpf_ctx_decl10(t,
x, args...) , t x __bpf_ctx_decl9(args)
#define __bpf_ctx_decl11(t, x, args...) , t x __bpf_ctx_decl10(args)
#define __bpf_ctx_decl12(t, x, args...) , t x __bpf_ctx_decl11(args)
#define __bpf_ctx_decl(args...) __bpf_apply(__bpf_ctx_decl, __bpf_narg2(args))(args)

/*
* BPF_PROG2 is an enhanced version of BPF_PROG in order to handle struct
* arguments. Since each struct argument might take one or two u64 values
* in the trampoline stack, argument type size is needed to place proper number
* of u64 values for each argument. Therefore, BPF_PROG2 has different
* syntax from BPF_PROG. For example, for the following BPF_PROG syntax:
*
* int BPF_PROG(test2, int a, int b) { ... }

```

```

*
* the corresponding BPF_PROG2 syntax is:
*
* int BPF_PROG2(test2, int, a, int, b) { ... }
*
* where type and the corresponding argument name are separated by comma.
*
* Use BPF_PROG2 macro if one of the arguments might be a struct/union larger
* than 8 bytes:
*
* int BPF_PROG2(test_struct_arg,
struct bpf_testmod_struct_arg_1, a, int, b,
* int, c, int, d, struct bpf_testmod_struct_arg_2, e, int, ret)
* {
* // access a, b, c, d, e, and ret directly
* ...
* }
*/
#define BPF_PROG2(name, args...) \
name(unsigned long long *ctx); \
static __always_inline typeof(name(0)) \
__##name(unsigned long long *ctx __bpf_ctx_decl(args)); \
typeof(name(0)) name(unsigned long long *ctx) \
{ \
return __##name(ctx __bpf_ctx_arg(args)); \
} \
static __always_inline typeof(name(0)) \
__##name(unsigned long long *ctx __bpf_ctx_decl(args))

struct pt_regs;

#define __bpf_kprobe_args0() ctx
#define __bpf_kprobe_args1(x) __bpf_kprobe_args0(), (void *)PT_REGS_PARM1(ctx)
#define __bpf_kprobe_args2(x, args...) __bpf_kprobe_args1(args), (void *)PT_REGS_PARM2(ctx)
#define __bpf_kprobe_args3(x, args...) __bpf_kprobe_args2(args), (void *)PT_REGS_PARM3(ctx)
#define __bpf_kprobe_args4(x, \
args...) __bpf_kprobe_args3(args), (void *)PT_REGS_PARM4(ctx)
#define __bpf_kprobe_args5(x, args...) __bpf_kprobe_args4(args), (void *)PT_REGS_PARM5(ctx)
#define __bpf_kprobe_args6(x, args...) __bpf_kprobe_args5(args), (void *)PT_REGS_PARM6(ctx)
#define __bpf_kprobe_args7(x, args...) __bpf_kprobe_args6(args), (void *)PT_REGS_PARM7(ctx)
#define __bpf_kprobe_args8(x, args...) __bpf_kprobe_args7(args), (void *)PT_REGS_PARM8(ctx)
#define __bpf_kprobe_args(args...) __bpf_apply(__bpf_kprobe_args, __bpf_narg(args))(args)

/*
* BPF_KPROBE serves the same purpose for kprobes as BPF_PROG for
* tp_btf/fentry/fexit BPF programs. It hides the underlying platform-specific
* low-level way of getting kprobe input arguments from struct pt_regs, and
* provides a familiar typed and named function arguments syntax and

```

```

* semantics of accessing kprobe input paremeters.
*
* Original struct pt_regs* context is preserved as 'ctx' argument. This might
* be necessary when using BPF helpers
  like bpf_perf_event_output().
*/
#define BPF_KPROBE(name, args...)      \
name(struct pt_regs *ctx);            \
static __always_inline typeof(name(0)) \
__##name(struct pt_regs *ctx, ##args); \
typeof(name(0)) name(struct pt_regs *ctx) \
{                                       \
    _Pragma("GCC diagnostic push")      \
    _Pragma("GCC diagnostic ignored \"-Wint-conversion\"") \
    return __##name(__bpf_kprobe_args(args)); \
    _Pragma("GCC diagnostic pop")       \
}                                       \
static __always_inline typeof(name(0)) \
__##name(struct pt_regs *ctx, ##args)

#define __bpf_kretprobe_args0()      ctx
#define __bpf_kretprobe_args1(x)     __bpf_kretprobe_args0(), (void *)PT_REGS_RC(ctx)
#define __bpf_kretprobe_args(args...) __bpf_apply(__bpf_kretprobe_args, __bpf_narg(args))(args)

/*
* BPF_KRETPROBE is similar to BPF_KPROBE, except, it only provides optional
* return value (in addition to `struct pt_regs *ctx`), but no input
* arguments, because they
  will be clobbered by the time probed function
* returns.
*/
#define BPF_KRETPROBE(name, args...)  \
name(struct pt_regs *ctx);            \
static __always_inline typeof(name(0)) \
__##name(struct pt_regs *ctx, ##args); \
typeof(name(0)) name(struct pt_regs *ctx) \
{                                       \
    _Pragma("GCC diagnostic push")      \
    _Pragma("GCC diagnostic ignored \"-Wint-conversion\"") \
    return __##name(__bpf_kretprobe_args(args)); \
    _Pragma("GCC diagnostic pop")       \
}                                       \
static __always_inline typeof(name(0)) __##name(struct pt_regs *ctx, ##args)

/* If kernel has CONFIG_ARCH_HAS_SYSCALL_WRAPPER, read pt_regs directly */
#define __bpf_syscall_args0()          ctx
#define __bpf_syscall_args1(x)         __bpf_syscall_args0(), (void *)PT_REGS_PARM1_SYSCALL(regs)
#define __bpf_syscall_args2(x, args...) __bpf_syscall_args1(args), (void *)PT_REGS_PARM2_SYSCALL(regs)

```

```

#define __bpf_syscall_args3(x, args...) __bpf_syscall_args2(args),
(void *)PT_REGS_PARM3_SYSCALL(regs)
#define __bpf_syscall_args4(x, args...) __bpf_syscall_args3(args), (void *)PT_REGS_PARM4_SYSCALL(regs)
#define __bpf_syscall_args5(x, args...) __bpf_syscall_args4(args), (void *)PT_REGS_PARM5_SYSCALL(regs)
#define __bpf_syscall_args6(x, args...) __bpf_syscall_args5(args), (void *)PT_REGS_PARM6_SYSCALL(regs)
#define __bpf_syscall_args7(x, args...) __bpf_syscall_args6(args), (void *)PT_REGS_PARM7_SYSCALL(regs)
#define __bpf_syscall_args(args...) __bpf_apply(__bpf_syscall_args, __bpf_narg(args))(args)

/* If kernel doesn't have CONFIG_ARCH_HAS_SYSCALL_WRAPPER, we have to BPF_CORE_READ from
pt_regs */
#define __bpf_syswrap_args0() ctx
#define __bpf_syswrap_args1(x) __bpf_syswrap_args0(), (void
*)PT_REGS_PARM1_CORE_SYSCALL(regs)
#define __bpf_syswrap_args2(x, args...) __bpf_syswrap_args1(args), (void
*)PT_REGS_PARM2_CORE_SYSCALL(regs)
#define __bpf_syswrap_args3(x, args...) __bpf_syswrap_args2(args), (void
*)PT_REGS_PARM3_CORE_SYSCALL(regs)
#define __bpf_syswrap_args4(x, args...) __bpf_syswrap_args3(args), (void
*)PT_REGS_PARM4_CORE_SYSCALL(regs)
#define __bpf_syswrap_args5(x, args...) __bpf_syswrap_args4(args), (void
*)PT_REGS_PARM5_CORE_SYSCALL(regs)
#define __bpf_syswrap_args6(x, args...) __bpf_syswrap_args5(args), (void
*)PT_REGS_PARM6_CORE_SYSCALL(regs)
#define __bpf_syswrap_args7(x, args...) __bpf_syswrap_args6(args), (void
*)PT_REGS_PARM7_CORE_SYSCALL(regs)
#define __bpf_syswrap_args(args...) __bpf_apply(__bpf_syswrap_args, __bpf_narg(args))(args)

/*
* BPF_KSYSCALL is a variant of BPF_KPROBE, which is intended for
* tracing syscall functions, like __x64_sys_close. It hides the underlying
* platform-specific low-level way of getting syscall input arguments from
* struct pt_regs, and provides a familiar typed and named function arguments
* syntax and semantics of accessing syscall input parameters.
*
* Original struct pt_regs * context is preserved as 'ctx' argument.
This might
* be necessary when using BPF helpers like bpf_perf_event_output().
*
* At the moment BPF_KSYSCALL does not transparently handle all the calling
* convention quirks for the following syscalls:
*
* - mmap(): __ARCH_WANT_SYS_OLD_MMAP.
* - clone(): CONFIG_CLONE_BACKWARDS, CONFIG_CLONE_BACKWARDS2 and
CONFIG_CLONE_BACKWARDS3.
* - socket-related syscalls: __ARCH_WANT_SYS_SOCKETCALL.
* - compat syscalls.
*
* This may or may not change in the future. User needs to take extra measures

```

```

* to handle such quirks explicitly, if necessary.
*
* This macro relies on BPF CO-RE support and virtual __kconfig externs.
*/
#define BPF_KSYSCALL(name, args...) \
    name(struct pt_regs *ctx); \
    extern _Bool LINUX_HAS_SYSCALL_WRAPPER __kconfig; \
    static __always_inline typeof(name(0)) \
    ____##name(struct pt_regs *ctx, ##args); \
    typeof(name(0)) name(struct pt_regs *ctx) \
    { \
        struct pt_regs *regs = LINUX_HAS_SYSCALL_WRAPPER \
            ? (struct pt_regs *)PT_REGS_PARM1(ctx) \
            : ctx; \
        _Pragma("GCC diagnostic push") \
        _Pragma("GCC diagnostic ignored \"-Wint-conversion\"") \
        if (LINUX_HAS_SYSCALL_WRAPPER) \
            return ____##name(__bpf_syswrap_args(args)); \
        else \
            return ____##name(__bpf_syscall_args(args)); \
        _Pragma("GCC diagnostic pop") \
    } \
    static __always_inline typeof(name(0)) \
    ____##name(struct pt_regs *ctx, ##args)

#define BPF_KPROBE_SYSCALL BPF_KSYSCALL

/* BPF_UPROBE and BPF_URETPROBE are identical to BPF_KPROBE and BPF_KRETPROBE,
 * but are named way less confusingly for SEC("uprobe") and SEC("uretprobe")
 * use cases.
 */
#define BPF_UPROBE(name, args...) BPF_KPROBE(name, ##args)
#define BPF_URETPROBE(name, args...) BPF_KRETPROBE(name, ##args)

```

#endif

Valid-License-Identifier: BSD-2-Clause

SPDX-URL: <https://spdx.org/licenses/BSD-2-Clause.html>

Usage-Guide:

To use the BSD 2-clause "Simplified" License put the following SPDX tag/value pair into a comment according to the placement guidelines in the licensing rules documentation:

SPDX-License-Identifier: BSD-2-Clause

License-Text:

Copyright (c) 2015 The Libbpf Authors. All rights reserved.

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

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

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

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

Copyright (C) 2013 Blake Mizerany

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

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

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

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

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

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

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

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

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

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices

stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Copyright (c) 2013 The Go Authors. All rights reserved.

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

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

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

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

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

Copyright (c) 2018 The Go Authors. All rights reserved.

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

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

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

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

Copyright (c) 2016 Mail.Ru Group

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

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

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

SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2012 Pter Surnyi. Portions Copyright (c) 2009 The Go Authors. All rights reserved.

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

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

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

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

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

Files other than internal/golang/* licensed under:

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication

on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
- You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and

hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You
may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

internal/golang/* files licensed under:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
- * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN

CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015, Gengo, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Gengo, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO

EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

When donating the randfill project to the CNCF, we could not reach all the gofuzz contributors to sign the CNCF CLA. As such, according to the CNCF rules to donate a repository, we must add a NOTICE referencing section 7 of the CLA with a list of developers who could not be reached.

`7. Should You wish to submit work that is not Your original creation, You may submit it to the Foundation separately from any Contribution, identifying the complete details of its source and of any license or other restriction (including, but not limited to, related patents, trademarks, and license agreements) of which you are personally aware, and conspicuously marking the work as "Submitted on behalf of a third-party: [named here]".`

Submitted on behalf of a third-party: @dnephin (Daniel Nephin)
Submitted on behalf of a third-party: @Aleksi (Alexey Palazhchenko)
Submitted on behalf of a third-party: @bbigras (Bruno Bigras)
Submitted on behalf of a third-party: @samirkut (Samir)
Submitted on behalf of a
third-party: @posener (Eyal Posener)
Submitted on behalf of a third-party: @Ashikpaul (Ashik Paul)
Submitted on behalf of a third-party: @kwongtailau (Kwongtai)

Submitted on behalf of a third-party: @ericcornelissen (Eric Cornelissen)

Submitted on behalf of a third-party: @eclipseo (Robert-Andr Mauchin)

Submitted on behalf of a third-party: @yanzhoupan (Andrew Pan)

Submitted on behalf of a third-party: @STRRL (Zhiqiang ZHOU)

Submitted on behalf of a third-party: @disconnect3d (Disconnect3d)

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2020 Bruno Scheufler

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object

form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

Copyright (c) 2012 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
- * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2017 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2015 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google LLC nor the names of its

contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2010 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices

stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2014 The gofuzz Authors

Copyright 2025 The Kubernetes Authors

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of
the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Prometheus instrumentation library for Go applications
Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

The following components are included in this product:

perks - a fork of <https://github.com/bmizerany/perks>
<https://github.com/beorn7/perks>
Copyright 2013-2015 Blake Mizerany, Bjrn Rabenstein
See <https://github.com/beorn7/perks/blob/master/README.md> for license details.

Go support for Protocol Buffers - Google's data interchange format
<http://github.com/golang/protobuf/>
Copyright 2010 The Go Authors
See source code for license details.
Copyright (c) 2012,2013 Ernest Micklei

MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,

EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
/* SPDX-License-Identifier: (LGPL-2.1 OR BSD-2-Clause) */
#ifndef __BPF_HELPERS__
#define __BPF_HELPERS__

/*
 * Note that bpf programs need to include either
 * vmlinux.h (auto-generated from BTF) or linux/types.h
 * in advance since bpf_helper_defs.h uses such types
 * as __u64.
 */
#include "bpf_helper_defs.h"

#define __uint(name, val) int (*name)[val]
#define __type(name, val) typeof(val) *name
#define __array(name, val) typeof(val) *name[]
#define __ulong(name, val) enum { __bpf_concat(__unique_value, __COUNTER__) = val } name

/*
 * Helper macro to place programs, maps, license in
 * different sections in elf_bpf file. Section names
 * are interpreted by libbpf depending on the context (BPF programs, BPF maps,
 * extern variables, etc).
 * To allow use of SEC() with externs (e.g., for extern .maps declarations),
 * make sure __attribute__((unused)) doesn't trigger compilation warning.
 */
#if __GNUC__ && !__clang__

/*
 * Pragma macros are broken on GCC
 * https://gcc.gnu.org/bugzilla/show_bug.cgi?id=55578
 * https://gcc.gnu.org/bugzilla/show_bug.cgi?id=90400
 */
#define SEC(name) __attribute__((section(name), used))

#else

#define SEC(name) \
    _Pragma("GCC diagnostic push") \
    _Pragma("GCC diagnostic ignored \"-Wignored-attributes\"") \
    __attribute__((section(name), used)) \
    _Pragma("GCC diagnostic pop")

#endif

#endif
```



```

/* Avoid 'linux/stddef.h' definition of '__always_inline'. */
#undef __always_inline
#define __always_inline inline __attribute__((always_inline))

#ifndef __noinline
#define __noinline __attribute__((noinline))
#endif
#ifndef __weak
#define __weak __attribute__((weak))
#endif

/*
 * Use __hidden attribute to mark a non-static BPF subprogram effectively
 * static for BPF verifier's verification algorithm purposes, allowing more
 * extensive and permissive BPF verification process, taking into account
 * subprogram's caller context.
 */
#define __hidden __attribute__((visibility("hidden")))

/* When utilizing vmlinux.h
 * with BPF CO-RE, user BPF programs can't include
 * any system-level headers (such as stddef.h, linux/version.h, etc), and
 * commonly-used macros like NULL and KERNEL_VERSION aren't available through
 * vmlinux.h. This just adds unnecessary hurdles and forces users to re-define
 * them on their own. So as a convenience, provide such definitions here.
 */
#ifndef NULL
#define NULL ((void *)0)
#endif

#ifndef KERNEL_VERSION
#define KERNEL_VERSION(a, b, c) (((a) << 16) + ((b) << 8) + ((c) > 255 ? 255 : (c)))
#endif

/*
 * Helper macros to manipulate data structures
 */

/* offsetof() definition that uses __builtin_offset() might not preserve field
 * offset CO-RE relocation properly, so force-redefine offsetof() using
 * old-school approach which works with CO-RE correctly
 */
#undef offsetof
#define offsetof(type, member) ((unsigned long)&((type *)0)->member)

/* redefined container_of() to ensure we use the above offsetof() macro */
#undef container_of

```

```

#define container_of(ptr, type, member) \
({ \
    void \
    *__mptr = (void *)(ptr); \
    ((type *)__mptr - offsetof(type, member)); \
})

/*
 * Compiler (optimization) barrier.
 */
#ifndef barrier
#define barrier() asm volatile("" ::: "memory")
#endif

/* Variable-specific compiler (optimization) barrier. It's a no-op which makes
 * compiler believe that there is some black box modification of a given
 * variable and thus prevents compiler from making extra assumption about its
 * value and potential simplifications and optimizations on this variable.
 *
 * E.g., compiler might often delay or even omit 32-bit to 64-bit casting of
 * a variable, making some code patterns unverifiable. Putting barrier_var()
 * in place will ensure that cast is performed before the barrier_var()
 * invocation, because compiler has to pessimistically assume that embedded
 * asm section might perform some extra operations on that variable.
 *
 * This is a variable-specific variant of more global barrier().
 */
#ifndef barrier_var
#define
    barrier_var(var) asm volatile("" : "+r"(var))
#endif

/*
 * Helper macro to throw a compilation error if __bpf_unreachable() gets
 * built into the resulting code. This works given BPF back end does not
 * implement __builtin_trap(). This is useful to assert that certain paths
 * of the program code are never used and hence eliminated by the compiler.
 *
 * For example, consider a switch statement that covers known cases used by
 * the program. __bpf_unreachable() can then reside in the default case. If
 * the program gets extended such that a case is not covered in the switch
 * statement, then it will throw a build error due to the default case not
 * being compiled out.
 */
#ifndef __bpf_unreachable
#define __bpf_unreachable() __builtin_trap()
#endif

```

```

/*
 * Helper function to perform a tail call with a constant/immediate map slot.
 */
#if __clang_major__ >= 8 && defined(__bpf__)
static __always_inline void
bpf_tail_call_static(void *ctx, const void *map, const __u32 slot)
{
    if
    (!__builtin_constant_p(slot))
        __bpf_unreachable();

/*
 * Provide a hard guarantee that LLVM won't optimize setting r2 (map
 * pointer) and r3 (constant map index) from _different paths_ ending
 * up at the _same_ call insn as otherwise we won't be able to use the
 * jmpq/nopl retpoline-free patching by the x86-64 JIT in the kernel
 * given they mismatch. See also d2e4c1e6c294 ("bpf: Constant map key
 * tracking for prog array pokes") for details on verifier tracking.
 *
 * Note on clobber list: we need to stay in-line with BPF calling
 * convention, so even if we don't end up using r0, r4, r5, we need
 * to mark them as clobber so that LLVM doesn't end up using them
 * before / after the call.
 */
asm volatile("r1 = %[ctx]\n\t"
             "r2 = %[map]\n\t"
             "r3 = %[slot]\n\t"
             "call 12"
             :: [ctx]"r"(ctx), [map]"r"(map), [slot]"i"(slot)
             : "r0", "r1", "r2", "r3", "r4", "r5");
}
#endif

enum libbpf_pin_type {
    LIBBPF_PIN_NONE,
    /* PIN_BY_NAME: pin
     maps by name (in /sys/fs/bpf by default) */
    LIBBPF_PIN_BY_NAME,
};

enum libbpf_tristate {
    TRI_NO = 0,
    TRI_YES = 1,
    TRI_MODULE = 2,
};

#define __kconfig __attribute__((section(".kconfig")))
#define __ksym __attribute__((section(".ksyms")))

```

```

#define __kptr_untrusted __attribute__((btf_type_tag("kptr_untrusted")))
#define __kptr __attribute__((btf_type_tag("kptr")))
#define __percpu_kptr __attribute__((btf_type_tag("percpu_kptr")))

#define bpf_ksym_exists(sym) ({ \
    _Static_assert(!__builtin_constant_p(!sym), #sym " should be marked as __weak"); \
    !!sym; \
})

#define __arg_ctx __attribute__((btf_decl_tag("arg:ctx")))
#define __arg_nonnull __attribute__((btf_decl_tag("arg:nonnull")))
#define __arg_nullable __attribute__((btf_decl_tag("arg:nullable")))
#define __arg_trusted __attribute__((btf_decl_tag("arg:trusted")))
#define __arg_arena __attribute__((btf_decl_tag("arg:arena")))

#ifndef __bpf_concat
#define __bpf_concat(a, b) a ## b
#endif
#ifndef __bpf_apply
#define
    __bpf_apply(fn, n) __bpf_concat(fn, n)
#endif
#ifndef __bpf_nth
#define __bpf_nth(, _1, _2, _3, _4, _5, _6, _7, _8, _9, _a, _b, _c, N, ...) N
#endif
#ifndef __bpf_narg
#define __bpf_narg(...) \
    __bpf_nth(, ##__VA_ARGS__, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0)
#endif

#define __bpf_fill0(arr, p, x) do { } while (0)
#define __bpf_fill1(arr, p, x) arr[p] = x
#define __bpf_fill2(arr, p, x, args...) arr[p] = x; __bpf_fill1(arr, p + 1, args)
#define __bpf_fill3(arr, p, x, args...) arr[p] = x; __bpf_fill2(arr, p + 1, args)
#define __bpf_fill4(arr, p, x, args...) arr[p] = x; __bpf_fill3(arr, p + 1, args)
#define __bpf_fill5(arr, p, x, args...) arr[p] = x; __bpf_fill4(arr, p + 1, args)
#define __bpf_fill6(arr, p, x, args...) arr[p] = x; __bpf_fill5(arr, p + 1, args)
#define __bpf_fill7(arr, p, x, args...) arr[p] = x; __bpf_fill6(arr, p + 1, args)
#define __bpf_fill8(arr, p, x, args...) arr[p] = x; __bpf_fill7(arr, p + 1, args)
#define __bpf_fill9(arr, p, x, args...) \
    arr[p] = x; __bpf_fill8(arr, p + 1, args)
#define __bpf_fill10(arr, p, x, args...) arr[p] = x; __bpf_fill9(arr, p + 1, args)
#define __bpf_fill11(arr, p, x, args...) arr[p] = x; __bpf_fill10(arr, p + 1, args)
#define __bpf_fill12(arr, p, x, args...) arr[p] = x; __bpf_fill11(arr, p + 1, args)
#define __bpf_fill(arr, args...) \
    __bpf_apply(__bpf_fill, __bpf_narg(args))(arr, 0, args)

/*

```

```

* BPF_SEQ_PRINTF to wrap bpf_seq_printf to-be-printed values
* in a structure.
*/
#define BPF_SEQ_PRINTF(seq, fmt, args...) \
({ \
    static const char __fmt[] = fmt; \
    unsigned long long __param[__bpf_narg(args)]; \
    \
    _Pragma("GCC diagnostic push") \
    _Pragma("GCC diagnostic ignored \"-Wint-conversion\"") \
    __bpf_fill(__param, args); \
    _Pragma("GCC diagnostic pop") \
    \
    bpf_seq_printf(seq, __fmt, sizeof(__fmt), \
        __param, sizeof(__param)); \
})

/*
* BPF_SNPRINTF wraps the bpf_snprintf helper with variadic arguments instead of
* an array of u64.
*/
#define BPF_SNPRINTF(out, out_size, fmt, args...) \
({ \
    static const char __fmt[] = fmt; \
    unsigned long long __param[__bpf_narg(args)]; \
    \
    _Pragma("GCC diagnostic push") \
    _Pragma("GCC diagnostic ignored \"-Wint-conversion\"") \
    __bpf_fill(__param, args); \
    _Pragma("GCC diagnostic pop") \
    \
    bpf_snprintf(out, out_size, __fmt, \
        __param, sizeof(__param)); \
})

#ifdef BPF_NO_GLOBAL_DATA
#define BPF_PRINTK_FMT_MOD
#else
#define BPF_PRINTK_FMT_MOD static const
#endif

#define __bpf_printk(fmt, ...) \
({ \
    BPF_PRINTK_FMT_MOD char ____fmt[] = fmt; \
    bpf_trace_printk(____fmt, sizeof(____fmt), \
        ##__VA_ARGS__); \
})

```

```

/*
 * __bpf_vprintk wraps the bpf_trace_vprintk helper with variadic arguments
 * instead of an array of u64.
 */
#define __bpf_vprintk(fmt, args...) \
({ \
    static const char __fmt[] = fmt; \
    unsigned long long __param[__bpf_narg(args)]; \
    \
    _Pragma("GCC \
diagnostic push") \
    _Pragma("GCC diagnostic ignored \"-Wint-conversion\"") \
    __bpf_fill(__param, args); \
    _Pragma("GCC diagnostic pop") \
    \
    bpf_trace_vprintk(__fmt, sizeof(__fmt), \
        __param, sizeof(__param)); \
})

/* Use __bpf_printk when bpf_printk call has 3 or fewer fmt args
 * Otherwise use __bpf_vprintk
 */
#define __bpf_pick_printk(...) \
    __bpf_nth(, ##__VA_ARGS__, __bpf_vprintk, __bpf_vprintk, __bpf_vprintk, \
        __bpf_vprintk, __bpf_vprintk, __bpf_vprintk, __bpf_vprintk, \
        __bpf_vprintk, __bpf_vprintk, __bpf_printk /*3*/, __bpf_printk /*2*/, \
        __bpf_printk /*1*/, __bpf_printk /*0*/)

/* Helper macro to print out debug messages */
#define bpf_printk(fmt, args...) __bpf_pick_printk(args)(fmt, ##args)

struct bpf_iter_num;

extern int bpf_iter_num_new(struct bpf_iter_num *it, int start, int end) __weak __ksym;
extern int *bpf_iter_num_next(struct bpf_iter_num *it) __weak __ksym;
extern void bpf_iter_num_destroy(struct bpf_iter_num
    *it) __weak __ksym;

#ifndef bpf_for_each
/* bpf_for_each(iter_type, cur_elem, args...) provides generic construct for
 * using BPF open-coded iterators without having to write mundane explicit
 * low-level loop logic. Instead, it provides for()-like generic construct
 * that can be used pretty naturally. E.g., for some hypothetical cgroup
 * iterator, you'd write:
 *
 * struct cgroup *cg, *parent_cg = <...>;
 *
 * bpf_for_each(cgroup, cg, parent_cg, CG_ITER_CHILDREN) {

```

```

*   bpf_printk("Child cgroup id = %d", cg->cgroup_id);
*   if (cg->cgroup_id == 123)
*       break;
* }
*
* I.e., it looks almost like high-level for each loop in other languages,
* supports continue/break, and is verifiable by BPF verifier.
*
* For iterating integers, the difference between bpf_for_each(num, i, N, M)
* and bpf_for(i, N, M) is in that bpf_for() provides additional proof to
* verifier that i is in [N, M) range, and in bpf_for_each() case i is `int
*`, not just `int`. So for integers
bpf_for() is more convenient.
*
* Note: this macro relies on C99 feature of allowing to declare variables
* inside for() loop, bound to for() loop lifetime. It also utilizes GCC
* extension: __attribute__((cleanup(<func>))), supported by both GCC and
* Clang.
*/
#define bpf_for_each(type, cur, args...) for (    \
/* initialize and define destructor */    \
struct bpf_iter_##type __it __attribute__((aligned(8), /* enforce, just in case */ \
cleanup(bpf_iter_##type##_destroy))), \
/* __p pointer is just to call bpf_iter_##type##_new() *once* to init __it */ \
*__p __attribute__((unused)) = (    \
bpf_iter_##type##_new(&__it, ##args), \
/* this is a workaround for Clang bug: it currently doesn't emit BTF */ \
/* for bpf_iter_##type##_destroy() when used from cleanup() attribute */ \
(void)bpf_iter_##type##_destroy, (void *)0); \
/* iteration and termination check */    \
(((cur) = bpf_iter_##type##_next(&__it)));    \
)
#endif
/* bpf_for_each */

#ifndef bpf_for
/* bpf_for(i, start, end) implements a for()-like looping construct that sets
* provided integer variable *i* to values starting from *start* through,
* but not including, *end*. It also proves to BPF verifier that *i* belongs
* to range [start, end), so this can be used for accessing arrays without
* extra checks.
*
* Note: *start* and *end* are assumed to be expressions with no side effects
* and whose values do not change throughout bpf_for() loop execution. They do
* not have to be statically known or constant, though.
*
* Note: similarly to bpf_for_each(), it relies on C99 feature of declaring for()
* loop bound variables and cleanup attribute, supported by GCC and Clang.

```

```

*/
#define bpf_for(i, start, end) for (    \
/* initialize and define destructor */    \
struct bpf_iter_num __it __attribute__((aligned(8), /* enforce, just in case */ \
cleanup(bpf_iter_num_destroy))), \
/* __p pointer is necessary to
call bpf_iter_num_new() *once* to init __it */ \
__p __attribute__((unused)) = (    \
bpf_iter_num_new(&__it, (start), (end)), \
/* this is a workaround for Clang bug: it currently doesn't emit BTF */ \
/* for bpf_iter_num_destroy() when used from cleanup() attribute */ \
(void)bpf_iter_num_destroy, (void *)0); \
({    \
/* iteration step */    \
int *__t = bpf_iter_num_next(&__it);    \
/* termination and bounds check */    \
(__t && ((i) = *__t, (i) >= (start) && (i) < (end)))); \
});    \
)
#endif /* bpf_for */

#ifndef bpf_repeat
/* bpf_repeat(N) performs N iterations without exposing iteration number
*
* Note: similarly to bpf_for_each(), it relies on C99 feature of declaring for()
* loop bound variables and cleanup attribute, supported by GCC and Clang.
*/
#define bpf_repeat(N) for (    \
/* initialize and define destructor */    \
struct bpf_iter_num __it __attribute__((aligned(8), /* enforce,
just in case */ \
cleanup(bpf_iter_num_destroy))), \
/* __p pointer is necessary to call bpf_iter_num_new() *once* to init __it */ \
__p __attribute__((unused)) = (    \
bpf_iter_num_new(&__it, 0, (N)), \
/* this is a workaround for Clang bug: it currently doesn't emit BTF */ \
/* for bpf_iter_num_destroy() when used from cleanup() attribute */ \
(void)bpf_iter_num_destroy, (void *)0); \
bpf_iter_num_next(&__it);    \
/* nothing here */    \
)
#endif /* bpf_repeat */

#endif

```

Copyright (c) 2013, The GoGo Authors. All rights reserved.

Protocol Buffers for Go with Gadgets

Go support for Protocol Buffers - Google's data interchange format

Copyright 2010 The Go Authors. All rights reserved.

<https://github.com/golang/protobuf>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2014 Sam Ghods

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2012 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The forked go-yaml.v3 library under this project is covered by two different licenses (MIT and Apache):

MIT License

The following files were ported to Go from C files of libyaml, and thus are still covered by their original MIT license, with the additional copyright starting in 2011 when the project was ported over:

apic.go emitterc.go parserc.go readerc.go scannerc.go
writerc.go yamlh.go yamlprivateh.go

Copyright (c)
2006-2010 Kirill Simonov
Copyright (c) 2006-2011 Kirill Simonov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License

All the remaining project files are covered by the Apache license:

Copyright (c) 2011-2019 Canonical Ltd

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The forked go-yaml.v2 library under the project is covered by an Apache license:

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative

Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section)

patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS"

BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and
limitations under the License.

Copyright (c) 2009,2014 Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.

- * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LGPL-2.1 OR BSD-2-Clause

1.23 protobuf 1.36.10

1.23.1 Available under license :

Copyright (c) 2018 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.24 utmps 0.1.3.1-r0

1.24.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This

General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in

whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an

announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you

received the program in object code
or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy
the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program

specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands

`show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

```
# Automatically generated by apkbuild-cpan, template 4
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkbuild-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
buildid="$srcdir/$_pkgreal-$pkgver"

build() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
        INSTALLDIRS=vendor \
        NO_PACKLIST=1 \
        NO_PERLLOCAL=1
    make
}

check() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    make test
}

package() {
    make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85"
```

```

BSD-Resource-1.2911.tar.gz
"
# Contributor: Maxim Karasev <begs@disroot.org>
# Maintainer: mio <miyopan@e.email>
pkgname=bsd-games
pkgver=3.3
pkgrel=2
pkgdesc="traditional text mode games from BSD"
url="https://bsd-games.sourceforge.io/"
arch="all"
license="BSD-3-Clause"
install="$pkgname.post-install"
makedepends="ncurses-dev coreutils" # configure causes busybox expr error
subpackages="$pkgname-doc"
source="https://sourceforge.net/projects/bsd-games/files/bsd-games-$pkgver.tar.gz
0001-bsd-games-scores-dir.patch
"
options="!check" # no tests

build() {
# some GNU autoconf options are ignored, but it works
./configure \
--build=$CBUILD \
--host=$CHOST \
--prefix=/usr \
--sysconfdir=/etc \
--mandir=/usr/share/man \
--scoredir=/usr/share/$pkgname
make
}

package() {
make DESTDIR="$pkgdir" install
install -Dm644 LICENSE "$pkgdir"/usr/share/licenses/bsd-games/LICENSE
}

sha512sums="
aaf36d09d4fe68514a5c279063d7e77a9a9a84c447037cba974b29faa07eb80c2aab2379d0699c196b5c27ffe1b2bb2c6
8d3390143e0e62d5e3d210ccfb61294
bsd-games-3.3.tar.gz
e1def0d4835a877242c4173fae862e278a7858738e0bacacbb045a42bcabec2aa7c72d000bbd2421ce35fe66cdc1f52c0
577b7e278029ca5ed50f7aab22319f6 0001-bsd-games-scores-dir.patch
"
MIT License

```

Copyright (c) 2023 Leaf Corcoran

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl
pkgver=2.3.21
pkgrel=8
pkgdesc="A library for high-performance 2D graphics"
url="https://www.levien.com/libart/"
arch="all"
options="!check" # No test suite.
license="LGPL-2.0-or-later"
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart-lgpl/2.3/libart-lgpl-$pkgver.tar.bz2"
"
```

```
builddir="$srcdir"/libart-lgpl-$pkgver
```

```
prepare() {
    update_config_sub
    default_prepare
}
```

```
build() {
    ./configure \
        --build=$CBUILD \
        --host=$CHOST \
        --prefix=/usr
    make
}
```

```
package() {
    make DESTDIR="$pkgdir" install
}
```

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"
mini_sendmail - accept email on behalf of real sendmail

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING,
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>

Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>

pkgname=spdx-licenses

pkgver=3.26.0

pkgrel=0

pkgdesc="Various data formats for the SPDX License List"

url="https://spdx.org/"

arch="noarch"

options="!check" # no test suite

license="CC-BY-3.0"

source="license-list-data-\$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v\$pkgver.tar.gz"

builddir="\$srcdir/license-list-data-\$pkgver"

subpackages="\$pkgname-list"

_types="html json rdfa rdnt rdfturtle rdfxml template text"

for type in \$_types; do

subpackages="\$subpackages \$pkgname-\$type:_subpkg"

done

```

package() {
    mkdir -p "$pkgdir"
}

_subpkg() {
    local type=${subpkgname/$pkgname-/}
    pkgdesc="$pkgdesc ($type)"
    install_if="$pkgname=$pkgver-r$pkgrel"
    mkdir -p "$subpkgdir"/usr/share/spdx
    cp -r "$builddir"/$type "$subpkgdir"/usr/share/spdx/
}

list() {
    pkgdesc="$pkgdesc (licence list)"
    mkdir -p "$subpkgdir"/usr/share/spdx
    local i; for i in "$builddir"/text/*.txt;
    do
        local license=${i##*/}
        echo ${license%.*} >> "$subpkgdir"/usr/share/spdx/license.lst
    done
}

sha512sums="
67e618d4642dfe4f366935d9514d6f7941d3711e77c49bc1b340ac572d36015981b8f10c72a9f80f1d83e132ae7e5093c
0155e0c0e786e4777ee16d1cb7b885a license-list-data-3.26.0.tar.gz
"

# Contributor: Patrycja Rosa <alpine@ptrcnull.me>
# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
cdefs.h
queue.h
tree.h
"

builddir="$srcdir"
options="!check" # just headers

package() {
    mkdir -p "$pkgdir"
    install -Dm644 -t "$pkgdir"/usr/include/sys \
        cdefs.h queue.h tree.h
}

```

```
}
sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178 cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750 queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3 tree.h
"
```

Original Copyright (c) 1985 by Supoj Sutanthavibul

Parts Copyright (c) 1989-2012 by Brian V. Smith

Parts Copyright (c) 1991 by Paul King

Other Copyrights may be found in various files

Main Xfig copyright notice:

Any party obtaining a copy of these files is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish and/or distribute copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.

No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Parts Copyright (C) 1993 by Alan Richardson (mppa3@uk.ac.sussex.syma)

The text rotation code in w_rottext.c was written by Alan Richardson.

The above copyright notice holds for this work as well.

Parts Copyright (c) 1994 by Anthony Dekker

The Kohonen neural network code for color optimization was written by Anthony Dekker.

Parts Copyright (c) 1995 by C. Blanc and C. Schlick

The X-Spline code was written Carole Blanc (blanc@labri.u-bordeaux.fr) and Christophe Schlick (schlick@labri.u-bordeaux.fr) starting from an initial implementation done by C. Feuille, S. Grosbois, L. Maziere and L. Minihot as a student practice (Universite Bordeaux, France). For additional information about X-splines, see:

"X-Splines: A Spline Model Designed for the End User" by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95
<http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/sig1.html>

Contributor: Dhruvin Gandhi <contact@dhruvin>

Maintainer: Patrycja Rosa <alpine@ptrcnull.me>

pkgname=py3-license-expression

_pkgname=license_expression

pkgver=30.4.1

pkgrel=0

pkgdesc="Library to parse, compare, simplify and normalize license expressions"

url="https://github.com/nexB/license-expression"

arch="noarch"

license="Apache-2.0"

```

depends="python3 py3-boolean.py"
makedepends="py3-setuptools py3-setuptools_scm py3-gpep517 py3-wheel"
checkdepends="py3-pytest py3-pytest-xdist"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/l/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir/${_pkgname}-${pkgver}"

build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}

check() {
    python3 -m venv --clear --without-pip --system-site-packages .testenv
    .testenv/bin/python3 -m installer .dist/*.whl
    .testenv/bin/python3 -m pytest
}

package() {
    python3 -m installer -d "$pkgdir" \
        .dist/*.whl
}

sha512sums="
114c630f098939757e49187e35b4e54883266a2ee9ca283724992420ef4c860f9606a8342da0ab74d93627c10805f312
fda3f699cbc7c4fbd47a71e220440fb0
    license_expression-30.4.1.tar.gz
"

/* $NetBSD: queue.h,v 1.70 2015/11/02 15:21:23 christos Exp $ */

/*
 * Copyright (c) 1991, 1993
 * The Regents of the University of California. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND

```

* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
 BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * @(#)queue.h 8.5 (Berkeley) 8/20/94
 */

```
#ifndef _SYS_QUEUE_H_
#define _SYS_QUEUE_H_
```

```
/*
 * This file defines five types of data structures: singly-linked lists,
 * lists, simple queues, tail queues, and circular queues.
 *
 * A singly-linked list is headed by a single forward pointer. The
 * elements are singly linked for minimum space and
 * pointer manipulation
 * overhead at the expense of O(n) removal for arbitrary elements. New
 * elements can be added to the list after an existing element or at the
 * head of the list. Elements being removed from the head of the list
 * should use the explicit macro for this purpose for optimum
 * efficiency. A singly-linked list may only be traversed in the forward
 * direction. Singly-linked lists are ideal for applications with large
 * datasets and few or no removals or for implementing a LIFO queue.
 *
 * A list is headed by a single forward pointer (or an array of forward
 * pointers for a hash table header). The elements are doubly linked
 * so that an arbitrary element can be removed without a need to
 * traverse the list. New elements can be added to the list before
 * or after an existing element or at the head of the list. A list
 * may only be traversed in the forward direction.
 *
 * A simple queue is headed by a pair of pointers, one the head of the
 * list and the
 * other to the tail of the list. The elements are singly
 * linked to save space, so elements can only be removed from the
 * head of the list. New elements can be added to the list after
 * an existing element, at the head of the list, or at the end of the
 * list. A simple queue may only be traversed in the forward direction.
 */
```



```

* A tail queue is headed by a pair of pointers, one to the head of the
* list and the other to the tail of the list. The elements are doubly
* linked so that an arbitrary element can be removed without a need to
* traverse the list. New elements can be added to the list before or
* after an existing element, at the head of the list, or at the end of
* the list. A tail queue may be traversed in either direction.
*
* A circle queue is headed by a pair of pointers, one to the head of the
* list and the other to the tail of the list. The elements are doubly
* linked so that an arbitrary element can be removed without a need to
* traverse the list. New elements
* can be added to the list before or after
* an existing element, at the head of the list, or at the end of the list.
* A circle queue may be traversed in either direction, but has a more
* complex end of list detection.
*
* For details on the use of these macros, see the queue(3) manual page.
*/

```

```

/*
* Include the definition of NULL only on NetBSD because sys/null.h
* is not available elsewhere. This conditional makes the header
* portable and it can simply be dropped verbatim into any system.
* The caveat is that on other systems some other header
* must provide NULL before the macros can be used.
*/
#ifdef __NetBSD__
#include <sys/null.h>
#endif

```

```

#if defined(QEUEDEBUG)
# if defined(_KERNEL)
# define QUEUEDEBUG_ABORT(...) panic(__VA_ARGS__)
# else
# include <err.h>
# define QUEUEDEBUG_ABORT(...) err(1, __VA_ARGS__)
# endif
#endif

```

```

/*
* Singly-linked List definitions.
*/
#define SLIST_HEAD(name, type) \
struct name { \
    struct type *slh_first; /* first \
    element */ \
}

```

```

#define SLIST_HEAD_INITIALIZER(head)  \
{ NULL }

#define SLIST_ENTRY(type)  \
struct {  \
    struct type *sle_next; /* next element */  \
}

/*
 * Singly-linked List access methods.
 */
#define SLIST_FIRST(head) ((head)->slh_first)
#define SLIST_END(head) NULL
#define SLIST_EMPTY(head) ((head)->slh_first == NULL)
#define SLIST_NEXT(elm, field) ((elm)->field.sle_next)

#define SLIST_FOREACH(var, head, field)  \
for((var) = (head)->slh_first;  \
    (var) != SLIST_END(head);  \
    (var) = (var)->field.sle_next)

#define SLIST_FOREACH_SAFE(var, head, field, tvar)  \
for ((var) = SLIST_FIRST((head));  \
    (var) != SLIST_END(head) &&  \
    ((tvar) = SLIST_NEXT((var), field), 1);  \
    (var) = (tvar))

/*
 * Singly-linked List functions.
 */
#define SLIST_INIT(head) do {  \
    (head)->slh_first = SLIST_END(head);  \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_AFTER(slistelm, elm, field) do {  \
    (elm)->field.sle_next = (slistelm)->field.sle_next;  \
    (slistelm)->field.sle_next  \
    = (elm);  \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_HEAD(head, elm, field) do {  \
    (elm)->field.sle_next = (head)->slh_first;  \
    (head)->slh_first = (elm);  \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_AFTER(slistelm, field) do {  \
    (slistelm)->field.sle_next =  \
        SLIST_NEXT(SLIST_NEXT((slistelm), field), field);  \
} while (/*CONSTCOND*/0)

```

```

} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_HEAD(head, field) do { \
    (head)->slh_first = (head)->slh_first->field.sle_next; \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE(head, elm, type, field) do { \
    if ((head)->slh_first == (elm)) { \
        SLIST_REMOVE_HEAD((head), field); \
    } \
    else { \
        struct type *curelm = (head)->slh_first; \
        while(curelm->field.sle_next != (elm)) \
            curelm = curelm->field.sle_next; \
        curelm->field.sle_next = \
            curelm->field.sle_next->field.sle_next; \
    } \
} while (/*CONSTCOND*/0)

/*
 * List definitions.
 */
#define LIST_HEAD(name, type) \
struct name { \
    struct type *lh_first; /* first element */ \
}

#define LIST_HEAD_INITIALIZER(head) \
{ NULL }

#define LIST_ENTRY(type) \
struct { \
    struct type *le_next; /* next element */ \
    struct type **le_prev; /* address of previous next element */ \
}

/*
 * List access methods.
 */
#define LIST_FIRST(head) ((head)->lh_first)
#define LIST_END(head) NULL
#define LIST_EMPTY(head) ((head)->lh_first == LIST_END(head))
#define LIST_NEXT(elm, field) ((elm)->field.le_next)

#define LIST_FOREACH(var, head, field) \
for ((var) = ((head)->lh_first); \
     (var) != LIST_END(head); \

```

```

    (var) = ((var)->field.le_next))

#define LIST_FOREACH_SAFE(var, head, field, tvar) \
for ((var) = LIST_FIRST((head)); \
    (var) != LIST_END(head) && \
    ((tvar) = LIST_NEXT((var), field), 1); \
    (var) = (tvar))

#define LIST_MOVE(head1, head2) do { \
    LIST_INIT((head2)); \
    if (!LIST_EMPTY((head1))) \
    { \
        (head2)->lh_first = (head1)->lh_first; \
        LIST_INIT((head1)); \
    } \
} while (/*CONSTCOND*/0)

/*
 * List functions.
 */
#ifdef QUEUEDEBDEBUG
#define QUEUEDEBDEBUG_LIST_INSERT_HEAD(head, elm, field) \
if ((head)->lh_first && \
    (head)->lh_first->field.le_prev != &(head)->lh_first) \
    QUEUEDEBDEBUG_ABORT("LIST_INSERT_HEAD %p %s:%d", (head), \
        __FILE__, __LINE__);
#define QUEUEDEBDEBUG_LIST_OP(elm, field) \
if ((elm)->field.le_next && \
    (elm)->field.le_next->field.le_prev != \
    &(elm)->field.le_next) \
    QUEUEDEBDEBUG_ABORT("LIST_* forw %p %s:%d", (elm), \
        __FILE__, __LINE__); \
if (*(elm)->field.le_prev != (elm)) \
    QUEUEDEBDEBUG_ABORT("LIST_* back %p %s:%d", (elm), \
        __FILE__, __LINE__);
#define QUEUEDEBDEBUG_LIST_POSTREMOVE(elm, field) \
(elm)->field.le_next = (void *)1L; \
(elm)->field.le_prev = (void *)1L;
#else
#define QUEUEDEBDEBUG_LIST_INSERT_HEAD(head, elm, field)
#define QUEUEDEBDEBUG_LIST_OP(elm, field)
#define QUEUEDEBDEBUG_LIST_POSTREMOVE(elm, field)
#endif

#define LIST_INIT(head) do { \
    (head)->lh_first = LIST_END(head); \
} while (/*CONSTCOND*/0)

```

```

#define LIST_INSERT_AFTER(listelm, elm, field) do { \
    QUEUEDEBUG_LIST_OP((listelm), field) \
    if (((elm)->field.le_next = (listelm)->field.le_next) != \
        LIST_END(head)) \
        (listelm)->field.le_next->field.le_prev = \
            &(elm)->field.le_next; \
    (listelm)->field.le_next = (elm); \
    (elm)->field.le_prev = &(listelm)->field.le_next; \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_BEFORE(listelm, elm, field) do { \
    QUEUEDEBUG_LIST_OP((listelm), field) \
    (elm)->field.le_prev = (listelm)->field.le_prev; \
    (elm)->field.le_next = (listelm); \
    *(listelm)->field.le_prev = (elm); \
    (listelm)->field.le_prev = &(elm)->field.le_next; \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBUG_LIST_INSERT_HEAD((head), (elm), field) \
    if (((elm)->field.le_next \
        = (head)->lh_first) != LIST_END(head)) \
        (head)->lh_first->field.le_prev = &(elm)->field.le_next; \
    (head)->lh_first = (elm); \
    (elm)->field.le_prev = &(head)->lh_first; \
} while (/*CONSTCOND*/0)

#define LIST_REMOVE(elm, field) do { \
    QUEUEDEBUG_LIST_OP((elm), field) \
    if ((elm)->field.le_next != NULL) \
        (elm)->field.le_next->field.le_prev = \
            (elm)->field.le_prev; \
    *(elm)->field.le_prev = (elm)->field.le_next; \
    QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

#define LIST_REPLACE(elm, elm2, field) do { \
    if (((elm2)->field.le_next = (elm)->field.le_next) != NULL) \
        (elm2)->field.le_next->field.le_prev = \
            &(elm2)->field.le_next; \
    (elm2)->field.le_prev = (elm)->field.le_prev; \
    *(elm2)->field.le_prev = (elm2); \
    QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

/*
* Simple queue definitions.

```

```

*/
#define SIMPLEQ_HEAD(name, type)  \
struct name {  \
    struct  \
    type *sqh_first; /* first element */  \
    struct type **sqh_last; /* addr of last next element */  \
}

#define SIMPLEQ_HEAD_INITIALIZER(head)  \
{ NULL, &(head).sqh_first }

#define SIMPLEQ_ENTRY(type)  \
struct {  \
    struct type *sqe_next; /* next element */  \
}

/*
 * Simple queue access methods.
 */
#define SIMPLEQ_FIRST(head) ((head)->sqh_first)
#define SIMPLEQ_END(head) NULL
#define SIMPLEQ_EMPTY(head) ((head)->sqh_first == SIMPLEQ_END(head))
#define SIMPLEQ_NEXT(elm, field) ((elm)->field.sqe_next)

#define SIMPLEQ_FOREACH(var, head, field)  \
for ((var) = ((head)->sqh_first);  \
     (var) != SIMPLEQ_END(head);  \
     (var) = ((var)->field.sqe_next))

#define SIMPLEQ_FOREACH_SAFE(var, head, field, next)  \
for ((var) = ((head)->sqh_first);  \
     (var) != SIMPLEQ_END(head) &&  \
     ((next = ((var)->field.sqe_next)), 1);  \
     (var) = (next))

/*
 * Simple queue functions.
 */
#define SIMPLEQ_INIT(head) do {  \
    (head)->sqh_first = NULL;  \
    (head)->sqh_last  \
    = &(head)->sqh_first;  \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_HEAD(head, elm, field) do {  \
    if (((elm)->field.sqe_next = (head)->sqh_first) == NULL)  \
        (head)->sqh_last = &(elm)->field.sqe_next;  \
    (head)->sqh_first = (elm);  \
}

```

```

} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_TAIL(head, elm, field) do { \
    (elm)->field.sqe_next = NULL; \
    *(head)->sqh_last = (elm); \
    (head)->sqh_last = &(elm)->field.sqe_next; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_AFTER(head, listelm, elm, field) do { \
    if (((elm)->field.sqe_next = (listelm)->field.sqe_next) == NULL) \
        (head)->sqh_last = &(elm)->field.sqe_next; \
    (listelm)->field.sqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_HEAD(head, field) do { \
    if (((head)->sqh_first = (head)->sqh_first->field.sqe_next) == NULL) \
        (head)->sqh_last = &(head)->sqh_first; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_AFTER(head, elm, field) do { \
    if \
        (((elm)->field.sqe_next = (elm)->field.sqe_next->field.sqe_next) \
         == NULL) \
        (head)->sqh_last = &(elm)->field.sqe_next; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE(head, elm, type, field) do { \
    if ((head)->sqh_first == (elm)) { \
        SIMPLEQ_REMOVE_HEAD((head), field); \
    } else { \
        struct type *curelm = (head)->sqh_first; \
        while (curelm->field.sqe_next != (elm)) \
            curelm = curelm->field.sqe_next; \
        if ((curelm->field.sqe_next = \
            curelm->field.sqe_next->field.sqe_next) == NULL) \
            (head)->sqh_last = &(curelm)->field.sqe_next; \
    } \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_CONCAT(head1, head2) do { \
    if (!SIMPLEQ_EMPTY((head2))) { \
        *(head1)->sqh_last = (head2)->sqh_first; \
        (head1)->sqh_last = (head2)->sqh_last; \
        SIMPLEQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_LAST(head, type, field) \

```

```

(SIMPLEQ_EMPTY((head)) ? \
NULL : \
    ((struct type
*)(void *) \
    ((char *)((head)->sqh_last) - offsetof(struct type, field))))

/*
 * Tail queue definitions.
 */
#define _TAILQ_HEAD(name, type, qual) \
struct name { \
    qual type *tqh_first; /* first element */ \
    qual type *qual *tqh_last; /* addr of last next element */ \
}
#define TAILQ_HEAD(name, type) _TAILQ_HEAD(name, struct type,)

#define TAILQ_HEAD_INITIALIZER(head) \
{ TAILQ_END(head), &(head).tqh_first }

#define _TAILQ_ENTRY(type, qual) \
struct { \
    qual type *tqe_next; /* next element */ \
    qual type *qual *tqe_prev; /* address of previous next element */ \
}
#define TAILQ_ENTRY(type) _TAILQ_ENTRY(struct type,)

/*
 * Tail queue access methods.
 */
#define TAILQ_FIRST(head) ((head)->tqh_first)
#define TAILQ_END(head) (NULL)
#define TAILQ_NEXT(elm, field) ((elm)->field.tqe_next)
#define TAILQ_LAST(head, headname) \
    (((struct headname *) (void *)((head)->tqh_last))->tqh_last)
#define TAILQ_PREV(elm, headname, field) \
    (((struct headname *) (void *)((elm)->field.tqe_prev))->tqh_last)
#define TAILQ_EMPTY(head) (TAILQ_FIRST(head) == TAILQ_END(head))

#define TAILQ_FOREACH(var, head, field) \
for ((var) = ((head)->tqh_first); \
    (var) != TAILQ_END(head); \
    (var) = ((var)->field.tqe_next))

#define TAILQ_FOREACH_SAFE(var, head, field, next) \
for ((var) = ((head)->tqh_first); \
    (var) != TAILQ_END(head) && \
    next = ((var)->field.tqe_next); \
    (var) = next)

```



```

((next) = TAILQ_NEXT(var, field), 1); (var) = (next))

#define TAILQ_FOREACH_REVERSE(var, head, headname, field) \
for ((var) = TAILQ_LAST((head), headname); \
    (var) != TAILQ_END(head); \
    (var) = TAILQ_PREV((var), headname, field))

#define TAILQ_FOREACH_REVERSE_SAFE(var, head, headname, field, prev) \
for ((var) = TAILQ_LAST((head), headname); \
    (var) != TAILQ_END(head) && \
    ((prev) = TAILQ_PREV((var), headname, field), 1); (var) = (prev))

/*
 * Tail queue functions.
 */
#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, \
    elm, field) \
    if ((head)->tqh_first && \
        (head)->tqh_first->field.tqe_prev != &(head)->tqh_first) \
        QUEUEDEBUG_ABORT("TAILQ_INSERT_HEAD %p %s:%d", (head), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field) \
    if (*(head)->tqh_last != NULL) \
        QUEUEDEBUG_ABORT("TAILQ_INSERT_TAIL %p %s:%d", (head), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_OP(elm, field) \
    if ((elm)->field.tqe_next && \
        (elm)->field.tqe_next->field.tqe_prev != \
        &(elm)->field.tqe_next) \
        QUEUEDEBUG_ABORT("TAILQ_* forw %p %s:%d", (elm), \
            __FILE__, __LINE__); \
    if (*(elm)->field.tqe_prev != (elm)) \
        QUEUEDEBUG_ABORT("TAILQ_* back %p %s:%d", (elm), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field) \
    if ((elm)->field.tqe_next == NULL && \
        (head)->tqh_last != &(elm)->field.tqe_next) \
        QUEUEDEBUG_ABORT("TAILQ_PREREMOVE head %p elm %p %s:%d", \
            (head), (elm), __FILE__, \
            __LINE__);
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field) \
    (elm)->field.tqe_next = (void *)1L; \
    (elm)->field.tqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field)
#define QUEUEDEBUG_TAILQ_OP(elm, field)

```

```

#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field)
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field)
#endif

#define TAILQ_INIT(head) do { \
    (head)->tqh_first = TAILQ_END(head); \
    (head)->tqh_last = &(head)->tqh_first; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBUG_TAILQ_INSERT_HEAD((head), (elm), field) \
    if (((elm)->field.tqe_next = (head)->tqh_first) != TAILQ_END(head)) \
    (head)->tqh_first->field.tqe_prev = \
        &(elm)->field.tqe_next; \
    else \
    (head)->tqh_last = &(elm)->field.tqe_next; \
    (head)->tqh_first = (elm); \
    (elm)->field.tqe_prev = &(head)->tqh_first; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_TAIL(head, \
    elm, field) do { \
    QUEUEDEBUG_TAILQ_INSERT_TAIL((head), (elm), field) \
    (elm)->field.tqe_next = TAILQ_END(head); \
    (elm)->field.tqe_prev = (head)->tqh_last; \
    *(head)->tqh_last = (elm); \
    (head)->tqh_last = &(elm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_AFTER(head, listelm, elm, field) do { \
    QUEUEDEBUG_TAILQ_OP((listelm), field) \
    if (((elm)->field.tqe_next = (listelm)->field.tqe_next) != \
        TAILQ_END(head)) \
    (elm)->field.tqe_next->field.tqe_prev = \
        &(elm)->field.tqe_next; \
    else \
    (head)->tqh_last = &(elm)->field.tqe_next; \
    (listelm)->field.tqe_next = (elm); \
    (elm)->field.tqe_prev = &(listelm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_BEFORE(listelm, elm, field) do { \
    QUEUEDEBUG_TAILQ_OP((listelm), field) \
    (elm)->field.tqe_prev = (listelm)->field.tqe_prev; \
    (elm)->field.tqe_next = (listelm); \
    *(listelm)->field.tqe_prev = (elm); \
    (listelm)->field.tqe_prev

```

```

= &(elm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_REMOVE(head, elm, field) do { \
    QUEUEDEBDEBUG_TAILQ_PREREMOVE((head), (elm), field) \
    QUEUEDEBDEBUG_TAILQ_OP((elm), field) \
    if (((elm)->field.tqe_next != TAILQ_END(head)) \
        (elm)->field.tqe_next->field.tqe_prev = \
            (elm)->field.tqe_prev; \
    else \
        (head)->tqh_last = (elm)->field.tqe_prev; \
        *(elm)->field.tqe_prev = (elm)->field.tqe_next; \
        QUEUEDEBDEBUG_TAILQ_POSTREMOVE((elm), field); \
    } while (/*CONSTCOND*/0)

#define TAILQ_REPLACE(head, elm, elm2, field) do { \
    if (((elm2)->field.tqe_next = (elm)->field.tqe_next) != \
        TAILQ_END(head)) \
        (elm2)->field.tqe_next->field.tqe_prev = \
            &(elm2)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm2)->field.tqe_next; \
        (elm2)->field.tqe_prev = (elm)->field.tqe_prev; \
        *(elm2)->field.tqe_prev = (elm2); \
        QUEUEDEBDEBUG_TAILQ_POSTREMOVE((elm), \
            field); \
    } while (/*CONSTCOND*/0)

#define TAILQ_CONCAT(head1, head2, field) do { \
    if (!TAILQ_EMPTY(head2)) { \
        *(head1)->tqh_last = (head2)->tqh_first; \
        (head2)->tqh_first->field.tqe_prev = (head1)->tqh_last; \
        (head1)->tqh_last = (head2)->tqh_last; \
        TAILQ_INIT((head2)); \
    } \
    } while (/*CONSTCOND*/0)

/*
 * Singly-linked Tail queue declarations.
 */
#define STAILQ_HEAD(name, type) \
struct name { \
    struct type *stqh_first; /* first element */ \
    struct type **stqh_last; /* addr of last next element */ \
}

#define STAILQ_HEAD_INITIALIZER(head) \
{ NULL, &(head).stqh_first }

```

```

#define STAILQ_ENTRY(type) \
struct { \
    struct type *stqe_next; /* next element */ \
}

/*
 * Singly-linked Tail queue access methods.
 */
#define STAILQ_FIRST(head) ((head)->stqh_first)
#define STAILQ_END(head) NULL
#define STAILQ_NEXT(elm, field) ((elm)->field.stqe_next)
#define STAILQ_EMPTY(head) (STAILQ_FIRST(head) \
    == STAILQ_END(head))

/*
 * Singly-linked Tail queue functions.
 */
#define STAILQ_INIT(head) do { \
    (head)->stqh_first = NULL; \
    (head)->stqh_last = &(head)->stqh_first; \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_HEAD(head, elm, field) do { \
    if (((elm)->field.stqe_next = (head)->stqh_first) == NULL) \
        (head)->stqh_last = &(elm)->field.stqe_next; \
    (head)->stqh_first = (elm); \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_TAIL(head, elm, field) do { \
    (elm)->field.stqe_next = NULL; \
    *(head)->stqh_last = (elm); \
    (head)->stqh_last = &(elm)->field.stqe_next; \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_AFTER(head, listelm, elm, field) do { \
    if (((elm)->field.stqe_next = (listelm)->field.stqe_next) == NULL) \
        (head)->stqh_last = &(elm)->field.stqe_next; \
    (listelm)->field.stqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define STAILQ_REMOVE_HEAD(head, field) do { \
    if (((head)->stqh_first = (head)->stqh_first->field.stqe_next) \
        == NULL) \
        (head)->stqh_last = &(head)->stqh_first; \
} while (/*CONSTCOND*/0)

#define STAILQ_REMOVE(head, elm, type, field) do { \

```

```

if ((head)->stqh_first == (elm)) { \
    STAILQ_REMOVE_HEAD((head), field); \
} else { \
    struct type *curelm = (head)->stqh_first; \
    while (curelm->field.stqe_next != (elm)) \
        curelm = curelm->field.stqe_next; \
    if ((curelm->field.stqe_next = \
        curelm->field.stqe_next->field.stqe_next) == NULL) \
        (head)->stqh_last = &(curelm->field.stqe_next); \
} \
} while (/*CONSTCOND*/0)

#define STAILQ_FOREACH(var, head, field) \
    for ((var) = ((head)->stqh_first); \
        (var); \
        (var) = ((var)->field.stqe_next))

#define STAILQ_FOREACH_SAFE(var, head, field, tvar) \
    for ((var) = STAILQ_FIRST((head)); \
        (var) && ((tvar) = STAILQ_NEXT((var), field), 1); \
        (var) = (tvar))

#define STAILQ_CONCAT(head1, head2) do { \
    if (!STAILQ_EMPTY((head2))) { \
        *(head1)->stqh_last \
        = (head2)->stqh_first; \
        (head1)->stqh_last = (head2)->stqh_last; \
        STAILQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)

#define STAILQ_LAST(head, type, field) \
    (STAILQ_EMPTY((head)) ? \
    NULL : \
    ((struct type *) (void *) \
    ((char *) ((head)->stqh_last) - offsetof(struct type, field))))

#ifdef _KERNEL
/*
 * Circular queue definitions. Do not use. We still keep the macros
 * for compatibility but because of pointer aliasing issues their use
 * is discouraged!
 */

/*
 * __launder_type(): We use this ugly hack to work around the the compiler
 * noticing that two types may not alias each other and elide tests in code.

```

* We hit this in the CIRCLEQ macros when comparing 'struct name *' and
 * 'struct type *' (see CIRCLEQ_HEAD()). Modern compilers (such as GCC
 * 4.8) declare these comparisons as always false, causing the code to
 * not run as designed.
 *
 * This hack is only to be used for comparisons and thus can be fully const.
 * Do not use for assignment.
 *
 * If we ever choose to change the ABI of the CIRCLEQ macros, we could fix
 * this by changing the head/tail sentinel values, but see the note above
 * this one.
 */

```
static __inline const void * __launder_type(const void *);
static __inline const void *
__launder_type(const void *__x)
{
    __asm __volatile("" : "+r" (__x));
    return __x;
}

#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field) \
    if ((head)->cqh_first != CIRCLEQ_ENDC(head) && \
        (head)->cqh_first->field.cqe_prev != CIRCLEQ_ENDC(head)) \
        QUEUEDEBUG_ABORT("CIRCLEQ head forw %p %s:%d", (head), \
            __FILE__, __LINE__); \
    if ((head)->cqh_last != CIRCLEQ_ENDC(head) && \
        (head)->cqh_last->field.cqe_next != CIRCLEQ_ENDC(head)) \
        QUEUEDEBUG_ABORT("CIRCLEQ head back %p %s:%d", (head), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_CIRCLEQ_ELM(head, elm, field) \
    if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) { \
        if ((head)->cqh_last != (elm)) \
            QUEUEDEBUG_ABORT("CIRCLEQ \
elm last %p %s:%d", \
                (elm), __FILE__, __LINE__); \
    } else { \
        if ((elm)->field.cqe_next->field.cqe_prev != (elm)) \
            QUEUEDEBUG_ABORT("CIRCLEQ elm forw %p %s:%d", \
                (elm), __FILE__, __LINE__); \
    } \
    if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) { \
        if ((head)->cqh_first != (elm)) \
            QUEUEDEBUG_ABORT("CIRCLEQ elm first %p %s:%d", \
                (elm), __FILE__, __LINE__); \
    } else { \
        if ((elm)->field.cqe_prev->field.cqe_next != (elm)) \
            QUEUEDEBUG_ABORT("CIRCLEQ elm prev %p %s:%d", \
```

```

        (elm), __FILE__, __LINE__); \
    }
#define QUEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field) \
    (elm)->field.cqe_next = (void *)1L; \
    (elm)->field.cqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field)
#define QUEUEDEBUG_CIRCLEQ_ELM(head, elm, field)
#define QUEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field)
#endif

#define CIRCLEQ_HEAD(name, type) \
struct name { \
    struct type *cqh_first; /* \
    first element */ \
    struct type *cqh_last; /* last element */ \
}

#define CIRCLEQ_HEAD_INITIALIZER(head) \
    { CIRCLEQ_END(&head), CIRCLEQ_END(&head) }

#define CIRCLEQ_ENTRY(type) \
struct { \
    struct type *cqe_next; /* next element */ \
    struct type *cqe_prev; /* previous element */ \
}

/*
 * Circular queue functions.
 */
#define CIRCLEQ_INIT(head) do { \
    (head)->cqh_first = CIRCLEQ_END(head); \
    (head)->cqh_last = CIRCLEQ_END(head); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_AFTER(head, listelm, elm, field) do { \
    QUEUEDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBUG_CIRCLEQ_ELM((head), (listelm), field) \
    (elm)->field.cqe_next = (listelm)->field.cqe_next; \
    (elm)->field.cqe_prev = (listelm); \
    if ((listelm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm); \
    else \
        (listelm)->field.cqe_next->field.cqe_prev = (elm); \
    (listelm)->field.cqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_BEFORE(head,

```

```

listelm, elm, field) do { \
    QUEUEDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBUG_CIRCLEQ_ELM((head), (listelm), field) \
    (elm)->field.cqe_next = (listelm); \
    (elm)->field.cqe_prev = (listelm)->field.cqe_prev; \
    if ((listelm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm); \
    else \
        (listelm)->field.cqe_prev->field.cqe_next = (elm); \
    (listelm)->field.cqe_prev = (elm); \
} while (/*CONSTCOND*/0)

```

```

#define CIRCLEQ_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBUG_CIRCLEQ_HEAD((head), field) \
    (elm)->field.cqe_next = (head)->cqh_first; \
    (elm)->field.cqe_prev = CIRCLEQ_END(head); \
    if ((head)->cqh_last == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm); \
    else \
        (head)->cqh_first->field.cqe_prev = (elm); \
    (head)->cqh_first = (elm); \
} while (/*CONSTCOND*/0)

```

```

#define CIRCLEQ_INSERT_TAIL(head, elm, field) do { \
    QUEUEDEBUG_CIRCLEQ_HEAD((head), field) \
    (elm)->field.cqe_next \
    = CIRCLEQ_END(head); \
    (elm)->field.cqe_prev = (head)->cqh_last; \
    if ((head)->cqh_first == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm); \
    else \
        (head)->cqh_last->field.cqe_next = (elm); \
    (head)->cqh_last = (elm); \
} while (/*CONSTCOND*/0)

```

```

#define CIRCLEQ_REMOVE(head, elm, field) do { \
    QUEUEDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBUG_CIRCLEQ_ELM((head), (elm), field) \
    if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
        (head)->cqh_last = (elm)->field.cqe_prev; \
    else \
        (elm)->field.cqe_next->field.cqe_prev = \
            (elm)->field.cqe_prev; \
    if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
        (head)->cqh_first = (elm)->field.cqe_next; \
    else \
        (elm)->field.cqe_prev->field.cqe_next = \
            (elm)->field.cqe_next; \
} while (/*CONSTCOND*/0)

```



```

QUEUEDEBUG_CIRCLEQ_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_FOREACH(var, head, field) \
for ((var) = ((head)->cqh_first); \
     (var) != CIRCLEQ_ENDC(head); \
     (var) \
     = ((var)->field.cqe_next))

#define CIRCLEQ_FOREACH_REVERSE(var, head, field) \
for ((var) = ((head)->cqh_last); \
     (var) != CIRCLEQ_ENDC(head); \
     (var) = ((var)->field.cqe_prev))

/*
 * Circular queue access methods.
 */

#define CIRCLEQ_FIRST(head) ((head)->cqh_first)
#define CIRCLEQ_LAST(head) ((head)->cqh_last)
/* For comparisons */
#define CIRCLEQ_ENDC(head) (__launter_type(head))
/* For assignments */
#define CIRCLEQ_END(head) ((void *)(head))
#define CIRCLEQ_NEXT(elm, field) ((elm)->field.cqe_next)
#define CIRCLEQ_PREV(elm, field) ((elm)->field.cqe_prev)
#define CIRCLEQ_EMPTY(head) \
    (CIRCLEQ_FIRST(head) == CIRCLEQ_ENDC(head))

#define CIRCLEQ_LOOP_NEXT(head, elm, field) \
    (((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
     ? ((head)->cqh_first) \
     : (elm->field.cqe_next))
#define CIRCLEQ_LOOP_PREV(head, elm, field) \
    (((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
     ? ((head)->cqh_last) \
     : \
     (elm->field.cqe_prev))
#endif /* !_KERNEL */

#endif /* !_SYS_QUEUE_H_ */

# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite

```

```

url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver

replaces="py-flake8-copyright"          # Backwards compatibility
provides="py-flake8-copyright=$pkgver-r$pkgrel" # Backwards compatibility

build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}

package() {
    gpep517 install-wheel --destdir "$pkgdir" \
        .dist/*.whl
}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
    flake8-copyright-0.2.4.tar.gz
"

#ifdef _SYS_CDEFS_H_
#define _SYS_CDEFS_H_

#warning usage of non-standard #include <sys/cdefs.h> is deprecated

#undef __P
#undef __PMT

#define __P(args) args
#define __PMT(args) args

#define __CONCAT(x,y) x ## y
#define __STRING(x) #x

#ifdef __cplusplus
# define __BEGIN_DECLS extern "C" {
# define __END_DECLS }
#else
# define __BEGIN_DECLS
# define __END_DECLS

```

```

#endif

#if defined(__GNUC__) && !defined(__cplusplus)
# define __THROW __attribute__ ((__nothrow__))
# define __NTH(fct) __attribute__ ((__nothrow__)) fct
#else
# define __THROW
# define __NTH(fct) fct
#endif

#endif /* _SYS_CDEFS_H_ */
# Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"
url="https://github.com/onur/cargo-license"
arch="all"
license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

prepare() {
    default_prepare

    cargo fetch --target="$CTARGET" --locked
}

build() {
    cargo auditable build --release --frozen
}

check() {
    cargo test --frozen
}

package() {
    install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5 cargo-license-0.6.1.tar.gz
"

/* $NetBSD: tree.h,v 1.20 2013/09/14 13:20:45 joerg Exp $ */
/* $OpenBSD: tree.h,v 1.13 2011/07/09 00:19:45 pirofti Exp $ */
/* Modified by Void Linux. */
/*

```

* Copyright 2002 Niels Provos <provos@citi.umich.edu>
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE
 * AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

```
#ifndef _SYS_TREE_H_
#define _SYS_TREE_H_

#ifdef __GNUC__
#define __GNUC_PREREQ__(x, y) \
  ((__GNUC__ == (x) && __GNUC_MINOR__ >= (y)) || \
   (__GNUC__ > (x)))
#else
#define __GNUC_PREREQ__(x, y) 0
#endif
```

```
#if __GNUC_PREREQ__(2, 7) || defined(__lint__)
#define _sys_tree_h_unused __attribute__((__unused__))
#else
#define _sys_tree_h_unused /* delete */
#endif
```

```
/*
 * This file defines data structures for different types of trees:
 * splay trees and red-black trees.
 *
 * A splay tree is a self-organizing
```

data structure. Every operation

- * on the tree causes a splay to happen. The splay moves the requested
- * node to the root of the tree and partly rebalances it.

*

- * This has the benefit that request locality causes faster lookups as
- * the requested nodes move to the top of the tree. On the other hand,
- * every lookup causes memory writes.

*

- * The Balance Theorem bounds the total access time for m operations
- * and n inserts on an initially empty tree as $O((m + n)\lg n)$. The
- * amortized cost for a sequence of m accesses to a splay tree is $O(\lg n)$;

*

- * A red-black tree is a binary search tree with the node color as an
- * extra attribute. It fulfills a set of conditions:

- * - every search path from the root to a leaf consists of the
- * same number of black nodes,
- * - each red node (except for the root) has a black parent,
- * - each leaf node is black.

*

- * Every operation on a red-black tree is bounded as $O(\lg n)$.

- * The maximum height of a red-black tree is $2\lg(n+1)$.

*/

```
#define SPLAY_HEAD(name, type) \
struct name { \
    struct type *sph_root; /* root of the tree */ \
}
```

```
#define SPLAY_INITIALIZER(root) \
{ NULL }
```

```
#define SPLAY_INIT(root) do { \
    (root)->sph_root = NULL; \
} while (/*CONSTCOND*/ 0)
```

```
#define SPLAY_ENTRY(type) \
struct { \
    struct type *spe_left; /* left element */ \
    struct type *spe_right; /* right element */ \
}
```

```
#define SPLAY_LEFT(elm, field) (elm)->field.spe_left
#define SPLAY_RIGHT(elm, field) (elm)->field.spe_right
#define SPLAY_ROOT(head) (head)->sph_root
#define SPLAY_EMPTY(head) (SPLAY_ROOT(head) == NULL)
```

```
/* SPLAY_ROTATE_{LEFT,RIGHT} expect that tmp hold SPLAY_{RIGHT,LEFT} */
```

```
#define SPLAY_ROTATE_RIGHT(head, tmp, field) do { \
```

```

    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(tmp, field); \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ROTATE_LEFT(head, tmp, field) do { \
    SPLAY_RIGHT((head)->sph_root, \
    field) = SPLAY_LEFT(tmp, field); \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKLEFT(head, tmp, field) do { \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \
    (head)->sph_root = SPLAY_LEFT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKRIGHT(head, tmp, field) do { \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \
    (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ASSEMBLE(head, node, left, right, field) do { \
    SPLAY_RIGHT(left, field) = SPLAY_LEFT((head)->sph_root, field); \
    SPLAY_LEFT(right, field) = SPLAY_RIGHT((head)->sph_root, field); \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(node, field); \
    SPLAY_RIGHT((head)->sph_root, field) = SPLAY_LEFT(node, field); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */

#define
    SPLAY_PROTOTYPE(name, type, field, cmp) \
    void name##_SPLAY(struct name *, struct type *); \
    void name##_SPLAY_MINMAX(struct name *, int); \
    struct type *name##_SPLAY_INSERT(struct name *, struct type *); \
    struct type *name##_SPLAY_REMOVE(struct name *, struct type *); \
    \
/* Finds the node with the same key as elm */ \
static __inline struct type * \
name##_SPLAY_FIND(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) \
        return(NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) \

```

```

    return (head->sph_root); \
return (NULL); \
} \
\
static __inline __sys_tree_h_unused struct type * \
name##_SPLAY_NEXT(struct name *head, struct type *elm) \
{ \
    name##_SPLAY(head, elm); \
    if (SPLAY_RIGHT(elm, field) != NULL) { \
        elm = SPLAY_RIGHT(elm, field); \
        while (SPLAY_LEFT(elm, field) != NULL) { \
            elm = SPLAY_LEFT(elm, field); \
        } \
    } \
    else \
        elm = NULL; \
    return (elm); \
} \
\
static __sys_tree_h_unused __inline struct type * \
name##_SPLAY_MIN_MAX(struct name *head, int val) \
{ \
    name##_SPLAY_MINMAX(head, val); \
    return (SPLAY_ROOT(head)); \
}

/* Main splay operation.
 * Moves node close to the key of elm to top
 */
#define SPLAY_GENERATE(name, type, field, cmp) \
struct type * \
name##_SPLAY_INSERT(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) { \
        SPLAY_LEFT(elm, field) = SPLAY_RIGHT(elm, field) = NULL; \
    } else { \
        int __comp; \
        name##_SPLAY(head, elm); \
        __comp = (cmp)(elm, (head)->sph_root); \
        if (__comp < 0) { \
            SPLAY_LEFT(elm, field) = SPLAY_LEFT((head)->sph_root, field); \
            SPLAY_RIGHT(elm, field) = (head)->sph_root; \
            SPLAY_LEFT((head)->sph_root, field) = NULL; \
        } else if (__comp > 0) { \
            SPLAY_RIGHT(elm, \
field) = SPLAY_RIGHT((head)->sph_root, field); \
            SPLAY_LEFT(elm, field) = (head)->sph_root; \
            SPLAY_RIGHT((head)->sph_root, field) = NULL; \
        } \
    } \
}

```

```

    } else \
        return ((head)->sph_root); \
    } \
    (head)->sph_root = (elm); \
    return (NULL); \
} \
\
struct type * \
name##_SPLAY_REMOVE(struct name *head, struct type *elm) \
{ \
    struct type *__tmp; \
    if (SPLAY_EMPTY(head)) \
        return (NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) { \
        if (SPLAY_LEFT((head)->sph_root, field) == NULL) { \
            (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field);\
        } else { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field);\
            (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);\
            name##_SPLAY(head, elm); \
            SPLAY_RIGHT((head)->sph_root, field) = __tmp; \
        } \
        return (elm); \
    } \
    return (NULL); \
} \
\
void \
name##_SPLAY(struct \
    name *head, struct type *elm) \
{ \
    struct type __node, *__left, *__right, *__tmp; \
    int __comp; \
    \
    SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
    __left = __right = &__node; \
    \
    while ((__comp = (cmp)(elm, (head)->sph_root)) != 0) { \
        if (__comp < 0) { \
            __tmp = SPLAY_LEFT((head)->sph_root, field);\
            if (__tmp == NULL) \
                break; \
            if ((cmp)(elm, __tmp) < 0){ \
                SPLAY_ROTATE_RIGHT(head, __tmp, field);\
                if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
                    break; \
            } \

```



```

    SPLAY_LINKLEFT(head, __right, field); \
} else if (__comp > 0) { \
    __tmp = SPLAY_RIGHT((head)->sph_root, field); \
    if (__tmp == NULL) \
        break; \
    if ((cmp)(elm, __tmp) > 0){ \
        SPLAY_ROTATE_LEFT(head, __tmp, field); \
        if (SPLAY_RIGHT((head)->sph_root, field) == NULL)\
            break; \
    } \
    SPLAY_LINKRIGHT(head, __left, field); \
} \
} \
SPLAY_ASSEMBLE(head,
&__node, __left, __right, field); \
} \
\
/* Splay with either the minimum or the maximum element \
* Used to find minimum or maximum element in tree. \
*/ \
void name##_SPLAY_MINMAX(struct name *head, int __comp) \
{ \
    struct type __node, *__left, *__right, *__tmp; \
    \
    SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
    __left = __right = &__node; \
    \
    while (1) { \
        if (__comp < 0) { \
            __tmp = SPLAY_LEFT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if (__comp < 0){ \
                SPLAY_ROTATE_RIGHT(head, __tmp, field); \
                if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
                    break; \
            } \
            SPLAY_LINKLEFT(head, __right, field); \
        } else if (__comp > 0) { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if (__comp > 0) { \
                SPLAY_ROTATE_LEFT(head, __tmp, field); \
                if (SPLAY_RIGHT((head)->sph_root,
field) == NULL)\
                    break; \
            } \

```

```

    SPLAY_LINKRIGHT(head, __left, field); \
} \
} \
SPLAY_ASSEMBLE(head, &__node, __left, __right, field); \
}

#define SPLAY_NEGINF -1
#define SPLAY_INF 1

#define SPLAY_INSERT(name, x, y) name##_SPLAY_INSERT(x, y)
#define SPLAY_REMOVE(name, x, y) name##_SPLAY_REMOVE(x, y)
#define SPLAY_FIND(name, x, y) name##_SPLAY_FIND(x, y)
#define SPLAY_NEXT(name, x, y) name##_SPLAY_NEXT(x, y)
#define SPLAY_MIN(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_NEGINF))
#define SPLAY_MAX(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_INF))

#define SPLAY_FOREACH(x, name, head) \
for ((x) = SPLAY_MIN(name, head); \
    (x) != NULL; \
    (x) = SPLAY_NEXT(name, head, x))

/* Macros that define a red-black tree */
#define RB_HEAD(name, type) \
struct name { \
    struct type *rbh_root; /* root of the tree */ \
}

#define RB_INITIALIZER(root) \
{ \
    NULL }

#define RB_INIT(root) do { \
    (root)->rbh_root = NULL; \
} while (/*CONSTCOND*/ 0)

#define RB_BLACK 0
#define RB_RED 1
#define RB_ENTRY(type) \
struct { \
    struct type *rbe_left; /* left element */ \
    struct type *rbe_right; /* right element */ \
    struct type *rbe_parent; /* parent element */ \
    int rbe_color; /* node color */ \
}

#define RB_LEFT(elm, field) (elm)->field.rbe_left

```

```

#define RB_RIGHT(elm, field) (elm)->field.rbe_right
#define RB_PARENT(elm, field) (elm)->field.rbe_parent
#define RB_COLOR(elm, field) (elm)->field.rbe_color
#define RB_ROOT(head) (head)->rbh_root
#define RB_EMPTY(head) (RB_ROOT(head) == NULL)

#define RB_SET(elm, parent, field) do { \
    RB_PARENT(elm, field) = parent; \
    RB_LEFT(elm, field) = RB_RIGHT(elm, field) = NULL; \
    RB_COLOR(elm, field) = RB_RED; \
} while (/*CONSTCOND*/ 0)

#define RB_SET_BLACKRED(black, red, field) do { \
    RB_COLOR(black, field) = RB_BLACK; \
    RB_COLOR(red, field) \
    = RB_RED; \
} while (/*CONSTCOND*/ 0)

#ifndef RB_AUGMENT
#define RB_AUGMENT(x) do {} while (/*CONSTCOND*/ 0)
#endif

#define RB_ROTATE_LEFT(head, elm, tmp, field) do { \
    (tmp) = RB_RIGHT(elm, field); \
    if ((RB_RIGHT(elm, field) = RB_LEFT(tmp, field)) != NULL) { \
        RB_PARENT(RB_LEFT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_LEFT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
    RB_AUGMENT(tmp); \
    if ((RB_PARENT(tmp, field))) \
        RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

#define RB_ROTATE_RIGHT(head, elm, tmp, field) do { \
    (tmp) = RB_LEFT(elm, field); \
    if ((RB_LEFT(elm, field) = RB_RIGHT(tmp, field)) \
        != NULL) { \
        RB_PARENT(RB_RIGHT(tmp, field), field) = (elm); \
    } \
}

```

```

RB_AUGMENT(elm); \
if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
    if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
        RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
    else \
        RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
} else \
    (head)->rbh_root = (tmp); \
RB_RIGHT(tmp, field) = (elm); \
RB_PARENT(elm, field) = (tmp); \
RB_AUGMENT(tmp); \
if ((RB_PARENT(tmp, field))) \
    RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */
#define RB_PROTOTYPE(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp,)
#define RB_PROTOTYPE_STATIC(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_PROTOTYPE_INTERNAL(name, type, field, cmp, attr) \
    attr void name##_RB_INSERT_COLOR(struct name *, struct \
        type *); \
    attr void name##_RB_REMOVE_COLOR(struct name *, struct type *, struct type *); \
    attr struct type *name##_RB_REMOVE(struct name *, struct type *); \
    attr struct type *name##_RB_INSERT(struct name *, struct type *); \
    attr struct type *name##_RB_FIND(struct name *, struct type *); \
    attr struct type *name##_RB_NFIND(struct name *, struct type *); \
    attr struct type *name##_RB_NEXT(struct type *); \
    attr struct type *name##_RB_PREV(struct type *); \
    attr struct type *name##_RB_MINMAX(struct name *, int); \
    \

/* Main rb operation.
 * Moves node close to the key of elm to top
 */
#define RB_GENERATE(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp,)
#define RB_GENERATE_STATIC(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_GENERATE_INTERNAL(name, type, field, cmp, attr) \
    attr void \
    name##_RB_INSERT_COLOR(struct name *head, struct type *elm) \
    { \
        struct type \
        *parent, *gparent, *tmp; \
        while ((parent = RB_PARENT(elm, field)) != NULL && \
            RB_COLOR(parent, field) == RB_RED) { \

```

```

gparent = RB_PARENT(parent, field); \
if (parent == RB_LEFT(gparent, field)) { \
    tmp = RB_RIGHT(gparent, field); \
    if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
        RB_COLOR(tmp, field) = RB_BLACK; \
        RB_SET_BLACKRED(parent, gparent, field);\
        elm = gparent; \
        continue; \
    } \
    if (RB_RIGHT(parent, field) == elm) { \
        RB_ROTATE_LEFT(head, parent, tmp, field);\
        tmp = parent; \
        parent = elm; \
        elm = tmp; \
    } \
    RB_SET_BLACKRED(parent, gparent, field); \
    RB_ROTATE_RIGHT(head, gparent, tmp, field); \
} else { \
    tmp = RB_LEFT(gparent, field); \
    if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
        RB_COLOR(tmp, field) = RB_BLACK; \
        RB_SET_BLACKRED(parent, gparent, field);\
        elm = gparent; \
        continue; \
    } \
    if (RB_LEFT(parent, field) == elm) \
{ \
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    tmp = parent; \
    parent = elm; \
    elm = tmp; \
} \
    RB_SET_BLACKRED(parent, gparent, field); \
    RB_ROTATE_LEFT(head, gparent, tmp, field); \
} \
} \
RB_COLOR(head->rbh_root, field) = RB_BLACK; \
} \
\
attr void \
name##_RB_REMOVE_COLOR(struct name *head, struct type *parent, struct type *elm) \
{ \
    struct type *tmp; \
    while ((elm == NULL || RB_COLOR(elm, field) == RB_BLACK) && \
        elm != RB_ROOT(head)) { \
        if (RB_LEFT(parent, field) == elm) { \
            tmp = RB_RIGHT(parent, field); \
            if (RB_COLOR(tmp, field) == RB_RED) { \

```

```

RB_SET_BLACKRED(tmp, parent, field); \
RB_ROTATE_LEFT(head, parent, tmp, field);\
tmp = RB_RIGHT(parent, field); \
} \
if ((RB_LEFT(tmp, field) == NULL || \
    RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
    (RB_RIGHT(tmp, field) == NULL || \
    RB_COLOR(RB_RIGHT(tmp,
field), field) == RB_BLACK)) {\
    RB_COLOR(tmp, field) = RB_RED; \
    elm = parent; \
    parent = RB_PARENT(elm, field); \
} else { \
    if (RB_RIGHT(tmp, field) == NULL || \
        RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK) {\
        struct type *oleft; \
        if ((oleft = RB_LEFT(tmp, field)) \
            != NULL) \
            RB_COLOR(oleft, field) = RB_BLACK;\
        RB_COLOR(tmp, field) = RB_RED; \
        RB_ROTATE_RIGHT(head, tmp, oleft, field);\
        tmp = RB_RIGHT(parent, field); \
    } \
    RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
    RB_COLOR(parent, field) = RB_BLACK; \
    if (RB_RIGHT(tmp, field)) \
        RB_COLOR(RB_RIGHT(tmp, field), field) = RB_BLACK;\
    RB_ROTATE_LEFT(head, parent, tmp, field);\
    elm = RB_ROOT(head); \
    break; \
} \
} else { \
    tmp = RB_LEFT(parent, field); \
    if (RB_COLOR(tmp, field) == RB_RED) { \
        RB_SET_BLACKRED(tmp, parent, field); \
        RB_ROTATE_RIGHT(head, parent, tmp, field);\
        tmp
= RB_LEFT(parent, field); \
    } \
    if ((RB_LEFT(tmp, field) == NULL || \
        RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
        (RB_RIGHT(tmp, field) == NULL || \
        RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
        RB_COLOR(tmp, field) = RB_RED; \
        elm = parent; \
        parent = RB_PARENT(elm, field); \
    } else { \
        if (RB_LEFT(tmp, field) == NULL || \

```

```

    RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) {\
struct type *oright; \
if ((oright = RB_RIGHT(tmp, field)) \
    != NULL) \
    RB_COLOR(oright, field) = RB_BLACK;\
    RB_COLOR(tmp, field) = RB_RED; \
    RB_ROTATE_LEFT(head, tmp, oright, field);\
    tmp = RB_LEFT(parent, field); \
} \
RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
RB_COLOR(parent, field) = RB_BLACK; \
if (RB_LEFT(tmp, field)) \
    RB_COLOR(RB_LEFT(tmp, field), field) = RB_BLACK;\
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    elm = RB_ROOT(head); \
    break; \
} \
} \
} \
if
(elm) \
    RB_COLOR(elm, field) = RB_BLACK; \
} \
\
attr struct type * \
name##_RB_REMOVE(struct name *head, struct type *elm) \
{ \
    struct type *child, *parent, *old = elm; \
    int color; \
    if (RB_LEFT(elm, field) == NULL) \
        child = RB_RIGHT(elm, field); \
    else if (RB_RIGHT(elm, field) == NULL) \
        child = RB_LEFT(elm, field); \
    else { \
        struct type *left; \
        elm = RB_RIGHT(elm, field); \
        while ((left = RB_LEFT(elm, field)) != NULL) \
            elm = left; \
        child = RB_RIGHT(elm, field); \
        parent = RB_PARENT(elm, field); \
        color = RB_COLOR(elm, field); \
        if (child) \
            RB_PARENT(child, field) = parent; \
        if (parent) { \
            if (RB_LEFT(parent, field) == elm) \
                RB_LEFT(parent, field) = child; \
            else \
                RB_RIGHT(parent, field) = child; \
        } \
    } \
}

```

```

    RB_AUGMENT(parent); \
} else \
    RB_ROOT(head)
= child; \
if (RB_PARENT(elm, field) == old) \
    parent = elm; \
(elm)->field = (old)->field; \
if (RB_PARENT(old, field)) { \
    if (RB_LEFT(RB_PARENT(old, field), field) == old)\
        RB_LEFT(RB_PARENT(old, field), field) = elm;\
    else \
        RB_RIGHT(RB_PARENT(old, field), field) = elm;\
    RB_AUGMENT(RB_PARENT(old, field)); \
} else \
    RB_ROOT(head) = elm; \
RB_PARENT(RB_LEFT(old, field), field) = elm; \
if (RB_RIGHT(old, field)) \
    RB_PARENT(RB_RIGHT(old, field), field) = elm; \
if (parent) { \
    left = parent; \
    do { \
        RB_AUGMENT(left); \
    } while ((left = RB_PARENT(left, field)) != NULL); \
} \
goto color; \
} \
parent = RB_PARENT(elm, field); \
color = RB_COLOR(elm, field); \
if (child) \
    RB_PARENT(child, field) = parent; \
if (parent) { \
    if (RB_LEFT(parent, field) == elm) \
        RB_LEFT(parent, field) = child; \
    else \
        RB_RIGHT(parent,
field) = child; \
    RB_AUGMENT(parent); \
} else \
    RB_ROOT(head) = child; \
color: \
if (color == RB_BLACK) \
    name##_RB_REMOVE_COLOR(head, parent, child); \
return (old); \
} \
\
/* Inserts a node into the RB tree */ \
attr struct type * \
name##_RB_INSERT(struct name *head, struct type *elm) \

```



```

{
    \
    struct type *tmp;    \
    struct type *parent = NULL;    \
    int comp = 0;    \
    tmp = RB_ROOT(head);    \
    while (tmp) {    \
        parent = tmp;    \
        comp = (cmp)(elm, parent);    \
        if (comp < 0)    \
            tmp = RB_LEFT(tmp, field);    \
        else if (comp > 0)    \
            tmp = RB_RIGHT(tmp, field);    \
        else    \
            return (tmp);    \
    }    \
    RB_SET(elm, parent, field);    \
    if (parent != NULL) {    \
        if (comp < 0)    \
            RB_LEFT(parent, field) = elm;    \
        else    \
            RB_RIGHT(parent, field) = elm;    \
        RB_AUGMENT(parent);    \
    } else    \
        RB_ROOT(head) = elm;    \
    name##_RB_INSERT_COLOR(head,
    elm);    \
    return (NULL);    \
}    \
    \
/* Finds the node with the same key as elm */    \
attr struct type *    \
name##_RB_FIND(struct name *head, struct type *elm)    \
{
    \
    struct type *tmp = RB_ROOT(head);    \
    int comp;    \
    while (tmp) {    \
        comp = cmp(elm, tmp);    \
        if (comp < 0)    \
            tmp = RB_LEFT(tmp, field);    \
        else if (comp > 0)    \
            tmp = RB_RIGHT(tmp, field);    \
        else    \
            return (tmp);    \
    }    \
    return (NULL);    \
}    \
    \
/* Finds the first node greater than or equal to the search key */    \

```

```

attr struct type * \
name##_RB_NFIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \
    struct type *res = NULL; \
    int comp; \
    while (tmp) { \
        comp = cmp(elm, tmp); \
        if (comp < 0) { \
            res = tmp; \
            tmp = RB_LEFT(tmp, field); \
        } \
        else if (comp > 0) \
            tmp = \
RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    return (res); \
} \
\
/* ARGSUSED */ \
attr struct type * \
name##_RB_NEXT(struct type *elm) \
{ \
    if (RB_RIGHT(elm, field)) { \
        elm = RB_RIGHT(elm, field); \
        while (RB_LEFT(elm, field)) \
            elm = RB_LEFT(elm, field); \
    } else { \
        if (RB_PARENT(elm, field) && \
            (elm == RB_LEFT(RB_PARENT(elm, field), field))) \
            elm = RB_PARENT(elm, field); \
        else { \
            while (RB_PARENT(elm, field) && \
                (elm == RB_RIGHT(RB_PARENT(elm, field), field))) \
                elm = RB_PARENT(elm, field); \
        } \
    } \
    return (elm); \
} \
\
/* ARGSUSED */ \
attr struct type * \
name##_RB_PREV(struct type *elm) \
{ \
    if (RB_LEFT(elm, field)) { \

```

```

elm = RB_LEFT(elm, field); \
while (RB_RIGHT(elm, field)) \
    elm = RB_RIGHT(elm, field); \
}
else { \
    if (RB_PARENT(elm, field) && \
        (elm == RB_RIGHT(RB_PARENT(elm, field), field))) \
        elm = RB_PARENT(elm, field); \
    else { \
        while (RB_PARENT(elm, field) && \
            (elm == RB_LEFT(RB_PARENT(elm, field), field))) \
            elm = RB_PARENT(elm, field); \
        elm = RB_PARENT(elm, field); \
    } \
} \
return (elm); \
} \
\
attr struct type * \
name##_RB_MINMAX(struct name *head, int val) \
{ \
    struct type *tmp = RB_ROOT(head); \
    struct type *parent = NULL; \
    while (tmp) { \
        parent = tmp; \
        if (val < 0) \
            tmp = RB_LEFT(tmp, field); \
        else \
            tmp = RB_RIGHT(tmp, field); \
    } \
    return (parent); \
}

#define RB_NEGINF -1
#define RB_INF 1

#define RB_INSERT(name, x, y) name##_RB_INSERT(x, y)
#define RB_REMOVE(name, x, y) name##_RB_REMOVE(x, y)
#define RB_FIND(name, x, y) name##_RB_FIND(x, y)
#define RB_NFIND(name, x, y) name##_RB_NFIND(x, \
y)
#define RB_NEXT(name, x, y) name##_RB_NEXT(y)
#define RB_PREV(name, x, y) name##_RB_PREV(y)
#define RB_MIN(name, x) name##_RB_MINMAX(x, RB_NEGINF)
#define RB_MAX(name, x) name##_RB_MINMAX(x, RB_INF)

#define RB_FOREACH(x, name, head) \
    for ((x) = RB_MIN(name, head); \

```

```

(x) != NULL; \
(x) = name##_RB_NEXT(x))

#define RB_FOREACH_FROM(x, name, y) \
for ((x) = (y); \
((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_SAFE(x, name, head, y) \
for ((x) = RB_MIN(name, head); \
((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_REVERSE(x, name, head) \
for ((x) = RB_MAX(name, head); \
(x) != NULL; \
(x) = name##_RB_PREV(x))

#define RB_FOREACH_REVERSE_FROM(x, name, y) \
for ((x) = (y); \
((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_REVERSE_SAFE(x, name, \
head, y) \
for ((x) = RB_MAX(name, head); \
((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
(x) = (y))

#endif /* _SYS_TREE_H_ */
pkgname=linux-headers
pkgver=6.14.2 # Follow the latest Linux stable
_kernver=${pkgver%.*}
pkgrel=0
pkgdesc="Linux system headers"
url="https://kernel.org/"
arch="all" # Headers are architecture specific
license="GPL-2.0-only"
makedepends="perl"
options="!check !dbg !strip !tracedeps !archcheck"
source="https://kernel.org/pub/linux/kernel/v6.x/linux-${_kernver}.tar.xz
revert-broken-uapi.patch
0003-remove-inclusion-of-sysinfo.h-in-kernel.h.patch

```

1.25 ca-certificates 20250911-r0

1.25.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/opt/cola/permits/2660733757_1759915527.765318/0/ca-certificates-20250911-r0-tar-gz/usr/share/ca-
certificates/mozilla/HARICA_TLS_RSA_Root_CA_2021.crt-----BEGIN CERTIFICATE-----
/opt/cola/permits/2660733757_1759915527.765318/0/ca-certificates-20250911-r0-tar-gz/usr/share/ca-
certificates/mozilla/QuoVadis_Root_CA_2.crt-----BEGIN CERTIFICATE-----
/opt/cola/permits/2660733757_1759915527.765318/0/ca-certificates-20250911-r0-tar-gz/usr/share/ca-
certificates/mozilla/TWCA_Global_Root_CA.crt-----BEGIN CERTIFICATE-----
/opt/cola/permits/2660733757_1759915527.765318/0/ca-certificates-20250911-r0-tar-gz/usr/share/ca-
certificates/mozilla/TrustAsia_TLS_RSA_Root_CA.crt-----BEGIN CERTIFICATE-----
```

Found in path(s):

*

No license file was found, but licenses were detected in source scan.

```
-----BEGIN CERTIFICATE-----
MIIFWjCCA0KgAwIBAgISEdK7udcjGJ5AXwqdLdDfJWfRMA0GCSqGSIb3DQEBAUA
MEYxChAJBgNVBAYTAkFMRkwFwYDVQQKExBHbG9iYWxTaWduIG52LXNhMRwwGgYD
VQQDExNHbG9iYWxTaWduIFJvb3QgUjQ2MB4XDTE5MDMyMDAwMDAwMFoXDTQ2MDMy
MDAwMDAwMFowRjELMAkGA1UEBhMCQkUxGTAXBgNVBAoTEEdsb2JhbFNpZ24gbnYt
c2ExHDAaBgNVBAMTE0dsb2JhbFNpZ24gUm9vdCBSNDYwgglMA0GCSqGSIb3DQEB
AQUAA4ICDwAwggIKAoICAQCsrHQy6LNI5brtQyYdpokNRbopiLKkHWPd08EsCveJ
OaFV6Wc0dwxu5FUdUiXSE2te4R2pt32JMI8Nnp8semNgQB+msLZ4j5lUghYruQG
vGIFAha/r6gjA7aUD7xubMLL1aa7DOn2wQL7Id5m3RerdELv8HQvJfTqa1VbkNud
316HCkD7rRlr+/fKYIje2sGP1q7Vf9Q8g+7XFkyDRTNrJ9CG0Bwta/OrffGFqfUo
0q3v84RLHif8E6M6cqJaESvWJ3En7YEtBwBkoe0G1h6zD8K+kZPTXhc+Ctl4wSE
y132tGqzZfxCnlEmIyDLPRt5ge1lFgBPGmSXZgjPjHvjK8Cd+RTyG/FWaha/LIWF
zXg4mutCagI0GIMXTpRW+LaCtfOW3T3zvn8gdz57GSNrLNRyc0NXfeD412IPFzYE
+cCQYDdF3uYM2HSNrpYibXRdQr4G9dlkbGfIQRImwTDsHTUB+JMWKmlJ5jqSngiCN
I/onccnfxkF0eE32kRbcRoxfKWMxWxEM2G/CtjJ9++ZdU6Z+Ffy7dXxd7Pj2Fxs
x2sZy/N78CsHpdlseVR2bJ0cpm4O6XkMqCNqo98bMDGfsVR7/mrLZqrcZdCinkqa
ByFrgY/bxFn63iLABJzjqls2k+g9vXqhnQt2sQvHnf3PmKgGwvqgo6GDoLclcqUC
4wIDAQABo0IwQDAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNV
HQ4EFgQUA1yrc4GHqMywptWU4jaWSf8FmSwwDQYJKoZIhvcNAQEMBQADggIBAHx4
7PYCLltbfpIrXTncvtgdokIzTfnvpCo7RGkerNlFo048p9gkUbJUHJNOxO97k4Vg
JuoJSOD1u8fpaNK7ajFzHmuEajwfmf3IH7wvqMxX63bElAZHU1VNaL8FpO7XJqti
2kM3S+LGteWygxk6x9PbTZ4IevPuzz5i+6zoYMzRx6Fcg0XERczF2sUyQQCPtlk
pnnpHs6i58FZFZ8d4kuaPp92CC1r2LpXFNqD6v6MVenQTqnMdzGxRBF6XLE+0xRF
FRhiJBPSy03OXIPBNvIQ6IbbjhVp+J3pZmOUdkLG5NrmJ7v2B0GbhWrJKsFjLt
rWhV/pi60zTe9Mlhww6G9kuEYO4Ne7UyWHmRVSyBQ7N0H3qqJZ4d16GLuc1CLgSk
ZoNNiTW2bKg2SnkheCLQQRzRQDGQob4Ez8pn7fXwgNNgyYMqIgXQBztSvwyeqiv5
u+YfjyW6hY0XHgI+XVAEV8/+LbZvXMAaq7afJMbfc2hIkCwU9D9SGuTSyXTDYWnP
4vkYxboznxSjBF25cfe1lNj2M8FawTSLfJvdkzmE6JwYZ+vj+vYxXX4M2bUdGc6
N3ec592kD3ZDZopD8p/7DEJ4Y9HiD2971KE9dJeFt0g5QdYg/NA6s/rob8SKunE3
vouXsXgXT7PntgMTzlSdriVZzH81Xwj3QEUxep6
-----END
```

CERTIFICATE-----

Found in path(s):

*

/ca-certificates-20250911-r0-tar-gz/usr/share/ca-certificates/mozilla/GlobalSign_Root_R46.crt

1.26 libssl3 3.5.4-r0

1.26.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and

disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

- a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and
- b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.
- d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

- a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions

either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so,

attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (a program to direct compilers to make passes at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of

the source form of the
Standard Version of this Package without restriction, provided that you
duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived
from the Public Domain or from the Copyright Holder. A Package modified in such
a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that
you insert a prominent notice in each changed file stating how and when you
changed that file, and provided that you do at least ONE of the following:

a) place your modifications in the Public Domain or otherwise make them
Freely Available, such as by posting said modifications to Usenet or an
equivalent medium, or placing the modifications on a major archive site
such as ftp.uu.net, or by allowing the Copyright Holder to include your
modifications in the Standard Version of the Package.

b) use the modified
Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with
standard executables, which must also be provided, and provide a separate
manual page for each non-standard executable that clearly documents how it
differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable
form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files,
together with instructions (in the manual page or equivalent) on where to
get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package
with your modifications.

c) accompany any non-standard executables with their corresponding Standard
Version executables, giving the non-standard executables non-standard
names,
and clearly documenting the differences in manual pages (or
equivalent), together with instructions on where to get the Standard
Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this

Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.27 musl 1.2.5-r10

1.27.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
 * iconv.c
 * Implementation of SUSv4 XCU iconv utility
 * Copyright © 2011 Rich Felker
 * Licensed under the terms of the GNU General Public License, v2 or later
 */
```

Found in path(s):

* /aports-master-main-musl-zip/aports-master-main-musl/main/musl/iconv.c

No license file was found, but licenses were detected in source scan.

```
/*-
 * Copyright (c) 1996, 1998 The NetBSD Foundation, Inc.
 * All rights reserved.
 *
 * This code is derived from software contributed to The NetBSD Foundation
 * by J.T. Conklin.
 *
 * Mostly rewritten to be used in Alpine Linux (with musl c-library)
 * by Timo Teräs.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
```

- * notice, this list of conditions and the following disclaimer.
- * 2. Redistributions in binary form must reproduce the above copyright
- * notice, this list of conditions and the following disclaimer in the
- * documentation and/or other materials provided with the distribution.
- *
- * THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
- * ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT
- NOT LIMITED
- * TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
- * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
- * BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
- * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
- * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
- * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
- * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
- * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
- * POSSIBILITY OF SUCH DAMAGE.
- */

Found in path(s):

- * /aports-master-main-musl-zip/aports-master-main-musl/main/musl/getconf.c

No license file was found, but licenses were detected in source scan.

/*-

- * Copyright (c) 2004-2006 The NetBSD Foundation, Inc.
- * All rights reserved.
- *
- * This code is derived from software contributed to The NetBSD Foundation
- * by Luke Mewburn.
- * Timo Teräs cleaned up the code for use in Alpine Linux with musl libc.
- *
- * Redistribution and use in source and binary forms, with or without
- * modification, are permitted provided that the following conditions
- * are met:
- * 1. Redistributions of source code must retain the above copyright
- * notice, this list of conditions and the following disclaimer.
- * 2. Redistributions in binary form must reproduce the above copyright
- * notice, this list of conditions and the following disclaimer in the
- * documentation and/or other materials provided with the distribution.
- *
- * THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
- * ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
- *
- TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
- * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
- * BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
- * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
- * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):

* /aports-master-main-musl-zip/aports-master-main-musl/main/musl/getent.c

1.28 protobuf-go 1.36.10

1.28.1 Available under license :

Copyright (c) 2018 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.29 iptutils-clockdiff 20240905-r0

1.29.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to

your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below,

refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but

does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source

code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this

License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software

Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,
THERE IS NO WARRANTY

FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR

THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this

is what you want to do, use the GNU Lesser General Public License instead of this License.

1.30 iputils-arping 20240905-r0

1.30.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public

License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such

interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution

and only if you
received the program in object code or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source
code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing
or modifying
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing

version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

1.31 apk-tools 2.14.9-r3

1.31.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or

for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence

of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works.

But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage

or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it

with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all

the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions.

You may

not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of

this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted

in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE

THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH

HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and an idea of what it does.>

Copyright (C)< yyyy> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon >, 1 April 1989 Ty Coon, President of Vice This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

1.32 kelseyhightower-envconfig 1.4.0

1.32.1 Available under license :

Copyright (c) 2013 Kelsey Hightower

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.33 go.yaml.in/yaml/v2 2.4.3

1.33.1 Available under license :

Copyright 2011-2016 Canonical Ltd.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.34 otel-metric 1.38.0

1.34.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.35 openssl 3.5.4-r0

1.35.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE

Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions
for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

- a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and
- b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).
- c) If the modified program normally reads commands interactively when

run, you must cause it, when started running for such interactive use

in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that

accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License.

Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS

TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so,
attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software

Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (a program to direct compilers to make passes
at assemblers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less

customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
 - a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
 - b) use the modified Package only within your corporation or organization.
 - c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it

differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.36 auto-sdk 1.2.1

1.36.1 Available under license :

MIT License

Copyright (c) 2019 Montgomery Edwards and Faye Amacker

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

- (a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices
stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The MIT License (MIT)

Copyright (c) 2014 Cenk Alt

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of

the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015, Dave Cheney <dave@cheney.net>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR

SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2019-present Faye Amacker

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License

for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks,
or product names of the Licensor, except as required for
reasonable and customary use in describing the origin of the Work and
reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall
any Contributor be
liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or
claims asserted against, such Contributor by reason of your
accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own

identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to
in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Copyright 2011-2016 Canonical Ltd.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Data model artifacts for Prometheus.
Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).
MIT License

Copyright (c) 2016 json-iterator

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2011, Open Knowledge Foundation Ltd.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Open Knowledge Foundation Ltd. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014 gRPC authors.

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2016 Caleb Spare

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH

THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

procfs provides functions to retrieve system, kernel and process metrics from the pseudo-filesystem proc.

Copyright 2014-2015 The Prometheus Authors

This product includes software developed at SoundCloud Ltd. (<http://soundcloud.com/>).

This project is covered by two different licenses: MIT and Apache.

MIT License

The following files were ported to Go from C files of libyaml, and thus are still covered by their original MIT license, with the additional copyright starting in 2011 when the project was ported over:

apic.go emitterc.go parserc.go readerc.go scannerc.go
writerc.go yamllh.go yamprivateh.go

Copyright (c) 2006-2010 Kirill Simonov

Copyright (c) 2006-2011 Kirill Simonov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License

All the remaining project files are covered by the Apache license:

Copyright (c) 2011-2019 Canonical Ltd

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.
Copyright (c) 2017-2021 Uber Technologies, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Common libraries shared by Prometheus Go components.

Copyright 2015 The Prometheus Authors

This product includes software developed at SoundCloud Ltd. (<http://soundcloud.com/>).
Copyright (C) 2014-2019, Matt Butcher and Matt Farina

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN

CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2017 Nathan Sweet
Copyright (c) 2018, 2019 Cloudflare
Copyright (c) 2019 Authors of Cilium

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2020 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/* This is auto-generated file. See bpf_doc.py for details. */

/* Forward declarations of BPF structs */

```

struct bpf_fib_lookup;
struct bpf_sk_lookup;
struct bpf_perf_event_data;
struct bpf_perf_event_value;
struct bpf_pidns_info;
struct bpf_redir_neigh;
struct bpf_sock;
struct bpf_sock_addr;
struct bpf_sock_ops;
struct bpf_sock_tuple;
struct bpf_spin_lock;
struct bpf_sysctl;
struct bpf_tcp_sock;
struct bpf_tunnel_key;
struct bpf_xfrm_state;
struct linux_binprm;
struct pt_regs;
struct sk_reuseport_md;
struct sockaddr;
struct tcphdr;
struct seq_file;
struct tcp6_sock;
struct tcp_sock;
struct tcp_timewait_sock;
struct tcp_request_sock;
struct udp6_sock;
struct unix_sock;
struct task_struct;
struct cgroup;
struct __sk_buff;
struct sk_msg_md;
struct xdp_md;
struct path;
struct btf_ptr;
struct inode;
struct socket;
struct file;
struct bpf_timer;
struct mptcp_sock;
struct bpf_dynptr;
struct iphdr;
struct ipv6hdr;

/*
 * bpf_map_lookup_elem
 *
 * Perform a lookup in *map* for an entry associated

```



```

to *key*.
*
* Returns
* Map value associated to *key*, or **NULL** if no entry was
* found.
*/
static void (* const bpf_map_lookup_elem)(void *map, const void *key) = (void *) 1;

/*
* bpf_map_update_elem
*
* Add or update the value of the entry associated to *key* in
* *map* with *value*. *flags* is one of:
*
* **BPF_NOEXIST**
* The entry for *key* must not exist in the map.
* **BPF_EXIST**
* The entry for *key* must already exist in the map.
* **BPF_ANY**
* No condition on the existence of the entry for *key*.
*
* Flag value **BPF_NOEXIST** cannot be used for maps of types
* **BPF_MAP_TYPE_ARRAY** or **BPF_MAP_TYPE_PERCPU_ARRAY** (all
* elements always exist), the helper would return an error.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_map_update_elem)(void *map, const void *key, const void *value, __u64 flags) = (void *)
2;

/*
* bpf_map_delete_elem
*
* Delete entry with *key* from
* *map*.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_map_delete_elem)(void *map, const void *key) = (void *) 3;

/*
* bpf_probe_read
*
* For tracing programs, safely attempt to read *size* bytes from
* kernel space address *unsafe_ptr* and store the data in *dst*.
*

```

```

* Generally, use bpf_probe_read_user() or
* bpf_probe_read_kernel() instead.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_probe_read)(void *dst, __u32 size, const void *unsafe_ptr) = (void *) 4;

/*
* bpf_ktime_get_ns
*
* Return the time elapsed since system boot, in nanoseconds.
* Does not include time the system was suspended.
* See: clock_gettime() (CLOCK_MONOTONIC)
*
* Returns
* Current ktime.
*/
static __u64 (* const bpf_ktime_get_ns)(void) = (void *) 5;

/*
* bpf_trace_printk
*
* This helper is a "printf()-like" facility for debugging. It
* prints a message
* defined by format fmt (of size fmt_size)
* to file /sys/kernel/tracing/trace from TraceFS, if
* available. It can take up to three additional u64
* arguments (as an eBPF helpers, the total number of arguments is
* limited to five).
*
* Each time the helper is called, it appends a line to the trace.
* Lines are discarded while /sys/kernel/tracing/trace is
* open, use /sys/kernel/tracing/trace_pipe to avoid this.
* The format of the trace is customizable, and the exact output
* one will get depends on the options set in
* /sys/kernel/tracing/trace_options (see also the
* README file under the same directory). However, it usually
* defaults to something like:
*
* ::
*
* telnet-470 [001] .N.. 419421.045894: 0x00000001: <formatted msg>
*
* In the above:
*
* telnet is the name of the current task.
* 470 is the PID of the current task.

```

```

* * ``001`` is the CPU number on which the task is
*   running.
* * In
*   ``.N.``, each character refers to a set of
*   options (whether irqs are enabled, scheduling
*   options, whether hard/softirqs are running, level of
*   preempt_disabled respectively). **N** means that
*   **TIF_NEED_RESCHED** and **PREEMPT_NEED_RESCHED**
*   are set.
* * ``419421.045894`` is a timestamp.
* * ``0x00000001`` is a fake value used by BPF for the
*   instruction pointer register.
* * ``<formatted msg>`` is the message formatted with
*   *fmt*.
*
* The conversion specifiers supported by *fmt* are similar, but
* more limited than for printk(). They are **%d**, **%i**,
* **%u**, **%x**, **%ld**, **%li**, **%lu**, **%lx**, **%lld**,
* **%lli**, **%llu**, **%llx**, **%p**, **%s**. No modifier (size
* of field, padding with zeroes, etc.) is available, and the
* helper will return **-EINVAL** (but print nothing) if it
* encounters an unknown specifier.
*
* Also, note that **bpf_trace_printk**() is slow, and should
* only be used
  for debugging purposes. For this reason, a notice
  block (spanning several lines) is printed to kernel logs and
  states that the helper should not be used "for production use"
  the first time this helper is used (or more precisely, when
  **trace_printk**() buffers are allocated). For passing values
  to user space, perf events should be preferred.
*
* Returns
* The number of bytes written to the buffer, or a negative error
* in case of failure.
*/
static long (* const bpf_trace_printk)(const char *fmt, __u32 fmt_size, ...) = (void *) 6;

/*
* bpf_get_prandom_u32
*
* Get a pseudo-random number.
*
* From a security point of view, this helper uses its own
* pseudo-random internal state, and cannot be used to infer the
* seed of other random functions in the kernel. However, it is
* essential to note that the generator used by the helper is not
* cryptographically secure.

```

```

*
* Returns
* A random 32-bit unsigned value.
*/
static __u32 (* const bpf_get_prandom_u32)(void)
= (void *) 7;

/*
* bpf_get_smp_processor_id
*
* Get the SMP (symmetric multiprocessing) processor id. Note that
* all programs run with migration disabled, which means that the
* SMP processor id is stable during all the execution of the
* program.
*
* Returns
* The SMP id of the processor running the program.
*/
static __u32 (* const bpf_get_smp_processor_id)(void) = (void *) 8;

/*
* bpf_skb_store_bytes
*
* Store *len* bytes from address *from* into the packet
* associated to *skb*, at *offset*. *flags* are a combination of
* **BPF_F_RECOMPUTE_CSUM** (automatically recompute the
* checksum for the packet after storing the bytes) and
* **BPF_F_INVALIDATE_HASH** (set *skb*->hash**, *skb*->
* **->swhash** and *skb*->l4hash** to 0).
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously done by the verifier are invalidated and must
be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_store_bytes)(struct __sk_buff *skb, __u32 offset, const void *from, __u32 len, __u64
flags) = (void *) 9;

/*
* bpf_l3_csum_replace
*
* Recompute the layer 3 (e.g. IP) checksum for the packet
* associated to *skb*. Computation is incremental, so the helper

```

- * must know the former value of the header field that was
- * modified (*from*), the new value of this field (*to*), and the
- * number of bytes (2 or 4) for this field, stored in *size*.
- * Alternatively, it is possible to store the difference between
- * the previous and the new values of the header field in *to*, by
- * setting *from* and *size* to 0. For both methods, *offset*
- * indicates the location of the IP checksum within the packet.
- *
- * This helper works in combination with **bpf_csum_diff**(),
- * which does not update
- the checksum in-place, but offers more
- * flexibility and can handle sizes larger than 2 or 4 for the
- * checksum to update.
- *
- * A call to this helper is susceptible to change the underlying
- * packet buffer. Therefore, at load time, all checks on pointers
- * previously done by the verifier are invalidated and must be
- * performed again, if the helper is used in combination with
- * direct packet access.
- *
- * Returns
- * 0 on success, or a negative error in case of failure.
- */

```
static long (* const bpf_l3_csum_replace)(struct __sk_buff *skb, __u32 offset, __u64 from, __u64 to, __u64 size) =
(void *) 10;
```

```
/*
 * bpf_l4_csum_replace
 *
 * Recompute the layer 4 (e.g. TCP, UDP or ICMP) checksum for the
 * packet associated to *skb*. Computation is incremental, so the
 * helper must know the former value of the header field that was
 * modified (*from*), the new value of this field (*to*), and the
 * number of bytes (2 or 4) for this field, stored on the lowest
 * four bits
 * of *flags*. Alternatively, it is possible to store
 * the difference between the previous and the new values of the
 * header field in *to*, by setting *from* and the four lowest
 * bits of *flags* to 0. For both methods, *offset* indicates the
 * location of the IP checksum within the packet. In addition to
 * the size of the field, *flags* can be added (bitwise OR) actual
 * flags. With BPF_F_MARK_MANGLED_0, a null checksum is left
 * untouched (unless BPF_F_MARK_ENFORCE is added as well), and
 * for updates resulting in a null checksum the value is set to
 * CSUM_MANGLED_0 instead. Flag BPF_F_PSEUDO_HDR indicates
 * the checksum is to be computed against a pseudo-header.
 *
 * This helper works in combination with bpf_csum_diff( ),
```

- * which does not update the checksum in-place, but offers more
- * flexibility and can handle sizes larger than 2 or 4 for the
- * checksum to update.
- *
- * A call to this helper is susceptible to change the underlying
- * packet buffer. Therefore, at load time, all checks on pointers
- * previously done by the verifier are invalidated and must be
- * performed again, if the helper is used in combination with
- * direct packet access.
- *
- * Returns
- * 0 on success, or a negative error in case of failure.
- */

```
static long (* const bpf_l4_csum_replace)(struct __sk_buff *skb, __u32 offset, __u64 from, __u64 to, __u64 flags)
= (void *) 11;
```

/*

- * bpf_tail_call
- *
- * This special helper is used to trigger a "tail call", or in
- * other words, to jump into another eBPF program. The same stack
- * frame is used (but values on stack and in registers for the
- * caller are not accessible to the callee). This mechanism allows
- * for program chaining, either for raising the maximum number of
- * available eBPF instructions, or to execute given programs in
- * conditional blocks. For security reasons, there is an upper
- * limit to the number of successive tail calls that can be
- * performed.
- *
- * Upon call
- of this helper, the program attempts to jump into a
- * program referenced at index *index* in *prog_array_map*, a
- * special map of type **BPF_MAP_TYPE_PROG_ARRAY**, and passes
- * *ctx*, a pointer to the context.
- *
- * If the call succeeds, the kernel immediately runs the first
- * instruction of the new program. This is not a function call,
- * and it never returns to the previous program. If the call
- * fails, then the helper has no effect, and the caller continues
- * to run its subsequent instructions. A call can fail if the
- * destination program for the jump does not exist (i.e. *index*
- * is superior to the number of entries in *prog_array_map*), or
- * if the maximum number of tail calls has been reached for this
- * chain of programs. This limit is defined in the kernel by the
- * macro **MAX_TAIL_CALL_CNT** (not accessible to user space),
- * which is currently set to 33.
- *
- * Returns

```

* 0 on success, or a negative error in case of failure.
*/
static long (* const
bpf_tail_call)(void *ctx, void *prog_array_map, __u32 index) = (void *) 12;

/*
* bpf_clone_redirect
*
* Clone and redirect the packet associated to *skb* to another
* net device of index *ifindex*. Both ingress and egress
* interfaces can be used for redirection. The BPF_F_INGRESS
* value in *flags* is used to make the distinction (ingress path
* is selected if the flag is present, egress path otherwise).
* This is the only flag supported for now.
*
* In comparison with bpf_redirect() helper,
* bpf_clone_redirect() has the associated cost of
* duplicating the packet buffer, but this can be executed out of
* the eBPF program. Conversely, bpf_redirect() is more
* efficient, but it is handled through an action code where the
* redirection happens only after the eBPF program has returned.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously
done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure. Positive
* error indicates a potential drop or congestion in the target
* device. The particular positive error codes are not defined.
*/
static long (* const bpf_clone_redirect)(struct __sk_buff *skb, __u32 ifindex, __u64 flags) = (void *) 13;

/*
* bpf_get_current_pid_tgid
*
* Get the current pid and tgid.
*
* Returns
* A 64-bit integer containing the current tgid and pid, and
* created as such:
* current_task->tgid << 32 |
* current_task->pid.
*/
static __u64 (* const bpf_get_current_pid_tgid)(void) = (void *) 14;

```

```

/*
 * bpf_get_current_uid_gid
 *
 * Get the current uid and gid.
 *
 * Returns
 * A 64-bit integer containing the current GID and UID, and
 * created as such: *current_gid* **<< 32 \|** *current_uid*.
 */
static __u64 (* const
bpf_get_current_uid_gid)(void) = (void *) 15;

/*
 * bpf_get_current_comm
 *
 * Copy the **comm** attribute of the current task into *buf* of
 * *size_of_buf*. The **comm** attribute contains the name of
 * the executable (excluding the path) for the current task. The
 * *size_of_buf* must be strictly positive. On success, the
 * helper makes sure that the *buf* is NUL-terminated. On failure,
 * it is filled with zeroes.
 *
 * Returns
 * 0 on success, or a negative error in case of failure.
 */
static long (* const bpf_get_current_comm)(void *buf, __u32 size_of_buf) = (void *) 16;

/*
 * bpf_get_cgroup_classid
 *
 * Retrieve the classid for the current task, i.e. for the net_cls
 * cgroup to which *skb* belongs.
 *
 * This helper can be used on TC egress path, but not on ingress.
 *
 * The net_cls cgroup provides an interface to tag network packets
 * based on a user-provided identifier for all traffic coming from
 * the tasks belonging to the related cgroup. See also the
 * related
 * kernel documentation, available from the Linux sources in file
 * *Documentation/admin-guide/cgroup-v1/net_cls.rst*.
 *
 * The Linux kernel has two versions for cgroups: there are
 * cgroups v1 and cgroups v2. Both are available to users, who can
 * use a mixture of them, but note that the net_cls cgroup is for
 * cgroup v1 only. This makes it incompatible with BPF programs
 * run on cgroups, which is a cgroup-v2-only feature (a socket can

```



```

* only hold data for one version of cgroups at a time).
*
* This helper is only available is the kernel was compiled with
* the CONFIG_CGROUP_NET_CLASSID configuration option set to
* "y" or to "m".
*
* Returns
* The classid, or 0 for the default unconfigured classid.
*/
static __u32 (* const bpf_get_cgroup_classid)(struct __sk_buff *skb) = (void *) 17;

/*
* bpf_skb_vlan_push
*
* Push a *vlan_tci* (VLAN tag control information) of protocol
* *vlan_proto* to the packet associated to *skb*, then update
* the checksum. Note that if *vlan_proto* is different from
* ETH_P_8021Q and ETH_P_8021AD, it is considered to
* be ETH_P_8021Q.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_vlan_push)(struct __sk_buff *skb, __be16 vlan_proto, __u16 vlan_tci) = (void *) 18;

/*
* bpf_skb_vlan_pop
*
* Pop a VLAN header from the packet associated to *skb*.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_vlan_pop)(struct __sk_buff *skb) = (void *) 19;

/*

```

```

* bpf_skb_get_tunnel_key
*
* Get tunnel metadata. This helper takes a pointer *key* to an
* empty **struct bpf_tunnel_key** of **size**, that will be
* filled with tunnel metadata for the packet associated to *skb*.
* The *flags* can be set to **BPF_F_TUNINFO_IPV6**, which
* indicates that the tunnel is based on IPv6 protocol instead of
* IPv4.
*
* The **struct bpf_tunnel_key** is an object that generalizes the
* principal parameters used by various tunneling protocols into a
* single struct. This way, it can be used to easily make a
* decision based on the contents of the encapsulation header,
* "summarized" in this struct. In particular, it holds the IP
* address of the remote end (IPv4 or IPv6, depending on the case)
* in *key* **->remote_ipv4** or *key* **->remote_ipv6**. Also,
* this struct
* exposes the *key* **->tunnel_id**, which is
* generally mapped to a VNI (Virtual Network Identifier), making
* it programmable together with the **bpf_skb_set_tunnel_key**\
* () helper.
*
* Let's imagine that the following code is part of a program
* attached to the TC ingress interface, on one end of a GRE
* tunnel, and is supposed to filter out all messages coming from
* remote ends with IPv4 address other than 10.0.0.1:
*
* ::
*
* int ret;
* struct bpf_tunnel_key key = {};
*
* ret = bpf_skb_get_tunnel_key(skb, &key, sizeof(key), 0);
* if (ret < 0)
*     return TC_ACT_SHOT; // drop packet
*
* if (key.remote_ipv4 != 0x0a000001)
*     return TC_ACT_SHOT; // drop packet
*
* return TC_ACT_OK; // accept packet
*
* This interface can also be used with all encapsulation devices
* that can operate in "collect metadata" mode: instead of having
* one network device per specific configuration, the "collect
* metadata" mode only requires a single
* device where the
* configuration can be extracted from this helper.
*

```

```

* This can be used together with various tunnels such as VXLAN,
* Geneve, GRE or IP in IP (IPIP).
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_get_tunnel_key)(struct __sk_buff *skb, struct bpf_tunnel_key *key, __u32 size, __u64
flags) = (void *) 20;

/*
* bpf_skb_set_tunnel_key
*
* Populate tunnel metadata for packet associated to *skb.* The
* tunnel metadata is set to the contents of *key*, of *size*. The
* *flags* can be set to a combination of the following values:
*
* **BPF_F_TUNINFO_IPV6**
* Indicate that the tunnel is based on IPv6 protocol
* instead of IPv4.
* **BPF_F_ZERO_CSUM_TX**
* For IPv4 packets, add a flag to tunnel metadata
* indicating that checksum computation should be skipped
* and checksum set to zeroes.
* **BPF_F_DONT_FRAGMENT**
* Add a flag to tunnel metadata indicating that the
* packet should
* not be fragmented.
* **BPF_F_SEQ_NUMBER**
* Add a flag to tunnel metadata indicating that a
* sequence number should be added to tunnel header before
* sending the packet. This flag was added for GRE
* encapsulation, but might be used with other protocols
* as well in the future.
* **BPF_F_NO_TUNNEL_KEY**
* Add a flag to tunnel metadata indicating that no tunnel
* key should be set in the resulting tunnel header.
*
* Here is a typical usage on the transmit path:
*
* ::
*
* struct bpf_tunnel_key key;
*     populate key ...
* bpf_skb_set_tunnel_key(skb, &key, sizeof(key), 0);
* bpf_clone_redirect(skb, vxlan_dev_ifindex, 0);
*
* See also the description of the bpf_skb_get_tunnel_key()
* helper for additional information.

```

```

*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_set_tunnel_key)(struct __sk_buff *skb, struct bpf_tunnel_key *key, __u32 size, __u64
flags) = (void *) 21;

/*
* bpf_perf_event_read
*
* Read the value of a perf event counter. This helper relies on a
* *map* of type BPF_MAP_TYPE_PERF_EVENT_ARRAY. The nature of
* the perf event counter is selected when *map* is updated with
* perf event file descriptors. The *map* is an array whose size
* is the number of available CPUs, and each cell contains a value
* relative to one CPU. The value to retrieve is indicated by
* *flags*, that contains the index of the CPU to look up, masked
* with BPF_F_INDEX_MASK. Alternatively, *flags* can be set to
* BPF_F_CURRENT_CPU to indicate that the value for the
* current CPU should be retrieved.
*
* Note that before Linux 4.13, only hardware perf event can be
* retrieved.
*
* Also, be aware that the newer helper
* bpf_perf_event_read_value() is recommended over
* bpf_perf_event_read() in general. The latter has some ABI
* quirks where error and counter value are used as a return code
* (which is wrong
to do since ranges may overlap). This issue is
* fixed with bpf_perf_event_read_value(), which at the same
* time provides more features over the bpf_perf_event_read\
* () interface. Please refer to the description of
* bpf_perf_event_read_value() for details.
*
* Returns
* The value of the perf event counter read from the map, or a
* negative error code in case of failure.
*/
static __u64 (* const bpf_perf_event_read)(void *map, __u64 flags) = (void *) 22;

/*
* bpf_redirect
*
* Redirect the packet to another net device of index *ifindex*.
* This helper is somewhat similar to bpf_clone_redirect\
* (), except that the packet is not cloned, which provides
* increased performance.

```

```

*
* Except for XDP, both ingress and egress interfaces can be used
* for redirection. The BPF_F_INGRESS value in *flags* is used
* to make the distinction (ingress path is selected if the flag
* is present, egress path otherwise). Currently, XDP only
*
    supports redirection to the egress interface, and accepts no
* flag at all.
*
* The same effect can also be attained with the more generic
* bpf_redirect_map(), which uses a BPF map to store the
* redirect target instead of providing it directly to the helper.
*
* Returns
* For XDP, the helper returns XDP_REDIRECT on success or
* XDP_ABORTED on error. For other program types, the values
* are TC_ACT_REDIRECT on success or TC_ACT_SHOT on
* error.
*/
static long (* const bpf_redirect)(__u32 ifindex, __u64 flags) = (void *) 23;

/*
* bpf_get_route_realm
*
* Retrieve the realm or the route, that is to say the
* tc_clsact field of the destination for the *skb*. The
* identifier retrieved is a user-provided tag, similar to the
* one used with the net_cls cgroup (see description for
* bpf_get_cgroup_classid() helper), but here this tag is
* held by a route (a destination entry), not by a task.
*
* Retrieving this identifier
    works with the clsact TC egress hook
* (see also tc-bpf(8)), or alternatively on conventional
* classful egress qdiscs, but not on TC ingress path. In case of
* clsact TC egress hook, this has the advantage that, internally,
* the destination entry has not been dropped yet in the transmit
* path. Therefore, the destination entry does not need to be
* artificially held via netif_keep_dst() for a classful
* qdisc until the *skb* is freed.
*
* This helper is available only if the kernel was compiled with
* CONFIG_IP_ROUTE_CLASSID configuration option.
*
* Returns
* The realm of the route for the packet associated to *skb*, or 0
* if none was found.
*/

```

```
static __u32 (* const bpf_get_route_realm)(struct __sk_buff *skb) = (void *) 24;
```

```
/*  
 * bpf_perf_event_output  
 *  
 * Write raw *data* blob into a special BPF perf event held by  
 * *map* of type **BPF_MAP_TYPE_PERF_EVENT_ARRAY**. This perf  
 * event must have the following attributes: **PERF_SAMPLE_RAW**  
 * as **sample_type**, **PERF_TYPE_SOFTWARE** as **type**, and  
 * **PERF_COUNT_SW_BPF_OUTPUT** as **config**.  
 *  
 * The *flags* are used to indicate the index in *map* for which  
 * the value must be put, masked with **BPF_F_INDEX_MASK**.  
 * Alternatively, *flags* can be set to **BPF_F_CURRENT_CPU**  
 * to indicate that the index of the current CPU core should be  
 * used.  
 *  
 * The value to write, of *size*, is passed through eBPF stack and  
 * pointed by *data*.  
 *  
 * The context of the program *ctx* needs also be passed to the  
 * helper.  
 *  
 * On user space, a program willing to read the values needs to  
 * call **perf_event_open**\ () on the perf event (either for  
 * one or for all CPUs) and to store the file descriptor into the  
 * *map*. This must be done before the eBPF program can send data  
 * into it. An example is available in file  
 * *samples/bpf/trace_output_user.c* in the Linux kernel source  
 * tree (the eBPF program counterpart is in  
 * *samples/bpf/trace_output_kern.c* ).  
 *  
 * **bpf_perf_event_output**\ () achieves better performance  
 * than **bpf_trace_printk**\ () for sharing data with user  
 * space, and is much better suitable for streaming data from eBPF  
 * programs.  
 *  
 * Note that this helper is not restricted to tracing use cases  
 * and can be used with programs attached to TC or XDP as well,  
 * where it allows for passing data to user space listeners. Data  
 * can be:  
 *  
 * * Only custom structs,  
 * * Only the packet payload, or  
 * * A combination of both.  
 *  
 * Returns  
 * 0 on success, or a negative error in case of failure.
```

```

*/
static long (* const bpf_perf_event_output)(void *ctx, void *map, __u64 flags, void *data, __u64 size) = (void *)
25;

/*
 * bpf_skb_load_bytes
 *
 * This helper was provided as an easy way to load data from a
 * packet. It can be used to load *len* bytes from *offset* from
 * the packet associated to *skb*, into the buffer pointed by
 * *to*.
 *
 * Since Linux 4.7, usage of this helper
 * has mostly been replaced
 * by "direct packet access", enabling packet data to be
 * manipulated with *skb->data** and *skb->data_end**
 * pointing respectively to the first byte of packet data and to
 * the byte after the last byte of packet data. However, it
 * remains useful if one wishes to read large quantities of data
 * at once from a packet into the eBPF stack.
 *
 * Returns
 * 0 on success, or a negative error in case of failure.
 */
static long (* const bpf_skb_load_bytes)(const void *skb, __u32 offset, void *to, __u32 len) = (void *) 26;

/*
 * bpf_get_stackid
 *
 * Walk a user or a kernel stack and return its id. To achieve
 * this, the helper needs *ctx*, which is a pointer to the context
 * on which the tracing program is executed, and a pointer to a
 * *map* of type **BPF_MAP_TYPE_STACK_TRACE**.
 *
 * The last argument, *flags*, holds the number of stack frames to
 * skip (from 0 to 255), masked with
 * **BPF_F_SKIP_FIELD_MASK**. The next bits can
 * be used to set
 * a combination of the following flags:
 *
 * **BPF_F_USER_STACK**
 * Collect a user space stack instead of a kernel stack.
 * **BPF_F_FAST_STACK_CMP**
 * Compare stacks by hash only.
 * **BPF_F_REUSE_STACKID**
 * If two different stacks hash into the same *stackid*,
 * discard the old one.
 */

```

- * The stack id retrieved is a 32 bit long integer handle which
- * can be further combined with other data (including other stack
- * ids) and used as a key into maps. This can be useful for
- * generating a variety of graphs (such as flame graphs or off-cpu
- * graphs).
- *
- * For walking a stack, this helper is an improvement over
- * `bpf_probe_read()`, which can be used with unrolled loops
- * but is not efficient and consumes a lot of eBPF instructions.
- * Instead, `bpf_get_stackid()` can collect up to
- * `PERF_MAX_STACK_DEPTH` both kernel and user frames. Note that
- * this limit can be controlled with the `sysctl` program, and
- * that it should
- be manually increased in order to profile long
- * user stacks (such as stacks for Java programs). To do so, use:
- *
- * ::
- *
- * `# sysctl kernel.perf_event_max_stack=<new value>`
- *
- * Returns
- * The positive or null stack id on success, or a negative error
- * in case of failure.
- */

```
static long (* const bpf_get_stackid)(void *ctx, void *map, __u64 flags) = (void *) 27;
```

/*

- * `bpf_csum_diff`
- *
- * Compute a checksum difference, from the raw buffer pointed by
- * `from`, of length `from_size` (that must be a multiple of 4),
- * towards the raw buffer pointed by `to`, of size `to_size`
- * (same remark). An optional `seed` can be added to the value
- * (this can be cascaded, the seed may come from a previous call
- * to the helper).
- *
- * This is flexible enough to be used in several ways:
- *
- * * With `from_size == 0`, `to_size > 0` and `seed` set to
- * checksum, it can be used when pushing new data.
- * * With `from_size > 0`, `to_size == 0` and `seed` set to
- * checksum,
- it can be used when removing data from a packet.
- * * With `from_size > 0`, `to_size > 0` and `seed` set to 0, it
- * can be used to compute a diff. Note that `from_size` and
- * `to_size` do not need to be equal.
- *
- * This helper can be used in combination with


```

* **bpf_l3_csum_replace**\ () and **bpf_l4_csum_replace**\ (), to
* which one can feed in the difference computed with
* **bpf_csum_diff**\ ().
*
* Returns
* The checksum result, or a negative error code in case of
* failure.
*/
static __s64 (* const bpf_csum_diff)(__be32 *from, __u32 from_size, __be32 *to, __u32 to_size, __wsum seed) =
(void *) 28;

/*
* bpf_skb_get_tunnel_opt
*
* Retrieve tunnel options metadata for the packet associated to
* *skb*, and store the raw tunnel option data to the buffer *opt*
* of *size*.
*
* This helper can be used with encapsulation devices that can
* operate in "collect metadata" mode (please refer to the related
* note in the description of **bpf_skb_get_tunnel_key**\
() for
* more details). A particular example where this can be used is
* in combination with the Geneve encapsulation protocol, where it
* allows for pushing (with **bpf_skb_get_tunnel_opt**\ () helper)
* and retrieving arbitrary TLVs (Type-Length-Value headers) from
* the eBPF program. This allows for full customization of these
* headers.
*
* Returns
* The size of the option data retrieved.
*/
static long (* const bpf_skb_get_tunnel_opt)(struct __sk_buff *skb, void *opt, __u32 size) = (void *) 29;

/*
* bpf_skb_set_tunnel_opt
*
* Set tunnel options metadata for the packet associated to *skb*
* to the option data contained in the raw buffer *opt* of *size*.
*
* See also the description of the **bpf_skb_get_tunnel_opt**\ ()
* helper for additional information.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_set_tunnel_opt)(struct __sk_buff *skb, void *opt, __u32 size) = (void *) 30;

```

```

/*
 *
 * bpf_skb_change_proto
 *
 * Change the protocol of the *skb* to *proto*. Currently
 * supported are transition from IPv4 to IPv6, and from IPv6 to
 * IPv4. The helper takes care of the groundwork for the
 * transition, including resizing the socket buffer. The eBPF
 * program is expected to fill the new headers, if any, via
 * **skb_store_bytes**\ () and to recompute the checksums with
 * **bpf_l3_csum_replace**\ () and **bpf_l4_csum_replace**\
 * (). The main case for this helper is to perform NAT64
 * operations out of an eBPF program.
 *
 * Internally, the GSO type is marked as dodgy so that headers are
 * checked and segments are recalculated by the GSO/GRO engine.
 * The size for GSO target is adapted as well.
 *
 * All values for *flags* are reserved for future usage, and must
 * be left at zero.
 *
 * A call to this helper is susceptible to change the underlying
 * packet buffer. Therefore, at load time, all checks on pointers
 * previously done by the verifier are
 * invalidated and must be
 * performed again, if the helper is used in combination with
 * direct packet access.
 *
 * Returns
 * 0 on success, or a negative error in case of failure.
 */
static long (* const bpf_skb_change_proto)(struct __sk_buff *skb, __be16 proto, __u64 flags) = (void *) 31;

/*
 * bpf_skb_change_type
 *
 * Change the packet type for the packet associated to *skb*. This
 * comes down to setting *skb* **->pkt_type** to *type*, except
 * the eBPF program does not have a write access to *skb*\
 * **->pkt_type** beside this helper. Using a helper here allows
 * for graceful handling of errors.
 *
 * The major use case is to change incoming *skb*s to
 * **PACKET_HOST** in a programmatic way instead of having to
 * recirculate via **redirect**\ (... , **BPF_F_INGRESS**), for
 * example.
 *
 * Note that *type* only allows certain values. At this time, they

```

```

* are:
*
* **PACKET_HOST**
* Packet is for us.
* **PACKET_BROADCAST**
* Send packet to all.
* **PACKET_MULTICAST**
* Send packet to group.
* **PACKET_OTHERHOST**
* Send packet to someone else.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_change_type)(struct __sk_buff *skb, __u32 type) = (void *) 32;

/*
* bpf_skb_under_cgroup
*
* Check whether *skb* is a descendant of the cgroup2 held by
* *map* of type **BPF_MAP_TYPE_CGROUP_ARRAY**, at *index*.
*
* Returns
* The return value depends on the result of the test, and can be:
*
* * 0, if the *skb* failed the cgroup2 descendant test.
* * 1, if the *skb* succeeded the cgroup2 descendant test.
* * A negative error code, if an error occurred.
*/
static long (* const bpf_skb_under_cgroup)(struct __sk_buff *skb, void *map, __u32 index) = (void *) 33;

/*
* bpf_get_hash_recalc
*
* Retrieve the hash of the packet, *skb* \ **->hash**. If it is
* not set, in particular if the hash was cleared due to mangling,
* recompute this hash. Later accesses
* to the hash can be done
* directly with *skb* \ **->hash**.
*
* Calling **bpf_set_hash_invalid**\ (), changing a packet
* prototype with **bpf_skb_change_proto**\ (), or calling
* **bpf_skb_store_bytes**\ () with the
* **BPF_F_INVALIDATE_HASH** are actions susceptible to clear
* the hash and to trigger a new computation for the next call to
* **bpf_get_hash_recalc**\ ().
*
* Returns

```

```

* The 32-bit hash.
*/
static __u32 (* const bpf_get_hash_recalc)(struct __sk_buff *skb) = (void *) 34;

/*
* bpf_get_current_task
*
* Get the current task.
*
* Returns
* A pointer to the current task struct.
*/
static __u64 (* const bpf_get_current_task)(void) = (void *) 35;

/*
* bpf_probe_write_user
*
* Attempt in a safe way to write *len* bytes from the buffer
* *src* to *dst* in memory. It only works for threads that are in
* user context, and *dst* must be a valid user space address.
*
* This helper should not be used to implement any kind of
* security mechanism
because of TOC-TOU attacks, but rather to
* debug, divert, and manipulate execution of semi-cooperative
* processes.
*
* Keep in mind that this feature is meant for experiments, and it
* has a risk of crashing the system and running programs.
* Therefore, when an eBPF program using this helper is attached,
* a warning including PID and process name is printed to kernel
* logs.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_probe_write_user)(void *dst, const void *src, __u32 len) = (void *) 36;

/*
* bpf_current_task_under_cgroup
*
* Check whether the probe is being run in the context of a given
* subset of the cgroup2 hierarchy. The cgroup2 to test is held by
* *map* of type **BPF_MAP_TYPE_CGROUP_ARRAY**, at *index*.
*
* Returns
* The return value depends on the result of the test, and can be:
*

```

```

* * 1, if current task belongs to the cgroup2.
* * 0, if current task does not belong to the cgroup2.
* * A negative error code, if an error occurred.
*/
static long (* const bpf_current_task_under_cgroup)(void *map, __u32 index) = (void *) 37;

/*
* bpf_skb_change_tail
*
* Resize (trim or grow) the packet associated to *skb* to the
* new *len*. The *flags* are reserved for future usage, and must
* be left at zero.
*
* The basic idea is that the helper performs the needed work to
* change the size of the packet, then the eBPF program rewrites
* the rest via helpers like **bpf_skb_store_bytes**\ (),
* **bpf_l3_csum_replace**\ (), **bpf_l3_csum_replace**\ ()
* and others. This helper is a slow path utility intended for
* replies with control messages. And because it is targeted for
* slow path, the helper itself can afford to be slow: it
* implicitly linearizes, unclones and drops offloads from the
* *skb*.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously done
by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_change_tail)(struct __sk_buff *skb, __u32 len, __u64 flags) = (void *) 38;

/*
* bpf_skb_pull_data
*
* Pull in non-linear data in case the *skb* is non-linear and not
* all of *len* are part of the linear section. Make *len* bytes
* from *skb* readable and writable. If a zero value is passed for
* *len*, then all bytes in the linear part of *skb* will be made
* readable and writable.
*
* This helper is only needed for reading and writing with direct
* packet access.
*
* For direct packet access, testing that offsets to access

```

```

* are within packet boundaries (test on *skb* \ **->data_end**) is
* susceptible to fail if offsets are invalid, or if the requested
* data is in non-linear parts of the *skb*. On failure the
* program
can just bail out, or in the case of a non-linear
* buffer, use a helper to make the data available. The
* **bpf_skb_load_bytes**() helper is a first solution to access
* the data. Another one consists in using **bpf_skb_pull_data**
* to pull in once the non-linear parts, then retesting and
* eventually access the data.
*
* At the same time, this also makes sure the *skb* is uncloned,
* which is a necessary condition for direct write. As this needs
* to be an invariant for the write part only, the verifier
* detects writes and adds a prologue that is calling
* **bpf_skb_pull_data**() to effectively unclone the *skb* from
* the very beginning in case it is indeed cloned.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
*
Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_pull_data)(struct __sk_buff *skb, __u32 len) = (void *) 39;

/*
* bpf_csum_update
*
* Add the checksum *csum* into *skb* \ **->csum** in case the
* driver has supplied a checksum for the entire packet into that
* field. Return an error otherwise. This helper is intended to be
* used in combination with **bpf_csum_diff**(), in particular
* when the checksum needs to be updated after data has been
* written into the packet through direct packet access.
*
* Returns
* The checksum on success, or a negative error code in case of
* failure.
*/
static __s64 (* const bpf_csum_update)(struct __sk_buff *skb, __wsum csum) = (void *) 40;

/*
* bpf_set_hash_invalid

```

```

*
* Invalidate the current *skb* **->hash**. It can be used after
* mangling on headers through direct packet access, in order to
* indicate that the hash is outdated and to trigger a
* recalculation the next
time the kernel tries to access this
* hash or when the **bpf_get_hash_recalc**() helper is called.
*
* Returns
* void.
*/
static void (* const bpf_set_hash_invalid)(struct __sk_buff *skb) = (void *) 41;

/*
* bpf_get_numa_node_id
*
* Return the id of the current NUMA node. The primary use case
* for this helper is the selection of sockets for the local NUMA
* node, when the program is attached to sockets using the
* **SO_ATTACH_REUSEPORT_EBPF** option (see also **socket(7)**),
* but the helper is also available to other eBPF program types,
* similarly to **bpf_get_smp_processor_id**().
*
* Returns
* The id of current NUMA node.
*/
static long (* const bpf_get_numa_node_id)(void) = (void *) 42;

/*
* bpf_skb_change_head
*
* Grows headroom of packet associated to *skb* and adjusts the
* offset of the MAC header accordingly, adding *len* bytes of
* space. It automatically extends and reallocates memory as
* required.
*
* This helper can be used
on a layer 3 *skb* to push a MAC header
* for redirection into a layer 2 device.
*
* All values for *flags* are reserved for future usage, and must
* be left at zero.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.

```

```

*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_change_head)(struct __sk_buff *skb, __u32 len, __u64 flags) = (void *) 43;

/*
* bpf_xdp_adjust_head
*
* Adjust (move) *xdp_md* **->data** by *delta* bytes. Note that
* it is possible to use a negative value for *delta*. This helper
* can be used to prepare the packet for pushing or popping
* headers.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all
* checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_xdp_adjust_head)(struct xdp_md *xdp_md, int delta) = (void *) 44;

/*
* bpf_probe_read_str
*
* Copy a NUL terminated string from an unsafe kernel address
* *unsafe_ptr* to *dst*. See **bpf_probe_read_kernel_str**\ () for
* more details.
*
* Generally, use **bpf_probe_read_user_str**\ () or
* **bpf_probe_read_kernel_str**\ () instead.
*
* Returns
* On success, the strictly positive length of the string,
* including the trailing NUL character. On error, a negative
* value.
*/
static long (* const bpf_probe_read_str)(void *dst, __u32 size, const void *unsafe_ptr) = (void *) 45;

/*
* bpf_get_socket_cookie
*
* If the **struct sk_buff** pointed by *skb* has a known socket,
* retrieve the cookie

```


(generated by the kernel) of this socket.

- * If no cookie has been set yet, generate a new cookie. Once
- * generated, the socket cookie remains stable for the life of the
- * socket. This helper can be useful for monitoring per socket
- * networking traffic statistics as it provides a global socket
- * identifier that can be assumed unique.

*

* Returns

- * A 8-byte long unique number on success, or 0 if the socket
- * field is missing inside *skb*.

*/

```
static __u64 (* const bpf_get_socket_cookie)(void *ctx) = (void *) 46;
```

/*

* bpf_get_socket_uid

*

- * Get the owner UID of the socket associated to *skb*.

*

* Returns

- * The owner UID of the socket associated to *skb*. If the socket
- * is **NULL**, or if it is not a full socket (i.e. if it is a
- * time-wait or a request socket instead), **overflowuid** value
- * is returned (note that **overflowuid** might also be the actual
- * UID value for the socket).

*/

```
static __u32 (* const bpf_get_socket_uid)(struct __sk_buff *skb) =  
(void *) 47;
```

/*

* bpf_set_hash

*

- * Set the full hash for *skb* (set the field *skb*->hash*)
- * to value *hash*.

*

* Returns

- * 0

*/

```
static long (* const bpf_set_hash)(struct __sk_buff *skb, __u32 hash) = (void *) 48;
```

/*

* bpf_setsockopt

*

- * Emulate a call to **setsockopt()** on the socket associated to
- * *bpf_socket*, which must be a full socket. The *level* at
- * which the option resides and the name *optname* of the option
- * must be specified, see **setsockopt(2)** for more information.
- * The option value of length *optlen* is pointed by *optval*.

*

```

* *bpf_socket* should be one of the following:
*
* * **struct bpf_sock_ops** for **BPF_PROG_TYPE_SOCK_OPS**.
* * **struct bpf_sock_addr** for **BPF_CGROUP_INET4_CONNECT**,
* * **BPF_CGROUP_INET6_CONNECT** and **BPF_CGROUP_UNIX_CONNECT**.
*
* This helper actually implements a subset of **setsockopt()**.
* It supports the following *level*\ s:
*
* * **SOL_SOCKET**, which supports the following *optname*\
  S:
* * **SO_RCVBUF**, **SO_SNDBUF**, **SO_MAX_PACING_RATE**,
* * **SO_PRIORITY**, **SO_RCVLOWAT**, **SO_MARK**,
* * **SO_BINDTODEVICE**, **SO_KEEPALIVE**, **SO_REUSEADDR**,
* * **SO_REUSEPORT**, **SO_BINDTOIFINDEX**, **SO_TXREHASH**.
* * **IPPROTO_TCP**, which supports the following *optname*\ s:
* * **TCP_CONGESTION**, **TCP_BPF_IW**,
* * **TCP_BPF_SNDCWND_CLAMP**, **TCP_SAVE_SYN**,
* * **TCP_KEEPIDLE**, **TCP_KEEPINTVL**, **TCP_KEEPCNT**,
* * **TCP_SYNCNT**, **TCP_USER_TIMEOUT**, **TCP_NOTSENT_LOWAT**,
* * **TCP_NODELAY**, **TCP_MAXSEG**, **TCP_WINDOW_CLAMP**,
* * **TCP_THIN_LINEAR_TIMEOUTS**, **TCP_BPF_DELACK_MAX**,
* * **TCP_BPF_RTO_MIN**.
* * **IPPROTO_IP**, which supports *optname* **IP_TOS**.
* * **IPPROTO_IPV6**, which supports the following *optname*\ s:
* * **IPV6_TCLASS**, **IPV6_AUTOFLOWLABEL**.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_setsockopt)(void *bpf_socket, int level,
int optname, void *optval, int optlen) = (void *) 49;

/*
* bpf_skb_adjust_room
*
* Grow or shrink the room for data in the packet associated to
* *skb* by *len_diff*, and according to the selected *mode*.
*
* By default, the helper will reset any offloaded checksum
* indicator of the skb to CHECKSUM_NONE. This can be avoided
* by the following flag:
*
* * **BPF_F_ADJ_ROOM_NO_CSUM_RESET**: Do not reset offloaded
* checksum data of the skb to CHECKSUM_NONE.
*
* There are two supported modes at this time:
*

```

```

* * **BPF_ADJ_ROOM_MAC**: Adjust room at the mac layer
* (room space is added or removed between the layer 2 and
* layer 3 headers).
*
* * **BPF_ADJ_ROOM_NET**: Adjust room at the network layer
* (room space is added or removed between the layer 3 and
* layer 4 headers).
*
* The following flags are supported at this time:
*
* * **BPF_F_ADJ_ROOM_FIXED_GSO**: Do not adjust gso_size.
* Adjusting mss in this way is not allowed for datagrams.
*
* * **BPF_F_ADJ_ROOM_ENCAP_L3_IPV4**,
* **BPF_F_ADJ_ROOM_ENCAP_L3_IPV6**:
* Any new space is reserved to hold a tunnel header.
* Configure skb offsets and other fields accordingly.
*
* * **BPF_F_ADJ_ROOM_ENCAP_L4_GRE**,
* **BPF_F_ADJ_ROOM_ENCAP_L4_UDP**:
* Use with ENCAP_L3 flags to further specify the tunnel type.
*
* * **BPF_F_ADJ_ROOM_ENCAP_L2**\ (*len*):
* Use with ENCAP_L3/L4 flags to further specify the tunnel
* type; *len* is the length of the inner MAC header.
*
* * **BPF_F_ADJ_ROOM_ENCAP_L2_ETH**:
* Use with BPF_F_ADJ_ROOM_ENCAP_L2 flag to further specify the
* L2 type as Ethernet.
*
* * **BPF_F_ADJ_ROOM_DECAP_L3_IPV4**,
* **BPF_F_ADJ_ROOM_DECAP_L3_IPV6**:
* Indicate the new IP header version after decapsulating the outer
* IP header. Used when the inner and outer IP versions are different.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all
checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_adjust_room)(struct __sk_buff *skb, __s32 len_diff, __u32 mode, __u64 flags) = (void
*) 50;

```

```

/*
 * bpf_redirect_map
 *
 * Redirect the packet to the endpoint referenced by *map* at
 * index *key*. Depending on its type, this *map* can contain
 * references to net devices (for forwarding packets through other
 * ports), or to CPUs (for redirecting XDP frames to another CPU;
 * but this is only implemented for native XDP (with driver
 * support) as of this writing).
 *
 * The lower two bits of *flags* are used as the return code if
 * the map lookup fails. This is so that the return value can be
 * one of the XDP program return codes up to XDP_TX, as chosen
 * by the caller. The higher bits of *flags*
 * can be set to
 * BPF_F_BROADCAST or BPF_F_EXCLUDE_INGRESS as defined below.
 *
 * With BPF_F_BROADCAST the packet will be broadcasted to all the
 * interfaces in the map, with BPF_F_EXCLUDE_INGRESS the ingress
 * interface will be excluded when do broadcasting.
 *
 * See also bpf_redirect( ), which only supports redirecting
 * to an ifindex, but doesn't require a map to do so.
 *
 * Returns
 * XDP_REDIRECT on success, or the value of the two lower bits
 * of the *flags* argument on error.
 */
static long (* const bpf_redirect_map)(void *map, __u64 key, __u64 flags) = (void *) 51;

/*
 * bpf_sk_redirect_map
 *
 * Redirect the packet to the socket referenced by *map* (of type
 * BPF_MAP_TYPE_SOCKMAP) at index *key*. Both ingress and
 * egress interfaces can be used for redirection. The
 * BPF_F_INGRESS value in *flags* is used to make the
 * distinction (ingress path is selected if the flag is present,
 * egress path otherwise). This is the only flag supported
 * for now.
 *
 * Returns
 * SK_PASS on success, or SK_DROP on error.
 */
static long (* const bpf_sk_redirect_map)(struct __sk_buff *skb, void *map, __u32 key, __u64 flags) = (void *) 52;

/*
 * bpf_sock_map_update

```

```

*
* Add an entry to, or update a *map* referencing sockets. The
* *skops* is used as a new value for the entry associated to
* *key*. *flags* is one of:
*
* **BPF_NOEXIST**
* The entry for *key* must not exist in the map.
* **BPF_EXIST**
* The entry for *key* must already exist in the map.
* **BPF_ANY**
* No condition on the existence of the entry for *key*.
*
* If the *map* has eBPF programs (parser and verdict), those will
* be inherited by the socket being added. If the socket is
* already attached to eBPF programs, this results in an error.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_sock_map_update)(struct bpf_sock_ops *skops, void *map, void *key, __u64 flags) = (void
*)
53;

/*
* bpf_xdp_adjust_meta
*
* Adjust the address pointed by *xdp_md* \ **->data_meta** by
* *delta* (which can be positive or negative). Note that this
* operation modifies the address stored in *xdp_md* \ **->data**,
* so the latter must be loaded only after the helper has been
* called.
*
* The use of *xdp_md* \ **->data_meta** is optional and programs
* are not required to use it. The rationale is that when the
* packet is processed with XDP (e.g. as DoS filter), it is
* possible to push further meta data along with it before passing
* to the stack, and to give the guarantee that an ingress eBPF
* program attached as a TC classifier on the same device can pick
* this up for further post-processing. Since TC works with socket
* buffers, it remains possible to set from XDP the **mark** or
* **priority** pointers, or other pointers for the socket buffer.
* Having this scratch space generic and programmable allows for
* more flexibility as the user is
free to store whatever meta
* data they need.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers

```

```

* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_xdp_adjust_meta)(struct xdp_md *xdp_md, int delta) = (void *) 54;

/*
* bpf_perf_event_read_value
*
* Read the value of a perf event counter, and store it into *buf*
* of size *buf_size*. This helper relies on a *map* of type
* **BPF_MAP_TYPE_PERF_EVENT_ARRAY**. The nature of the perf event
* counter is selected when *map* is updated with perf event file
* descriptors. The *map* is an array whose size is the number of
* available CPUs, and each cell contains a value relative to one
* CPU. The value to retrieve is indicated by *flags*, that
* contains the index of the CPU to look up, masked with
* **BPF_F_INDEX_MASK**. Alternatively, *flags* can be set to
* **BPF_F_CURRENT_CPU** to indicate that the value for the
* current CPU should be retrieved.
*
* This helper behaves in a way close to
* **bpf_perf_event_read**() helper, save that instead of
* just returning the value observed, it fills the *buf*
* structure. This allows for additional data to be retrieved: in
* particular, the enabled and running times (in *buf*
* **->enabled** and *buf* **->running**, respectively) are
* copied. In general, **bpf_perf_event_read_value**() is
* recommended over **bpf_perf_event_read**(), which has some
* ABI issues and provides fewer functionalities.
*
* These values are interesting, because hardware PMU (Performance
* Monitoring Unit) counters are limited resources. When there are
* more PMU based perf events opened than available counters,
* kernel will multiplex these events so each event
* gets certain
* percentage (but not all) of the PMU time. In case that
* multiplexing happens, the number of samples or counter value
* will not reflect the case compared to when no multiplexing
* occurs. This makes comparison between different runs difficult.
* Typically, the counter value should be normalized before
* comparing to other experiments. The usual normalization is done
* as follows.
*
* ::

```

```

*
* normalized_counter = counter * t_enabled / t_running
*
* Where t_enabled is the time enabled for event and t_running is
* the time running for event since last normalization. The
* enabled and running times are accumulated since the perf event
* open. To achieve scaling factor between two invocations of an
* eBPF program, users can use CPU id as the key (which is
* typical for perf array usage model) to remember the previous
* value and do the calculation inside the eBPF program.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_perf_event_read_value)(void *map, __u64 flags, struct bpf_perf_event_value *buf, __u32
buf_size) = (void *) 55;

/*
* bpf_perf_prog_read_value
*
* For an eBPF program attached to a perf event, retrieve the
* value of the event counter associated to *ctx* and store it in
* the structure pointed by *buf* and of size *buf_size*. Enabled
* and running times are also stored in the structure (see
* description of helper **bpf_perf_event_read_value**\ () for
* more details).
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_perf_prog_read_value)(struct bpf_perf_event_data *ctx, struct bpf_perf_event_value *buf,
__u32 buf_size) = (void *) 56;

/*
* bpf_getsockopt
*
* Emulate a call to **getsockopt()** on the socket associated to
* *bpf_socket*, which must be a full socket. The *level* at
* which the option resides and the name *optname* of the option
* must be specified, see **getsockopt(2)**
for more information.
* The retrieved value is stored in the structure pointed by
* *opval* and of length *optlen*.
*
* *bpf_socket* should be one of the following:
*
* * **struct bpf_sock_ops** for **BPF_PROG_TYPE_SOCKET_OPS**.
* * **struct bpf_sock_addr** for **BPF_CGROUP_INET4_CONNECT**,

```

```

* **BPF_CGROUP_INET6_CONNECT** and **BPF_CGROUP_UNIX_CONNECT**.
*
* This helper actually implements a subset of **getsockopt()**.
* It supports the same set of *optname*s that is supported by
* the **bpf_setsockopt**() helper. The exceptions are
* **TCP_BPF_*** is **bpf_setsockopt**() only and
* **TCP_SAVED_SYN** is **bpf_getsockopt**() only.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_getsockopt)(void *bpf_socket, int level, int optname, void *optval, int optlen) = (void *) 57;

/*
* bpf_override_return
*
* Used for error injection, this helper uses kprobes to override
* the return value of the probed
function, and to set it to *rc*.
* The first argument is the context *regs* on which the kprobe
* works.
*
* This helper works by setting the PC (program counter)
* to an override function which is run in place of the original
* probed function. This means the probed function is not run at
* all. The replacement function just returns with the required
* value.
*
* This helper has security implications, and thus is subject to
* restrictions. It is only available if the kernel was compiled
* with the **CONFIG_BPF_KPROBE_OVERRIDE** configuration
* option, and in this case it only works on functions tagged with
* **ALLOW_ERROR_INJECTION** in the kernel code.
*
* Also, the helper is only available for the architectures having
* the CONFIG_FUNCTION_ERROR_INJECTION option. As of this writing,
* x86 architecture is the only one to support this feature.
*
* Returns
* 0
*/
static long (* const bpf_override_return)(struct pt_regs *regs, __u64 rc) = (void
*) 58;

/*
* bpf_sock_ops_cb_flags_set
*
* Attempt to set the value of the **bpf_sock_ops_cb_flags** field

```



```

* for the full TCP socket associated to *bpf_sock_ops* to
* *argval*.
*
* The primary use of this field is to determine if there should
* be calls to eBPF programs of type
* **BPF_PROG_TYPE_SOCKET_OPS** at various points in the TCP
* code. A program of the same type can change its value, per
* connection and as necessary, when the connection is
* established. This field is directly accessible for reading, but
* this helper must be used for updates in order to return an
* error if an eBPF program tries to set a callback that is not
* supported in the current kernel.
*
* *argval* is a flag array which can combine these flags:
*
* ***BPF_SOCKET_OPS_RTO_CB_FLAG** (retransmission time out)
* ***BPF_SOCKET_OPS_RETRANS_CB_FLAG** (retransmission)
* ***BPF_SOCKET_OPS_STATE_CB_FLAG** (TCP state change)
* ***BPF_SOCKET_OPS_RTT_CB_FLAG** (every RTT)
*
* Therefore,
* this function can be used to clear a callback flag by
* setting the appropriate bit to zero. e.g. to disable the RTO
* callback:
*
* **bpf_sock_ops_cb_flags_set(bpf_sock,**
* **bpf_sock->bpf_sock_ops_cb_flags & ~BPF_SOCKET_OPS_RTO_CB_FLAG)**
*
* Here are some examples of where one could call such eBPF
* program:
*
* * When RTO fires.
* * When a packet is retransmitted.
* * When the connection terminates.
* * When a packet is sent.
* * When a packet is received.
*
* Returns
* Code **-EINVAL** if the socket is not a full TCP socket;
* otherwise, a positive number containing the bits that could not
* be set is returned (which comes down to 0 if all bits were set
* as required).
*/
static long (* const bpf_sock_ops_cb_flags_set)(struct bpf_sock_ops *bpf_sock, int argval) = (void *) 59;

/*
* bpf_msg_redirect_map
*

```

```

* This helper is used in programs implementing policies at the
* socket level. If the message *msg* is allowed to pass (i.e. if
* the verdict eBPF program returns **SK_PASS**), redirect it to
* the socket referenced by *map* (of type
* **BPF_MAP_TYPE_SOCKMAP**) at index *key*. Both ingress and
* egress interfaces can be used for redirection. The
* **BPF_F_INGRESS** value in *flags* is used to make the
* distinction (ingress path is selected if the flag is present,
* egress path otherwise). This is the only flag supported for now.
*
* Returns
* **SK_PASS** on success, or **SK_DROP** on error.
*/
static long (* const bpf_msg_redirect_map)(struct sk_msg_md *msg, void *map, __u32 key, __u64 flags) = (void *)
60;

/*
* bpf_msg_apply_bytes
*
* For socket policies, apply the verdict of the eBPF program to
* the next *bytes* (number of bytes) of message *msg*.
*
* For example, this helper can be used in the following cases:
*
* * A single **sendmsg**() or **sendfile**() system call
* contains multiple logical messages that the eBPF program is
* supposed to read and for which
it should apply a verdict.
* * An eBPF program only cares to read the first *bytes* of a
* *msg*. If the message has a large payload, then setting up
* and calling the eBPF program repeatedly for all bytes, even
* though the verdict is already known, would create unnecessary
* overhead.
*
* When called from within an eBPF program, the helper sets a
* counter internal to the BPF infrastructure, that is used to
* apply the last verdict to the next *bytes*. If *bytes* is
* smaller than the current data being processed from a
* **sendmsg**() or **sendfile**() system call, the first
* *bytes* will be sent and the eBPF program will be re-run with
* the pointer for start of data pointing to byte number *bytes*
* **+ 1**. If *bytes* is larger than the current data being
* processed, then the eBPF verdict will be applied to multiple
* **sendmsg**() or **sendfile**() calls until *bytes* are
* consumed.
*
* Note that if a socket closes with the

```

```

internal counter holding
* a non-zero value, this is not a problem because data is not
* being buffered for *bytes* and is sent as it is received.
*
* Returns
* 0
*/
static long (* const bpf_msg_apply_bytes)(struct sk_msg_md *msg, __u32 bytes) = (void *) 61;

/*
* bpf_msg_cork_bytes
*
* For socket policies, prevent the execution of the verdict eBPF
* program for message *msg* until *bytes* (byte number) have been
* accumulated.
*
* This can be used when one needs a specific number of bytes
* before a verdict can be assigned, even if the data spans
* multiple **sendmsg**\ () or **sendfile**\ () calls. The extreme
* case would be a user calling **sendmsg**\ () repeatedly with
* 1-byte long message segments. Obviously, this is bad for
* performance, but it is still valid. If the eBPF program needs
* *bytes* bytes to validate a header, this helper can be used to
* prevent the eBPF program to be called again until *bytes* have
* been accumulated.
*
* Returns
* 0
*/
static long (* const bpf_msg_cork_bytes)(struct sk_msg_md *msg, __u32 bytes) = (void *) 62;

/*
* bpf_msg_pull_data
*
* For socket policies, pull in non-linear data from user space
* for *msg* and set pointers *msg* **->data** and *msg* \
* **->data_end** to *start* and *end* bytes offsets into *msg*,
* respectively.
*
* If a program of type **BPF_PROG_TYPE_SK_MSG** is run on a
* *msg* it can only parse data that the (**data**, **data_end**)
* pointers have already consumed. For **sendmsg**\ () hooks this
* is likely the first scatterlist element. But for calls relying
* on the **sendpage** handler (e.g. **sendfile**\ ()) this will
* be the range (**0**, **0**) because the data is shared with
* user space and by default the objective is to avoid allowing
* user space to modify data while (or after) eBPF verdict is
* being decided. This helper can be used to pull in data and to

```

- * set the start and end pointer to given values. Data will be
- * copied if necessary (i.e. if data was not linear and if start
- * and end pointers do not point to the same chunk).
- *
- * A call to this helper is susceptible to change the underlying
- * packet buffer. Therefore, at load time, all checks on pointers
- * previously done by the verifier are invalidated and must be
- * performed again, if the helper is used in combination with
- * direct packet access.
- *
- * All values for **flags** are reserved for future usage, and must
- * be left at zero.
- *
- * Returns
- * 0 on success, or a negative error in case of failure.
- */

```
static long (* const bpf_msg_pull_data)(struct sk_msg_md *msg, __u32 start, __u32 end, __u64 flags) = (void *) 63;
```



```
/*
 * bpf_bind
 *
```

- * Bind the socket associated to **ctx** to the address pointed by
- * **addr**, of length **addr_len**. This allows for making outgoing
- * connection from the desired IP address, which can be useful for
- * example when all processes inside a cgroup should use one
- * single IP address on a host
- that has multiple IP configured.
- *
- * This helper works for IPv4 and IPv6, TCP and UDP sockets. The
- * domain (**addr**\ ***->sa_family***) must be ***AF_INET*** (or
- * ***AF_INET6***). It's advised to pass zero port (***sin_port***
- * or ***sin6_port***) which triggers IP_BIND_ADDRESS_NO_PORT-like
- * behavior and lets the kernel efficiently pick up an unused
- * port as long as 4-tuple is unique. Passing non-zero port might
- * lead to degraded performance.
- *
- * Returns
- * 0 on success, or a negative error in case of failure.
- */

```
static long (* const bpf_bind)(struct bpf_sock_addr *ctx, struct sockaddr *addr, int addr_len) = (void *) 64;
```



```
/*
 * bpf_xdp_adjust_tail
 *
```

- * Adjust (move) **xdp_md**\ ***->data_end*** by **delta** bytes. It is
- * possible to both shrink and grow the packet tail.
- * Shrink done via **delta** being a negative integer.

```

*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously
done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_xdp_adjust_tail)(struct xdp_md *xdp_md, int delta) = (void *) 65;

/*
* bpf_skb_get_xfrm_state
*
* Retrieve the XFRM state (IP transform framework, see also
* **ip-xfrm(8)**) at *index* in XFRM "security path" for *skb*.
*
* The retrieved value is stored in the **struct bpf_xfrm_state**
* pointed by *xfrm_state* and of length *size*.
*
* All values for *flags* are reserved for future usage, and must
* be left at zero.
*
* This helper is available only if the kernel was compiled with
* **CONFIG_XFRM** configuration option.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_get_xfrm_state)(struct __sk_buff *skb, __u32 index, struct bpf_xfrm_state *xfrm_state,
__u32 size, __u64
flags) = (void *) 66;

/*
* bpf_get_stack
*
* Return a user or a kernel stack in bpf program provided buffer.
* To achieve this, the helper needs *ctx*, which is a pointer
* to the context on which the tracing program is executed.
* To store the stacktrace, the bpf program provides *buf* with
* a nonnegative *size*.
*
* The last argument, *flags*, holds the number of stack frames to
* skip (from 0 to 255), masked with
* **BPF_F_SKIP_FIELD_MASK**. The next bits can be used to set
* the following flags:

```

```

*
* **BPF_F_USER_STACK**
* Collect a user space stack instead of a kernel stack.
* **BPF_F_USER_BUILD_ID**
* Collect (build_id, file_offset) instead of ip for user
* stack, only valid if **BPF_F_USER_STACK** is also
* specified.
*
* *file_offset* is an offset relative to the beginning
* of the executable or shared object file backing the vma
* which the *ip* falls in. It is *not* an offset relative
* to that object's base address. Accordingly,
it must be
* adjusted by adding (sh_addr - sh_offset), where
* sh_{addr,offset} correspond to the executable section
* containing *file_offset* in the object, for comparisons
* to symbols' st_value to be valid.
*
* **bpf_get_stack**\ () can collect up to
* **PERF_MAX_STACK_DEPTH** both kernel and user frames, subject
* to sufficient large buffer size. Note that
* this limit can be controlled with the **sysctl** program, and
* that it should be manually increased in order to profile long
* user stacks (such as stacks for Java programs). To do so, use:
*
* ::
*
* # sysctl kernel.perf_event_max_stack=<new value>
*
* Returns
* The non-negative copied *buf* length equal to or less than
* *size* on success, or a negative error in case of failure.
*/
static long (* const bpf_get_stack)(void *ctx, void *buf, __u32 size, __u64 flags) = (void *) 67;

/*
* bpf_skb_load_bytes_relative
*
* This helper is similar to **bpf_skb_load_bytes**\ () in that
* it provides an easy way to load *len* bytes from *offset*
* from the packet associated to *skb*, into the buffer pointed
* by *to*. The difference to **bpf_skb_load_bytes**\ () is that
* a fifth argument *start_header* exists in order to select a
* base offset to start from. *start_header* can be one of:
*
* **BPF_HDR_START_MAC**
* Base offset to load data from is *skb*'s mac header.
* **BPF_HDR_START_NET**

```

```

* Base offset to load data from is *skb*'s network header.
*
* In general, "direct packet access" is the preferred method to
* access packet data, however, this helper is in particular useful
* in socket filters where *skb* \ **->data** does not always point
* to the start of the mac header and where "direct packet access"
* is not available.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/

static long (* const bpf_skb_load_bytes_relative)(const void *skb, __u32 offset, void *to, __u32 len, __u32
start_header) = (void *) 68;

/*
* bpf_fib_lookup
*
* Do FIB lookup in kernel tables using parameters in *params*.
* If lookup is successful and result shows packet is to be
* forwarded, the neighbor tables are searched for the nexthop.
* If successful (ie., FIB lookup shows forwarding and nexthop
* is resolved), the nexthop address is returned in ipv4_dst
* or ipv6_dst based on family, smac is set to mac address of
* egress device, dmac is set to nexthop mac address, rt_metric
* is set to metric from route (IPv4/IPv6 only), and ifindex
* is set to the device index of the nexthop from the FIB lookup.
*
* *plen* argument is the size of the passed in struct.
* *flags* argument can be a combination of one or more of the
* following values:
*
* **BPF_FIB_LOOKUP_DIRECT**
* Do a direct table lookup vs full lookup using FIB
* rules.
* **BPF_FIB_LOOKUP_TBID**
* Used with BPF_FIB_LOOKUP_DIRECT.
* Use the routing table ID present in *params*->tbid
* for the fib lookup.
* **BPF_FIB_LOOKUP_OUTPUT**
* Perform lookup from an egress perspective (default is
* ingress).
* **BPF_FIB_LOOKUP_SKIP_NEIGH**
* Skip the neighbour table lookup. *params*->dmac
* and *params*->smac will not be set as output. A common
* use case is to call **bpf_redirect_neigh**() after
* doing **bpf_fib_lookup**().
* **BPF_FIB_LOOKUP_SRC**
* Derive and set source IP addr in *params*->ipv{4,6}_src

```

```

* for the nexthop. If the src addr cannot be derived,
* **BPF_FIB_LKUP_RET_NO_SRC_ADDR** is returned. In this
* case, *params*->dmac and *params*->smac are not set either.
*
* *ctx* is either **struct xdp_md** for XDP programs or
* **struct sk_buff** tc cls_act programs.
*
* Returns
* * < 0 if any input argument is invalid
* * 0 on success (packet is forwarded, nexthop neighbor exists)
* * > 0 one of **BPF_FIB_LKUP_RET_** codes explaining why the
* packet is not forwarded or needs assist from full stack
*
* If lookup fails with BPF_FIB_LKUP_RET_FRAG_NEEDED,
* then the MTU
* was exceeded and output params->mtu_result contains the MTU.
*/
static long (* const bpf_fib_lookup)(void *ctx, struct bpf_fib_lookup *params, int plen, __u32 flags) = (void *) 69;

/*
* bpf_sock_hash_update
*
* Add an entry to, or update a sockhash *map* referencing sockets.
* The *skops* is used as a new value for the entry associated to
* *key*. *flags* is one of:
*
* **BPF_NOEXIST**
* The entry for *key* must not exist in the map.
* **BPF_EXIST**
* The entry for *key* must already exist in the map.
* **BPF_ANY**
* No condition on the existence of the entry for *key*.
*
* If the *map* has eBPF programs (parser and verdict), those will
* be inherited by the socket being added. If the socket is
* already attached to eBPF programs, this results in an error.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_sock_hash_update)(struct bpf_sock_ops *skops, void *map, void
*key, __u64 flags) = (void *) 70;

/*
* bpf_msg_redirect_hash
*
* This helper is used in programs implementing policies at the
* socket level. If the message *msg* is allowed to pass (i.e. if

```



```

* the verdict eBPF program returns **SK_PASS**), redirect it to
* the socket referenced by *map* (of type
* **BPF_MAP_TYPE_SOCKHASH**) using hash *key*. Both ingress and
* egress interfaces can be used for redirection. The
* **BPF_F_INGRESS** value in *flags* is used to make the
* distinction (ingress path is selected if the flag is present,
* egress path otherwise). This is the only flag supported for now.
*
* Returns
* **SK_PASS** on success, or **SK_DROP** on error.
*/
static long (* const bpf_msg_redirect_hash)(struct sk_msg_md *msg, void *map, void *key, __u64 flags) = (void *)
71;

/*
* bpf_sk_redirect_hash
*
* This helper is used in programs implementing policies at the
* skb socket level. If the sk_buff *skb* is allowed to pass (i.e.
* if the verdict eBPF
program returns **SK_PASS**), redirect it
* to the socket referenced by *map* (of type
* **BPF_MAP_TYPE_SOCKHASH**) using hash *key*. Both ingress and
* egress interfaces can be used for redirection. The
* **BPF_F_INGRESS** value in *flags* is used to make the
* distinction (ingress path is selected if the flag is present,
* egress otherwise). This is the only flag supported for now.
*
* Returns
* **SK_PASS** on success, or **SK_DROP** on error.
*/
static long (* const bpf_sk_redirect_hash)(struct __sk_buff *skb, void *map, void *key, __u64 flags) = (void *) 72;

/*
* bpf_lwt_push_encap
*
* Encapsulate the packet associated to *skb* within a Layer 3
* protocol header. This header is provided in the buffer at
* address *hdr*, with *len* its size in bytes. *type* indicates
* the protocol of the header and can be one of:
*
* **BPF_LWT_ENCAP_SEG6**
* IPv6 encapsulation with Segment Routing Header
* (**struct ipv6_sr_hdr**). *hdr* only contains the SRH,
* the IPv6 header is computed by the kernel.
* **BPF_LWT_ENCAP_SEG6_INLINE**
* Only works if *skb* contains an IPv6 packet. Insert a
* Segment Routing Header (**struct ipv6_sr_hdr**) inside

```

```

* the IPv6 header.
* **BPF_LWT_ENCAP_IP**
* IP encapsulation (GRE/GUE/IPIP/etc). The outer header
* must be IPv4 or IPv6, followed by zero or more
* additional headers, up to **LWT_BPF_MAX_HEADROOM**
* total bytes in all prepended headers. Please note that
* if **skb_is_gso** (*skb*) is true, no more than two
* headers can be prepended, and the inner header, if
* present, should be either GRE or UDP/GUE.
*
* **BPF_LWT_ENCAP_SEG6**\ types can be called by BPF programs
* of type **BPF_PROG_TYPE_LWT_IN** ; **BPF_LWT_ENCAP_IP** type can
* be called by bpf programs of types **BPF_PROG_TYPE_LWT_IN** and
* **BPF_PROG_TYPE_LWT_XMIT** .
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on
pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_lwt_push_encap)(struct __sk_buff *skb, __u32 type, void *hdr, __u32 len) = (void *) 73;

/*
* bpf_lwt_seg6_store_bytes
*
* Store *len* bytes from address *from* into the packet
* associated to *skb*, at *offset*. Only the flags, tag and TLVs
* inside the outermost IPv6 Segment Routing Header can be
* modified through this helper.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_lwt_seg6_store_bytes)(struct
__sk_buff *skb, __u32 offset, const void *from, __u32 len) = (void *) 74;

/*

```

```

* bpf_lwt_seg6_adjust_srh
*
* Adjust the size allocated to TLVs in the outermost IPv6
* Segment Routing Header contained in the packet associated to
* *skb*, at position *offset* by *delta* bytes. Only offsets
* after the segments are accepted. *delta* can be as well
* positive (growing) as negative (shrinking).
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_lwt_seg6_adjust_srh)(struct __sk_buff *skb, __u32 offset, __s32 delta) = (void *) 75;

/*
* bpf_lwt_seg6_action
*
* Apply an IPv6 Segment Routing action of type *action* to the
* packet associated to *skb*. Each action takes a parameter
* contained at address *param*, and of length *param_len* bytes.
* *action* can be one of:
*
* **SEG6_LOCAL_ACTION_END_X**
* End.X action: Endpoint with Layer-3 cross-connect.
* Type of *param*: **struct in6_addr**.
* **SEG6_LOCAL_ACTION_END_T**
* End.T action: Endpoint with specific IPv6 table lookup.
* Type of *param*: **int**.
* **SEG6_LOCAL_ACTION_END_B6**
* End.B6 action: Endpoint bound to an SRv6 policy.
* Type of *param*: **struct ipv6_sr_hdr**.
* **SEG6_LOCAL_ACTION_END_B6_ENCAP**
* End.B6.Encap action: Endpoint bound to an SRv6
* encapsulation policy.
* Type of *param*: **struct ipv6_sr_hdr**.
*
* A call to this helper is susceptible to change the underlying
* packet buffer. Therefore, at load time, all checks on pointers
* previously done by the verifier are invalidated and must be
* performed again, if the helper is used in combination with
* direct packet access.
*
* Returns

```

```

* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_lwt_seg6_action)(struct __sk_buff *skb, __u32 action, void *param, __u32 param_len) =
(void *) 76;

/*
* bpf_rc_repeat
*
* This helper is used in programs implementing IR decoding, to
* report a successfully decoded repeat key message. This delays
* the generation of a key up event for previously generated
* key down event.
*
* Some IR protocols like NEC have a special IR message for
* repeating last button, for when a button is held down.
*
* The *ctx* should point to the lirc sample as passed into
* the program.
*
* This helper is only available is the kernel was compiled with
* the **CONFIG_BPF_LIRC_MODE2** configuration option set to
* "y".
*
* Returns
* 0
*/
static long (* const bpf_rc_repeat)(void *ctx) = (void *) 77;

/*
* bpf_rc_keydown
*
* This helper is used in programs implementing IR decoding, to
* report a successfully decoded key
press with *scancode*,
* *toggle* value in the given *protocol*. The scancode will be
* translated to a keycode using the rc keymap, and reported as
* an input key down event. After a period a key up event is
* generated. This period can be extended by calling either
* **bpf_rc_keydown**\ () again with the same values, or calling
* **bpf_rc_repeat**\ ().
*
* Some protocols include a toggle bit, in case the button was
* released and pressed again between consecutive scancodes.
*
* The *ctx* should point to the lirc sample as passed into
* the program.
*
* The *protocol* is the decoded protocol number (see

```

```

* **enum rc_proto** for some predefined values).
*
* This helper is only available if the kernel was compiled with
* the **CONFIG_BPF_LIRC_MODE2** configuration option set to
* "y".
*
* Returns
* 0
*/
static long (* const bpf_rc_keydown)(void *ctx, __u32 protocol, __u64 scancode, __u32 toggle) = (void *) 78;

/*
* bpf_skb_cgroup_id
*
* Return
* the cgroup v2 id of the socket associated with the *skb*.
* This is roughly similar to the **bpf_get_cgroup_classid**()
* helper for cgroup v1 by providing a tag resp. identifier that
* can be matched on or used for map lookups e.g. to implement
* policy. The cgroup v2 id of a given path in the hierarchy is
* exposed in user space through the f_handle API in order to get
* to the same 64-bit id.
*
* This helper can be used on TC egress path, but not on ingress,
* and is available only if the kernel was compiled with the
* **CONFIG_SOCK_CGROUP_DATA** configuration option.
*
* Returns
* The id is returned or 0 in case the id could not be retrieved.
*/
static __u64 (* const bpf_skb_cgroup_id)(struct __sk_buff *skb) = (void *) 79;

/*
* bpf_get_current_cgroup_id
*
* Get the current cgroup id based on the cgroup within which
* the current task is running.
*
* Returns
* A 64-bit integer containing the current cgroup id based
* on the cgroup within which
* the current task is running.
*/
static __u64 (* const bpf_get_current_cgroup_id)(void) = (void *) 80;

/*
* bpf_get_local_storage
*

```

- * Get the pointer to the local storage area.
- * The type and the size of the local storage is defined
- * by the `*map*` argument.
- * The `*flags*` meaning is specific for each map type,
- * and has to be 0 for cgroup local storage.
- *
- * Depending on the BPF program type, a local storage area
- * can be shared between multiple instances of the BPF program,
- * running simultaneously.
- *
- * A user should care about the synchronization by himself.
- * For example, by using the `**BPF_ATOMIC**` instructions to alter
- * the shared data.
- *
- * Returns
- * A pointer to the local storage area.
- */

```
static void>(* const bpf_get_local_storage)(void *map, __u64 flags) = (void *) 81;
```



```
/*
 * bpf_sk_select_reuseport
 *
 * Select a **SO_REUSEPORT** socket from a
 * **BPF_MAP_TYPE_REUSEPORT_SOCKARRAY** *map*.
 * It checks the selected socket is matching the incoming
 * request in the socket buffer.
 *
 * Returns
 * 0 on success, or a negative error in case of failure.
 */
```

```
static long (* const bpf_sk_select_reuseport)(struct sk_reuseport_md *reuse, void *map, void *key, __u64 flags) =
(void *) 82;
```



```
/*
 * bpf_skb_ancestor_cgroup_id
 *
 * Return id of cgroup v2 that is ancestor of cgroup associated
 * with the *skb* at the *ancestor_level*. The root cgroup is at
 * *ancestor_level* zero and each step down the hierarchy
 * increments the level. If *ancestor_level* == level of cgroup
 * associated with *skb*, then return value will be same as that
 * of **bpf_skb_cgroup_id** ().
 *
 * The helper is useful to implement policies based on cgroups
 * that are upper in hierarchy than immediate cgroup associated
 * with *skb*.
 *
 * The format of returned id and helper limitations are same as in
```

```

* **bpf_skb_cgroup_id**\ ().
*
* Returns
* The id is returned or 0 in case the id could not be retrieved.
*/

static __u64 (* const bpf_skb_ancestor_cgroup_id)(struct
__sk_buff *skb, int ancestor_level) = (void *) 83;

/*
* bpf_sk_lookup_tcp
*
* Look for TCP socket matching *tuple*, optionally in a child
* network namespace *netns*. The return value must be checked,
* and if non-**NULL**, released via **bpf_sk_release**\ ().
*
* The *ctx* should point to the context of the program, such as
* the skb or socket (depending on the hook in use). This is used
* to determine the base network namespace for the lookup.
*
* *tuple_size* must be one of:
*
* **sizeof**\ (*tuple* **->ipv4**)
* Look for an IPv4 socket.
* **sizeof**\ (*tuple* **->ipv6**)
* Look for an IPv6 socket.
*
* If the *netns* is a negative signed 32-bit integer, then the
* socket lookup table in the netns associated with the *ctx*
* will be used. For the TC hooks, this is the netns of the device
* in the skb. For socket hooks, this is the netns of the socket.
* If *netns* is any other signed 32-bit value greater than or
* equal to zero then it specifies the ID of the netns relative to
* the netns associated with the *ctx*. *netns* values beyond the
* range of 32-bit integers are reserved for future use.
*
* All values for *flags* are reserved for future usage, and must
* be left at zero.
*
* This helper is available only if the kernel was compiled with
* **CONFIG_NET** configuration option.
*
* Returns
* Pointer to **struct bpf_sock**, or **NULL** in case of failure.
* For sockets with reuseport option, the **struct bpf_sock**
* result is from *reuse* **->socks**\ [] using the hash of the
* tuple.
*/

static struct bpf_sock (* const bpf_sk_lookup_tcp)(void *ctx, struct bpf_sock_tuple *tuple, __u32 tuple_size,

```

```

__u64 netns, __u64 flags) = (void *) 84;

/*
 * bpf_sk_lookup_udp
 *
 * Look for UDP socket matching *tuple*, optionally in a child
 * network namespace *netns*. The return value must be checked,
 * and if non-**NULL**, released via **bpf_sk_release**\ ().
 *
 * The
 * *ctx* should point to the context of the program, such as
 * the skb or socket (depending on the hook in use). This is used
 * to determine the base network namespace for the lookup.
 *
 * *tuple_size* must be one of:
 *
 * **sizeof**\ (*tuple*\ **->ipv4**)
 * Look for an IPv4 socket.
 * **sizeof**\ (*tuple*\ **->ipv6**)
 * Look for an IPv6 socket.
 *
 * If the *netns* is a negative signed 32-bit integer, then the
 * socket lookup table in the netns associated with the *ctx*
 * will be used. For the TC hooks, this is the netns of the device
 * in the skb. For socket hooks, this is the netns of the socket.
 * If *netns* is any other signed 32-bit value greater than or
 * equal to zero then it specifies the ID of the netns relative to
 * the netns associated with the *ctx*. *netns* values beyond the
 * range of 32-bit integers are reserved for future use.
 *
 * All values for *flags* are reserved for future usage, and must
 * be left at zero.
 *
 * This helper is available
 * only if the kernel was compiled with
 * **CONFIG_NET** configuration option.
 *
 * Returns
 * Pointer to **struct bpf_sock**, or **NULL** in case of failure.
 * For sockets with reuseport option, the **struct bpf_sock**
 * result is from *reuse*\ **->socks**\ [] using the hash of the
 * tuple.
 */
static struct bpf_sock *(* const bpf_sk_lookup_udp)(void *ctx, struct bpf_sock_tuple *tuple, __u32 tuple_size,
__u64 netns, __u64 flags) = (void *) 85;

/*
 * bpf_sk_release

```



```

*
* Release the reference held by *sock*. *sock* must be a
* non-**NULL** pointer that was returned from
* **bpf_sk_lookup_xxx**().
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_sk_release)(void *sock) = (void *) 86;

/*
* bpf_map_push_elem
*
* Push an element *value* in *map*. *flags* is one of:
*
* **BPF_EXIST**
* If the queue/stack is full, the oldest element is
* removed to make room for this.
*
* Returns
* 0 on success, or
* a negative error in case of failure.
*/
static long (* const bpf_map_push_elem)(void *map, const void *value, __u64 flags) = (void *) 87;

/*
* bpf_map_pop_elem
*
* Pop an element from *map*.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_map_pop_elem)(void *map, void *value) = (void *) 88;

/*
* bpf_map_peek_elem
*
* Get an element from *map* without removing it.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_map_peek_elem)(void *map, void *value) = (void *) 89;

/*
* bpf_msg_push_data
*

```

```

* For socket policies, insert *len* bytes into *msg* at offset
* *start*.
*
* If a program of type **BPF_PROG_TYPE_SK_MSG** is run on a
* *msg* it may want to insert metadata or options into the *msg*.
* This can later be read and used by any of the lower layer BPF
* hooks.
*
* This helper may fail if under memory pressure (a malloc
* fails) in these cases BPF programs
will get an appropriate
* error and BPF programs will need to handle them.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_msg_push_data)(struct sk_msg_md *msg, __u32 start, __u32 len, __u64 flags) = (void *) 90;

/*
* bpf_msg_pop_data
*
* Will remove *len* bytes from a *msg* starting at byte *start*.
* This may result in **ENOMEM** errors under certain situations if
* an allocation and copy are required due to a full ring buffer.
* However, the helper will try to avoid doing the allocation
* if possible. Other errors can occur if input parameters are
* invalid either due to *start* byte not being valid part of *msg*
* payload and/or *pop* value being to large.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_msg_pop_data)(struct sk_msg_md *msg, __u32 start, __u32 len, __u64 flags) = (void *) 91;

/*
* bpf_rc_pointer_rel
*
* This helper is used in programs implementing
IR decoding, to
* report a successfully decoded pointer movement.
*
* The *ctx* should point to the lirc sample as passed into
* the program.
*
* This helper is only available is the kernel was compiled with
* the **CONFIG_BPF_LIRC_MODE2** configuration option set to
* **y**".

```

```

*
* Returns
* 0
*/
static long (* const bpf_rc_pointer_rel)(void *ctx, __s32 rel_x, __s32 rel_y) = (void *) 92;

/*
* bpf_spin_lock
*
* Acquire a spinlock represented by the pointer *lock*, which is
* stored as part of a value of a map. Taking the lock allows to
* safely update the rest of the fields in that value. The
* spinlock can (and must) later be released with a call to
* **bpf_spin_unlock**\ (\ *lock*\ ).
*
* Spinlocks in BPF programs come with a number of restrictions
* and constraints:
*
* * **bpf_spin_lock** objects are only allowed inside maps of
* types **BPF_MAP_TYPE_HASH** and **BPF_MAP_TYPE_ARRAY** (this
* list could be extended in the future).
* * BTF description of the map is mandatory.
* * The BPF program can take ONE lock at a time, since taking two
* or more could cause dead locks.
* * Only one **struct bpf_spin_lock** is allowed per map element.
* * When the lock is taken, calls (either BPF to BPF or helpers)
* are not allowed.
* * The **BPF_LD_ABS** and **BPF_LD_IND** instructions are not
* allowed inside a spinlock-ed region.
* * The BPF program MUST call **bpf_spin_unlock**\ () to release
* the lock, on all execution paths, before it returns.
* * The BPF program can access **struct bpf_spin_lock** only via
* the **bpf_spin_lock**\ () and **bpf_spin_unlock**\ ()
* helpers. Loading or storing data into the **struct
* bpf_spin_lock** *lock*\ **,** field of a map is not allowed.
* * To use the **bpf_spin_lock**\ () helper, the BTF description
* of the map value must be a struct and have **struct
* bpf_spin_lock** *anyname*\ **,** field at the top level.
* Nested
lock inside another struct is not allowed.
* * The **struct bpf_spin_lock** *lock* field in a map value must
* be aligned on a multiple of 4 bytes in that value.
* * Syscall with command **BPF_MAP_LOOKUP_ELEM** does not copy
* the **bpf_spin_lock** field to user space.
* * Syscall with command **BPF_MAP_UPDATE_ELEM**, or update from
* a BPF program, do not update the **bpf_spin_lock** field.
* * **bpf_spin_lock** cannot be on the stack or inside a
* networking packet (it can only be inside of a map values).

```

```

* * **bpf_spin_lock** is available to root only.
* * Tracing programs and socket filter programs cannot use
* * **bpf_spin_lock**\ () due to insufficient preemption checks
* (but this may change in the future).
* * **bpf_spin_lock** is not allowed in inner maps of map-in-map.
*
* Returns
* 0
*/
static long (* const bpf_spin_lock)(struct bpf_spin_lock *lock) = (void *) 93;

/*
* bpf_spin_unlock
*
* Release the *lock* previously locked
by a call to
* **bpf_spin_lock**\ (\ *lock*\ ).
*
* Returns
* 0
*/
static long (* const bpf_spin_unlock)(struct bpf_spin_lock *lock) = (void *) 94;

/*
* bpf_sk_fullsock
*
* This helper gets a **struct bpf_sock** pointer such
* that all the fields in this **bpf_sock** can be accessed.
*
* Returns
* A **struct bpf_sock** pointer on success, or **NULL** in
* case of failure.
*/
static struct bpf_sock *(* const bpf_sk_fullsock)(struct bpf_sock *sk) = (void *) 95;

/*
* bpf_tcp_sock
*
* This helper gets a **struct bpf_tcp_sock** pointer from a
* **struct bpf_sock** pointer.
*
* Returns
* A **struct bpf_tcp_sock** pointer on success, or **NULL** in
* case of failure.
*/
static struct bpf_tcp_sock *(* const bpf_tcp_sock)(struct bpf_sock *sk) = (void *) 96;

/*

```

```

* bpf_skb_ecn_set_ce
*
* Set ECN (Explicit Congestion Notification) field of IP header
* to **CE** (Congestion Encountered) if current value is **ECT**
* (ECN Capable Transport). Otherwise, do nothing.
Works with IPv6
* and IPv4.
*
* Returns
* 1 if the **CE** flag is set (either by the current helper call
* or because it was already present), 0 if it is not set.
*/
static long (* const bpf_skb_ecn_set_ce)(struct __sk_buff *skb) = (void *) 97;

/*
* bpf_get_listener_sock
*
* Return a **struct bpf_sock** pointer in **TCP_LISTEN** state.
* **bpf_sk_release**\ () is unnecessary and not allowed.
*
* Returns
* A **struct bpf_sock** pointer on success, or **NULL** in
* case of failure.
*/
static struct bpf_sock *(* const bpf_get_listener_sock)(struct bpf_sock *sk) = (void *) 98;

/*
* bpf_skc_lookup_tcp
*
* Look for TCP socket matching *tuple*, optionally in a child
* network namespace *netns*. The return value must be checked,
* and if non-**NULL**, released via **bpf_sk_release**\ ().
*
* This function is identical to **bpf_sk_lookup_tcp**\ (), except
* that it also returns timewait or request sockets. Use
* **bpf_sk_fullsock**\ () or **bpf_tcp_sock**\
() to access the
* full structure.
*
* This helper is available only if the kernel was compiled with
* **CONFIG_NET** configuration option.
*
* Returns
* Pointer to **struct bpf_sock**, or **NULL** in case of failure.
* For sockets with reuseport option, the **struct bpf_sock**
* result is from *reuse*\ **->socks**\ [] using the hash of the
* tuple.
*/

```

```

static struct bpf_sock *(* const bpf_sk_lookup_tcp)(void *ctx, struct bpf_sock_tuple *tuple, __u32 tuple_size,
__u64 netns, __u64 flags) = (void *) 99;

/*
 * bpf_tcp_check_syncookie
 *
 * Check whether *iph* and *th* contain a valid SYN cookie ACK for
 * the listening socket in *sk*.
 *
 * *iph* points to the start of the IPv4 or IPv6 header, while
 * *iph_len* contains **sizeof**\ (**struct iphdr**) or
 * **sizeof**\ (**struct ipv6hdr**).
 *
 * *th* points to the start of the TCP header, while *th_len*
 * contains the length of the TCP header (at least
 * **sizeof**\ (**struct tcphdr**)).
 *
 * Returns
 * 0 if *iph* and *th* are a valid SYN cookie ACK, or a negative
 * error otherwise.
 */
static long (* const bpf_tcp_check_syncookie)(void *sk, void *iph, __u32 iph_len, struct tcphdr *th, __u32 th_len)
= (void *) 100;

/*
 * bpf_sysctl_get_name
 *
 * Get name of sysctl in /proc/sys/ and copy it into provided by
 * program buffer *buf* of size *buf_len*.
 *
 * The buffer is always NUL terminated, unless it's zero-sized.
 *
 * If *flags* is zero, full name (e.g. "net/ipv4/tcp_mem") is
 * copied. Use **BPF_F_SYSCTL_BASE_NAME** flag to copy base name
 * only (e.g. "tcp_mem").
 *
 * Returns
 * Number of character copied (not including the trailing NUL).
 *
 * **_E2BIG** if the buffer wasn't big enough (*buf* will contain
 * truncated name in this case).
 */
static long (* const bpf_sysctl_get_name)(struct bpf_sysctl *ctx, char *buf, unsigned long buf_len, __u64 flags) =
(void *) 101;

/*
 * bpf_sysctl_get_current_value
 *

```

```

* Get current value of sysctl as it is presented in
/proc/sys
* (incl. newline, etc), and copy it as a string into provided
* by program buffer *buf* of size *buf_len*.
*
* The whole value is copied, no matter what file position user
* space issued e.g. sys_read at.
*
* The buffer is always NUL terminated, unless it's zero-sized.
*
* Returns
* Number of character copied (not including the trailing NUL).
*
* **E2BIG** if the buffer wasn't big enough (*buf* will contain
* truncated name in this case).
*
* **EINVAL** if current value was unavailable, e.g. because
* sysctl is uninitialized and read returns -EIO for it.
*/
static long (* const bpf_sysctl_get_current_value)(struct bpf_sysctl *ctx, char *buf, unsigned long buf_len) = (void *) 102;

/*
* bpf_sysctl_get_new_value
*
* Get new value being written by user space to sysctl (before
* the actual write happens) and copy it as a string into
* provided by program buffer *buf* of size *buf_len*.
*
* User space may write new value at file position >
0.
*
* The buffer is always NUL terminated, unless it's zero-sized.
*
* Returns
* Number of character copied (not including the trailing NUL).
*
* **E2BIG** if the buffer wasn't big enough (*buf* will contain
* truncated name in this case).
*
* **EINVAL** if sysctl is being read.
*/
static long (* const bpf_sysctl_get_new_value)(struct bpf_sysctl *ctx, char *buf, unsigned long buf_len) = (void *) 103;

/*
* bpf_sysctl_set_new_value
*

```

```

* Override new value being written by user space to sysctl with
* value provided by program in buffer *buf* of size *buf_len*.
*
* *buf* should contain a string in same form as provided by user
* space on sysctl write.
*
* User space may write new value at file position > 0. To override
* the whole sysctl value file position should be set to zero.
*
* Returns
* 0 on success.
*
* **-E2BIG** if the *buf_len* is too big.
*
* **-EINVAL** if sysctl is being read.
*/
static long (* const bpf_sysctl_set_new_value)(struct
bpf_sysctl *ctx, const char *buf, unsigned long buf_len) = (void *) 104;

/*
* bpf_strtol
*
* Convert the initial part of the string from buffer *buf* of
* size *buf_len* to a long integer according to the given base
* and save the result in *res*.
*
* The string may begin with an arbitrary amount of white space
* (as determined by **isspace**\ (3)) followed by a single
* optional '**_**' sign.
*
* Five least significant bits of *flags* encode base, other bits
* are currently unused.
*
* Base must be either 8, 10, 16 or 0 to detect it automatically
* similar to user space **strtol**\ (3).
*
* Returns
* Number of characters consumed on success. Must be positive but
* no more than *buf_len*.
*
* **-EINVAL** if no valid digits were found or unsupported base
* was provided.
*
* **-ERANGE** if resulting value was out of range.
*/
static long (* const bpf_strtol)(const char *buf, unsigned long buf_len, __u64 flags, long *res) = (void *) 105;

/*

```



```

*
bpf_strtoul
*
* Convert the initial part of the string from buffer *buf* of
* size *buf_len* to an unsigned long integer according to the
* given base and save the result in *res*.
*
* The string may begin with an arbitrary amount of white space
* (as determined by isspace(3)).
*
* Five least significant bits of *flags* encode base, other bits
* are currently unused.
*
* Base must be either 8, 10, 16 or 0 to detect it automatically
* similar to user space strtoul(3).
*
* Returns
* Number of characters consumed on success. Must be positive but
* no more than *buf_len*.
*
* -EINVAL if no valid digits were found or unsupported base
* was provided.
*
* -ERANGE if resulting value was out of range.
*/
static long (* const bpf_strtoul)(const char *buf, unsigned long buf_len, __u64 flags, unsigned long *res) = (void *)
106;

/*
* bpf_sk_storage_get
*
* Get a bpf-local-storage from a *sk*.
*
* Logically, it could be thought of getting
the value from
* a *map* with *sk* as the key. From this
* perspective, the usage is not much different from
* bpf_map_lookup_elem(*map, &*sk*) except this
* helper enforces the key must be a full socket and the map must
* be a BPF_MAP_TYPE_SK_STORAGE also.
*
* Underneath, the value is stored locally at *sk* instead of
* the *map*. The *map* is used as the bpf-local-storage
* "type". The bpf-local-storage "type" (i.e. the *map*) is
* searched against all bpf-local-storages residing at *sk*.
*
* *sk* is a kernel struct sock pointer for LSM program.
* *sk* is a struct bpf_sock pointer for other program types.

```

```

*
* An optional *flags* (**BPF_SK_STORAGE_GET_F_CREATE**) can be
* used such that a new bpf-local-storage will be
* created if one does not exist. *value* can be used
* together with **BPF_SK_STORAGE_GET_F_CREATE** to specify
* the initial value of a bpf-local-storage. If *value* is
* **NULL**, the new bpf-local-storage
will be zero initialized.
*
* Returns
* A bpf-local-storage pointer is returned on success.
*
* **NULL** if not found or there was an error in adding
* a new bpf-local-storage.
*/
static void>(* const bpf_sk_storage_get)(void *map, void *sk, void *value, __u64 flags) = (void *) 107;

/*
* bpf_sk_storage_delete
*
* Delete a bpf-local-storage from a *sk*.
*
* Returns
* 0 on success.
*
* **ENOENT** if the bpf-local-storage cannot be found.
* **EINVAL** if sk is not a fullsock (e.g. a request_sock).
*/
static long(* const bpf_sk_storage_delete)(void *map, void *sk) = (void *) 108;

/*
* bpf_send_signal
*
* Send signal *sig* to the process of the current task.
* The signal may be delivered to any of this process's threads.
*
* Returns
* 0 on success or successfully queued.
*
* **EBUSY** if work queue under nmi is full.
*
* **EINVAL** if *sig* is invalid.
*
* **EPERM** if no permission to send the *sig*.
*
* **EAGAIN** if bpf
program can try again.
*/

```

```

static long (* const bpf_send_signal)(__u32 sig) = (void *) 109;

/*
 * bpf_tcp_gen_syncookie
 *
 * Try to issue a SYN cookie for the packet with corresponding
 * IP/TCP headers, *iph* and *th*, on the listening socket in *sk*.
 *
 * *iph* points to the start of the IPv4 or IPv6 header, while
 * *iph_len* contains **sizeof**\ (**struct iphdr**) or
 * **sizeof**\ (**struct ipv6hdr**).
 *
 * *th* points to the start of the TCP header, while *th_len*
 * contains the length of the TCP header with options (at least
 * **sizeof**\ (**struct tcphdr**)).
 *
 * Returns
 * On success, lower 32 bits hold the generated SYN cookie in
 * followed by 16 bits which hold the MSS value for that cookie,
 * and the top 16 bits are unused.
 *
 * On failure, the returned value is one of the following:
 *
 * **_EINVAL** SYN cookie cannot be issued due to error
 *
 * **_ENOENT** SYN cookie should not be issued (no SYN flood)
 *
 * **_EOPNOTSUPP** kernel configuration
    does not enable SYN cookies
 *
 * **_EPROTONOSUPPORT** IP packet version is not 4 or 6
 */
static __s64 (* const bpf_tcp_gen_syncookie)(void *sk, void *iph, __u32 iph_len, struct tcphdr *th, __u32 th_len) =
(void *) 110;

/*
 * bpf_skb_output
 *
 * Write raw *data* blob into a special BPF perf event held by
 * *map* of type **BPF_MAP_TYPE_PERF_EVENT_ARRAY**. This perf
 * event must have the following attributes: **PERF_SAMPLE_RAW**
 * as **sample_type**, **PERF_TYPE_SOFTWARE** as **type**, and
 * **PERF_COUNT_SW_BPF_OUTPUT** as **config**.
 *
 * The *flags* are used to indicate the index in *map* for which
 * the value must be put, masked with **BPF_F_INDEX_MASK**.
 * Alternatively, *flags* can be set to **BPF_F_CURRENT_CPU**
 * to indicate that the index of the current CPU core should be

```

```

* used.
*
* The value to write, of *size*, is passed through eBPF stack and
* pointed by *data*.
*
* *ctx* is a pointer to in-kernel struct sk_buff.
*
* This helper is similar to
**bpf_perf_event_output**\ () but
* restricted to raw_tracepoint bpf programs.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_skb_output)(void *ctx, void *map, __u64 flags, void *data, __u64 size) = (void *) 111;

/*
* bpf_probe_read_user
*
* Safely attempt to read *size* bytes from user space address
* *unsafe_ptr* and store the data in *dst*.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_probe_read_user)(void *dst, __u32 size, const void *unsafe_ptr) = (void *) 112;

/*
* bpf_probe_read_kernel
*
* Safely attempt to read *size* bytes from kernel space address
* *unsafe_ptr* and store the data in *dst*.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_probe_read_kernel)(void *dst, __u32 size, const void *unsafe_ptr) = (void *) 113;

/*
* bpf_probe_read_user_str
*
* Copy a NUL terminated string from an unsafe
user address
* *unsafe_ptr* to *dst*. The *size* should include the
* terminating NUL byte. In case the string length is smaller than
* *size*, the target is not padded with further NUL bytes. If the
* string length is larger than *size*, just *size*-1 bytes are
* copied and the last byte is set to NUL.

```

```

*
* On success, returns the number of bytes that were written,
* including the terminal NUL. This makes this helper useful in
* tracing programs for reading strings, and more importantly to
* get its length at runtime. See the following snippet:
*
* ::
*
* SEC("kprobe/sys_open")
* void bpf_sys_open(struct pt_regs *ctx)
* {
*     char buf[PATHLEN]; // PATHLEN is defined to 256
*     int res = bpf_probe_read_user_str(buf, sizeof(buf),
*                                     ctx->di);
*
*     // Consume buf, for example push it to
*     // userspace via bpf_perf_event_output(); we
*     // can use res (the string length) as event
*     // size,
*     after checking its boundaries.
* }
*
* In comparison, using bpf_probe_read_user() helper here
* instead to read the string would require to estimate the length
* at compile time, and would often result in copying more memory
* than necessary.
*
* Another useful use case is when parsing individual process
* arguments or individual environment variables navigating
* *current* **->mm->arg_start** and *current*
* **->mm->env_start**: using this helper and the return value,
* one can quickly iterate at the right offset of the memory area.
*
* Returns
* On success, the strictly positive length of the output string,
* including the trailing NUL character. On error, a negative
* value.
*/
static long (* const bpf_probe_read_user_str)(void *dst, __u32 size, const void *unsafe_ptr) = (void *) 114;

/*
* bpf_probe_read_kernel_str
*
* Copy a NUL terminated string from an unsafe kernel address *unsafe_ptr*
* to *dst*. Same semantics as with bpf_probe_read_user_str()
* apply.
*
* Returns

```

```

* On success, the strictly positive length of the string, including
* the trailing NUL character. On error, a negative value.
*/
static long (* const bpf_probe_read_kernel_str)(void *dst, __u32 size, const void *unsafe_ptr) = (void *) 115;

/*
* bpf_tcp_send_ack
*
* Send out a tcp-ack. *tp* is the in-kernel struct **tcp_sock**.
* *rcv_nxt* is the ack_seq to be sent out.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_tcp_send_ack)(void *tp, __u32 rcv_nxt) = (void *) 116;

/*
* bpf_send_signal_thread
*
* Send signal *sig* to the thread corresponding to the current task.
*
* Returns
* 0 on success or successfully queued.
*
* **EBUSY** if work queue under nmi is full.
*
* **EINVAL** if *sig* is invalid.
*
* **EPERM** if no permission to send the *sig*.
*
* **EAGAIN** if bpf program can try again.
*/
static long (* const bpf_send_signal_thread)(__u32 sig) = (void *) 117;

/*
* bpf_jiffies64
*
* Obtain the 64bit jiffies
*
* Returns
* The 64 bit jiffies
*/
static __u64 (* const bpf_jiffies64)(void) = (void *) 118;

/*
* bpf_read_branch_records
*

```

```

* For an eBPF program attached to a perf event, retrieve the
* branch records (**struct perf_branch_entry**) associated to *ctx*
* and store it in the buffer pointed by *buf* up to size
* *size* bytes.
*
* Returns
* On success, number of bytes written to *buf*. On error, a
* negative value.
*
* The *flags* can be set to **BPF_F_GET_BRANCH_RECORDS_SIZE** to
* instead return the number of bytes required to store all the
* branch entries. If this flag is set, *buf* may be NULL.
*
* **-EINVAL** if arguments invalid or **size** not a multiple
* of **sizeof** (**struct perf_branch_entry**).
*
* **-ENOENT** if architecture does not support branch records.
*/
static long (* const bpf_read_branch_records)(struct bpf_perf_event_data *ctx, void *buf, __u32 size, __u64
flags) = (void *) 119;

/*
* bpf_get_ns_current_pid_tgid
*
* Returns 0 on success, values for *pid* and *tgid* as seen from the current
* *namespace* will be returned in *nsdata*.
*
* Returns
* 0 on success, or one of the following in case of failure:
*
* **-EINVAL** if dev and inum supplied don't match dev_t and inode number
* with nsfs of current task, or if dev conversion to dev_t lost high bits.
*
* **-ENOENT** if pidns does not exists for the current task.
*/
static long (* const bpf_get_ns_current_pid_tgid)(__u64 dev, __u64 ino, struct bpf_pidns_info *nsdata, __u32 size)
= (void *) 120;

/*
* bpf_xdp_output
*
* Write raw *data* blob into a special BPF perf event held by
* *map* of type **BPF_MAP_TYPE_PERF_EVENT_ARRAY**. This perf
* event must have the following attributes: **PERF_SAMPLE_RAW**
* as **sample_type**, **PERF_TYPE_SOFTWARE** as **type**, and
* **PERF_COUNT_SW_BPF_OUTPUT** as **config**.
*
* The *flags* are used to indicate the index

```

```

in *map* for which
* the value must be put, masked with **BPF_F_INDEX_MASK**.
* Alternatively, *flags* can be set to **BPF_F_CURRENT_CPU**
* to indicate that the index of the current CPU core should be
* used.
*
* The value to write, of *size*, is passed through eBPF stack and
* pointed by *data*.
*
* *ctx* is a pointer to in-kernel struct xdp_buff.
*
* This helper is similar to **bpf_perf_eventoutput**\() but
* restricted to raw_tracepoint bpf programs.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_xdp_output)(void *ctx, void *map, __u64 flags, void *data, __u64 size) = (void *) 121;

/*
* bpf_get_netns_cookie
*
* Retrieve the cookie (generated by the kernel) of the network
* namespace the input *ctx* is associated with. The network
* namespace cookie remains stable for its lifetime and provides
* a global identifier that can be assumed unique. If *ctx* is
* NULL, then the helper returns the cookie
for the initial
* network namespace. The cookie itself is very similar to that
* of **bpf_get_socket_cookie**\() helper, but for network
* namespaces instead of sockets.
*
* Returns
* A 8-byte long opaque number.
*/
static __u64 (* const bpf_get_netns_cookie)(void *ctx) = (void *) 122;

/*
* bpf_get_current_ancestor_cgroup_id
*
* Return id of cgroup v2 that is ancestor of the cgroup associated
* with the current task at the *ancestor_level*. The root cgroup
* is at *ancestor_level* zero and each step down the hierarchy
* increments the level. If *ancestor_level* == level of cgroup
* associated with the current task, then return value will be the
* same as that of **bpf_get_current_cgroup_id**\().
*
* The helper is useful to implement policies based on cgroups

```



```

* that are upper in hierarchy than immediate cgroup associated
* with the current task.
*
* The format of returned id and helper limitations are same as in
* **bpf_get_current_cgroup_id**\
().
*
* Returns
* The id is returned or 0 in case the id could not be retrieved.
*/
static __u64 (* const bpf_get_current_ancestor_cgroup_id)(int ancestor_level) = (void *) 123;

/*
* bpf_sk_assign
*
* Helper is overloaded depending on BPF program type. This
* description applies to **BPF_PROG_TYPE_SCHED_CLS** and
* **BPF_PROG_TYPE_SCHED_ACT** programs.
*
* Assign the *sk* to the *skb*. When combined with appropriate
* routing configuration to receive the packet towards the socket,
* will cause *skb* to be delivered to the specified socket.
* Subsequent redirection of *skb* via **bpf_redirect**\ (),
* **bpf_clone_redirect**\ () or other methods outside of BPF may
* interfere with successful delivery to the socket.
*
* This operation is only valid from TC ingress path.
*
* The *flags* argument must be zero.
*
* Returns
* 0 on success, or a negative error in case of failure:
*
* **-EINVAL** if specified *flags* are not supported.
*
* **-ENOENT**
if the socket is unavailable for assignment.
*
* **-ENETUNREACH** if the socket is unreachable (wrong netns).
*
* **-EOPNOTSUPP** if the operation is not supported, for example
* a call from outside of TC ingress.
*/
static long (* const bpf_sk_assign)(void *ctx, void *sk, __u64 flags) = (void *) 124;

/*
* bpf_ktime_get_boot_ns
*

```

```

* Return the time elapsed since system boot, in nanoseconds.
* Does include the time the system was suspended.
* See: clock_gettime (CLOCK_BOOTTIME)
*
* Returns
* Current ktime.
*/
static __u64 (* const bpf_ktime_get_boot_ns)(void) = (void *) 125;

/*
* bpf_seq_printf
*
* bpf_seq_printf () uses seq_file seq_printf () to print
* out the format string.
* The m represents the seq_file. The fmt and fmt_size are for
* the format string itself. The data and data_len are format string
* arguments. The data are a u64 array and corresponding format string
* values are stored in the array. For strings
* and pointers where pointees
* are accessed, only the pointer values are stored in the data array.
* The data_len is the size of data in bytes - must be a multiple of 8.
*
* Formats %s, %p{i,I}{4,6} requires to read kernel memory.
* Reading kernel memory may fail due to either invalid address or
* valid address but requiring a major memory fault. If reading kernel memory
* fails, the string for %s will be an empty string, and the ip
* address for %p{i,I}{4,6} will be 0. Not returning error to
* bpf program is consistent with what bpf_trace_printk () does for now.
*
* Returns
* 0 on success, or a negative error in case of failure:
*
* -EBUSY if per-CPU memory copy buffer is busy, can try again
* by returning 1 from bpf program.
*
* -EINVAL if arguments are invalid, or if fmt is invalid/unsupported.
*
* -E2BIG if fmt contains too many format specifiers.
*
* -EOVERFLOW if an overflow happened: The same
* object will be tried again.
*/
static long (* const bpf_seq_printf)(struct seq_file *m, const char *fmt, __u32 fmt_size, const void *data, __u32
data_len) = (void *) 126;

/*
* bpf_seq_write
*

```

```

* **bpf_seq_write**\ () uses seq_file **seq_write**\ () to write the data.
* The *m* represents the seq_file. The *data* and *len* represent the
* data to write in bytes.
*
* Returns
* 0 on success, or a negative error in case of failure:
*
* **-EOVERFLOW** if an overflow happened: The same object will be tried again.
*/
static long (* const bpf_seq_write)(struct seq_file *m, const void *data, __u32 len) = (void *) 127;

/*
* bpf_sk_cgroup_id
*
* Return the cgroup v2 id of the socket *sk*.
*
* *sk* must be a non-NULL pointer to a socket, e.g. one
* returned from **bpf_sk_lookup_xxx**\ (),
* **bpf_sk_fullsock**\ (), etc. The format of returned id is
* same as in **bpf_skb_cgroup_id**\ ().
*
* This helper is available only if the kernel was compiled with
* the
* **CONFIG_SOCK_CGROUP_DATA** configuration option.
*
* Returns
* The id is returned or 0 in case the id could not be retrieved.
*/
static __u64 (* const bpf_sk_cgroup_id)(void *sk) = (void *) 128;

/*
* bpf_sk_ancestor_cgroup_id
*
* Return id of cgroup v2 that is ancestor of cgroup associated
* with the *sk* at the *ancestor_level*. The root cgroup is at
* *ancestor_level* zero and each step down the hierarchy
* increments the level. If *ancestor_level* == level of cgroup
* associated with *sk*, then return value will be same as that
* of **bpf_sk_cgroup_id**\ ().
*
* The helper is useful to implement policies based on cgroups
* that are upper in hierarchy than immediate cgroup associated
* with *sk*.
*
* The format of returned id and helper limitations are same as in
* **bpf_sk_cgroup_id**\ ().
*
* Returns

```

```

* The id is returned or 0 in case the id could not be retrieved.
*/
static __u64 (* const bpf_sk_ancestor_cgroup_id)(void *sk, int ancestor_level)
= (void *) 129;

/*
* bpf_ringbuf_output
*
* Copy *size* bytes from *data* into a ring buffer *ringbuf*.
* If **BPF_RB_NO_WAKEUP** is specified in *flags*, no notification
* of new data availability is sent.
* If **BPF_RB_FORCE_WAKEUP** is specified in *flags*, notification
* of new data availability is sent unconditionally.
* If **0** is specified in *flags*, an adaptive notification
* of new data availability is sent.
*
* An adaptive notification is a notification sent whenever the user-space
* process has caught up and consumed all available payloads. In case the user-space
* process is still processing a previous payload, then no notification is needed
* as it will process the newly added payload automatically.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_ringbuf_output)(void *ringbuf, void *data, __u64 size, __u64 flags) = (void *) 130;

/*
* bpf_ringbuf_reserve
*
* Reserve *size* bytes of payload
in a ring buffer *ringbuf*.
* *flags* must be 0.
*
* Returns
* Valid pointer with *size* bytes of memory available; NULL,
* otherwise.
*/
static void (* const bpf_ringbuf_reserve)(void *ringbuf, __u64 size, __u64 flags) = (void *) 131;

/*
* bpf_ringbuf_submit
*
* Submit reserved ring buffer sample, pointed to by *data*.
* If **BPF_RB_NO_WAKEUP** is specified in *flags*, no notification
* of new data availability is sent.
* If **BPF_RB_FORCE_WAKEUP** is specified in *flags*, notification
* of new data availability is sent unconditionally.
* If **0** is specified in *flags*, an adaptive notification

```

```

* of new data availability is sent.
*
* See 'bpf_ringbuf_output()' for the definition of adaptive notification.
*
* Returns
* Nothing. Always succeeds.
*/
static void (* const bpf_ringbuf_submit)(void *data, __u64 flags) = (void *) 132;

/*
* bpf_ringbuf_discard
*
* Discard reserved ring buffer sample, pointed to by *data*.
* If **BPF_RB_NO_WAKEUP**
is specified in *flags*, no notification
* of new data availability is sent.
* If **BPF_RB_FORCE_WAKEUP** is specified in *flags*, notification
* of new data availability is sent unconditionally.
* If **0** is specified in *flags*, an adaptive notification
* of new data availability is sent.
*
* See 'bpf_ringbuf_output()' for the definition of adaptive notification.
*
* Returns
* Nothing. Always succeeds.
*/
static void (* const bpf_ringbuf_discard)(void *data, __u64 flags) = (void *) 133;

/*
* bpf_ringbuf_query
*
* Query various characteristics of provided ring buffer. What
* exactly is queried is determined by *flags*:
*
* **BPF_RB_AVAIL_DATA**: Amount of data not yet consumed.
* **BPF_RB_RING_SIZE**: The size of ring buffer.
* **BPF_RB_CONS_POS**: Consumer position (can wrap around).
* **BPF_RB_PROD_POS**: Producer(s) position (can wrap around).
*
* Data returned is just a momentary snapshot of actual values
* and could be inaccurate,
so this facility should be used to
* power heuristics and for reporting, not to make 100% correct
* calculation.
*
* Returns
* Requested value, or 0, if *flags* are not recognized.
*/

```

```

static __u64 (* const bpf_ringbuf_query)(void *ringbuf, __u64 flags) = (void *) 134;

/*
 * bpf_csum_level
 *
 * Change the skbs checksum level by one layer up or down, or
 * reset it entirely to none in order to have the stack perform
 * checksum validation. The level is applicable to the following
 * protocols: TCP, UDP, GRE, SCTP, FCOE. For example, a decap of
 * | ETH | IP | UDP | GUE | IP | TCP | into | ETH | IP | TCP |
 * through bpf_skb_adjust_room() helper with passing in
 * BPF_F_ADJ_ROOM_NO_CSUM_RESET flag would require one call
 * to bpf_csum_level() with BPF_CSUM_LEVEL_DEC since
 * the UDP header is removed. Similarly, an encap of the latter
 * into the former could be accompanied by a helper call to
 * bpf_csum_level() with BPF_CSUM_LEVEL_INC
 * if the
 * skb is still intended to be processed in higher layers of the
 * stack instead of just egressing at tc.
 *
 * There are three supported level settings at this time:
 *
 * BPF_CSUM_LEVEL_INC: Increases skb->csum_level for skbs
 * with CHECKSUM_UNNECESSARY.
 * BPF_CSUM_LEVEL_DEC: Decreases skb->csum_level for skbs
 * with CHECKSUM_UNNECESSARY.
 * BPF_CSUM_LEVEL_RESET: Resets skb->csum_level to 0 and
 * sets CHECKSUM_NONE to force checksum validation by the stack.
 * BPF_CSUM_LEVEL_QUERY: No-op, returns the current
 * skb->csum_level.
 *
 * Returns
 * 0 on success, or a negative error in case of failure. In the
 * case of BPF_CSUM_LEVEL_QUERY, the current skb->csum_level
 * is returned or the error code -EACCES in case the skb is not
 * subject to CHECKSUM_UNNECESSARY.
 */
static long (* const bpf_csum_level)(struct __sk_buff *skb, __u64 level) = (void *) 135;

/*
 * bpf_skc_to_tcp6_sock
 *
 * Dynamically cast a *sk* pointer to
 * a *tcp6_sock* pointer.
 *
 * Returns
 * *sk* if casting is valid, or NULL otherwise.
 */

```

```

static struct tcp6_sock *(* const bpf_skc_to_tcp6_sock)(void *sk) = (void *) 136;

/*
 * bpf_skc_to_tcp_sock
 *
 * Dynamically cast a *sk* pointer to a *tcp_sock* pointer.
 *
 * Returns
 * *sk* if casting is valid, or **NULL** otherwise.
 */
static struct tcp_sock *(* const bpf_skc_to_tcp_sock)(void *sk) = (void *) 137;

/*
 * bpf_skc_to_tcp_timewait_sock
 *
 * Dynamically cast a *sk* pointer to a *tcp_timewait_sock* pointer.
 *
 * Returns
 * *sk* if casting is valid, or **NULL** otherwise.
 */
static struct tcp_timewait_sock *(* const bpf_skc_to_tcp_timewait_sock)(void *sk) = (void *) 138;

/*
 * bpf_skc_to_tcp_request_sock
 *
 * Dynamically cast a *sk* pointer to a *tcp_request_sock* pointer.
 *
 * Returns
 * *sk* if casting is valid, or **NULL** otherwise.
 */
static struct tcp_request_sock *(* const bpf_skc_to_tcp_request_sock)(void *sk) = (void *) 139;

/*
 * bpf_skc_to_udp6_sock
 *
 * Dynamically cast a *sk* pointer to a *udp6_sock* pointer.
 *
 * Returns
 * *sk* if casting is valid, or **NULL** otherwise.
 */
static struct udp6_sock *(* const bpf_skc_to_udp6_sock)(void *sk) = (void *) 140;

/*
 * bpf_get_task_stack
 *
 * Return a user or a kernel stack in bpf program provided buffer.
 * Note: the user stack will only be populated if the *task* is
 * the current task; all other tasks will return -EOPNOTSUPP.

```

- * To achieve this, the helper needs `*task*`, which is a valid
- * pointer to `**struct task_struct**`. To store the stacktrace, the
- * bpf program provides `*buf*` with a nonnegative `*size*`.
- *
- * The last argument, `*flags*`, holds the number of stack frames to
- * skip (from 0 to 255), masked with
- * `**BPF_F_SKIP_FIELD_MASK**`. The next bits can be used to set
- * the following flags:
- *
- * `**BPF_F_USER_STACK**`
- * Collect a user space stack instead of a kernel stack.
- * The `*task*` must be the current task.
- * `**BPF_F_USER_BUILD_ID**`
- * Collect
- buildid+offset instead of ips for user stack,
- * only valid if `**BPF_F_USER_STACK**` is also specified.
- *
- * `**bpf_get_task_stack**\()` can collect up to
- * `**PERF_MAX_STACK_DEPTH**` both kernel and user frames, subject
- * to sufficient large buffer size. Note that
- * this limit can be controlled with the `**sysctl**` program, and
- * that it should be manually increased in order to profile long
- * user stacks (such as stacks for Java programs). To do so, use:
- *
- * ::
- *
- * # sysctl kernel.perf_event_max_stack=<new value>
- *
- * Returns
- * The non-negative copied `*buf*` length equal to or less than
- * `*size*` on success, or a negative error in case of failure.
- */

```
static long (* const bpf_get_task_stack)(struct task_struct *task, void *buf, __u32 size, __u64 flags) = (void *) 141;
```



```
/*
 * bpf_load_hdr_opt
 *
 * Load header option. Support reading a particular TCP header
 * option for bpf program (**BPF_PROG_TYPE_SOCKET_OPS**).
 *
 * If *flags* is 0, it will search the option from
 the
 * *skops* \ **->skb_data**. The comment in **struct bpf_sock_ops**
 * has details on what skb_data contains under different
 * *skops* \ **->op**.
 *
 * The first byte of the *searchby_res* specifies the
 * kind that it wants to search.
```


- *
 - * If the searching kind is an experimental kind
 - * (i.e. 253 or 254 according to RFC6994). It also
 - * needs to specify the "magic" which is either
 - * 2 bytes or 4 bytes. It then also needs to
 - * specify the size of the magic by using
 - * the 2nd byte which is "kind-length" of a TCP
 - * header option and the "kind-length" also
 - * includes the first 2 bytes "kind" and "kind-length"
 - * itself as a normal TCP header option also does.
- *
 - * For example, to search experimental kind 254 with
 - * 2 byte magic 0xB9F, the searchby_res should be
 - * [254, 4, 0xB, 0x9F, 0, 0, ... 0].
- *
 - * To search for the standard window scale option (3),
 - * the *searchby_res* should be [3, 0, 0, ... 0].
 - * Note, kind-length must be 0 for regular
 - option.
- *
 - * Searching for No-Op (0) and End-of-Option-List (1) are
 - * not supported.
- *
 - * *len* must be at least 2 bytes which is the minimal size
 - * of a header option.
- *
 - * Supported flags:
- *
 - * *BPF_LOAD_HDR_OPT_TCP_SYN* to search from the
 - * saved_syn packet or the just-received syn packet.
- *
 - * Returns
 - * > 0 when found, the header option is copied to *searchby_res*.
 - * The return value is the total length copied. On failure, a
 - * negative error code is returned:
- *
 - * *-EINVAL* if a parameter is invalid.
- *
 - * *-ENOMSG* if the option is not found.
- *
 - * *-ENOENT* if no syn packet is available when
 - * *BPF_LOAD_HDR_OPT_TCP_SYN* is used.
- *
 - * *-ENOSPC* if there is not enough space. Only *len* number of
 - * bytes are copied.
- *
 - * *-EFAULT* on failure to parse the header options in the

```

* packet.
*
* **_EPERM** if the helper cannot be used under the current
* *skops*\ **->op**.
*/
static long (* const bpf_load_hdr_opt)(struct bpf_sock_ops
*skops, void *searchby_res, __u32 len, __u64 flags) = (void *) 142;

/*
* bpf_store_hdr_opt
*
* Store header option. The data will be copied
* from buffer *from* with length *len* to the TCP header.
*
* The buffer *from* should have the whole option that
* includes the kind, kind-length, and the actual
* option data. The *len* must be at least kind-length
* long. The kind-length does not have to be 4 byte
* aligned. The kernel will take care of the padding
* and setting the 4 bytes aligned value to th->doff.
*
* This helper will check for duplicated option
* by searching the same option in the outgoing skb.
*
* This helper can only be called during
* **BPF_SOCK_OPS_WRITE_HDR_OPT_CB**.
*
*
* Returns
* 0 on success, or negative error in case of failure:
*
* **_EINVAL** If param is invalid.
*
* **_ENOSPC** if there is not enough space in the header.
* Nothing has been written
*
* **_EEXIST** if the option already exists.
*
* **_EFAULT**
on failure to parse the existing header options.
*
* **_EPERM** if the helper cannot be used under the current
* *skops*\ **->op**.
*/
static long (* const bpf_store_hdr_opt)(struct bpf_sock_ops *skops, const void *from, __u32 len, __u64 flags) =
(void *) 143;

/*

```

```

* bpf_reserve_hdr_opt
*
* Reserve *len* bytes for the bpf header option. The
* space will be used by bpf_store_hdr_opt() later in
* BPF SOCK OPS WRITE HDR OPT CB.
*
* If bpf_reserve_hdr_opt() is called multiple times,
* the total number of bytes will be reserved.
*
* This helper can only be called during
* BPF SOCK OPS HDR OPT LEN CB.
*
*
* Returns
* 0 on success, or negative error in case of failure:
*
* -EINVAL if a parameter is invalid.
*
* -ENOSPC if there is not enough space in the header.
*
* -EPERM if the helper cannot be used under the current
* skops ->op.
*/
static long (* const bpf_reserve_hdr_opt)(struct bpf_sock_ops *skops,
__u32 len, __u64 flags) = (void *) 144;

/*
* bpf_inode_storage_get
*
* Get a bpf_local_storage from an *inode*.
*
* Logically, it could be thought of as getting the value from
* a *map* with *inode* as the key. From this
* perspective, the usage is not much different from
* bpf_map_lookup_elem(*map*, &*inode*) except this
* helper enforces the key must be an inode and the map must also
* be a BPF_MAP_TYPE_INODE_STORAGE.
*
* Underneath, the value is stored locally at *inode* instead of
* the *map*. The *map* is used as the bpf-local-storage
* "type". The bpf-local-storage "type" (i.e. the *map*) is
* searched against all bpf_local_storage residing at *inode*.
*
* An optional *flags* (BPF_LOCAL_STORAGE_GET_F_CREATE) can be
* used such that a new bpf_local_storage will be
* created if one does not exist. *value* can be used
* together with BPF_LOCAL_STORAGE_GET_F_CREATE to specify
* the initial value of a bpf_local_storage.

```

```

    If *value* is
    * **NULL**, the new bpf_local_storage will be zero initialized.
    *
    * Returns
    * A bpf_local_storage pointer is returned on success.
    *
    * **NULL** if not found or there was an error in adding
    * a new bpf_local_storage.
    */
static void (* const bpf_inode_storage_get)(void *map, void *inode, void *value, __u64 flags) = (void *) 145;

/*
 * bpf_inode_storage_delete
 *
 * Delete a bpf_local_storage from an *inode*.
 *
 * Returns
 * 0 on success.
 *
 * **ENOENT** if the bpf_local_storage cannot be found.
 */
static int (* const bpf_inode_storage_delete)(void *map, void *inode) = (void *) 146;

/*
 * bpf_d_path
 *
 * Return full path for given **struct path** object, which
 * needs to be the kernel BTF *path* object. The path is
 * returned in the provided buffer *buf* of size *sz* and
 * is zero terminated.
 *
 * Returns
 * On success, the strictly positive length of the string,
 * including the trailing NUL character. On error, a negative
 * value.
 */
static long (* const bpf_d_path)(struct path *path, char *buf, __u32 sz) = (void *) 147;

/*
 * bpf_copy_from_user
 *
 * Read *size* bytes from user space address *user_ptr* and store
 * the data in *dst*. This is a wrapper of **copy_from_user**().
 *
 * Returns
 * 0 on success, or a negative error in case of failure.
 */

```

```

static long (* const bpf_copy_from_user)(void *dst, __u32 size, const void *user_ptr) = (void *) 148;

/*
 * bpf_snprintf_btf
 *
 * Use BTF to store a string representation of *ptr*->ptr in *str*,
 * using *ptr*->type_id. This value should specify the type
 * that *ptr*->ptr points to. LLVM __builtin_btf_type_id(type, 1)
 * can be used to look up vmlinux BTF type ids. Traversing the
 * data structure using BTF, the type information and values are
 * stored in the first *str_size* - 1 bytes of *str*. Safe copy of
 * the pointer data is carried out to avoid kernel crashes during
 * operation. Smaller types can use string space on the stack;
 * larger
programs can use map data to store the string
 * representation.
 *
 * The string can be subsequently shared with userspace via
 * bpf_perf_event_output() or ring buffer interfaces.
 * bpf_trace_printk() is to be avoided as it places too small
 * a limit on string size to be useful.
 *
 * *flags* is a combination of
 *
 * **BTF_F_COMPACT**
 * no formatting around type information
 * **BTF_F_NONAME**
 * no struct/union member names/types
 * **BTF_F_PTR_RAW**
 * show raw (unobfuscated) pointer values;
 * equivalent to printk specifier %px.
 * **BTF_F_ZERO**
 * show zero-valued struct/union members; they
 * are not displayed by default
 *
 * Returns
 * The number of bytes that were written (or would have been
 * written if output had to be truncated due to string size),
 * or a negative error in cases of failure.
 */
static long (* const bpf_snprintf_btf)(char *str, __u32 str_size, struct btf_ptr *ptr, __u32 btf_ptr_size, __u64 flags) =
(void *) 149;

/*
 * bpf_seq_printf_btf
 *
 * Use BTF to write to seq_write a string representation of

```

```

* *ptr*->ptr, using *ptr*->type_id as per bpf_snprintf_btf().
* *flags* are identical to those used for bpf_snprintf_btf.
*
* Returns
* 0 on success or a negative error in case of failure.
*/
static long (* const bpf_seq_printf_btf)(struct seq_file *m, struct btf_ptr *ptr, __u32 ptr_size, __u64 flags) = (void *) 150;

/*
* bpf_skb_cgroup_classid
*
* See **bpf_get_cgroup_classid**\ () for the main description.
* This helper differs from **bpf_get_cgroup_classid**\ () in that
* the cgroup v1 net_cls class is retrieved only from the *skb*'s
* associated socket instead of the current process.
*
* Returns
* The id is returned or 0 in case the id could not be retrieved.
*/
static __u64 (* const bpf_skb_cgroup_classid)(struct __sk_buff *skb) = (void *) 151;

/*
* bpf_redirect_neigh
*
* Redirect the packet to another net device of index *ifindex*
* and fill in L2 addresses from neighboring
* subsystem. This helper
* is somewhat similar to **bpf_redirect**\ (), except that it
* populates L2 addresses as well, meaning, internally, the helper
* relies on the neighbor lookup for the L2 address of the nexthop.
*
* The helper will perform a FIB lookup based on the skb's
* networking header to get the address of the next hop, unless
* this is supplied by the caller in the *params* argument. The
* *plen* argument indicates the len of *params* and should be set
* to 0 if *params* is NULL.
*
* The *flags* argument is reserved and must be 0. The helper is
* currently only supported for tc BPF program types, and enabled
* for IPv4 and IPv6 protocols.
*
* Returns
* The helper returns **TC_ACT_REDIRECT** on success or
* **TC_ACT_SHOT** on error.
*/
static long (* const bpf_redirect_neigh)(__u32 ifindex, struct bpf_redir_neigh *params, int plen, __u64 flags) =
(void *) 152;

```

```

/*
 * bpf_per_cpu_ptr
 *
 * Take a pointer to a percpu ksym, *percpu_ptr*,
 * and return a
 * pointer to the percpu kernel variable on *cpu*. A ksym is an
 * extern variable decorated with '__ksym'. For ksym, there is a
 * global var (either static or global) defined of the same name
 * in the kernel. The ksym is percpu if the global var is percpu.
 * The returned pointer points to the global percpu var on *cpu*.
 *
 * bpf_per_cpu_ptr() has the same semantic as per_cpu_ptr() in the
 * kernel, except that bpf_per_cpu_ptr() may return NULL. This
 * happens if *cpu* is larger than nr_cpu_ids. The caller of
 * bpf_per_cpu_ptr() must check the returned value.
 *
 * Returns
 * A pointer pointing to the kernel percpu variable on *cpu*, or
 * NULL, if *cpu* is invalid.
 */
static void *(* const bpf_per_cpu_ptr)(const void *percpu_ptr, __u32 cpu) = (void *) 153;

```

```

/*
 * bpf_this_cpu_ptr
 *
 * Take a pointer to a percpu ksym, *percpu_ptr*, and return a
 * pointer to the percpu kernel variable on this cpu. See the
 * description of 'ksym' in **bpf_per_cpu_ptr**\
().
 *
 * bpf_this_cpu_ptr() has the same semantic as this_cpu_ptr() in
 * the kernel. Different from **bpf_per_cpu_ptr**\ (), it would
 * never return NULL.
 *
 * Returns
 * A pointer pointing to the kernel percpu variable on this cpu.
 */
static void *(* const bpf_this_cpu_ptr)(const void *percpu_ptr) = (void *) 154;

```

```

/*
 * bpf_redirect_peer
 *
 * Redirect the packet to another net device of index *ifindex*.
 * This helper is somewhat similar to **bpf_redirect**\ (), except
 * that the redirection happens to the *ifindex* peer device and
 * the netns switch takes place from ingress to ingress without
 * going through the CPU's backlog queue.

```

```

*
* The *flags* argument is reserved and must be 0. The helper is
* currently only supported for tc BPF program types at the
* ingress hook and for veth and netkit target device types. The
* peer device must reside in a different network namespace.
*
* Returns
* The helper returns TC_ACT_REDIRECT on success or
*
* TC_ACT_SHOT on error.
*/
static long (* const bpf_redirect_peer)(__u32 ifindex, __u64 flags) = (void *) 155;

/*
* bpf_task_storage_get
*
* Get a bpf_local_storage from the *task*.
*
* Logically, it could be thought of as getting the value from
* a *map* with *task* as the key. From this
* perspective, the usage is not much different from
* bpf_map_lookup_elem(*map*, &task) except this
* helper enforces the key must be a task_struct and the map must also
* be a BPF_MAP_TYPE_TASK_STORAGE.
*
* Underneath, the value is stored locally at *task* instead of
* the *map*. The *map* is used as the bpf-local-storage
* "type". The bpf-local-storage "type" (i.e. the *map*) is
* searched against all bpf_local_storage residing at *task*.
*
* An optional *flags* (BPF_LOCAL_STORAGE_GET_F_CREATE) can be
* used such that a new bpf_local_storage will be
* created if one does not exist. *value* can be used
* together with BPF_LOCAL_STORAGE_GET_F_CREATE
* to specify
* the initial value of a bpf_local_storage. If *value* is
* NULL, the new bpf_local_storage will be zero initialized.
*
* Returns
* A bpf_local_storage pointer is returned on success.
*
* NULL if not found or there was an error in adding
* a new bpf_local_storage.
*/
static void (* const bpf_task_storage_get)(void *map, struct task_struct *task, void *value, __u64 flags) = (void *)
156;

/*

```



```

* bpf_task_storage_delete
*
* Delete a bpf_local_storage from a *task*.
*
* Returns
* 0 on success.
*
* **ENOENT** if the bpf_local_storage cannot be found.
*/
static long (* const bpf_task_storage_delete)(void *map, struct task_struct *task) = (void *) 157;

/*
* bpf_get_current_task_btf
*
* Return a BTF pointer to the "current" task.
* This pointer can also be used in helpers that accept an
* *ARG_PTR_TO_BTF_ID* of type *task_struct*.
*
* Returns
* Pointer to the current task.
*/
static struct task_struct *(* const
bpf_get_current_task_btf)(void) = (void *) 158;

/*
* bpf_bprm_opts_set
*
* Set or clear certain options on *bprm*:
*
* **BPF_F_BPRM_SECUREEXEC** Set the secureexec bit
* which sets the **AT_SECURE** auxv for glibc. The bit
* is cleared if the flag is not specified.
*
* Returns
* **EINVAL** if invalid *flags* are passed, zero otherwise.
*/
static long (* const bpf_bprm_opts_set)(struct linux_binprm *bprm, __u64 flags) = (void *) 159;

/*
* bpf_ktime_get_coarse_ns
*
* Return a coarse-grained version of the time elapsed since
* system boot, in nanoseconds. Does not include time the system
* was suspended.
*
* See: **clock_gettime**\ (**CLOCK_MONOTONIC_COARSE**)
*
* Returns

```

```

* Current *ktime*.
*/
static __u64 (* const bpf_ktime_get_coarse_ns)(void) = (void *) 160;

/*
* bpf_ima_inode_hash
*
* Returns the stored IMA hash of the *inode* (if it's available).
* If the hash is larger than *size*, then only *size*
* bytes will be copied to *dst*
*
* Returns
* The **hash_algo** is returned on success,
* **-EOPNOTSUPP** if IMA is disabled or **-EINVAL** if
* invalid arguments are passed.
*/
static long (* const bpf_ima_inode_hash)(struct inode *inode, void *dst, __u32 size) = (void *) 161;

/*
* bpf_sock_from_file
*
* If the given file represents a socket, returns the associated
* socket.
*
* Returns
* A pointer to a struct socket on success or NULL if the file is
* not a socket.
*/
static struct socket *(* const bpf_sock_from_file)(struct file *file) = (void *) 162;

/*
* bpf_check_mtu
*
* Check packet size against exceeding MTU of net device (based
* on *ifindex*). This helper will likely be used in combination
* with helpers that adjust/change the packet size.
*
* The argument *len_diff* can be used for querying with a planned
* size change. This allows to check MTU prior to changing packet
* ctx. Providing a *len_diff* adjustment that is larger than the
* actual packet size (resulting in negative
* packet size) will in
* principle not exceed the MTU, which is why it is not considered
* a failure. Other BPF helpers are needed for performing the
* planned size change; therefore the responsibility for catching
* a negative packet size belongs in those helpers.
*
* Specifying *ifindex* zero means the MTU check is performed

```

- * against the current net device. This is practical if this isn't
- * used prior to redirect.
- *
- * On input `*mtu_len*` must be a valid pointer, else verifier will
- * reject BPF program. If the value `*mtu_len*` is initialized to
- * zero then the ctx packet size is use. When value `*mtu_len*` is
- * provided as input this specify the L3 length that the MTU check
- * is done against. Remember XDP and TC length operate at L2, but
- * this value is L3 as this correlate to MTU and IP-header `tot_len`
- * values which are L3 (similar behavior as `bpf_fib_lookup`).
- *
- * The Linux kernel route table can configure MTUs on a more
- * specific per route level,
- which is not provided by this helper.
- * For route level MTU checks use the `**bpf_fib_lookup**\()`
- * helper.
- *
- * `*ctx*` is either `**struct xdp_md**` for XDP programs or
- * `**struct sk_buff**` for tc `cls_act` programs.
- *
- * The `*flags*` argument can be a combination of one or more of the
- * following values:
- *
- * `**BPF_MTU_CHK_SEGS**`
- * This flag will only works for `*ctx*` `**struct sk_buff**`.
- * If packet context contains extra packet segment buffers
- * (often knows as GSO skb), then MTU check is harder to
- * check at this point, because in transmit path it is
- * possible for the skb packet to get re-segmented
- * (depending on net device features). This could still be
- * a MTU violation, so this flag enables performing MTU
- * check against segments, with a different violation
- * return code to tell it apart. Check cannot use `len_diff`.
- *
- * On return `*mtu_len*` pointer contains the MTU value of the net
- * device. Remember the net device configured MTU is the
- L3 size,
- * which is returned here and XDP and TC length operate at L2.
- * Helper take this into account for you, but remember when using
- * MTU value in your BPF-code.
- *
- *
- * Returns
- * * 0 on success, and populate MTU value in `*mtu_len*` pointer.
- *
- * * < 0 if any input argument is invalid (`*mtu_len*` not updated)
- *
- * MTU violations return positive values, but also populate MTU

```

* value in *mtu_len* pointer, as this can be needed for
* implementing PMTU handling:
*
* ***BPF_MTU_CHK_RET_FRAG_NEEDED***
* ***BPF_MTU_CHK_RET_SEGS_TOOBIG***
*/
static long (* const bpf_check_mtu)(void *ctx, __u32 ifindex, __u32 *mtu_len, __s32 len_diff, __u64 flags) = (void
*) 163;

/*
* bpf_for_each_map_elem
*
* For each element in **map**, call **callback_fn** function with
* **map**, **callback_ctx** and other map-specific parameters.
* The **callback_fn** should be a static function and
* the **callback_ctx** should be a pointer to the stack.
* The **flags** is used to control
certain aspects of the helper.
* Currently, the **flags** must be 0.
*
* The following are a list of supported map types and their
* respective expected callback signatures:
*
* BPF_MAP_TYPE_HASH, BPF_MAP_TYPE_PERCPU_HASH,
* BPF_MAP_TYPE_LRU_HASH, BPF_MAP_TYPE_LRU_PERCPU_HASH,
* BPF_MAP_TYPE_ARRAY, BPF_MAP_TYPE_PERCPU_ARRAY
*
* long (*callback_fn)(struct bpf_map *map, const void *key, void *value, void *ctx);
*
* For per_cpu maps, the map_value is the value on the cpu where the
* bpf_prog is running.
*
* If **callback_fn** return 0, the helper will continue to the next
* element. If return value is 1, the helper will skip the rest of
* elements and return. Other return values are not used now.
*
*
* Returns
* The number of traversed map elements for success, **-EINVAL** for
* invalid **flags**.
*/
static long (* const bpf_for_each_map_elem)(void *map, void *callback_fn, void *callback_ctx, __u64 flags) =
(void *) 164;

/*
* bpf_snprintf
*
* Outputs a string into the **str** buffer of size **str_size**

```

```

* based on a format string stored in a read-only map pointed by
* fmt.
*
* Each format specifier in fmt corresponds to one u64 element
* in the data array. For strings and pointers where pointees
* are accessed, only the pointer values are stored in the data
* array. The data_len is the size of data in bytes - must be
* a multiple of 8.
*
* Formats %s and %p{i,I}{4,6} require to read kernel
* memory. Reading kernel memory may fail due to either invalid
* address or valid address but requiring a major memory fault. If
* reading kernel memory fails, the string for %s will be an
* empty string, and the ip address for %p{i,I}{4,6} will be 0.
* Not returning error to bpf program is consistent with what
* bpf_trace_printk() does for now.
*
*
* Returns
* The strictly positive length of the formatted string, including
* the trailing zero character. If
the return value is greater than
* str_size, str contains a truncated string, guaranteed to
* be zero-terminated except when str_size is 0.
*
* Or -EBUSY if the per-CPU memory copy buffer is busy.
*/
static long (* const bpf_snprintf)(char *str, __u32 str_size, const char *fmt, __u64 *data, __u32 data_len) = (void *)
165;

/*
* bpf_sys_bpf
*
* Execute bpf syscall with given arguments.
*
* Returns
* A syscall result.
*/
static long (* const bpf_sys_bpf)(__u32 cmd, void *attr, __u32 attr_size) = (void *) 166;

/*
* bpf_btf_find_by_name_kind
*
* Find BTF type with given name and kind in vmlinux BTF or in module's BTFs.
*
* Returns
* Returns btf_id and btf_obj_fd in lower and upper 32 bits.
*/

```

```
static long (* const bpf_btf_find_by_name_kind)(char *name, int name_sz, __u32 kind, int flags) = (void *) 167;
```

```
/*
```

```
* bpf_sys_close
```

```
*
```

```
* Execute close syscall for given FD.
```

```
*
```

```
* Returns
```

```
* A syscall result.
```

```
*/
```

```
static long (* const bpf_sys_close)(__u32
```

```
fd) = (void *) 168;
```

```
/*
```

```
* bpf_timer_init
```

```
*
```

```
* Initialize the timer.
```

```
* First 4 bits of *flags* specify clockid.
```

```
* Only CLOCK_MONOTONIC, CLOCK_REALTIME, CLOCK_BOOTTIME are allowed.
```

```
* All other bits of *flags* are reserved.
```

```
* The verifier will reject the program if *timer* is not from
```

```
* the same *map*.
```

```
*
```

```
* Returns
```

```
* 0 on success.
```

```
* **EBUSY** if *timer* is already initialized.
```

```
* **EINVAL** if invalid *flags* are passed.
```

```
* **EPERM** if *timer* is in a map that doesn't have any user references.
```

```
* The user space should either hold a file descriptor to a map with timers
```

```
* or pin such map in bpffs. When map is unpinned or file descriptor is
```

```
* closed all timers in the map will be cancelled and freed.
```

```
*/
```

```
static long (* const bpf_timer_init)(struct bpf_timer *timer, void *map, __u64 flags) = (void *) 169;
```

```
/*
```

```
* bpf_timer_set_callback
```

```
*
```

```
* Configure the timer to call *callback_fn* static function.
```

```
*
```

```
* Returns
```

```
* 0 on success.
```

```
* **EINVAL** if *timer* was not  
initialized with bpf_timer_init() earlier.
```

```
* **EPERM** if *timer* is in a map that doesn't have any user references.
```

```
* The user space should either hold a file descriptor to a map with timers
```

```
* or pin such map in bpffs. When map is unpinned or file descriptor is
```

```
* closed all timers in the map will be cancelled and freed.
```

```
*/
```

```
static long (* const bpf_timer_set_callback)(struct bpf_timer *timer, void *callback_fn) = (void *) 170;
```

```
/*
```

```
 * bpf_timer_start
```

```
 *
```

```
 * Set timer expiration N nanoseconds from the current time. The  
 * configured callback will be invoked in soft irq context on some cpu  
 * and will not repeat unless another bpf_timer_start() is made.  
 * In such case the next invocation can migrate to a different cpu.  
 * Since struct bpf_timer is a field inside map element the map  
 * owns the timer. The bpf_timer_set_callback() will increment refcnt  
 * of BPF program to make sure that callback_fn code stays valid.  
 * When user space reference to a map reaches zero all  
 timers  
 * in a map are cancelled and corresponding program's refcnts are  
 * decremented. This is done to make sure that Ctrl-C of a user  
 * process doesn't leave any timers running. If map is pinned in  
 * bpf's the callback_fn can re-arm itself indefinitely.  
 * bpf_map_update/delete_elem() helpers and user space sys_bpf commands  
 * cancel and free the timer in the given map element.  
 * The map can contain timers that invoke callback_fn-s from different  
 * programs. The same callback_fn can serve different timers from  
 * different maps if key/value layout matches across maps.  
 * Every bpf_timer_set_callback() can have different callback_fn.
```

```
 *
```

```
 * *flags* can be one of:
```

```
 *
```

```
 * **BPF_F_TIMER_ABS**
```

```
 * Start the timer in absolute expire value instead of the  
 * default relative one.
```

```
 * **BPF_F_TIMER_CPU_PIN**
```

```
 * Timer will be pinned to the CPU of the caller.
```

```
 *
```

```
 *
```

```
 * Returns
```

```
 * 0 on success.
```

```
 * **EINVAL** if *timer* was not initialized with bpf_timer_init() earlier
```

```
 * or
```

```
 invalid *flags* are passed.
```

```
 */
```

```
static long (* const bpf_timer_start)(struct bpf_timer *timer, __u64 nsecs, __u64 flags) = (void *) 171;
```

```
/*
```

```
 * bpf_timer_cancel
```

```
 *
```

```
 * Cancel the timer and wait for callback_fn to finish if it was running.
```

```
 *
```

```
 * Returns
```

```

* 0 if the timer was not active.
* 1 if the timer was active.
* **EINVAL** if *timer* was not initialized with bpf_timer_init() earlier.
* **EDEADLK** if callback_fn tried to call bpf_timer_cancel() on its
* own timer which would have led to a deadlock otherwise.
*/
static long (* const bpf_timer_cancel)(struct bpf_timer *timer) = (void *) 172;

/*
* bpf_get_func_ip
*
* Get address of the traced function (for tracing and kprobe programs).
*
* When called for kprobe program attached as uprobe it returns
* probe address for both entry and return uprobe.
*
* Returns
* Address of the traced function for kprobe.
* 0 for kprobes placed within the function (not at the entry).
* Address of the probe for uprobe
and return uprobe.
*/
static __u64 (* const bpf_get_func_ip)(void *ctx) = (void *) 173;

/*
* bpf_get_attach_cookie
*
* Get bpf_cookie value provided (optionally) during the program
* attachment. It might be different for each individual
* attachment, even if BPF program itself is the same.
* Expects BPF program context *ctx* as a first argument.
*
* Supported for the following program types:
* - kprobe/uprobe;
* - tracepoint;
* - perf_event.
*
* Returns
* Value specified by user at BPF link creation/attachment time
* or 0, if it was not specified.
*/
static __u64 (* const bpf_get_attach_cookie)(void *ctx) = (void *) 174;

/*
* bpf_task_pt_regs
*
* Get the struct pt_regs associated with **task**.

```



```

*
* Returns
* A pointer to struct pt_regs.
*/
static long (* const bpf_task_pt_regs)(struct task_struct *task) = (void *) 175;

/*
* bpf_get_branch_snapshot
*
* Get branch trace from hardware engines like Intel LBR. The
* hardware engine is stopped
* shortly after the helper is
* called. Therefore, the user need to filter branch entries
* based on the actual use case. To capture branch trace
* before the trigger point of the BPF program, the helper
* should be called at the beginning of the BPF program.
*
* The data is stored as struct perf_branch_entry into output
* buffer *entries*. *size* is the size of *entries* in bytes.
* *flags* is reserved for now and must be zero.
*
*
* Returns
* On success, number of bytes written to *buf*. On error, a
* negative value.
*
* **EINVAL** if *flags* is not zero.
*
* **ENOENT** if architecture does not support branch records.
*/
static long (* const bpf_get_branch_snapshot)(void *entries, __u32 size, __u64 flags) = (void *) 176;

/*
* bpf_trace_vprintk
*
* Behaves like **bpf_trace_printk**\ () helper, but takes an array of u64
* to format and can handle more format args as a result.
*
* Arguments are to be used as in **bpf_seq_printf**\ () helper.
*
*
* Returns
* The number of bytes written to the buffer, or a negative error
* in case of failure.
*/
static long (* const bpf_trace_vprintk)(const char *fmt, __u32 fmt_size, const void *data, __u32 data_len) = (void *) 177;

```

```

/*
 * bpf_skc_to_unix_sock
 *
 * Dynamically cast a *sk* pointer to a *unix_sock* pointer.
 *
 * Returns
 * *sk* if casting is valid, or **NULL** otherwise.
 */
static struct unix_sock *(* const bpf_skc_to_unix_sock)(void *sk) = (void *) 178;

/*
 * bpf_kallsyms_lookup_name
 *
 * Get the address of a kernel symbol, returned in *res*. *res* is
 * set to 0 if the symbol is not found.
 *
 * Returns
 * On success, zero. On error, a negative value.
 *
 * **EINVAL** if *flags* is not zero.
 *
 * **EINVAL** if string *name* is not the same size as *name_sz*.
 *
 * **ENOENT** if symbol is not found.
 *
 * **EPERM** if caller does not have permission to obtain kernel address.
 */
static long (* const bpf_kallsyms_lookup_name)(const char *name, int name_sz, int
flags, __u64 *res) = (void *) 179;

/*
 * bpf_find_vma
 *
 * Find vma of *task* that contains *addr*, call *callback_fn*
 * function with *task*, *vma*, and *callback_ctx*.
 * The *callback_fn* should be a static function and
 * the *callback_ctx* should be a pointer to the stack.
 * The *flags* is used to control certain aspects of the helper.
 * Currently, the *flags* must be 0.
 *
 * The expected callback signature is
 *
 * long (*callback_fn)(struct task_struct *task, struct vm_area_struct *vma, void *callback_ctx);
 *
 * Returns
 * 0 on success.
 * **ENOENT** if *task->mm* is NULL, or no vma contains *addr*.

```

```

* ** -EBUSY** if failed to try lock mmap_lock.
* ** -EINVAL** for invalid **flags**.
*/
static long (* const bpf_find_vma)(struct task_struct *task, __u64 addr, void *callback_fn, void *callback_ctx,
__u64 flags) = (void *) 180;

/*
* bpf_loop
*
* For **nr_loops**, call **callback_fn** function
* with **callback_ctx** as the context parameter.
* The
**callback_fn** should be a static function and
* the **callback_ctx** should be a pointer to the stack.
* The **flags** is used to control certain aspects of the helper.
* Currently, the **flags** must be 0. Currently, nr_loops is
* limited to 1 << 23 (~8 million) loops.
*
* long (*callback_fn)(u32 index, void *ctx);
*
* where **index** is the current index in the loop. The index
* is zero-indexed.
*
* If **callback_fn** returns 0, the helper will continue to the next
* loop. If return value is 1, the helper will skip the rest of
* the loops and return. Other return values are not used now,
* and will be rejected by the verifier.
*
*
* Returns
* The number of loops performed, ** -EINVAL** for invalid **flags**,
* ** -E2BIG** if **nr_loops** exceeds the maximum number of loops.
*/
static long (* const bpf_loop)(__u32 nr_loops, void *callback_fn, void *callback_ctx, __u64 flags) = (void *) 181;

/*
* bpf_strncmp
*
* Do strncmp() between **s1** and
**s2**. **s1** doesn't need
* to be null-terminated and **s1_sz** is the maximum storage
* size of **s1**. **s2** must be a read-only string.
*
* Returns
* An integer less than, equal to, or greater than zero
* if the first **s1_sz** bytes of **s1** is found to be
* less than, to match, or be greater than **s2**.
*/

```

```

static long (* const bpf_strncmp)(const char *s1, __u32 s1_sz, const char *s2) = (void *) 182;

/*
 * bpf_get_func_arg
 *
 * Get **n**-th argument register (zero based) of the traced function (for tracing programs)
 * returned in **value**.
 *
 *
 * Returns
 * 0 on success.
 * **EINVAL** if n >= argument register count of traced function.
 */
static long (* const bpf_get_func_arg)(void *ctx, __u32 n, __u64 *value) = (void *) 183;

/*
 * bpf_get_func_ret
 *
 * Get return value of the traced function (for tracing programs)
 * in **value**.
 *
 *
 * Returns
 * 0 on success.
 * **EOPNOTSUPP** for tracing programs other than BPF_TRACE_FEXIT or BPF_MODIFY_RETURN.
 */
static long (* const bpf_get_func_ret)(void *ctx, __u64 *value) = (void *) 184;

/*
 * bpf_get_func_arg_cnt
 *
 * Get number of registers of the traced function (for tracing programs) where
 * function arguments are stored in these registers.
 *
 *
 * Returns
 * The number of argument registers of the traced function.
 */
static long (* const bpf_get_func_arg_cnt)(void *ctx) = (void *) 185;

/*
 * bpf_get_retval
 *
 * Get the BPF program's return value that will be returned to the upper layers.
 *
 *
 * This helper is currently supported by cgroup programs and only by the hooks
 * where BPF program's return value is returned to the userspace via errno.
 */

```

```

* Returns
* The BPF program's return value.
*/
static int (* const bpf_get_retval)(void) = (void *) 186;

/*
* bpf_set_retval
*
* Set the BPF program's return value that will be returned to the upper layers.
*
* This helper is currently supported by cgroup programs and only by the hooks
* where BPF program's
* return value is returned to the userspace via errno.
*
* Note that there is the following corner case where the program exports an error
* via bpf_set_retval but signals success via 'return 1':
*
* bpf_set_retval(-EPERM);
* return 1;
*
* In this case, the BPF program's return value will use helper's -EPERM. This
* still holds true for cgroup/bind{4,6} which supports extra 'return 3' success case.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static int (* const bpf_set_retval)(int retval) = (void *) 187;

/*
* bpf_xdp_get_buff_len
*
* Get the total size of a given xdp buff (linear and paged area)
*
* Returns
* The total size of a given xdp buffer.
*/
static __u64 (* const bpf_xdp_get_buff_len)(struct xdp_md *xdp_md) = (void *) 188;

/*
* bpf_xdp_load_bytes
*
* This helper is provided as an easy way to load data from a
* xdp buffer. It can be used to load *len* bytes from *offset* from
* the frame associated to *xdp_md*,
* into the buffer pointed by
* *buf*.
*

```

```

* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_xdp_load_bytes)(struct xdp_md *xdp_md, __u32 offset, void *buf, __u32 len) = (void *)
189;

/*
* bpf_xdp_store_bytes
*
* Store *len* bytes from buffer *buf* into the frame
* associated to *xdp_md*, at *offset*.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_xdp_store_bytes)(struct xdp_md *xdp_md, __u32 offset, void *buf, __u32 len) = (void *)
190;

/*
* bpf_copy_from_user_task
*
* Read *size* bytes from user space address *user_ptr* in *tsk*'s
* address space, and stores the data in *dst*. *flags* is not
* used yet and is provided for future extensibility. This helper
* can only be used by sleepable programs.
*
* Returns
* 0 on success, or a negative error in case of failure. On error
* *dst* buffer is zeroed out.
*/
static long (* const bpf_copy_from_user_task)(void *dst,
__u32 size, const void *user_ptr, struct task_struct *tsk, __u64 flags) = (void *) 191;

/*
* bpf_skb_set_timestamp
*
* Change the __sk_buff->timestamp_type to *timestamp_type*
* and set *timestamp* to the __sk_buff->timestamp together.
*
* If there is no need to change the __sk_buff->timestamp_type,
* the timestamp value can be directly written to __sk_buff->timestamp
* instead.
*
* BPF_SKB_TIMESTAMP_DELIVERY_MONO is the only timestamp that
* will be kept during bpf_redirect_*. A non zero
* *timestamp* must be used with the BPF_SKB_TIMESTAMP_DELIVERY_MONO
* *timestamp_type*.
*

```

```

* A BPF_SKB_TIMESTAMP_UNSPEC *timestamp_type* can only be used
* with a zero *timestamp*.
*
* Only IPv4 and IPv6 skb->protocol are supported.
*
* This function is most useful when it needs to set a
* mono delivery time to __sk_buff->timestamp and then
* bpf_redirect_*(*) to the egress of an iface. For example,
* changing the (rcv) timestamp in __sk_buff->timestamp at
* ingress to a mono delivery time and then bpf_redirect_*(*)
* to sch_fq@phy-dev.
*
* Returns
* 0 on success.
* **EINVAL** for invalid input
* **EOPNOTSUPP** for unsupported protocol
*/
static long (* const bpf_skb_set_timestamp)(struct __sk_buff *skb, __u64 timestamp, __u32 timestamp_type) = (void *) 192;

/*
* bpf_ima_file_hash
*
* Returns a calculated IMA hash of the *file*.
* If the hash is larger than *size*, then only *size*
* bytes will be copied to *dst*
*
* Returns
* The **hash_algo** is returned on success,
* **EOPNOTSUPP** if the hash calculation failed or **EINVAL** if
* invalid arguments are passed.
*/
static long (* const bpf_ima_file_hash)(struct file *file, void *dst, __u32 size) = (void *) 193;

/*
* bpf_kptr_xchg
*
* Exchange kptr at pointer *map_value* with *ptr*, and return the
* old value. *ptr* can be NULL, otherwise it must be a referenced
* pointer which will be released when this helper is called.
*
* Returns
* The old value of kptr (which can be NULL). The returned pointer
* if
* not NULL, is a reference which must be released using its
* corresponding release function, or moved into a BPF map before
* program exit.
*/
static void *(* const bpf_kptr_xchg)(void *map_value, void *ptr) = (void *) 194;

```

```

/*
 * bpf_map_lookup_percpu_elem
 *
 * Perform a lookup in *percpu map* for an entry associated to
 * *key* on *cpu*.
 *
 * Returns
 * Map value associated to *key* on *cpu*, or **NULL** if no entry
 * was found or *cpu* is invalid.
 */
static void *(* const bpf_map_lookup_percpu_elem)(void *map, const void *key, __u32 cpu) = (void *) 195;

/*
 * bpf_skc_to_mptcp_sock
 *
 * Dynamically cast a *sk* pointer to a *mptcp_sock* pointer.
 *
 * Returns
 * *sk* if casting is valid, or **NULL** otherwise.
 */
static struct mptcp_sock *(* const bpf_skc_to_mptcp_sock)(void *sk) = (void *) 196;

/*
 * bpf_dynptr_from_mem
 *
 * Get a dynptr to local memory *data*.
 *
 * *data* must be a ptr to a map value.
 * The maximum *size* supported is DYNPTR_MAX_SIZE.
 * *flags*
 * is currently unused.
 *
 * Returns
 * 0 on success, -E2BIG if the size exceeds DYNPTR_MAX_SIZE,
 * -EINVAL if flags is not 0.
 */
static long (* const bpf_dynptr_from_mem)(void *data, __u32 size, __u64 flags, struct bpf_dynptr *ptr) = (void *)
197;

/*
 * bpf_ringbuf_reserve_dynptr
 *
 * Reserve *size* bytes of payload in a ring buffer *ringbuf*
 * through the dynptr interface. *flags* must be 0.
 *
 * Please note that a corresponding bpf_ringbuf_submit_dynptr or
 * bpf_ringbuf_discard_dynptr must be called on *ptr*, even if the

```



```

* reservation fails. This is enforced by the verifier.
*
* Returns
* 0 on success, or a negative error in case of failure.
*/
static long (* const bpf_ringbuf_reserve_dynptr)(void *ringbuf, __u32 size, __u64 flags, struct bpf_dynptr *ptr) =
(void *) 198;

/*
* bpf_ringbuf_submit_dynptr
*
* Submit reserved ring buffer sample, pointed to by *data*,
* through the dynptr interface. This is a no-op if the dynptr is
* invalid/null.
*
*
For more information on *flags*, please see
* 'bpf_ringbuf_submit'.
*
* Returns
* Nothing. Always succeeds.
*/
static void (* const bpf_ringbuf_submit_dynptr)(struct bpf_dynptr *ptr, __u64 flags) = (void *) 199;

/*
* bpf_ringbuf_discard_dynptr
*
* Discard reserved ring buffer sample through the dynptr
* interface. This is a no-op if the dynptr is invalid/null.
*
* For more information on *flags*, please see
* 'bpf_ringbuf_discard'.
*
* Returns
* Nothing. Always succeeds.
*/
static void (* const bpf_ringbuf_discard_dynptr)(struct bpf_dynptr *ptr, __u64 flags) = (void *) 200;

/*
* bpf_dynptr_read
*
* Read *len* bytes from *src* into *dst*, starting from *offset*
* into *src*.
* *flags* is currently unused.
*
* Returns
* 0 on success, -E2BIG if *offset* + *len* exceeds the length
* of *src*'s data, -EINVAL if *src* is an invalid dynptr or if

```

```

* *flags* is not 0.
*/
static long (* const bpf_dynptr_read)(void *dst, __u32 len, const struct bpf_dynptr
*src, __u32 offset, __u64 flags) = (void *) 201;

/*
* bpf_dynptr_write
*
* Write *len* bytes from *src* into *dst*, starting from *offset*
* into *dst*.
*
* *flags* must be 0 except for skb-type dynptrs.
*
* For skb-type dynptrs:
*   * All data slices of the dynptr are automatically
*     invalidated after **bpf_dynptr_write**\ (). This is
*     because writing may pull the skb and change the
*     underlying packet buffer.
*
*   * For *flags*, please see the flags accepted by
*     **bpf_skb_store_bytes**\ ().
*
* Returns
* 0 on success, -E2BIG if *offset* + *len* exceeds the length
* of *dst*'s data, -EINVAL if *dst* is an invalid dynptr or if *dst*
* is a read-only dynptr or if *flags* is not correct. For skb-type dynptrs,
* other errors correspond to errors returned by **bpf_skb_store_bytes**\ ().
*/
static long (* const bpf_dynptr_write)(const struct bpf_dynptr *dst, __u32 offset, void *src, __u32 len, __u64 flags)
= (void *)
202;

/*
* bpf_dynptr_data
*
* Get a pointer to the underlying dynptr data.
*
* *len* must be a statically known value. The returned data slice
* is invalidated whenever the dynptr is invalidated.
*
* skb and xdp type dynptrs may not use bpf_dynptr_data. They should
* instead use bpf_dynptr_slice and bpf_dynptr_slice_rdwr.
*
* Returns
* Pointer to the underlying dynptr data, NULL if the dynptr is
* read-only, if the dynptr is invalid, or if the offset and length
* is out of bounds.
*/

```

```

static void (* const bpf_dynptr_data)(const struct bpf_dynptr *ptr, __u32 offset, __u32 len) = (void *) 203;

/*
 * bpf_tcp_raw_gen_syncookie_ipv4
 *
 * Try to issue a SYN cookie for the packet with corresponding
 * IPv4/TCP headers, *iph* and *th*, without depending on a
 * listening socket.
 *
 * *iph* points to the IPv4 header.
 *
 * *th* points to the start of the TCP header, while *th_len*
 * contains the length of the TCP header (at least
 * **sizeof**\ (**struct tcphdr**)).
 *
 * Returns
 * On success, lower 32 bits hold the generated SYN cookie in
 * followed by 16 bits which hold the MSS value for that cookie,
 * and the top 16 bits are unused.
 *
 * On failure, the returned value is one of the following:
 *
 * **-EINVAL** if *th_len* is invalid.
 */
static __s64 (* const bpf_tcp_raw_gen_syncookie_ipv4)(struct iphdr *iph, struct tcphdr *th, __u32 th_len) = (void *) 204;

/*
 * bpf_tcp_raw_gen_syncookie_ipv6
 *
 * Try to issue a SYN cookie for the packet with corresponding
 * IPv6/TCP headers, *iph* and *th*, without depending on a
 * listening socket.
 *
 * *iph* points to the IPv6 header.
 *
 * *th* points to the start of the TCP header, while *th_len*
 * contains the length of the TCP header (at least
 * **sizeof**\ (**struct tcphdr**)).
 *
 * Returns
 * On success, lower 32 bits hold the generated SYN cookie in
 * followed by 16 bits which hold the MSS value for that cookie,
 * and the top 16 bits are unused.
 *
 * On failure,
 * the returned value is one of the following:
 *

```

```

* **EINVAL** if *th_len* is invalid.
*
* **-EPROTONOSUPPORT** if CONFIG_IPV6 is not builtin.
*/
static __s64 (* const bpf_tcp_raw_gen_syncookie_ipv6)(struct ipv6hdr *iph, struct tcphdr *th, __u32 th_len) =
(void *) 205;

/*
* bpf_tcp_raw_check_syncookie_ipv4
*
* Check whether *iph* and *th* contain a valid SYN cookie ACK
* without depending on a listening socket.
*
* *iph* points to the IPv4 header.
*
* *th* points to the TCP header.
*
* Returns
* 0 if *iph* and *th* are a valid SYN cookie ACK.
*
* On failure, the returned value is one of the following:
*
* **-EACCES** if the SYN cookie is not valid.
*/
static long (* const bpf_tcp_raw_check_syncookie_ipv4)(struct iphdr *iph, struct tcphdr *th) = (void *) 206;

/*
* bpf_tcp_raw_check_syncookie_ipv6
*
* Check whether *iph* and *th* contain a valid SYN cookie ACK
* without depending on a listening socket.
*
* *iph* points to the IPv6 header.
*
* *th*
points to the TCP header.
*
* Returns
* 0 if *iph* and *th* are a valid SYN cookie ACK.
*
* On failure, the returned value is one of the following:
*
* **-EACCES** if the SYN cookie is not valid.
*
* **-EPROTONOSUPPORT** if CONFIG_IPV6 is not builtin.
*/
static long (* const bpf_tcp_raw_check_syncookie_ipv6)(struct ipv6hdr *iph, struct tcphdr *th) = (void *) 207;

```

```

/*
 * bpf_ktime_get_tai_ns
 *
 * A nonsettable system-wide clock derived from wall-clock time but
 * ignoring leap seconds. This clock does not experience
 * discontinuities and backwards jumps caused by NTP inserting leap
 * seconds as CLOCK_REALTIME does.
 *
 * See: **clock_gettime**\ (**CLOCK_TAI**)
 *
 * Returns
 * Current *ktime*.
 */
static __u64 (* const bpf_ktime_get_tai_ns)(void) = (void *) 208;

/*
 * bpf_user_ringbuf_drain
 *
 * Drain samples from the specified user ring buffer, and invoke
 * the provided callback for each such sample:
 *
 * long (*callback_fn)(const struct bpf_dynptr *dynptr,
 * void *ctx);
 *
 * If **callback_fn** returns 0, the helper will continue to try
 * and drain the next sample, up to a maximum of
 * BPF_MAX_USER_RINGBUF_SAMPLES samples. If the return value is 1,
 * the helper will skip the rest of the samples and return. Other
 * return values are not used now, and will be rejected by the
 * verifier.
 *
 * Returns
 * The number of drained samples if no error was encountered while
 * draining samples, or 0 if no samples were present in the ring
 * buffer. If a user-space producer was epoll-waiting on this map,
 * and at least one sample was drained, they will receive an event
 * notification notifying them of available space in the ring
 * buffer. If the BPF_RB_NO_WAKEUP flag is passed to this
 * function, no wakeup notification will be sent. If the
 * BPF_RB_FORCE_WAKEUP flag is passed, a wakeup notification will
 * be sent even if no sample was drained.
 *
 * On failure, the returned value is one of the following:
 *
 * **-EBUSY**
 *
 * if the ring buffer is contended, and another calling
 * context was concurrently draining the ring buffer.
 */

```

```

* **EINVAL** if user-space is not properly tracking the ring
* buffer due to the producer position not being aligned to 8
* bytes, a sample not being aligned to 8 bytes, or the producer
* position not matching the advertised length of a sample.
*
* **E2BIG** if user-space has tried to publish a sample which is
* larger than the size of the ring buffer, or which cannot fit
* within a struct bpf_dynptr.
*/
static long (* const bpf_user_ringbuf_drain)(void *map, void *callback_fn, void *ctx, __u64 flags) = (void *) 209;

/*
* bpf_cgrp_storage_get
*
* Get a bpf_local_storage from the *cgroup*.
*
* Logically, it could be thought of as getting the value from
* a *map* with *cgroup* as the **key**. From this
* perspective, the usage is not much different from
* **bpf_map_lookup_elem**(*map*, **&***cgroup*) except this
* helper enforces the
key must be a cgroup struct and the map must also
* be a **BPF_MAP_TYPE_CGRP_STORAGE**.
*
* In reality, the local-storage value is embedded directly inside of the
* *cgroup* object itself, rather than being located in the
* **BPF_MAP_TYPE_CGRP_STORAGE** map. When the local-storage value is
* queried for some *map* on a *cgroup* object, the kernel will perform an
* O(n) iteration over all of the live local-storage values for that
* *cgroup* object until the local-storage value for the *map* is found.
*
* An optional *flags* (**BPF_LOCAL_STORAGE_GET_F_CREATE**) can be
* used such that a new bpf_local_storage will be
* created if one does not exist. *value* can be used
* together with **BPF_LOCAL_STORAGE_GET_F_CREATE** to specify
* the initial value of a bpf_local_storage. If *value* is
* **NULL**, the new bpf_local_storage will be zero initialized.
*
* Returns
* A bpf_local_storage pointer is returned on success.
*
* **NULL** if not found or there was
an error in adding
* a new bpf_local_storage.
*/
static void (* const bpf_cgrp_storage_get)(void *map, struct cgroup *cgroup, void *value, __u64 flags) = (void *)
210;

```

```

/*
 * bpf_cgrp_storage_delete
 *
 * Delete a bpf_local_storage from a *cgroup*.
 *
 * Returns
 * 0 on success.
 *
 * ** -ENOENT ** if the bpf_local_storage cannot be found.
 */
static long (* const bpf_cgrp_storage_delete)(void *map, struct cgroup *cgroup) = (void *) 211;
ISC License

```

Copyright (c) 2012-2016 Dave Collins <dave@davec.name>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 2014, Evan Phoenix
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the Evan Phoenix nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR

SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/* SPDX-License-Identifier: (LGPL-2.1 OR BSD-2-Clause) */

#ifndef __BPF_TRACING_H__

#define __BPF_TRACING_H__

#include "bpf_helpers.h"

/* Scan the ARCH passed in from ARCH env variable (see Makefile) */

#if defined(__TARGET_ARCH_x86)

#define bpf_target_x86

#define bpf_target_defined

#elif defined(__TARGET_ARCH_s390)

#define bpf_target_s390

#define bpf_target_defined

#elif defined(__TARGET_ARCH_arm)

#define bpf_target_arm

#define bpf_target_defined

#elif defined(__TARGET_ARCH_arm64)

#define bpf_target_arm64

#define bpf_target_defined

#elif defined(__TARGET_ARCH_mips)

#define bpf_target_mips

#define bpf_target_defined

#elif defined(__TARGET_ARCH_powerpc)

#define bpf_target_powerpc

#define bpf_target_defined

#elif defined(__TARGET_ARCH_sparc)

#define bpf_target_sparc

#define bpf_target_defined

#elif defined(__TARGET_ARCH_riscv)

#define bpf_target_riscv

#define bpf_target_defined

#elif defined(__TARGET_ARCH_arc)

#define bpf_target_arc

#define bpf_target_defined

#elif defined(__TARGET_ARCH_loongarch)

#define

bpf_target_loongarch

#define bpf_target_defined

#else

/* Fall back to what the compiler says */

#if defined(__x86_64__)

#define bpf_target_x86

#define bpf_target_defined


```

#elif defined(__s390__)
#define bpf_target_s390
#define bpf_target_defined
#elif defined(__arm__)
#define bpf_target_arm
#define bpf_target_defined
#elif defined(__aarch64__)
#define bpf_target_arm64
#define bpf_target_defined
#elif defined(__mips__)
#define bpf_target_mips
#define bpf_target_defined
#elif defined(__powerpc__)
#define bpf_target_powerpc
#define bpf_target_defined
#elif defined(__sparc__)
#define bpf_target_sparc
#define bpf_target_defined
#elif defined(__riscv) && __riscv_xlen == 64
#define bpf_target_riscv
#define bpf_target_defined
#elif defined(__arc__)
#define bpf_target_arc
#define bpf_target_defined
#elif defined(__loongarch__)
#define bpf_target_loongarch
#define bpf_target_defined
#endif /* no compiler target */

#endif

#ifndef __BPF_TARGET_MISSING
#define
__BPF_TARGET_MISSING "GCC error \"Must specify a BPF target arch via __TARGET_ARCH_XXX\""
#endif

#if defined(bpf_target_x86)

/*
 * https://en.wikipedia.org/wiki/X86_calling_conventions#System_V_AMD64_ABI
 */

#if defined(__KERNEL__) || defined(__VMLINUX_H__)

#define __PT_PARM1_REG di
#define __PT_PARM2_REG si
#define __PT_PARM3_REG dx
#define __PT_PARM4_REG cx

```

```

#define __PT_PARM5_REG r8
#define __PT_PARM6_REG r9
/*
 * Syscall uses r10 for PARM4. See arch/x86/entry/entry_64.S:entry_SYSCALL_64
 * comments in Linux sources. And refer to syscall(2) manpage.
 */
#define __PT_PARM1_SYSCALL_REG __PT_PARM1_REG
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG r10
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG __PT_PARM6_REG

#define __PT_RET_REG sp
#define __PT_FP_REG bp
#define __PT_RC_REG ax
#define __PT_SP_REG sp
#define __PT_IP_REG ip

#else

#ifdef __i386__

/*
 * i386 kernel is built with -mregparm=3 */
#define __PT_PARM1_REG eax
#define __PT_PARM2_REG edx
#define __PT_PARM3_REG ecx
/* i386 syscall ABI is very different, refer to syscall(2) manpage */
#define __PT_PARM1_SYSCALL_REG ebx
#define __PT_PARM2_SYSCALL_REG ecx
#define __PT_PARM3_SYSCALL_REG edx
#define __PT_PARM4_SYSCALL_REG esi
#define __PT_PARM5_SYSCALL_REG edi
#define __PT_PARM6_SYSCALL_REG ebp

#define __PT_RET_REG esp
#define __PT_FP_REG ebp
#define __PT_RC_REG eax
#define __PT_SP_REG esp
#define __PT_IP_REG eip

#else /* __i386__ */

#define __PT_PARM1_REG rdi
#define __PT_PARM2_REG rsi
#define __PT_PARM3_REG rdx
#define __PT_PARM4_REG rcx

```

```

#define __PT_PARM5_REG r8
#define __PT_PARM6_REG r9

#define __PT_PARM1_SYSCALL_REG __PT_PARM1_REG
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG r10
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG __PT_PARM6_REG

#define __PT_RET_REG rsp
#define
__PT_FP_REG rbp
#define __PT_RC_REG rax
#define __PT_SP_REG rsp
#define __PT_IP_REG rip

#endif /* __i386__ */

#endif /* __KERNEL__ || __VMLINUX_H__ */

#elif defined(bpf_target_s390)

/*
 * https://github.com/IBM/s390x-abi/releases/download/v1.6/lzsabi_s390x.pdf
 */

struct pt_regs__s390 {
    unsigned long orig_gpr2;
};

/* s390 provides user_pt_regs instead of struct pt_regs to userspace */
#define __PT_REGS_CAST(x) ((const user_pt_regs *) (x))
#define __PT_PARM1_REG gprs[2]
#define __PT_PARM2_REG gprs[3]
#define __PT_PARM3_REG gprs[4]
#define __PT_PARM4_REG gprs[5]
#define __PT_PARM5_REG gprs[6]

#define __PT_PARM1_SYSCALL_REG orig_gpr2
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG gprs[7]
#define PT_REGS_PARM1_SYSCALL(x) PT_REGS_PARM1_CORE_SYSCALL(x)
#define PT_REGS_PARM1_CORE_SYSCALL(x) \
    BPF_CORE_READ((const

```

```

struct pt_regs__s390 *) (x), __PT_PARM1_SYSCALL_REG)

#define __PT_RET_REG gprs[14]
#define __PT_FP_REG gprs[11] /* Works only with CONFIG_FRAME_POINTER */
#define __PT_RC_REG gprs[2]
#define __PT_SP_REG gprs[15]
#define __PT_IP_REG psw.addr

#elif defined(bpf_target_arm)

/*
 * https://github.com/ARM-software/abi-aa/blob/main/aapcs32/aapcs32.rst#machine-registers
 */

#define __PT_PARM1_REG uregs[0]
#define __PT_PARM2_REG uregs[1]
#define __PT_PARM3_REG uregs[2]
#define __PT_PARM4_REG uregs[3]

#define __PT_PARM1_SYSCALL_REG __PT_PARM1_REG
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG uregs[4]
#define __PT_PARM6_SYSCALL_REG uregs[5]
#define __PT_PARM7_SYSCALL_REG uregs[6]

#define __PT_RET_REG uregs[14]
#define __PT_FP_REG uregs[11] /* Works only with CONFIG_FRAME_POINTER */
#define __PT_RC_REG uregs[0]
#define __PT_SP_REG uregs[13]
#define __PT_IP_REG uregs[12]

#elif
defined(bpf_target_arm64)

/*
 * https://github.com/ARM-software/abi-aa/blob/main/aapcs64/aapcs64.rst#machine-registers
 */

struct pt_regs__arm64 {
    unsigned long orig_x0;
};

/* arm64 provides struct user_pt_regs instead of struct pt_regs to userspace */
#define __PT_REGS_CAST(x) ((const struct user_pt_regs *) (x))
#define __PT_PARM1_REG regs[0]
#define __PT_PARM2_REG regs[1]

```

```

#define __PT_PARM3_REG regs[2]
#define __PT_PARM4_REG regs[3]
#define __PT_PARM5_REG regs[4]
#define __PT_PARM6_REG regs[5]
#define __PT_PARM7_REG regs[6]
#define __PT_PARM8_REG regs[7]

#define __PT_PARM1_SYSCALL_REG orig_x0
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG __PT_PARM6_REG
#define PT_REGS_PARM1_SYSCALL(x) PT_REGS_PARM1_CORE_SYSCALL(x)
#define PT_REGS_PARM1_CORE_SYSCALL(x) \
    BPF_CORE_READ((const struct pt_regs __arm64 *) (x), \
        __PT_PARM1_SYSCALL_REG)

#define __PT_RET_REG regs[30]
#define __PT_FP_REG regs[29] /* Works only with CONFIG_FRAME_POINTER */
#define __PT_RC_REG regs[0]
#define __PT_SP_REG sp
#define __PT_IP_REG pc

#elif defined(bpf_target_mips)

/*
 * N64 ABI is assumed right now.
 * https://en.wikipedia.org/wiki/MIPS_architecture#Calling_conventions
 */

#define __PT_PARM1_REG regs[4]
#define __PT_PARM2_REG regs[5]
#define __PT_PARM3_REG regs[6]
#define __PT_PARM4_REG regs[7]
#define __PT_PARM5_REG regs[8]
#define __PT_PARM6_REG regs[9]
#define __PT_PARM7_REG regs[10]
#define __PT_PARM8_REG regs[11]

#define __PT_PARM1_SYSCALL_REG __PT_PARM1_REG
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG /* only N32/N64 */
#define __PT_PARM6_SYSCALL_REG __PT_PARM6_REG /* only N32/N64 */

#define __PT_RET_REG regs[31]

```

```

#define __PT_FP_REG regs[30] /* Works only with CONFIG_FRAME_POINTER
*/
#define __PT_RC_REG regs[2]
#define __PT_SP_REG regs[29]
#define __PT_IP_REG cp0_epc

#elif defined(bpf_target_powerpc)

/*
* http://refspecs.linux-foundation.org/elf/elfspec\_ppc.pdf (page 3-14,
* section "Function Calling Sequence")
*/

#define __PT_PARM1_REG gpr[3]
#define __PT_PARM2_REG gpr[4]
#define __PT_PARM3_REG gpr[5]
#define __PT_PARM4_REG gpr[6]
#define __PT_PARM5_REG gpr[7]
#define __PT_PARM6_REG gpr[8]
#define __PT_PARM7_REG gpr[9]
#define __PT_PARM8_REG gpr[10]

/* powerpc does not select ARCH_HAS_SYSCALL_WRAPPER. */
#define PT_REGS_SYSCALL_REGS(ctx) ctx
#define __PT_PARM1_SYSCALL_REG orig_gpr3
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG __PT_PARM6_REG
#if !defined(__arch64__)
#define __PT_PARM7_SYSCALL_REG __PT_PARM7_REG /* only powerpc (not powerpc64) */
#endif

#define __PT_RET_REG regs[31]
#define
__PT_FP_REG __unsupported__
#define __PT_RC_REG gpr[3]
#define __PT_SP_REG sp
#define __PT_IP_REG nip

#elif defined(bpf_target_sparc)

/*
* https://en.wikipedia.org/wiki/Calling\_convention#SPARC
*/

#define __PT_PARM1_REG u_regs[UREG_I0]

```

```

#define __PT_PARM2_REG u_regs[UREG_I1]
#define __PT_PARM3_REG u_regs[UREG_I2]
#define __PT_PARM4_REG u_regs[UREG_I3]
#define __PT_PARM5_REG u_regs[UREG_I4]
#define __PT_PARM6_REG u_regs[UREG_I5]

#define __PT_PARM1_SYSCALL_REG __PT_PARM1_REG
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG __PT_PARM6_REG

#define __PT_RET_REG u_regs[UREG_I7]
#define __PT_FP_REG __unsupported__
#define __PT_RC_REG u_regs[UREG_I0]
#define __PT_SP_REG u_regs[UREG_FP]
/* Should this also be a bpf_target check for the sparc case? */
#if defined(__arch64__)
#define __PT_IP_REG tpc
#else
#define
__PT_IP_REG pc
#endif

#elif defined(bpf_target_riscv)

/*
 * https://github.com/riscv-non-isa/riscv-elf-psabi-doc/blob/master/riscv-cc.adoc#risc-v-calling-conventions
 */

/* riscv provides struct user_regs_struct instead of struct pt_regs to userspace */
#define __PT_REGS_CAST(x) ((const struct user_regs_struct*)(x))
#define __PT_PARM1_REG a0
#define __PT_PARM2_REG a1
#define __PT_PARM3_REG a2
#define __PT_PARM4_REG a3
#define __PT_PARM5_REG a4
#define __PT_PARM6_REG a5
#define __PT_PARM7_REG a6
#define __PT_PARM8_REG a7

#define __PT_PARM1_SYSCALL_REG __PT_PARM1_REG
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG __PT_PARM6_REG

```

```

#define __PT_RET_REG ra
#define __PT_FP_REG s0
#define __PT_RC_REG a0
#define __PT_SP_REG sp
#define __PT_IP_REG pc

#elif defined(bpf_target_arc)

/*
 * Section "Function Calling Sequence" (page
 24):
 * https://raw.githubusercontent.com/wiki/foss-for-synopsys-dwc-arc-processors/toolchain/files/ARCV2\_ABI.pdf
 */

/* arc provides struct user_regs_struct instead of struct pt_regs to userspace */
#define __PT_REGS_CAST(x) ((const struct user_regs_struct *) (x))
#define __PT_PARM1_REG scratch.r0
#define __PT_PARM2_REG scratch.r1
#define __PT_PARM3_REG scratch.r2
#define __PT_PARM4_REG scratch.r3
#define __PT_PARM5_REG scratch.r4
#define __PT_PARM6_REG scratch.r5
#define __PT_PARM7_REG scratch.r6
#define __PT_PARM8_REG scratch.r7

/* arc does not select ARCH_HAS_SYSCALL_WRAPPER. */
#define PT_REGS_SYSCALL_REGS(ctx) ctx
#define __PT_PARM1_SYSCALL_REG __PT_PARM1_REG
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG __PT_PARM6_REG

#define __PT_RET_REG scratch.blink
#define __PT_FP_REG scratch.fp
#define __PT_RC_REG scratch.r0
#define
  __PT_SP_REG scratch.sp
#define __PT_IP_REG scratch.ret

#elif defined(bpf_target_loongarch)

/*
 * https://docs.kernel.org/loongarch/introduction.html
 * https://loongson.github.io/LoongArch-Documentation/LoongArch-ELF-ABI-EN.html
 */

```



```

/* loongarch provides struct user_pt_regs instead of struct pt_regs to userspace */
#define __PT_REGS_CAST(x) ((const struct user_pt_regs *) (x))
#define __PT_PARM1_REG regs[4]
#define __PT_PARM2_REG regs[5]
#define __PT_PARM3_REG regs[6]
#define __PT_PARM4_REG regs[7]
#define __PT_PARM5_REG regs[8]
#define __PT_PARM6_REG regs[9]
#define __PT_PARM7_REG regs[10]
#define __PT_PARM8_REG regs[11]

/* loongarch does not select ARCH_HAS_SYSCALL_WRAPPER. */
#define PT_REGS_SYSCALL_REGS(ctx) ctx
#define __PT_PARM1_SYSCALL_REG __PT_PARM1_REG
#define __PT_PARM2_SYSCALL_REG __PT_PARM2_REG
#define __PT_PARM3_SYSCALL_REG __PT_PARM3_REG
#define __PT_PARM4_SYSCALL_REG __PT_PARM4_REG
#define __PT_PARM5_SYSCALL_REG __PT_PARM5_REG
#define __PT_PARM6_SYSCALL_REG
__PT_PARM6_REG

#define __PT_RET_REG regs[1]
#define __PT_FP_REG regs[22]
#define __PT_RC_REG regs[4]
#define __PT_SP_REG regs[3]
#define __PT_IP_REG csr_era

#endif

#if defined(bpf_target_defined)

struct pt_regs;

/* allow some architectures to override `struct pt_regs` */
#ifndef __PT_REGS_CAST
#define __PT_REGS_CAST(x) (x)
#endif

/*
 * Different architectures support different number of arguments passed
 * through registers. i386 supports just 3, some arches support up to 8.
 */
#ifndef __PT_PARM4_REG
#define __PT_PARM4_REG __unsupported__
#endif
#ifndef __PT_PARM5_REG
#define __PT_PARM5_REG __unsupported__

```

```

#endif
#ifndef __PT_PARM6_REG
#define __PT_PARM6_REG __unsupported__
#endif
#ifndef __PT_PARM7_REG
#define __PT_PARM7_REG __unsupported__
#endif
#ifndef __PT_PARM8_REG
#define __PT_PARM8_REG __unsupported__
#endif
/*
* Similarly, syscall-specific conventions might differ between function call
* conventions within each architecture. All supported architectures pass
* either
* 6 or 7 syscall arguments in registers.
*
* See syscall(2) manpage for succinct table with information on each arch.
*/
#ifndef __PT_PARM7_SYSCALL_REG
#define __PT_PARM7_SYSCALL_REG __unsupported__
#endif

#define PT_REGS_PARM1(x) (__PT_REGS_CAST(x)->__PT_PARM1_REG)
#define PT_REGS_PARM2(x) (__PT_REGS_CAST(x)->__PT_PARM2_REG)
#define PT_REGS_PARM3(x) (__PT_REGS_CAST(x)->__PT_PARM3_REG)
#define PT_REGS_PARM4(x) (__PT_REGS_CAST(x)->__PT_PARM4_REG)
#define PT_REGS_PARM5(x) (__PT_REGS_CAST(x)->__PT_PARM5_REG)
#define PT_REGS_PARM6(x) (__PT_REGS_CAST(x)->__PT_PARM6_REG)
#define PT_REGS_PARM7(x) (__PT_REGS_CAST(x)->__PT_PARM7_REG)
#define PT_REGS_PARM8(x) (__PT_REGS_CAST(x)->__PT_PARM8_REG)
#define PT_REGS_RET(x) (__PT_REGS_CAST(x)->__PT_RET_REG)
#define PT_REGS_FP(x) (__PT_REGS_CAST(x)->__PT_FP_REG)
#define PT_REGS_RC(x) (__PT_REGS_CAST(x)->__PT_RC_REG)
#define PT_REGS_SP(x) (__PT_REGS_CAST(x)->__PT_SP_REG)
#define PT_REGS_IP(x) (__PT_REGS_CAST(x)->__PT_IP_REG)

#define PT_REGS_PARM1_CORE(x)
BPF_CORE_READ(__PT_REGS_CAST(x), __PT_PARM1_REG)
#define PT_REGS_PARM2_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_PARM2_REG)
#define PT_REGS_PARM3_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_PARM3_REG)
#define PT_REGS_PARM4_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_PARM4_REG)
#define PT_REGS_PARM5_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_PARM5_REG)
#define PT_REGS_PARM6_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_PARM6_REG)
#define PT_REGS_PARM7_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_PARM7_REG)
#define PT_REGS_PARM8_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_PARM8_REG)
#define PT_REGS_RET_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_RET_REG)
#define PT_REGS_FP_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_FP_REG)
#define PT_REGS_RC_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_RC_REG)

```

```

#define PT_REGS_SP_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_SP_REG)
#define PT_REGS_IP_CORE(x) BPF_CORE_READ(__PT_REGS_CAST(x), __PT_IP_REG)

#if defined(bpf_target_powerpc)

#define
BPF_KPROBE_READ_RET_IP(ip, ctx) ({ (ip) = (ctx)->link; })
#define BPF_KRETPROBE_READ_RET_IP BPF_KPROBE_READ_RET_IP

#elif defined(bpf_target_sparc)

#define BPF_KPROBE_READ_RET_IP(ip, ctx) ({ (ip) = PT_REGS_RET(ctx); })
#define BPF_KRETPROBE_READ_RET_IP BPF_KPROBE_READ_RET_IP

#else

#define BPF_KPROBE_READ_RET_IP(ip, ctx) \
({ bpf_probe_read_kernel(&(ip), sizeof(ip), (void *)PT_REGS_RET(ctx)); })
#define BPF_KRETPROBE_READ_RET_IP(ip, ctx) \
({ bpf_probe_read_kernel(&(ip), sizeof(ip), (void *) (PT_REGS_FP(ctx) + sizeof(ip))); })

#endif

#ifndef PT_REGS_PARM1_SYSCALL
#define PT_REGS_PARM1_SYSCALL(x) (__PT_REGS_CAST(x)->__PT_PARM1_SYSCALL_REG)
#define PT_REGS_PARM1_CORE_SYSCALL(x) BPF_CORE_READ(__PT_REGS_CAST(x), \
__PT_PARM1_SYSCALL_REG)
#endif

#ifndef PT_REGS_PARM2_SYSCALL
#define PT_REGS_PARM2_SYSCALL(x) (__PT_REGS_CAST(x)->__PT_PARM2_SYSCALL_REG)
#define PT_REGS_PARM2_CORE_SYSCALL(x) BPF_CORE_READ(__PT_REGS_CAST(x), \
__PT_PARM2_SYSCALL_REG)
#endif

#ifndef PT_REGS_PARM3_SYSCALL
#define PT_REGS_PARM3_SYSCALL(x) (__PT_REGS_CAST(x)->__PT_PARM3_SYSCALL_REG)
#define PT_REGS_PARM3_CORE_SYSCALL(x) BPF_CORE_READ(__PT_REGS_CAST(x), \
__PT_PARM3_SYSCALL_REG)
#endif

#ifndef PT_REGS_PARM4_SYSCALL
#define PT_REGS_PARM4_SYSCALL(x) (__PT_REGS_CAST(x)->__PT_PARM4_SYSCALL_REG)
#define PT_REGS_PARM4_CORE_SYSCALL(x) BPF_CORE_READ(__PT_REGS_CAST(x), \
__PT_PARM4_SYSCALL_REG)
#endif

#ifndef PT_REGS_PARM5_SYSCALL
#define PT_REGS_PARM5_SYSCALL(x) (__PT_REGS_CAST(x)->__PT_PARM5_SYSCALL_REG)
#define PT_REGS_PARM5_CORE_SYSCALL(x) BPF_CORE_READ(__PT_REGS_CAST(x), \
__PT_PARM5_SYSCALL_REG)

```

```

#endif
#ifndef PT_REGS_PARM6_SYSCALL
#define PT_REGS_PARM6_SYSCALL(x) (__PT_REGS_CAST(x)->__PT_PARM6_SYSCALL_REG)
#define PT_REGS_PARM6_CORE_SYSCALL(x) BPF_CORE_READ(__PT_REGS_CAST(x),
__PT_PARM6_SYSCALL_REG)
#endif
#ifndef PT_REGS_PARM7_SYSCALL
#define PT_REGS_PARM7_SYSCALL(x) (__PT_REGS_CAST(x)->__PT_PARM7_SYSCALL_REG)
#define PT_REGS_PARM7_CORE_SYSCALL(x) BPF_CORE_READ(__PT_REGS_CAST(x),
__PT_PARM7_SYSCALL_REG)
#endif

#else /* defined(bpf_target_defined) */

#define PT_REGS_PARM1(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM2(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM3(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM4(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM5(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM6(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM7(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM8(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_RET(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_FP(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_RC(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_SP(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_IP(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })

#define PT_REGS_PARM1_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l;
})
#define PT_REGS_PARM2_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM3_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM4_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM5_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM6_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM7_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM8_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_RET_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_FP_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_RC_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_SP_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_IP_CORE(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })

#define BPF_KPROBE_READ_RET_IP(ip, ctx) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define BPF_KRETPROBE_READ_RET_IP(ip, ctx) ({ _Pragma(__BPF_TARGET_MISSING);
0l; })

#define PT_REGS_PARM1_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })

```

```

#define PT_REGS_PARM2_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM3_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM4_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM5_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM6_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM7_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })

#define PT_REGS_PARM1_CORE_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM2_CORE_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM3_CORE_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM4_CORE_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM5_CORE_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define PT_REGS_PARM6_CORE_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })
#define
PT_REGS_PARM7_CORE_SYSCALL(x) ({ _Pragma(__BPF_TARGET_MISSING); 0l; })

#endif /* defined(bpf_target_defined) */

/*
 * When invoked from a syscall handler kprobe, returns a pointer to a
 * struct pt_regs containing syscall arguments and suitable for passing to
 * PT_REGS_PARMn_SYSCALL() and PT_REGS_PARMn_CORE_SYSCALL().
 */
#ifndef PT_REGS_SYSCALL_REGS
/* By default, assume that the arch selects ARCH_HAS_SYSCALL_WRAPPER. */
#define PT_REGS_SYSCALL_REGS(ctx) ((struct pt_regs *)PT_REGS_PARM1(ctx))
#endif

#ifndef __bpf_concat
#define __bpf_concat(a, b) a ## b
#endif
#ifndef __bpf_apply
#define __bpf_apply(fn, n) __bpf_concat(fn, n)
#endif
#ifndef __bpf_nth
#define __bpf_nth(_, _1, _2, _3, _4, _5, _6, _7, _8, _9, _a, _b, _c, N, ...) N
#endif
#ifndef __bpf_narg
#define __bpf_narg(...) __bpf_nth(_, ##__VA_ARGS__, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0)
#endif

#define __bpf_ctx_cast0()      ctx
#define __bpf_ctx_cast1(x)    __bpf_ctx_cast0(), (void *)ctx[0]
#define
__bpf_ctx_cast2(x, args...) __bpf_ctx_cast1(args), (void *)ctx[1]
#define __bpf_ctx_cast3(x, args...) __bpf_ctx_cast2(args), (void *)ctx[2]
#define __bpf_ctx_cast4(x, args...) __bpf_ctx_cast3(args), (void *)ctx[3]
#define __bpf_ctx_cast5(x, args...) __bpf_ctx_cast4(args), (void *)ctx[4]

```

```

#define __bpf_ctx_cast6(x, args...) __bpf_ctx_cast5(args), (void *)ctx[5]
#define __bpf_ctx_cast7(x, args...) __bpf_ctx_cast6(args), (void *)ctx[6]
#define __bpf_ctx_cast8(x, args...) __bpf_ctx_cast7(args), (void *)ctx[7]
#define __bpf_ctx_cast9(x, args...) __bpf_ctx_cast8(args), (void *)ctx[8]
#define __bpf_ctx_cast10(x, args...) __bpf_ctx_cast9(args), (void *)ctx[9]
#define __bpf_ctx_cast11(x, args...) __bpf_ctx_cast10(args), (void *)ctx[10]
#define __bpf_ctx_cast12(x, args...) __bpf_ctx_cast11(args), (void *)ctx[11]
#define __bpf_ctx_cast(args...) __bpf_apply(__bpf_ctx_cast, __bpf_narg(args))(args)

/*
 * BPF_PROG is a convenience wrapper for generic
 * tp_btf/fentry/fexit and
 * similar kinds of BPF programs, that accept input arguments as a single
 * pointer to untyped u64 array, where each u64 can actually be a typed
 * pointer or integer of different size. Instead of requiring user to write
 * manual casts and work with array elements by index, BPF_PROG macro
 * allows user to declare a list of named and typed input arguments in the
 * same syntax as for normal C function. All the casting is hidden and
 * performed transparently, while user code can just assume working with
 * function arguments of specified type and name.
 *
 * Original raw context argument is preserved as well as 'ctx' argument.
 * This is useful when using BPF helpers that expect original context
 * as one of the parameters (e.g., for bpf_perf_event_output()).
 */
#define BPF_PROG(name, args...) \
    name(unsigned long long *ctx); \
    static __always_inline typeof(name(0)) \
    ____##name(unsigned long long *ctx, ##args); \
    typeof(name(0)) \
    name(unsigned long long *ctx) \
    { \
        _Pragma("GCC diagnostic push") \
        _Pragma("GCC diagnostic ignored \"-Wint-conversion\"") \
        return ____##name(__bpf_ctx_cast(args)); \
        _Pragma("GCC diagnostic pop") \
    } \
    static __always_inline typeof(name(0)) \
    ____##name(unsigned long long *ctx, ##args)

#ifndef __bpf_nth2
#define __bpf_nth2(_, _1, _2, _3, _4, _5, _6, _7, _8, _9, _10, _11, _12, _13, \
    _14, _15, _16, _17, _18, _19, _20, _21, _22, _23, _24, N, ...) N
#endif
#ifndef __bpf_narg2
#define __bpf_narg2(...) \
    __bpf_nth2(_, ##__VA_ARGS_, 12, 12, 11, 11, 10, 10, 9, 9, 8, 8, 7, 7, \
    6, 6, 5, 5, 4, 4, 3, 3, 2, 2, 1, 1, 0)

```

```

#endif

#define __bpf_treg_cnt(t) \
__builtin_choose_expr(sizeof(t) == 1, 1, \
__builtin_choose_expr(sizeof(t) == 2, 1, \
__builtin_choose_expr(sizeof(t) == 4, 1, \
__builtin_choose_expr(sizeof(t) == 8, 1, \
__builtin_choose_expr(sizeof(t) == 16, 2, \
(void)0))))))

#define
__bpf_reg_cnt0() (0)
#define __bpf_reg_cnt1(t, x) (__bpf_reg_cnt0() + __bpf_treg_cnt(t))
#define __bpf_reg_cnt2(t, x, args...) (__bpf_reg_cnt1(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt3(t, x, args...) (__bpf_reg_cnt2(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt4(t, x, args...) (__bpf_reg_cnt3(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt5(t, x, args...) (__bpf_reg_cnt4(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt6(t, x, args...) (__bpf_reg_cnt5(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt7(t, x, args...) (__bpf_reg_cnt6(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt8(t, x, args...) (__bpf_reg_cnt7(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt9(t, x, args...) (__bpf_reg_cnt8(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt10(t, x, args...) (__bpf_reg_cnt9(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt11(t, x, args...) (__bpf_reg_cnt10(args) + __bpf_treg_cnt(t))
#define __bpf_reg_cnt12(t, x, args...) (__bpf_reg_cnt11(args)
+ __bpf_treg_cnt(t))
#define __bpf_reg_cnt(args...) __bpf_apply(__bpf_reg_cnt, __bpf_narg2(args))(args)

#define __bpf_union_arg(t, x, n) \
__builtin_choose_expr(sizeof(t) == 1, ({ union { __u8 z[1]; t x; } __t = { .z = {ctx[n]} }; __t.x; }, \
__builtin_choose_expr(sizeof(t) == 2, ({ union { __u16 z[1]; t x; } __t = { .z = {ctx[n]} }; __t.x; }, \
__builtin_choose_expr(sizeof(t) == 4, ({ union { __u32 z[1]; t x; } __t = { .z = {ctx[n]} }; __t.x; }, \
__builtin_choose_expr(sizeof(t) == 8, ({ union { __u64 z[1]; t x; } __t = { .z = {ctx[n]} }; __t.x; }, \
__builtin_choose_expr(sizeof(t) == 16, ({ union { __u64 z[2]; t x; } __t = { .z = {ctx[n], ctx[n + 1]} }; __t.x; }, \
(void)0))))))

#define __bpf_ctx_arg0(n, args...)
#define __bpf_ctx_arg1(n, t, x) , __bpf_union_arg(t, x, n - __bpf_reg_cnt1(t, x))
#define __bpf_ctx_arg2(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt2(t, x, args)) __bpf_ctx_arg1(n,
args)
#define __bpf_ctx_arg3(n,
t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt3(t, x, args)) __bpf_ctx_arg2(n, args)
#define __bpf_ctx_arg4(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt4(t, x, args)) __bpf_ctx_arg3(n,
args)
#define __bpf_ctx_arg5(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt5(t, x, args)) __bpf_ctx_arg4(n,
args)
#define __bpf_ctx_arg6(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt6(t, x, args)) __bpf_ctx_arg5(n,
args)
#define __bpf_ctx_arg7(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt7(t, x, args)) __bpf_ctx_arg6(n,

```

```

args)
#define __bpf_ctx_arg8(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt8(t, x, args)) __bpf_ctx_arg7(n,
args)
#define __bpf_ctx_arg9(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt9(t, x, args)) __bpf_ctx_arg8(n,
args)
#define __bpf_ctx_arg10(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt10(t, x, args))
__bpf_ctx_arg9(n, args)
#define __bpf_ctx_arg11(n,
t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt11(t, x, args)) __bpf_ctx_arg10(n, args)
#define __bpf_ctx_arg12(n, t, x, args...) , __bpf_union_arg(t, x, n - __bpf_reg_cnt12(t, x, args))
__bpf_ctx_arg11(n, args)
#define __bpf_ctx_arg(args...) __bpf_apply(__bpf_ctx_arg, __bpf_narg2(args))(__bpf_reg_cnt(args), args)

#define __bpf_ctx_decl0()
#define __bpf_ctx_decl1(t, x) , t x
#define __bpf_ctx_decl2(t, x, args...) , t x __bpf_ctx_decl1(args)
#define __bpf_ctx_decl3(t, x, args...) , t x __bpf_ctx_decl2(args)
#define __bpf_ctx_decl4(t, x, args...) , t x __bpf_ctx_decl3(args)
#define __bpf_ctx_decl5(t, x, args...) , t x __bpf_ctx_decl4(args)
#define __bpf_ctx_decl6(t, x, args...) , t x __bpf_ctx_decl5(args)
#define __bpf_ctx_decl7(t, x, args...) , t x __bpf_ctx_decl6(args)
#define __bpf_ctx_decl8(t, x, args...) , t x __bpf_ctx_decl7(args)
#define __bpf_ctx_decl9(t, x, args...) , t x __bpf_ctx_decl8(args)
#define __bpf_ctx_decl10(t,
x, args...) , t x __bpf_ctx_decl9(args)
#define __bpf_ctx_decl11(t, x, args...) , t x __bpf_ctx_decl10(args)
#define __bpf_ctx_decl12(t, x, args...) , t x __bpf_ctx_decl11(args)
#define __bpf_ctx_decl(args...) __bpf_apply(__bpf_ctx_decl, __bpf_narg2(args))(args)

/*
* BPF_PROG2 is an enhanced version of BPF_PROG in order to handle struct
* arguments. Since each struct argument might take one or two u64 values
* in the trampoline stack, argument type size is needed to place proper number
* of u64 values for each argument. Therefore, BPF_PROG2 has different
* syntax from BPF_PROG. For example, for the following BPF_PROG syntax:
*
* int BPF_PROG(test2, int a, int b) { ... }
*
* the corresponding BPF_PROG2 syntax is:
*
* int BPF_PROG2(test2, int, a, int, b) { ... }
*
* where type and the corresponding argument name are separated by comma.
*
* Use BPF_PROG2 macro if one of the arguments might be a struct/union larger
* than 8 bytes:
*
* int BPF_PROG2(test_struct_arg,

```



```

struct bpf_testmod_struct_arg_1, a, int, b,
*   int, c, int, d, struct bpf_testmod_struct_arg_2, e, int, ret)
*   {
*       // access a, b, c, d, e, and ret directly
*       ...
*   }
*/

#define BPF_PROG2(name, args...) \
name(unsigned long long *ctx); \
static __always_inline typeof(name(0)) \
__##name(unsigned long long *ctx __bpf_ctx_decl(args)); \
typeof(name(0)) name(unsigned long long *ctx) \
{ \
return __##name(ctx __bpf_ctx_arg(args)); \
} \
static __always_inline typeof(name(0)) \
__##name(unsigned long long *ctx __bpf_ctx_decl(args))

struct pt_regs;

#define __bpf_kprobe_args0()      ctx
#define __bpf_kprobe_args1(x)    __bpf_kprobe_args0(), (void *)PT_REGS_PARM1(ctx)
#define __bpf_kprobe_args2(x, args...) __bpf_kprobe_args1(args), (void *)PT_REGS_PARM2(ctx)
#define __bpf_kprobe_args3(x, args...) __bpf_kprobe_args2(args), (void *)PT_REGS_PARM3(ctx)
#define __bpf_kprobe_args4(x, \
args...) __bpf_kprobe_args3(args), (void *)PT_REGS_PARM4(ctx)
#define __bpf_kprobe_args5(x, args...) __bpf_kprobe_args4(args), (void *)PT_REGS_PARM5(ctx)
#define __bpf_kprobe_args6(x, args...) __bpf_kprobe_args5(args), (void *)PT_REGS_PARM6(ctx)
#define __bpf_kprobe_args7(x, args...) __bpf_kprobe_args6(args), (void *)PT_REGS_PARM7(ctx)
#define __bpf_kprobe_args8(x, args...) __bpf_kprobe_args7(args), (void *)PT_REGS_PARM8(ctx)
#define __bpf_kprobe_args(args...) __bpf_apply(__bpf_kprobe_args, __bpf_narg(args))(args)

/*
 * BPF_KPROBE serves the same purpose for kprobes as BPF_PROG for
 * tp_btf/fentry/fexit BPF programs. It hides the underlying platform-specific
 * low-level way of getting kprobe input arguments from struct pt_regs, and
 * provides a familiar typed and named function arguments syntax and
 * semantics of accessing kprobe input paremeters.
 *
 * Original struct pt_regs* context is preserved as 'ctx' argument. This might
 * be necessary when using BPF helpers
 * like bpf_perf_event_output().
 */

#define BPF_KPROBE(name, args...) \
name(struct pt_regs *ctx); \
static __always_inline typeof(name(0)) \
__##name(struct pt_regs *ctx, ##args); \
typeof(name(0)) name(struct pt_regs *ctx) \

```

```

{
    \
    _Pragma("GCC diagnostic push")
    _Pragma("GCC diagnostic ignored \\"-Wint-conversion\\")
    return __##name(__bpf_kprobe_args(args));
    _Pragma("GCC diagnostic pop")
}

static __always_inline typeof(name(0))
__##name(struct pt_regs *ctx, ##args)

#define __bpf_kretprobe_args0() ctx
#define __bpf_kretprobe_args1(x) __bpf_kretprobe_args0(), (void *)PT_REGS_RC(ctx)
#define __bpf_kretprobe_args(args...) __bpf_apply(__bpf_kretprobe_args, __bpf_narg(args))(args)

/*
 * BPF_KRETPROBE is similar to BPF_KPROBE, except, it only provides optional
 * return value (in addition to `struct pt_regs *ctx`), but no input
 * arguments, because they
 * will be clobbered by the time probed function
 * returns.
 */
#define BPF_KRETPROBE(name, args...) \
    name(struct pt_regs *ctx); \
    static __always_inline typeof(name(0)) \
    __##name(struct pt_regs *ctx, ##args); \
    typeof(name(0)) name(struct pt_regs *ctx) \
    { \
        \
        _Pragma("GCC diagnostic push") \
        _Pragma("GCC diagnostic ignored \\"-Wint-conversion\\") \
        return __##name(__bpf_kretprobe_args(args)); \
        _Pragma("GCC diagnostic pop") \
    } \
    static __always_inline typeof(name(0)) __##name(struct pt_regs *ctx, ##args)

/* If kernel has CONFIG_ARCH_HAS_SYSCALL_WRAPPER, read pt_regs directly */
#define __bpf_syscall_args0() ctx
#define __bpf_syscall_args1(x) __bpf_syscall_args0(), (void *)PT_REGS_PARM1_SYSCALL(regs)
#define __bpf_syscall_args2(x, args...) __bpf_syscall_args1(args), (void *)PT_REGS_PARM2_SYSCALL(regs)
#define __bpf_syscall_args3(x, args...) __bpf_syscall_args2(args), \
    (void *)PT_REGS_PARM3_SYSCALL(regs)
#define __bpf_syscall_args4(x, args...) __bpf_syscall_args3(args), (void *)PT_REGS_PARM4_SYSCALL(regs)
#define __bpf_syscall_args5(x, args...) __bpf_syscall_args4(args), (void *)PT_REGS_PARM5_SYSCALL(regs)
#define __bpf_syscall_args6(x, args...) __bpf_syscall_args5(args), (void *)PT_REGS_PARM6_SYSCALL(regs)
#define __bpf_syscall_args7(x, args...) __bpf_syscall_args6(args), (void *)PT_REGS_PARM7_SYSCALL(regs)
#define __bpf_syscall_args(args...) __bpf_apply(__bpf_syscall_args, __bpf_narg(args))(args)

/* If kernel doesn't have CONFIG_ARCH_HAS_SYSCALL_WRAPPER, we have to BPF_CORE_READ from
pt_regs */
#define __bpf_syswrap_args0() ctx

```

```

#define __bpf_syswrap_args1(x)      __bpf_syswrap_args0(), (void
*)PT_REGS_PARM1_CORE_SYSCALL(regs)
#define __bpf_syswrap_args2(x, args...) __bpf_syswrap_args1(args), (void
*)PT_REGS_PARM2_CORE_SYSCALL(regs)
#define __bpf_syswrap_args3(x, args...) __bpf_syswrap_args2(args), (void
*)PT_REGS_PARM3_CORE_SYSCALL(regs)
#define __bpf_syswrap_args4(x, args...) __bpf_syswrap_args3(args), (void
*)PT_REGS_PARM4_CORE_SYSCALL(regs)
#define __bpf_syswrap_args5(x, args...) __bpf_syswrap_args4(args), (void
*)PT_REGS_PARM5_CORE_SYSCALL(regs)
#define __bpf_syswrap_args6(x, args...) __bpf_syswrap_args5(args), (void
*)PT_REGS_PARM6_CORE_SYSCALL(regs)
#define __bpf_syswrap_args7(x, args...) __bpf_syswrap_args6(args), (void
*)PT_REGS_PARM7_CORE_SYSCALL(regs)
#define __bpf_syswrap_args(args...)  __bpf_apply(__bpf_syswrap_args, __bpf_narg(args))(args)

/*
 * BPF_KSYSCALL is a variant of BPF_KPROBE, which is intended for
 * tracing syscall functions, like __x64_sys_close. It hides the underlying
 * platform-specific low-level way of getting syscall input arguments from
 * struct pt_regs, and provides a familiar typed and named function arguments
 * syntax and semantics of accessing syscall input parameters.
 *
 * Original struct pt_regs * context is preserved as 'ctx' argument.
 * This might
 * be necessary when using BPF helpers like bpf_perf_event_output().
 *
 * At the moment BPF_KSYSCALL does not transparently handle all the calling
 * convention quirks for the following syscalls:
 *
 * - mmap(): __ARCH_WANT_SYS_OLD_MMMap.
 * - clone(): CONFIG_CLONE_BACKWARDS, CONFIG_CLONE_BACKWARDS2 and
 *   CONFIG_CLONE_BACKWARDS3.
 * - socket-related syscalls: __ARCH_WANT_SYS_SOCKETCALL.
 * - compat syscalls.
 *
 * This may or may not change in the future. User needs to take extra measures
 * to handle such quirks explicitly, if necessary.
 *
 * This macro relies on BPF CO-RE support and virtual __kconfig externs.
 */
#define BPF_KSYSCALL(name, args...) \
    name(struct pt_regs *ctx); \
    extern _Bool LINUX_HAS_SYSCALL_WRAPPER __kconfig; \
    static __always_inline typeof(name(0)) \
    __##name(struct pt_regs *ctx, ##args); \
    typeof(name(0)) name(struct pt_regs *ctx) \
    { \

```

```

struct pt_regs *regs = LINUX_HAS_SYSCALL_WRAPPER
\
    ? (struct pt_regs *)PT_REGS_PARM1(ctx) \
    : ctx; \
_Pragma("GCC diagnostic push") \
_Pragma("GCC diagnostic ignored \"-Wint-conversion\"") \
if (LINUX_HAS_SYSCALL_WRAPPER) \
    return ____##name(__bpf_syswrap_args(args)); \
else \
    return ____##name(__bpf_syscall_args(args)); \
_Pragma("GCC diagnostic pop") \
} \
static __always_inline typeof(name(0)) \
____##name(struct pt_regs *ctx, ##args)

#define BPF_KPROBE_SYSCALL BPF_KSYSCALL

/* BPF_UPROBE and BPF_URETPROBE are identical to BPF_KPROBE and BPF_KRETPROBE,
 * but are named way less confusingly for SEC("uprobe") and SEC("uretprobe")
 * use cases.
 */
#define BPF_UPROBE(name, args...) BPF_KPROBE(name, ##args)
#define BPF_URETPROBE(name, args...) BPF_KRETPROBE(name, ##args)

```

#endif

Valid-License-Identifier: BSD-2-Clause

SPDX-URL: <https://spdx.org/licenses/BSD-2-Clause.html>

Usage-Guide:

To use the BSD 2-clause "Simplified" License put the following SPDX tag/value pair into a comment according to the placement guidelines in the licensing rules documentation:

SPDX-License-Identifier: BSD-2-Clause

License-Text:

Copyright (c) 2015 The Libbpf Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,

INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2013 Blake Mizerany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding
those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Copyright (c) 2013 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2018 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2016 Mail.Ru Group

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2012 Pter Surnyi. Portions Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files other than internal/golang/* licensed under:

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices

contained

within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional
or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

internal/golang/* files licensed under:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
- * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
MIT License

Copyright (c) 2019 Josh Bleecher Snyder

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN

CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015, Gengo, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Gengo, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO

EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above

copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

When donating the randfill project to the CNCF, we could not reach all the gofuzz contributors to sign the CNCF CLA. As such, according to the CNCF rules to donate a repository, we must add a NOTICE referencing section 7 of the CLA with a list of developers who could not be reached.

`7. Should You wish to submit work that is not Your original creation, You may submit it to the Foundation separately from any Contribution, identifying the complete details of its source and of any license or other restriction (including, but not limited to, related patents, trademarks, and license agreements) of which you are personally aware, and conspicuously marking the work as "Submitted on behalf of a third-party: [named here]".`

Submitted on behalf of a third-party: @dnephin (Daniel Nephin)
Submitted on behalf of a third-party: @AlekSi (Alexey Palazhchenko)
Submitted on behalf of a third-party: @bbigras (Bruno Bigras)
Submitted on behalf of a third-party: @samirkut (Samir)
Submitted on behalf of a
third-party: @posener (Eyal Posener)
Submitted on behalf of a third-party: @Ashikpaul (Ashik Paul)
Submitted on behalf of a third-party: @kwongtailau (Kwongtai)
Submitted on behalf of a third-party: @ericcornelissen (Eric Cornelissen)
Submitted on behalf of a third-party: @eclipseo (Robert-Andr Mauchin)
Submitted on behalf of a third-party: @yanzhoupian (Andrew Pan)
Submitted on behalf of a third-party: @STRRL (Zhiqiang ZHOU)
Submitted on behalf of a third-party: @disconnect3d (Disconnect3d)
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are

met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2020 Bruno Scheufler

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

AWS SDK for Go

Copyright 2015 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Copyright 2014-2015 Stripe, Inc.

Copyright (c) 2012 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
- * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2017 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2015 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2010 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2014 The gofuzz Authors
Copyright 2025 The Kubernetes Authors

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of
the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Prometheus instrumentation library for Go applications
Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

The following components are included in this product:

perks - a fork of <https://github.com/bmizerany/perks>
<https://github.com/beorn7/perks>
Copyright 2013-2015 Blake Mizerany, Bjrn Rabenstein
See <https://github.com/beorn7/perks/blob/master/README.md> for license details.

Go support for Protocol Buffers - Google's data interchange format
<http://github.com/golang/protobuf/>
Copyright 2010 The Go Authors
See source code for license details.
Copyright (c) 2012,2013 Ernest Micklei

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/* SPDX-License-Identifier: (LGPL-2.1 OR BSD-2-Clause) */

#ifndef __BPF_HELPERS__

```

#define __BPF_HELPERS__

/*
 * Note that bpf programs need to include either
 * vmlinux.h (auto-generated from BTF) or linux/types.h
 * in advance since bpf_helper_defs.h uses such types
 * as __u64.
 */
#include "bpf_helper_defs.h"

#define __uint(name, val) int (*name)[val]
#define __type(name, val) typeof(val) *name
#define __array(name, val) typeof(val) *name[]
#define __ulong(name, val) enum { __bpf_concat(__unique_value, __COUNTER__) = val } name

/*
 * Helper macro to place programs, maps, license in
 * different sections in elf_bpf file. Section names
 * are interpreted by libbpf depending on the context (BPF programs, BPF maps,
 * extern variables, etc).
 * To allow use of SEC() with externs (e.g., for extern .maps declarations),
 * make sure __attribute__((unused)) doesn't trigger compilation warning.
 */
#if __GNUC__ && !__clang__

/*
 * Pragma macros are broken on GCC
 * https://gcc.gnu.org/bugzilla/show_bug.cgi?id=55578
 * https://gcc.gnu.org/bugzilla/show_bug.cgi?id=90400
 */
#define SEC(name) __attribute__((section(name), used))

#else

#define SEC(name) \
    _Pragma("GCC diagnostic push") \
    _Pragma("GCC diagnostic ignored \"-Wignored-attributes\"") \
    __attribute__((section(name), used)) \
    _Pragma("GCC diagnostic pop") \

#endif

/* Avoid 'linux/stddef.h' definition of '__always_inline'. */
#undef __always_inline
#define __always_inline inline __attribute__((always_inline))

#ifndef __noinline
#define __noinline __attribute__((noinline))

```

```

#endif
#ifndef __weak
#define __weak __attribute__((weak))
#endif

/*
 * Use __hidden attribute to mark a non-static BPF subprogram effectively
 * static for BPF verifier's verification algorithm purposes, allowing more
 * extensive and permissive BPF verification process, taking into account
 * subprogram's caller context.
 */
#define __hidden __attribute__((visibility("hidden")))

/* When utilizing vmlinux.h
 * with BPF CO-RE, user BPF programs can't include
 * any system-level headers (such as stddef.h, linux/version.h, etc), and
 * commonly-used macros like NULL and KERNEL_VERSION aren't available through
 * vmlinux.h. This just adds unnecessary hurdles and forces users to re-define
 * them on their own. So as a convenience, provide such definitions here.
 */
#ifndef NULL
#define NULL ((void *)0)
#endif

#ifndef KERNEL_VERSION
#define KERNEL_VERSION(a, b, c) (((a) << 16) + ((b) << 8) + ((c) > 255 ? 255 : (c)))
#endif

/*
 * Helper macros to manipulate data structures
 */

/* offsetof() definition that uses __builtin_offset() might not preserve field
 * offset CO-RE relocation properly, so force-redefine offsetof() using
 * old-school approach which works with CO-RE correctly
 */
#undef offsetof
#define offsetof(type, member) ((unsigned long)&((type *)0)->member)

/* redefined container_of() to ensure we use the above offsetof() macro */
#undef container_of
#define container_of(ptr, type, member) \
({ \
    void \
    *__mptr = (void *) (ptr); \
    ((type *) (__mptr - offsetof(type, member))); \
})

```

```

/*
 * Compiler (optimization) barrier.
 */
#ifndef barrier
#define barrier() asm volatile("" ::: "memory")
#endif

/* Variable-specific compiler (optimization) barrier. It's a no-op which makes
 * compiler believe that there is some black box modification of a given
 * variable and thus prevents compiler from making extra assumption about its
 * value and potential simplifications and optimizations on this variable.
 *
 * E.g., compiler might often delay or even omit 32-bit to 64-bit casting of
 * a variable, making some code patterns unverifiable. Putting barrier_var()
 * in place will ensure that cast is performed before the barrier_var()
 * invocation, because compiler has to pessimistically assume that embedded
 * asm section might perform some extra operations on that variable.
 *
 * This is a variable-specific variant of more global barrier().
 */
#ifndef barrier_var
#define
    barrier_var(var) asm volatile("" : "+r"(var))
#endif

/*
 * Helper macro to throw a compilation error if __bpf_unreachable() gets
 * built into the resulting code. This works given BPF back end does not
 * implement __builtin_trap(). This is useful to assert that certain paths
 * of the program code are never used and hence eliminated by the compiler.
 *
 * For example, consider a switch statement that covers known cases used by
 * the program. __bpf_unreachable() can then reside in the default case. If
 * the program gets extended such that a case is not covered in the switch
 * statement, then it will throw a build error due to the default case not
 * being compiled out.
 */
#ifndef __bpf_unreachable
#define __bpf_unreachable() __builtin_trap()
#endif

/*
 * Helper function to perform a tail call with a constant/immediate map slot.
 */
#if __clang_major__ >= 8 && defined(__bpf__)
static __always_inline void
bpf_tail_call_static(void *ctx, const void *map, const __u32 slot)
{

```

```

if
(!__builtin_constant_p(slot))
__bpf_unreachable();

/*
 * Provide a hard guarantee that LLVM won't optimize setting r2 (map
 * pointer) and r3 (constant map index) from _different paths_ ending
 * up at the _same_ call insn as otherwise we won't be able to use the
 * jmpq/nopl retpoline-free patching by the x86-64 JIT in the kernel
 * given they mismatch. See also d2e4c1e6c294 ("bpf: Constant map key
 * tracking for prog array pokes") for details on verifier tracking.
 *
 * Note on clobber list: we need to stay in-line with BPF calling
 * convention, so even if we don't end up using r0, r4, r5, we need
 * to mark them as clobber so that LLVM doesn't end up using them
 * before / after the call.
 */
asm volatile("r1 = %[ctx]\n\t"
             "r2 = %[map]\n\t"
             "r3 = %[slot]\n\t"
             "call 12"
             :: [ctx]"r"(ctx), [map]"r"(map), [slot]"i"(slot)
             : "r0", "r1", "r2", "r3", "r4", "r5");
}
#endif

enum libbpf_pin_type {
LIBBPF_PIN_NONE,
/* PIN_BY_NAME: pin
maps by name (in /sys/fs/bpf by default) */
LIBBPF_PIN_BY_NAME,
};

enum libbpf_tristate {
TRI_NO = 0,
TRI_YES = 1,
TRI_MODULE = 2,
};

#define __kconfig __attribute__((section(".kconfig")))
#define __ksym __attribute__((section(".ksyms")))
#define __kptr_untrusted __attribute__((btf_type_tag("kptr_untrusted")))
#define __kptr __attribute__((btf_type_tag("kptr")))
#define __percpu_kptr __attribute__((btf_type_tag("percpu_kptr")))

#define bpf_ksym_exists(sym) ({ \
    _Static_assert(!__builtin_constant_p(!sym), #sym " should be marked as __weak"); \
    !!sym; \
}

```



```

}))

#define __arg_ctx __attribute__((btf_decl_tag("arg:ctx")))
#define __arg_nonnull __attribute__((btf_decl_tag("arg:nonnull")))
#define __arg_nullable __attribute__((btf_decl_tag("arg:nullable")))
#define __arg_trusted __attribute__((btf_decl_tag("arg:trusted")))
#define __arg_arena __attribute__((btf_decl_tag("arg:arena")))

#ifndef __bpf_concat
#define __bpf_concat(a, b) a ## b
#endif
#ifndef __bpf_apply
#define
__bpf_apply(fn, n) __bpf_concat(fn, n)
#endif
#ifndef __bpf_nth
#define __bpf_nth(, _1, _2, _3, _4, _5, _6, _7, _8, _9, _a, _b, _c, N, ...) N
#endif
#ifndef __bpf_narg
#define __bpf_narg(...) \
__bpf_nth(, ##__VA_ARGS__, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0)
#endif

#define __bpf_fill0(arr, p, x) do {} while (0)
#define __bpf_fill1(arr, p, x) arr[p] = x
#define __bpf_fill2(arr, p, x, args...) arr[p] = x; __bpf_fill1(arr, p + 1, args)
#define __bpf_fill3(arr, p, x, args...) arr[p] = x; __bpf_fill2(arr, p + 1, args)
#define __bpf_fill4(arr, p, x, args...) arr[p] = x; __bpf_fill3(arr, p + 1, args)
#define __bpf_fill5(arr, p, x, args...) arr[p] = x; __bpf_fill4(arr, p + 1, args)
#define __bpf_fill6(arr, p, x, args...) arr[p] = x; __bpf_fill5(arr, p + 1, args)
#define __bpf_fill7(arr, p, x, args...) arr[p] = x; __bpf_fill6(arr, p + 1, args)
#define __bpf_fill8(arr, p, x, args...) arr[p] = x; __bpf_fill7(arr, p + 1, args)
#define __bpf_fill9(arr, p, x, args...) \
arr[p] = x; __bpf_fill8(arr, p + 1, args)
#define __bpf_fill10(arr, p, x, args...) arr[p] = x; __bpf_fill9(arr, p + 1, args)
#define __bpf_fill11(arr, p, x, args...) arr[p] = x; __bpf_fill10(arr, p + 1, args)
#define __bpf_fill12(arr, p, x, args...) arr[p] = x; __bpf_fill11(arr, p + 1, args)
#define __bpf_fill(arr, args...) \
__bpf_apply(__bpf_fill, __bpf_narg(args))(arr, 0, args)

/*
 * BPF_SEQ_PRINTF to wrap bpf_seq_printf to-be-printed values
 * in a structure.
 */
#define BPF_SEQ_PRINTF(seq, fmt, args...) \
({ \
static const char __fmt[] = fmt; \
unsigned long long __param[__bpf_narg(args)]; \

```

```

    \
    _Pragma("GCC diagnostic push") \
    _Pragma("GCC diagnostic ignored \\"-Wint-conversion\\") \
    __bpf_fill(__param, args); \
    _Pragma("GCC diagnostic pop") \
    \
    bpf_seq_printf(seq, __fmt, sizeof(__fmt), \
        __param, sizeof(__param)); \
})

/*
 * BPF_SNPRINTF wraps the bpf_snprintf helper with variadic arguments instead of
 * an array of u64.
 */
#define BPF_SNPRINTF(out, out_size, fmt, args...) \
({ \
    static const char __fmt[] = fmt; \
    unsigned long long __param[__bpf_narg(args)]; \
    \
    _Pragma("GCC diagnostic push") \
    _Pragma("GCC diagnostic ignored \\"-Wint-conversion\\") \
    __bpf_fill(__param, args); \
    _Pragma("GCC diagnostic pop") \
    \
    bpf_snprintf(out, out_size, __fmt, \
        __param, sizeof(__param)); \
})

#ifdef BPF_NO_GLOBAL_DATA
#define BPF_PRINTK_FMT_MOD
#else
#define BPF_PRINTK_FMT_MOD static const
#endif

#define __bpf_printk(fmt, ...) \
({ \
    BPF_PRINTK_FMT_MOD char ____fmt[] = fmt; \
    bpf_trace_printk(____fmt, sizeof(____fmt), \
        ##__VA_ARGS__); \
})

/*
 * __bpf_vprintf wraps the bpf_trace_vprintf helper with variadic arguments
 * instead of an array of u64.
 */
#define __bpf_vprintf(fmt, args...) \
({ \
    static const char __fmt[] = fmt; \

```

```

unsigned long long __param[__bpf_narg(args)]; \
\
#pragma("GCC
diagnostic push") \
#pragma("GCC diagnostic ignored \"-Wint-conversion\"") \
__bpf_fill(__param, args); \
#pragma("GCC diagnostic pop") \
\
bpf_trace_vprintk(__fmt, sizeof(__fmt), \
    __param, sizeof(__param)); \
})

/* Use __bpf_printk when bpf_printk call has 3 or fewer fmt args
 * Otherwise use __bpf_vprintk
 */
#define __bpf_pick_printk(...) \
    __bpf_nth(__VA_ARGS__, __bpf_vprintk, __bpf_vprintk, __bpf_vprintk, \
        __bpf_vprintk, __bpf_vprintk, __bpf_vprintk, __bpf_vprintk, \
        __bpf_vprintk, __bpf_vprintk, __bpf_printk /*3*/, __bpf_printk /*2*/, \
        __bpf_printk /*1*/, __bpf_printk /*0*/)

/* Helper macro to print out debug messages */
#define bpf_printk(fmt, args...) __bpf_pick_printk(args)(fmt, ##args)

struct bpf_iter_num;

extern int bpf_iter_num_new(struct bpf_iter_num *it, int start, int end) __weak __ksym;
extern int *bpf_iter_num_next(struct bpf_iter_num *it) __weak __ksym;
extern void bpf_iter_num_destroy(struct bpf_iter_num
    *it) __weak __ksym;

#ifndef bpf_for_each
/* bpf_for_each(iter_type, cur_elem, args...) provides generic construct for
 * using BPF open-coded iterators without having to write mundane explicit
 * low-level loop logic. Instead, it provides for()-like generic construct
 * that can be used pretty naturally. E.g., for some hypothetical cgroup
 * iterator, you'd write:
 *
 * struct cgroup *cg, *parent_cg = <...>;
 *
 * bpf_for_each(cgroup, cg, parent_cg, CG_ITER_CHILDREN) {
 *     bpf_printk("Child cgroup id = %d", cg->cgroup_id);
 *     if (cg->cgroup_id == 123)
 *         break;
 * }
 *
 * I.e., it looks almost like high-level for each loop in other languages,
 * supports continue/break, and is verifiable by BPF verifier.

```

```

*
* For iterating integers, the difference between bpf_for_each(num, i, N, M)
* and bpf_for(i, N, M) is in that bpf_for() provides additional proof to
* verifier that i is in [N, M) range, and in bpf_for_each() case i is `int
*`, not just `int`. So for integers
bpf_for() is more convenient.
*
* Note: this macro relies on C99 feature of allowing to declare variables
* inside for() loop, bound to for() loop lifetime. It also utilizes GCC
* extension: __attribute__((cleanup(<func>))), supported by both GCC and
* Clang.
*/
#define bpf_for_each(type, cur, args...) for (    \
/* initialize and define destructor */    \
struct bpf_iter_##type __it __attribute__((aligned(8), /* enforce, just in case */ \
      cleanup(bpf_iter_##type##_destroy))), \
/* __p pointer is just to call bpf_iter_##type##_new() *once* to init __it */ \
    *__p __attribute__((unused)) = (    \
      bpf_iter_##type##_new(&__it, ##args), \
/* this is a workaround for Clang bug: it currently doesn't emit BTF */ \
/* for bpf_iter_##type##_destroy() when used from cleanup() attribute */ \
      (void)bpf_iter_##type##_destroy, (void *)0); \
/* iteration and termination check */    \
    (((cur) = bpf_iter_##type##_next(&__it)));    \
)
#endif
/* bpf_for_each */

#ifndef bpf_for
/* bpf_for(i, start, end) implements a for()-like looping construct that sets
* provided integer variable *i* to values starting from *start* through,
* but not including, *end*. It also proves to BPF verifier that *i* belongs
* to range [start, end), so this can be used for accessing arrays without
* extra checks.
*
* Note: *start* and *end* are assumed to be expressions with no side effects
* and whose values do not change throughout bpf_for() loop execution. They do
* not have to be statically known or constant, though.
*
* Note: similarly to bpf_for_each(), it relies on C99 feature of declaring for()
* loop bound variables and cleanup attribute, supported by GCC and Clang.
*/
#define bpf_for(i, start, end) for (    \
/* initialize and define destructor */    \
struct bpf_iter_num __it __attribute__((aligned(8), /* enforce, just in case */ \
      cleanup(bpf_iter_num_destroy))), \
/* __p pointer is necessary to

```

```

call bpf_iter_num_new() *once* to init ____it */ \
    *____p __attribute__((unused)) = (    \
    bpf_iter_num_new(&____it, (start), (end)), \
/* this is a workaround for Clang bug: it currently doesn't emit BTF */ \
/* for bpf_iter_num_destroy() when used from cleanup() attribute */ \
    (void)bpf_iter_num_destroy, (void *)0); \
({    \
/* iteration step */    \
    int *____t = bpf_iter_num_next(&____it);    \
/* termination and bounds check */    \
    (____t && ((i) = *____t, (i) >= (start) && (i) < (end)))); \
});    \
)
#endif /* bpf_for */

#ifndef bpf_repeat
/* bpf_repeat(N) performs N iterations without exposing iteration number
 *
 * Note: similarly to bpf_for_each(), it relies on C99 feature of declaring for()
 * loop bound variables and cleanup attribute, supported by GCC and Clang.
 */
#define bpf_repeat(N) for (    \
/* initialize and define destructor */    \
    struct bpf_iter_num ____it __attribute__((aligned(8), /* enforce,
just in case */ \
        cleanup(bpf_iter_num_destroy))), \
/* ____p pointer is necessary to call bpf_iter_num_new() *once* to init ____it */ \
    *____p __attribute__((unused)) = (    \
    bpf_iter_num_new(&____it, 0, (N)), \
/* this is a workaround for Clang bug: it currently doesn't emit BTF */ \
/* for bpf_iter_num_destroy() when used from cleanup() attribute */ \
    (void)bpf_iter_num_destroy, (void *)0); \
    bpf_iter_num_next(&____it);    \
/* nothing here */    \
)
#endif /* bpf_repeat */

#endif

Copyright (c) 2013, The GoGo Authors. All rights reserved.

```

Protocol Buffers for Go with Gadgets

Go support for Protocol Buffers - Google's data interchange format

Copyright 2010 The Go Authors. All rights reserved.

<https://github.com/golang/protobuf>

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2014 Sam Ghods

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2012 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The forked go-yaml.v3 library under this project is covered by two different licenses (MIT and Apache):

MIT License

The following files were ported to Go from C files of libyaml, and thus are still covered by their original MIT license, with the additional copyright starting in 2011 when the project was ported over:

apic.go emitterc.go parserc.go readerc.go scannerc.go
writerc.go yamlh.go yamlprivateh.go

Copyright (c)

2006-2010 Kirill Simonov

Copyright (c) 2006-2011 Kirill Simonov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in

the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License

All the remaining project files are covered by the Apache license:

Copyright (c) 2011-2019 Canonical Ltd

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The forked go-yaml.v2 library under the project is covered by an Apache license:

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS"

BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2009,2014 Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LGPL-2.1 OR BSD-2-Clause

1.37 bash 5.2.37-r0

1.37.1 Available under license :

Shellfloat is copyright (c) 2020 by Michael Wood.

=====

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code.

And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed

this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the

public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for

the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met.

This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid

circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may

convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the

Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no

charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply

if neither you nor
any third party retains the ability to install
modified object code on the User Product (for example, the work has
been installed in ROM).

The requirement to provide Installation Information does not include a
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed. Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement
the terms of this

License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law. If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option
remove any additional permissions from that copy, or from any part of
it. (Additional permissions may be written to require their own
removal in certain cases when you modify the work.) You may place
additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you
add to a covered work, you may (if
authorized by the copyright holders of
that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the
terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or
author attributions in that material or in the Appropriate Legal

Notices displayed by works containing it; or

- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your

license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation

by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept

this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor

has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent

license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number.

If the

Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY

APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF

ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>  
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <https://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <https://www.gnu.org/licenses/why-not-lgpl.html>.

GNU Free Documentation License
Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
<http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it,

with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We

have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of

the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The

Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated

as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains

nothing but

endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections

Entitled "History"

in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4.

Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include

the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of

the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed

to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it,

either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies
of the Program's

source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole

or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to

control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent

license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN

IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

@c The GNU Free Documentation License.

@center Version 1.3, 3 November 2008

@c This file is intended to be included within another document,
@c hence no sectioning command or @node.

@display

Copyright @copyright{} 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

@uref{http://fsf.org/}

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

@end display

@enumerate 0

@item

PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document @dfn{free} in the sense of freedom: to
assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way
to get credit for their work, while not being considered responsible
for modifications made by others.

This License is a kind of ``copyleft'', which means that derivative
works of the document
must themselves be free in the same sense. It
complements the GNU General Public License, which is a copyleft
license designed for free software.

We have designed this License in order to use it for manuals for free
software, because free software needs free documentation: a free
program should come with manuals providing the same freedoms that the
software does. But this License is not limited to software manuals;
it can be used for any textual work, regardless of subject matter or
whether it is published as a printed book. We recommend this License
principally for works whose purpose is instruction or reference.

@item

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that
contains a notice placed by the copyright holder saying it can be
distributed under the terms of this License. Such a notice grants a
world-wide, royalty-free license, unlimited in duration, to use that
work under the conditions stated herein. The ``Document'', below,
refers

to any such manual or work. Any member of the public is a licensee, and is addressed as ``you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A ``Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ``Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ``Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The ``Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A ``Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not ``Transparent" is called ``Opaque".

Examples of suitable formats for Transparent copies include plain `@sc{ascii}` without markup, Texinfo input format, `LaTeX{} input format`, `@acronym{SGML}` or `@acronym{XML}` using a publicly available `@acronym{DTD}`, and standard-conforming simple `@acronym{HTML}`, PostScript or `@acronym{PDF}` designed for human modification. Examples of transparent image formats include `@acronym{PNG}`, `@acronym{XCF}` and `@acronym{JPG}`. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, `@acronym{SGML}` or `@acronym{XML}` for which the `@acronym{DTD}` and/or processing tools are not generally available, and the machine-generated `@acronym{HTML}`, PostScript or `@acronym{PDF}` produced by some word processors for output purposes only.

The ``Title Page'' means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, ``Title Page'' means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The ``publisher'' means any person or entity that distributes copies of the Document to the public.

A section ``Entitled XYZ'' means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as ``Acknowledgements'', ``Dedications'', ``Endorsements'', or ``History''.) To ``Preserve the Title'' of such a section when you modify the Document means that it remains a section ``Entitled XYZ'' according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

@item

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever

to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

@item

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible.

You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that

this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

@item

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

@enumerate A

@item

Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

@item

List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

@item

State on the Title page the name of the publisher of the Modified Version, as the publisher.

@item

Preserve all the copyright notices of the Document.

@item

Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

@item

Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

@item

Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

@item

Include an unaltered copy of this License.

@item

Preserve the section Entitled ``History'', Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled ``History'' in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

@item

Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the ``History'' section. You may omit a network location for a work that was published at least

four years before the Document itself, or if the original publisher of the version it refers to gives permission.

@item

For any section Entitled ``Acknowledgements'' or ``Dedications'', Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

@item

Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

@item

Delete any section Entitled ``Endorsements''. Such a section may not be included in the Modified Version.

@item

Do not retitle any existing section to be Entitled ``Endorsements'' or to conflict in title with any Invariant Section.

@item

Preserve any Warranty Disclaimers.

@end enumerate

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document,

you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled ``Endorsements'', provided it contains nothing but endorsements of your Modified Version by various parties---for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

@item

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ``History'' in the various original documents, forming one section Entitled

``History"; likewise combine any sections Entitled ``Acknowledgements", and any sections Entitled ``Dedications". You must delete all sections Entitled ``Endorsements."

@item

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract

a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

@item

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an ``aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

@item

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the

original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled ``Acknowledgements'', ``Dedications'', or ``History'', the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

@item

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

@item

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new

versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See [@uref{http://www.gnu.org/copyleft/}](http://www.gnu.org/copyleft/).

Each version of the License is given a distinguishing version number.

If the Document specifies that

a particular numbered version of this

License ``or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

@item

RELICENSING

``Massive Multiauthor Collaboration Site" (or ``MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A ``Massive Multiauthor Collaboration" (or ``MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

``CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

``Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is ``eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the

MMC is eligible for relicensing.

@end enumerate

@page

@heading ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

@smallexample

@group

Copyright (C) @var{year} @var{your name}.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled ``GNU Free Documentation License".

@end group

@end smallexample

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the ``with@dots{ } Texts." line with this:

@smallexample

@group

with the Invariant Sections being @var{list their titles}, with the Front-Cover Texts being @var{list}, and with the Back-Cover Texts being @var{list}.

@end group

@end smallexample

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

@c Local Variables:

@c ispell-local-pdict: "ispell-dict"

@c End:

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to
freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights. Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received. You must make sure that they, too, receive
or can get the source code. And you must show them these terms so they
know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software. For both users' and
authors' sake, the GPL requires that modified versions be marked as

changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without

permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally

available free

programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered

by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed;

section 10

makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or

similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated

place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the

User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or

limiting liability differently from the terms of sections 15 and 16 of this License; or

- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on

those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or

modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that

transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights

granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this

License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may

not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS"

WITHOUT WARRANTY

OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by

the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>  
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library. If this is what you want to
do, use the GNU Lesser General
Public License instead of this License. But first, please read
<<http://www.gnu.org/philosophy/why-not-lgpl.html>>.
Unless otherwise stated, all files in this directory are Copyright (C)
1991,1992,1993,1994,1995,1996,1997,1998,1999,2000,2001,2002,2003,
2004,2005,2006,2007,2008,2009,2010,2011
Free Software Foundation, Inc.

See the file COPYING in the bash distribution root directory for copying
and usage restrictions.

The file ifs-posix.tests is Copyright (C) 2005 Glen Fowler.

1.38 go.yaml.in-yaml 2.4.3

1.38.1 Available under license :

Copyright 2011-2016 Canonical Ltd.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.39 pam 1.7.0-r4

1.39.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in

whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete

machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates

the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands

`show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

```
# Automatically generated by apkgbuild-cpan, template 4
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkgbuild-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
buildidir="$srcdir/${_pkgreal}-$pkgver"
```

```
build() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
        INSTALLDIRS=vendor \
        NO_PACKLIST=1 \
        NO_PERLLOCAL=1
    make
}
```

```
check() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    make test
}
```

```

package() {
    make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
    BSD-Resource-1.2911.tar.gz
"

# Contributor: Maxim Karasev <begs@disroot.org>
# Maintainer: mio <miyopan@e.email>
pkgname=bsd-games
pkgver=3.3
pkgrel=2
pkgdesc="traditional text mode games from BSD"
url="https://bsd-games.sourceforge.io/"
arch="all"
license="BSD-3-Clause"
install="$pkgname.post-install"
makedepends="ncurses-dev coreutils" # configure causes busybox expr error
subpackages="$pkgname-doc"
source="https://sourceforge.net/projects/bsd-games/files/bsd-games-$pkgver.tar.gz
0001-bsd-games-scores-dir.patch
"

options="!check" # no tests

build() {
    # some GNU autoconf options are ignored, but it works
    ./configure \
        --build=$CBUILD \
        --host=$CHOST \
        --prefix=/usr \
        --sysconfdir=/etc \
        --mandir=/usr/share/man \
        --sourcedir=/usr/share/$pkgname
    make
}

package() {
    make DESTDIR="$pkgdir" install
    install -Dm644 LICENSE "$pkgdir"/usr/share/licenses/bsd-games/LICENSE
}

sha512sums="
aaf36d09d4fe68514a5c279063d7e77a9a9a84c447037cba974b29faa07eb80c2aab2379d0699c196b5c27ffe1b2bb2c6
8d3390143e0e62d5e3d210ccfb61294
    bsd-games-3.3.tar.gz
e1def0d4835a877242c4173fae862e278a7858738e0bacacbb045a42bcabec2aa7c72d000bbd2421ce35fe66cdc1f52c0

```

577b7e278029ca5ed50f7aab22319f6 0001-bsd-games-scores-dir.patch

"

MIT License

Copyright (c) 2023 Leaf Corcoran

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Maintainer: Natanael Copa <ncopa@alpinelinux.org>

pkgname=libart-lgpl

pkgver=2.3.21

pkgrel=8

pkgdesc="A library for high-performance 2D graphics"

url="https://www.levien.com/libart/"

arch="all"

options="!check" # No test suite.

license="LGPL-2.0-or-later"

subpackages="\$pkgname-dev"

source="https://download.gnome.org/sources/libart-lgpl/2.3/libart-lgpl-\$pkgver.tar.bz2"

"

builddir="\$srcdir"/libart-lgpl-\$pkgver

```
prepare() {  
    update_config_sub  
    default_prepare  
}
```

```
build() {  
    ./configure \  
    --build=$CBUILD \  
    --host=$CHOST \  
    --
```

```

--prefix=/usr
make
}

package() {
    make DESTDIR="$pkgdir" install
}

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"
mini_sendmail - accept email on behalf of real sendmail

```

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```

# Contributor: Patrycja Rosa <alpine@ptrcnull.me>
# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
cdefs.h
queue.h

```

```

tree.h
"

builddir="$srcdir"
options="!check" # just headers

package() {
    mkdir -p "$pkgdir"
    install -Dm644 -t "$pkgdir"/usr/include/sys \
        cdefs.h queue.h tree.h
}

sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178 cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750 queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3 tree.h
"

# Contributor: Carlo Landmeter <clandmeter@alpinelinux.org>
# Maintainer: Kevin Daudt <kdaudt@alpinelinux.org>
pkgname=spdx-licenses
pkgver=3.26.0
pkgrel=0
pkgdesc="Various data formats for the SPDX License List"
url="https://spdx.org/"
arch="noarch"
options="!check" # no test suite
license="CC-BY-3.0"
source="license-list-data-$pkgver.tar.gz::https://github.com/spdx/license-list-data/archive/v$pkgver.tar.gz"
builddir="$srcdir/license-list-data-$pkgver"
subpackages="$pkgname-list"

_types="html json rdfa rdft rdfturtle rdfxml template text"

for type in $_types; do
    subpackages="$subpackages $pkgname-$type:_subpkg"
done

package() {
    mkdir -p "$pkgdir"
}

_subpkg() {
    local type=${subpkgname/$pkgname-/}
    pkgdesc="$pkgdesc ($type)"
    install_if="$pkgname=$pkgver-r$pkgrel"
    mkdir -p "$subpkgdir"/usr/share/spdx
    cp -r "$builddir"/$type "$subpkgdir"/usr/share/spdx/

```



```

}

list() {
  pkgdesc="$pkgdesc (licence list)"
  mkdir -p "$subpkgdir"/usr/share/spdx
  local i; for i in "$builddir"/text/*.txt;
  do
    local license=${i##*/}
    echo ${license%.*} >> "$subpkgdir"/usr/share/spdx/license.lst
  done
}

sha512sums="
67e618d4642dfe4f366935d9514d6f7941d3711e77c49bc1b340ac572d36015981b8f10c72a9f80f1d83e132ae7e5093c
0155e0c0e786e4777ee16d1cb7b885a  license-list-data-3.26.0.tar.gz
"

```

Original Copyright (c) 1985 by Supoj Sutanthavibul
 Parts Copyright (c) 1989-2012 by Brian V. Smith
 Parts Copyright (c) 1991 by Paul King
 Other Copyrights may be found in various files

Main Xfig copyright notice:

Any party obtaining a copy of these files is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish and/or distribute copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.

No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Parts Copyright (C) 1993 by Alan Richardson (mppa3@uk.ac.sussex.syma)
 The text rotation code in w_rottext.c was written by Alan Richardson.
 The above copyright notice holds for this work as well.

Parts Copyright (c) 1994 by Anthony Dekker
 The Kohonen neural network code for color optimization was written by Anthony Dekker.

Parts Copyright (c) 1995 by C. Blanc and C. Schlick
 The X-Spline code was written Carole Blanc (blanc@labri.u-bordeaux.fr) and Christophe Schlick (schlick@labri.u-bordeaux.fr) starting from an initial implementation done by C. Feuille, S. Grosbois, L. Maziere and L. Minihot as a student practice (Universite Bordeaux, France). For additional information about X-splines, see:
 "X-Splines: A Spline Model Designed for the End User" by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95
<http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/sig1.html>
 # Contributor: Dhruvin Gandhi <contact@dhruvin>
 # Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
 pkgname=py3-license-expression

```

_pkgname=license_expression
pkgver=30.4.1
pkgrel=0
pkgdesc="Library to parse, compare, simplify and normalize license expressions"
url="https://github.com/nexB/license-expression"
arch="noarch"
license="Apache-2.0"
depends="python3 py3-boolean.py"
makedepends="py3-setuptools py3-setuptools_scm py3-gpep517 py3-wheel"
checkdepends="py3-pytest py3-pytest-xdist"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/l/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir/${_pkgname}-${pkgver}"

build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}

check() {
    python3 -m venv --clear --without-pip --system-site-packages .testenv
    .testenv/bin/python3 -m installer .dist/*.whl
    .testenv/bin/python3 -m pytest
}

package() {
    python3 -m installer -d "$pkgdir" \
        .dist/*.whl
}

sha512sums="
114c630f098939757e49187e35b4e54883266a2ee9ca283724992420ef4c860f9606a8342da0ab74d93627c10805f312
fda3f699cbc7c4fbd47a71e220440fb0
    license_expression-30.4.1.tar.gz
"

/* $NetBSD: queue.h,v 1.70 2015/11/02 15:21:23 christos Exp $ */

/*
 * Copyright (c) 1991, 1993
 * The Regents of the University of California. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright

```

- * notice, this list of conditions and the following disclaimer in the
- * documentation and/or other materials provided with the distribution.
- * 3. Neither the name of the University nor the names of its contributors
- * may be used to endorse or promote products derived from this software
- * without specific prior written permission.
- *
- * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
- * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
- BUT NOT LIMITED TO, THE
- * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
- * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
- * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
- * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
- * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
- * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
- * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
- * SUCH DAMAGE.
- *
- * @(#)queue.h 8.5 (Berkeley) 8/20/94
- */

```
#ifndef _SYS_QUEUE_H_
#define _SYS_QUEUE_H_
```

```
/*
 * This file defines five types of data structures: singly-linked lists,
 * lists, simple queues, tail queues, and circular queues.
 *
 * A singly-linked list is headed by a single forward pointer. The
 * elements are singly linked for minimum space and
 * pointer manipulation
 * overhead at the expense of O(n) removal for arbitrary elements. New
 * elements can be added to the list after an existing element or at the
 * head of the list. Elements being removed from the head of the list
 * should use the explicit macro for this purpose for optimum
 * efficiency. A singly-linked list may only be traversed in the forward
 * direction. Singly-linked lists are ideal for applications with large
 * datasets and few or no removals or for implementing a LIFO queue.
 *
 * A list is headed by a single forward pointer (or an array of forward
 * pointers for a hash table header). The elements are doubly linked
 * so that an arbitrary element can be removed without a need to
 * traverse the list. New elements can be added to the list before
 * or after an existing element or at the head of the list. A list
 * may only be traversed in the forward direction.
 *
 * A simple queue is headed by a pair of pointers, one the head of the
```

- * list and the other to the tail of the list. The elements are singly linked to save space, so elements can only be removed from the head of the list. New elements can be added to the list after an existing element, at the head of the list, or at the end of the list. A simple queue may only be traversed in the forward direction.
- *
- * A tail queue is headed by a pair of pointers, one to the head of the list and the other to the tail of the list. The elements are doubly linked so that an arbitrary element can be removed without a need to traverse the list. New elements can be added to the list before or after an existing element, at the head of the list, or at the end of the list. A tail queue may be traversed in either direction.
- *
- * A circle queue is headed by a pair of pointers, one to the head of the list and the other to the tail of the list. The elements are doubly linked so that an arbitrary element can be removed without a need to traverse the list. New elements can be added to the list before or after an existing element, at the head of the list, or at the end of the list.
- * A circle queue may be traversed in either direction, but has a more complex end of list detection.
- *
- * For details on the use of these macros, see the queue(3) manual page.
- */

```
/*
 * Include the definition of NULL only on NetBSD because sys/null.h
 * is not available elsewhere. This conditional makes the header
 * portable and it can simply be dropped verbatim into any system.
 * The caveat is that on other systems some other header
 * must provide NULL before the macros can be used.
 */
```

```
#ifdef __NetBSD__
#include <sys/null.h>
#endif
```

```
#if defined(QUEUEDEBUG)
# if defined(_KERNEL)
# define QUEUEDEBUG_ABORT(...) panic(__VA_ARGS__)
# else
# include <err.h>
# define QUEUEDEBUG_ABORT(...) err(1, __VA_ARGS__)
# endif
#endif
```

```
/*
 * Singly-linked List definitions.
```

```

*/
#define SLIST_HEAD(name, type) \
struct name { \
    struct type *slh_first; /* first \
    element */ \
}

#define SLIST_HEAD_INITIALIZER(head) \
{ NULL }

#define SLIST_ENTRY(type) \
struct { \
    struct type *sle_next; /* next element */ \
}

/*
 * Singly-linked List access methods.
 */
#define SLIST_FIRST(head) ((head)->slh_first)
#define SLIST_END(head) NULL
#define SLIST_EMPTY(head) ((head)->slh_first == NULL)
#define SLIST_NEXT(elm, field) ((elm)->field.sle_next)

#define SLIST_FOREACH(var, head, field) \
for((var) = (head)->slh_first; \
    (var) != SLIST_END(head); \
    (var) = (var)->field.sle_next)

#define SLIST_FOREACH_SAFE(var, head, field, tvar) \
for ((var) = SLIST_FIRST((head)); \
    (var) != SLIST_END(head) && \
    ((tvar) = SLIST_NEXT((var), field), 1); \
    (var) = (tvar))

/*
 * Singly-linked List functions.
 */
#define SLIST_INIT(head) do { \
    (head)->slh_first = SLIST_END(head); \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_AFTER(slistelm, elm, field) do { \
    (elm)->field.sle_next = (slistelm)->field.sle_next; \
    (slistelm)->field.sle_next \
    = (elm); \
} while (/*CONSTCOND*/0)

#define SLIST_INSERT_HEAD(head, elm, field) do { \

```

```

(elm)->field.sle_next = (head)->slh_first; \
(head)->slh_first = (elm); \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_AFTER(slistelm, field) do { \
    (slistelm)->field.sle_next = \
        SLIST_NEXT(SLIST_NEXT((slistelm), field), field); \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE_HEAD(head, field) do { \
    (head)->slh_first = (head)->slh_first->field.sle_next; \
} while (/*CONSTCOND*/0)

#define SLIST_REMOVE(head, elm, type, field) do { \
    if ((head)->slh_first == (elm)) { \
        SLIST_REMOVE_HEAD((head), field); \
    } \
    else { \
        struct type *curelm = (head)->slh_first; \
        while(curelm->field.sle_next != (elm)) \
            curelm = curelm->field.sle_next; \
        curelm->field.sle_next = \
            curelm->field.sle_next->field.sle_next; \
    } \
} while (/*CONSTCOND*/0)

/*
 * List definitions.
 */
#define LIST_HEAD(name, type) \
struct name { \
    struct type *lh_first; /* first element */ \
}

#define LIST_HEAD_INITIALIZER(head) \
{ NULL }

#define LIST_ENTRY(type) \
struct { \
    struct type *le_next; /* next element */ \
    struct type **le_prev; /* address of previous next element */ \
}

/*
 * List access methods.
 */
#define LIST_FIRST(head) ((head)->lh_first)

```

```

#define LIST_END(head) NULL
#define LIST_EMPTY(head) ((head)->lh_first == LIST_END(head))
#define LIST_NEXT(elm, field) ((elm)->field.le_next)

#define LIST_FOREACH(var, head, field) \
    for ((var) = ((head)->lh_first); \
         (var) != LIST_END(head); \
         (var) = ((var)->field.le_next))

#define LIST_FOREACH_SAFE(var, head, field, tvar) \
    for ((var) = LIST_FIRST((head)); \
         (var) != LIST_END(head) && \
         ((tvar) = LIST_NEXT((var), field), 1); \
         (var) = (tvar))

#define LIST_MOVE(head1, head2) do { \
    LIST_INIT((head2)); \
    if (!LIST_EMPTY((head1))) \
    { \
        (head2)->lh_first = (head1)->lh_first; \
        LIST_INIT((head1)); \
    } \
} while (/*CONSTCOND*/0)

/*
 * List functions.
 */

#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field) \
    if ((head)->lh_first && \
        (head)->lh_first->field.le_prev != &(head)->lh_first) \
        QUEUEDEBUG_ABORT("LIST_INSERT_HEAD %p %s:%d", (head), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_OP(elm, field) \
    if ((elm)->field.le_next && \
        (elm)->field.le_next->field.le_prev != \
        &(elm)->field.le_next) \
        QUEUEDEBUG_ABORT("LIST_* forw %p %s:%d", (elm), \
            __FILE__, __LINE__); \
    if (*(elm)->field.le_prev != (elm)) \
        QUEUEDEBUG_ABORT("LIST_* back %p %s:%d", (elm), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field) \
    (elm)->field.le_next = (void *)1L; \
    (elm)->field.le_prev = (void *)1L;
#else
#define QUEUEDEBUG_LIST_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_LIST_OP(elm,

```

```

field)
#define QUEUEDEBUG_LIST_POSTREMOVE(elm, field)
#endif

#define LIST_INIT(head) do { \
    (head)->lh_first = LIST_END(head); \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_AFTER(listelm, elm, field) do { \
    QUEUEDEBUG_LIST_OP((listelm), field) \
    if (((elm)->field.le_next = (listelm)->field.le_next) != \
        LIST_END(head)) \
        (listelm)->field.le_next->field.le_prev = \
            &(elm)->field.le_next; \
    (listelm)->field.le_next = (elm); \
    (elm)->field.le_prev = &(listelm)->field.le_next; \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_BEFORE(listelm, elm, field) do { \
    QUEUEDEBUG_LIST_OP((listelm), field) \
    (elm)->field.le_prev = (listelm)->field.le_prev; \
    (elm)->field.le_next = (listelm); \
    *(listelm)->field.le_prev = (elm); \
    (listelm)->field.le_prev = &(elm)->field.le_next; \
} while (/*CONSTCOND*/0)

#define LIST_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBUG_LIST_INSERT_HEAD((head), (elm), field) \
    if (((elm)->field.le_next \
        = (head)->lh_first) != LIST_END(head)) \
        (head)->lh_first->field.le_prev = &(elm)->field.le_next; \
    (head)->lh_first = (elm); \
    (elm)->field.le_prev = &(head)->lh_first; \
} while (/*CONSTCOND*/0)

#define LIST_REMOVE(elm, field) do { \
    QUEUEDEBUG_LIST_OP((elm), field) \
    if ((elm)->field.le_next != NULL) \
        (elm)->field.le_next->field.le_prev = \
            (elm)->field.le_prev; \
    *(elm)->field.le_prev = (elm)->field.le_next; \
    QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

#define LIST_REPLACE(elm, elm2, field) do { \
    if (((elm2)->field.le_next = (elm)->field.le_next) != NULL) \
        (elm2)->field.le_next->field.le_prev = \
            &(elm2)->field.le_next; \

```



```

(elm2)->field.le_prev = (elm)->field.le_prev; \
*(elm2)->field.le_prev = (elm2); \
QUEUEDEBUG_LIST_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

/*
 * Simple queue definitions.
 */
#define SIMPLEQ_HEAD(name, type) \
struct name { \
    struct \
    type *sqh_first; /* first element */ \
    struct type **sqh_last; /* addr of last next element */ \
}

#define SIMPLEQ_HEAD_INITIALIZER(head) \
{ NULL, &(head).sqh_first }

#define SIMPLEQ_ENTRY(type) \
struct { \
    struct type *sqe_next; /* next element */ \
}

/*
 * Simple queue access methods.
 */
#define SIMPLEQ_FIRST(head) ((head)->sqh_first)
#define SIMPLEQ_END(head) NULL
#define SIMPLEQ_EMPTY(head) ((head)->sqh_first == SIMPLEQ_END(head))
#define SIMPLEQ_NEXT(elm, field) ((elm)->field.sqe_next)

#define SIMPLEQ_FOREACH(var, head, field) \
for ((var) = ((head)->sqh_first); \
     (var) != SIMPLEQ_END(head); \
     (var) = ((var)->field.sqe_next))

#define SIMPLEQ_FOREACH_SAFE(var, head, field, next) \
for ((var) = ((head)->sqh_first); \
     (var) != SIMPLEQ_END(head) && \
     ((next = ((var)->field.sqe_next)), 1); \
     (var) = (next))

/*
 * Simple queue functions.
 */
#define SIMPLEQ_INIT(head) do { \
    (head)->sqh_first = NULL; \
    (head)->sqh_last

```

```

= &(head)->sqh_first; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_HEAD(head, elm, field) do { \
if (((elm)->field.sqe_next = (head)->sqh_first) == NULL) \
(head)->sqh_last = &(elm)->field.sqe_next; \
(head)->sqh_first = (elm); \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_TAIL(head, elm, field) do { \
(elm)->field.sqe_next = NULL; \
*(head)->sqh_last = (elm); \
(head)->sqh_last = &(elm)->field.sqe_next; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_INSERT_AFTER(head, listelm, elm, field) do { \
if (((elm)->field.sqe_next = (listelm)->field.sqe_next) == NULL) \
(head)->sqh_last = &(elm)->field.sqe_next; \
(listelm)->field.sqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_HEAD(head, field) do { \
if (((head)->sqh_first = (head)->sqh_first->field.sqe_next) == NULL) \
(head)->sqh_last = &(head)->sqh_first; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE_AFTER(head, elm, field) do { \
if \
(((elm)->field.sqe_next = (elm)->field.sqe_next->field.sqe_next) \
== NULL) \
(head)->sqh_last = &(elm)->field.sqe_next; \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_REMOVE(head, elm, type, field) do { \
if ((head)->sqh_first == (elm)) { \
SIMPLEQ_REMOVE_HEAD((head), field); \
} else { \
struct type *curelm = (head)->sqh_first; \
while (curelm->field.sqe_next != (elm)) \
curelm = curelm->field.sqe_next; \
if ((curelm->field.sqe_next = \
curelm->field.sqe_next->field.sqe_next) == NULL) \
(head)->sqh_last = &(curelm)->field.sqe_next; \
} \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_CONCAT(head1, head2) do { \
if (!SIMPLEQ_EMPTY((head2))) { \

```

```

*(head1)->sqh_last = (head2)->sqh_first; \
(head1)->sqh_last = (head2)->sqh_last; \
SIMPLEQ_INIT((head2)); \
} \
} while (/*CONSTCOND*/0)

#define SIMPLEQ_LAST(head, type, field) \
(SIMPLEQ_EMPTY((head)) ? \
NULL : \
((struct type \
*)(void *) \
((char *)((head)->sqh_last) - offsetof(struct type, field))))

/*
 * Tail queue definitions.
 */
#define _TAILQ_HEAD(name, type, qual) \
struct name { \
    qual type *tqh_first; /* first element */ \
    qual type *qual *tqh_last; /* addr of last next element */ \
}
#define TAILQ_HEAD(name, type) _TAILQ_HEAD(name, struct type,)

#define TAILQ_HEAD_INITIALIZER(head) \
{ TAILQ_END(head), &(head).tqh_first }

#define _TAILQ_ENTRY(type, qual) \
struct { \
    qual type *tqe_next; /* next element */ \
    qual type *qual *tqe_prev; /* address of previous next element */ \
}
#define TAILQ_ENTRY(type) _TAILQ_ENTRY(struct type,)

/*
 * Tail queue access methods.
 */
#define TAILQ_FIRST(head) ((head)->tqh_first)
#define TAILQ_END(head) (NULL)
#define TAILQ_NEXT(elm, field) ((elm)->field.tqe_next)
#define TAILQ_LAST(head, headname) \
(*(((struct headname *) (void *)((head)->tqh_last))->tqh_last))
#define TAILQ_PREV(elm, headname, field) \
(*(((struct headname \
*)(void *)((elm)->field.tqe_prev))->tqh_last))
#define TAILQ_EMPTY(head) (TAILQ_FIRST(head) == TAILQ_END(head))

#define TAILQ_FOREACH(var, head, field) \

```

```

for ((var) = ((head)->tqh_first); \
    (var) != TAILQ_END(head); \
    (var) = ((var)->field.tqe_next))

#define TAILQ_FOREACH_SAFE(var, head, field, next) \
for ((var) = ((head)->tqh_first); \
    (var) != TAILQ_END(head) && \
    ((next) = TAILQ_NEXT(var, field), 1); (var) = (next))

#define TAILQ_FOREACH_REVERSE(var, head, headname, field) \
for ((var) = TAILQ_LAST((head), headname); \
    (var) != TAILQ_END(head); \
    (var) = TAILQ_PREV((var), headname, field))

#define TAILQ_FOREACH_REVERSE_SAFE(var, head, headname, field, prev) \
for ((var) = TAILQ_LAST((head), headname); \
    (var) != TAILQ_END(head) && \
    ((prev) = TAILQ_PREV((var), headname, field), 1); (var) = (prev))

/*
 * Tail queue functions.
 */
#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, \
    elm, field) \
    if ((head)->tqh_first && \
        (head)->tqh_first->field.tqe_prev != &(head)->tqh_first) \
        QUEUEDEBUG_ABORT("TAILQ_INSERT_HEAD %p %s:%d", (head), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field) \
    if (*(head)->tqh_last != NULL) \
        QUEUEDEBUG_ABORT("TAILQ_INSERT_TAIL %p %s:%d", (head), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_OP(elm, field) \
    if ((elm)->field.tqe_next && \
        (elm)->field.tqe_next->field.tqe_prev != \
            &(elm)->field.tqe_next) \
        QUEUEDEBUG_ABORT("TAILQ_* forw %p %s:%d", (elm), \
            __FILE__, __LINE__); \
    if (*(elm)->field.tqe_prev != (elm)) \
        QUEUEDEBUG_ABORT("TAILQ_* back %p %s:%d", (elm), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field) \
    if ((elm)->field.tqe_next == NULL && \
        (head)->tqh_last != &(elm)->field.tqe_next) \
        QUEUEDEBUG_ABORT("TAILQ_PREREMOVE head %p elm %p %s:%d", \
            (head), (elm), __FILE__,

```

```

__LINE__);
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field) \
    (elm)->field.tqe_next = (void *)1L; \
    (elm)->field.tqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_TAILQ_INSERT_HEAD(head, elm, field)
#define QUEUEDEBUG_TAILQ_INSERT_TAIL(head, elm, field)
#define QUEUEDEBUG_TAILQ_OP(elm, field)
#define QUEUEDEBUG_TAILQ_PREREMOVE(head, elm, field)
#define QUEUEDEBUG_TAILQ_POSTREMOVE(elm, field)
#endif

#define TAILQ_INIT(head) do { \
    (head)->tqh_first = TAILQ_END(head); \
    (head)->tqh_last = &(head)->tqh_first; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_HEAD(head, elm, field) do { \
    QUEUEDEBUG_TAILQ_INSERT_HEAD((head), (elm), field) \
    if (((elm)->field.tqe_next = (head)->tqh_first) != TAILQ_END(head)) \
        (head)->tqh_first->field.tqe_prev = \
            &(elm)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm)->field.tqe_next; \
    (head)->tqh_first = (elm); \
    (elm)->field.tqe_prev = &(head)->tqh_first; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_TAIL(head, \
    elm, field) do { \
    QUEUEDEBUG_TAILQ_INSERT_TAIL((head), (elm), field) \
    (elm)->field.tqe_next = TAILQ_END(head); \
    (elm)->field.tqe_prev = (head)->tqh_last; \
    *(head)->tqh_last = (elm); \
    (head)->tqh_last = &(elm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

#define TAILQ_INSERT_AFTER(head, listelm, elm, field) do { \
    QUEUEDEBUG_TAILQ_OP((listelm), field) \
    if (((elm)->field.tqe_next = (listelm)->field.tqe_next) != \
        TAILQ_END(head)) \
        (elm)->field.tqe_next->field.tqe_prev = \
            &(elm)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm)->field.tqe_next; \
    (listelm)->field.tqe_next = (elm); \
    (elm)->field.tqe_prev = &(listelm)->field.tqe_next; \
} while (/*CONSTCOND*/0)

```

```
#define TAILQ_INSERT_BEFORE(listelm, elm, field) do { \
    QUEUEDEBUG_TAILQ_OP((listelm), field) \
    (elm)->field.tqe_prev = (listelm)->field.tqe_prev; \
    (elm)->field.tqe_next = (listelm); \
    *(listelm)->field.tqe_prev = (elm); \
    (listelm)->field.tqe_prev \
    = &(elm)->field.tqe_next; \
} while (/*CONSTCOND*/0)
```

```
#define TAILQ_REMOVE(head, elm, field) do { \
    QUEUEDEBUG_TAILQ_PREREMOVE((head), (elm), field) \
    QUEUEDEBUG_TAILQ_OP((elm), field) \
    if (((elm)->field.tqe_next != TAILQ_END(head)) \
        (elm)->field.tqe_next->field.tqe_prev = \
            (elm)->field.tqe_prev; \
    else \
        (head)->tqh_last = (elm)->field.tqe_prev; \
    *(elm)->field.tqe_prev = (elm)->field.tqe_next; \
    QUEUEDEBUG_TAILQ_POSTREMOVE((elm), field); \
} while (/*CONSTCOND*/0)
```

```
#define TAILQ_REPLACE(head, elm, elm2, field) do { \
    if (((elm2)->field.tqe_next = (elm)->field.tqe_next) != \
        TAILQ_END(head)) \
        (elm2)->field.tqe_next->field.tqe_prev = \
            &(elm2)->field.tqe_next; \
    else \
        (head)->tqh_last = &(elm2)->field.tqe_next; \
    (elm2)->field.tqe_prev = (elm)->field.tqe_prev; \
    *(elm2)->field.tqe_prev = (elm2); \
    QUEUEDEBUG_TAILQ_POSTREMOVE((elm), \
        field); \
} while (/*CONSTCOND*/0)
```

```
#define TAILQ_CONCAT(head1, head2, field) do { \
    if (!TAILQ_EMPTY(head2)) { \
        *(head1)->tqh_last = (head2)->tqh_first; \
        (head2)->tqh_first->field.tqe_prev = (head1)->tqh_last; \
        (head1)->tqh_last = (head2)->tqh_last; \
        TAILQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)
```

```
/*
 * Singly-linked Tail queue declarations.
 */
```

```
#define STAILQ_HEAD(name, type) \
```

```

struct name {      \
    struct type *stqh_first; /* first element */ \
    struct type **stqh_last; /* addr of last next element */ \
}

#define STAILQ_HEAD_INITIALIZER(head)      \
    { NULL, &(head).stqh_first }

#define STAILQ_ENTRY(type)      \
    struct {      \
        struct type *stqe_next; /* next element */ \
    }

/*
 * Singly-linked Tail queue access methods.
 */
#define STAILQ_FIRST(head) ((head)->stqh_first)
#define STAILQ_END(head) NULL
#define STAILQ_NEXT(elm, field) ((elm)->field.stqe_next)
#define STAILQ_EMPTY(head) (STAILQ_FIRST(head) == STAILQ_END(head))

/*
 * Singly-linked Tail queue functions.
 */
#define STAILQ_INIT(head) do {      \
    (head)->stqh_first = NULL;      \
    (head)->stqh_last = &(head)->stqh_first; \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_HEAD(head, elm, field) do {      \
    if (((elm)->field.stqe_next = (head)->stqh_first) == NULL) \
        (head)->stqh_last = &(elm)->field.stqe_next; \
    (head)->stqh_first = (elm);      \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_TAIL(head, elm, field) do {      \
    (elm)->field.stqe_next = NULL;      \
    *(head)->stqh_last = (elm);      \
    (head)->stqh_last = &(elm)->field.stqe_next; \
} while (/*CONSTCOND*/0)

#define STAILQ_INSERT_AFTER(head, listelm, elm, field) do {      \
    if (((elm)->field.stqe_next = (listelm)->field.stqe_next) == NULL) \
        (head)->stqh_last = &(elm)->field.stqe_next; \
    (listelm)->field.stqe_next = (elm);      \
} while (/*CONSTCOND*/0)

```

```

#define STAILQ_REMOVE_HEAD(head, field) do { \
    if (((head)->stqh_first = (head)->stqh_first->field.stqe_next) \
        == NULL) \
        (head)->stqh_last = &(head)->stqh_first; \
} while (/*CONSTCOND*/0)

#define STAILQ_REMOVE(head, elm, type, field) do { \
    if ((head)->stqh_first == (elm)) { \
        STAILQ_REMOVE_HEAD((head), field); \
    } else { \
        struct type *curelm = (head)->stqh_first; \
        while (curelm->field.stqe_next != (elm)) \
            curelm = curelm->field.stqe_next; \
        if ((curelm->field.stqe_next = \
            curelm->field.stqe_next->field.stqe_next) == NULL) \
            (head)->stqh_last = &(curelm)->field.stqe_next; \
    } \
} while (/*CONSTCOND*/0)

#define STAILQ_FOREACH(var, head, field) \
    for ((var) = ((head)->stqh_first); \
        (var); \
        (var) = ((var)->field.stqe_next))

#define STAILQ_FOREACH_SAFE(var, head, field, tvar) \
    for ((var) = STAILQ_FIRST((head)); \
        (var) && ((tvar) = STAILQ_NEXT((var), field), 1); \
        (var) = (tvar))

#define STAILQ_CONCAT(head1, head2) do { \
    if (!STAILQ_EMPTY((head2))) { \
        *(head1)->stqh_last \
        = (head2)->stqh_first; \
        (head1)->stqh_last = (head2)->stqh_last; \
        STAILQ_INIT((head2)); \
    } \
} while (/*CONSTCOND*/0)

#define STAILQ_LAST(head, type, field) \
    (STAILQ_EMPTY((head)) ? \
    NULL : \
    ((struct type *) (void *) \
    ((char *) ((head)->stqh_last) - offsetof(struct type, field))))

#ifdef _KERNEL
/*
 * Circular queue definitions. Do not use. We still keep the macros

```



```

* for compatibility but because of pointer aliasing issues their use
* is discouraged!
*/

/*
* __launder_type(): We use this ugly hack to work around the the compiler
* noticing that two types may not alias each other and elide tests in code.
* We hit this in the CIRCLEQ macros when comparing 'struct name *' and
* 'struct type *' (see CIRCLEQ_HEAD()). Modern compilers (such as GCC
* 4.8) declare these comparisons as always false, causing the code to
* not run as designed.
*
* This hack is only to be used for comparisons and thus can be fully const.
* Do not use for assignment.
*
* If we ever choose to change the ABI of the CIRCLEQ macros, we could fix
* this by changing the head/tail sentinel values, but see the note above
* this one.
*/
static __inline const void * __launder_type(const void *);
static __inline const void *
__launder_type(const void *__x)
{
    __asm __volatile("" : "+r" (__x));
    return __x;
}

#if defined(QUEUEDEBUG)
#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field) \
    if ((head)->cqh_first != CIRCLEQ_ENDC(head) && \
        (head)->cqh_first->field.cqe_prev != CIRCLEQ_ENDC(head)) \
        QUEUEDEBUG_ABORT("CIRCLEQ head forw %p %s:%d", (head), \
            __FILE__, __LINE__); \
    if ((head)->cqh_last != CIRCLEQ_ENDC(head) && \
        (head)->cqh_last->field.cqe_next != CIRCLEQ_ENDC(head)) \
        QUEUEDEBUG_ABORT("CIRCLEQ head back %p %s:%d", (head), \
            __FILE__, __LINE__);
#define QUEUEDEBUG_CIRCLEQ_ELM(head, elm, field) \
    if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) { \
        if ((head)->cqh_last != (elm)) \
            QUEUEDEBUG_ABORT("CIRCLEQ \
elm last %p %s:%d", \
                (elm), __FILE__, __LINE__); \
    } else { \
        if ((elm)->field.cqe_next->field.cqe_prev != (elm)) \
            QUEUEDEBUG_ABORT("CIRCLEQ elm forw %p %s:%d", \
                (elm), __FILE__, __LINE__); \
    } \

```

```

if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) { \
    if ((head)->cqh_first != (elm)) \
        QUEUEDEBUG_ABORT("CIRCLEQ elm first %p %s:%d", \
            (elm), __FILE__, __LINE__); \
} else { \
    if ((elm)->field.cqe_prev->field.cqe_next != (elm)) \
        QUEUEDEBUG_ABORT("CIRCLEQ elm prev %p %s:%d", \
            (elm), __FILE__, __LINE__); \
}
#define QUEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field) \
    (elm)->field.cqe_next = (void *)1L; \
    (elm)->field.cqe_prev = (void *)1L;
#else
#define QUEUEDEBUG_CIRCLEQ_HEAD(head, field)
#define QUEUEDEBUG_CIRCLEQ_ELM(head, elm, field)
#define QUEUEDEBUG_CIRCLEQ_POSTREMOVE(elm, field)
#endif

#define CIRCLEQ_HEAD(name, type) \
    struct name { \
        struct type *cqh_first; /* \
        first element */ \
        struct type *cqh_last; /* last element */ \
    }

#define CIRCLEQ_HEAD_INITIALIZER(head) \
    { CIRCLEQ_END(&head), CIRCLEQ_END(&head) }

#define CIRCLEQ_ENTRY(type) \
    struct { \
        struct type *cqe_next; /* next element */ \
        struct type *cqe_prev; /* previous element */ \
    }

/*
 * Circular queue functions.
 */
#define CIRCLEQ_INIT(head) do { \
    (head)->cqh_first = CIRCLEQ_END(head); \
    (head)->cqh_last = CIRCLEQ_END(head); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_AFTER(head, listelm, elm, field) do { \
    QUEUEDEBUG_CIRCLEQ_HEAD((head), field) \
    QUEUEDEBUG_CIRCLEQ_ELM((head), (listelm), field) \
    (elm)->field.cqe_next = (listelm)->field.cqe_next; \
    (elm)->field.cqe_prev = (listelm); \
    if ((listelm)->field.cqe_next == CIRCLEQ_ENDC(head)) \

```

```

(head)->cqh_last = (elm); \
else \
(listelm)->field.cqe_next->field.cqe_prev = (elm); \
(listelm)->field.cqe_next = (elm); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_BEFORE(head,
listelm, elm, field) do { \
QUEUEDEBBUG_CIRCLEQ_HEAD((head), field) \
QUEUEDEBBUG_CIRCLEQ_ELM((head), (listelm), field) \
(elm)->field.cqe_next = (listelm); \
(elm)->field.cqe_prev = (listelm)->field.cqe_prev; \
if ((listelm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
(head)->cqh_first = (elm); \
else \
(listelm)->field.cqe_prev->field.cqe_next = (elm); \
(listelm)->field.cqe_prev = (elm); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_HEAD(head, elm, field) do { \
QUEUEDEBBUG_CIRCLEQ_HEAD((head), field) \
(elm)->field.cqe_next = (head)->cqh_first; \
(elm)->field.cqe_prev = CIRCLEQ_END(head); \
if ((head)->cqh_last == CIRCLEQ_ENDC(head)) \
(head)->cqh_last = (elm); \
else \
(head)->cqh_first->field.cqe_prev = (elm); \
(head)->cqh_first = (elm); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_INSERT_TAIL(head, elm, field) do { \
QUEUEDEBBUG_CIRCLEQ_HEAD((head), field) \
(elm)->field.cqe_next
= CIRCLEQ_END(head); \
(elm)->field.cqe_prev = (head)->cqh_last; \
if ((head)->cqh_first == CIRCLEQ_ENDC(head)) \
(head)->cqh_first = (elm); \
else \
(head)->cqh_last->field.cqe_next = (elm); \
(head)->cqh_last = (elm); \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_REMOVE(head, elm, field) do { \
QUEUEDEBBUG_CIRCLEQ_HEAD((head), field) \
QUEUEDEBBUG_CIRCLEQ_ELM((head), (elm), field) \
if ((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
(head)->cqh_last = (elm)->field.cqe_prev; \
else \

```

```

(elm)->field.cqe_next->field.cqe_prev = \
    (elm)->field.cqe_prev; \
if ((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
    (head)->cqh_first = (elm)->field.cqe_next; \
else \
    (elm)->field.cqe_prev->field.cqe_next = \
        (elm)->field.cqe_next; \
QUEUEDEBUG_CIRCLEQ_POSTREMOVE((elm), field) \
} while (/*CONSTCOND*/0)

#define CIRCLEQ_FOREACH(var, head, field) \
for ((var) = ((head)->cqh_first); \
    (var) != CIRCLEQ_ENDC(head); \
    (var) \
    = ((var)->field.cqe_next))

#define CIRCLEQ_FOREACH_REVERSE(var, head, field) \
for ((var) = ((head)->cqh_last); \
    (var) != CIRCLEQ_ENDC(head); \
    (var) = ((var)->field.cqe_prev))

/*
 * Circular queue access methods.
 */
#define CIRCLEQ_FIRST(head) ((head)->cqh_first)
#define CIRCLEQ_LAST(head) ((head)->cqh_last)
/* For comparisons */
#define CIRCLEQ_ENDC(head) (__launter_type(head))
/* For assignments */
#define CIRCLEQ_END(head) ((void *)(head))
#define CIRCLEQ_NEXT(elm, field) ((elm)->field.cqe_next)
#define CIRCLEQ_PREV(elm, field) ((elm)->field.cqe_prev)
#define CIRCLEQ_EMPTY(head) \
    (CIRCLEQ_FIRST(head) == CIRCLEQ_ENDC(head))

#define CIRCLEQ_LOOP_NEXT(head, elm, field) \
    (((elm)->field.cqe_next == CIRCLEQ_ENDC(head)) \
    ? ((head)->cqh_first) \
    : (elm)->field.cqe_next)
#define CIRCLEQ_LOOP_PREV(head, elm, field) \
    (((elm)->field.cqe_prev == CIRCLEQ_ENDC(head)) \
    ? ((head)->cqh_last) \
    : \
    (elm)->field.cqe_prev)
#endif /* !_KERNEL */

#endif /* !_SYS_QUEUE_H_ */

```

```

#ifndef _SYS_CDEFS_H_
#define _SYS_CDEFS_H_

#warning usage of non-standard #include <sys/cdefs.h> is deprecated

#undef __P
#undef __PMT

#define __P(args) args
#define __PMT(args) args

#define __CONCAT(x,y) x ## y
#define __STRING(x) #x

#ifdef __cplusplus
# define __BEGIN_DECLS extern "C" {
# define __END_DECLS }
#else
# define __BEGIN_DECLS
# define __END_DECLS
#endif

#if defined(__GNUC__) && !defined(__cplusplus)
# define __THROW __attribute__((__nothrow__))
# define __NTH(fct) __attribute__((__nothrow__)) fct
#else
# define __THROW
# define __NTH(fct) fct
#endif

#endif /* _SYS_CDEFS_H_ */

# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir"/$_pkgname-$pkgver

```

```

replaces="py-flake8-copyright"          # Backwards compatibility
provides="py-flake8-copyright=$pkgver-r$pkgrel" # Backwards compatibility

build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}

package() {
    gpep517 install-wheel --destdir "$pkgdir" \
        .dist/*.whl
}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
    flake8-copyright-0.2.4.tar.gz
"

# Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"
url="https://github.com/onur/cargo-license"
arch="all"
license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

prepare() {
    default_prepare

    cargo fetch --target="$CTARGET" --locked
}

build() {
    cargo auditable build --release --frozen
}

check() {
    cargo test --frozen
}

package() {
    install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

```

```

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5 cargo-license-0.6.1.tar.gz
"

/* $NetBSD: tree.h,v 1.20 2013/09/14 13:20:45 joerg Exp $ */
/* $OpenBSD: tree.h,v 1.13 2011/07/09 00:19:45 pirofti Exp $ */
/* Modified by Void Linux. */
/*
* Copyright 2002 Niels Provos <provos@citi.umich.edu>
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE
* AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

#ifndef _SYS_TREE_H_
#define _SYS_TREE_H_

#ifdef __GNUC__
#define __GNUC_PREREQ__(x, y) \
    ((__GNUC__ == (x) && __GNUC_MINOR__ >= (y)) || \
     (__GNUC__ > (x)))
#else
#define __GNUC_PREREQ__(x, y) 0
#endif

#if __GNUC_PREREQ__(2, 7) || defined(__lint__)
#define _sys_tree_h_unused __attribute__((__unused__))
#else
#define _sys_tree_h_unused /* delete */

```

```

#endif

/*
 * This file defines data structures for different types of trees:
 * splay trees and red-black trees.
 *
 * A splay tree is a self-organizing
 * data structure. Every operation
 * on the tree causes a splay to happen. The splay moves the requested
 * node to the root of the tree and partly rebalances it.
 *
 * This has the benefit that request locality causes faster lookups as
 * the requested nodes move to the top of the tree. On the other hand,
 * every lookup causes memory writes.
 *
 * The Balance Theorem bounds the total access time for m operations
 * and n inserts on an initially empty tree as  $O((m + n)\lg n)$ . The
 * amortized cost for a sequence of m accesses to a splay tree is  $O(\lg n)$ ;
 *
 * A red-black tree is a binary search tree with the node color as an
 * extra attribute. It fulfills a set of conditions:
 * - every search path from the root to a leaf consists of the
 *   same number of black nodes,
 * - each red node (except for the root) has a black parent,
 * - each leaf node is black.
 *
 * Every operation on a red-black tree is bounded as  $O(\lg n)$ .
 * The maximum height of a red-black tree is  $2\lg(n+1)$ .
 */

#define SPLAY_HEAD(name, type) \
struct name { \
    struct type *sph_root; /* root of the tree */ \
}

#define SPLAY_INITIALIZER(root) \
{ NULL }

#define SPLAY_INIT(root) do { \
    (root)->sph_root = NULL; \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ENTRY(type) \
struct { \
    struct type *spe_left; /* left element */ \
    struct type *spe_right; /* right element */ \
}

```



```

#define SPLAY_LEFT(elm, field) (elm)->field.spe_left
#define SPLAY_RIGHT(elm, field) (elm)->field.spe_right
#define SPLAY_ROOT(head) (head)->sph_root
#define SPLAY_EMPTY(head) (SPLAY_ROOT(head) == NULL)

/* SPLAY_ROTATE_{LEFT,RIGHT} expect that tmp hold SPLAY_{RIGHT,LEFT} */
#define SPLAY_ROTATE_RIGHT(head, tmp, field) do { \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(tmp, field); \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ROTATE_LEFT(head, tmp, field) do { \
    SPLAY_RIGHT((head)->sph_root, \
    field) = SPLAY_LEFT(tmp, field); \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKLEFT(head, tmp, field) do { \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \
    (head)->sph_root = SPLAY_LEFT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKRIGHT(head, tmp, field) do { \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \
    (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ASSEMBLE(head, node, left, right, field) do { \
    SPLAY_RIGHT(left, field) = SPLAY_LEFT((head)->sph_root, field); \
    SPLAY_LEFT(right, field) = SPLAY_RIGHT((head)->sph_root, field); \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(node, field); \
    SPLAY_RIGHT((head)->sph_root, field) = SPLAY_LEFT(node, field); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */

#define
    SPLAY_PROTOTYPE(name, type, field, cmp) \
void name##_SPLAY(struct name *, struct type *); \
void name##_SPLAY_MINMAX(struct name *, int); \
struct type *name##_SPLAY_INSERT(struct name *, struct type *); \
struct type *name##_SPLAY_REMOVE(struct name *, struct type *); \
\
/* Finds the node with the same key as elm */ \

```

```

static __inline struct type * \
name##_SPLAY_FIND(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) \
        return(NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) \
        return (head->sph_root); \
    return (NULL); \
} \
\
static __inline _sys_tree_h_unused struct type * \
name##_SPLAY_NEXT(struct name *head, struct type *elm) \
{ \
    name##_SPLAY(head, elm); \
    if (SPLAY_RIGHT(elm, field) != NULL) { \
        elm = SPLAY_RIGHT(elm, field); \
        while (SPLAY_LEFT(elm, field) != NULL) { \
            elm = SPLAY_LEFT(elm, field); \
        } \
    } \
    else \
        elm = NULL; \
    return (elm); \
} \
\
static _sys_tree_h_unused __inline struct type * \
name##_SPLAY_MIN_MAX(struct name *head, int val) \
{ \
    name##_SPLAY_MINMAX(head, val); \
    return (SPLAY_ROOT(head)); \
}

/* Main splay operation.
 * Moves node close to the key of elm to top
 */
#define SPLAY_GENERATE(name, type, field, cmp) \
struct type * \
name##_SPLAY_INSERT(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) { \
        SPLAY_LEFT(elm, field) = SPLAY_RIGHT(elm, field) = NULL; \
    } else { \
        int __comp; \
        name##_SPLAY(head, elm); \
        __comp = (cmp)(elm, (head)->sph_root); \
        if (__comp < 0) { \
            SPLAY_LEFT(elm, field) = SPLAY_LEFT((head)->sph_root, field); \

```

```

        SPLAY_RIGHT(elm, field) = (head)->sph_root; \
        SPLAY_LEFT((head)->sph_root, field) = NULL; \
    } else if (__comp > 0) { \
        SPLAY_RIGHT(elm,
field) = SPLAY_RIGHT((head)->sph_root, field);\
        SPLAY_LEFT(elm, field) = (head)->sph_root; \
        SPLAY_RIGHT((head)->sph_root, field) = NULL; \
    } else \
        return ((head)->sph_root); \
    } \
    (head)->sph_root = (elm); \
    return (NULL); \
} \
\
struct type * \
name##_SPLAY_REMOVE(struct name *head, struct type *elm) \
{ \
    struct type *__tmp; \
    if (SPLAY_EMPTY(head)) \
        return (NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) { \
        if (SPLAY_LEFT((head)->sph_root, field) == NULL) { \
            (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field);\
        } else { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field); \
            (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);\
            name##_SPLAY(head, elm); \
            SPLAY_RIGHT((head)->sph_root, field) = __tmp; \
        } \
        return (elm); \
    } \
    return (NULL); \
} \
\
void \
name##_SPLAY(struct
name *head, struct type *elm) \
{ \
    struct type __node, *__left, *__right, *__tmp; \
    int __comp; \
    \
    SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
    __left = __right = &__node; \
    \
    while ((__comp = (cmp)(elm, (head)->sph_root)) != 0) { \
        if (__comp < 0) { \
            __tmp = SPLAY_LEFT((head)->sph_root, field); \

```

```

if (__tmp == NULL) \
    break; \
if ((cmp)(elm, __tmp) < 0){ \
    SPLAY_ROTATE_RIGHT(head, __tmp, field); \
    if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
        break; \
} \
SPLAY_LINKLEFT(head, __right, field); \
} else if (__comp > 0) { \
    __tmp = SPLAY_RIGHT((head)->sph_root, field); \
    if (__tmp == NULL) \
        break; \
    if ((cmp)(elm, __tmp) > 0){ \
        SPLAY_ROTATE_LEFT(head, __tmp, field); \
        if (SPLAY_RIGHT((head)->sph_root, field) == NULL)\
            break; \
    } \
    SPLAY_LINKRIGHT(head, __left, field); \
} \
} \
SPLAY_ASSEMBLE(head,
&__node, __left, __right, field); \
} \
\
/* Splay with either the minimum or the maximum element \
* Used to find minimum or maximum element in tree. \
*/ \
void name##_SPLAY_MINMAX(struct name *head, int __comp) \
{ \
    struct type __node, *__left, *__right, *__tmp; \
    \
    SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL; \
    __left = __right = &__node; \
    \
    while (1) { \
        if (__comp < 0) { \
            __tmp = SPLAY_LEFT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if (__comp < 0){ \
                SPLAY_ROTATE_RIGHT(head, __tmp, field); \
                if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
                    break; \
            } \
            SPLAY_LINKLEFT(head, __right, field); \
        } else if (__comp > 0) { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field); \
            if (__tmp == NULL) \

```

```

        break; \
    if (__comp > 0) { \
        SPLAY_ROTATE_LEFT(head, __tmp, field); \
        if (SPLAY_RIGHT((head)->sph_root,
field) == NULL)\
            break; \
    } \
    SPLAY_LINKRIGHT(head, __left, field); \
} \
} \
SPLAY_ASSEMBLE(head, &__node, __left, __right, field); \
}

#define SPLAY_NEGINF -1
#define SPLAY_INF 1

#define SPLAY_INSERT(name, x, y) name##_SPLAY_INSERT(x, y)
#define SPLAY_REMOVE(name, x, y) name##_SPLAY_REMOVE(x, y)
#define SPLAY_FIND(name, x, y) name##_SPLAY_FIND(x, y)
#define SPLAY_NEXT(name, x, y) name##_SPLAY_NEXT(x, y)
#define SPLAY_MIN(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_NEGINF))
#define SPLAY_MAX(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_INF))

#define SPLAY_FOREACH(x, name, head) \
    for ((x) = SPLAY_MIN(name, head); \
        (x) != NULL; \
        (x) = SPLAY_NEXT(name, head, x))

/* Macros that define a red-black tree */
#define RB_HEAD(name, type) \
struct name { \
    struct type *rbh_root; /* root of the tree */ \
}

#define RB_INITIALIZER(root) \
{ \
    NULL }

#define RB_INIT(root) do { \
    (root)->rbh_root = NULL; \
} while (/*CONSTCOND*/ 0)

#define RB_BLACK 0
#define RB_RED 1
#define RB_ENTRY(type) \
struct { \

```

```

struct type *rbe_left; /* left element */ \
struct type *rbe_right; /* right element */ \
struct type *rbe_parent; /* parent element */ \
int rbe_color; /* node color */ \
}

#define RB_LEFT(elm, field) (elm)->field.rbe_left
#define RB_RIGHT(elm, field) (elm)->field.rbe_right
#define RB_PARENT(elm, field) (elm)->field.rbe_parent
#define RB_COLOR(elm, field) (elm)->field.rbe_color
#define RB_ROOT(head) (head)->rbh_root
#define RB_EMPTY(head) (RB_ROOT(head) == NULL)

#define RB_SET(elm, parent, field) do { \
    RB_PARENT(elm, field) = parent; \
    RB_LEFT(elm, field) = RB_RIGHT(elm, field) = NULL; \
    RB_COLOR(elm, field) = RB_RED; \
} while (/*CONSTCOND*/ 0)

#define RB_SET_BLACKRED(black, red, field) do { \
    RB_COLOR(black, field) = RB_BLACK; \
    RB_COLOR(red, field) \
    = RB_RED; \
} while (/*CONSTCOND*/ 0)

#ifndef RB_AUGMENT
#define RB_AUGMENT(x) do {} while (/*CONSTCOND*/ 0)
#endif

#define RB_ROTATE_LEFT(head, elm, tmp, field) do { \
    (tmp) = RB_RIGHT(elm, field); \
    if ((RB_RIGHT(elm, field) = RB_LEFT(tmp, field)) != NULL) { \
        RB_PARENT(RB_LEFT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_LEFT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
    RB_AUGMENT(tmp); \
    if ((RB_PARENT(tmp, field))) \
        RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

```

```

#define RB_ROTATE_RIGHT(head, elm, tmp, field) do { \
    (tmp) = RB_LEFT(elm, field); \
    if ((RB_LEFT(elm, field) = RB_RIGHT(tmp, field)) \
    != NULL) { \
        RB_PARENT(RB_RIGHT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_RIGHT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
    RB_AUGMENT(tmp); \
    if ((RB_PARENT(tmp, field))) \
        RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */
#define RB_PROTOTYPE(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp,)
#define RB_PROTOTYPE_STATIC(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_PROTOTYPE_INTERNAL(name, type, field, cmp, attr) \
    attr void name##_RB_INSERT_COLOR(struct name *, struct \
    type *); \
    attr void name##_RB_REMOVE_COLOR(struct name *, struct type *, struct type *); \
    attr struct type *name##_RB_REMOVE(struct name *, struct type *); \
    attr struct type *name##_RB_INSERT(struct name *, struct type *); \
    attr struct type *name##_RB_FIND(struct name *, struct type *); \
    attr struct type *name##_RB_NFIND(struct name *, struct type *); \
    attr struct type *name##_RB_NEXT(struct type *); \
    attr struct type *name##_RB_PREV(struct type *); \
    attr struct type *name##_RB_MINMAX(struct name *, int); \
    \

/* Main rb operation.
 * Moves node close to the key of elm to top
 */
#define RB_GENERATE(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp,)
#define RB_GENERATE_STATIC(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_GENERATE_INTERNAL(name, type, field, cmp, attr) \

```

```

attr void \
name##_RB_INSERT_COLOR(struct name *head, struct type *elm) \
{ \
    struct type
    *parent, *gparent, *tmp; \
    while ((parent = RB_PARENT(elm, field)) != NULL && \
        RB_COLOR(parent, field) == RB_RED) { \
        gparent = RB_PARENT(parent, field); \
        if (parent == RB_LEFT(gparent, field)) { \
            tmp = RB_RIGHT(gparent, field); \
            if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
                RB_COLOR(tmp, field) = RB_BLACK; \
                RB_SET_BLACKRED(parent, gparent, field);\
                elm = gparent; \
                continue; \
            } \
            if (RB_RIGHT(parent, field) == elm) { \
                RB_ROTATE_LEFT(head, parent, tmp, field);\
                tmp = parent; \
                parent = elm; \
                elm = tmp; \
            } \
            RB_SET_BLACKRED(parent, gparent, field); \
            RB_ROTATE_RIGHT(head, gparent, tmp, field); \
        } else { \
            tmp = RB_LEFT(gparent, field); \
            if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
                RB_COLOR(tmp, field) = RB_BLACK; \
                RB_SET_BLACKRED(parent, gparent, field);\
                elm = gparent; \
                continue; \
            } \
            if (RB_LEFT(parent, field) == elm)
{ \
                RB_ROTATE_RIGHT(head, parent, tmp, field);\
                tmp = parent; \
                parent = elm; \
                elm = tmp; \
            } \
            RB_SET_BLACKRED(parent, gparent, field); \
            RB_ROTATE_LEFT(head, gparent, tmp, field); \
        } \
    } \
    RB_COLOR(head->rbh_root, field) = RB_BLACK; \
} \
\
attr void \
name##_RB_REMOVE_COLOR(struct name *head, struct type *parent, struct type *elm) \

```



```

{
    \
    struct type *tmp;
    \
    while ((elm == NULL || RB_COLOR(elm, field) == RB_BLACK) && \
        elm != RB_ROOT(head)) {
        \
        if (RB_LEFT(parent, field) == elm) {
            \
            tmp = RB_RIGHT(parent, field);
            \
            if (RB_COLOR(tmp, field) == RB_RED) {
                \
                RB_SET_BLACKRED(tmp, parent, field);
                \
                RB_ROTATE_LEFT(head, parent, tmp, field);
                \
                tmp = RB_RIGHT(parent, field);
            }
            \
            if ((RB_LEFT(tmp, field) == NULL || \
                RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) && \
                (RB_RIGHT(tmp, field) == NULL || \
                RB_COLOR(RB_RIGHT(tmp,
field), field) == RB_BLACK)) {
                \
                RB_COLOR(tmp, field) = RB_RED;
                \
                elm = parent;
                \
                parent = RB_PARENT(elm, field);
            } else {
                \
                if (RB_RIGHT(tmp, field) == NULL || \
                    RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK) {
                    \
                    struct type *oleft;
                    \
                    if ((oleft = RB_LEFT(tmp, field)) \
                        != NULL) {
                        \
                        RB_COLOR(oleft, field) = RB_BLACK;
                        \
                        RB_COLOR(tmp, field) = RB_RED;
                        \
                        RB_ROTATE_RIGHT(head, tmp, oleft, field);
                        \
                        tmp = RB_RIGHT(parent, field);
                    }
                    \
                    RB_COLOR(tmp, field) = RB_COLOR(parent, field);
                    \
                    RB_COLOR(parent, field) = RB_BLACK;
                    \
                    if (RB_RIGHT(tmp, field)) {
                        \
                        RB_COLOR(RB_RIGHT(tmp, field), field) = RB_BLACK;
                        \
                        RB_ROTATE_LEFT(head, parent, tmp, field);
                        \
                        elm = RB_ROOT(head);
                        \
                        break;
                    }
                    \
                } else {
                    \
                    tmp = RB_LEFT(parent, field);
                    \
                    if (RB_COLOR(tmp, field) == RB_RED) {
                        \
                        RB_SET_BLACKRED(tmp, parent, field);
                        \
                        RB_ROTATE_RIGHT(head, parent, tmp, field);
                        \
                        tmp
= RB_LEFT(parent, field);
                    }
                    \
                    if ((RB_LEFT(tmp, field) == NULL || \
                        RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) && \

```

```

        (RB_RIGHT(tmp, field) == NULL || \
        RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
    RB_COLOR(tmp, field) = RB_RED; \
    elm = parent; \
    parent = RB_PARENT(elm, field); \
} else { \
    if (RB_LEFT(tmp, field) == NULL || \
        RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) {\
        struct type *oright; \
        if ((oright = RB_RIGHT(tmp, field)) \
            != NULL) \
            RB_COLOR(oright, field) = RB_BLACK;\
        RB_COLOR(tmp, field) = RB_RED; \
        RB_ROTATE_LEFT(head, tmp, oright, field);\
        tmp = RB_LEFT(parent, field); \
    } \
    RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
    RB_COLOR(parent, field) = RB_BLACK; \
    if (RB_LEFT(tmp, field)) \
        RB_COLOR(RB_LEFT(tmp, field), field) = RB_BLACK;\
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    elm = RB_ROOT(head); \
    break; \
} \
} \
} \
if
(elm) \
    RB_COLOR(elm, field) = RB_BLACK; \
} \
\
attr struct type * \
name##_RB_REMOVE(struct name *head, struct type *elm) \
{ \
    struct type *child, *parent, *old = elm; \
    int color; \
    if (RB_LEFT(elm, field) == NULL) \
        child = RB_RIGHT(elm, field); \
    else if (RB_RIGHT(elm, field) == NULL) \
        child = RB_LEFT(elm, field); \
    else { \
        struct type *left; \
        elm = RB_RIGHT(elm, field); \
        while ((left = RB_LEFT(elm, field)) != NULL) \
            elm = left; \
        child = RB_RIGHT(elm, field); \
        parent = RB_PARENT(elm, field); \
        color = RB_COLOR(elm, field); \

```

```

if (child)    \
    RB_PARENT(child, field) = parent; \
if (parent) {    \
    if (RB_LEFT(parent, field) == elm) \
        RB_LEFT(parent, field) = child; \
    else    \
        RB_RIGHT(parent, field) = child; \
    RB_AUGMENT(parent); \
} else    \
    RB_ROOT(head)
= child; \
if (RB_PARENT(elm, field) == old) \
    parent = elm; \
(elm)->field = (old)->field; \
if (RB_PARENT(old, field)) { \
    if (RB_LEFT(RB_PARENT(old, field), field) == old)\
        RB_LEFT(RB_PARENT(old, field), field) = elm;\
    else    \
        RB_RIGHT(RB_PARENT(old, field), field) = elm;\
    RB_AUGMENT(RB_PARENT(old, field)); \
} else    \
    RB_ROOT(head) = elm; \
RB_PARENT(RB_LEFT(old, field), field) = elm; \
if (RB_RIGHT(old, field)) \
    RB_PARENT(RB_RIGHT(old, field), field) = elm; \
if (parent) {    \
    left = parent; \
    do {    \
        RB_AUGMENT(left); \
    } while ((left = RB_PARENT(left, field)) != NULL); \
}    \
goto color; \
}    \
parent = RB_PARENT(elm, field); \
color = RB_COLOR(elm, field); \
if (child)    \
    RB_PARENT(child, field) = parent; \
if (parent) {    \
    if (RB_LEFT(parent, field) == elm) \
        RB_LEFT(parent, field) = child; \
    else    \
        RB_RIGHT(parent,
field) = child; \
    RB_AUGMENT(parent); \
} else    \
    RB_ROOT(head) = child; \
color:    \
if (color == RB_BLACK)    \

```

```

    name##_RB_REMOVE_COLOR(head, parent, child); \
    return (old); \
} \
\
/* Inserts a node into the RB tree */ \
attr struct type * \
name##_RB_INSERT(struct name *head, struct type *elm) \
{ \
    struct type *tmp; \
    struct type *parent = NULL; \
    int comp = 0; \
    tmp = RB_ROOT(head); \
    while (tmp) { \
        parent = tmp; \
        comp = (cmp)(elm, parent); \
        if (comp < 0) \
            tmp = RB_LEFT(tmp, field); \
        else if (comp > 0) \
            tmp = RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    RB_SET(elm, parent, field); \
    if (parent != NULL) { \
        if (comp < 0) \
            RB_LEFT(parent, field) = elm; \
        else \
            RB_RIGHT(parent, field) = elm; \
        RB_AUGMENT(parent); \
    } else \
        RB_ROOT(head) = elm; \
    name##_RB_INSERT_COLOR(head, \
    elm); \
    return (NULL); \
} \
\
/* Finds the node with the same key as elm */ \
attr struct type * \
name##_RB_FIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \
    int comp; \
    while (tmp) { \
        comp = cmp(elm, tmp); \
        if (comp < 0) \
            tmp = RB_LEFT(tmp, field); \
        else if (comp > 0) \
            tmp = RB_RIGHT(tmp, field); \
    }

```

```

else \
    return (tmp); \
} \
return (NULL); \
} \
\
/* Finds the first node greater than or equal to the search key */\
attr struct type * \
name##_RB_NFIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \
    struct type *res = NULL; \
    int comp; \
    while (tmp) { \
        comp = cmp(elm, tmp); \
        if (comp < 0) { \
            res = tmp; \
            tmp = RB_LEFT(tmp, field); \
        } \
        else if (comp > 0) \
            tmp = \
RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    return (res); \
} \
\
/* ARGSUSED */ \
attr struct type * \
name##_RB_NEXT(struct type *elm) \
{ \
    if (RB_RIGHT(elm, field)) { \
        elm = RB_RIGHT(elm, field); \
        while (RB_LEFT(elm, field)) \
            elm = RB_LEFT(elm, field); \
    } else { \
        if (RB_PARENT(elm, field) && \
            (elm == RB_LEFT(RB_PARENT(elm, field), field))) \
            elm = RB_PARENT(elm, field); \
        else { \
            while (RB_PARENT(elm, field) && \
                (elm == RB_RIGHT(RB_PARENT(elm, field), field))) \
                elm = RB_PARENT(elm, field); \
            elm = RB_PARENT(elm, field); \
        } \
    } \
    return (elm); \
} \

```

```

} \
\
/* ARGSUSED */ \
attr struct type * \
name##_RB_PREV(struct type *elm) \
{ \
if (RB_LEFT(elm, field)) { \
elm = RB_LEFT(elm, field); \
while (RB_RIGHT(elm, field)) \
elm = RB_RIGHT(elm, field); \
}
else { \
if (RB_PARENT(elm, field) && \
(elm == RB_RIGHT(RB_PARENT(elm, field), field))) \
elm = RB_PARENT(elm, field); \
else { \
while (RB_PARENT(elm, field) && \
(elm == RB_LEFT(RB_PARENT(elm, field), field))) \
elm = RB_PARENT(elm, field); \
elm = RB_PARENT(elm, field); \
} \
} \
return (elm); \
} \
\
attr struct type * \
name##_RB_MINMAX(struct name *head, int val) \
{ \
struct type *tmp = RB_ROOT(head); \
struct type *parent = NULL; \
while (tmp) { \
parent = tmp; \
if (val < 0) \
tmp = RB_LEFT(tmp, field); \
else \
tmp = RB_RIGHT(tmp, field); \
} \
return (parent); \
}

#define RB_NEGINF -1
#define RB_INF 1

#define RB_INSERT(name, x, y) name##_RB_INSERT(x, y)
#define RB_REMOVE(name, x, y) name##_RB_REMOVE(x, y)
#define RB_FIND(name, x, y) name##_RB_FIND(x, y)
#define RB_NFIND(name, x, y) name##_RB_NFIND(x,

```

```

y)
#define RB_NEXT(name, x, y) name##_RB_NEXT(y)
#define RB_PREV(name, x, y) name##_RB_PREV(y)
#define RB_MIN(name, x) name##_RB_MINMAX(x, RB_NEGINF)
#define RB_MAX(name, x) name##_RB_MINMAX(x, RB_INF)

#define RB_FOREACH(x, name, head) \
    for ((x) = RB_MIN(name, head); \
         (x) != NULL; \
         (x) = name##_RB_NEXT(x))

#define RB_FOREACH_FROM(x, name, y) \
    for ((x) = (y); \
         ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
         (x) = (y))

#define RB_FOREACH_SAFE(x, name, head, y) \
    for ((x) = RB_MIN(name, head); \
         ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
         (x) = (y))

#define RB_FOREACH_REVERSE(x, name, head) \
    for ((x) = RB_MAX(name, head); \
         (x) != NULL; \
         (x) = name##_RB_PREV(x))

#define RB_FOREACH_REVERSE_FROM(x, name, y) \
    for ((x) = (y); \
         ((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
         (x) = (y))

#define RB_FOREACH_REVERSE_SAFE(x, name, \
    head, y) \
    for ((x) = RB_MAX(name, head); \
         ((x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
         (x) = (y))

#endif /* _SYS_TREE_H_ */
pkgname=linux-headers
pkgver=6.14.2 # Follow the latest Linux stable
_kernver=${pkgver%.*}
pkgrel=0
pkgdesc="Linux system headers"
url="https://kernel.org/"
arch="all" # Headers are architecture specific
license="GPL-2.0-only"
makedepends="perl"
options="!check !dbg !strip !tracedeps !archcheck"

```

source="https://kernel.org/pub/linux/kernel/v6.x/linux-\$_kernver.tar.xz
revert-broken-uapi.patch
0003-remove-inclusion-of-sysinfo.h-in-kernel.h.patch

1.40 readline 8.2.13-r1

1.40.1 Available under license :

@c The GNU Free Documentation License.
@center Version 1.3, 3 November 2008

@c This file is intended to be included within another document,
@c hence no sectioning command or @node.

@display
Copyright @copyright{} 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
@uref{http://fsf.org/}

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
@end display

@enumerate 0
@item
PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document @dfn{free} in the sense of freedom: to
assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way
to get credit for their work, while not being considered responsible
for modifications made by others.

This License is a kind of ``copyleft'', which means that derivative
works of the document
must themselves be free in the same sense. It
complements the GNU General Public License, which is a copyleft
license designed for free software.

We have designed this License in order to use it for manuals for free
software, because free software needs free documentation: a free
program should come with manuals providing the same freedoms that the
software does. But this License is not limited to software manuals;
it can be used for any textual work, regardless of subject matter or
whether it is published as a printed book. We recommend this License
principally for works whose purpose is instruction or reference.

@item

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The ``Document'', below, refers

to any such manual or work. Any member of the public is a licensee, and is addressed as ``you''. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A ``Modified Version'' of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ``Secondary Section'' is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ``Invariant Sections'' are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The ``Cover Texts'' are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A ``Transparent'' copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for

images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not ``Transparent" is called ``Opaque".

Examples of suitable formats for Transparent copies include plain `@sc{ascii}` without markup, Texinfo input format, `La@TeX{}` input format, `@acronym{SGML}` or `@acronym{XML}` using a publicly available `@acronym{DTD}`, and standard-conforming simple `@acronym{HTML}`, PostScript or `@acronym{PDF}` designed for human modification. Examples of transparent image formats include `@acronym{PNG}`, `@acronym{XCF}` and `@acronym{JPG}`. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, `@acronym{SGML}` or `@acronym{XML}` for which the `@acronym{DTD}` and/or processing tools are not generally available, and the machine-generated `@acronym{HTML}`, PostScript or `@acronym{PDF}` produced by some word processors for output purposes only.

The ``Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, ``Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The ``publisher" means any person or entity that distributes copies of the Document to the public.

A section ``Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as ``Acknowledgements", ``Dedications", ``Endorsements", or ``History".) To ``Preserve the Title" of such a section when you modify the Document means that it remains a section ``Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has

no effect on the meaning of this License.

@item

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

@item

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible.

You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material.

If you use the latter option, you must take reasonably prudent steps,

when you begin distribution of Opaque copies in quantity, to ensure that

this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

@item

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

@enumerate A

@item

Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions

(which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

@item

List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

@item

State on the Title page the name of the publisher of the Modified Version, as the publisher.

@item

Preserve all the copyright notices of the Document.

@item

Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

@item

Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

@item

Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

@item

Include an unaltered copy of this License.

@item

Preserve the section Entitled ``History'', Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled ``History'' in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

@item

Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the ``History'' section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

@item

For any section Entitled ``Acknowledgements'' or ``Dedications'', Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

@item

Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

@item

Delete any section Entitled ``Endorsements''. Such a section may not be included in the Modified Version.

@item

Do not retitle any existing section to be Entitled ``Endorsements" or to conflict in title with any Invariant Section.

@item

Preserve any Warranty Disclaimers.

@end enumerate

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document,
you may at your option designate some or all
of these sections as invariant. To do this, add their titles to the
list of Invariant Sections in the Modified Version's license notice.
These titles must be distinct from any other section titles.

You may add a section Entitled ``Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties---for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of,
you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

@item

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single

copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ``History" in the various original documents, forming one section Entitled ``History"; likewise combine any sections Entitled ``Acknowledgements", and any sections Entitled ``Dedications". You must delete all sections Entitled ``Endorsements."

@item

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

@item

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an ``aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole

aggregate.

@item

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled ``Acknowledgements'', ``Dedications'', or ``History'', the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

@item

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the

licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

@item

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See [{http://www.gnu.org/copyleft/}](http://www.gnu.org/copyleft/).

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

@item

RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is ``eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

@end enumerate

@page

@heading ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

@smallexample

@group

Copyright (C) @var{year} @var{your name}.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled ``GNU Free Documentation License".

@end group

@end smallexample

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the ``with@dots{ }Texts." line with this:

@smallexample

@group

with the Invariant Sections being @var{list their titles}, with the Front-Cover

Texts being @var{list}, and with the Back-Cover Texts being @var{list}.

@end group

@end smallexample

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

@c Local Variables:

@c ispell-local-pdict: "ispell-dict"

@c End:

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty;

and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program

with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is

void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed

through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

EXCEPT WHEN

OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS

TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to

any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1)

assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand

ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to

avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a

menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The

"System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered

by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed;

section 10

makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately

publish on each copy an appropriate copyright notice;

keep intact all notices stating that this License and any

non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided,

in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of

it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The

work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will

be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it

under certain conditions; type `show c` for details.

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<http://www.gnu.org/philosophy/why-not-lgpl.html>>.

1.41 ca-certificates-bundle 20250911-r0

1.41.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):

*

No license file was found, but licenses were detected in source scan.

-----BEGIN CERTIFICATE-----

```
MIIH0zCCBbugAwIBAgIIXsO3pkN/pOAwdQYJKoZIhvcNAQEFBQAwwQIESMBAGA1UE
AwwJQUUNDVlJBBSVoxMRAwDgYDVQQQLDAdQS0lBQ0NWMQ0wCwYDVQQKDARBQ0NWMQsw
CQYDVQQGEwJFUzAeFw0xMTA1MDUwOTMzMzdaFw0zMDEyMzEwOTMzMzdaMEIxEjAQ
BgNVBAMMCUFdQ1ZSQUlaMTEQMA4GA1UECwwHUETJQUUNDVjENMAAsGA1UECgwEQUND
VjELMAkGA1UEBhMCRVMwggLiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCb
qau/YUqXry+XZpp0X9DZlv3P4uRm7x8fRzPCRKPfmt4ftVTdFXxpNRFvu8gMjmoY
HtiP2Ra8EEg2XPBjs5BaXCQ316PWYwIxfEBcoSwfdtNgM3802/J+Nq2DoLSRYWo
G2ioPej0RGy9ocLLA76MPhMAhN9KSMDjIgro6TenGEyxCCQ0jVn8ETdkXhBilyNpA
lHPrzg5XPAOBOP0KoVdDaaxXbXmQeOW1tDvYvEyNKKGno6e6Ak4l0Squ7a4DIrhr
IA8wKFSVf+DuzgpmndFALW4ir50awQUZ0m/A8p/4e7MCQvtQqR0tkw8jq8bBD5L/
0KIV9VMJcRz/RROE5iZe+OCiHAr8Fraocwa48GOEAqDGWuzndN9wrqODJerWx5eH
k6fGioozl2A3ED6XPm4pFdahD9GILBKfb6qkxkLrQaLjIUPtAYVtjrs78yM2x/47
4KElB0iryYl0/wiPgL/AlmXz7uxLaL2diMMxs0Dx6M/2OLuc5NF/1OVYm3z61PMO
m3WR5LpSLhl+0fXNWhn8ugb2+1KoS5kE3fj5tItQo05iifCHJPqDQsGH+tUtKSpa
cXpkatcnYGMN285J9Y0fkIkyF/hzQ7jSWpOGYdbhdQrqeWZ2iE9x6wQl1gpaepPl
uUsXQA+xtrn13k/c4LOsOxFwYIRKQ26ZIMApCqRAZQIDAQABo4ICyzCCAscwfQYI
```

KwYBBQUHAQEEcTBvMEwGCCsGAQUFBzAChkBodHRwOi8vd3d3LmFjY3YyZXMvZmls
ZWfkbWluL0FyY2hpdm9zL2NlcnRpZmljYWRvcy9yYWI6YWNjdjEuY3J0MB8GCCsG
AQUFBzABhhNodHRwOi8vb2NzcC5hY2N2LmVzMB0GA1UdDgQWBBSht7Tj3zcnk1X2
VuqB5TbMjB4/vTAPBgNVHRMBAf8EBTADAQH/MB8GA1UdIwQYMBaAFNKHtOPfNyeT
VfZW6oHlNsyMHj+9MIIBcwYDVR0gBIIBajCCAWYwggFiBgRVHSAAMIIBWDCCASIG
CCsGAQUFBwICMIIBFB6CARAAQQB1AHQAAbwByAGkAZABhAGQAIABkAGUAIABDAGUA
cgB0AGkAZgBpAGMAYQBjAGkA8wBuACAAUgBhAO0AegAgAGQAZQAgAgwAYQAgAEEA
QwBDAFYAIAAoAEEAZwBLAG4AYwBpAGEAIAABkAGUAIABUAGUAYwBuAG8AbABvAGcA
7QBhACAAeQAgAEMAZQByAHQAaQBmAGkAYwBhAGMAaQDzAG4AIABFAGwAZQBjAHQA
cgDzAG4AaQBjAGEALAAgAEMASQBGACAAUQA0ADYAMAAxADEANQA2AEUAKQAuACAA
QwBQAFMAIABIAG4AIABoAHQAdABwADoALwAvAHcAdwB3AC4AYQBjAGMAdgAuAGUA
czAwBggrBgEFBQcCARYkaHR0cDovL3d3dy5hY2N2LmVzL2xlZ2lzbGFjaW9uX2Mu
aHRtMFUGA1UdHwROMEwwSqBIOEaGRGh0dHA6Ly93d3cuYWNjdj5lcY9maWxlYWRT
aW4vQXJjaGl2b3MvY2VydGlmaWNhZG9zL3JhaXphY2N2MV9kZXIuY3JsMA4GA1Ud
DwEB/wQEAwIBBjAXBgNVHREEDAOgQxhY2N2QGFjY3YyZXMwDQYJKoZIhvcNAQEF
BQADggIBAxcAp/n/UNnSEQU5CmH7UwoZtCPNdpNYbdKI02125DgBS4OxnnQ8pdp
D70ER9m+27Up2pvZrqmZ1dM8MJP1jaGo/AaNRPTKFpV8M9xii6g3+CfYCS0b78gU
JyCpZET/LtZ1qmxNYEASUNUY9rizLpm5U9EelvZaoErQNV/+QEnWCzI7UiRfD+m
AM/EKXMRNt6GGT6d7hmKG9Ww7Y49nCrADdg9ZuM8Db3VIFzi4qc1GwQA9j9ajepD
vV+JHanBsMyZ4k0ACtrJJ1vnE5Bc5PUzolVt3OAJTS+xJlsndQAjxGJ3KQhfnlms
tn6tn1QwIgpBHnFk/vk4CpYY3QIUrCPLBhwepH2NDd4nQeit2hW3sCPdK6jT2iWH
7ehVRE2I9DZ+hJp4rPcOVkkO1jMl1oRQQmwgEh0q1b688nCBpHBgvgW1m54ERL5h
I6zppSSMEYCUWqKiuUnSwdzRp+0xESyeGabu4VXhwOrPDYTkF7eifKXeVSUG7szA
h1xA2syVP1XgNce4hL60Xc16gwFy7ofmXx2utYXGJt/mwZrpHgJHnyqobalbz+xF
d3+YJ5oyXSrjhO7FmGYvliAd3djDJ9ew+f7Zfc3Qn48LFFhRny+Lwzgt3uiP1o2H
pPVWQxaZLPSkVrQ0uGE3ycJYgBugl6H8WY3pEfbRD0tVNEYqi4Y7

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFgzCCA2ugAwIBAgIPXZONMGc2yAYdGsdUhGkHMA0GCSqGSIb3DQEBCwUAMDsx
CzAJBgNVBAYTAkVMTREwDwYDVQQKDAhGtK1ULVJDTTEZMBcGA1UECwwQQUMgUkFJ
WiBGtK1ULVJDTTAeFw0wODEwMjIxNTU5NTZaFw0zMdAxMDEwMDAwMDBaMDsxZAJ
BgNVBAYTAkVMTREwDwYDVQQKDAhGtK1ULVJDTTEZMBcGA1UECwwQQUMgUkFJWiBG
Tk1ULVJDTTCCAILwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBALpxgHpMhm5/
yBNtwMZ9HACXjyWMI7sQmkCpGreHiPibVmr75nuOi5KOpyVdWRHbNi63URcfqQgf
BBckWKo3Shjf5TnUV/3XwSyRAZHiItQDwFj8d0fsjz50Q7qsNI1NOHZnjdIbzAz
WHFctPVrbtQBULgTfmXKo0nRIBnuvMApGGWn3v7v3QqQIecaZ5JCEJhfTzC8PhxF
tBDXaEAUwED653cXeuYLj2VbPNmaUtu1vZ5Gzz3rkQUCwJaydkxNEJY7kvqcfw+Z
374jNUUeAlz+taibmSXaXvMiwzn15Cou08YfxGyqxRxqAQVKL9LFwag0Jl1mpdIC
IfkYtwb1TplvqKtMUejPUBjFd8g5CSxJkjKZqLsXF3mwWsXmo8RZZUc1g16p6DUL
mbvkzSDGm0oGOBVo/CK67lWMK07q87Hj/LaZmtVC+nFNcm+HHmpxeffnTtOmlcYF7
wk5HlqX2doWjKI/pgG6BU6VtX7hI+cL5NqYuSf+4lsKMB7ObiFj86xsc3i1w4peS
MKGJ47xVqCfWS+2QrYv6YyVZLag13cqXM7zlcZced0evzXg5KkAYmY6252TuTb7p2
ZSysV4999AeU14ECll2jB0nVetBX+RvnU0Z1qrB5QstocQjpYL05ac70r8NWQMet
UqIJ5G+GR4of6ygnXYMgrwTJbFaai0b1AgMBAAAgjgYmWgYAwDwYDVR0TAQH/BAUw
AwEB/zAOBgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFpD9xf3E6Jobd2Sn9R2gzL+H
YJptMD4GA1UdIAQ3MDUwMwYEVROgADArMCKGCCsGAQUFBwIBFh1odHRwOi8vd3d3

W7i1o0TJrH93PB0j7IKppuLIBkwC/qxcmZkLLxCKpvR/1Yd0DVIJRfbwCvW5Kda/
SiOL9V8BY9KHcyi1Swr1+KuCLH5zJtIdC2MKF4EA/7Z2Xue0sUDKlBvVgFHlSFJn
LNJhiQcND85Cd8BEc5xEUKDbEAotlRyBr+Qc5RQe8TZBAQIvfXOn3kLMTOMJDVb3
n5HUA8ZsyY/b2BzgQJhdZpmYgG4t/wHFzstGH6wCkxPmrqKEPMVOHj1tyRRM4y5B
u8o5vzY8KhmqQYdOpc5LMnndkEl/AgMBAAGjYzBhMB8GA1UdIwQYMBaAFJxf0Gxj
o1+TypOYCK2Mh6UsXME3MB0GA1UdDgQWBBSX9BsY6Nfk8qTmAitJleLFzBNzAO
BgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOC
AgEATH65isagmD9uw2nAalxJUqzLK114OMHVVISfk/CHGT0sZonrDUL8zPB1hT+L
9IBdeeUXZ701guLyPI59WzbLWoAAKfLOKyzxj6ptBZNscsdW699QIyJlRRA96Gej
rw5VD5AJYU9LWaL2U/HANeQvwSS9eS9OICI7/RogsKQOLHDtdD+4E5UGUcjohybK
pFtqFiGS3XNgnhAY3jyB6ugYw3yJ8otQPr0R4hUDqDZ9MwFsSBXXiJCZBMXM5gfo
vPSQ7RPI6ovDj6MzD8EpTBNO2hVWcXNygID2mjN8orGoGjR0ZVzO0eurU+AagNjq
OknkJjCb5RyKqKkVMoaZkgoQI1YS4PbOTOK7vtuNknMBZi9iPrJyJ0U27U1W45eZ
/zo1PqVUSIJZS2Db7v54EX9K3BR5YLZrZAPbFYPhor72I5dQ8AkzNqdxliXzuUJ9
2zg/LFis6ELhDjtTO0wugumDLmsx2d1Hhk9tl5EuT+IocTUW0fJz/iUrB0ckYyfl
+PbZa/wSMVYIwFNCr5zQM378BvAxRAMU8Vjq8moNqRGyg77FG8H6lnc04g175x2
MjxNBiLOFeXdntiP2t7SxDnlF4HPOEfrf4htWRvfn0IUr7PqLBmZdo3r5+qPeoo
tt7VMVgWglvquxl1AnMaykgaIZOQCo6ThKd9OyMYkomgjaw=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIIFuzCCA6OgAwIBAgIIVwoRI0LE48wwDQYJKoZIhvcNAQELBQAwwELMAkGA1UE
BhMCSVQxDjAMBgNVBACMBU1pbGFuMSMwIQYDVQQKDBpBY3RhbGlzIFMucC5BLi8w
MzM1ODUyMDk2NzEnMCUGA1UEAwweQWN0YWxpcyBBdXR0ZW50aWNhdGlvbiBSb290
IENBMB4XDTEyMDk2NzEnMCUGA1UEAwweQWN0YWxpcyBBdXR0ZW50aWNhdGlvbiBSb290IENB
SVQxDjAMBgNVBACMBU1pbGFuMSMwIQYDVQQKDBpBY3RhbGlzIFMucC5BLi8wMzM1
ODUyMDk2NzEnMCUGA1UEAwweQWN0YWxpcyBBdXR0ZW50aWNhdGlvbiBSb290IENB
MIICljANBgkqhkiG9w0BAQEFAAOCAg8AMIICcGKCAgEAp8bEpSmkLO/lGMWwUKNv
UTufClrJwkg4CsIcoBh/kbWHuUA/3R1oHwiD1S0eiKD4j1aPbZkCkpAW1V8IbInX
4ay8IMKx4INRimlNAJZaby/ARH6jDuSRzVju3PvHHkVH3Se5CAGfpiEd9UEtL0z9
KK3giq0itFZljoZUj5NDKd45RnijMCO6zfB9E1fAXdKDa0hMxKufgFpbOr3JpyI/
gCczWw63igxdBzclY2zSekciRDXFzMwuJt0q7bd9Zg1fYVEiVRvjRuPjPdA1Yprb
rxTIW6HMiRvhMCb8oJsfgadHHwTrozmSBp+Z07/T6k9QnBn+locePGX2oxgkg4YQ
51Q+qDp2JE+BicXjDwL4k5RHILv+1A7TaLndxHqEguNTVHnd25zS8gebLra8Pu2F
be8IEfKXGkJh90qX6IuxEAf6ZYGYojnP9zz/GPvG8VqLWeICrHuS0E4UT11F9gxe
KF+w6D9Fz8+vm2/7hNN3WpVvrJSEnu68wEqPSpP4RCHiMUvhUE4Q2OM1fEwZtN4F
v6MGn8i1zeQf1xcGDXqVdFUNaBr8EBtiZJ1t4JWgw5QHvW0U5r0F+7if5t+L4sbn
fPb2U8WANFAoWPASUHEXMLrmeGO89LKtmuyy/uE5jF66CyCU3nuDuP/jVo23Eek7
jPKxwV2dpAtMK9myGPW1n0sCAwEAAANjMGEwHQYDVR0OBBYEFFLYiDrIn3hm7Ynz
ezhw1MkCAjbQMA8GA1UdEwEB/wQFMAMBAf8wHwYDVR0jBBgwFoAUUtiOSifeGbt
ifN7OHCUyQICNtAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4ICAQAL
e3KHwGCmSUyIWOYdiPcUZEim2FgKDk8Tnd81HdTtBjHlGt5q1d07GjLukD0R0i70
jsNjLiNmsGe+b7bAEzlgqqI0JZN1Ut6nna0Oh4lScWoWPBkdg/iaKWW+9D+a2fDz
WochcYBNy+A4mz+7+uAwTc+G02UQGRjRlwKxK3JCaKygvU5a2hi/a5iB0P2avl4V
SM0RFbnAKVy06Ij3Pjaut2L9HmLecHgQHEhb2rykOLpn7VU+Xlff1ANATIGk0k9j
pwlCCRT8AKnCGHNPLsBA2RF7SOp6AsDT6ygBJlh0wcBzIm2TlF05fbsq4/aC4yyX
X04fkZT6/iyj2HYauE2yOE+b+h1IYHkm4vP9qdCa6HCPSXrW5b0KDtst842/6+Ok

CERTIFICATE-----

CERTIFICATE-----

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIFRjCCAy6gAwIbYwURrGmCu4wDQYJKoZlIhvcNAQEMBAQAwQTELMAkGA1UE
 BhMCVVMxFDASBgNVBAoMCOFmZmlybVRydXN0MRwwGgYDVQQDDDBNBZmZpcmlUcnVz
 dCBQcmVtaXVtMB4XDTEwMDEyOTE0MTAzNloXDTEwMDEyOTE0MTAzNlowQTELMAkG
 A1UEBhMCVVMxFDASBgNVBAoMCOFmZmlybVRydXN0MRwwGgYDVQQDDDBNBZmZpcmlU
 cnVzdCBQcmVtaXVtMIICljANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAGeAxBLf
 qV/+Qd3d9Z+K4/as4Tx4mrzY8H96oDMq3I0gW64tb+eT2TZwamjPjlGjhVtnBKAQ
 JG9dKILB1fYSCkTtuG+kU3fhQxTGJoeJKJPj/CihQvL9Cl/0qRY7iZNYaqoe5rZ
 +jjeRfCv5fiMyNII4g0WJx0eyIOFJbe6qlVBzAMiSy2RjYvmia9mx+n/K+k8rNrS
 s8PhaJyJ+HoAVt70VZVs+7pk3WKL3wt3MutizCaam7uqYoNMtAZ6MMgpv+0GTZe5
 HMQxK9VfvFMSF5yZVylmd2EhMQcuJUmdGPLu8ytxjLW6OQdJd/zvLpKQBY0tL3d7
 70O/Nbua2Plzpyzy0FfuKE4mX4+QaAkvuPjcBukumj5Rp9EixAqnOEhss/n/fauG
 V+O61oV4d7pD6kh/9ti+I20ev9E2bFhc8e6kGVQa9QPSdubhjL08s9NIS+LI+H+S
 qHZGnEJlPqQewQcDWkYtuJfzt9WyVSHvutxMAJf7FJUnM7/oQ0dG0giZFmA7mn7S
 5u046uwBHjxIVkkJx0w3AJ6IDsBz4W9m6XJHMD4Q5QsDyZpCAGzFIH5hxIrf4Ia
 C1nEWJTJ3s7xgaVY5/bQGEyzWZDbZvUjthB9+pSKPKrhC9IK31FOQeE4tGv2Bb0TX
 OwF0lkLgAOIua+rF7nKsu7/+6qqo+Nz2snmKtmcCAwEAAaNCMEAwHQYDVR0OBBYE
 FJ3AZ6YMItkm9UWrpmVSESfYRaxjMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/
 BAQDAgEGMA0GCSqGSIb3DQEBDAUAA4ICAQCzV00QYk465KzquByvMiPIs0laUZx2
 KI15qldGF9X1Uva3ROgIRL8YhNILgM3FEv0AVQVhh0HctSSePMTYyPtwni94loMg
 Nt58D2kTiKV1NpgIpsbfrM7jWNa3Pt668+s0QNiigfV4Py/VpfzZotReBA4Xrf5B
 8OWycvpEgjNC6C1Y91aMYj+6QrCcDFx+LmUmXFNPALJ4fqENmS2NuB2OosSw/WDQ
 MKSOyARiqcTtNd56l+0OOF6SL5Nwpamcb6d9Ex1+xghIsV5n61EIJenmJWtSKZGc
 0jlzCFfemQa0W50QBuHCAki4HEoCChtQwUHK+4w1IX2COPKpVJEZNZOUbWo6xbLQ
 u4mGk+ibyQ86p3q4ofB4Rvr8Ny/lioTz3/4E2aFooC8k4gmVBtWVyuEklut89pMF
 u+1z6S3RdTnX5yTb2E5fQ4+e0BQ5v1VwSJlXmBSc7kqYA5YwH2AG7hsj/oFgIxpH
 YoWlzBk0gG+zrBrjn/B7SK3VAdlntqlyk+otZrWyuOQ9PLLvTlZq6we/qzWaVYa8
 GKa1qF60g2xraUDTn9zxw2lrueFtCfTxqlB2Cnp9ehevVZZCmTEJ3WARjQUwfuaO
 RtGdFNrHF+QFlozEJLUBzxQHskD4o55BhrwE0GuWyCqANP2/7waj3VjFhT0+j/6e
 KeC2uAloGRwYQw==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MlIB/jCCAYWgAwIBAgIIdJclsc/elQwCgYIKoZlZj0EAWMwRTELMakGA1UEBhMC
VVMxFDASBgNVBAoMC0FmZmlybVRydXN0MSAwHgYDVQQDDBdBZmZpcm1UcnVzdCBQ
cmVtaXVtIEVDQzAeFw0xMDAxMjkxNDIwMjRaFw00MDEyMzExNDIwMjRaMEUxCzAJ
BgNVBAYTAiVTMRQwEgYDVQQKDAkBZmZpcm1UcnVzdDEgMB4GA1UEAwwXQWZmaXJt
VHJ1c3QgUHJlbWl1bSBFQ0MwdjAQBgqhkJOPQIBBgUrgQQAIGNiAAQNMF4bFZOD
0KF5Nbc6PJJ6yhUczWLznCZcBz3lVPqj1swS6vQUX+iOGasvLkjmrbhDeKzQN8O9
ss0s5kfiGuZjuD0uL3jET9v0D6RoTFVy5UdThhClXjMNzyR4ptlKymjQjBAMB0G
A1UdDgQWBBSarylwBE1NSZRMADDav5A1a7WPDAPBgNVHRMBAf8EBTADAQH/MA4G
A1UdDwEB/wQEAWIBBJAKBggqhkJOPQQDAwNnADBKAjAXCfOHIFBar8jAQR9HX/Vs
aobgxCd05DhT1wV/GzTxji+zygk8N53X57hG8f2h4nECMEJZh0PUUd+60wkvyWs6I

-----END CERTIFICATE-----

MIDQTCcAimgAwAIBAgITBmyfz5m/jAo54vB4ikPmljZbyjANBgkqhkiG9w0BAQSF
 ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWFi
 b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAwMFowOTEL
 MAkGA1UEBhMCVVMxZDZANBgNVBAoTBkFtYXpjb2ZEMBcGA1UEAxMQQW1hem9uIFJv
 b3QgQ0EgMTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4gHHKEnXj
 ca9HgFB0fW7Y14h29Jlo91ghYPl0hAEvrAlthrOgQ3pOsqTQNroBvo3bSMgHFzZM
 9O6II8c+6zf1tRn4SWiw3te5djdYZ6k/oI2peVKVuRF4fn9tBb6dNqcmzU5L/qw
 IFAGbHrQgLK+m+/sRxpPUDgH3KKHOVj4utWp+UhnMJbulHheb4mjUcAwhmahRWa6
 VOujw5H5SNz/0egwLX0tdHA114gk957EWW67c4cX8jJGKLhD+rcdqsq08p8kDi1L
 93FcXmn/6pUCyziKrlA4b9v7LWlBxcceVOF34GfID5yHI9Y/QCB/IIDEgEw+OyQm
 jgSubJrIqg0CAwEAAaNCMEAwdwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
 AYYwHQYDVR0OBBYEFiQYzIU07LwMlJQuCFmxc7IQTgoIMA0GCSqGSIB3DQEBcWUA
 A4IBAQCYY8jdaQZChGsV2USggNiMOruYou6r4lK5IpDB/G/wkjUu0yKGX9rbxenDI
 U5PMCCjjmCXPI6T53iHTfIUJrU6adTrCC2qJeHZERxhlbI1Bjjt/msv0tadQ1wUs
 N+gDS63pYaACbvXy8MWy7Vu33PqUXHeeE6V/Uq2V8viTO96LXFvKWlJbYK8U90vv
 o/ufQJVtMVT8QtPHR8hjrkdPSHca2XV4cdFyQzR1bldZwgJcJmApzyMZFo6IQ6XU
 5Msl+yMRQ+hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy
 rqXRfboQnoZsG4q5WTP468SQvvG5

CERTIFICATE-----

MIIFQTCCAymsgAwIbAGITBmyf0pY1hp8KD+WGePhbJruKNzANBgkqhkiG9w0BAQwF
ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6
b24gUm9vdCBDQSAyMB4XDTE1MDUyNjAwMDAwMFoXDTQwMDUyNjAwMDAwMFowOTEL
MAkGA1UEBhMCVVMxZDZANBgNVBAoTBkFtYXpjbjEzMBCGA1UEAxMQQW1hem9uIFJv
b3QgQ0EgMjCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAK2Wny2cSkxK
gXlRmeyKy2tgURO8TW0G/LAljd0ZEGrHJgw12MBvIITpILGbhQPDW9tK6Mj4kHbZ
W0/jTOGnK3Mmqw9DJArktQGGWCsN0R5hYGCrVo34A3MnaZMUnbqQ523BNFQ9lXg
1dKmSYXpN+nKfq5clU1Imj+uIFptiJXZNlHSGkOQsL9sBbm2eLfqq0OQ6PBJTYv9K
8nu+nQWpEjTj82R0Yiw9AElaKP4yRLuH3WUnAnE72kr3H9rN9yFVke8P7K6C4Z9r
2UXTu/Bfh+08LDmG2j/e7HJV63mjrdvdfLC6HM783k81ds8P+HgfajZRRidhW+me
z/CiVX18JYpvL7TFz4QuK/0NURBs+18bvBt+xa47mAExkv8LV/SasrlX6avvDXbR
8O70zoan4G7ptGmh32n2M8ZpLpcTnqWHsFcQgTfJU7O7f/aS0ZzQGPsSbtqDT6Zj
mUyl+17vIWR6IF9sZIUUVyzfpYgwLKhbcAS4y2j5L9Z469hdAlO+ekQiG+r5jqFoz
7Mt0Q5X5bGLSNscpb/xVA1wf+5+9R+vnSUEVC06JlgIJ4PVhHvG/LopyboBZ/1c6
+XUyo05f7O0oYtNc/LMgRdg7c3r3NunysV+Ar3yVAhU/bQtCSwXVEqY0VThUWcl
0u1ufm8/0i2BWSlmy5A5lREedCf+3euvAgMBAAGjQjBAMA8GA1UdEwEB/wQFMAMB
Af8wDgYDVR0PAQH/BAQDAgGGMB0GA1UdDgQWBBSwDPBMMPQFWAJI/TPIUq9LhONm
UjANBgkqhkiG9w0BAQwFAAOCAGEAqqiAjw54o+Ci1M3m9Zh6O+oAA7CXDP08Wqj2
LIxyh6mx/H9z/WNxeKWHWc8w4Q0QshNabYL1auaAn6AFC2jkR2vHat+2/XcycuUY
+gn0oJMsXdKMdYV2ZZAMA3m3MSNjrXiDCYZohMr/+c8mmpJ5581LxedhpxfL86kS
k5Nrp+gvU5LEYFiwzAJRGFuFjWJZY7attN6a+yb3ACfAXVU3dJnJUH/jWS5E4ywl
7uxMMne0nxrps10gxdR9HicWxkPo1LsmmkVwXqkLN1PiRnsn/eBG8om3zEK2yygm

btmlyTrIQRNg91CMFa6ybRoVGld45pIq2WWQgj9sAq+uEjonljYE1x2igGOpm/Hl
urR8FLBOybEfF849lHqm/osohHUqS0nGkWxr7JOcQ3AWEbWaQbLU8uz/mtBzUF+
fUwPfhJ5elnNXkoOrJupmHN5fLT0zLm4BwydFy4x2+IoZCn9Kr5v2c69BoVYh63
n749sSmvZ6ES8lgQGVMDMBu4Gon2nL2XA46jCfMdiyHxtN/kHNGfZQIG6lzWE7OE
76KlXlX3KadowGuuQNKotOrN8I1LOJwZmhsoVLIJkO/KdYE+HvJkJMcYr07/R54H
9jVlpNMKVv/1F2Rs76giJUmtt8AF9pYfl3uxRuw0dFfIRDH+fO6AgonB8Xx1sfT
4PsJYGw=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIBtjCCAVugAwIBAgITBmyf1XSXNmY/Owua2eiedgPySjAKBggqhkJOPQQDAjA5
MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6b24g
Um9vdCBDQSAzMB4XDTE1MDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAwMFowOTELMAkG
A1UEBhMCVVMxMzE0MjE1MDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAwMFowOTELMAkG
A1UEBhMCVVMxMzE0MjE1MDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAwMFowOTELMAkG
Q0EgMzBZMBMGBYqGSM49AgEGCCqGSM49AwEHA0IABCMxp8ZBf8ANm+gBG1bG8IKl
ui2yEujSLtf6ycXYqm0fc4E7O5hrOXwzpcVOho6AF2hiRVd9RFgdszflZwjZt6j
QjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAAQH/BAQDAgGGMBOGA1UdDgQWBBSr
ttvXBp43rDCGB5Fwx5zEGbF4wDAKBggqhkJOPQQDAgNJADBGAiEA4IWSoxe3jflkr
BqWTrBqYaGFy+uGh0PsceGCMQ5nFuMQCIQCcAu/xlJyzlvnrxir4tiz+OpAUfTeM
YyRIHN8wfdVoOw==

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIB8jCCAXigAwIBAgITBmyf18G7EEwpQ+Vxe3ssyBrBDjAKBggqhkJOPQQDAzA5
MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6b24g
Um9vdCBDQSA0MB4XDTE1MDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAwMFowOTELMAkG
A1UEBhMCVVMxMzE0MjE1MDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAwMFowOTELMAkG
Q0EgNDB2MBAGByqGSM49AgEGBSuBBAAiA2IABNKrijdPo1MN/sGKe0uoe0ZLY7Bi
9i0b2whxIdIA6GO9mif78DluXeo9pcmBqqNbIJhFXRbb/egQbeOc4OO9X4Ri83Bk
M6DLJC9wuoihKqB1+IGuYgbEgds5bimwHvouXKNCMEAwDwYDVR0TAQH/BAUwAwEB
/zAOBgNVHQ8BAf8EBAMCAYYwHQYDVR0OBBYEFNPsxzplbszh2naaVvuc84ZtV+WB
MAoGCCqGSM49BAMDA2gAMGUCMDqLIgG9fhGt0O9Yli/W651+kI0rz2ZVwyzjKKlw
CkcO8DdZE8tmZQoTipPNU0zWgIxAOp1AE47xDqUEpHJWEadIRNyp4iciuRMStuW
1KyLa2tJElMzrdfkviT8tQp21KW8EA==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDdzCCA1+gAwIBAgIIXDPLYixfszIwDQYJKoZIhvcNAQELBQAwPDEeMBwGA1UE
AwwVQXRvcyBUcnVzdGVkUm9vdCAyMDEwCwYDVQQKDARBdG9zMQswCQYDVQQG
EwJERTAEFw0xMTA3MDcxNDU4MzBaFw0zMDUyMzE0MjE1MDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAwMFowOTELMAkG
A1UEBhMCVVMxMzE0MjE1MDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAwMFowOTELMAkG
A1UEBhMCVVMxMzE0MjE1MDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAwMFowOTELMAkG
REUwggEiMA0GCSqGSIb3DQEBAAQUAA4IBDwAwggEKAoIBAQCvHtuXbyo7LjvPpvMp
Nb7PGKw+qtn4TaA+Gke5vJrf8v7MPkfoepbCJI419KkM/IL9bcFyYie96mvr54rM
VD6QUM+A1JX76LWC1BTftqlVJVfbsVD2sGBkWXppzwO3bw2+yj5vdHLqqjAqc2K+
SZFhyBH+DgMq92og3AIVDV4VavzjgsG1xZ1kCWyjWZgHJ8cblithdHFsQ/H3NYkQ
4J7sVaE3IqKHBAUsR320HLliKWYoyrfhk/WklAOZuXCFteZI6o1Q/NnezG8HDt0L

cp2AMBYHIT8oDv3FdU9T1nSatCQujgKRz3bFmx5VdJx4IbHwLfELn8LVlhgf8FQi
eowHAgMBAAGjfTB7MB0GA1UdDgQWBBSnpQaxLKYJYO7Rl+lwrrw7GWzbITAPBgNV
HRMBAf8EBTADAQH/MB8GA1UdIwQYMBaAFKELBrEspglg7tGX6XCuvDsZbNshMBgG
A1UdIAQRMA8wDQYLKwYBBAGwLQMEAQEwDgYDVR0PAQH/BAQDAgGGMA0GCSqGS1b3
DQEBcWUAA4IBAQAmdzTbIEiGKkGdLD4GkGDEjKwLVlgfuXvTBznk+j57sj1O7Z8j
vZfza1zv7v1Apt+hk6EKhqzvINB5Ab149xnYJDE0BAGmuhWawyfc2E8PzBhj/5kP
DpFrdRbhIfzYJsdHt6bPWHJxfrhTZVHO8mvbaG0weyJ9rQPOLXiZNwzl6bb65pc
maHFCN795trV1lpFDMS3wrUU77QR/w4VtfX128a961qn8FYiqTxlVMYVqL2Gns2D
lmh6cYgJ4Qvh6hEbaAjMaZ7snkGeRDImeuKHCnE96+RapNLbxc3G3mB/ufNPRJLv
KrcYPqcZ2Qt9sTdBQrC6YB3y/gkRsPCHe6ed

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICFTCCA ZugAwIBAgIQPZg7pmY9kGP3fiZXOATvADAKBggqhkJOPQQDAzBMMS4w
LAYDVQDDCVBdG9zIFRydXN0ZWRSb290IFJvb3QgQ0EgRUNDIFRMUyAyMDIxMQ0w
CwYDVQQKDARBdG9zMQswCQYDVQQGEwJERTAeFw0yMTA0MjIwOTI2MjNaFw00MTA0
MTcwOTI2MjJaMEwxLjAsBgNVBAMMJUF0b3MgVHJ1c3RlZFJvb3QgUm9vdCBDQSBF
Q0MgVExTIDIwMjExDTALBgNVBAoMBEF0b3MxCzAJBgNVBAYTAkRFMHYwEAYHKoZI
zj0CAQYFK4EEACIDYgAEloZYKDcKZ9Cg3iQZGGeHkBQcfl+3oZIK59sRxUM6KDP/X
tXa7oWyTbIOiaG6l2b4siJBzV3dscqDY4PMwL502eCdpO5KTlbgmClBk1IQ1SQ4
AjJn8ZQSB+/Xxd4u/RmAo0IwQDAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBRR2
KCXWfeBmmnoJsmo7jjPXNtNPojAOBgNVHQ8BAf8EBAMCAYYwCgYIKoZIzj0EAwMD
aAAwZQIwW5kp85wxtolrbNa9d+F851F+uDrNozZffPc8dz7kUK2o59JZDCaOMDtu
CCrCp1rIAjEAMEMM56PDr9NJLkaCI2ZdyQAUEv049OGYa3cpetskz2VAv9LcjBHo
9H1/IIspQuQo

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFZDCCA0ygAwIBAgIQU9XP5hmTC/srBRLYwqipDANBgkqhkiG9w0BAQwFADBM
MS4wLAYDVQDDCVBdG9zIFRydXN0ZWRSb290IFJvb3QgQ0EgUINBIFRMUyAyMDIx
MQ0wCwYDVQQKDARBdG9zMQswCQYDVQQGEwJERTAeFw0yMTA0MjIwOTIxMTBaFw00
MTA0MTcwOTIxMDIAMEwxLjAsBgNVBAMMJUF0b3MgVHJ1c3RlZFJvb3QgUm9vdCBD
QSBSU0EgVExTIDIwMjExDTALBgNVBAoMBEF0b3MxCzAJBgNVBAYTAkRFMIICjAN
BgkqhkiG9w0BAQEFAAOCAg8AMIICGcKCAgEAtoAOxHm9BYx9sKODTSJNy/BBl01Z
4NH+VoyX8te9j2y3I49f1cTYQcvyAh5x5en2XssIKl4w8i1mx4QbZFc4nXUtVsYv
Ye+W/CBGvevUez8/fEc4BKkbbqLlfeZfTFRVOvV98r61jx3ncCHvVoOX3W3WsgFWZ
kmGbzSoXfduP9LVq6hdKZChmFSlsAvFr1bqjM9xaZ6cF4r9lthawEO3NUDPJcFDs
GY6wx/J0W2tExn2WuZgIWWbeKQGb9Cpt0xU6kGpn8bRrZtkh68rZYnxGEFzedUln
nkL5/nWpo63/dgpnQOPF943HhZpZnmKaaU1Fh5hnstVKPNe0OwANwI8f4UDErmwh
3El+fsqjyW22v5MvoVw+j8rtgI5Y4dtXz4U2OLJxpAmMkokIIEjxQGMYSluMWuPD
0xeqqxmjlBvk1cbiZnrXghmmOxYsL3GHX0WelXOTwkKBIROW1527k2gV+p2kHYzy
geBYBr3JtuP2iV2J+axEoctr+hbx1A9JNr3w+SH1VbxT5Aw+kUJWdo0zuATHAR8
ANSbhqRAvNncTFd+rrcztl524WWLZt+NyteYr842mlycg5kDcPOvdO3GDjbnvezB
c6eUWsuSZIKmAMFwoW4sKeFYV+xafJlrJaSQOoD0IJ2azsct+bJLKZWD6TWNp0II
pw9MGZHQ9b8Q4HECAwEAaNCMEAwDwYDVR0TAAQH/BAUwAwEB/zAdBgNVHQ4EFgQU
dEmZ0f+0emhFdcN+tNzMzjkz2ggwDgYDVR0PAQH/BAQDAgGGMA0GCSqGS1b3DQEB

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFdDCCA1ygAwIBAgIQVW9l47TZkGobCdFsPsBsIDANBgkqhkiG9w0BAQsFADBU
MQswCQYDVQQGEwJDTjEmMCQGA1UECgwdQkVJSklORyBDRVJUSUZJQ0FURSBBVVRI
T1JJVFkxHTAbBgNVBAMMFEJKQ0EgR2xvYmFsIFJvb3QgQ0ExMB4XDTE5MTIxOTAz
MTYxN1oXDTQ0MTIxMjAzMTYxN1owVDELMakGA1UEBhMCQ04xJjAkBgNVBAoMHUJF
SUpJTkcqQ0VSVElGSUNBVEUgQVVUSE9SSVRZMR0wGwYDVQQDDDBRCSknBIEdsb2Jh
bCBSb290IENBMTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAPFmCL3Z
xRVhy4QEQaVpN3cdwbB7+sN3SJATcmTRuHyQNZ0YeYjJlW8R4HyDqKYDZ4/N+AZ
spDyRhySsTphzvq3Rp4Dhtczbu33RYx2N95ulpH3134rhxfVizXuhJFyV9xgw8O5
58dnJCNPYwpj9mZ9S1WnP3hkSWkSl+BMDdMJJoDIwOvqfwPKcxRIqLhy1BDPapDgR
at7GGPZHOiJBhyL8xIkoVNiMpTAK+BcWyqw3/XmnkRd4OJmtWO2y3syJfQOcs4ll
5+M7sSKGjwZteAf9krJ/sGsciQ35uMt0WwfCyPQ10WRjeulumijWML3mG90Vr4Tq
nMfK9Q7q8l0ph49pczm+LiRvRSGsxdRpJQaDrXplhRMsDQa4bHlW/KNnMoH1V6XK
V0Jp6VwkYe/iMBhORJhVb3rCk9gZtt58R4oRTklH2yiUAguUSiz5EtBP6DF+bHq/
pj+bOT0CFqMYs2esWz8sgytnOYFcuX6U1WTdno9uruh8W7TXakdl136z1C2OVnZO
z2nxbkRs1CTqjSShGL+9V/6pmTW12xB3uD1IutbB5/EjPtffhZ0nPNRAvQoMvfXn
jSXWgXSHRtQpdaJCbPdied9v3pKH9MiyRVVz99vfFXQpIsHETdfg6YmV6YBW37+
WGgHqel62bno/1Afq8K0wM7o6v0PvY1NuLxxAgMBAAGjQjBAMB0GA1UdDgQWBbTF
7+3M2I0hXkj49cULqcWk+WYATAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQE
AwIBBjANBgkqhkiG9w0BAQsFAAOCAgEAOuKsITQfI/Ki2Pm4rzc2IInRNwPWaZ+4
YRC6ojGYWUfo0Q0lHhVBDOAqVdVXUsv45MdpoX1NcQJeXyFFYEhcCY5JEMEE3Kli
awLwQ8hOnThJdMkycFRtwUf8jrQ2ntScvd0g1lPJGKm1Vrl2i5VnZu69mP6u775u
+2D2/VnGKhs/I0qUJDAnyIm860Qkmss9vk/Ves6OF8tiwdneHg56/0OGNFK8YT88
X7vZdrRTvJez/opMEi4r89fO4aL/3Xtw+zuhTaRjAv04l5U/BXCga99igUOLtFkN
SoxUnMW7gZ/NfaXvCyUeOiDbHPwfmGcCCtRzRBPbUYQaVQNW4AB+dAb/OMRyHdOo
P2gxXdMjxy6MW2Pg6Nwe0uxhHvLe5e/2mXZgLR6UcnHGCyoyx5JO1UbXHfmpGQrI
+pXObSOYqgs4rZpWDW+N8TEAiMEXnM0ZNjX+VVOg4DwzX5Ze4jLp3zO7Bkqp2IRz
znfSxqxx4VyyHqY7Ct9f4qNx2No3WqB4K/TUfet27fJhcKVlmtOJNBir+3I+17Q9
eVzYH6Eze9mCUAYTF6ps3MKCuwJXNq+YJyo5UOGwifUll35HaBC07HPKs5fRJNz2
YqAo07WjuGS3iGJCz51TzZm+ZGiPTx4SSPfSKcOYKMryMguTjClPPGAyzQWWYezy
r/6zcCwupvI=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICJTCCAaugAwIBAgIQLBcIfWQqWP6FGfKgz7RK6zAKBggqhkiOPQQDAzBUMQsw
CQYDVQQGEwJDTjEmMCQGA1UECgwdQkVJSklORyBDRVJUSUZJQ0FURSBBVVRI1JJ
VFkxHTAbBgNVBAMMFEJKQ0EgR2xvYmFsIFJvb3QgQ0EyMB4XDTE5MTIxOTAzMTgy
MVoxDTQ0MTIxMjAzMTYxN1owVDELMakGA1UEBhMCQ04xJjAkBgNVBAoMHUJFJSUpJ
TkcgQ0VSVElGSUNBVEUgQVVUSE9SSVRZMR0wGwYDVQQDDDBRCSknBIEdsb2JhbCBS
b290IENBMjBMBAGByqGSM49AgEGBSuBBAAiA2IABJ3LgJGNU2e1uVCxA/jlSR9B
IgmwUVJY1is0j8USRhTFiy8shP8sbqjV8QnjAyEUxEM9fMEsxEtqSs3ph+B99iK+
+kpRuDCK/eHeGBIK9ke35xe/J4rUQUyWPGCWwf0VHKNCMEAwHQYDVR00OBBYEfNJK
sVF/BvDRgh9Obl+rg/xI1LCRMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQD
AgEGMAoGCCqGSM49BAMDA2gAMGUCMBq8W9f+qdJUDkpd0m2xQNz0Q9XSSpkZElaA

94M04TVOSG0ED1cxMDAtsaqdAzjbBgIxAMvMh1PLet8gUXOQwKhbYdDFUDn9hf7B

43j4ptZLvZuHjw/1l1OWqzzIQNph91Oj9w==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFWTCCA0GgAwIBAgIBAjANBgkqhkiG9w0BAQsFADBOMQswCQYDVQQGEwJOTzEd
MBsGA1UECgwUQnV5cGFzcyBBUy05ODMxNjMzMjcjIDAeBgNVBAMMF0J1eXBhc3Mg
Q2xhc3MgMiBsb290IENBMB4XDTEwMTAyNjA4MzgwM1oXDTQwMTAyNjA4MzgwM1ow
TjELMAkGA1UEBhMCTk8xHTAbBgNVBAoMFEJ1eXBhc3MgQVMtOTgzMTYzMzI3MSAw
HgYDVQQDDDBdCdXlwYXNzIENsYXNzIDIGUm9vdCBDQTCCAiIwDQYJKoZIhvcNAQEB
BQADggIPADCCAgOCggIBANfHXvfBB9R3+0Mh9PT1aeTuMgHbo4Yf5FkNuud1g1Lr
6hxxFUi7HQfKjK6w3Jad6sNgkoaCKHOcVgb/S2TwDCo3SbXlzx87vFKu3MwZfPV
L4O2fuPn9Z6rYpNt8Z2SdlrkHJasW4DptfQxh6NR/Md+oW+OU3fUl8FVM5I+GC91
1K2GScuVr1QGbnGGE41b/+EmGVnAJLqBcXmQRFB0JJRfuLMR8SIBYaNBYYM21cHx
MIAQTn/0hpPshNOOvEu/XAFOBz3cFIqUCqTqc/sLUegTBxj6DvEr0VQVfTzh97QZ
QmdiXnfgolXsttlpF9U6r0TtSsWe5HonfOV116rLJeffawrbD02TTqigzXsu8lkB
arcNuAeBfos4GzjmCleZPe4h6KP1DBbdi+w0jpwqHAaVF41og9JwnxgIzRF01clr
Us3ERo/ctfPYV3Me6ZQ5BL/T3jjetFPsaRyifsSP5BtwrfKi+fv3FmRmaZ9JUaLi
FRhnBkp/1Wy1TbMz4GhrXb7pmA8y1x1LPC5aAVKRCfLf6o3YBkBjqhHk/sM3nhRS
P/TizPJhk9H9Z2vXUq6/aKtAQ6BXNVN48FP4YUIHZMbXb5tMOA1jrGKvNouicwoN
9SG9dKpN6nIDSdvHXx1iY8f93ZHsM+71bbRuMGjeyNYmsHVee7QHJihdjK4TWxP
AgMBAAGjQjBAMA8GA1UdEwEB/wQFMAMBAf8wHQYDVROBBYEFMmAd+BikoL1Rpzz
uvdMw964o605MA4GA1UdDwEB/wQEAwIBBjANBgkqhkiG9w0BAQsFAAOCAgEUAU18h
9bqwOli5LJKwbADJ784g7wbylp7ppHR/ehb8t/W2+xUbP6umwHJdELFx7rxP462s
A20ucS6vxOOto70MEae/0qyexAQH6dXQbLArvQsWdZHEIjzIVEpMMpghq9Gqx3t
OluwlN5E40EIosHsHdb9T7bWR9AUC8rmyrV7d35BH16Dx7aMOZawP5aBQW9gkOLO
+fsicdl9sz1Gv7SEr5AcD48Saq/v7h56rgJKihcrdv6sVlkkLE8/trKnToyokZf7
KcZ7XC25y2a2t6hbElGfTqI+Ynhw/qlqYLYdDnkM/crqJIByw5c/8nerQyIKx+u2
DISCLIBrQYoIwOula9+ZEsuK1V6ADJHgJgg2SMX6OBE1/yWDLfJ6v9r9jv6ly0Us
H8SIU653DtmadsWOLB2jutXsMq7Aqqz30XpN69QH4kj3Io6wpJ9qzo6ysmD0oyLQ
I+uUWnnp3Q+/QFesa1lQ2aOZ4W7+jQF5JyMV3pKdewlNWudLSDBaGOYKbeaP4NK7
5t98biGCwWg5TbSYWgzZizEqXsP6JwSxeRV0mcy+rSDeJmAc61ZRppPq5KM/p/9h
3PFaTWwyI0PurKju7koSCTxdccK+efrCh2gdC/1cacwG0Jp9VJkqyTkaGa9LKkPz
Y11aWOIv4x3kqdbQCtCev9eBCfHJxyYNrJgWVqA=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFWTCCA0GgAwIBAgIBAjANBgkqhkiG9w0BAQsFADBOMQswCQYDVQQGEwJOTzEd
MBsGA1UECgwUQnV5cGFzcyBBUy05ODMxNjMzMjcjIDAeBgNVBAMMF0J1eXBhc3Mg
Q2xhc3MgMyBsb290IENBMB4XDTEwMTAyNjA4MzgwM1oXDTQwMTAyNjA4MzgwM1ow
TjELMAkGA1UEBhMCTk8xHTAbBgNVBAoMFEJ1eXBhc3MgQVMtOTgzMTYzMzI3MSAw
HgYDVQQDDDBdCdXlwYXNzIENsYXNzIDMgUm9vdCBDQTCCAiIwDQYJKoZIhvcNAQEB
BQADggIPADCCAgOCggIBAKXaCpUWUOOV8l6ddjEGMnqb8RB2uACatVI2zSRHsJ8Y
ZLYa9vrVediQYkwiL944PdbgqOkcLnt4EemOaFEVcsfzM4fkoF0LXOBXByow9c3E
N3coTRiR5r/VUv1xLXA+58bEiuPwKAv0dphi4dVsjoT/Lc+JzeOIuOoTyrvYLS9
tznDDgFHMV0ST9tD+leh7fmdvhFHJlsTmKtdFoqwNxxXnUX/iJY2v7vKB3tvh2PX

0DJq1l1sDPGzbjniazEuOQAnFN44wOwZZoYS6J1yFhNkUsepNxx9giDthBgd9K5c
/3ATAOUx9TN6S9ZV+AWNS2mw9bMoNlwUxFFzTWsL8TQH2xc519woe2v1n/MuwU8X
KhDzzMro6/1rqy6any2CbgTUUGTLT2G/H783+9CHaZr77kgxve9oKeV/afmiSTY
zIw0bOIjL9kSGiG5VZFvC5F5GQytQIgLC0J60g7YaEi7ghM5EFjp2CoHxhLbWNvS
O1UQRwUVZ2J+GGOmRj8JDIQyXr8NYnon74Do29lLBlo3WiXQCBJ31G8JUJc9yB3D
34xFMFbG02SrZvPAXpacw8Tvw3xrzp5f7Njzz3iiZ+gMEuFuZyUJHmPfWupRWgP
K9Dx2zhLabjKSWJtyNBjYt1gD1iqj6G8BaVmos8bdrKEZLFMOVLAMLrwjEsCsLa3
AgMBAAGjQjBAMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFEE4zf/lb+74suwv
Tg75JbCOPGvDMA4GA1UdDwEB/wQEAwIBBjANBgkqhkiG9w0BAQsFAAOCAgEAACAj
QTUEkMJAYmDv4jVM1z+s4jSQuKFvdvoWFqRINyzpkMLyPPgKn9iB5btb2iUspKdV
cSQy9sgL8rxq+JOssgfCX5/bzMiKqr5qb+FJEMwx14C7u8jYog5kV+qi9cKpMRXS
IGrs/CIBKM+GuIAeqcwRpTzyFrNHnfzSgCHEy9BHcEGhyoMZCCxt8l13nIoUE9Q2
HJLw5QY33KbmkJs4j1xrG0aGQ0JfPgEHU1RdZX33inOhmlRaHylDFCfChQ+1iHsa
O5S3HWCntZznKWIXWpuTekMwGwPXYshApqr8ZORK15FTAaggiG6cX0S5y2CBNOxv
033aSF/rtJC8LakcC6wc1aJoIAE1vyxjy+7SjENSoYc6+I2KSb12tjE8nVhz36u
dmNKEkBlk4f4HoCMhuWG1o8O/FMsYOGWYRqiPkN7zTlgVGr18okmAWiDSKIz6MkE
kbIRNBE+6tBDGR8Dk5AM/1E9V/RBbuHL0L7ryWPNbczk+DaqaJ3tvV2XcEQNtg41
3OEMXbugUZTLfhbrES+jkkXITHHZvMmZUldGL1DPvTVp9D0VzgalLA8+9oG6lLvD
u79leNKGef9JOxqDDPDeeOzI8k1MGt6CKfjBWtrt7uYnXuhF0J0cUahq0Tj0ltq
4/g7u9xN12TyUb7mqqa6THuBrxzvxNiCp/HuZc=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFaTCCA1GgAwIBAgIJAJK4iNuwisFjMA0GCSqGSIb3DQEBCwUAMF1xCzAJBgNV
BAYTAiNLMRMwEQYDVQQHEwpCcmF0aXNzYXZlMRMwEQYDVQQKEwpEaXNpZyBhLnMu
MRkwFwYDVQQDExBdQSBGaXNpZyBSb290IFlyMB4XDTEyMDcxOTA5MTUzMFoXDTQy
MDcxOTA5MTUzMFoUjELMAkGA1UEBhMCU0sxZzARBgNVBAcTCkRjYXRrc2xhdmEx
EzARBgNVBAoTCkRjc2lnIGUucy4xGTAXBgNVBAMTEENBIERpc2lnIFJvb3QgUjIw
ggLiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCio8QACdaFXS1tFPbCw3Oe
NcJxVX6B+6tGUODBFel45qt5WDza/3wcn9iXAng+a0EE6UG9vgMsRfYvZNSrXaNH
PWSb6WiaxswbP7q+sos0Ai6YVRn8jG+qX9pMzk0DIaPY0jSTVpbLTAWAFjxfG3I
x2ymrdMxp7zo5eFm1tL7A7RBZckQrg4FY8aAamkw/dLukO8NJ9+flXP04SXabBbe
QTg06ov80egEFGEtQX6sx3dOy1FU+16SGBsEWmjGycT6txOgmLcRK7fWV8x8nhfR
yyX+hk4kLlYMeE2eARKmK6cBZW58Yh2EhN/qwGu1pSqVg8NTEQxzHQuyRpDRQjrO
QG6Vrf/GLK1ul4SOfW+eioANSW1z4nuSHsPzwfPrLgVv2RvPN3YEyLRa5Beny912
H9AZdugsBbPWnDTYltxhh5EF5EQIM8HauQhl1K6yNg3ruji6DOWbnuuNZt2Zz9aJ
QfYEkoopKW1rOhzndX0CcQ7zwOe9yxndnWCywmZgtrEE7snmhrmaZkCo5xHtgUUD
i/ZnWejBBhG93c+AAk9lQHhcR1DIm+YfgXvkRKhhbZri3lrVx/k6RGZL5DJUfORs
nLMOPReisjQS1n6yqEm70XooQL6iFh/f5DcfEXP7kAplQ6INfPgGAVUzfbANuPT1
rqVCV3w2EYx7XsQDnYx5nQIDAQABo0lwQDAPBgNVHRMBAf8EBTADAQH/MA4GA1Ud
DwEB/wQEAwIBBjAdBgNVHQ4EFgQUtZn4r7CU9eMg1gqtk5WpC5uQu0wDQYJKoZI
hvcNAQELBQADggIBACYGXnDnZTPIgm7ZnBc6G3pmsgH2eDtpXi/q/075KMOYKmfM
tCQSin1tERT3nLXK5ryeJ45MGcipvXrA1zYObYVybqjGom32+nNjf7xueQgcnYqf
GopTpti72TVVsRHFqQOzVju5hJMiXn7B9hJSi+osZ7z+Nkz1uM/Rs0mSO9MpDpkb
lvduHvEK7Z4bLQjb/D907JedR+Zlais9trhxTF7+9FGs9K8Z7RiVL0J92Owk6Ka
+elSLOTgEqv89WBW7xBci8QaQtyDW2QOy7W81k/BfDxujRNt+3vrMNDcTa/F1bal
TFtxyegxvug4BkihGuLq0t4SOVga/4AOgnXmt8kHbA7v/zjxmHHEt38OFdAlab0i

YXRlciBNYW5jaGVzdGVyMRAwDgYDVQQHEwdTYWxmb3JkMRowGAYDVQQKEExFDT01P
RE8gQ0EgTGltaxRlZDnMCUGA1UEAxMeQ09NT0RPIENlcnRpZmljYXRpb24gQXV0
aG9yaXR5MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0ECLi3LjkRv3
UCebVASy06m/weaKXTuH+7uIzg3jLz8GlvCiKVCZrts7oVewdFFxze1CkU1B/qnI
2GqGd0S7WWaXUF601CwRM/aN5VCaTwwxHGzUvAhTaHYujl8HJ6jJJ3ygxaYqhZ8
Q5sVW7euNJH+1GIImGEaaP+vB+fGQV+useg2L23IwambV4EajcNxo2f8ESil33rXp
+2dtQem8Ob0y2WIC8bGoPW43nOlV4tOiJovGuFVDiOEjPqXSJDlqR6sA1KGzqSX+
DT+nHbrTUcELpNqsOO9VUCQFZUaTNE8tja3G1CEZ0o7KBWfXb3NH5YoZE0ETc5O
nKVlrLsm9wIDAQABo4GOMIGLMB0GA1UdDgQWBBQLWOWLxkwVN6RAqTCpIb5HNlpW
/zAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAAQH/BAUwAwEB/zBJBgNVHR8EQjBAMD6g
PKA6hjhodHRwOi8vY3JsLmNvbW9kb2NhLmNvbS9DT01PRE9DZXJ0aWZpY2F0aW9u
QXV0aG9yaXR5LmNybDANBgkqhkiG9w0BAQUFAAOCAQEAAPpiem/Yb6dc5t3iuHXIY
SdOH5EOC6z/JqvWote9VfCFSZfnVDeFs9D6Mk3ORLgLETgdx8CPOGEIqB6BCsAv
IC9Bi5HcSEW88cbeunZrM8gALTFGTO3nnc+IIP8zwFboJIYmuNg4ON8qa90SzM/
RxdMosIGlnW2/4/PEZB31jiVg88O8EckzXZOFKs7sjsLjBOLDW0JB9LeGna8gl4
zJVSkl/BwJVmcIGfE7vmlV2H0knZ9P4SNVbfo5azV8fUZVqZa+5Acr5Pr5RzUZ5dd
BA6+C4OmF4O5MBKgxTMVBbkN+8cFduPYSo38NBejxiEovjBFMR7HeL5YYTisO+IB
ZQ==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICiTCCAg+gAwIBAgIQH0evqmIAcFBUTAGem2OZKjAKBggqhkiOPQQDAzCBhTEL
MAkGA1UEBhMCR0IxGzAZBgNVBAGTEkdyZWf0ZXIgtWTFuY2hlc3RlcjEQMA4GA1UE
BxMHU2FsZm9yZDEaMBGGA1UEChMRQ09NT0RPIENBIExpbWl0ZWQxKzApBgNVBAMT
IkNPTU9ETyBFQ0MgQ2VydGlmaWNhdGlvbiBBdXR0b3JpdHkwHhcNMDDgWmZa2MDAw
MDAwWhcNMzgwMTE4MjM1OTU5WjCBhTElMAkGA1UEBhMCR0IxGzAZBgNVBAGTEkdy
ZWf0ZXIgtWTFuY2hlc3RlcjEQMA4GA1UEBxMHU2FsZm9yZDEaMBGGA1UEChMRQ09N
T0RPIENBIExpbWl0ZWQxKzApBgNVBAMTIkNPTU9ETyBFQ0MgQ2VydGlmaWNhdGlv
biBBdXR0b3JpdHkwZjAQBgcqhkiOPQIBBgUrgQAIAiNAAQDR3svdcmCFYX7deSR
FtSrYpn1PIILBs5BAH+X4QokPB0BBO490o0JlwzgdET6+3eKKvUDYEs2ixYjFq0J
cfRK9ChQtP6IHG4/bC8vCVlbpVsLM5niwz2J+Wos77LTBumjQjBAMB0GA1UdDgQW
BBR1cacZSBm8nZ3qQUfflMRId5nTeTAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAAQH/
BAUwAwEB/zAKBggqhkiOPQQDAwNoADBlAJEA7wNbeqy3eApyt4jf/7VGFakK+qDm
fQjGGoe9GKhzvSbKYAydzpmfz1wPMOG+FDHqAJAU9JM8SaczepBGR7NjfrObTrdv
GDeAU/7dIOA1mjbRxxG55tZd8/8dLDoWV9mSOdY=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIF2DCCA8CgAwIBAgIQTKr5yttjb+Af907YWwOGnTANBgkqhkiG9w0BAQwFADCB
hTElMAkGA1UEBhMCR0IxGzAZBgNVBAGTEkdyZWf0ZXIgtWTFuY2hlc3RlcjEQMA4G
A1UEBxMHU2FsZm9yZDEaMBGGA1UEChMRQ09NT0RPIENBIExpbWl0ZWQxKzApBgNV
BAMTIkNPTU9ETyBSU0EgQ2VydGlmaWNhdGlvbiBBdXR0b3JpdHkwHhcNMATAwMTE5
MDAwMDAwWhcNMzgwMTE4MjM1OTU5WjCBhTElMAkGA1UEBhMCR0IxGzAZBgNVBAGT
EkdyZWf0ZXIgtWTFuY2hlc3RlcjEQMA4GA1UEBxMHU2FsZm9yZDEaMBGGA1UEChMR
Q09NT0RPIENBIExpbWl0ZWQxKzApBgNVBAMTIkNPTU9ETyBSU0EgQ2VydGlmaWNh
dGlvbiBBdXR0b3JpdHkwggLiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCR

6FSS0gpWsaWNJN3Fz0RndJkrN6N9I3AAcbxT38T6KhKPS38QVr2fcHK3YX/JSw8X
pz3jsARh7v8Rl8f0hj4K+j5c+ZPmNHRZFGvnnLOFoIJ6dq9xkNfs/Q36nGz637CC
9BR++b7Epi9Pf5l/txnQ3K9DADWietrLNPtj5gcFKt+5eNu/Nio5Jlk2kNrYrhV
/erBvGy2i/MOjZrkm2xpmfh4SDBF1a3hDTxFYPwyllEnvGfDyi62a+pGx8cgoLEf
Zd5ICLQkTqnyg0Y3hOvozIFIQ2dOciqbXL1MGyiKXCJ7tKuY2e7gUYPCDUZObT6Z
+pUX2nzwV0E8jVHTC7ZcryxjGt9XyD+86V3Em69FmeKjWiS0uqlWPc9vqv9JWL7w
qP/0uK3pN/u6uPQLOvnoQ0IeidiEyxPx2bvhiWC4jChWrBQdnArncevPDt09qZah
SL0896+1DSJMwBGB7FY79tOi4lu3sgQiUpWak2nojkl8ZEDLXB0AuqLZxUpaVIC
u9ffUGpVRr+goyhhf3DQw6KqLCCgQR84onAZFdr+CGCe01a60y1Dma/RMhnEw6abf
Fobg2P9A3fvQOq/ozM6LlweQRGBY84YcWsr7KaKtzFcOmpH4MN5WdYgGq/yapiq
crxXstJLnsQ/LBMQeXtHT1eKJ2czL+zUdqnR+WEUwIDAQABo0IwQDAdBgNVHQ4E
FgQUu69+Aj36pvE8hI6t7jiY7NkyMtQwDgYDVR0PAQH/BAQDAgEGMA8GA1UdEwEB
/wQFMAMBAf8wDQYJKoZIhvcNAQEMBAQDggIBAARx1UaEt65Ru2yyTUEUAJNMnMvl
wFTPoCWOAvn9sKIN9SCYPBMtrFaisNZ+EZLpLrqeLppysb0ZRGxhNaKatBYSaVqM
4dc+pBroLwP0rmEdEBsqpl6xf4FpuHA1sj+nq6PK7o9mfjYcwlYRm6mnPTXJ9OV
2jeDchzTc+Cir5kDOF3VSXkAKRzH7JsgHACKaVd4sjn8OoSgtZx8jb8uk2Intzna
FxiuvTwJaP+EmzzV1gsD41eeFPfR60/IvYcjt7ZJQ3mFXLrrkguhxuhqEwWsRqZ
CuhTLJK7oQkYdQxlqHvLI7cawiiFwxv/0Cti76R7CZGYZ4wUAc1oBmpjIXUDGIiK
boHGhfKppC3n9KUkEEeDys30jXIYsQab5xoq2Z0B15R97QNKYvDb6KkBPvVWmcke
jkk9u+UJueBPSZI9FoJAzMxZxuY67RlUaTxslbH9qh17f4a+Hg4yRvv7E491f0yL
S0Zj/gA0QHDBw7mh3aZw4gSzQbzpgJHqZJx64SIDqZxubw5IT2yHh17zbqD5daWb
QOHtsiedSrnAdyGN/4fy3ryM7xft0kL0fJuMAsaDk527RH89elWsn2/x20Kk4yl
0MC2Hb46TpSi125sC8KKfPog88Tk5c0NqMuRkrF8hey1FGlmdoLnzc7ILaZRfyHB
NVOFBkpdn627G190

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIB9zCCAX2gAwIBAgIQBiUzsUcdMydc+Y2aub/M+DAKBBggqhkJOPQQDAzA9MQsw
CQYDVQQGEwJVUzESMBAGA1UEChMJQ2VydGFpbmx5MR0wGAYDVQQDExFDZXJ0YWlu
bHkgUm9vdCBFMTAeFw0yMTA0MDEwMDAwMDBaFw00NjA0MDEwMDAwMDBaMD0xCzAJ
BgNVBAYTAiVTMRIeAYDVQQKEwldZXJ0YWlubHkgGjA5BgNVBAMTEUNlcnRhaW5s
eSBSb290IEUxMhYwEAYHKoZIzj0CAQYFK4EEACIDYgAE3m/4fxzf7flHh4axpmCK
+IKXgOqPyEpeKn2IaKcBYhSRJHpcnqMxfYqGITQYUBsQ3tA3SybHGwCA6TS9YBk2
QNYphwk8kXr2vBMj3VIOBF7PyAicGFPBMDjaIOIejeR2o0IwQDAOBgNVHQ8BAf8E
BAMCAQYwDwYDVR0TAAQH/BAUwAwEB/zAdBgNVHQ4EFgQU8ygYy2R17ikq6+2uI1g4
hevIIgcwCgYIKoZIzj0EAwMDAAwZQIxALGOWiDDshliTd6wT99u0nCK8Z9+aozm
ut6Dacpps6kFtZaSF4fC0urQe87YQVt8rgIwRt7qy12a7DLCZRawTDBcMPPaTnOG
BtjOiQRINzf43TNRnXCve1XYAS59BWQOhriR

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFRzCCAy+gAwIBAgIRAI4P+UuQcWhlM1T01EQ5t+AwDQYJKoZIhvcNAQELBQAw
PTELMAMGA1UEBhMCVVMxEjAQBgNVBAoTCUNlcnRhaW5seTEaMBGGA1UEAxMRQ2Vy
dGFpbmx5IFJvb3QgUjEwHhcNMjEwNDExMDAwMDAwWhcNNDYwNDExMDAwMDAwWjA9
MQswCQYDVQQGEwJVUzESMBAGA1UEChMJQ2VydGFpbmx5MR0wGAYDVQQDExFDZXJ0
YWlubHkgUm9vdCBFMTAeFw0yMTA0MDEwMDAwMDBaFw00NjA0MDEwMDAwMDBaMD0xCzAJ
BgNVBAYTAiVTMRIeAYDVQQKEwldZXJ0YWlubHkgGjA5BgNVBAMTEUNlcnRhaW5s

1B/q3avk0bbm+yLA3RMNansiExyXPGhjZjKcA7WNpIGD2ngwEc/csiu+kr+O5MQT
vqRoTNoCaBZ0vrLdBORrKt03H2As2/X3oXyVtwxwhi7xOu9S98zTm/mLvg7fMbed
aFySpvXl8wo0tf97ouSHocavFwDvA5HtqRxOcT3Si2yJ9HiG5mpJoM610rCrm/b0
1C7jcvk2xusVtyWMOvwlDbMicyF0yEqWYZL1LwsYpfSt4u5BvQF5+paMjRcCMLT5
r3gajLQ2EBAHBXDQ9DQGIlHFhiZ5shGIXsXwCITNSaa/ApzSRKft43jvRl5tcdF5
cBxGX1HpyTfcX35pe0HfNEXgO4T0oYoKNp43zGJS4YkNKPl6I7ENPT2a/Z2B7yyQ
wHtETrtJ4A5KVpK8y7XdeReJkd5hiXSSqOMyhb5OhaRLWcsrxXiOcVTQAjeZjOVJ
6uBUcQQRBi8LjMFbvrWhsFNunLhgkR9Za/kt9JQKl7XsxXYDVBtlUrpMklZRNAba
2CnblJ2Oy0wQJuK0EJWtLeIAaSHO1OWzaMWj/Nmqhexx2DgwUMFDO6bW2BvBlyH
Wyf5QBGGenDPBt+U1VwV/J84XIIwc/PH72jEpSe31C4SnT8H2TsIonPru4K8H+zMR
eiFPCyEQtkA6qyI6BjYlM4SGcprSp6XEtHWRqSsjAgMBAAGjQjBAMA4GA1UdDwEB
/wQEAwIBBjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBbTgqj8ljZ9EXME66C6u
d0yEPmcM9DANBgkqhkiG9w0BAQsFAAOCAgEAuVevuBLaV4OPaAszHQNTVfSVcOQr
PbA56/qJYv331hgELyE03fFo8NWWwt7CgKPBjcZq91l3rhVlz1t5BXdm6ozTaw3d
8VksWTOlMIAVRQdFGjEitpIAq5lNo093r6kiyi9jyhXWx8bwPWz8HA2YEGGeEali
1wrykXprOQ4vMMM2SZ/g6Q8CRFA3lFV96p/2O7qUpUzpvD5RtOjKkjZUbVwlKNrd
rRT90+7iIgXr0PK3aBLXWopBGsaSpVo7Y0VPv+E6dyIvXL9G+VoDhRNCX8reU9di
taY1BMJH/5n9hN9czulegChB8n3nHpDYT3Y+gjnW/KUD+nsa2UUeYNrEjvn8K8l7
lcUq/6qJ34IXD3L/DCfXCh5WAFaEJDJBlrXYFIW7pw0WwfgHJBu6haEaBQmAupVj
yTrsJZ9/nbqkRxWbRHDxakvWOF5D8xh+UG7pWijmZeZ3Gzr9Hb4DJqPb1OG7fpYn
Kx3upPvaJVQTA945xsMfTZDsxtK0hzthZU4UHIg1sGQUdGpXJpuHfUzVounmdLy
yCwzk5Iwx06MZTMQZBF9JBeW0Y3COMor6xOLRPIh80oat3df1+2IpHLIOR+Vnb5n
wXARpVb0+Em34yaXOp/SX3z7wJl8OSngex2/DaeP0ik0biQVY96QXr8axGbqwua6
OV+KmalBWQewLK8=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDqDCCApCgAwIBAgIJAP7c4wEPyUj/MA0GCSqGSIb3DQEBBQUAMdQxCzAJBgNV
BAYTAkZSMRlwEAYDVQQKDAIEaGlteW90aXMxETAPBgNVBAMMCENlcnRpZ25hMB4X
DTA3MDYyOTE1MTMwNVoXDTI3MDYyOTE1MTMwNVowNDELMAkGA1UEBhMCRIExEjAQ
BgNVBAoMCURoaW15b3RpciERMA8GA1UEAwwIQ2VydGlnbmEwggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQDIaPHJ1tazNHUmgh7stL7qXOEm7RFHYeGifBZ4
QCHkYJ5ayGPhxLGWkv8YbWkj4Sti993iNi+RB7lIzw7sebYs5zRLcAglozyHGxny
gQcPOJAZ0xH+hrTy0V4eHpbNgGzOOzGTtVkg0KmVEn2lmsxryIRWijOp5yIVUxbw
zBfsV1/pogqYCd7jX5xv3EjjhQsVWqa6n6xI4wmy9/Qy3l40vhx4XUJbzg4ij02Q
130yGLMLLGq/jj8UEYkgDncUtT2UCIf3JR7VsmAA7G8qKCvUk4YYxclPz5EIBb2
JsglrgVKtOdJLPOMFIN+XPsRGgjBRmKfIrjxwo1p3Po6WAbfAgMBAAGjgbwwgbkw
DwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUGu3+QTmQtCRZvgHyUtVF9lo53BEw
ZAYDVR0jBF0wW4AUGu3+QTmQtCRZvgHyUtVF9lo53BGhOKQ2MDQxCzAJBgNVBAYT
AkZSMRlwEAYDVQQKDAIEaGlteW90aXMxETAPBgNVBAMMCENlcnRpZ25hggkA/tzj
AQ/JSP8wDgYDVR0PAQH/BAQDAgEGMBEGCWCsGAGG+EIBAQQEAwIABzANBgkqhkiG
9w0BAQUFAAOCAQEAAhQMeknH2Qq/hO2Ge6/PAD/Kl1NqV5ta+aDY9fm4fTlrv0Q8h
bV6lUmPOEjvKtpv6zf+EwLHyzs+ImvaYS5/1HI93TDhHkxAGYwP15zRgzB7mFnc
fca5DCIMoTOi62c6ZYTTluLtdkVwj7Ur3vkj1kluPBS1xp81HIDQwY9qcEQCYsuu
HWhBp6pX6FOqB9IG9tUUBguRA3UsbHK1YZWaDYu5Def131TN3ubY1gkIl2PlwS6w
t0QmwCbAr1UwnjvVNioZBPRCvHv/PLlF/0P2HQBHVESO7SMAhqaQoLf0V+LBOK/Qw
WyH8EZE0vkHve52Xdf+XlcCWWC/qu0bXu+TZLg==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIGWzCCBEOgAwIBAgIRAMrpG4nxVQMNo+ZBbcTjpuEwDQYJKoZIhvcNAQELBQAw
WjELMAkGA1UEBhMCRIxEjAQBgNVBAoMCURoaW15b3RpczEcMBoGA1UECwwTMDDAw
MiA0ODE0NjMwODEwMDAzNjEzMBCGA1UEAwwQQ2VydGlnbmEgUm9vdCBDQTAeFw0x
MzEwMDEwODMyMjdaFw0zMzEwMDEwODMyMjdaMFoxCzAJBgNVBAYTAkZSMRIwEAYD
VQQKDAIEaGlteW90aXMxHDAaBgNVBAsMEzAwMDIgNDgxNDYzMDgxMDAwMzYxGTAX
BgNVBAMMEENlcnRpZ25hIFJvb3QgQ0EwggLiMA0GCsQGSib3DQEBAQUAA4ICDwAw
ggIKAoICAQDNGDIlGlmx6mQWDoyUJJV8g9PFOSbcDO8WV43X2KyjQn+Cyu3NW9sO
ty3tRQgXstmzy9YXUUnIo245Onoq2C/mehJpNdt4iKVzSs9IGPjA5qXSjklYcoW9M
CiBtNyN6tMbaLOQdLNyzKNAT8kxOAkmhVECe5uUFoC2EyP+YbNDrihqECB63aCPu
I9Vwzm1RaRDuoXrC0SIxwoKF0vJVdlB8JXrJhFwLrN1CTivngqIkicuQstDuI7pm
TLtipPlTWmR7fJj6o0ieD5Wupxj0auwuA0Wv8HT4Ks16XdG+RCYyKfHx9WzMfglh
C59vpD++nVPiz32pLHxYGpfhPTc3GGY0okDFUYqMwy3OU4gkWGQwFsWq4NYKpkDf
ePb1BHxpE4S80dGnBs8B92jAqFe7OmGtBIyT46388NtEbVncSVmurJqZNjBBE3Yz
IoejwpKGbvlw7q6Hh5UbxHq9MfPU0uWZ/75I7HX1eBYdpnDBfzwb0ZL7z8g81sWT
Co/1VTp2lc5ZmIoJlXcymoO6LAQ6l73UL77XbJuiyn1tJslV1c/DeVIICZkHJC1k
JWumIWmbat10TWuXekG9qxf5kBDIjzb5LdXF2+6qhUVB+s06RbFo5jZMm5BX7CO5
hwjCxAnxl4YqKE3idMDaxIzb3+KhF1nOJfI0Mdp//TBt2dzhouH8XwIDAQAB04IB
GjCCARYwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYE
FBiHVuBud+4kNTxOc5of1uHieX4rMB8GA1UdIwQYMBaAFBiHVuBud+4kNTxOc5of
1uHieX4rMEQGA1UdIAQ9MDswOQYEVVR0gADAXMC8GCCsGAQUFBwIBFiNodHRwcwov
L3d3d3cuY2VydGlnbmEuZnIvYXV0b3JpdGVzLzBtBgNVHR8EZjBkMC+gLaArhilo
dHRwOi8vY3JsLmNlcnRpZ25hLmZyL2NlcnRpZ25hcm9vdGNhLmNybdAxoC+gLYYr
aHR0cDovL2Nybc5kaGlteW90aXMxY29tL2NlcnRpZ25hcm9vdGNhLmNybdANBgkq
hkiG9w0BAQsFAAOCAgEAILieT/DjIQgi581oQfccVdV8AOItOoldaDgvUSILSo3L
6btdPrtcPbEo/uRTVRPPoZAbAh1fZkYJMyjhDSSXcNMQH+pkV5a7XdrnxIxPTGRG
HVyH41neQtGbqH6mid2PHMkwgu07nM3A6RngatgCdTer9zQoKJHyBApPNeNgJgH6
0BGM+RFq7q89w1DTj18zeTyGqHNfKlwgtnJzFyO+B2XleJINugHA64wcZr+shncB
lA2c5uk5jR+mUYyZDDl34bSb+hxnV29qao6pK0xXeXpXIs/NX2NGjVxZOob4Mkdi
o2cNGJHc+6Zr9UhhcyNZjgKnvETq9Emd8VRY+WCV2hikLyhF3HqgiIZd8zvn/yk1
gPxxKQ5Tm4xxvvq0OKmOZK8l+hfZx6AYDlf7ej0gcWtSS6Cvu5zHbugRqh5jnxV/v
faci9wHYTfmJ0A6aBVmknpiZbyvKcL5kwIwJ9Omvw5Ip3IgWJJk8jSaYtlu3zM63
Nwf9JtmYhST/WSMDmu2dnajkXjjO11INb9I/bbEFa0nOipFGc/T2L/Coc3cOZayh
jWZSaX5LaAZHHjcn6WMxwLkFM1JAbBzs/3GkDpv0mztO+7skb6iQ12LAEpmJURw
3kAP+HwV96LOPNdeE4yBFxgX0b3xdx4A61GU5wSesVywlVP+i2k+KYTLerj1KjL0=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICZTCCAeugAwIBAgIQeI8nXIESUiClBNAt3bpz9DAKBggqhkJOPQQDAZB0MQsw
CQYDVQQGEwJQTDEhMB8GA1UEChMYQXNzZWNVIERhdGEgU3lzdGVtYyBTLkEuMScw
JQYDVQQLEx5DZXJ0dW0gQ2VydGlnaWNhdGlviBBdXR0b3JpdHkxGTAXBgNVBAMT
EENlcnR1bSBFQy0zODQgQ0EwHhcNMjgwMzI2MDcyNDU0WhcNNDMwMzI2MDcyNDU0
WjB0MQswCQYDVQQGEwJQTDEhMB8GA1UEChMYQXNzZWNVIERhdGEgU3lzdGVtYyBT
LkEuMScwJQYDVQQLEx5DZXJ0dW0gQ2VydGlnaWNhdGlviBBdXR0b3JpdHkxGTAX

BgNVBAMTEENlcnR1bSBFQy0zODQgQ0EwdjAQBgcqhkJOPQIBBgUrgQQAiGniAATE
Kl6rGFtqvm5kN2PkzeyrOvfMobgOgknXhimfoZTy42B4mIF4Bk3y7JoOV2CDn7Tm
Fy8as10CW4kJPIMIRBSqniBMY81CE1700LCeJVf/OTOffph8oxPBUw7l8t1Ot68Kj
QjBAMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFi0GZnQkdjrzife81r1HfS+8
EF9LMA4GA1UdDwEB/wQEAwIBBjAKBggqhkJOPQQDAwNoADBlAjADVS2m5hjEfO/J
UG7BJw+ch69u1RsIGL2SKcHvIJF40jocVYli5RsJHrpka/F2tNQCMQC0QoSZ/6vn
nvuRlydd3LBbMHHOXjgaatkI5+r3YZJW+OraNsKHZZYuciUvf9/DE8k=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDuzCCAqOgAwIBAgIDBETAMA0GCSqSIlb3DQEBBQUAMH4xCzAJBgNVBAYTAiBM
MSIwIAAYDVQQKExlVbml6ZXRVIFRlY2hub2xvZ2llcyBTLkEuMScwJQYDVQQLEx5D
ZXJ0dW0gQ2VydGlmaWNhdGlubiBBdXR0b3JpdHkxIjAgBgNVBAMTGUNlcnR1bSBU
cnVzdGVvIE5ldHdvcmsgQ0EwHhcNMjE5MDYxMDIyMTIwNzYyMjE5MDYxMDIyMTIwNzYy
WjB+MQswCQYDVQQGEwJQTEiMCAGA1UEChMZVW5pemV0byBUZWNobm9sb2dpZXMG
Uy5BLjEnMCUGA1UECXMhY2VydHVtIENlcnR1bSBjYXRpb24gQXV0aG9yaXR5MSIw
IAAYDVQQDExlDZXJ0dW0gVHJ1c3RlZCBOZXR3b3JrIENBMBIIBIjANBgkqhkiG9w0B
AQEFAAOCAQ8AMIIBCgKCAQEAA4/t9o3K6wvDJFIf1awFO4W5AB7ptJ11/91sts1rH
UV+rpDKmYYe2bg+G0jACl/jXaVehGDldamR5xgFZrDwxSjh80gTSSyjoIF87B6LM
TXPb865Px1bVWqeWifr2jUI4ZZJ88JJ7ysbnKDHDBy3+Ci6dLhdHUZvSqeexVU
BBvXQzmtVSjF4hq79MDkrjhJM8x2hZ85RdKknvISjFH4fOQtf/WsX+sWn7Et0brM
kUJ3TCXJkDhy2/DM+44el1k+1WBO5gUo7UI5E0u6SNsv+XLTOcr+H9g0cvW0QM8x
AcPs3hEtF10fuFDRXhmnad4HMjyKUIX5p1TLVIZQRan5SQIDAQABo0IwQDAPBgNV
HRMBAf8EBTADAQH/MB0GA1UdDgQWBbQids3LB/8k9sXN7buQvOKEN0Z19zAOBgNV
HQ8BAf8EBAMCAQYwDQYJKoZIhvcNAQEFBQADggEBAKAorSLOAT2mo/9i0Eidi15y
sHhE49wcrwn9I0j6vSrEuVUEtRCjjSfeC4Jj0O7eDDd5QVsisrCaQVymcODU0HfL
I9MA4GxWL+FpDQ3Zqr8hgVDZBqWo/5U30Kr+4rP1mS1FhIrlQgnXdAlv94nYmem8
J9RHjboNRhx3zxSkHLMkMcScKHQDNP8zGSal6Q10tz6XxnboJ5ajZt3hrvJBW8qY
VoNzcOSGGtIxQbovvi0TWnZvTuhOgQ4/WwMioBK+ZlgRSssDxLQqKi2WF+A5VLxI
03YnnZotBqBJ7DnSq9ufmgnsAjUpsUCV5/nonFWIGUbWtzT1fs45mtk48VH3Tyw=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIF0jCCA7qgAwIBAgIQIdbQSk8lD8kyN/yqXhKN6TANBgkqhkiG9w0BAQ0FADCB
gDELMaKGA1UEBhMCUEwXlAgBgNVBAoTGUVuaXpldG8gVGvjaG5vbG9naWVzIFMu
QS4xJzAlBgNVBA5THkNlcnR1bSBBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTEkMCIG
A1UEAxMbQ2VydHVtIFRydXN0ZWQgTmV0d29yayBDQSAyMCIYDzIwMTExMDA2MDgz
OTU2WhgPMjA0NjEwMDYwODM5NTZaMIGAMQswCQYDVQQGEwJQTEiMCAGA1UEChMZ
VW5pemV0byBUZWNobm9sb2dpZXMGUy5BLjEnMCUGA1UECXMhY2VydHVtIENlcnR1
ZmljYXRpb24gQXV0aG9yaXR5MSQwIgYDVQQDExtDZXJ0dW0gVHJ1c3RlZCBOZXR3
b3JrIENBIDIwMjE5MDYxMDIyMTIwNzYyMjE5MDYxMDIyMTIwNzYyMjE5MDYxMDIyMTIwNzYy
DGSDhhuWZGc/IjoedQF97/tcZ4zJzFxrqZHmuULIEub2pt7uZld2ZuAS9eEQCsN
0+i6MLs+CRqnSZXvK0AkwpfHp+6bJe+cGcXhVqqndwpyeI1B+twTUrwWbNWuKFB
OJvR+zFj/Bf4bE/D44WSWDXB0Y+aomEKsq09DRZ40bRr5HMNUuctHfY9rnY3IE
fktjJlmgLjQ/KUxSiyqnwOKRKIm5wFv5HdnnJ63/mgKXwcZQkpsCLL2puTRZCr+E
Sv/f/rOf69me4Jgj7KZrdxYq28ytOxykh9xGc14ZYmhFV+SQgkK7QtbwYeDBoz1m

o130GO6IyY0XRSmZMnUCMe4pJshrAua1YkV/NxVaI2iJ1D7eTiew8EAMvE0Xy02i
sx7QBld9pPPV3WZ9fqGGmd4s7+W/jTcvedSVuWz5XV710GRBdxdaeOVDDUO5/IOW
OZV7bIBaTxNyxtD9KXpEulKkKtVBRgkg/iKgtlswjbyJDNXXcPiHUv3a76xRLgez
Tv7QCdpw75j6VuZt27VXS9zLLCUVyJ4ueE742pyehizKV/Ma5ciSixqClnrDvFAS
adgOWkaLOusm+iPJtrCBvkIApPjW/jAux9JG9uWOdf3yzLnQh1vMBhBgu4M1t15n
3kfsuUjxpKEV/q2MYo45VU85FrmxY53/twIDAQABo0IwQDAPBgNVHRMBAf8EBTAD
AQH/MB0GA1UdDgQWBBS2oVQ5AsOgP46KvPrU+Bym0ToO/TAOBgNVHQ8BAf8EBAMC
AQYwDQYJKoZIhvcNAQENBQADggIBAHLGIDs7k6b8/ONWJWsQCYftMxRQXLyYtPU2sQ
F/xlhMcQSZDe28cmk4gmb3DWA145oPePq5a1pRNcgRRtDoGCERuKTsZPpd1iHkTf
CVn0W3cLN+mLIMb4Ck4uWBzrM9DPhmDJ2vuAL55MYIR4PSFk1vtBHxgP58l1cb29
XN40hz5BsA72udY/CROWFC/emh1auVbONTqwX3BNXuMp8SMoclm2q8KMZiYcdywm
djWLKKdpPk79SPdhRB0yZADVPHnr7pH1BKXESLjokmUbOe3IEu6LaTaM4tMpkT/
WjzGHWTYtTHkpx6qFcL2+1hGsvxznN3Y6SHb0xRONbkX8eftoEq5IVIeVheO/jb
AoJnwTnbw3RLPTYe+SmTiGhbqEQZIfCn6IENLOiTNrQ3ssqwGyZ6miUfmpqAnksq
P/ujmv5zMnHCnsZy4YpoJ/HkD7TETKVhk/iXEAcqMCWpuchxuO9ozC1+9eB+D4Ko
b7a6bINDd82Kkhehnl4Fj1F4jNy3eFmynpTycUm/Q1oBEautmbjL4ZvrHG8hnj
XALKLNhvSgfZyTXaQHXYxKcZb55CEJh15pWLYLztxRLXis7VmFxWlgPF7ncGNf/P
5O4/E2Hu29othfDNrp2yGAlFw5Khchf8R7agCyzxxN5DaAhqXzvwmdmP7zAYspsbi
DrW5viSP
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFwDCCA6igAwIBAgIQHr9ZULjJgDdMBvfrVU+17TANBgkqhkiG9w0BAQ0FADB6
MQswCQYDVQQGEwJQTDEhMB8GA1UEChMYQXNzZWNVIERhdGEgU3lzdGVtcyBTLkEu
MScwJQYDVQQLEw5DZXJ0dW0gQ2VydGlmaWNhdGlubiBBdXR0b3JpdHkxH2AdBgNV
BAMTFkNlcnR1bSBUCnVzdGVkIFJvb3QgQ0EwHhcNMjgwMzE2MTIwMDEzWhcNNDMw
MzE2MTIwMDEzWjB6MQswCQYDVQQGEwJQTDEhMB8GA1UEChMYQXNzZWNVIERhdGEg
U3lzdGVtcyBTLkEuMScwJQYDVQQLEw5DZXJ0dW0gQ2VydGlmaWNhdGlubiBBdXR0
b3JpdHkxH2AdBgNVBAMTFkNlcnR1bSBUCnVzdGVkIFJvb3QgQ0EwggLiMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQDRLY67tzbqTeRn06TpwXkKQmIzhyC93yZ
n0EGze2jusDbCsZBfN8pftlL5On1AFrAygYo9idBcEq2EXxkd7f09CAAozPOA/q
p1x4EaTBByIVcJdPTsuclzxFUl6s1wB52HO8AU5853BSILCils3Jy/I2z5T4IHhQq
NwuIPMqw9MjCoa68wb4pZ1Xi/K1ZXP69VyywkI3C7Te2fJmItDUDmj0VDT06qKhF
8JVOJVkdZzhp9PMMsmN74H+rX2Ju7pgE8pllWeg8xn2A1bUatMn4qGtg/BKEiJ3
HAVz4hlxQsDsUaakFjgao4rpUYwBI4Zshfjvqm6f1bxJAPXsiEodg42MEx51UGa
mqi4NboMOvJEGyCi98U11z3G4z5D3Yf+xOr1Uz5MZf87Sst4WmsXXw3Hw09Omiqi
7VdNIuJGmj8PkTQkfVXjjJU30xrwCSss0smNtA0Aq2cpKNgB9RkEth2+dv5yXMSF
ytKAQd8FqKPvHJBPC/PgP5sZ0jeJP/J7UhyM9uH3PAeXjA6iWYEMspA90+NZRu0P
qafegGtaqge2Gcu8V/OXIXoMsSt0Puvap2ctTMSYnjYJdmZm/Bo/6khUHL4wvYBQ
v3y1zgD2DGHZ5yQD4OMBgQ692IU0iL2yNqh7XAJlRICMb/gv1SHKHRzQ+8S1h9E6
Tsd2tTVItQIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBSM+xx1
vALTn04uSNn5YFSqxLNP+jAOBgNVHQ8BAf8EBAMCAQYwDQYJKoZIhvcNAQENBQAD
ggIBAEii1QALLtA/vBzVtVRJHlpr9OTy4EA34MwUe7nJ+jW1dReTagVphZzNTxl4
WxmB82M+w85bj/UvXgF2Ez8sALnIIl5SW0ETsXpD4YN4fqzX4IS8TrOZgYkNCvo
zMrnadyHncI013nR03e4qllY/p0m+jiGPp2Kh2RX5Rc64vmNueMzeMGQ2Ljdt4NR
5MTMI9UGfOZR0800McD2RrsLrfw9EAUqO0qRJe6M1ISHgCq8CYyqOhNf6DR5UMeq
GfnTKB7U0VEwKbOukGfWHwpjscWpxkIxYxeU72nLL/qMFH3EQxiJ2fAyQOaA4kZf

5ePBAFmo+eggvIksDkc0C+pXwlM2/KfUrzHN/gLldfq5Jwn58/U7yn2fqSLLiMmq
0Uc9NneoWWRrJ8/vJ8HjJLWG965+Mk2weWjROeiQWMODvA8s1pfrzgzhiMfat7D
P78v3DSk+yshzWePS/Tj6tQ/50+6uaWTRRxmHyH6ZF5v4HaUMst19W7l9o/HuKTM
qJZ9ZPskWkoDbGs4xugDQ5r3V7mzKWmTOPQD8rv7gmsHINF5SH5pkAnuYZttcTVoP
0ISVoDwUQwbKytu4QTbaakRnh6+v40URFWkIsr4WOZckbxJF0WddCajJFdr60qZf
E2Efv4WstK2tBZQIgx51F9NxO5NQI1mg7TyRVJ12AMXDuDjb

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICHTCCAaOgAwIBAgIUQ3CCd89NXTTxyq4yLzf39H91oJ4wCgYIKoZIzj0EAwMw
TjELMAkGA1UEBhMCVVMxEjAQBgNVBAoMCUNvbW1TY29wZTERMCKGA1UEAwwiQ29t
bVNjb3BlIFB1Ym9pYyBUcnVzdCBFQ0MgUm9vdC0wMTAeFw0yMTA0MjgxNzM1NDNa
Fw00NjA0MjgxNzM1NDJaME4xCzAJBgNVBAYTAiVTMRIwEAYDVQQKDAIDb21tU2Nv
cGUxKzApBgNVBAMMIkNvbW1TY29wZSBQdWJsaWMgVHJ1c3QgRUNDIFJvb3QtMDEw
djAQBgcqhkiOPQIBBgUrgQQAIGNiAARLNumuV16ocNfQj3Rid8NeeqrltqLxeP0C
flfdkXmcblLISiF58LwS+uM32ENEp7LXQoMPwiXAZu1FlxUOcW5tjnSCDPgYLpkJE
hRGnSjot6dZoL0hOUysHP029uax3OVejQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYD
VR0PAQH/BAQDAgEGMB0GA1UdDgQWBBSOB2LAUN3GGQYARnQE9/OufXVNMDAKBggq
hkJOPQDAwNoADBlAjEAnDPfQeMjQEI2Jpc1XHvr20v4qotzVRVcrHgpD7oh2MSg
2NED3W3ROT3Ek2DS43KyAjB8xX6I01D1HiXo+k515liWpDVfG2XqYZpwI7UNo5uS
Um9poIyNStDuiw7LR47QjRE=

-----END CERTIFICATE-----

-----BEGIN

CERTIFICATE-----

MIICHDCCAaOgAwIBAgIUkP2ZYEfHpgE6yhR7H+/5aAiDXX0wCgYIKoZIzj0EAwMw
TjELMAkGA1UEBhMCVVMxEjAQBgNVBAoMCUNvbW1TY29wZTERMCKGA1UEAwwiQ29t
bVNjb3BlIFB1Ym9pYyBUcnVzdCBFQ0MgUm9vdC0wMjAeFw0yMTA0MjgxNzQ0NTRa
Fw00NjA0MjgxNzQ0NTNaME4xCzAJBgNVBAYTAiVTMRIwEAYDVQQKDAIDb21tU2Nv
cGUxKzApBgNVBAMMIkNvbW1TY29wZSBQdWJsaWMgVHJ1c3QgRUNDIFJvb3QtMDIw
djAQBgcqhkiOPQIBBgUrgQQAIGNiAAR4MIHoYx7l63FRD/cHB8o5mXxO1Q/MMDAL
j2aTPs+9xYa9+bG3tD60B8jzljHz7aRP+KNOjSkVWLjVb3/ubCK1sK9IRQq9qEmU
v4RDsNuESgmJGwdqb8FuvAY5N9GIUvejQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYD
VR0PAQH/BAQDAgEGMB0GA1UdDgQWBbTmGHX/72DehKT1RsfeSlXjMjZ59TAKBggq
hkJOPQDAwNnADBkAjAmc0l6tqvmSfr9Uj/UQQSugEODZXW5hYA4O9Zv5JOGq4/n
ich/m35rChJVYaoR4HkCMHfoMXGsPHED1oQmHhS48zs73u1Z/GtMMH9ZzkXpc2AV
mkzw5l4lhVtwodZ0LKOag==

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFbDCCA1SgAwIBAgIUgXUwdDGOTKvVxZApIsU5EN0wDQYJKoZIhvcNAQEL
BQAwTjELMAkGA1UEBhMCVVMxEjAQBgNVBAoMCUNvbW1TY29wZTERMCKGA1UEAwwi
Q29tbVNjb3BlIFB1Ym9pYyBUcnVzdCBUSU0EgUm9vdC0wMTAeFw0yMTA0MjgxNjQ1
NTRaFw00NjA0MjgxNjQ1NTNaME4xCzAJBgNVBAYTAiVTMRIwEAYDVQQKDAIDb21t
U2NvcGUxKzApBgNVBAMMIkNvbW1TY29wZSBQdWJsaWMgVHJ1c3QgUlNBIFJvb3Qt
MDEwggliMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCwSGWjDR1C45FtnYSk
YZYSwu3D2iM0GXb26v1VWvZVAVMP8syMl0+5UMuzAURWlv2bKOx7dAvnQmtVzslh

suitQDy6uUEKBU8bJoWPQ7VAtYXR1HHcg0Hz9kXHgKKEUJdGzqAMxGBWBB0HW0al
DrJLpA6lfO741GIDuZNqihS4cPgugkY4lw50x2tBt9Apo52AsH53k2NC+zSDO3Oj
WiE260f6GBfZumbCk6SP/F2krfxQapWsvCQz0b2lf4b19bJzKo98rwjyGpg/qYfI
P8GMicWWMJoKz/TUyDTtnS+8jTiGU+6Xn6myY5QXjQ/cZip8UIF1y5mO6D1cv547
KI2DAg+pn3LiLCuz3GaXAEDQpFSOm117RTYm1nJD68/A6g3czhLmfTifBSeolz7p
UcZsBSjBAG/pGG3svZwG1KdJ9FQFa2ww8esD1eo9anbCyxooSU1/ZOD6K9pzg4H/
kQO9ILvkuI6cMmPNn7togbGEW682v3fuHX/3SZtS7NJ3Wn2RnU3COS3kuoL4b/JO
Hg9O5j9ZpSPcPYeoKFgo0fEbNttPxP/hjFtyjMcmAyejOQoBqsCyMWCDIqFPEgkB
Ea801M/XrmLTBQe0MXXgDW1XT2mH+VepuhX2yFJtocucH+X8eKg1mp9BFM6ltM6U
CBwJrVbl2rZJmkrqYxhTnCWuwwIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4G
A1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUUN12mmnQywsL5x6YVEFm45P3luG0wDQYJ
KoZIHvcNAQELBQADggIBAK+nz97/4L1CjU3IpbfaOp9TSp90K09FlxD533Ahuh6
NWPxzIHlxgvoLl1pKZJkGNRRDSsBTtXAOntYtPZKdVUvhwQkZyybf5Z/Xn36lbQ
nmhUQo8mUuJM3y+Xpi/SB5io82BdS5pYV4jvguX6r2yBS5KPQJqTRlnLX3gWsWc+
QgvfKNmwrZggvkN80V4aCRckjXtdlemrwWCrWxhkgPut4AZ9HcpZuPN4KWfGVh2v
trV0Knahp/t1MJ+UXjulYPPLXAZiDslg+MkfFoom3ecnf+slpoq9uCO2EJqxWE2a
aE9gVOX2RhOOiKy8IUISrcZKiX2bwdgt6ZYD9KJ0DLwAHb/WNyVntHKLr4W96ioD
j8z7PEQkguIBpQtZtjSNMgsSDesnvw1B10A8ckYpwlzqg/xBpMu95yo9GA+o/E4
Xo4TwbM6l4c/ksp4qRyv0LAbJh6+cOx69TOY6lz/KwsETkPdY34Op054A5U+1C0w
lREQKC6/oAI+/15Z0wUOIv9TRe9rh9VlzRamloPh37MG88EU26fsHltdkJANclHn
YfkUyq+Dj7+vsQpZXdxcl+SWrVtgHdql7I52Qb1dgAT+GhMIbA1xNxVssnBQVoc
icCMb3SgazNNtQEO/a2tiRc7ppqEvOuM6sRxJKi6KfkIsidWNTJf6jn7MZRVGczw
-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFbDCCA1SgAwIBAgIUUVBa/O345lXGN0aoApYYNK496BU4wDQYJKoZIhvcNAQEL
BQAwTjELMAkGA1UEBhMCVVMxEjAQBgNVBAoMCUNvbW1TY29wZTERMCKGA1UEAwwi
Q29tbVNjb3BlIFB1YmtpYyBUcnVzdCBSU0EgUm9vdC0wMjAeFw0yMTA0Mjg1NzE2
NDNaFw00NjA0Mjg1NzE2NDJaME4xCzAJBgNVBAYTAiVtMRIwEAYDVQQKDAIDb21t
U2NvcGUxKzApBgNVBAMMIkNvbW1TY29wZSBQdWJsaWMgVHJ1c3QgUINBIFJvb3Qt
MDIwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDh+g77aAASyE3VrCLE
NQE7xVTIWXZjpX/rwcRqmL0yjReA61260WI9JSMZNRTPf4mnG2I81lDnNJUDMrG0
kyI9p+Kx7eZ7T6Hmw0zdQreqjXnfuU2mKKuJZ6VszKWpCtYHu8//mI0SFHrt1C
rWDaSWqVcN3SAOLMV2MCe5bdSZdbkk6V0/nLKR8YSvgBKtJjCW4k6YnS5cciTNxz
hkcAqg2Ijq6FfUrpuzNPDlJwnZXjfG2WWy09X6GDRI224yW4fKcZgBzqZUPckXk2
LHR88mcGyYnJ27/aaL8j7dxrrSiDeS/sOKUNNWFnJ5rpM9kzXzehxfCrPfp4sOcs
n/Y+n2Dg70jpkEUeBVF4GiwSLFworA2iI540jwXmojPOEXcT1A6kHkIfhs1w/tku
FT0du7jyU1fbzMZ0KZwYszZ1OC4PVKH4kh+Jlk+71O6d6Ts2QrUKOyrUZHk2EOH5
kQMreyBUzQ0ZGshBMjTRsJnhkB4BQDa1t/qp5Xd1pCKBXbCL5CcSD1SIxtuFdOa3
wNemKfrb3vOTlycEVS8KbzfFPROvCgCpLlscgSjX74Yxqa7ybrjKaixUR9gqiC6v
wQcQeKwRoi9C8DdfF8rhW3Q5iLc4tVn5V8qdE9isy9COoR+jUKgF4z2rDN6ieZdlS
5fq6M8EGRPbmz6UNp2YINios8wIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4G
A1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUR9DnsSL/nSz12Vdgs7GxcJXvYXowDQYJ
KoZIHvcNAQELBQADggIBAIzpsU0v6Z9PIpNojuQhmaPORVMbc0RTAIFhzTHjCLqB
KCh6krm2qMhDnscTJk3C2OVVnJJdUnjCK9v+5qiXz1I6JMNlZFxHMaNINRPdk7n3
+VGXu6TwYof1gbTl4MgqX67tiHCpQ2EAOHyJxCDut0DgdXdaMNmEMjRdrSzbyme
APnCKfwxkx1SaRosTKCL4BwaMS/TiJVZbuXEs1DIFAhKm4sTg7GkcrI7djNB3Nyp

RIUWaqZLKfm7lVa/Rh3sHZMdwGWyH6FDrlaeoLGPaxK3YG14C8qKXO0elg6DpkiV
jTujIcSuWMYASoS0I6SWhjW42J7YrDRJmGOVxcttSEfi8i4YHtAxq9107PncjLgc
jmgjutDzUNzPZY9zOjLHfP7KgiJPvo5iR2blzYfi6NUPGJ/lBHJLRjwQ8kTCZFzx
TnXonMkmdMV9WdEKWw9t/p51HBjGGjp82A0EzM23RWV6sY+4roRIPrN6TagD4uJ+
ARZZaBhDM7DS3LAaQzXupdqprlyuhoFBAUp0JuyfBr/CBTdkdXgpaP3F9ev+R/nk
hbDhezGdpn9yo7nELC7MmVcOIQxFAZRl62UJxmMiCzNJkkg8/M3OsD6Onov4/knF
NXJHAgMBAAGjgY4wgYswDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUqvyREBuH
kV8Wub9PS5FeAByxMoAwDgYDVR0PAQH/BAQDAgEGMEkGA1UdHwRCMEAwPqA8oDqG
OGh0dHA6Ly9jcmwuZC10cnVzdC5uZXQvY3JsL2QtdHJ1c3RfZXZfcm9vdF9jYV8y
XzIwMjY3JmMA0GCSqGSIb3DQEBDQUAA4ICAQCTy6UfmrHsmg1fLBWTxj++EI14
QvBukEdHjqOSMo1wj/Zbjb6JzkcBahsgIllbyIIQbODnmaprxigYzWRaoUlrRc4
pZt+UPJ26oUFKIDBK7GB0aL2QHWpDsvxVUjY7NHss+jOFKE17MJeNRqrphYBB07q
3C+jisoksetSjl8MmxPy3MHGcRqwnNU73xDUmPBEcrCRbH0O1P1aa4846XerOhU
t7KR/aypH/KH5BfGSah82ApB9PI+53c0BFLd6IHYS9URZ0V4U/M5d40VxDJl3IX
cI1QcB9WbMy5/zpaT2N6w25lBx2Eof+pDGOJbbJAiDnXH3dotfyc1dZnaVuodNv8
ifYbMvekJKZ2t0dT741J6m2g1qlpBFYfXeA08mD6iL8AOWsKwV0HFaanuU5nCT
2vFp4LJiTZ6P/4mdm13NRemUAiKN4DV/6PEEeXFsVIP4M7kFMhtYVRFP0OUnR3Hs
7dpn1mKmS00PaaLJvOwiS5THaJQXfuKOKD62xur1NGyfn4gHONuGcfrNIUhDbqNP
gofXNjhuS5N5YHVpD/Aa1VP6IQzCP+k/HxiMkl14p3ZnGbuy6n/pcAlWVqOwDAst
NI7F6cTVg8uGF5csbBNvh1qvSaYd2804BC5f4ko1Di1L+KIkBI3Y4WNeApI02pHH
XBxvWHZks/wCuPWdCg==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIEHzCCAxugAwIBAgIDCYPzMA0GCSqGSIb3DQEBDQUAAQME0xCzAJBgNVBAYTAkRF
MRUwEwYDVQQKDAxELVRydXN0IEEdtYkgxJzAlBgNVBAMMHkQtVFJlVU1QgUm9vdCBD
bGFZcyAzIENBIDIgMjAwOTAeFw0wOTExMDUwODM1NThaFw0yOTExMDUwODM1NTha
ME0xCzAJBgNVBAYTAkRFMRUwEwYDVQQKDAxELVRydXN0IEEdtYkgxJzAlBgNVBAMM
HkQtVFJlVU1QgUm9vdCBDDbGFZcyAzIENBIDIgMjAwOTCCASlwdQYJKoZIhvcNAQEB
BQADggEPADCCAQoCggEBANoySs96R+91myP6Oi/WUEWJNTrGa9v+2wBoQOADER03
UAiftUpolDWzU9GUY6cgVq/eUXjsKj3zSehQPgrfRIWLJ23DEE0NkVJD2IfgXU42
tSHKXzlABF9bfsyxiupQB7ZNoTWSPOSHjRGICTBpFGOShrvUD9pXRI/RcPHAY9R
ySPocq60vFYJfXLLHLGvKZAKyVXMD9O0Gu1HNVPk7ZxzBCHQqr0ME7UAyiZsxGsM
lFqVlNpQmVh/pStmMaTJOKDfHR+4CS7zp+hnUquVH+BGptikw8paxTGA6Eian5Rp
/hnd2HN8gcqW3o7tszIFZYQ05ub9VxC1X3a/L7AQDcUCAwEAAaOCARowggEWMA8G
A1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFp3aFMSfMN4hvR5COFYrYyNJ4PGEMA4G
A1UdDwEB/wQEAwIBBjCB0wYDVR0fBIHLMiHIMIGAoh6gfIZ6bGRhcDovL2RpcmVj
dG9yeS5kLXRydXN0Lm5ldC9DTj1ELVRSVVNUJTIwUm9vdCUyMENsYXNzJTIwMyUy
MENBJTIwMiUyMDIwMDksTz1ELVRydXN0JTIwR21iSCxDPURFP2NlcnRpZmljYXRl
cmV2b2NhdGlvbmxpc3QwQ6BBOD+GPWh0dHA6Ly93d3cuZC10cnVzdC5uZXQvY3Js
L2QtdHJ1c3Rfcm9vdF9jYGFzc18zX2NhXzJfMjAwOS5jcmwwDQYJKoZIhvcNAQEL
BQADggEBAH+X2zDI36ScfSF6gHDOFBjpiBSVYEQBrLLpME+bUMJm2H6NMLVwMeni
acfcZnsgFYbQDfC+rAF1hM5+n02/t2A7nPPKHeJeaNijnZflQGDSNiH+0LS4F9p0
o3/U37CYAqxva2ssJSRyoWXuJVrl5jLn8t+rSfrzkGkj2wTZ51xY/GXU177M/C4K
zCUqNQ4YJEVdT1B/yMfGchs64JTBKbkTCJNjYy6zltz7GRUUG3RnFX7acM2w4y8
PIWmawomDeCTmGCufsYkl4phX5GOZPlJhzbNi5stPvZR1FDUWSi9g/LMKHtThm3Y
Johw1+qRzT65ysCQblRGXnRl11z+o+I=

CERTIFICATE-----

MIEQzCCAyugAwIBAgIDICYP0MA0GCSqGSIb3DQEBCwUAMFAxSzAJBgNVBAYTAkRF
 MRUwEwYDVQQKDAxELVRydXN0IEdtYkxgKjAoBgNVBAMMIUQtVFJlVU1QgUm9vdCBD
 bGFzcyAzIENBIDIGRVYgMjAwOTAeFw0wOTExMDUwODUwNDZaFw0yOTExMDUwODUw
 NDZaMFAxSzAJBgNVBAYTAkRFMRUwEwYDVQQKDAxELVRydXN0IEdtYkxgKjAoBgNV
 BAMMIUQtVFJlVU1QgUm9vdCBDbGFzcyAzIENBIDIGRVYgMjAwOTCCASIwDQYJKoZI
 hvCNAQEBBQADggEPADCCAQoCggEBAJnxhDRwui+3MKCOvXwEz75ivJn9gpfSegpn
 lJgJ9hBOISJzmY3aFS3nBfwZcyK3jpgAvDw9rKFs+9Z5JUut8Mxk2og+KbgPCdM0
 3TP1YtHhzRnp7hhPTFiu4h7WDFsVWtg6uMQYZB7jM7K1iXdODL/ZlGsTl28So/6Z
 qQTMFexgaDbtChu39b+T7WYxg4zGcTSHThfqr4uRjRxWQa4iN1438h3Z0S0NL2lR
 p75mpoo6Kr3HGrHhFPC+Oh25z1uxav60sUYgovseO3Dvk5h9jHOW8sXvhXCtKsB8
 HgQ+HKDYD8tSg2J87otTlZCpV6LqYQXY+U3EJ/pure3511H3a6UCAwEAAaOCASQw
 ggEgMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBByEFNOuikxiEyoZLsyvcp09Ntea
 HNxnMA4GA1UdDwEB/wQEAwIBBjCB3QYDVR0fBIHVMiHSMIGHoIGeOIGBhn9sZGFw
 Oi8vZGlyZWNo63J5LmQtdHJ1c3QubmV0L0NOPUQtVFJlVU1QIMjBSb290JTIwQ2xh
 c3MlMjAzJTlwQ0ElMjAyJTlwRVYlMjAyMDA5LE89RC1UcnVzdCUyMEdtYkgsQz1E
 RT9jZXJ0aWZpY2F0ZXJldm9jYXRpb25saXN0MEagRKBCkHkHrWoi8vd3d3LmQt
 dHJ1c3QubmV0L2Nybc9K9LXRydXN0X3Jvb3RfY2xhc3NfM19jYV8yX2V2XzlwMDku
 Y3JsMA0GCSqGSIb3DQEBCwUAA4IBAQA07XtaPKSUiO8aEXUHL7P+PPoeUSbrh/Yp
 3uDx1MYkCenBz1UbtDDZzhr+BlGmFaQt77JLvyAoJUnRpjZ3NOhk31KxEcdzes05
 nsKtjHEh8lpr988TIWvsorlFIIm5d8sqMb7Po23Pb0iUMkZv53GMOkaEGTcH8gNF
 CSuGdXzfX2IXANtu2KZYlktQ1HWYVt+3GP9DQ1CuekR78HlR10M9p9OB0/DJT7na
 xpeG0ILD5EJt/rDiZE4OJudANCa1CInXCGNjOCd1HjPqbjdn5lPdE2BiYBL3ZqX
 KVwvvoFBuYz/6n1gBp7N1z3TLqMVvKjmJuVvw9y4AyHqnxbxLFS1

CERTIFICATE-----

MIIDtZCCAp+gAwIBAgIQDOfg5RfYRv6P5WD8G/AwOTANBgkqhkiG9w0BAQUFADBl
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLEwB3
d3cuZGlnaWNlcnQuY29tMSQwIgYDVQQDEwEaWdpQ2VydCBBC3N1cmVkieIEIFJv
b3QgQ0EwHhcNMDYxMTEwMDAwMDAwWhcNMzExMTEwMDAwMDAwWjBlMQswCQYDVQQG
EwJVUzEVMBMGA1UEChMMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLEwB3d3cuZGlnaWNl
cnQuY29tMSQwIgYDVQQDEwEaWdpQ2VydCBBC3N1cmVkieIEIFJvb3QgQ0EwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIABAQCtDhXO5EOAXLGH87dg+XESpa7c
JpSIqvTO9SA5KFhgDPiA2qkVITJhPLWxKISKityfCgyDF3qPkKyK53lTXDGEKvYP
mDI2dsze3Tyooou9q+yHyUmHfnyDXH+Kx2f4YZNISW1/5WBg1vEfNoTb5a3/UsDg+
wRvDjDPZ2C8Y/igPs6eD1sNuRMBhNZYW/lmci3Zt1/GiSw0r/wty2p5g0I6QNcZ4
VYcgoc/lbQrISXwxmDNsIumH0DJaoroTghHtORedmTpyoeb6pNnVFzF1roV9Iq4/
AUaG9ih5yLHa5FcXxH4cDrC0kqZW572yl+2qp/C3xag/lRbQ/6GW6whfGHdPAgMB
AAGjYzBhMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQW
BBRF66Kv9JLLgjEtUYunpyGd823IDzAfBgNVHSMEGDAWgBRF66Kv9JLLgjEtUYun
pyGd823IDzANBgkqhkiG9w0BAQUFAAOCAQEAAog683+Lt8ONyc3pkIL/3cmbYMuRC
dWKuh+vy1dneVrOfzM4UKLkNl2BcEkxY5NM9g0lFWJc1aRqoR+pWxnmrEthngYtf
fwk8lOa4JiwgvT2zKIn3X/8i4peEH+ll74fg38FnSbNd67lJKusm7Xi+ft8R87cm

NW1fiQG2SVufAQWbqz0lwcy2f8Lxb4bG+mRo64EtlOtCt/qMHt1i8b5QZ7dsvfPx
H2sMNgcWfzd8qVttevESRmCD1ycEvkvOI77DZypoEd+A5wwzZr8TDRRu838fYxAe
+o0bJW1sj6W3YQGx0qMmoRBxna3iw/nDmVG3Kwclzi7mULKn+gpFL6Lw8g==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDljCCAn6gAwIBAgIQC5McOtY5Z+pnI7/Dr5r0SzANBgkqhkiG9w0BAQsFADBl
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSQwIgYDVQQDEExtEaWdpQ2VydCBBC3N1cmVkiEIEIFJv
b3QgRzIwHhcNMTMwODAxMTIwMDAwWhcNMzgwMTE1MTIwMDAwWjBlMQswCQYDVQQG
EwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cuZGlnaWNl
cnQuY29tMSQwIgYDVQQDEExtEaWdpQ2VydCBBC3N1cmVkiEIEIFJvb3QgRzIwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDZ5ygvUj82ckmlkzTz+GoeMVSA
n61UQbVH35ao1K+ALbkKz3X9iaV9JPrjIgwrvJUXCzO/GU1BBpAAvQxNEP4Htecc
biJVMWwXvdMX0h5i89vqbFCMP4QMls+3ywPgym2hFEwbid3tALBSfK+RbLE4E9Hp
EgjAALAcKxHad3A2m67OeYfcgnDmCXRwVWmvo2ifv922ebPynXApVfSr/5Vh88lA
bx3RvpO704gqu52/clpWcTs/1PPRCv4o76Pu2ZmvA9OPYLfykqGxvYmJHzDNw6Yu
YjOuFgJ3RfrngQo8p0Quebg/BLxcoIfhG69Rjs3sLPr4/m3wOnyqi+RnlTGNAgMB
AAGjQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgGGMB0GA1UdDgQW
BBTOw0q5mVXyuNtgvl+vVa1lzan1jANBgkqhkiG9w0BAQsFAAOCAQEAYqVVjOPI
QW5pJ6d1Ee88hjZv0p3GeDgdaZaikmkuOGybfQTUiaWxMTeKySHMq2zNixya1r9I
0jJmwYrA8y8678Dj1JGG0VDjA9tzd29KOVpt3ibHtX2vK0LRdWLjSisCx1BL4Gni
lmwORGYQRI+tBev4eaymG+g3NJ1TyWGqolKvSnAWhsI6yLETcDbYz+70CjTVW0z9
B5yiutkBclzzTcHdDrEcDcRjvq30FPuJ7KJBDkzMyFdA0G4Dqs0MjomZmWzwPDCv
ON9vvKO+KSAnq3T/EyJ43pdSVR6DtVQgA+6uwE9W3jfMw3+qBCe703e4YtsXfJwo
IhNzbM8m9Yop5w==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICRjCCAc2gAwIBAgIQC6Fa+h3foLVJRK/NJKBs7DAKBggqhkiOPQQDAzBlMQsw
CQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cu
ZGlnaWNlcnQuY29tMSQwIgYDVQQDEExtEaWdpQ2VydCBBC3N1cmVkiEIEIFJvb3Qg
RzMwHhcNMTMwODAxMTIwMDAwWhcNMzgwMTE1MTIwMDAwWjBlMQswCQYDVQQGEwJV
UzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cuZGlnaWNlcnQu
Y29tMSQwIgYDVQQDEExtEaWdpQ2VydCBBC3N1cmVkiEIEIFJvb3QgRzMwdjAQBgcq
hkjOPQIBBgUrgQQAIGNiAAQZ57ysRGXtzbG/WPuNsVepRC0FFfLvC/8QdJ+1YlJf
Zn4f5dwbRXkLzMTZCp2NXQLZqVneAlr2lSoOjThKiknGvMYDOAdfVdp+CW7if17Q
RSAPWXYQ1qAk8C3eNvJsKTmjQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/
BAQDAgGGMB0GA1UdDgQWBbTL0L2p4ZgFUaFNN6KDec6NHSrkhDAKBggqhkiOPQQD
AwNnADBkAjAlpIFFAmsSS3V0T8gj43DydXLEfInwz5FyYZ5eEJJZVrmDxxDnOOlY
JjZ91eQ0hjkCMHw2U/Aw5WJjOpnitqM7mzT6HtoQknFekRON3aRukswy1vUhZscv
6pZjamVFkpUBtA==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDrzCCApEgAwIBAgIQCDvgVpBCRRGhdWrJWZHHSjANBgkqhkiG9w0BAQUFADBh
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLEExB3
d3cuZGlnaWNLcnQuY29tMSAwHgYDVQQDEXdEaWdpQ2VydCBHbG9iYWwgUm9vdCBD
QTAEFw0wNjExMTAwMDAwMDBaFw0zMTEwMTAwMDAwMDBaMGExCzAJBgNVBAYTAIVT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j
b20xIDAeBgNVBAMTF0RpZ2lDZXJ0IEdsb2JhbCBScSb290IENBMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4jvhEXLeqKTT01eqUKKPC3eQyaKl7hL0llsB
CSDMAZOnTjC3U/dDxGkAV53ijSLdhwZAAIEJzs4bg7/fzTtxRuLWZscFs3YnFo97
nh6Vfe63SKMI2tavegw5BmV/SI0fvBf4q77uKNd0f3p4mVmFaG5cIzJLv07A6Fpt
43C/dxC//AH2hdmoRBBYmql1GNXRor5H4idq9Joz+EkIYIvUX7Q6hL+hqkpMfT7P
T19sdl6gSzeRntwi5m3OFBqOasv+zbMUZBFHWymeMr/y7vrTC0LUq7dBMtoM1O/4
gdW7jVg/tRvoSSiicNoxBN33shbyTApOB6jtSj1etX+jkMOvJwIDAQAB02MwYTAO
BgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUA95QNVbR
TLtm8KPiGxvDI7I90VUwHwYDVR0jBBgwFoAU95QNVbRtLtm8KPiGxvDI7I90VUw
DQYJKoZIhvcNAQEFBQADggEBAMucN6pIExIK+t1EnE9SsPTfrgT1eXkloyQY/Esr
hMATudXH/vTBH1jLuG2cenTnmCmrEbXjKChzUyImZOMkXDiqw8cvpOp/2PV5Adg
06O/nVsJ8dWO41P0jmP6P6ftGbfYmbW0W5Bjflttep3Sp+dWOIrWcBAI+0tKIJF
PnlUkiaY4IBIqDfv8NZ5YBberOgOzW6sRBc4L0na4UU+Krk2U886UAb3LujEV0ls
YSEY1QSteDwsOoBrp+uvFRTp2InBuThs4pFsiv9kuXclVzDAGySj4d3p30d8tbQk
CAUw7C29C79Fv1C5qfPrmAESrciXpg0X40KPMbp1ZWVbd4=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDjjCCAnagAwIBAgIQAzrx5qcRqaC7KGSxHQn65TANBgkqhkiG9w0BAQsFADBh
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLEExB3
d3cuZGlnaWNLcnQuY29tMSAwHgYDVQQDEXdEaWdpQ2VydCBHbG9iYWwgUm9vdCBH
MjAEFw0xMzA4MDEwMjAwMDBaFw0zODAxMTUxMjAwMDBaMGExCzAJBgNVBAYTAIVT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j
b20xIDAeBgNVBAMTF0RpZ2lDZXJ0IEdsb2JhbCBScSb290IEcyMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuzfNNN7a8myaJCtSnX/RrohCgiN9RIUyful
2/Ou8jqJkTx65qsGGmvPrC3oXgkkRLpimn7Wo6h+4FR1IAWsULecYxpsMNzaHxm
1x7e/dfgy5SDN67sH0NO3Xss0r0upS/kqbitOtSZpLYl6ZtrAGCSYP9PIUkY92eQ
q2EGnI/yuum06ZIya7XzV+hdG82MHauVBJVJ8zUtlunJbd134/tJS7SsVQepj5Wz
tCO7TG1F8PapsPuwtp1MVYwnSlcUfIKdzXOS0xZKBgyMUNGPHgm+F6HmIcr9g+UQ
vIOICsRnKPZzFBQ9RnbDhxSJITRNrw9FDKZJobq7nMWxM4MphQIDAQAB0IwQDAP
BgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBhjAdBgNVHQ4EFgQUtIUIBiV
5uNu5g/6+rkS7QYXjzkWdQYJKoZIhvcNAQELBQADggEBAGBnKJRvDkhj6zHd6mcY
1Yl9PMWLSn/pvtsrF9+wX3N3KjITOYFnQoQj8kVnNeyIv/iPsGEMNKSulEyExtv4
NeF22d+mQrvHRAiGfzZ0JFrabA0UWTW98kndth/Jsw1HKj2ZL7tcu7XUIOGZX1NG
Fdtom/DzMNU+MeKNhJ7jitalj41E6Vf8PlwUHBHQRFXGU7Aj64GxJUTFy8bJZ91
8rGOmaFvE7FBcf6IKshPECBV1/MURexGRPTqh5Uykw7+U0b6LJ3/iyK5S9kJRaTe
pLiaWN0bfVKfjllDiIGknibVb63dDcY3fe0Dkhvld1927jyNxF1WW6LZZm6zNTfl
MrY=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICPzCCAcWgAwIBAgIQBVVWvPJepDU1w6QP1atFcjAKBggqhkhjOPQQDAzBhMQsw
CQYDVQQGEwJVUzEVMBMGA1UEChMMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cu
ZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBHMzAe
Fw0xMzA4MDEwMjAwMDBaFw0zODAxMTUxMjAwMDBaMGEwCzAJBgNVBAYTAiVTMRUw
EwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5jb20x
IDAeBgNVBAMTF0RpZ2lDZXJ0IEdsb2JhbCBSc290IEczMHYwEAYHKoZIzj0CAQYF
K4EEACIDYgAE3afZu4q4C/sLfYHS8L6+c/MzXRq8NOrexp80JX28MzQC7phW1FG
fp4tn+6OYwwX7Adw9c+ELkCDnOg/QW07rdOkFFk2eJ0DQ+4QE2xy3q6Ip6FrtUPO
Z9wj/wMco+I+o0IwQDAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBhjAd
BgNVHQ4EFgQU9tIpPmhxdIuNkHMEWNpYim8S8YwCgYIKoZIzj0EAwMDaAAwZQIx
AK288mw/EkrRLTnDCgmXc/SINoyIJ7vml1Qhadj+Z4y3maTD/HMsQmP3Wyr+mt/
oAIwOWZbwmSNuJ5Q3KjVSaLtx9zRSX8XAbjIho9OjIgrqJqpisXRAL34VOKa5Vt8
sycX

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDxTCCAq2gAwIBAgIQAxqJmoLQJuPC3nyrkYldzANBgkqhkiG9w0BAQUFADBs
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDEyJEAWdpQ2VydCBlaWdoIEFzc3VyYW5j
ZSBFViBSb290IENBMB4XDTA2MTEwMDAwMDAwMFoXDTEwMDAwMDAwMDAwMFowbDEL
MAkGA1UEBhMCVVMxFTATBgNVBAoTDERpZ2lDZXJ0IEluYzEZMBcGA1UECzMqd3d3
LmRpZ2ljZXJ0LmNvbTErMCKGA1UEAxMiRGlnaUNlcnQgSGlnaCBBC3N1cmFuY2Ug
RVYgUm9vdCBDQTCCASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMbM5XPm
+9S75S0tMqbf5YE/yc0lSbZxKsPVIDRnogocF9ppkCxxLeyj9CYpKIBWTrt3JTW
PNt0OKRKzE0lgvdKpVMSO07zSW1xkX5jtqumX8OkhPhPYIG++MXs2ziS4wblCJEM
xChBVfvLWokVfnHoNb9Ncgk9vjo4Uft3MRuNs8ckRZqnrG0AFFoEt7oT61EKmEFB
Ik5IYYeBQVCmeVyJ3hlKV9Uu5l0cUyx+mM0aBhakaHPQNAQTXXFx01p8VdteZOE3
hzBWBOUrtCmAeVf5OYiiAhF8J2a3iLd48soKqDirCmTCv2ZdlYTBosUeh10aUAsg
EsxBu24LUTi4S8sCAwEAAANjMGEwDgYDVROPAQH/BAQDAgGGMA8GA1UdEwEB/wQF
MAMBAf8wHQYDVROBBYEFLE+w2kD+L9HADsYJhoIAu9jZCvDMB8GA1UdIwQYMBAA
FLE+w2kD+L9HADsYJhoIAu9jZCvDMA0GCSqGSIb3DQEBAQUAA4IBAQAACGgaX3Nec
nzyIZgYIVyHbIUf4KmeqvxygdkAQV8GK83rZEWwONfqe/EW1ntlMMUu4kehDLI6z
eM7b41N5cdblIZQB2lWHmiRk9opmzN6cN82oNLFpmyPinngiK3BD41VHMWEZ71jF
hS9OMPagMRYjyOfiZRYzy78aG6A9+MpeizGLYAiJLQwGXFK3xPkKmNEVX58Svnw2
Yzi9RKR/5CYrCsSXAQ3pjOLAEFe4yHYSkVXySGnYvCoCWw9E1CAx2/S6cCZdkGce
vEsXCS+0yx5DaMkHJ8HSXPfqIbloEpw8nL+e/IBcm2PN7EeqJSdnoDfzAIJ9VNep
+OkuE6N36B9K

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICGTCCAaz+gAwIBAgIQCeCTZaz32ci5PhwLBCou8zAKBggqhkhjOPQQDAzBOMQsw
CQYDVQQGEwJVUzEXMBUGA1UEChMMORGlnaUNlcnQsIEluYy4xJjAkBgNVBAMTHURp
Z2lDZXJ0IFRMUyBFQ0MGUDM4NCBSb290IEc1MB4XDTIxMDEwNTAwMDAwMFoXDTEw
MDEwNDIzNTk1OVowTjELMAkGA1UEBhMCVVMxFTATBgNVBAoTDkRlZ2lDZXJ0LCBJ
bmMuMSYwJAYDVQQDEx1EaWdpQ2VydCBUTFMgRUNDIFAzODQgUm9vdCBHNTB2MBAG
ByqGSM49AgEGBSuBBAAiA2IABMFEOc8Rl1Ca3iOCNqfN0MsYndLxf3c1TzvdIHJS
7cl7+Oz6e2tYIOyZrsn8aLN1udsJ7MgT9U7GCh1mMEy7H0cKPGEQQil8pQgO4CLp

0zVozptjn4S1mU1YoI71VOeVyaNCMEAwHQYDVR0OBByEFMFRRVBZqz7nLFr6ICIS
B4CIfBFqMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MAoGCCqGSM49
BAMDA2gAMGUCMQCJao1H5+z8blUD2WdsJk6Dxv3J+ysTvLd6jLRL0mlpYxNjOyZQ
LgGheQaRnUi/wr4CMEfDFXuxoJGZSZOoPHzoRgaLLPIxAJSdYsiJvRmEFOMl+wG4
DXZDjC5Ty3zfDBeWUA==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFZjCCA06gAwIBAgIQCPm0eKj6ftpqMzeJ3nzPijANBgkqhkiG9w0BAQwFADBN
MQswCQYDVQQGEwJVUzEXMBUGA1UEChMORGlnaUNlcnQsIEluYy4xJTAjBgNVBAMT
HERpZ2lDZXJ0IFRmUyBSU0E0MDk2IFJvb3QgRzUwHhcNMjEwMTE1MDAwMDAwWhcN
NDYwMTE0MjM1OTU5WjBjBNMQswCQYDVQQGEwJVUzEXMBUGA1UEChMORGlnaUNlcnQs
IEluYy4xJTAjBgNVBAMTHERpZ2lDZXJ0IFRmUyBSU0E0MDk2IFJvb3QgRzUwggli
MA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCz0PTJeRGd/fxmgefM1eS87IE+
ajWOLrfn3q/5B03PMJ3qCQuZvWxX2hhKuHisOjmopkisLnLlvevxGs3npAOPxG0
2C+JFvuUAT27L/gTBaF4HI4o4EXgg/RZG5Wzrn4DReW+wkL+7vI8toUTmDKdFqgp
wgscONyFMXdcvyej/Cestyu9dJsXLfKB2l2w4SMXPohKEiPQ6s+d3gMXsUJKoBZM
pG2T6T867jp8nVid9E6P/DsjyG244gXazOvswzH016cpVIDPRFtMbZCe88zdH5RD
nU1/cHAN1DrRN/BsnZvAFJNY781BOHW8EwOVfH/jXOnVDdXifBBiqmvwPXbzP6Po
sMH976pXTayGpxi0KcEsDr9kvimM2AltVwv8n/vFfQMfawKsPHTDU9qTXeXAaDx
Zre3zu/O7Oyldcqs4+Fj97ihBMi8ez9dLRYiVu1ISf6nL3kwJZu6ay0/nTvEF+cd
Lvvyz6b84xQslpgHjLSR6Rlgg/IwKwZzUNWYOwbpx4oMYIwo+FKbbuH2TbsGJJvX
KyY//SovcfXWJL5/MZ4PbeiPT02jP/816t9JXkGPhvnx3d3Lg7SjXi/7RgLQZhNe
XoVPzthwiHvOAbWW19fNff2C+MIkwcoBOU+NosEUQB+cZtUMCUbW8tDRSHZWOkPL
tgoRObqME2wGtZ7P6wIDAQABo0IwQDAdBgNVHQ4EFgQUUTMc7TZArxfTJc1paPKv
TiM+s0EwDgYDVR0PAAQH/BAQDAgGGMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcN
AQEMBAQDggIBAGCmr1tfV9qJ20tQqcQjNSH/0GEwhJG3PxDPJY7Jv0Y02cEhJhxxw
GXleo8mH/qlDZJY6yFMECrZBu8RHANmfGBg7sg7zNOok992vIGCukihfNudd5N7H
PNtQOa27PShNlnx2xlv0wdsUpasZYgcYQF+Xkdycx6u1UQ3maVNVzDI92sURVXLf
O4uJ+DQtpBfIf+aZfTCiITfNMBc9uPK8qHWgQ9w+iUuQrm0D4ByjoJYJu32jtyoQ
REtGBzRj7TG5BO6jm5qu5jF49OokYTurWGT/u4cnYiWB39yhL/btp/96j1EuMPik
AdKFOV8BmZZvWltwGUb+hmA+rYAQCd05JS9Yf7vSdPD3Rh9GOUrYU9DzLjtxpdRv
/PNn5AeP3SYZ4Y1b+qOTEZvpyDrDVWiaKuFSdjo4bq9+0/V77PnSIMx8IIh47a+
p6tv75/ftM8BuGJqIz3nCU2AG3swpMPdB380vqQmsvZB6Akd4yCYqjdP//fx4ilw
MUc/dNAUFvohigLVigmUdy7yWSiLfFCSCmZ4OIN1xLVaqBHG5cGdZlXPU8Sv13WF
qUITVuwhd4GTWgzqltJyqEI8pc7bZsEGCREjnwB8twl2F6GmrE52/WRMmrRpnCK
ovfepEWFJqgejF0pW8hL2JpqA15w8oVPbEtoL8pU9ozaMv7Da4M/OMZ+

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFkDCCA3igAwIBAgIQBZsbV56OITLiOQe9p3d1XDANBgkqhkiG9w0BAQwFADBi
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLEXB3
d3cuZGlnaWNlcnQuY29tMSEwHwYDVQQDEXhEaWdpQ2VydCBUCnVzdGVkIFJvb3Qg
RzQwHhcNMTEwMDAwMDAwWhcNMTEwMDAwMDAwWjBiMQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLEXB3d3cuZGlnaWNlcnQuY29tMSEwHwYDVQQDEXhEaWdpQ2VydCBUCnVzdGVkIFJvb3QgRzQwggliMA0GCSqG

SIb3DQEBAQUAA4ICDwAwggIKAoICAQC/5pBzaN675F1KPDAiMGkz7MKnJS7JIT3y
ithZwuEppz1Yq3aaza57G4QNxDaf8xukOBbrVsaXbR2rsnnyyhHS5F/WBTxSD1If
xp4VpX6+n6lXFllVcq9ok3DCsrp1mWpzMpTREEQQLt+C8weE5nQ7bXHiLQwb7iDV
ySAdYyktzuxeTsiT+CFhmzTrBcZe7FsavOvJz82sNEBfsXpm7nfISKhmV1efVFiO
DCu3T6cw2Vbuyntd463JT17lNecxy9qTXtyOj4DatpGYQJB5w3jHtrHEtWoYOAMQ
jdjUN6QuBX2I9YI+EJFwq1WCQTLX2wRzKm6RAXwhTNS8rhsDdV14Ztk6MUSaM0C/
CNdaSaTC5qmgZ92kJ7yhTzm1EVgX9yRcRo9k98FpiHaYdj1ZXUJ2h4mXaXpI8OCi
EhtmmnTK3kse5w5jrubU75KSOp493ADkRSWJtppEGSt+wJS00mFt6zPZxd9LBADM
fRyVw4/3IbKyEbe7f/LVjHAsQWCqsWMYRJUadmJ+9oCw++hkpjPRiQfhvbfmQ6QY
uKZ3AeEPIAwHbJUKSWJbOUOUIFHdL4mrLZBdd56rF+NP8m800EREIvIEFDrMcXK
chYiCd98THU/Y+whX8QgUWtvsauGi0/C1kVfnSD8oR7FwI+isX4KJpn15GkvmB0t
9dmpsh3lGwIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIB
hjAdBgNVHQ4EFgQU7NfjgtJxXWRM3y5nP+e6mK4cD08wDQYJKoZIhvcNAQEMBQAD
ggIBALth2X2pbL4XxJEbw6GiAI3jZGgPVs93rnD5/ZpKmbnJeFwMDF/k5hQpVgs2
SV1EY+CtnJYYZhsjDT156W1r1lT40jzBQ0CuHVD1UvyQO7uYmWlrx8GnqGikJ9yd
+SeuMIW59mdNOj6PWTkiU0TryF0Dyu1Qen1iIQqAyHNm0aAFYF/opbSnr6j3bTWc
fFqK1q14mfN4i/RN0iAL3gTujJtHgXINwBQy7zBZLq7gcfJW5GqXb5JQbZaNaHqa
sjYUegbyJLkJEVDXCLG4iXqEI2FCKeWjzaIgQdfRnGTZ6iahixTXTBmyUEFXPT9N
cCOGDercgdlMMpSEDQgJlxxPwO5rIHQw0uA5NBCFIRUBCOhVMt5xSdKoF1BN5r5N
0XWs0Mr7QbhDparTwwVETyw2m+L64kW4I1NsBm9nVX9GtUw/bihaeSbSpKhil9Ie
4u1Ki7wb/UdKDd9nZn6yW0HQO+T0O/QEY+nvwlQAUaCKKsnOeMzV6ocEGLP0r0mI
r/OSmbaz5mEP0oUA51Aa5BuVnRmhuZyxm7EAHu/QD09CbMkKvO5D+jpxpchnJqU1
/YldvIViHTLSocT07ZpXwdv6EM8Zt4tKG48BtieVU+i2iW1bvGjUI+iLUaJW+fCm
gKDWHR08Dw9TdSmq6hN35N6MgSGtBxBHEa2HPQfRdbzP82Z+

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIEkTCCA3mgAwIBAgIERWtQVDANBgkqhkiG9w0BAQUFADCBsDELMAkGA1UEBhMC
VVMxVjFjAUBgNVBAoTUDUvVudHJ1c3QsIEluYy4xOTA3BgNVBAsTMHd3dy5lbnRydXN0
Lm5ldC9DUFMgaXMgaW5jb3Jwb3JhdGVkIGJ5IHJlZmVyZW5jZTEfMB0GA1UECzMW
KGMpIDIwMDYyYzUwcnVzdCwgSW5jLjEtMCsGA1UEAxMkRW50cnVzdCBSb290IENl
cnRpZmljYXRpb24gQXV0aG9yaXR5MB4XDTA2MTEyNzIwMloXDTE2MTEyNzIwMlo
NTM0MlowgbAxCzAJBgNVBAYTAiVTMRYwFAYDVQQKEw1FbnRydXN0LCBjbmMuMTkw
NwYDVQQLEzB3d3cuZW50cnVzdC5uZXQvQ1BTIGlzlGluY29ycG9yYXRlZCBieSBY
ZWZlcmVuY2UxHjAdBgNVBAsTFihjKSAyMDA2IEVudHJ1c3QsIEluYy4xLTAuRjBmNV
BAMTJEVudHJ1c3QgUm9vdCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTCCASlwdQYJ
KoZIhvcNAQEBBQADggEPADCCAQoCggEBALaVtkNC+sZtKm9I35RMOVcf7sN5EUFO
Nu3s/poBj6E4KPz3EEZmLk0eGrEaTsbRwJWIsMn/MYszA9u3g3s+IRe7bJWKKf4
4LIACtFfY0cOlypowCKVYhXbR9n10Cv/gkvJrT7eTNUqGFA/CYqEAOWwCj0Yzfv9
KlmaI5UXLEWeH25DeW0MXJj+SKfFI0dcXv1u5x609mhF0YaDW6KKjbHjKYD+JXGI
rb68j6xSlkuqUY3kEzEZ6E5Nn9uss2rVvDIUccp6en+Q3X0dgNmBu1kmwhH+5pPi
94DkZfs0Nw4pgHBNrziGLp5/V6+eF67rHMSolV+2HNjnogQi+dPa2MsCAwEAAoOB
sDCBrTAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUwAwEB/zArBgNVHRAEJDAi
gA8yMDA2MTEyNzIwMloqBDZlWmJYxMTI3MjA1MzQyWjAfBgNVHSMEGDAWgBRo
kORnpKZTgMeGZqTx90tD+4S9bTAdBgNVHQ4EFgQUaJdkZ6SmU4DHhmak8fdLQ/uE
vW0wHQYJKoZIhVZ9B0EABBAwDhsIVjcuMT0LjADAgSQMA0GCSqGSIb3DQEBBQUA
A4IBAQCT1DCw1wMgKtD5Y+iRDAUgqV8ZyntYtTsX29CW+1RaGSwMCPeyvIWonX9t

Rkfz6/djwUAFQKXSt/S1mja/qYh2iARVBCuch38aNzx+LaUa2NSJXsq9rD1s2G2v
1fN2D807iDginWyTmsQ9v4IbZT+mD12q/OWyFcq1rca8PdCE6OoGcrBNOTJ4vz4R
nAuknZoh8/CbCzB428Hch0P+vGOaysXCHMnHjf87ElgI5rY97HosTvuDls4MPGmH
VHOkc8KT/1EQrBVUAdj8BbGJoX90g5pJ19xOe4pIb4tF9g==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIcejCCAgCgAwIBAgIQMZch7a+JQn81QYehZ1ZMbTAKBggqhkJOPQQDAzBuMQsw
CQYDVQQGEwJFUzEcMBoGA1UECgwTRmlybWFwcm9mZXNpb25hbCBTQTEYMBYGA1UE
YQwPVkFURVMtQTYyNjM0MDY4MScwJQYDVQQDDB5GSVJNQVBST0ZFU0IPTkFMIENB
IFJPT1QtQSBXRUIwHhcNMjIwMDkxMTM2WWhcNNDcwMzIxMDkxMTM2WjBuMQsw
CQYDVQQGEwJFUzEcMBoGA1UECgwTRmlybWFwcm9mZXNpb25hbCBTQTEYMBYGA1UE
YQwPVkFURVMtQTYyNjM0MDY4MScwJQYDVQQDDB5GSVJNQVBST0ZFU0IPTkFMIENB
IFJPT1QtQSBXRUIwdjAQBgcqhkJOPQIBBgUrgQQAIGNiAARHU+osEaR3xyrq89Zf
e9MEkVz6iMYiuYMQYneEMy3pA4jU4DP37XcsSmDq5G+tbT4TIqk5B/K6k84Si6C
cyvHZpsKjECcflr28jgst7L7Ljkb+qbXbdTkBgYVcUgt5SjYzBhMA8GA1UdEwEB
/wQFMAMBAf8wHwYDVR0jBBgwFoAUk+FDY1w8ndYn81LsF7Kpryz3dvvgWQYDVR0O
BBYEFJPhQ2NcPJ3WJ/NS7Beyqa8s93b4MA4GA1UdDwEB/wQEAwIBBjAKBggqhkJOP
PQQDAwNoADBlAjAdfKR7w4l1M+E7qUW/Runpod3JIha3RxEL2Jq68cgLcFBTApFw
hVmpHqTm6iMxoAACMQD94vizrxa5HnPEluPMBBnYfubDI94cT7iJLzPrSA8Z94dG
XSaQpYXFuXqUPoeovQA=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFiDCCA3CgAwIBAgIIIfQmX/vBH6nowDQYJKoZIhvcNAQELBQAwYjELMAkGA1UE
BhMCQ04xMjAwBgNVBAoMKUdVQU5HIERPTkcgQ0VSVElGSUNBVEUgQVUUSE9SSVRZ
IENPLixMVEQuMR8wHQYDVQQDDBZHRENBIFRydXN0QVVUSCBSNSBST09UMB4XDTE0
MTEyNjA1MTMxNV0XDTQwMTIzMTk1OVowYjELMAkGA1UEBhMCQ04xMjAwBgNV
BAoMKUdVQU5HIERPTkcgQ0VSVElGSUNBVEUgQVUUSE9SSVRZ IENPLixMVEQuMR8w
HQYDVQQDDBZHRENBIFRydXN0QVVUSCBSNSBST09UMIICIjANBgkqhkiG9w0BAQEF
AAOCAg8AMIICCGKCAgEA2aMW8Mh0dHeb7zMN0wZ+Vfy1YI92hhJCfVZmPoiC7XJj
Dp6L3TQsAlFRwxn9WVSEyFrs0yw6ehGXTjGoqcuEVe6ghWin9tsJlKCvLriXBj
TnnEt1u9ol2x8kECK62pOqPseQrsXzrj/e+APK00mxqriCZ7VqKChh/rNYmDf1+u
KU49tm7srsHwJ5uu4/Ts765/94Y9cnrrpftZTqfrlYwiOXnhLQiPzLyRuEH3FMEj
qcOtmkVEs7LXLM3GKeJQEK5cy4KOFxg2fZfmiJqwTTQJ9Cy5WmYqsBebnh52nUpm
MUHfP/vFBu8btm4aRjb3ZGM74zkYI+dndRTVdVeSN72+ahsmUPI2JgaQxXABZG12
ZuGR224HwGGALrIuL4xwp9E7PLOR5G62xDtw8mySlwnNR30YwPO7ng/Wi64HtloP
zgsMR6flPri9fcebNaBhlzpBdRfMK5Z3KPlhHtmVdiBnaM8Nvd/WHwlqmuLMc3Gk
L30SgLDTMEZeS1SZD2fJpcjyIMGC7J0R38IC+xo70e0gmu9lZJIQDSri3nDxGGeC
jGHeuLzRL5z7D9Ar7Rt2ueQ5Vfj4oR24qoAATILnsn8JuLwwoc8N9VKejveSswoA
HQBULwbgsQfZxw9cZX08bVlX5O2ljelAU58VS6Bx9hoh49pwBiFYFIeFd3mqgnkC
AwEAAANCMEEAwHQYDVR0OBBYEFOLJQJ9NzuiaoXzPDj9lxSmIahlRMA8GA1UdEwEB
/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgGGMA0GCSqGS1b3DQEBCwUAA4ICAQDRSVfg
p8xoWLoBDysZzY2wYUWsEe1jUGn4H3++Fo/9nesLqjJHdtJnJO29fDMylrHBYZm
DRd9FBUB1Ov9H5r2XpdptxolpAqzkT9fNqyL7FeoPueBihhXOYV0GkLH6VsTX4/5
COmSdl31R9KRO9b7eGZONn356ZLpBN79SWP8bfsUcZNnL0dKt7n/HipzcEYwv1ry

L3ml4Y0M2fmyYzeMN2WfCgpcWwlyua1jPLHd+PwyvzeG5LuOmCd+uh8W4XAR8gPf
JWlyJyYYMoSf/wA6E7qaTfRPuBRwIrHKK5DOKcFw9C+df/KQHtZa37dG/OaG+svg
IHZ6uqbL9XzeYqWxi+7egmaKTjowHz+Ay60nugxe19CxVsp3cbK1daFQqUBDF8Io
2c9Si1vIY9RCPqAzekYu9wogRIR+ak8x8YF+QnQ4ZXMn7sZ8uI7XpTrXmKGcjBBV
09tL7ECQ8s1uV9JiDnxXk7Gnbc2dg7sq5+W2O3FYrf3RRbxake5TFW/TRQl1brqQ
XR4EzfffHqhmsYzmlGrv/EhOdJhCrylvLmrH+33RZjEizIYAfmaDDEL0vTSSwxrq
T8p+ck0LcIymSLumoRT2+1hEmRSuqguTaaApJUqlyyvdimYHFngVV3Eb7PVHhPOe
MTd61X8kreS8/f3MboPoDKi3QWwH3b08hpcv0g==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFGjCCA2qgAwIBAgILWku9WvtPilv6ZeUwDQYJKoZIhvcNAQELBQAwwTTELMakG
A1UEBhMCQVQxIzAhBgNVBAoTGmUtY29tbWVyY2UgbW9uaXRvcmluZyBHbWJIMRkw
FwYDVQQDExBHTE9CQUxUUIVTVCAyMDIwMB4XDTIwMDIxMDAwMDAwMFoXDTQwMDYx
MDAwMDAwMFowTTELMakGA1UEBhMCQVQxIzAhBgNVBAoTGmUtY29tbWVyY2UgbW9u
aXRvcmluZyBHbWJIMRkwFwYDVQQDExBHTE9CQUxUUIVTVCAyMDIwMIIICjANBgkq
hkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAri5WrRsc7/aVj6B3GyvTY4+ETUWiD59b
RatZe1E0+eyLinjF3WuvvcTfk0Uev5E4C64OFudBc/jbu9G4UeDLgztzOG53ig9Z
YybNpyrOVPu44sB8R85gfD+yc/LAGbaKkoc1DZAoouQVBGM+uq/ufF7MpotQsjj3
QWPKzv9pj2gOITblzLmMCcpL3TGQlsjMH/1WljTbjhzqLL6FLmPdqqmV0/0plRPw
yJiT2S0WR5ARg6I6IqIoV6Lr/sCMKKCmfecqQjuCgGOIYx8ZzHyyZqjC0203b+J+
BlHZRYQfEs4kUmSFC0iAToexIiIwquuuuuAC4EDosEKAA1GqtH6qRNdDYfOiaxaJ
SaSjpCuKAsR49GiKweR6NrFvG5Ybd0mN1MkGco/PU+PcF4UgStyYJ9ORJitHHmkH
r96i5OTUawuzXnzUJIBHKWk7buis/UDr2O1xcSvy6Fgd60GXIsUf1DnQJ4+H4xj0
4KlGdFv0OoIu0G4skaMxXDtG6nsEEFZegB31pWXogvziB4xiRfUg3kZwhqG8k9Me
dKZssCz3AwylDMvUclOGvGBG85hqwvG/Q/lwIHfKN0F5VVJjVsSn8VoxlIdrPIw
q7ejMZdnrY8XD2zHc+0klGvIg5rQmjdBKuxFshsSUktq6HQjJLyQUp5ISXbY9e2
nKd+Qmn7OmMCAwEAAANjMGEwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AQYwHQYDVR0OBBYEFNwuH9FhN3nkq9XVsxJxaD1qaJwiMB8GA1UdIwQYMBaAFNwu
H9FhN3nkq9XVsxJxaD1qaJwiMA0GCSqGSIb3DQEBCwUAA4ICAQCR8EICaEDuw2jA
VC/f7GLDw56KoDEoqoOOpFaWEhCGVrqXctJUMHyGdUdaG/7FELYjQ7ztdG14wJC
XtzoRlghNQIw4Lx0SsFDKv/bGtCwr2zD/cuz9X9tAy5ZVp0tLTWMstZDFyySCstd
6IwPS3BD0IL/qMy/pJTAvoe9iuOTE8aPmxadJ2W8esVCgmxcB9CpwYhgROmYhRZf
+I/KARDOJcP5YBugxZfD0yyIMaK9MOzQ0MAS8cE54+X1+NZK3TTN+2/BT+MAi1bi
kvcoskI3ciNnxz8RFbLEAwW+uxF7Cr+obuf/WEPPm2eggAe2HcqtbePBEX4tdJP7
wry+UUTF72glJ4DjyKDUEuzZpTcdN3y0kcra1LGWge9oXHYQSa9+pTeAsRxSvTOB
TI/53WXZFM2KJVj04sWDpQmQ1GwUY7VA3+vA/MRYfg0UFodUJ25W5HCEuGwyEn6C
MUO+1918oa2u1qsgEu8KwxCMSZY13At1XrFP1U80DhEgB3VDRemjEdqso5nCtnkn
4rnvyOL2NSI6dPrFf4IFYqYK6miyeUcGbvJXqBUzxvd4Sj1Ce2t+/vdG6tHrju+I
aFvowdlxfv1k7/9nR4hYJS8+hge9+6jlgqispdNpQ80xiEmEU5LAsTkbOYMBMMTy
qfrQA71yN2BWHZz8vTmR9W0Nv3vXkg==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFVZCAAz+gAwIBAgIINAgPlk28xsBNJiGuiFzANBgkqhkiG9w0BAQwFADBHMQsw
CQYDVQQGEwJVUzEiMCAgA1UEChMZR29vZ2xIIFRydXN0IFNlcnZpY2VzIEExMQZEU

MBIGA1UEAxMLR1RTIFJvb3QgUjEwHhcNMTYwNjIyMDAwMDAwWhcNMzYwNjIyMDAw
MDAwWjBHMqswCQYDVQGEwJVUzEiMCAGA1UEChMZR29vZ2x1IFRydXN0IFNlcnZp
Y2VzIEExMQZEUUMBIGA1UEAxMLR1RTIFJvb3QgUjEwggLiMA0GCSqGSIsb3DQEBAQUA
A4ICDwAwggIKAoICAQC2EQLHuOhd5s73L+UPreVp0A8of2C+X0yBoJx9vaMf/vo
27xqLpeXo4xL+Sv2sfnohB2x+cWX3u+58qPpvBKJXqeqUqv4IyflPLGcY9vXmX7w
Cl7raKb0xlpHDU0QM+NOsROjyBhsS+z8CZDfnWQpJSMHobTSPS5g4M/SCYe7zUjw
TcLCeoiKu7rPWRnWr4+wB7CeMfGCwcDfLqZtbBkOtdh+JhpFAz2weaSUKK0Pfybl
qAj+lug8aJRT7oM6iCsVlgmy4HqMLnXWnOunVmSPlk9orj2XwoSPwLxAwAtcvfaH
szVsrBhQf4TgTM2S0yDpM7xSma8ytSmzJSq0SPly4cpk9+aCEI3oncKKiPo4Zor8
Y/kB+Xj9e1x3+naH+uzfsQ55lVe0vSbv1gHR6xYKu44LtcXFilWr06zqkUspzBmk
MiVOKvFIRNACzqrOSbTqn3yDsEB750Orp2yjj32JgfpMpf/VjsPOS+C12LOORc92
wO1AK/1TD7Cn1TsNsYqiA94xrcx36m97PtbkSIS5r762DL8EGMUUXLeXdYWk70p
aDPvOmbsB4om3xPXV2V4J95eSRQAogB/mqghtqmxlbCluQ0WEdrHbEg8QOB+DVrN
VjzRlwW5y0vtOUucxD/SVRNuJLDWcfr0wbrM7Rv1/oFB2ACYPTIrImqYNxgFIQID
AQABo0IwQDAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0TAAQH/BAUwAwEB/zAdBgNVHQ4E
FgQU5K8rJnEaK0gnhS9SZizv8IkTcT4wDQYJKoZIhvcNAQEMBAQADggIBA+jQibb
C5u+/x6Wki4+omVKapi6Ist9wTrYggoGxval3sBOh2Z5ofmmWJyq+bXmYOfg6LEe
QkEzCzc9zolzFcq1JKjPa7XSQCGYzyI0zzvFIoTgxQ6KfF2I5DUkzps+GlQebtuy
h6f88/qBVRRIclmpIguXpOLW7ttXNLwzldMXG+gnoot7TiYaelpkttGsN/H9oPM4
7HLwEXWdyzRSjeZ2axfG34arJ45JK3VmgRAhpua+9K4l/3wV3s6MJT/KYnAK9y8J
ZgflPxz88NtFMN9iiMG1D53Dn0reWVIHxYciNuaCp+0KueIHoi17eko8cdLiA6Ef
MgfdG+RCzgwARWGAtQsgWSl4vflVy2PFPEz0tv/bal8xa5meLMFrUKTX5hgUvYU/
Z6tGn6D/Qqc6f1zLXbBwHS09dR2CQzreExZBfMzQsNhFRAbd03OIozUhfJFbdt
6u9AWpQKXCbfTkBdYiJ23//OYb2MI3jSNwLgjt7REteJ9r/tSQdirpLsQBqvFAnZ
0E6yove+7u7Y/9waLd64NnHi/Hm3lCXRSNboTXns5lndcEZoitHTtNCjv0xyBZm
2tIMPNUzjsmhDYAPexZ3FL//2wmUspO8IFgV6dtxQ/PeEMMA3KqglbbC1j+Qa3bb
bP6MvPJwNQzcmRk13NfIRmPVNnGuV/u3gm3c

-----END

CERTIFICATE-----

-----BEGIN

CERTIFICATE-----

MIIFVzCCAz+gAwIBAgINAgPlrsWNBCUaqxElqjANBgkqhkiG9w0BAQwFADBHMqsw
CQYDVQGEwJVUzEiMCAGA1UEChMZR29vZ2x1IFRydXN0IFNlcnZpY2VzIEExMQZEU
MBIGA1UEAxMLR1RTIFJvb3QgUjEwHhcNMTYwNjIyMDAwMDAwWhcNMzYwNjIyMDAw
MDAwWjBHMqswCQYDVQGEwJVUzEiMCAGA1UEChMZR29vZ2x1IFRydXN0IFNlcnZp
Y2VzIEExMQZEUUMBIGA1UEAxMLR1RTIFJvb3QgUjEwggLiMA0GCSqGSIsb3DQEBAQUA
A4ICDwAwggIKAoICAQDO3v2m++zsFDQ8BwZabFn3GTxd98GdVarTzTukk3LvCvpt
nfbwhYBboUhSnznFt+4orO/LdmguUd+tAWyZH8QiHZ/+cnfgLFuv5AS/T3KgGjSY
6Dlo7JUle3ah5mm5hRm9iYz+re026nO8/4Piy33B0s5Ks40FnotJk9/BW9BuXvAu
MC6C/Pq8tBcKSOWIm8Wba96wyrQD8Nr0kLhlZPdcTK3ofmZemde4wj7I0BODre7k
RXuJVfeKH2JShBKzwcCX44ofR5GmdFrS+LFjKBC4swm4VndAoiaYecb+3yXuPuWg
f9RhD1FLPD+M2uFwdNjCaKH5wQzpoeJ/u1U8dgbuak7MkogwTZq9TwtImoS1mKPV
+3PBV2HdKFZ1E66HjucMUQkQdYhMvI35ezzUIkgfKtzra7tEscszcTJGr61K8Yzo
dDqs5xoic4DSMPclQsciOzsSrZYuxsN2B6ogtzVJV+mSSeh2FnIxZyuWfojx5RW
Ir9qS34BIbIjMt/kmkRtWVtd9QCgHJvGeJeNkP+byKq0rxFROV7Z+2et1VsRnTKa
G73VululycsLaVNVJ1zgyjbLiGH7HrfQy+4W+9OmTN6SpdT3/UGVN4unUu0kzCq
gc7dGtxRcw1PcOnlthYhGXmy5okLdWTK1au8CcEYof/UVKGFPP0UJA0yh9OktwID

AQABo0lwQDAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4E
FgQUu//KjiOfT5nK2+JopqUVJxce2Q4wDQYJKoZIhvcNAQEMBQADggIBAB/Kzt3H
vqGf2SdMC9wXmBFqiN495nFWcrKeGk6c1SuYJF2ba3uwM4IJvd8lRuqYnrYb/oM8
0mJhwQTztuDFycgTE1XnqG0tjHsB/ncw4c5omwX4Eu55MaBBRTUoCnGkJE+M3DyC
B19m3H0Q/gxhswWV7uGugQ+o+MePTagjAiZrHYNSVc61LwDKgEDg4XSsYPWHgJ2u
NmSRXbBoGOqKYcl3qJfEycel/FVL8/B/uWU9J2jQzGv6U53hkRrJXRqWbTKH7QMg
yALOWr7Z6v2yTcQvG99fevX4i8buMTolUVVnjWQye+mew4K6Ki3pHrTgSAai/Gev
HyICc/sgCq+dVEuhzf9gR7A/Xe8bVr2XIZYtCtFenTgCR2y59PYjJbigapordwj6
xLEokCZYCDzifqrXPW+6MYgKBesntaFJ7qBFVHvmJ2WZICGoo7z7GJa7Um8M7YNR
TOlZ4iBgxcJlkoKM8xAfDoqXvneCbT+PHV28SSe9zE8P4c52hgQjxcCMElv924Sg
JPFI/2R80L5cFtHvma3AH/vLrrw4IgYmZNralw4/KBVEqE8AyyCazM90arQ+POuV
7LXTWtiBmelDGDfrs7vRWGJB82bSj6p4lVQgw1oudCvV0b4YacCs1aTPObpRhANl
6WLAYv7YTVVWW4tAR+kg0Eeye7QUd5MjWHYbL

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICCTCCAY6gAwIBAgINAgPluILrIPglJ209ZjAKBggqhkJOPQQDAzBHMqSwCQYD
VQQGEwJVUzEiMCAGA1UEChMZRR29vZ2x1IFRydXN0IFNlcnZpY2VzIEExMQZEU
MBIG A1UEAxMLR1RTIFJvb3QgUjMwHhcNMTYwNjIyMDAwMDAwWhcNMzYwNjIyMDAwMDAw
WjBHMqSwCQYDVQQGEwJVUzEiMCAGA1UEChMZRR29vZ2x1IFRydXN0IFNlcnZpY2Vz
IEExMQZEU MBIG A1UEAxMLR1RTIFJvb3QgUjMwZjAQBgcqhkJOPQIBBgUrgQQAIGNi
AAQfTzOHMymKoYTeY8chWEGJ6ladK0uFxb1MJ7x/JlFyb+Kf1qPKzEUURout736G
jOyxfi//qXGdGIRFBEFVbivqJn+7kAHjSxm65FSWRQmx1WyRRK2EE46ajA2ADDL2
4CejQjBAMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQW
BBTB8Sa6oC2uhYHP0/EqEr24Cmf9vDAKBggqhkJOPQQDAwNpADBMAjEA9uEglRR7
VKOQFhG/hMjqb2sXnh5GmCCbn9MN2azTL818+FsuVbu/3ZL3pAzcMeGiAIEA/Jdm
ZuVDFhOD3cffL74UOO0BzrEXGhF16b0DjyZ+hOXJYKaV11RZt+cRLInUue4X

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICCTCCAY6gAwIBAgINAgPlwGjvYxqccpBQUjAKBggqhkJOPQQDAzBHMqSwCQYD
VQQGEwJVUzEiMCAGA1UEChMZRR29vZ2x1IFRydXN0IFNlcnZpY2VzIEExMQZEU
MBIG A1UEAxMLR1RTIFJvb3QgUjQwHhcNMTYwNjIyMDAwMDAwWhcNMzYwNjIyMDAwMDAw
WjBHMqSwCQYDVQQGEwJVUzEiMCAGA1UEChMZRR29vZ2x1IFRydXN0IFNlcnZpY2Vz
IEExMQZEU MBIG A1UEAxMLR1RTIFJvb3QgUjQwZjAQBgcqhkJOPQIBBgUrgQQAIGNi
AATzdHOnaltgrkO4NcWBMHtLSZ37wWHO5t5GvWvVYRg1rkDdc/eJkTBa6zzuhXyi
QHY7qca4R9gq55KRanPpsXI5nymfopjTX15YhmUPoYRlBtHci8nHc8iMai/lxKvR
HYqjQjBAMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQW
BBSATNbrdP9JNqPV2Py1PsVq8JQdjDAKBggqhkJOPQQDAwNpADBMAjEA6ED/g94D
9J+uHXqnLrmvT/aDHQ4thQEd0dlq7A/Cr8deVl5c1RxYIigL9zc2L7F8AjEA8GE8
p/SgguMh1YQdc4acLa/KNJvxn7kjNuK8YAODgLOaVsjs4rsUecrNIdSUtUId

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIB3DCCAYOGAwIBAgINAgPlfvU/k/2lCSGypjAKBggqhkJOPQQDAjBQMSQwIgYD

VQQLExtHbG9iYWxTaWduIEVDQyBSb290IENBIC0gUjQxEzARBgNVBAoTCkdsb2Jh
bFNpZ24xEzARBgNVBAMTCkdsb2JhbFNpZ24wHhcNMTIxMTEzMDAwMDAwWhcNMzgw
MTE5MDMxNDA3WjBQMSQwIgYDVQQLExtHbG9iYWxTaWduIEVDQyBSb290IENBIC0g
UjQxEzARBgNVBAoTCkdsb2JhbFNpZ24xEzARBgNVBAMTCkdsb2JhbFNpZ24wWTAT
BgcqhkJOPQIBBgqhkJOPQMBBwNCAAS4xnnTj2wlDp8uORkcA6SumuU5BwkWymOx
uYb4ilfBV85C+nOh92VC/x7BALJucw7/xyHlGKSq2XE/qNS5zowdo0IwQDAOBgNV
HQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUVLB7rUW44kB/
+wpu+74zyTyjhNUwCgYIKoZIZj0EAwIDRwAwRAIgIk90crlgr/HmnKAWBVBfw147
bmF0774BxL4YSFlhglICICadVGNA3jdgUM/I2O2dgq43mLyjj0xMqTQrbO/7lZsm
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICHjCCAAaSgAwIBAgIRYFIJ4CYuu1X5CneKcflK2GwwCgYIKoZIZj0EAwMwUDEk
MCIGA1UECxMbr2xvYmFsU2lnbiBFQ0MgUm9vdCBDQSAAtIFI1MRMwEQYDVQQKEwpH
bG9iYWxTaWduMRMwEQYDVQQDEwpHbG9iYWxTaWduMB4XDTEyMTEzMDAwMDAwMFoX
DTM4MDExOTAzMTQwN1owUDEkMCIGA1UECxMbr2xvYmFsU2lnbiBFQ0MgUm9vdCBD
QSAAtIFI1MRMwEQYDVQQKEwpHbG9iYWxTaWduMRMwEQYDVQQDEwpHbG9iYWxTaWdu
MHYwEAYHKoZIzj0CAQYFK4EEACIDYgAER0UOlvt9Xb/pOdEh+J8LttV7HpI6SFkc
8GIxLcB6KP4ap1yztSYX50XUWPrRd21DosCHZTQKH3rd6zwcWdTaRvQZU4f8ke
hOvRnkmSh5SHDDqFSmafVnTTZdhBoZKo0IwQDAOBgNVHQ8BAf8EBAMCAQYwDwYD
VR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUPEyPjSjvqB8ohREom3m7e0oPQn1kwCgYI
KoZIZj0EAwMDAAAwZQIXAOVpEsLu28YxuglB4Zf4+/2a4n0Sye18ZNPLBSWLvtmg
515dTguDnFt2KaAJJiFqYgIwcdK1j1zqO+F4CYWodZI7yFz9SO8NdCKoCOJuxUnO
xwy8p2Fp8fc74SrL+SvzZpA3
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDXzCCAkegAwIBAgILBAAAAAABIVhTCKIwDQYJKoZIhvcNAQELBQAwtDEgMB4G
A1UECxMXR2xvYmFsU2lnbiBSb290IENBIC0gUjMxEzARBgNVBAoTCkdsb2JhbFNp
Z24xEzARBgNVBAMTCkdsb2JhbFNpZ24wHhcNMDEwMzE4MTAwMDAwWhcNMjkwMzE4
MTAwMDAwWjBMMSAwHgYDVQQLEXdHbG9iYWxTaWduIFJvb3QgQ0EgLSBSMzEtMBEG
A1UEChMKR2xvYmFsU2lnbjETMBEGA1UEAxMKR2xvYmFsU2lnbjCCASlwDQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBAMwldpB5BngiFvXAg7aEyiie/QV2EcWtiHL8
RgJDx7KKnQRfJMsuS+FggkbhUqsMgUdwbN1k0ev1LKMPgj0MK66X17YUhhB5uzsT
gHeMCOFJ0mpiLx9e+pZo34knltifBtc+ycsmWQ1z3rDI6SYOgxXG71uL0gRgykmm
KPZpO/bLyCiR5Z2KYVc3rHQU3HTgOu5yLy6c+9C7v/U9AOEGM+iCK65TpjoWc4zd
QQ4gOsC0p6Hpsk+QLjJg6VfLuQSSaGjLOCZgdbKfd/+RFO+uIE8rUAVSNECMWEZ
XriX7613tSaer9fwrPvm2L7DWzgVGkVwQPabumDk3F2xmmFghcCAwEAAaNCMEAw
DgYDVR0PAQH/BAQDAgEGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFI/wS3+o
LkUkrk1Q+mOai97i3Ru8MA0GCSqGSIb3DQEBCwUAA4IBAQLQNvAUKr+yAzv95ZU
RUm7lgAJQayzE4aGKAczymvmdLm6AC2upArT9fHxD4q/c2dKg8dEe3jgr25sbwMp
jjM5RcOO5LIXbKr8EpbsU8Yt5CRsuZRj+9xTaGdWPoO4zzUhw8lo/s7awlOqzJCK
6fBdRoyV3XpYKBovHd7NADdBj+1EbddTKJd+82cEHhXXipa0095MJ6RMG3NzdvQX
mcIfeg7jLQitChws/zyrVQ4PkX4268NXSb7hLi18YIvDQVETI53O9zJrlAGomecs
Mx86OyXShkDOOyyGeMlhLxS67ttVb9+E7gUJTb0o2HLO02JQZR7rkpeDMdmztcPH
WD9f
-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFgzCCA2ugAwIBAgIORea7A4Mzw4VISOB/RVEwDQYJKoZIhvcNAQEMBAwTDEg
MB4GA1UECxMXR2xvYmFsU2lnbiBSb290IENBIC0gUjYxEzARBgNVBAoTCkdsb2Jh
bFNpZ24xEzARBgNVBAMTCkdsb2JhbFNpZ24wHhcNMTQxMjEwMDAwMDAwWhcNMzQx
MjEwMDAwMDAwWjBMMSAwHgYDVQQLExdHbG9iYWxTaWduIFJvb3QgQ0EgLSBSNjET
MBEGA1UEChMKR2xvYmFsU2lnbjETMBEGA1UEAxMKR2xvYmFsU2lnbjCCAiIwDQYJ
KoZIhvcNAQEBBQADggIPADCCAgoCggIBAJUH6HPKZvnsFMp7PPcNCPG0RQssgrRI
xutbPK6DuEGSMxSkb3/pKszGsIhrxbaJ0cay/xTOURQh7ErdG1rG1ofuTTToVBu1k
ZguSgMpE3nOUTvOniX9PeGMiYBJQbUJmL025eShNUhqKGoC3GYEOfsSKvGRMIRxD
aNc9PIrFsmBvKJq3MQbFvuJtMgamHvm566qjuL++gmNQ0PAYid/kD3n16qIfKtJw
LnnvJO7bVPiSHyMEAc4/2ayd2F+4OqMPKq0pPbzUoSB239jLKJz9CgYXfIWHSw
1CM69106yqLbnQneXUQtPGbZVeS+n68UARjNN9rkxi+azayOeSsJDa38O+2HBNX
k7besvjihbdzorg1qkXy4J02oW9UivFyVm4uiMVRQkQVlO6jxTiWm05OWgtH8wY2
SXcwwHE35absIQh1/OZhFj931dmRl4QKbNQCTXTAFO39OfuD8l4UoQSwC+n+7o/h
bguyCLNhZglqsQY6ZZZZwPA1/cnaKI0aEYdwgQqomnUdnjqGBQCe24DWJfncBZ4n
WUx2OVvq+aWh2IMP0f/fMBH5hc8zSPXKbWQULHpYT9NLCEnFIWQaYw55PfWzjMpY
rZxCRXluDocZXSxZba/jJvcE+kNb7gu3GduyYsRtYQUigAZcIN5kZeR1Bonvzce
MgfYFGM8KEyvAgMBAAGjYzBhMA4GA1UdDwEB/wQEAwIBBjAPBgNVHRMBAf8EBTAD
AQH/MB0GA1UdDgQWBBSUBAWjKxPioufi1xzWx/B/yGdToDAfBgNVHSMEGDAWgBSu
bAWjKxPioufi1xzWx/B/yGdToDANBgkqhkiG9w0BAQwFAAOCAGeAgyXt6NH9IVLN
nsAEoJFp5lzQhN7craJP6Ed41mWYqVuoPid8AorRbrcWc+ZfwFSY1XS+wc3iEZGt
Igx93eFyRJa0IV7Ae46ZeBZDE1ZXs6KzO7V33EByrKPrmzU+sQghoefEQzd5Mr61
55wsTLxDKZmOMNOsIeDjHfrYBzN2VAAiKrlNIC5waNrlU/yDXNOd8v9EDERm8tLj
vUYAGm0CuiVdjaExUd1URhxN25mW7xocBFymFe944Hn+Xds+qkxV/ZoVqW/hpvpf
cDDpw+5CRu3CkwWJ+n1jez/QcYF8AOiYrg54NMML+68KnyBr3TsTjxKM4kEaSHpz
oHdpX7Zcf4LIHv5YGygrqGytXm3ABdJ7t+uA/iU3/gKbaKxCXcPu9czc8FB10jZp
nOZ7BN9uBmm23goJSFmH63sUYHpkqmlD75HHTOwY3WzvUy2MmeFe8nl+z1TivWfs
pA9MRf/TuTAjB0yPEL+GltmZWrsZVxykzLsViVO6LAUP5MSeGbEYNNVMnbrt9x+v
JJUEeKgDu+6B5dpffltKoZB0JaezPkvILFa9x8jvOOJckvB595yEunQtYQEgfn7R
8k8HWV+LLUNS60YMI0H1Zkd5d9VUWx+tJDfLRVpOoERiYNiwmcUVhAn21klJwGW4
5hpxbqCo8YLoRT5s1gLXCmeDBVrJpBA=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICCzCAZGgAwIBAgISEdK7ujNu1LzmJGjFDYQdmOhDMAoGCCqGSM49BAMDMEYx
CzAJBgNVBAYTAkFMRkwFwYDVQQKExBHbG9iYWxTaWduIG52LXNhMRwwGgYDVQQD
ExNHbG9iYWxTaWduIFJvb3QgRTQ2MB4XDTE5MDMyMDAwMDAwMFoXDTE5MDMyMDAw
MDAwMFowRjELMAkGA1UEBhMCQkUxGTAXBgNVBAoTEEdsb2JhbFNpZ24gbnYtc2Ex
HDAaBgNVBAMTE0dsb2JhbFNpZ24gUm9vdCBFNDYwdjAQBgcqhkiOPQIBBgUrgQQA
IgNiAAAScDrHPt+ieUnd1NPqlRqetMhkytAepJ8qUuwzSchDH2omwlwxwEwkBjtjq
R+q+soArzfwoDdusvKSGN+1wCAB16pMLey5SnCNoIwZD7JiVU4Tb+0cUB+hflGdd
yXqBPCCjQjBAMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MB0GA1Ud
DgQWBbBQxCPcPtsad0kRLgLWi5h+xEk8blTAKBggqhkiOPQIDAwwNoADBIAjEA31SQ
7Zvvi5QCkxeCmb6zniz2C5GMn0oUsfZkvLtoURMMA/cVi4RguYv/Uo7njLwcAjA8
+RHUjE7AwWHCFUyqqx0LMV87HOIAloQx5v5zli/altP+CAezNI8BZ/3Hobui3A=

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFWjCCA0KgAwIBAgISEdK7udcjGJ5AXwqdLdDfJWfRMA0GCSqGSIb3DQEBAUA
MEYxChAJBgNVBAYTAkFMRkwFwYDVQQKExBHbG9iYWxTaWduIG52LXNhMRwwGgYD
VQQDExNHbG9iYWxTaWduIFJvb3QgUjQ2MB4XDTE5MDMyMDAwMDAwMFoXDTQ2MDMy
MDAwMDAwMFowRjELMAkGA1UEBhMCQkUxGTAXBgNVBAoTEEdsb2JhbFNpZ24gbnYt
c2ExHDAaBgNVBAMTE0dsb2JhbFNpZ24gUm9vdCBSNDYwggLiMA0GCSqGSIb3DQEB
AQUAA4ICDwAwggIKAoICAQCsrHQy6LNl5brtQyYdpokNRbopiLKkHWPd08EsCVeJ
OaFV6Wc0dwxu5FUdUiXSE2te4R2pt32Jm18Nnp8semNgQB+msLZ4j5lUlgHYruQG
vGIFaHa/r6gjA7aUD7xubMLL1aa7DOn2wQL7ld5m3RerdELv8HQvJfTqa1VbkNud
316HCkD7rRlr+/fKYIje2sGP1q7Vf9Q8g+7XFkyDRTNrJ9CG0Bwta/OrffGFqfUo
0q3v84RLHif8E6M6cqJaESvWJ3En7YEtbWaBkoe0G1h6zD8K+kZPTXhc+Ctl4wSE
y132tGqzZfxCnlEmIyDLPRt5ge1lFgBPGmSXZgjPjHvjK8Cd+RTyG/FWaha/LIWF
zXg4mutCagI0GIMXTpRW+LaCtfOW3T3zvn8gdz57GSNrLNRyc0NXfeD412IPFzYE
+cCQYDdF3uYM2HSNrpYibXRdQr4G9dlkgbIQRImwTDsHTUB+JMWKmlJ5jqSngiCN
l/onccnfxkF0oE32kRbcRoxfKWMxWxEM2G/CtjJ9++ZdU6Z+Ffy7dXxd7Pj2Fxs
x2sZy/N78CsHpdlseVR2bJ0cpm4O6XkMqCNqo98bMDGfsVR7/mrLZqrcZdCinkqa
ByFrgY/bxFn63iLABJzjqls2k+g9vXqhnQt2sQvHnf3PmKgGwvqo6GDoLclcqUC
4wIDAQABO0IwQDAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNV
HQ4EFgQUA1yrc4GHqMywptWU4jaWSf8FmSwwDQYJKoZIhvcNAQEMBQADggIBAHx4
7PYCLltbfpIrXTncvtgdokIzTfnvpCo7RGkerNlFo048p9gkUbJUHJNOxO97k4Vg
JuoJSOD1u8fpaNK7ajFxzHmuEajwfmf3IH7wvqMxX63bElazHU1VNaL8FpO7XJqti
2kM3S+LGteWygxk6x9PbTZ4IevPuzz5i+6zoYMzRx6Fcg0XERczzF2sUyQQCPtlk
pnnpHs6i58FZFZ8d4kuaPp92CC1r2LpXFNqD6v6MVenQTqnMdzGxRBF6XLE+0xRF
FRhiJBPSy03OXIPBNvIQ6IbbjhVp+J3pZmOUdkLG5NrmJ7v2B0GbhWrJKsFjLt
rWhV/pi60zTe9Mlhw6G9kuEYO4Ne7UyWHmRVSyBQ7N0H3qqJZ4d16GLuc1CLgSk
ZoNNiTW2bKg2SnkheCLQQRzRQDGQob4Ez8pn7fXwgNNgyYMqIgXQBztSvwyeqiv5
u+YfjyW6hY0XHgL+XVAEV8/+LbzvXMAaq7afJMbfc2hIkCwU9D9SGuTSyxTDYWnP
4vkYxboznxSjBF25cfe1lNj2M8FawTSLfJvdkzrnE6JwYZ+vYxXX4M2bUdGc6
N3ec592kd3ZDZopD8p/7DEJ4Y9HiD2971KE9dJeFt0g5QdYg/NA6s/rob8SKunE3
vouXsXgxT7PntgMTzlSdriVZzH81Xwj3QEUxCp6

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDxTCCAq2gAwIBAgIBADANBgkqhkiG9w0BAQsFADCBBgZELMAkGA1UEBhMCVVMx
EDAOBgNVBAgTB0FyaXpvbmExEzARBgNVBAcTClnjb3R0c2RhbgUxGjA5YBgNVBAoT
EUdvRGFkZHkuY29tLCBjbmuMTEwLwYDVQQDEyhHbyBEYWRkeSBSb290IENlcnRp
ZmljYXRlIEF1dGhvcml0eSAteCycMB4XDTA5MDkwMTAwMDAwMFoXDTM3MTIzMTIz
NTk1OVowgYmxCzAJBgNVBAYTAiVTRMAwDgYDVQQIEwdBcm16b25hMRMwEQYDVQQH
EwpTY290dHNkYWxIMRowGAYDVQQKEExFHb0RhZGR5LmNvbSwgSW5jLjExMC8GA1UE
AxMoR28gRGFkZHkgUm9vdCBDZXJ0aWZpY2F0ZSBDbXR0b3JpdHkgLSBHMjCCASIw
DQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAL9xYgix+lk09xvJGKP3gEly6SKD
E6bFIEMBO4Tx5oVJnyfq9oQbTqC023CYxzIBsQU+B07u9PpPL1kwIuerGVZr4oAH
/PMWdYA5UXvl+TW2dE6pjYIT5LY/qQOD+qK+ihVqf94Lw7YZFAXK6sOoBJQ7Rnwy
DfMAZiLiJWltNowRGLfTshxgtDj6AozO091GB94KPutdfMh8+7ArU6SSYmlRJQVh
GkSBjCypQ5Yj36w6gZoOKcUcqeldHraenjAKOc7xiID7S13MMuyFYkMlNAJWJwGR

MAMBAf8wHQYDVR0OBBYEFAPiI6ZgpJIKM+qTW8VX6iVNvRLuMA4GA1UdDwEB/wQE
AwIBhJANBgkqhkiG9w0BAQsFAAOCAGeAPpBIqm5iFSVmewzVjIuJndftTgfvnNAU
X15QvWiWkKQUEapobQk1OUAJ2vQJLDSle1mESSmXdMgHHkdt8s4cUCbjnj1AUz/3
f5Z2EMVGpdAgS1D0NTsY9FVqQRtHBmg8uwkIYtlfVUKqrFOFrJVWNlar5AWMxaja
H6NpvVMPxP/cyuN+8kyIhkdGGvMA9YCRotxDQpSbIPDRzbLrLFPCU3hKTwsUQZqP
JzLB5UkZv/HywouoCjkkLR9YjYsTewfM7Z+d21+UPCfDtcRj88YxeMn/ibvBZ3P
zzfF0HvaO7AWHaw6k9a+F9sPPg4ZeAnHqQJyIkV3N3a6dcSFA1pj1bF1BcK5vZSt
jBWZp5N99sXzqnTPBIWUmAD04vnKJGW/4GKvyMX6ssmeVkjaf2WdhW+o45WxLM0
/L5H9MG0qPzVMiHo7suuyWPEdr6sOBjHxlzPrjoiUevRi7PzKzMHVIf6tLITe7pT
BGIBnfHAT+7hOtSLIBD6Alfm78ELt5BGnBkpjNxvoEppaZS3JGWg/6w/zgH7IS79
aPib8qXPMThcFarmlwDB31qlpzmq6YR/PFGGoOtmUW4y/Twhx5duoXNTSpv4Ao8YW
xw/ogM4cKGR0GQjTQuPOAF1/sdwTsOEFy9EgqoZ0njnnkf3/W9b3raYvAwtt41dU
63ZTGIOrmLo=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICwzCCAkqgAwIBAgIBADAKBggqhkiOPQQDAjCBqjELMAkGA1UEBhMCR1Ix DzAN
BgNVBACtBkF0aGVucZFEMEIGA1UEChM7SGVsbGVuaWMgQWNhZGVtaWMgYW5kIFJl
c2VhcmNoIEluc3RpdHV0aW9ucyBDZXJ0LiBBdXR0b3JpdHkxRDBCBgNVBAMTO0hl
bGxlbmljIEFjYWRLbWljIGFuZCBSZXNlYXJjaCBJbnN0aXR1dGlbnMgRUNDIFJv
b3RDQSAyMDE1MB4XDTE1MDcwNzEwMzcwMloXDTQwMDYzMDEwMzcwMlowgaOxCzAJ
BgNVBAYTAkdSMQ8wDQYDVQQHEwZBdGhlbnMxRDBCBgNVBAoTO0hlbGxlbmljIEFj
YWRLbWljIGFuZCBSZXNlYXJjaCBJbnN0aXR1dGlbnMgQ2VydC4gQXV0aG9yaXR5
MUQwQgYDVQQDEztlZWxsZW5pYyBBY2FkZW1pYyBhbmQgUmVzZWYyY2ggSW5zdGl0
dXRpb25zIEVDQyBSb290Q0EgMjAxNTB2MBAGByqGSM49AgEGBSuBBAAiA2IABJKg
QehLgoRc4vgxEZmGZE4JJS+dQS8KjrVPdJWYUWRrjWvmP3CV8AVER6ZyOFB2lQJa
jq4onvktPtnvLEhvTCUp6NFxW98dwXU3tNf6e3pCnGoKVlp8aQuqgAkkbH7BRqNC
MEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFLLQI
C4KZJAEOnLvkDv2/+5cgk5kqMAoGCCqGSM49BAMCA2cAMGQCMGfOFmI4oqxiRaep
lSTAGiecMjvAwNW6qef4BENThe5SId6d9SWDPp5YSy/XZxMOIQIwBeF1Ad5o7Sof
TUwJCA3sS61kFyjndc5FZXIhF8siQQ6ME5g4mlRtm8rifOoCWCKR

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIGCzCCA/OgAwIBAgIBADANBgkqhkiG9w0BAQsFAADCbPjELMAkGA1UEBhMCR1Ix
DzANBgNVBACtBkF0aGVucZFEMEIGA1UEChM7SGVsbGVuaWMgQWNhZGVtaWMgYW5k
IFJlc2VhcmNoIEluc3RpdHV0aW9ucyBDZXJ0LiBBdXR0b3JpdHkxQDA+BgNVBAMT
N0hlbGxlbmljIEFjYWRLbWljIGFuZCBSZXNlYXJjaCBJbnN0aXR1dGlbnMgUm9v
dENBIDIwMTUwHhcNMTUwNzA3MTAxMTIxWhcNNDANjMwMTAxMTIxWjCBPjELMAkG
A1UEBhMCR1Ix DzANBgNVBACtBkF0aGVucZFEMEIGA1UEChM7SGVsbGVuaWMgQWNh
ZGVtaWMgYW5kIFJlc2VhcmNoIEluc3RpdHV0aW9ucyBDZXJ0LiBBdXR0b3JpdHkx
QDA+BgNVBAMTN0hlbGxlbmljIEFjYWRLbWljIGFuZCBSZXNlYXJjaCBJbnN0aXR1
dGlbnMgUm9vdENBIDIwMTUwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoIC
AQDC+Kk/G4n8PDwEXT2QNrCRONk8ZlrvtKBSRq0t89/TSNTt5AA4xMqKKYx8ZEA
4yjsriFBzh/a/X0SWwGDD7mwX5nh8hKDgE0Gpt+sr+ehiGsxr/CL0BgzuNtFajT0
AoAkKAoCFZVedioNmToUW/bLy1O8E00BiDeUJRtCvCLYjqOWXjrZMts+6PAQZe10

4S+nfK8nNLspfZu2zwnI5dMK/IhlZXQK3HMcXM1AsRzUtoSMTFDPaI6oWa7CJ06C
ojXdFPQf/7J31Ycvqm59JCFnxssm5uX+Zwdj2EUN3TpZZTIYepKZcj2chF6IIBjV
9Cz82XBST3i4vTwri5WY9bPRaM8gFH5MXF/ni+X1NYEZN9cRCLdmvtNKzoNXADrD
gfgXy5I2XdGj2HUb4Ysn6npIQf1FGQatJ5lOwXBH3bWfgVMS5bGMSF0xQxfjjMZ6
Y5ZLKTBOhE5iGV48zpeQpX8B653g+IuJ3SWYPZK2fu/Z8VFRfS0myGlZYeCsargq
NhEEelC9MoS+L9xy1dcdFkfkR2YgP/SWxa+OAXqlD3pk9Q0Yh9muiNX6hME6wGko
LfiNaFGq46V3xqSQDqE3izEjR8EJC0tu93ib14L8hCCZSRm2Ekax+0VVfQmjZayc
Bw/qa9wfLgZy7IaIEuQt218FL+TwA9MmM+eAws1CoRc0CwIDAQABo0IwQDAPBgNV
HRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUcRVnyMjJvXVd
ctA4GGqd83EkVAswDQYJKoZIhvcNAQELBQADggIBAHW7bVRLqhBYRjTyYtcWNlOI
XtVsyle9tC5G8jH4fOpCtZMWVdyhDBKg2mF+D1hYc2Ryx+hFjtyp8iY/xnmMsVMI
M4GwVhO+5lFc2JsKT0ucVIMC6U/2DWDqTUJV6HwbISHTGzrMd/K4kPFox/la/vot
9L/J9UUbzjgQKjeKeaO04wlshYaT/4mWJ3iBj2fjRnRUjtkNaeJK9E10A/+yd+2V
Z5fkscWrv2oj6NSU4kQoYsRL4vDY4ilrGnB+JGGTe08DMiUNRSQrlrRGar9KC/ea
j8GsGsVn82800vpzY4zvFrCopEYq+OsS7HK07/grfoxSwIuEVpKvPuNVqNxmsdnh
X9izjFk0WaSrT2y7HxjbdavYy5LNIIdhhDgcGH0tGEPEVvo2FXDtKK4F5D7Rpn0lQ
l033DIzdwJVqwbDG2jJ9SrcR5q+ss7FJej6A7na+RZukYT1HCjI/CbM1xyQVqdf
bzoEvM14iQuODy+jqk+iGxI9FghAD/FGTNeqewjBCvVtJ94Cj8rDtSvK6evIIVM4
pcw72Hc3MKJP2W/R8kCtQXoXxdZKNYm3QdV8hn9VTYNKpXMgWdqvKPGaJI7ZjnHK
e7iG2rKPMt4dEw0SEe7Uq/DpFXYC5ODfqiAeW2GFZECpkJcNrVPSWh2HagCXZWK0
vm9qp/UsQu0yrbYhnr68
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFajCCA1KgAwIBAgIQld2szmKXlKFD6LDNdmpeYDANBgkqhkiG9w0BAQsFADBP
MQswCQYDVQQGEwJUVzEjMCEGA1UECgwaQ2h1bmdod2EgVGVsZWNVbSBDby4sIEx0
ZC4xGzAZBgNVBAMMEkhpUEtJIFJvb3QgQ0EgLSBHMTAEFw0xOTAyMjIwOTQ2MDRa
Fw0zNzEyMzExNTU5NTlaME8xCzAJBgNVBAYTAiRlXMSMwIQYDVQQKDBpDaHVuZ2h3
YSBUZWx1Y29tIENvLiwgTHRkLjEibMBkGA1UEAwwSSG1QS0kgUm9vdCBDQSAIEcx
MIICljANBgkqhkiG9w0BAQEFAAOCAg8AMIICCGKCAgEA9B5/UnMyDHPkvRN0o9Qw
qNCuS9i233VHZvR85zkEHmpwINJaR3JnVfSl6J3VHiGh8Ge6zCFovkRTv4354twv
Vcg3Px+kwJyz5HdcoEb+d/oaoDjq7Zpy3iu9lFc6uux55199QmQ5eiY29yTw1S+6
lZgRZq2XNdZ1AYDgr/SEYYwNHl98h5ZeQa/rh+r4XfEuiAU+TCK72h8q3VJGZDnz
Qs7ZngyzsHeXZJzA9KMuH5UHsBffMNSAgJZMoYFL3QRtU6M9/Aes1MU3guvklQgZ
KILSQjqj2FPseYlSGDIcpJQ3AOPgz+yQlda22rpEZfdhSi8MEyr48KxRURHH+CK
FgeW0iEPU8DtqX7UTuybCeyvQqww1r/REEXgphaypcXTT3OUM3ECOWqj1jOXTyFj
HluP2cFeRXF3D4FdXyGarYPM+l7WjSNfGz1BryB1ZlpK9p/7qxj3ccC2HTHSOyDr
y+K49a6SsvfhfEvyovKTmiKe0xRvNlS9H15ZFblzqMF8b3ti6RZsR1pl8w4Rm0bZ
/W3c1pzAtH2lsN0/Vm+h+fbkEkj9Bn8SV7apI09bA8PgcSojt/ewsTu8mL3WmKgM
a/aOEmem8rJY5AIJEzypuxC00jBF8ez3ABHfZfjcK0NVvxaXxA/VLGGEqnKG/uY6
fsI/fe78LxQ+5oXdUG+3Se0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAdBgNV
HQ4EFgQU8ncX+l6o/vY9cdVouSlGDDjYr7AwDgYDVR0PAQH/BAQDAgGGMA0GCSqG
SIb3DQEBCwUAA4ICAQBQUfB13HAE4/+qddRxosuej6ip0691x1TPOhwEmSKsxBH
7zNKpiMdDg1H2DfHb680f0+BazVP6XKlMeJ45/dOlBhbQH3PayFUhuaVevvGyuqc
SE5XCv0vrPslTJczWNWseanMX/mF+lLJfjRfOs6DRfQUJ748JzjkZ4Bjgs6Fza
ZsT0pPBWGTMPwMWSBUdGSquEwx4noR8RkpkndZMPvDY711ePJlsMu5wP1G4wB9Tc
XzZoZjmDlicmisjEO6aIW/Vcobpf2LlI07QJNBAsNB1CI69aO4I1258EHBGG3zg

iLKecoaZAeO/n0kZtCW+VmWuF2PIHt/o/0elv+EmBYTksMCv5wiZqAxeJoBF1Pho
L5aPruJKHJwWDBNvOif2u8g0X5IDUXlwpt/L9ZlNec1OvFefQ05rLisY+GpzjLrF
Ne85akEez3GoorKGB1s6yeHvP2UEgEcyRHCVTjFnanRbEEV16rCf0OY1/k6fi8wr
kkVbbiVghUbN0aqwdmaTd5a+g744tiROJgvM7XpWGuDpWsZkrUx6AEhEL7lAuxM+
vhV4nYWBSipX3tUZQ9rbyltHhoMLP7YNdnhzeSJesYAfz77RP1YQmCuVh6EfnWQU
YDksswBVLuT1sw5XxJFBAJw/6KXf6vb/yPCtbVKoF6ubYfwSUTXkJf2vqmqGOQ==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFzzCCA7egAwIBAgIUCBZfikyl7ADJk0DfxMauI7gcWqQwDQYJKoZIhvcNAQEL
BQAwbzELMAkGA1UEBhMCSEsxEjAQBgNVBAgTCUhvbmcgS29uZzESMBAGA1UEBxMJ
SG9uZyBLb25nMRYwFAyDVQQKEw1lb25na29uZyBQb3N0MSAwHgYDVQQDExdlb25n
a29uZyBQb3N0IFJvb3QgQ0EgMzAeFw0xNzA2MDMwMjI5NDZaFw00MjA2MDMwMjI5
NDZaMG8xCzAJBgNVBAYTAkhLMRIwEAYDVQQIEw1lb25nIetvbmcxEjAQBgNVBAcT
CUhvbmcgS29uZzEWMBQGA1UEChMNSG9uZ2tvmcGU9zdDEgMB4GA1UEAxMXSG9u
Z2tvmcGU9zdCBSb290IENBIDMwggLiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIK
AoICAQCziNfqzg8gTr7m1gNt7ln8wlffKWihgw4+aMdoWJwcYEuJQwy51BWY7sFO
dem1p+/l6TWZ5Mwc50tfjTMwIDNT2aa71T4Tjukfh0mtUC1Qyhi+AViiE3CWu4mI
VoBc+L0sPOFMV4i707mV78vH9toxdlCim5lSJ9UExyUumGs2C4HDAOym71QP1mbpV
9WTRYA6ziUm4ii8F0oRfKHypaFASepwLtVPLwpgchKOesL4jpNrcyCse2m5FHomY
2vkALgbpDDtw1VAliJnLzXNg99X/NWfFobxeq81KuEXryGgeDQ0URhLj0mRiikKY
vLTGCAj4/ahMZJx2Ab0vqWwzD9g/KLg8aQFChn5pwckGyuV6RmXpwtZQQS4/t+Tt
bNe/JgERohYpSms0BpDsE9K2+2p20jzt8NYt3eEV7KObLyzJPivkaTv/ciWxNoZb
x39ri1UbSsUgYT2uy1DhCDq+sI9jQVMwCFk8mB13umOResoQUGC/8Ne8lYePl8X+
l2oBlKN8W4UdKjk60FSh0Tlxnf0h+bV78OLgAo9uliQLKAeLKjEiafv7ZkGL7YK
TE/bosw3Gq9HhS2KX8Q0NEwA/RiTzXPRN+ZlItSgXvd7GYYKecsAyVKvQv83j+Gj
Hno9UKtjBucVtT+2RTEUN7F+8kjDf8V1/peNRY8apxpyKBpADwIDAQABO2MwYTAP
BgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBBjAfBgNVHSMEGDAWgBQXnc0e
i9Y5K3DTXNSguB+wAPzFYTAdBgNVHQ4EFgQUF53NHovWOSTw01zUoLgfsAD8xWEw
DQYJKoZIhvcNAQELBQADggIBAfbVe27mIghSQpsY1Q7XZiNc4/6gx5LS6ZStS6LG
7BJ8dNVI0lkUmcDrudHr9EgwW62nV3OZqdPlt9EuWSRY3GguLmLYauRwCy0gUCCk
MpXRAJi70/33MvJJrsZ64Ee+bs7Lo3l6LWldy8joRTnU+kLBEUx3XZL7av9YROXr
gZ6voJmtvqkBZss4HTzfQx/0TW60uhdG/H39h4F5ag0zD/ov+BS5gLNdTaqX4fnk
GMX41TiMjz98iji7lpJiCzfeT2OnpA8vUFKOt1b9pq0zj8lMH8yfaIDINDceqFS
3m6TjRgm/VWsvY+b0s+v54Ysyx8Jb6NvqYTUc79NoXQbTiNg8swOqn+knEwlqLJm
Ozj/2ZQw9nKEvmhVEA/GcywWaZMH/rFF7buiVWqw2rVKAiUnhde3t4ZEFolsgCs+
l6mc1X5VTMbeRRAC6uk7nwNT7u56AQIWeNTowr5GdogTPyK7SBIdUgC0An4hGh6c
JfTzPV4e0hz5sy229zdcxsshTrD3mUcYhcErulWuBurQB7Lcq9CClnXO0ID+mefP
L5/ndtFhKvshuzHQqp9HpLIiyhY6UFfEW0NnxWViA0kB60PZ2Pierc+xYw5F9KBa
LJstxabArahH9CdMOA0uG0k7UvToiIMrVCjU8jVStDKDYmlkJGcn5fqdBb9HxEG
mpv0
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFazCCA1OgAwIBAgIRAIQz7DSQONZRGpGu2OCiwAwDQYJKoZIhvcNAQELBQAw
TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNlY3VyaXR5IFJlc2Vh

cmNoIEdyb3VwMRUwEwYDVQQDEwxJU1JHIFJvb3QgWDEwHhcNMTUwNjA0MTEwNDM4
WhcNMzUwNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJu
ZXQgU2VjdXJpdHkgUmVzZWZyY2ggR3JvdXAxFtATBgNVBAMTDElTUkcgUm9vdCBY
MTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAK3oJHP0FDfzm54rVygC
h77ct984kIxuPOZXoHj3dcKi/vVqbvYATyjb3miGbESTtrFj/RQSa78f0uoxmyF+
0TM8ukj13Xnfs7j/EvEhmkvBioZxaUpmZmyPfxwv60pIgbz5MDmgK7iS4+3mX6U
A5/TR5d8mUgU+g4rk8Kb4Mu0U1XjIB0ttov0DiNewNwIRt18jA8+o+u3dpjq+sW
T8KOEUt+zwvo/7V3LvSye0rgTBILDHCNAymg4VMk7BPZ7hm/ELNKjD+Jo2FR3qyH
B5T0Y3HsLuJvW5iB4YlcNHlsdu87kGJ55tukmi8mxdAQ4Q7e2RCOFvu396j3x+UC
B5iPNgiV5+I3lg02dZ77DnKxHZu8A/IJBdiB3QW0KtZB6awBdpUKD9jf1b0SHzUv
KBds0pjBqAlkd25HN7rOrFleaJ1/ctaJxQZBKT5ZPt0m9STJEadao0xAH0ahmbWn
OlFuhjuefXKnEgV4We0+UXgVCwOPjdAvBbI+e0ocS3MFEvzG6uBQE3xDk3SzynTn
jh8BCNAw1FtxNrQHusEwMFxIt4I7mKZ9YIqioymCzLq9gwQbooMDQaHWBfEbwrwb
qHyGO0aoSCqI3Haadr8faqU9GY/rOPNk3sgrDQoo//fb4hVC1CLQJ13hef4Y53CI
rU7m2Ys6xt0nUW7/vGT1M0NPAgMBAAGjQjBAMA4GA1UdDwEB/wQEAwIBBjAPBgNV
HRMBAf8EBTADAQH/MB0GA1UdDgQWBRR5tFnme7bl5AFzgAiIyBpY9umbbjANBgkq
hkiG9w0BAQsFAAOCAgEAVR9YqbyyqFDQDLHYGmkGJykIrGF1XIpu+ILlaS/V9lZL
ubhzEFnTIZd+50xx+7LSYK05qAvqFyFWhfFQDlnrzuBZ6brJFe+GnY+EgPbk6ZGQ
3BebYhtF8GaV0nxvwuo77x/Py9auJ/GpsMiu/X1+mvoiBOv/2X/qkSsisRcOj/KK
NFtY2PwByVS5uCbMiozziUwthDyC3+6WVwW6LLv3xLfhTjuCvjHIInNzktHCgKQ5
ORAZl4JMPJ+GslWYHb4phowim57iazXOoJwTdwJx4nLCgdNbOhdjsnvzqvHu7Ur
TkXWStAmzOVyyghqpZXjFaH3pO3JLF+l+/sKAIuvtd7u+Nxe5AW0wdeRlN8NwdC
jNPElpzVmbUq4JUagEiuTDkHszxHpFKVK7q4+63SM1N95R1NbdWhscdCb+ZAJzVc
oyi3B43njTOQ5yOf+1CceWxG1bQVs5ZufpsMlj4Ui0/1lvh+wjChP4kqKOJ2qxq
4RgqsahDYVvTH9w7jXbyLeiNdd8XM2w9U/t7y0Ff/9yi0GE44Za4rF2LN9d11TPA
mRGunUHBcnWEvgJBQl9nJEiU0Zsnvgc/ubhPgXRR4Xq37Z0j4r7g1SgEEzwxA57d
emyPxgcYxn/eR44/KJ4EBs+lVDR3veyJm+kXQ99b21/+jh5Xos1AnX5iltreGCc=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICGzCCAAGgAwIBAgIQQdKd0XLq7qeAwSxs6S+HUjAKBgqhkhjOPQQDAzBPMQsw
CQYDVQQGEwJVUzEpMCCGA1UEChMgSW50ZXJuZXQgU2VjdXJpdHkgUmVzZWZyY2gg
R3JvdXAxFtATBgNVBAMTDElTUkcgUm9vdCBYMTCCAiIwDQYJKoZIhvcNAQEBBQAD
ggIPADCCAgoCggIBAK3oJHP0FDfzm54rVygCh77ct984kIxuPOZXoHj3dcKi/vVqbv
YATyjb3miGbESTtrFj/RQSa78f0uoxmyF+0TM8ukj13Xnfs7j/EvEhmkvBioZxaUpm
ZmyPfxwv60pIgbz5MDmgK7iS4+3mX6UA5/TR5d8mUgU+g4rk8Kb4Mu0U1XjIB0ttov
0DiNewNwIRt18jA8+o+u3dpjq+sWT8KOEUt+zwvo/7V3LvSye0rgTBILDHCNAymg4
VMk7BPZ7hm/ELNKjD+Jo2FR3qyHB5T0Y3HsLuJvW5iB4YlcNHlsdu87kGJ55tukmi8
mxdAQ4Q7e2RCOFvu396j3x+UCB5iPNgiV5+I3lg02dZ77DnKxHZu8A/IJBdiB3QW0
KtZB6awBdpUKD9jf1b0SHzUvKBds0pjBqAlkd25HN7rOrFleaJ1/ctaJxQZBKT5ZPt
0m9STJEadao0xAH0ahmbWnOlFuhjuefXKnEgV4We0+UXgVCwOPjdAvBbI+e0ocS3M
FEvzG6uBQE3xDk3SzynTnjh8BCNAw1FtxNrQHusEwMFxIt4I7mKZ9YIqioymCzLq9
gwQbooMDQaHWBfEbwrwbqHyGO0aoSCqI3Haadr8faqU9GY/rOPNk3sgrDQoo//fb4
hVC1CLQJ13hef4Y53CIrU7m2Ys6xt0nUW7/vGT1M0NPAgMBAAGjQjBAMA4GA1UdD
wEB/wQEAwIBBjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBRR5tFnme7bl5AFzgA
iIyBpY9umbbjANBgkqhkiG9w0BAQsFAAOCAgEAVR9YqbyyqFDQDLHYGmkGJykIrGF
1XIpu+ILlaS/V9lZLubhzEFnTIZd+50xx+7LSYK05qAvqFyFWhfFQDlnrzuBZ6brJ
Fe+GnY+EgPbk6ZGQ3BebYhtF8GaV0nxvwuo77x/Py9auJ/GpsMiu/X1+mvoiBOv/2
X/qkSsisRcOj/KKNFtY2PwByVS5uCbMiozziUwthDyC3+6WVwW6LLv3xLfhTjuCvjH
IInNzktHCgKQ5ORAZl4JMPJ+GslWYHb4phowim57iazXOoJwTdwJx4nLCgdNbOhdjs
nvzqvHu7UrTkXWStAmzOVyyghqpZXjFaH3pO3JLF+l+/sKAIuvtd7u+Nxe5AW0wde
RlN8NwdCjNPElpzVmbUq4JUagEiuTDkHszxHpFKVK7q4+63SM1N95R1NbdWhscdCb
+ZAJzVcoyi3B43njTOQ5yOf+1CceWxG1bQVs5ZufpsMlj4Ui0/1lvh+wjChP4kqKO
J2qxq4RgqsahDYVvTH9w7jXbyLeiNdd8XM2w9U/t7y0Ff/9yi0GE44Za4rF2LN9d1
1TPAmRGunUHBcnWEvgJBQl9nJEiU0Zsnvgc/ubhPgXRR4Xq37Z0j4r7g1SgEEzwx
A57demyPxgcYxn/eR44/KJ4EBs+lVDR3veyJm+kXQ99b21/+jh5Xos1AnX5iltreGC
c=

-----END CERTIFICATE-----

-----BEGIN

CERTIFICATE-----

MIIFYDCCA0igAwIBAgIQCgFCgAAAAUujyES1AAAAAANBgkqhkiG9w0BAQsFADBK

MQswCQYDVQQGEwJVUzESMBAGA1UEChMJSWRlblRydXN0MScwJQYDVQQDEx5JZGVu
VHJ1c3QgQ29tbWVyY2lhbCBSb290IENBIDEwHhcNMTQwMTE2MTgxMjIzWhcNMzQw
MTE2MTgxMjIzWjBKMqswCQYDVQQGEwJVUzESMBAGA1UEChMJSWRlblRydXN0MScw
JQYDVQQDEx5JZGVuVHJ1c3QgQ29tbWVyY2lhbCBSb290IENBIDEwggLiMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQCnUBneP5k91DNG8W9RYYKyqU+PZ4ldhNIT
3Qwo2dfw/66VQ3KZ+bVdfIrBQuExUHTRgQ18zZshq0PirK1ehm7zCYofWjK9ouuU
+ehcCuz/mNKvcbO0U59Oh++SvL3sTzIwiEsXXlFEU8L2ApeN2WlrvyQfYo3fw7gp
S0l4PJNgiCL8mdo2yMKi1CxUAGc1bnO/AljwpN3lsKImesrgNqUZFvX9t++uP0D1
bVoE/c40yiTcdCMbXTMTEL3EASX2MN0CXZ/g1Ue9tOsbobtJSdifWwLziuQkkORi
T0/Br4sOdBeo0XKIanoBScy0RnnGF7HamB4HWfp1IYVl3ZBWzvurpWCdxJ35UrCL
vYf5jysjCiN2O/cz4ckA82n5S6LgTrx+kzmEB/dEcH7+B1rlsazRGMzyNeVJSQjK
Vsk9+w8YfYs7wRPCTY/JTw436R+hDmrfYi7LNQZReSzIJTj0+kuniVyc0uMNOYZK
dHzVWYfCP04MXFL0PfdSgvHqo6z9STQaKPNBiDoT7uje/5kdX7rL6B7yuVBgwDHT
c+XvvqDtMwt0viAgxGds8AgDelWaf0ZOlf0Hj7h9tgJ4TNkK2PXMl6f+cB7D3hv
l7yTmvmcEpB4eoCHFddydJxVdHixuuFucAS6T6C6aMN7/zHwcZ09lCqxC0EOoP5N
iGVreTO01wIDAQABo0IwQDAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUwAwEB
/zAdBgNVHQ4EFgQU7UQZwNPwBovupHu+QucmVMiONnYwDQYJKoZIhvcNAQELBQAD
ggIBAA2ukDL2pkt8RHYZYR4nKM1eVO8lvOMIkPkp165oCOGUAFjvLi5+U1KMtlwH
6oi6mYtQlNeCgN9hCQCTrQ0U5s7B8jeUeLBfnLOic7iPBZM4zY0+sLj7wM+x8uwt
LRvM7Kqas6pgghstO8OEPVeKlh6cdbjTMM1gCIOQ045U8U1mwF10A0Cj7oV+wh93
nAbowacYXVKV7cndJZ5t+qntozo00Fl72u1Q8zW/7esUTTHHYPTa8Yec4kjixsU3
+wYQ+nVZZjFHKdp2mhZpgq7vmrlR94gjmVYjzLVYA211QC//G5Xc7UI2/YRYRK
W2XviQzdFKcgyxIlJbQN+QHwotL0AMh0jEqSI5l2xPE4iUXfeu+h1sXIFRRk0pT
AwvsXcoz7WL9RccvW9xYoIA55vrX/hMUpu09lEpCdNTDd1lzzY9GvIU47/rokTLq
l1gElt44w8y8bckzOmoKaT+gyOpyj4xjhiO9bTyWnpXgSUyqorkqG5w2gXjtw+hG
4iZZRHUe2XWJUc0QhJ1hYMTd+ZciTY6Y5uN/9lu7rs3KSoFrXgvzUeF0K+l+J6fZ
mUIO+KWA2yUPHGNiiskzZ2s8EIPGrd6ozRaOjFAHN3Gf8qv8QfXBi+wAN10J5U6A
7/qxXDgGpRtK4dw4LTzcqx+QGtVKnO7RcGzM7vRX+Bi6hG6H

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFZjCCA06gAwIBAgIQCgFCgAAAAUujz0Z8AAAAAajANBgkqhkiG9w0BAQsFADBN
MQswCQYDVQQGEwJVUzESMBAGA1UEChMJSWRlblRydXN0MSowKAYDVQQDEyFJZGVu
VHJ1c3QgUHVibGljIFNlY3RvciBSb290IENBIDEwHhcNMTQwMTE2MTc1MzMyWhcN
MzQwMTE2MTc1MzMyWjBNNMqswCQYDVQQGEwJVUzESMBAGA1UEChMJSWRlblRydXN0
MSowKAYDVQQDEyFJZGVuVHJ1c3QgUHVibGljIFNlY3RvciBSb290IENBIDEwggLi
MA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQC2IpT8pEiv6EdrCvsnduTyP4o7
ekosMSqMjbCpwwFrqHd2hCa2rIFCDQjrVVi7evi8ZX3yoG2LqEfpYnYeEe4IFNGy
RBb06tD6Hi9e28tzQa68ALBKK0CyrOE7S8ItneShm+waOh7wCLPQ5CQ1B5+ctMIS
bdsHyo+1W/CD80/HLaXlrcuVIKQxKfDYWuSNG5qrmg0M8gozOSI5Cpcu81N3uURF
/YTLNiCBWS2ab21ISGHKTN9T0a9SvEsfqy9rg3LvdYDaBjMbXcjaY8ZNzaxmMc3R
3j6HEDbhuaR672BQssvKplbgN6+rNBM5Jeg5ZuSYeqoSmJxZZoY+rfGwyj4GD3vw
EU3oERte8uojHH01bWRNsZWfCYr3lEXsZdMUD2xlvI8BX0tIdUAvwFno157plzy
9yLxkA2T26pEUWbMfXYD62qoKjgZl3YN4ph+bz27nb9cCvdKTz4Ch5bQhyLVi9V
GxyhLrXHFub4qjySjmm2AcG1hp2JDws4lFT06tyePSW8Uybt1as5qsVATFSsrTZ
2fjXctscvG29ZV/viDUqZi/u9rNl8DONfJhBaUYYPQxxp+pu10GFqzcpL2UyQRqsV
WaFHVCKugyhFMKiq3IXAAaOReyL4jM9f9oZRORicsPflsbyVtTdX5Vy7W1f90gD

W/3FKqD2cyOEEBsB5wIDAQABo0lwQDAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/
BAUwAwEB/zAdBgNVHQ4EFgQU43HgntinQtnbcZFrlJPw6PRFKMwDQYJKoZIhvcN
AQELBQADggIBAef63QqwEZE4rU1d9+UOl1QZgkiHVlyqZJnYWv6IAcVYpZmxI1Qj
t2odIFlAWJBF9MJ23XLblSQdf4an4EKwt3X9wnQW3IV5B4Jaj0z8yGa5hV+rVHV
DRDtfULAj+7AmgjVQdZcDiFpboBhDhXAuM/FSRJSzL46zNQuOAXeNf0fb7iAaJg9
TaDKQGXS3z1i9kKIT/YPyNtGtEqJBnZhbMX73huqVjRI9PHE+1yJX9dsXNw0H8G
lwmEKYBhHfpe/3OsoOOJuBxxFcbeMX8S3OFtm6/n6J91eEyrRjuazr8FGF1NFTwW
mhlQBjQymmm9li1JfPFgEKCAZmExfrngdbkaqIHWchezxQMxNRF4eKLg6TCMf4Df
WN88uieW4oA0beOY02QnrEh+KHdcxiVhJfiFDGX6xDIvpZgF5PgLZxYWxoK4Mhn5
+bl53B/N66+rDt0b20XkeucC4pVd/GnwU2lhlXV5C15V5jgclKlZM57IcXR5f1GJ
tshquDDIajjDbp7hNxbqBWJMWxJH7ae0s1hWx0nzfxJoCTFx8G34Tkf71oXuxVhA
GaQdp/ILQzfcFpPz+vCZHTetBXZ9FRUGi8c15dxVJCO2SCdUyt/q4/i6jC8UDfv
8Ue1fXwsBOxonbRJRBD0ckscZOf85muQ3Wl9af0AVqW3rLatt8o+Ae+c

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIF8TCCA9mgAwIBAgIQALC3WhZIX7/hy/WL1xnmfTANBgkqhkiG9w0BAQsFADA4
MQswCQYDVQQGEwJFUzEUMBIGA1UECgwLSVpFTlBFIFMuQS4xEzARBgNVBAMMCKl6
ZW5wZS5jb20wHhcNMDCxMjEzMTMwODI4WchcNMzcwMjEzMDgyNzI1WjA4MQswCQYD
VQQGEwJFUzEUMBIGA1UECgwLSVpFTlBFIFMuQS4xEzARBgNVBAMMCKl6ZW5wZS5j
b20wgglMA0GCSqSIB3DQEBAQUAA4ICDwAwggIKAoICAQDJO3rKDX6sp4boFmVq
scIbRTJxldn+EFvMr+eleQGPicPK8lVx93e+d5TzcqQsRNiekpsUOqHnJJAKClAO
xdgmlOHZSOEtPtoKct2jmRXagaKH9HtuJneJWK3W6wyyQXpzbm3benhB6QiIEn6H
LmYRY2xU+zydcsC8Lv/Ct90NduM61/e0aL6i9eOBbsFGb12N4E3GVFWJGjMxCrFX
uaOKmMPsOzTFIUFpfnXCPCDFYbpRR6AgkJOhkEvzTnyFRVSA0QUmQbC1TR0zvsQD
yCV8wXDbo/QJLVQnSKwv4cSsPsjLkxTOTOcj7NMB+eAJRE1NZMDhDVqHlrytG6P+
JrUV86f8hBnp7KGItERphIPzidF0BqnMC9bC3ieFUCbKF7jJeodWLBoBHmy+E60Q
rLUk9TiRodZL2vG70t5HtfG8gfZZa88ZU+mNFctKy6lvROUbQc/hhqqfK0GqfvEyN
BjNaooXlkDWgYlwWTVdJovoDGrQscbNYLN57C9saD+veIR8GdwYDsMnvmfzAuU8L
hij+0rnq49qlw0dpEuDb8PYZi+17cNcC1u2HGCgsBCRMd+RlihrGO5rUD8r6ddIB
QFqNeb+Lz0vPqhbBleStIo+F5HUsWLlguWABKQDfo2/2n+iD5dPDNMN+9fR5XJ+
HMh3/1uaD7euBUbl8agW7EekFwIDAQABo4H2MIHzMIGwBgNVHREEgagwgaWBD2lu
Zm9AaXplbnBILmNvbaSBkTCBjjFHMEUGA1UECgw+SVpFTlBFIFMuQS4gLSBDSUYg
QTAXMzM3MjYwLVJNZXJjLlZpdG9yaWEtR2FzdGVpeiBUMTA1NSBGNjIgUzgxQzBB
BgNVBAkMOkF2ZGEgZGVsIE1lZGl0ZXJyYW5lbyBFdG9yYmlkZWVMTQgLSAwMTAx
MCBWAxRvcmlhLWdhc3RlaXowDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AQYwHQYDVIR0OBBYEFB0cZQ6o8iV7tJHP5LGx5r1VdGwFMA0GCSqSIB3DQEBCwUA
A4ICAQB4pgwWSp9MiDrAyw6lFn2fuUhfGI8NYjb2zRlrrKvV9pF9rnHzP7MOeIWb
laQnIUdCSnxIOvVfLMMjlf4rJUT3sb9fbgakEyrkgPH7UIBzg/YsfqikuFgba56
awmqxinuaElnMIAkejEWovt+8Rwu3WwJrfIwYJOubv5vr8qhT/AQKM6WfxZSzw
JNu0FXWuDYi6LnPAvViH5ULy617uHjAimcs30cQbIHsvm0m5hzkQiCeR7Csg1lw
LDXWrzY0tM07+DKo7+N4ifuNRSzanLh+QBxh5z6ikixL8s36mLYp//Pye6kflQCT
VyvehQP5aTfLnnhqBbTFMXiJ7HqnheG5ezzevh55hM6fcA5ZwjUukCox2eRFekGk
LhObNA5me0mrZJfQRsN5nXJQY6aYWwa9SG3YOYNw6DXwBdGqvOPbyALqfP2C2sJb
UjWumDqtujWTI6cfSN01RpiyEGjkpTHCClguGYEQyVB1/OpaFs4R1+7vUIgtYf8/
QnMFIEPVjjxOAToZpR9GTnfQXewBliGH/pR9hNiTrdZoQ0iy2+tzJOeRf1SktoA+
naM8THLCV8Sg1Mw4J87VBp6iSNnpn86CcDaTmjvfliHjWbcM2pE38P1ZWroZYgls

QyYBNWNgVYkDOnXYukrZVP/u3oDYLdE41V4tC5h9Pmzb/CaIxw==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIECjCCAvKgAwIBAgIJAMJ+QwRORz8ZMA0GCSqGSIb3DQEBCwUAMIGCMQswCQYD
VQQGEwJIVTERMA8GA1UEBwwIQnVkYXBlc3QxZjAUBGNVBAoMDU1pY3Jvc2VjIEx0
ZC4xJzAlBgNVBAMMHk1pY3Jvc2VjIGUtU3ppZ25vIFJvb3QgQ0EgMjAwOTefMB0G
CSqGSIb3DQEJARYQaW5mb0BILXN6aWduby5odTAeFw0wOTA2MTYxMTMwMThaFw0y
OTEyMzAxMTMwMThaMIGCMQswCQYDVQQGEwJIVTERMA8GA1UEBwwIQnVkYXBlc3Qx
ZjAUBGNVBAoMDU1pY3Jvc2VjIEx0ZC4xJzAlBgNVBAMMHk1pY3Jvc2VjIGUtU3pp
Z25vIFJvb3QgQ0EgMjAwOTefMB0GCSqGSIb3DQEJARYQaW5mb0BILXN6aWduby5o
dTCASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAOn4j/NjrdqG2KfgQvvP
kd6mJviZpWNwrZuuyjNAfW2WbqEORO7hE52UQIKavXWFdCyoDh2Tthi3jCyoZ/tc
cbna7P7of0/kLx2yqHWH2Lh5TvPmUpG0IMZfcChEhyVbUr02MeITTMuhTlAdX4U
fIASmFDHQWe4oIBhVKZsTh/gnQ4H6cm6M+f+wFUoLAKApxn1ntxVUwOXewdl/5n7
N4okxFnMUBBjjqqpGrCEGob5X7uxUG6k0QrM1XF+H6cbfPVTbiJfyvym1HxdrtbC
xkzIBQHZ7Vf8wSN5/PrIJOV87VqUQHqD9bpEqH5GoP7ghu5sJf0dgYzQ0mg/wu1
+rUCAwEAAaOBgDB+MA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB0G
A1UdDgQWBbTLd8bfQkPMPcu1SCOhGnqmKrs0aDAfBgNVHSMEGDAWgBTLd8bfQkPM
Pcu1SCOhGnqmKrs0aDAfBgNVHREEFdASgRBpbmZvQGUTC3ppZ25vLmh1MA0GCSqG
SIb3DQEBCwUAA4IBAQDJ0Q5eLtXMs3w+y/w9/w0lZMEyL/azXm4Q5DwpL7v8u8h
mLzU1F0G9u5C7DBsoKqpyvGvivo/C3NqPuouQH4frlRheesuCDfXI/OMn74dseGk
ddug4lQUsbocKaQY9hK6ohQU4zE1yED/t+AFdlfBHFny+L/k7SViXITwfn4fs775
tyERzAMBvNcnEJIeGzSBHq2cGsMEPO0CYdYeBvNfOofyK/FFh+U9rNHV4S9a67c
2Pm2G2JwCz02yULyMtd6YebS2z3PyKnJm9zbWETXbzivf3jTo60adbocwTZ8jx5t
HMN1Rq41Bab2XD0h7lbwyYiLXpUq3DDfSJlgnCW

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICWTCCAd+gAwIBAgIQZvI9r4fei7FK6gxXMqHC7DAKBggqhkJOPQQDAzBlMQsw
CQYDVQQGEwJVUzEeMBwGA1UEChMVTVWlJcm9zb2Z0IENvcnBvcnF0aW9uMTYwNAYD
VQQDEy1NaWNyb3NvZnQgRUNDIFJvb3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5IDIw
MTcwHhcnMTkxMjE4MjMwNjQ1WhcnNDIwNzE4MjMxNjA0WjBlMQswCQYDVQQGEwJV
UzEeMBwGA1UEChMVTVWlJcm9zb2Z0IENvcnBvcnF0aW9uMTYwNAYDVQQDEy1NaWNy
b3NvZnQgRUNDIFJvb3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5IDIwMTcwZjAQBgcq
hkjOPQIBBgUrgQQAIGNiAATUvD0CQnVBeyPNgASGAIEvaqiBYgtlzPbKnR5vSmZR
ogPZnZH6thaxjG7efM3beaYvzrvOcS/lpaso7GMEZpn4+vKTEAXhgShC48Zo9OYb
hGBKia/teQ87zvH2RPUBeMCjVDBSMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8E
BTADAQH/MB0GA1UdDgQWBbTIy5lycFIM+Oa+sgRXKSrPQhDtNTAQBgkrBgEEAYI3
FQEEAwIBADAKBggqhkJOPQQDAwNoADBlAjBY8k3qDPlfXu5gKcs68tvWMOQZP3zV
L8KxzJOUlSJMsbG7X7JNpQS5GiFBqlb0C8CMQCZ6Ra0DvpWSNSkMBaReNtUjGUB
iudQZsIxtzm6uBoiB078a1QWIP8rtedMDE2mT3M=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFqDCCA5CgAwIBAgIQHtOXCv/YtLNHcB6qvn9FszANBgkqhkiG9w0BAQwFADBl
MQswCQYDVQQGEwJVUzEeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcmlF0aW9uMTYw
NAYDVQQDEy1NaWNyb3NvZnQgUINBIFJvb3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5
IDIwMTcwHhcNMtKxMjE4MjI1MTIyWhcNNDIwNzE4MjMwMDIzWjBlMQswCQYDVQQG
EwJVUzEeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcmlF0aW9uMTYwNAYDVQQDEy1N
aWNyb3NvZnQgUINBIFJvb3QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5IDIwMTcwggLi
MA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDKW76UM4wplZEWCPw9R2LBifOZ
Nt9GkMml7Xhqb0eRaPgnZ1AzHaGm++DIQ6OEAlcBXZxIQIJTEly/xztokLaCLeX0
ZdDMbRnMlfl7rEqUrQ7eS0MdhweSE5CAg2Q1OQT85elss7YfUJQ4ZVBcF0a5toW1
HLUX6NZFndiyJrDKxHBKrmCk3bPZ7Pw71VdyvD/IybLeS2v4I2wDwAW9lcfNcztm
gGTjGqwu+UcF8ga2m3P1eDNbx6H7JyqhtJqRjJHTOOI+dkC0zVJhUXAoP8XFWvLJ
jEm7FFtNyP9nTUwSlq31/niol4fX/V4ggNyhSyL71Imtus5Hl0dVe49FyGcohJUc
aDDv70ngNXtk55iwlNpNhTs+VcQor1fznhPbRiefHqJeRIOkpcrVE7NLP8TjwuaG
YaRSMlI6IE9vDzhTyzMMeyuP1pq9KsgtsRx9S1HKR9FIJ3Jdh+vVReZIZZ2vUpC6
W6IYZVcSn2i51BVrlMRpIpj0M+Dt+VGOQVDJNE92kKz8OMHY4Xu54+OU4UZpyw4K
UGsTuqwPN1q3ErWQgR5WrlcihtnJ0tHXUeOrO8ZV/R4O03QK0dqq6mm4lyiPSMQH
+FJDOvTKVTUssKZqwJz58oHhEmrARdlns87/I6KJCITUFLkqqNfs+avNJVgyeY+Q
W5g5xAgGwax/Dj0ApQIDAQABo1QwUjAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/
BAUwAwEB/zAdBgNVHQ4EFgQUcCtZf4aycl8awznjwNnpv7tNsiMwEAYJKwYBBAGC
NxUBBAMCAQAwDQYJKoZIhvcNAQEMBAQgIBAKyvpI3CEZaJjqPnktaXFbgToqZC
LgLNfGvVZJ8og6Lq46BrsTaiXVq5lQ7GPAJtSzVXNUzltYkyLDVt8LkS/gxCP81OC
gMNPOsduET/m4xaRhPtthH80dK2Jp86519efhGSSvpWhrQITM93uCupKUY5vVau6
tZRGrox/2KJQJWVvgEbbMwSubLWYdFQl3Jpk+ONVFT24bcMKpBLBaYVu32TxU5nh
SnUgnZUP5NbcA/FZGOHhibJXWpS2qdgXKxdJ5XbLwVaZOjex/2kskZGT4d9Mozd2
TaGf+G0eHdP67Pv0RR0Tbc/3WeUiJ3IrhvNXuzDtJE3cfVa7o7P4NHmJweDyAmH3
pvwPuxwXC65B2Xy9J6P9LjrRk5Sxc0ki69bIImtt2dmefU6xqaWM/5TkshGsRGR
xpl/j8nWZjEgQRCHLQzWwa80mMpkg/sTV9HB8Dx6jKXB/ZUhoHHBk2dxEuqPiApp
GWSZI1b7rCoucL5mxAyE7+WL85MB+GqQk2dLsmijtWKP6T+MejteD+eMuMZ87zf9
dOLITzNy4ZQ5bb0Sr74MTnB8G2+NszKTc0QWbej09+CVgI+WXTik9KveCjCHk9hN
AHFiRSdLokKEW39lt2c0Ui2cFmuqqNh7o0JMcccMyj6D5KbvtwEwXlGjefVwaaZB
RA+GsCyRxj3qrg+E
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFojCCA4qgAwIBAgIUAZQwHqIL3fXfMyqxQ0Rx+NZQTQ0wDQYJKoZIhvcNAQEM
BQAwAaTELMAkGA1UEBhMCS1IkJAkBgNVBAoMHU5BVkVSIEJVU0IORVNTIFBMQVRG
T1JNIEVncnAuMTIwMAYDVQQDDCQlOQVZFUlBiHbG9iYWwUgUm9vdCBDZXJ0aWZpY2F0
aW9uIEF1dGhvcml0eTAeFw0xNzA4MTgwODU4NDJaFw0zNzA4MTgyMzU5NTlaMGkx
CzAJBgNVBAYTAktSMSYwJAYDVQQKDB1OQVZFUlBCVNVNjTKVTUyBQTEFURk9STSBD
b3JwLjEyMDAGA1UEAwppTkFWRVVlR2xvYmFsIFJvb3QgQ2VydGlmaWNhdGlvbiBB
dXR0b3JpdHkwggLiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQC21PGTXLVA
iQqrDZBbUGOUkJR0F0Vy1ntlWilP1agS7gvQnXp2XskWjFlqxcX0TM62RHcQDaH
38dq6SZeWYp34+hInDEW+j6RscrJo+KfziFTowI2MMtSAuXaMl3Dxeb57hHHI8IE
HoSTGEq0n+USZGnQJoViAbbJAh2+g1G7XNr4rRVqmfeSVpC0W+m/6imBEtRTkZaz
kVrd/pBzKPswRrXKCAfHcXLJZtM0l/aM9BhK4dA9WkW2aacc+yPOiNgSnABlqKYP
szuSjXEODMWLyEz59JuOuDXp7W87UC9Y7cSw0BwbagzivESq2M0UXZR4Yb8Obtoq
vC8MC3GmsxY/nOb5zJ9TNeIDoKAYv7vxvTWjLcNQvcGufFt7QSUqP620wbGQGhf

nZ3zVHbOUzoBppJB7ASjjw2i1QnK1sua8e9DXcCrpUHPXFNwcMmIpi3Ua2FzUCaG
YQ5fG8lr4ozVu53BA0K6lNpfqbDKzE0K70dpAy8i+/Eozr9dUGWokG2zdLAIx6yo
0es+nPxdGoMuK8u180SdOqcXYZaicdNwlhVNt0xz7hlcxVs+Qf6sdWA7G2POAN3a
CJBItOUt7kinaxeZVL6HSuOpXgRM6xBtVNbv8ejyYhbLgGvtPe31HzClrkvJE+2K
AQHJuFFYwGY6sWZLxNUxAmLpdIQM201GLQIDAQABo0IwQDAdBgNVHQ4EFgQU0p+I
36HNL3s9TsBAZMzJ7LrYEsWdgYDVR0PAQH/BAQDAgEGMA8GA1UdEwEB/wQFMAMB
Af8wDQYJKoZIhvcNAQEMBAQggIBADLKgLodPVQG3dLSLvCkASELZ0jKbY7gyKoN
qo0hV4/GPnrK21HUUrPUloSIWGB/5QuOH/XcChWB5Tu2tyIvCZwTfrFsDDUIbatj
cu3cvuzHV+YwIHHW1xDBE1UBjCpD5EHxzzp6U5LOogMFDTjfArsQLtk70pt6wKGm
+LUx5vR1yblTmXVHIloUFcd4G7ad6Qz4G3bxhYTEodoS76TiEJd6eN4MUZeoIUCL
hr0N8F5OSza7OyAfikJW4Qsav3vQIkMsRIz75Sq0bBwcupTgE34h5prCy8VCZLQe
lHsIJchxzIdFV4XTnyliIoNRlwAYl3dqmJLJfGBs32x9SuRwTMKeuB330DTHD8z7
p/8Dvq1wkNoL3chtl1+afwkyQf3NosxabUzyqkn+Zvjp2DXrDige7kgvOtB5CTh8
piKCK5XQA76+AqAF3SAi428diDRgxuYKuQl1C/AH6GmWNcf7I4GOODm4RStDeKLR
LBT/DSHycpWbXgnbiUSYqqFJu3FS8r/2/yehNq+4tneI3TqkbZs0kNwUXTc/t+sX
5Ie3cdCh13cV1ELX8vMxmV2b3RZtP+oGI/hGoiLtk/bdmuYqh7GYVPEi92tF4+KO
dh2ajcQGjTa3FPOdVGm3jjzVpG2Tgbet9r1ke8LJaDmgkpzNNIaRkPpkUZ3+/uul
9XXeifdy

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIEFTCCAav2gAwIBAgIGSUEs5AAQMA0GCSqGSIb3DQEBCwUAMIGnMQswCQYDVQQG
EwJIVTERMA8GA1UEBwwIQnVkYXBlc3QxFTATBgNVBAoMDE5ldExvY2sgS2Z0LjE3
MDUGA1UECwwuVGfuw7pzw610dsOhbnlraWFkw7NriChDZXJ0aWZpY2F0aW9uIFNI
cnZpY2VzKTE1MDMGA1UEAwwsTmV0TG9jayBBcmFueSAoQ2xhc3MgR29sZCkgRsrWR
dGFuw7pzw610dsOhbnkwHhcnMDgxMjExMTUwODIxWhcNMjg1MjA2MTUwODIxWjCB
pzELMAkGA1UEBhMCSFUxETAPBgNVBACMECEJ1ZGFwZXN0MRUwEwYDVQQKDAxOZXRM
b2NriEtmdC4xNzA1BgNVBAsMLIRhbsO6c8OtdHbDoW55a2lhZMOzayAoQ2VydGlm
aWNhdGlvb2Z0LjE3MDUwODIxWjCBMAwGA1UdEwQFMAMBAQggIBADLKgLodPVQG3dLSLvCkASELZ0jKbY7gyKoN
qo0hV4/GPnrK21HUUrPUloSIWGB/5QuOH/XcChWB5Tu2tyIvCZwTfrFsDDUIbatj
cu3cvuzHV+YwIHHW1xDBE1UBjCpD5EHxzzp6U5LOogMFDTjfArsQLtk70pt6wKGm
+LUx5vR1yblTmXVHIloUFcd4G7ad6Qz4G3bxhYTEodoS76TiEJd6eN4MUZeoIUCL
hr0N8F5OSza7OyAfikJW4Qsav3vQIkMsRIz75Sq0bBwcupTgE34h5prCy8VCZLQe
lHsIJchxzIdFV4XTnyliIoNRlwAYl3dqmJLJfGBs32x9SuRwTMKeuB330DTHD8z7
p/8Dvq1wkNoL3chtl1+afwkyQf3NosxabUzyqkn+Zvjp2DXrDige7kgvOtB5CTh8
piKCK5XQA76+AqAF3SAi428diDRgxuYKuQl1C/AH6GmWNcf7I4GOODm4RStDeKLR
LBT/DSHycpWbXgnbiUSYqqFJu3FS8r/2/yehNq+4tneI3TqkbZs0kNwUXTc/t+sX
5Ie3cdCh13cV1ELX8vMxmV2b3RZtP+oGI/hGoiLtk/bdmuYqh7GYVPEi92tF4+KO
dh2ajcQGjTa3FPOdVGm3jjzVpG2Tgbet9r1ke8LJaDmgkpzNNIaRkPpkUZ3+/uul
9XXeifdy

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDtTCCAp2gAwIBAgIQdrEgUnTwhYdGs/gjGvbCwDANBgkqhkiG9w0BAQsFADBt
MQswCQYDVQQGEwJDSDEQMA4GA1UEChMHV0lTZUtleTEiMCAGA1UECxmZT0lTVEUg
Rm91bmRhdGlvb1BFbmRvcnNlZDEoMCIYGA1UEAxMfT0lTVEUgV0lTZUtleSBHbG9i
YWwgUm9vdCBHQiBDQTAeFw0xNDEyMDExNTAwMzJaFw0zOTEyMDExNTEwMzFaMG0x
CzAJBgNVBAYTAkNIMRAwDgYDVQQKEwdXSVNIS2V5MSIwIAAYDVQQLExlPSVNURSBG
b3VuZGF0aW9uIEVuzG9yc2VkbMSgwJgYDVQQDEx9PSVNURSBXSXSVNIS2V5IEdsb2Jh
bCBSb290IEDiENBMBIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2Be3
HEokKtaXscriHvt9OO+Y9bI5mE4nuBFde9lIlliCFSZqGzG7qFshISvYD06fWvGx
WuR51jIjK+FTZJlFXHtPrby/h0oLS5daqPZI7H17Dc0hBt+eFf1Biki3IPShehtX
1F1Q/7pn2COZH8g/497/b1t3sWtuuMlk9+HKQUYOKXHQuSP8yYFfTvdv37+ErXNk
u7dCjmn21HYdfp2nuFeKUWdy19SouJVUQHMD9ur06/4oQnc/nSMbsrY9gBQHTC5P
99UKFg29ZkM3fiNDecNAhvVMKdqOmQ0NpQSHiB6F4+IT1ZvliwNjeOvgGUpuuy9r
M2RYk61pv48b74JixwIDAQAB01EwTzALBgNVHQ8EBAMCAYYwDwYDVR0TAQH/BAUw
AwEB/zAdBgNVHQ4EFgQUNQ/INmNe4qPs+TtmFc5RUuORmj0wEAYJKwYBBAGCNxUB
BAMCAQAwDQYJKoZIhvcNAQELBQADggEBAEBM+4eymYGQfp3FsLAmzYh7KzKNbrgh
cViXfa43FK8+5/ea4n32cZiZBKpDdHij40lhPnOMTzTg+XHEthYOU3gf1qKHLwI5
gSk8rxWYITD+KJAAjNHhy/peyP34EEY7onhCkRd0VQreUGdNZtGn//3ZwLWoo4rO
ZvUPQ82nK1d7Y0Zqqi5S2PTt4W2tKZB4SLrhI6qjiey1q5bAtEuiHZeeevJuQHf
aPFIc58Bd9Tzaml8LGBHAVRgOY1NK/VLSgWH1Sb9pWJmLU2NuJMW8c8CLC02Ic
Nc1MaRVUGpCY3useX8p3x8uOPUNpnJpY0CQ73xtAln41rYHHTnG6iBM=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICaTCCAe+gAwIBAgIQISpWdK7aDKtArb8roi066jAKBggqhkiOPQQDAzBtMQsw
CQYDVQQGEwJDSDEQMA4GA1UEChMHV0lTZUtleTEiMCAGA1UECxmZT0lTVEUgRm91
bmRhdGlvb1BFbmRvcnNlZDEoMCIYGA1UEAxMfT0lTVEUgV0lTZUtleSBHbG9iYWwg
Um9vdCBHQyBDQTAeFw0xNzA1MDkwOTQ4MzRaFw00MjA1MDkwOTU4MzNaMG0xCzAJ
BgNVBAYTAkNIMRAwDgYDVQQKEwdXSVNIS2V5MSIwIAAYDVQQLExlPSVNURSBGb3Vu
ZGF0aW9uIEVuzG9yc2VkbMSgwJgYDVQQDEx9PSVNURSBXSXSVNIS2V5IEdsb2JhbCBS
b290IEDiENBMBHYwEAYHKoZIzj0CAQYFK4EEACIDYgAETOlQwMYPchi82PG6s4ni
eUqjFqdrVCTbUf/q9Akkwwsin8tqJ4KBDdLArzHkdIJuyiXZjHWd8dvQmqJLIX4W
p2OQ0jnUsYd4XxiWD1AbNTcPasbc2RNNpI6QN+a9WzGRo1QwUjAOBgNVHQ8BAf8E
BAMCAQYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUSIcUrOPDnpBgOtfKie7T
rYy0UGYwEAYJKwYBBAGCNxUBBAMCAQAwCgYIKoZIzj0EAwMDAAwZQIwJsdpW9zV
57LnyAyMjMPdeYwbY9XJUpROTYJKcx6ygISpJcBMWm1JKWB4E+J+SOTkAJEA2zQg
Mgj/mkkCtojeFK9dbJlxjRo/i9fgojaGHAeConZT/cKi7e97sIBPWA9LUzm9
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFYDCCA0igAwIBAgIUeFhfLq0sGUvjNwc1NBMotZbUZZMwDQYJKoZIhvcNAQEL
BQAwSDELMakGA1UEBhMCQk0xGTAXBgNVBAoTEFFf1b1ZhZGlzIExpblWl0ZWQxHjAc
BgNVBAMTFVF1b1ZhZGlzIFJvb3QgQ0EgMSBHMzAeFw0xMjA1MTIxNzI3NDRaFw00
MjA1MTIxNzI3NDRaMEgxCzAJBgNVBAYTAkNIMRkwFwYDVQQKEwBRdW9WYW9WYWRpcyBM
aW1pdGVkMR4wHAAYDVQQDExVRdW9WYW9WYWRpcyBSb290IENBIDEgRzRwggLiMA0GCsQg
Sib3DQEBAAUAA4ICDwAwggIKAoICAQCgvlAQjunybEC0BJyFuTHK3C3kEakEPBtV
wedYMB0ktMPvhd6MLOHBPd+C5k+tR4ds7FtJwUrVu4/sh6x/gpqG7D0DmVIB0jWe

rNrwU8lmPNSsAgHaJNM7qAJGr6Qc4/hzWHa39g6QDbXwz8z6+cZM5cOGMAqNF341
68Xfuw6cwI2H44g4hWf6Pser4BOcBRiYz5P1sZK0/CPTz9XEJ0ngnjybCKOLXSoh
4Pw5qlPafX7PGglTvF0FBM+hSo+LdoINofjSxxR3W5A2B4GbPgb6U15jxaYA/qXp
UhtStZI5cgMJYr2wYBZupt0lwGnm3fME0UDiTouG9G/lg6AnhF4EwfWQvTA9xO+o
abw4m6SkltFi2mnAAZauy8RRNOoMqv8hjlmpSlzkYZqn0ukqeI1RPToV7qJZjqlc
3sX5kCLliEVx3ZGZbHqfPT2Yff72vhZooF6uCyP8Wg+qInYtyaEQHeTTRCOQIj/G
KubX9ZqzWB4vMikIG1SitZgj7Ah3HJVdYdHLiZxfokqRmu8hqkkWCKi9YSgxyXSt
hfbZxbGL0eUQMk1fiyA6PEkfM4VZDdvLCXVDAxp7a3F98N/ETH3Goy7IlXnLc6KO
Tk0k+17kBL5yG6YnLUlamXrXXAkg3+UuU/xDRxeiElbEbfkduebPRq34wGmAot
zCjvpUfzUwIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAWIB
BjAdBgNVHQ4EFgQUo5fW816iEOGrRZ88F2Q87gFwnMwwDQYJKoZIhvcNAQELBQAD
ggIBABj6W3X8PnrHX3fHyt/PX8MSxEBd1DKquGrX1RUVRpjpeaQWxiZTOOtQqOC
MTalzen7xASWSIsBx40Bz1szBpZGZnQdT+3Btrm0DWHMY37XLneMlhqwI2hrhVd2
cDMT/uFPpiN3GPOajOi9ZcnPP/TJF9zrx7zABC4tRi9pZsMbj/7sPtPKIL92CiUN
qXsCHKnQO18LwIE6PWThv6ctTr1NxNgpxiIY0MWscgKCP6o6joilzHdCGPDdRS5
YCgtW2jgFqlmgiNR9etT2DGbe+m3nUvriBbP+V04ikkwj+3x6xn0dxoxGE1nVGww
b2X52z3sIexe9PSLyMBIVNFxZPT5pqOBMzYzcfCkeF9OrYMh3jRjehZrJ3ydl02
8hP0r+AJx2EqbPfgna67hkooby7utHnNkDPDs3b69fBsnQGG+P6Q9pxyz0fawx/k
NSBT8lTR32GDpgLiJTjehTItXnOQUl1CxM49S+H5GYQd1aJQZEH7QRTDvdbJWqNj
ZgKAuVQU6O0ec7AAmTPWIUb+oI38YB7AL7YsmoWTTYUrrXJ/es69nA7Mf3W1daWhp
q1467HxpvMc7hU6eFbm0FU/DIXpY18ls6Wy58yljXrQs8C097Vpl4KlbQMJIImYFt
nh8GKjwStIsPm6lk8KaN1nrgS7ZklmOVhMJKzRwuJicYOXD

-----END

CERTIFICATE-----

-----BEGIN

CERTIFICATE-----

MIIftZCCA5+gAwIBAgICBQkwDQYJKoZIhvcNAQEFBQAwrTELMakGA1UEBhMCQk0x
GTAXBgNVBAoTEFF1b1ZlZGlzIExpblw0ZWQxGzAZBgNVBAMTElF1b1ZlZGlzIFJv
b3QgQ0EgMjAeFw0wNjExMjQxODI3MDBaFw0zMTEyMjQxODIzMzNaMEUxGzAJBgNV
BAYTAkNMRkwFwYDVQQKEwBRdW9WYW9WYWRpcyBMAw1pdGVkMRswGQYDVQQDEwJRdW9W
YWRpcyBSb290IENBIDlwggLiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCa
GMpLIA0ALa8DKYrwD4HlrkwZhR0In6spRIXzL4GtMh6QRr+jhiYaHv5+HBg6XJxg
Fyo6dIMzMH1hVBHL7avg5tKifvVrbxi3Cgst/ek+7wrGsxDP3MJGF/hd/aTa/55J
WpzmM+Yklvc/ulsrHHo1wtZn/qtmUlTKGAr79dgw8eTvl02kfn/+NsRE8Scd3bB
rrcCaoF6qUWD4gXmuVbBlDePSHFjIuwXZQeVikvfj8ZaCuWw419eagGrDPmF60Tp
+ARz8un+XJiM9XOva7R+zdRcAitMOeGylZUtQofX1bOQQ7dsE/He3fbE+Ik/0XX1
ksOR1YqI0JDs3G3eicJlcZaLDQP9nL9bFqyS2+r+eXyt66/3FsvbzSUr5R/7mp/i
Ucw6UwxI5g69ybR2BILmEROFcmMDBOAENisgGQLodKcftslWZvB1JdxnwQ5hYIiz
PtGo/KPaHbDRsSNU30R2be1B2MGyIrZTHN81Hdyhdyox5C315eXbyOD/5YDXC2Og
/zOhD7osFRXql7PSorW+8oyWHhqPHWykYTE5hnMz15eWniN9gqRMgeKh0bnpX5UH
oycr7hYQe7xFSkyyBNKr79X9DFHOUGoImfmR2gyPZFwDwzqLID9ujWc90tb+fVuI
yV77zGHcizN300QyNQliBIWENieJ0f7OyHj+OsdWwIDAQABo4GwMIGtMA8GA1Ud
EwEB/wQFMAMBAf8wCwYDVR0PBAQDAgEGMB0GA1UdDgQWBbQahGK8SEwzJQTU7tD2
A8QZRtGUazBuBgNVHSMEZzBlBQahGK8SEwzJQTU7tD2A8QZRtGUa6FJpEcwRTEL
MAkGA1UEBhMCQk0xGTAXBgNVBAoTEFF1b1ZlZGlzIExpblw0ZWQxGzAZBgNVBAMT
ElF1b1ZlZGlzIFJvY3QgQ0EgMoICBQkwDQYJKoZIhvcNAQEFBQAAdgglBAD4KFk2f
BluornFdLwUvZ+YTRYPENvbzwCYMDbVHZF34tHLJRqUDGCdViXh9duqWNIAXINzn

g/iN/Ae42l9NLmeyhP3ZRPx3UIHmfLTJDQtyU/h2BwdBR5YM++CCJpNVjP4iH2Bl
fF/nJrP3MpCYUNQ3cVX2kiF495V5+vgJodmVjB3pjd4M1IQWK4/YY7yarHvGH5K
WWPKjaJW1acvvFYfzznB4vsKqBUsfU16Y8Zsl0Q80m/DShcK+JDSV6IZUaUtl0Ha
B0+pUNqQjZRG4T7wlP0QADj1O+hA4bRuVhogzG9Yje0uRY/W6ZM/57Es3zrWIoZc
hLsib9D45MY56QSIPMO661V6bYCYJPVsAfv4l7CUW+v90m/xd2gNNWQjrLhVoQPR
TUIZ3Ph1WVaj+ahJefivDrkRoHy3au000LYmYjgahwz46P0u05B/B5EqHdZ+XIWD
mbA4CD/pXvk1B+TJYm5Xf6dQlfe6yJvmjqIBxdZmv3lh8zwc4bmCXF2gw+nYSL0Z
ohEUGW6yhhToPkg3Goi3XZZenMfvJ2II4pEZXLNlXId26F0KCl3GBUzGpn/Z9Yr9y
4aOTHcyKJloJONDO1w2AFrR4pTqHTI2KpdVGl/IsELm8VCLAABpQ570su9t+Oza
8eOx79+Rj1QqCyXBjhnEUhAFZdWCEOrCMc0u

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFYDCCA0igAwIBAgIUrfC0JFuBiZs18s64KztbpybwdSgwDQYJKoZIhvcNAQEL
BQAwSDELMAkGA1UEBhMCQk0xGTAXBgNVBAoTEFF1b1ZlZGlzIExpbl0ZWQxHjAc
BgNVBAMTFVF1b1ZlZGlzIFJvb3QgQ0EgMiBHMzAeFw0xMjAxMTIxODU5MzJaFw00
MjAxMTIxODU5MzJaMEGxCzAJBgNVBAYTAkNMRkwFwYDVQQKEExBRdW9WYWWRpcyBM
aW1pdGVkMR4wHAAYDVQQDExVRdW9WYWWRpcyBSb290IENBIDlgRzMWggliMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQChriWyARjcV4g/Ruv5r+Lrl3HimtFhZiFf
qq8nUeVuGxbULX1QsFN3vXg6YOJkApt8hvpWGo6t/x8Vf9WVHhLL5hSEBMHfNrMW
n4rjyduYNM7YMXcoRvynyfDStNVNCXJJ+fKH46nafaf9a7I6JaltUkSs+L5u+9ym
c5GQYayYDFCDy54ejiK2toIz/pgslUiXnFgHVy7g1gQyJO/Dh4fxaXc6AcW34Sas+
O7q414AB+6XrW7PFXmAqMaCvN+ggOp+oMiwMzAkd056OXbxMmO7FGmh77FOM6RQ1
o9/NgJ8MSPsc9PG/Srj61YxxSscfrf5BmrODXfKEVU+IV0POKa2Mq1W/xPtbAd0j
IaFYAI7D0GoT7RPjEiuA3GfmlbLNHiJuKvB1PLKFAeNilUSxmn1uIZoL1NesNKq
IcGY5jDjZ1XHm26sGahVpkUG0CM62+tLXSoREfA7T8pt9DTEceT/AFr2XK4jYIVz
8eQQsSWu1ZK7E8EM4DnatDlXtas1qnIhO4M15zHfeiFuuDIIIfR0ykRVKYnLP43eh
vNURG3YBZwjgQQvD6xVu+KQZ2aKrr+InUIYrAoosFCT5v0ICvybIxo/gbjh9Uy3l
7ZizlWNof/k19N+IxWA1ksB8aRxlRbQ694Lrz4EEEVIWFA4r0jyWbYW8jwNkALG
cC4BrTwV1wIDAQABo0IwQDAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIB
BjAdBgNVHQ4EFgQU7edvdlq/YOxJW8ald7tyFnGbxD0wDQYJKoZIhvcNAQELBQAD
ggIBAIIHfD9DCX5xwvfrs4iP4VGyvD11+ShdyLyZm3tdquXK4Qr36LLTn91nMX66
AarHake7kNQIXLJgapDwyM4DYvmL7ftuKtwGTTwpD4kWilhMSA/ohGHqPHKmd+RC
roijQ1h5fq7KpVMNqT1wvSAZYaRsOPxDMuHBR//47PERIjKWnML2W2mWeyAMQ0Ga
W/ZZGYjeVYg3UQt4XAoeo0L9x52ID8DyeAlkVJOviYeIyUqAHerQbj5hLja7NQ4n
lv1mNDthcnPxFlxHBIRJAHpYErAK74X9sbgzdWqTHBLmYF5vHX/JHyPLhGGfHoJE
+V+tYIUkmlKY7VHnoX6XOuYvHxHaU4AshZ6rNRDbll9qxV6XU/IyAgkwo1jwDQHV
csaxfGl7w/U2Rcxhbl5MLMVerugOXou/983g7aEOGzPuVBj+D77vfoRrQ+NwmNtd
dbINWQeFFSM51vHfqSYP1kjHs6Yi9TM3WpVHn3u6GBVv/9YUZINJ0gpnIdsPNWNg
KCLjsZWDzYWm3S8P52dSbrsvhXz1SnPnxT7AvSESBT/8twNJAlvIjebiVDj1eYeM
HVOyToV7BjjHLPj4sHKNJeV3UvQDHEimUF+IIDBu8oJDqz2XhOdT+yHBTw8imoa4
WSr2Rz0ZiC3oheGe7IUlarFsNMkd7EgrO3jtZsSOeWmD3n+M

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIGnTCCBIWgAwIBAgICBcYwDQYJKoZIhvcNAQEFBQAwRTELMAkGA1UEBhMCQk0x

GTAXBgNVBAoTEFF1b1ZhZGlzIExpWl0ZWQxGzAZBgNVBAMTElF1b1ZhZGlzIFJv
b3QgQ0EgMzAeFw0wNjExMjQxOTExMjNaFw0zMTEwMjQxOTA2NDRaMEUxGzAJBgNV
BAYTAkJKMRkwFwYDVQQKEsBRdW9WYWRpcyBMAW1pdGVkMRswGQYDVQQDEsJRdW9W
YWRpcyBSb290IENBIDMwgglMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDM
V0lWVJzmmNPTTe7+7cefQzlKZbPoFog02w1ZkXTPkrgEQK0CSzGrvI2RaNggDhoB
4hp7Thdd4oq3P5kazethq8Jlph+3t723j/z9cI8LoGe+AaJZz3HmDyl2/7FWeUUr
H556VOijKTVopAFPD6QuN+8bv+OPEKhyq1hX51SGyMnzW9os2l2ObjyPtr7guXd
8lyyBTNvijbO0BNO/79KDDRMpsMhvVAEVeuxu537RR5kFd5VAYwCdrXLoT9Cabwv
vWhDFlaJKjdHkf2mrk7AyxRlIDdLkgbvBNDInIjbC3uBr7E9KsRIOni27tyAsdLT
mZw67mtaa7ONt9XOnMK+pUsvFrGeaDsGb659n/je7Mwpp5ijJUMv7/FfJuGITfhe
btfZFG4ZM2mnO4SJk8RTVROhUXhA+LjJou57ulJCg54U7QVSWllWp5f8nT8KKDjc
T5EOE7zelaTfi5m+rJsziO+1ga8bxiJTYPbH7pcUsMV8eFLI8M5ud2CEpukqdiDt
WAEXMJPPGovgc2PZapKUSU60rUqFxmMiMPwJ7Wgic6aIDFUhWMXhOp8q3crhkODZ
c6tsgLjoC2SToJyMGf+z0gzskSaHirOi4XCPLArlzW1oUevaPwV/izLmE1xr/19A
4iLlTLRkT9a6fUg+qGkM17uGclzuD87nSVL2v9A6wIDAQABo4IBITCCAZEwDwYD
VR0TAQH/BAUwAwEB/zCB4QYDVR0gBIH2MIH2MIHTBgkrBgEEAb5YAAMwgcUwgZMG
CCsGAQUFBwICMIGGGGoGDQW55IHVzZSBvZiB0aGlzIENlcnRpZmljYXRlIGNvbnN0
aXR1dGVzIGFjY2VwdGFuY2Ugb2YgdGhlIFF1b1ZhZGlzIFJvY3QgQ0EgMyBDZXJ0
aWZpY2F0ZSBQb2xpY3kgLyBDZXJ0aWZpY2F0aW9uIFByYWN0aWNlIFN0YXRlZWVw
dC4wLQYIKwYBBQUHAQEWIWh0dHA6Ly93d3cucXVvdmFkaXNnbG9iYWwY29tL2Nw
czALBgNVHQ8EBAMCAQYwHQYDVR0OBBYEFPLAE+CCQz777i9nMpY1XNu4ywLQMG4G
A1UdIwRnMGWAFPLAE+CCQz777i9nMpY1XNu4ywLQoUmkRzBFMQswCQYDVQQGEwJC
TTEZMBcGA1UEChMQUXVvVmFkaXMGtGltXRlZDEbMBkGA1UEAxMSUXVvVmFkaXMG
Um9vdCBDQSAzggIFxjANBgkqhkiG9w0BAQUFAAOCAgEAT62gLEz6wPJv92ZVqyM0
7ucp2sNbtrCD2dDQ4iH782CnO11gUyeim/YIirn6By5ZwkajGxkHon24QRiSem
d1o417+shvzuXYO8BsbRd2sPbSQvS3pspweWyuOEn62lix2rFo1bZhfZFvSLgNLd
+LJ2w/w4E6oM3kJPk27zPOuAJ9v1pkQNN1pVWQvVDVJlxa6f8i+AxeoyUDUSly7B
4f/xI4hROJ/yZlZ25w9Rl6VSDE1JUZU2Pb+iSwwQHYaZTKrzchGT5Or2m9qoXadN
t54CrnMAyNojA+j56hl0YgCUyyIgvpsnWbWCar6ZeXqp8kokUvd0/bpO5qgdAm6x
DYBEwa7TlZdfu4V8K5Iu6H6li92Z4b8nby1dquH/grdS/yO9SbkbnBCbjPsMZ57
k8HkyWkaPcBrTiJt7qtYTcbQQcEr6k8Sh17rRdhs9ZgC06DYVYoGmRmioHfRMJ6s
zHXug/WwYjnPbFfiTNKRCw51KBuav/0aQ/HKd/s7j2G4aSgWQgRecCocIdiP4b0j
Wy10QJLZYxkNc91pvGJHvOB0K7Lrfb5BG7XARsWhlStfTsEokt4YutUqKLsRixeT
mJlglFwjz1onl14LBQaTNx47aTbrqZ5hHY8y2o4M1nQ+ewkk2gF3R8Q7zTSMmfXK
4SVhM7JZG+Ju1zdXtg2pEto=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFYDCCA0igAwIBAgIUlvWbAiin23r/1aOp7r0DoM8Sah0wDQYJKoZIhvcNAQEL
BQAwSDELMAkGA1UEBhMCQk0xGTAXBgNVBAoTEFF1b1ZhZGlzIExpWl0ZWQxHjAc
BgNVBAMTFVF1b1ZhZGlzIFJvY3QgQ0EgMyBHMzAeFw0xMjAxMTIyMDI2MzJaFw00
MjAxMTIyMDI2MzJaMEGxGzAJBgNVBAYTAkJKMRkwFwYDVQQKEsBRdW9WYWRpcyBM
aW1pdGVkMR4wHAAYDVQQDEsVRdW9WYWRpcyBSb290IENBIDMgRzMwgglMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQCzyw4QZ47qfJenMioKVjZ/aEzHs286IxSR
/xl/pqcs7rN2nXrpixurazHb+gtTTK/FpRp5PIpM/6zfJd5O2YIyC0TeytuMrKNu
FoM7pmRLMon7FhY4futD4tN0SsJiCnMK3UmzV9KwCoWdcTzeo8vAMvMBOSBDGzXR
U7Ox7sWTaYI+FrUoRqHe6okJ7UO4BUAKhvVZR74bbwEhELn9qdIoyhA5CcoTNs+c

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFiTCCA3GgAwIBAgIQb77arXO9CEDii02+1PdbkTANBgkqhkiG9w0BAQsFADBO
MQswCQYDVQQGEwJVUzEYMBYGA1UECgwPU1NMIENvcnBvcnF0aW9uMSUwIwYDVQQD
DBxTU0wuY29tIFRmUyBSU0EgUm9vdCBDQSAyMDIyMB4XDTIyMDgyNTE2MzQyMloX
DTQ2MDgxOTE2MzQyMVowTjELMAkGA1UEBhMCVVMxGDAWBgNVBAoMD1NTTCBDb3Jw
b3JhdGlvbjElMCMGA1UEAwwcU1NMLmNvbSBUTFMgUINBIFJvb3QgQ0EgMjAyMjCC
AiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBANckCXJPQIgSYT41I57u9nTP
L3tYPc48DRAokC+X94xI2KDYJbFMsbFMF3NQ0CJkY7uB0ylu1bUJPIYYf7ISf5OY
t6/wNr/y7hienDtSxUcZXXtZGbvXcdotL8bHAajvI9AI7YexoS9UcQbOcGV0ins
S657Lb85/bRi3pZ7QcacoOAGcvvwB5cJOYF0r/c0WRFXCsjbwST0MXMwgsadugL3
PnxEX4MN8/HdIGkWCVDi1FW24IBydm5MR7d1VVm0U3TZIMZBrViKMWYPHqIbKUBO
L9975hYsLfY/7PO0+r4Y9ptJ1O4Fbtk085zx7AGL0SDGD6C1vBdOSHtRwvzpXGk3
R2azaPgVKPC506QVzFpPulJwoxJF3ca6Tvvc0PeoUidtnm1jPx7jMEWTO6Af77w
dr5BUxIzrlo4QqvXDz5BjXYHMTWrfZOZ9mxQnUjbvPNQrL8VfVThxc7wDNY8VLS
+Yck80jwO4s4zKTGkH8Pn2L0aPP2oOnaclQntVcBdIKQXTbYxE3waWglksejBYS
d66UNHsef8JmAOSqg+qKkK3ONkRN0VHpvB/zagX9wHQfJRIAuw7qglFA35u5CCoG
AtUjHBPW6dvbxbR6y3snm/vg1UYk7RBLy0ulBY+6uB0rpvqR4pJSvezrZ5dtmi2f
gTIFZzL7SAg/2SW4BCUvAgMBAAGjYzBhMA8GA1UdEwEB/wQFMAMBAf8wHwYDVR0j
BBgwFoAU+y437uOEeicuzRk1sTN8/9REQRkwHQYDVR0OBBYEFpsuN+7jhHonLs0Z
NbEzfp/UREK5MA4GA1UdDwEB/wQEAwIBhjANBgkqhkiG9w0BAQsFAAOCAgEAjYlt
hEUy8U+zoO9opMAdrDC8Z2awms22qyIZZtM7QbUQnRC6cm4pJCAcAZli05bg4vsM
QtfhWsSWTVTNj8pDU/0quOr4ZcoBwq1gaAafORpR2eCNJvkLTqVTJXojpBzOCBvf
R4iyrT7gJ4eLSYwfqUdYe5byiB0YrrPRpgqU+tvT5TgKa3kSM/tKWTcWQA673vWJ
DPFs0/dRa1419dvAJuoSc06pkZCmF8NsLzjUo3KUQyxi4U5cMj29TH0ZR6LDSeeW
P4+a0zvKEdiLA9z2tmBVGKaBUfPhqBVq6+AL8BQx1rmMRTqoENjwuSfr98t67wVy
lrXEj5ZzxOhWc5y8aVFjvO9nHEMaX3cZHxj4HCUp+UmZKbaSPaKDN7EgkaibMOlq
bLQjk2UEqxHzDh1TJElTHaE/nUiSEeJ9DU/1172iWD54nR4fK/4huxoTtrEoZP2w
AgDhblCivRZQIA9ygV/MIP+7mea6kMvq+cYMwq7FGc4zoWtcu358NFcXrfA/rs3q
r5nsLFR+jM4uElZI7xc7P0peYNLcdDa8pUNjyw9bowJWCZ4kLOGGgYz+qxcs+sji
Mho6/4UlyYOf8kpIEFR3N+2ivEC+5BB09+Rbu7nzifmPQdjH5FCQNYA+HLhNkNPU
98OwoX6EyneSMSy4kLGCenROmxMmtNVQZIR4rmA=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDcjCCAlqgAwIBAgIUPopdB+xV0jLVt+O2XwHrLdzk1uQwDQYJKoZIhvcNAQEL
BQAwUTELMAkGA1UEBhMCUEwKDAmbGNVBAoMH0tyYWpvd2EgSXpiYSBSb3psaWN6
ZW5pb3dhIFMuQS4xGDAWBgNVBAMMD1NaQUZJUjBST09UIENBMjAeFw0xNTEwMTkw
NzQzMzBaFw0zNTEwMTkwNzQzMzBaMFExCzAJBgNVBAYTAiBMMSgwJgYDVQQKDDB9L
cmFqb3dhIEl6YmEgUm96bGljemVuaW93YSBTLkEuMRgwFgYDVQQDDA9TWkFGSVI
Uk9PVCBQDQTIwggEiMA0GCSqGSIb3DQEBQUAA4IBDwAwggEKAoIBAQC3vD5QqEvN
QLXOYeeWyrSh2gwisPq1e3YAd4wLz32ohswmUeQgPYUM1lj5/QqGJ3a0a4m7utT
3PSQ1hNKDJA8w/Ta0o4NkjrcsbH/ON7Dui1fgLkCvUqdGw+0w8LBZwPd3BucPbOw
3gAeqDRHu5rr/gsUvTaE2g0gv/pby6kWIK05YO4vdbbnl5z5Pv1+TW9NL++IDWr6
3fE9biCloBK0TXC5ztdyO4mTp4CEHCdJckm1/zuVnsHMyAHs6A6KCpbns6aH5db5

BSsNl0BwPLqsdVqc1U2dAgrSS5tmS0YHF2Wtn2yIANwiieDhZNRnvDF5YTy7ykHN
XGoAyDw4jlivAgMBAAGjQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAAQH/BAQD
AgEGMB0GA1UdDgQWBbQuFqlKGLXLzPVvUPMjX/hd56zwyDANBgkqhkiG9w0BAQSF
AAOCAQEAtXP4A9xZWx126aMqe5Aosk3AM0+qmrHUuOQn/6mWmc5G4G18TKI4pAZw
8PRBEew/R40/cof5O/2kbytTAOD/OblqBw7rHRz2onKQy4I9EYKL0rufKq8h5mOG
nXkZ7/e7DDWQw4rtTw/1zBLZpD67oPwglV9PJi8RI4NOdQcPv5vRtB3pEAT+ymCP
oky4rc/hkA/NrgRHXxU3UNLUYfrVFdvXn4dRVoul4+vJhaAlIDf7js4MNIThPIGy
d05DpYhfhmehPea0XGG2Ptv+tyjFogeutcrKjSoS75ftwjCkySp6+/NNIxuZMzSg
LvWpCz/UXeHPhJ/iGcJfitYgHuNztw==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIICOjCCAcGgAwIBAgIQQvLM2htpN0RfFf51KBC49DAKBggqhkiOPQQDAzBfMQsw
CQYDVQQGEwJHQjEYMBYGA1UEChMPU2VjdGlnbyBMAW1pdGVkMTYwNAYDVQQDEy1T
ZWN0aWdvIFB1YmXpYyBTZXJ2ZXIgcXV0aGVudGljYXRpb24gUm9vdCBFNNDYwHhcN
MjEwMzIyMDAwMDAwWhcNNDYwMzIxMjM1OTU5WjBfMQswCQYDVQQGEwJHQjEYMBYG
A1UEChMPU2VjdGlnbyBMAW1pdGVkMTYwNAYDVQQDEy1TZWN0aWdvIFB1YmXpYyBT
ZXJ2ZXIgcXV0aGVudGljYXRpb24gUm9vdCBFNNDYwdjAQBgcqhkiOPQIBBgUrgQQA
IgNiAAR2+pmpbiDt+dd34wc7qNs9Xzjoq1WmVh/WSOrsfy2qw7LFeeyZYX8Qeccc
WvkEN/U0NSt3zn8gj1KjAIns1aeibVvjS5KToID1AZTc8GgHHs3u/iVStSBDHBv+
6xnOQ6OjQjBAMB0GA1UdDgQWBbTRItPmWfFLXyY4qp3W7usNw/upYTAOBgNVHQ8B
Af8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAKBggqhkiOPQQDAwNnADBkAjaN7qRa
qCG76UeXlImldCBteU/IvZNeWbj7LRoAasm4PdCkT0RHlAFWovgzJQxC36oCMB3q
4S6ILuH5px0CMk7yn2xVdOOurvulGu7t0vzCAxHrRVxgED1cf5kDW21USAGKcw==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIFijCCA3KgAwIBAgIQdY39i658BwD6qSWn4cetFDANBgkqhkiG9w0BAQwFADBf
MQswCQYDVQQGEwJHQjEYMBYGA1UEChMPU2VjdGlnbyBMAW1pdGVkMTYwNAYDVQQD
Ey1TZWN0aWdvIFB1YmXpYyBTZXJ2ZXIgcXV0aGVudGljYXRpb24gUm9vdCBSNDYw
HhcNMjEwMzIyMDAwMDAwWhcNNDYwMzIxMjM1OTU5WjBfMQswCQYDVQQGEwJHQjEY
MBYGA1UEChMPU2VjdGlnbyBMAW1pdGVkMTYwNAYDVQQDEy1TZWN0aWdvIFB1YmXp
YyBTZXJ2ZXIgcXV0aGVudGljYXRpb24gUm9vdCBSNDYwgglMA0GCSqGSIb3DQEB
AQUAA4ICDwAwggIKAoICAQCTvtU2UnXYASOGHEdCSe5jtrch/cSV1UgrJnwUUXDa
ef0rty2k1Cz66jLdScK5vQ9IPXtamFSvnl0xdE8H/FAh3aTPaE8bEmNtJZIMKpnz
SDBh+oF8HqclStw+KxwfGExxqjWMrfhu6DtK2eWUAtaJhBOqbchPM8xQljeSM9xf
iOefVNII8JhD1mb9nxc4Q8UBUQvX4yMPFF1bFOdLvt30yNoDN9HWOaEhUTCDsG3X
ME6WW5HwcCsrv0WBZEMNvSE6Lzppng3LILVCJ8zab5vuZDCQOc2TZYehMbUjUDM3
IuM47fgxMMxF/mL50V0yeUKH32rMVhlATc6qu/m1dkmU8Sf4kaWD5QazYw6A3OAS
VYCM02a0OYctyPDQ0RTp5A1NDvZdV3LFOxxHVp3i1fuBYyzMTYcQNFu31xR13NgE
SJ/AwSiItOkcyqex8Va3e0IMWeUgFaiEAin6OJRpmkkGj80feRQXEgyDet4fsZfu
+Zd4KKTIRJLpfSYFplhym3kT2BFfrsU4YjRosoYwjviQYZ4ybPUHNs2iTg7sijbt
8uaZFURww3y8nDnAtOf94MII1fZEoDISfB1D++N6xybVCi0ITz8fAr/73trdf+L
HaAZBav6+CuBQug4urv7qv094PPK306Xlynt8xhW6aWWrL3DkJiy4Pmi1KZHQ3xt
zwIDAQABo0IwQDAdBgNVHQ4EFgQUVnNYZJX5khqwEioEYnmhQBWIIUkwDgYDVR0P
AQH/BAQDAgGGMMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEMBQADggIBAC9c

mTz8Bl6MlC5w6tIyMY208FHVvArzZJ8HXtXBc2hkeqK5Duj5XYUtqDdFqij0lgVQ
YKlJfp/imTYpEORHap1VIDzYm/EDMrraQKFz6oOht0SmDpkBm+S8f74TlH7Kph52
gDY9hAaLMyZlbcP+nv4fjFg4exqDsQ+8FxG75gbMY/qB8oFM2gsQa6H61SilzwZA
Fv97fRheORKkU55+MkIQpiGRqRxoF3yEvJ+M0ejf5lG5Nkc/kLnHvALcWxxPDkjB
JYOcCj+esQMzEhonrPcibCTRAUH4WAP+JWgiH5paPHxsnnVI84HxZmduTILA7rpX
DhjvLpr3Etiga+kFpaHpaPi8TD8SHkXoUsCjvxInebnMMTzD9joiFgOgyY9mpFui
TdaBJQbpdqQACj7LzTWb4OE4y2BThihCQRxEV+ioratF4yUQvNs+ZUH7G6aXD+u5
dHn5HrwdVw1Hr8Mvn4dGp+smWg9WY7ViYG4A++MnESLn/pmPNPW56MORcr3Ywx65
LvKRRRFHQV80MNNVIIb/bE/FmJUNS0nAiNs2fxBx1IK1jcmMGDw4nztJqDby1ORrp
0XZ60Vzk50lJLVU3aPaAaOpg+VBeHVOMmJ1CJeyAvP/+oYtKR5j/K3tJPsmPmRmAY
QqsZKbrAKbkTidOIjJlBO8n9pu0f9GBj39ItVQGL

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDcjCCAlqgAwIBAgIUZvnHwa/swlG07VOX5uaCwysckBYwDQYJKoZIhvcNAQEL
BQAwUTELMAkGA1UEBhMCSlAxIzAhBgNVBAoTGkN5YmVydHJ1c3QgSmFwYW4gQ28u
LCBMDGQuMR0wGwYDVQQDExRTZW51cmVtaWduIFJvb3QgQ0ExMjAeFw0yMDA0MDgw
NTM2NDZaFw00MDA0MDgwNTM2NDZaMFExCzAJBgNVBAYTAkpQMSMwIQYDVQQKExpD
eWJlcnRydXN0IEphcGFuIENvLiwgTHRKLjEdMBsGA1UEAxMUU2VjdXJlU2lnbiBS
b290IENBMTIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC6OcE3emhF
KxS06+QT61d1I02PJC0W6K6OyX2kVzsqdiUzg2zqMoqUm048luT9Ub+ZyZN+v/mt
p7JIKwccJ/VMvHASd6SFVLX9kHrko+RRWAPNEHl57muTH2SOa2SroxPjcf59q5zd
J1M3s6oYwlkm7Fsf0uZlfo+TvdhYXAvA42VvPMfKWeP+bl+sg779XSVOKik71gur
FzJ4pOE+IEa+Ym6b3kaosRbnhW70CEBFEEaCeVESE99g2zvVQR9wsMJvuWPWW0v4J
hscGwa5Pro4RmHvzC1KqYiaqlD+OJTN5lxZJfU+1UefNzFJM3IFTQy2VYzxV4+K
h9GtxRESOaCtAgMBAAGjQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAAQH/BAQD
AgEGMB0GA1UdDgQWBRRXNPN0zwRL1SXm8UC2LEzZLemgrTANBgkqhkiG9w0BAQsF
AAOCAQEAPrvbFxbS8hQBICw4g0utvsqFepq2m2um4fylOqyttCg6r9cBg0krY6Ld
mmQOmFvx3Y67ilQILUoT865AQ9tPkbeGGuWAtEGBpE/6aouIs3YIcipJQMPTw4WJ
mBClnW8Zt7vPemVV2zfrPIpyMpcemik+rY3moxtt9XUa5rBouVui7mlHJzWhhpmA
8zNL4WukJsPvdFseqJkth5Ew1DgDzk9qTPxpfpSVWKErI4cqclavTc7bgoitPQV
55FYxTpE05Uo2cBl6XLK0A+9H7MV2anjPecJnuDLN/v9vZfVvhgaaaI5gdka9at/
yOPiZwud9AazqVN/Ssq+xlVeg37xEHA==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFcjCCA1qgAwIBAgIUZNtaDCBO6Ncpd8hQJ6JaJ90t8sswDQYJKoZIhvcNAQEM
BQAwUTELMAkGA1UEBhMCSlAxIzAhBgNVBAoTGkN5YmVydHJ1c3QgSmFwYW4gQ28u
LCBMDGQuMR0wGwYDVQQDExRTZW51cmVtaWduIFJvb3QgQ0ExNDAAeFw0yMDA0MDgw
NzA2MTlaFw00NTA0MDgwNzA2MTlaMFExCzAJBgNVBAYTAkpQMSMwIQYDVQQKExpD
eWJlcnRydXN0IEphcGFuIENvLiwgTHRKLjEdMBsGA1UEAxMUU2VjdXJlU2lnbiBS
b290IENBMTQwggEiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDF0nqh1oq/
FjHqMNE6lPxauG4iwWL3pwon71D2LrGeaBLwbCRjOfHw3xDG3rdSINVSW0KZnvOg
vIfX8xbacuUKLBl422+JX1sLrcneC+y9/3OPJH9aaakpUqYllQC6KxNedlsmGy
6pJxaeQp8E+BgQQ8sqVb1MWOwWd7VRxJq3qdwudzTe/NCcLEVxLbAQ4jeQkHO6Lo
/IrPj8BGJJw4J+CDnRugv3gVEOuGTgpa/d/aLIJ+7sr2KeH6caH3iGicnPCNvg9J

kdjqOvn90Ghx2+m1K06Ckm9mH+Dw3EzsythqunQG+bOEKJTRX45zGRBdAuVwpcAQ
0BB8b8VYSbSbWbprafZX1zNoCr7gsfXmPvkPx+SgojQlD+Ajda8iLLCSxjVIHvXib
y8posqTddEx5YMaZ0ZPxMBoH064iwurO8YQJzOAUbn8/ftKChazcqRZOhaBgy/ac
18izju3Gm5h1DVXoX+WVWwKkrkMpKBGk5hIwAUt1ax5mnXkvpXYvHUC0bcl9eQjs
0Wq2XSqypWa9a4X0dFbD9ed1UigspF9mR6XU/v6eVL9lfgHwMI+lNpyiUBzuOIA
SMbHdPTGrMNASRZhdCyvjG817XsYAFs2PJxQDcqSMxDxJklt33UkN4li1+iW/RVL
ApY+B3KVfqs9TC7XyvDf4Fg/LS8EmjijAQIDAQABo0IwQDAPBgNVHRMBAf8EBTAD
AQH/MA4GA1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUBpOjCl4oaTeqYR3r6/wtbyPk
86AwDQYJKoZIhvcNAQEMBAQDggIBAJaAcgkGfpzMkwQWu6A6jZJOtxEaCnFxEM0E
rX+IRVAQZk5KQaID2RFPeje5S+LGjzJmdSX7684/AykmbjgWHfYfM25I5uj4V7Ib
ed87hwriZLoAymzvftAj63iP/2SbNDefNWWipAA9EiOWWF3KY4fGoweITedpdpT
zfFP7ELyk+OZpDc8h7hi2/DsHzc/N19DzFGdtfCXwreFamGLRB7IUe6TzktuhsHS
DCRZNhqlfJGP4xjblJUK7ZGqDpncllPjYYPGFrojutzdFhrGe0K22VoF3Jpf1d+4
2kd92jjbrDnVHmtsKheMYc2xbXIBw8MgAGJoFjHVdqqGuw6qnsb58Nn4DSEC5MUo
FlkRudlpcyqSeLiSV5sI8jrlL5WwWLdrIBRtFO8KvH7YVdiI2i/6GaX7i+B/OfVY
K4XELKzvGUWSTLNB9xNH27SgRNcmvMSZ4PPmz+Ln52kuaiWA3rF7iDeM9ovnhp6
dB7h7sxaOgTdsxoEqBRjrLdHEoOabPXm6RUVkRqEGQ6UROcSjiVbgGcZ3GOTEAtl
Lor6CZpO2oYofaphNdgOpygau1LgePhsumywbmHXumZNTfxPWQrqaA0k89jL9WB
365jJ6UeTo3cKXhZ+PmhIIynJkBugnLNeLLIjzwec+fBH7/PzqUqm9tEZDKgu39c
JRNIx+S

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICIZCCAamGAwIBAgIUfhXHw9hJp75pDIqI7fBw+d23PocwCgYIKoZlZj0EAwMw
UTELMAkGA1UEBhMCSlAxIzAhBgNVBAoTGKn5YmVydHJ1c3QgSmFwYW4gQ28uLCBM
dGQuMR0wGwYDVQQDExRTZWV1cmVtaWduIFJvb3QgQ0ExNTAeFw0yMDA0MDgwODMy
NTZaFw00NTA0MDgwODMyNTZaMFExCzAJBgNVBAYTAkpQMSMwIQYDVQQKEExpDeWJl
cnRydXN0IEphcGFuIEVhNvLiwgTHRkLjEdMBsGA1UEAxMUU2VjdXJlU2lnbiBSb290
IENBMTUwdjAQBgcqhkiOPQIBBgUrgGQAIGNiAAQLUHSNZDKZmbPSYAi4Io5GdCx4
wCtELW1fHcmuS1Iggz24FG1Th2CeX2yF2wYUleDHKP+dX+Sq8bOLbe1PL0vJSpSR
ZHX+AezB2Ot6IHhWGENfa4HL9rzatAy2KZMIaY+jQjBAMA8GA1UdEwEB/wQFMAMB
Af8wDgYDVR0PAQH/BAQDAgEGMB0GA1UdDgQWBbTrQciu/NWeUUj1vYv0hyCTQSVt
9DAKBggqhkiOPQQAawNoADBlAjEA2S6Jf15OpBEHvVnCB96rMjhTKkZEBhd6zlHp
4P9mLQlO4E/0BdGF9jVg3PVys0Z9AjBEmEYagoUeYWmJSwdLZrWeqrqgHkHZAXQ6
bkU6iYAZezKYVWOr62Nuk22rGwlgMU4=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDuDCCAqCgAwIBAgIQDPCOXAgWpa1Cf/DrJxhZ0DANBgkqhkiG9w0BAQUFADBI
MQswCQYDVQQGEwJVUzEgMB4GA1UEChMXU2VjdXJlVHJ1c3QgQ29ycG9yYXRpb24x
FzAVBgNVBAMTDINiY3VyZVRydXN0IENBMB4XDTA2MTEwNzE5MzExOFoXDTE5MTIz
MTE5NDA1NVowSDELMakGA1UEBhMCVVMxIDAeBgNVBAoTF1NiY3VyZVRydXN0IENv
cnBvcnF0aW9uMRcwFQYDVQQDEw5TZWN1cmVUcnVzdCBDQTCCASlwdQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBAKukgeWVzfX2FI7CT8rU4niVWJxB4Q2ZQCQXOZEz
Zum+4YOvYlyJ0fwkW2Gz4BERQRwdbvC4u/jep4G6pkjGnx29vo6pQT64lO0pGtSO
0gMdA+9tDWccV9cGrcri9f4Or2YlSASWC12juhbDCE/RRvgUXPLIXgGZbf2IzIao

wW8xQmxSPmjL8xk037uHGFaAJSTQ3MBv396gwpEWoGQRS0S8Hvbn+mPeZqx2pHGj
7DaUaHp3pLHnDi+BeuK1cobvomuL8A/b01k/unK8RCSc43Oz969XL0Imnal0ugBS
8kvNU3xHCzaFDmapCJcWNfFBZveA4+1wVMeT4C4oFVmHursCAwEAAaOBnTCBmjAT
BgkrBgEEAYI3FAIEBh4EAEMAQTALBgNVHQ8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB
/zAdBgNVHQ4EFgQUQjK2FvoE/f5dS3rD/fdMQB1aQ68wNAYDVR0fBC0wKzApoCeg
JYYjaHR0cDovL2NybC5zZWV0cnVzdC5jb20vU1RDQS5jcmwwEAYJKwYBBAGC
NxUBBAMCAQAwDQYJKoZIhvcNAQEFBQADggEBADdtT0rhWDpSclu1pqNIGKa7UTt3
6Z3q059c4EVlew3KW+JwULKUBRSuSceNQqcSc5R+DCMh/bwQf2AQWnL1mA6s7Ll/
3XpvXdMc9P+IBWlCqQVxyLesJugutIxq/3HcuLHfmbx8IVQr5Fiiu1cprp6poxkm
D5kuCLDv/WnPmRoJjeOnnyvJNjR7JLN4TJUXpAYmHrZkUjZfYGFZnMUFdAvnZyPS
CPyI6a6Lf+Ew9Dd+/cYy2i2eRDAwbO4H3tI0/NL/QPZL9GZGBISm8jIKYyYwa5vR
3ItHuuG51WLQoqD0ZwV4KWMabwTW+MZMo5qxN7SN5ShLHZ4swrhovO0C7jE=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDvDCCAqSgAwIBAgIQB1YipOjUiolN9BPI8PjqpTANBgkqhkiG9w0BAQUFADBK
MQswCQYDVQQGEwJVUzEgMB4GA1UEChMXU2VjdXJlVHJ1c3QgQ29ycG9yYXRpb24x
GTAXBgNVBAMTEFNiY3VyZSBHbG9iYWwgQ0EwHhcNMdYxMTA3MTk0MjI0WWhcNMjYx
MjMxMTk1MjA2WjBKMqswCQYDVQQGEwJVUzEgMB4GA1UEChMXU2VjdXJlVHJ1c3Qg
Q29ycG9yYXRpb24xGTAXBgNVBAMTEFNiY3VyZSBHbG9iYWwgQ0EwggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQCvNS7YrGxVaQZx5RNoJLNP2MwhR/jxYDiJ
iQPpvepeRlMJ3Fz1Wuj3RSoC6zFh1yzTM7HfAo3fg+6MpjhHZevj8feyTiW89sa
/FHtaMbQbqR8JNGuQsiWUGMu4P51/pinX0kuleM5M2SOHqRfkNjNPLlZ/kG5VacJ
jnIFHovdRIWCQtBJwB1g8NEXLJXr9qXBkqPFwqcIYA1gBBCWeZ4WNOaptvolRTnI
HmX5k/Wq8VLcmZg9pYYaDDUz+kulBAYVHDGA76oYa8J719rO+TMg1fW9ajMtgQT7
sFzUnKPiXB3jqUJ1XnvUd+85VLrJChgbEplJL4hL/VBi0XPnj3pDAgMBAAGjZ0w
gZowEwYJKwYBBAGCNxQCBAYeBABDAEEwCwYDVR0PBAQDAgGGMA8GA1UdEwEB/wQF
MAMBAf8wHQYDVR0OBBYEfK9EBMJBfkiD2045AuzshHrmzsmkMDQGA1UdHwQtMCsw
KaAnoCWGI2h0dHA6Ly9jcmwuc2VjdXJldHJ1c3QuY29tL1NHQ0EuY3JsMBAGCSsG
AQQBgjcVAQQDAgEAMA0GCSqGSIb3DQEBAQUAA4IBAQBJGghAfaReUw132HquHw0L
URYD7xh8yOOvaliTfGCRsoTciE6+OYo68+aCiV0BN7OrJKQVDpI1WkpEXk5X+nXO
H0jOZvQ8QCaSmGwb7iRGDBezUqXbpZGRzzfTb+cnCDpOGR86p1hcF895P4vkp9Mm
I50mD1hp/Ed+stCni5O/KU9DaXR2Z0vPB4zmAve14bRDtUstFJ/53CYNv6ZHdAbY
iNE6KTCEztI5gGlbqMdXSbxqVVFuFuq+NQfk1XWYN3kwFNspnWzFacxHVAIw98xc
f8LDmBxrThaA63p4ZUWiABqvDA1VZDRIuJK58bRQKfJPIx/abKwfROHdI3hRW8cW
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICODCCAb6gAwIBAgIJANZdm7N4gS7rMAoGCCqGSM49BAMDMGExCzAJBgNVBAYT
AkpQMSUwIwYDVQQKExxTRUNPTSBUCnVzdCBTeXN0ZW1zIENPLixMVEQuMSswKQYD
VQQDEyJTZWV0cD0eSBDb21tdW5pY2F0aW9uIEVDQyBSb290Q0ExMB4XDTE2MDYx
NjA1MTUyOFoXDTM4MDEwODA1MTUyOFowYTELMakGA1UEBhMCSIAxJTAjBgNVBAoT
HFNFQ09NIFRydXN0IFN5c3RlbXMgQ08uLExURC4xKzApBgNVBAMTIiNiY3VyYXR5
IENvbW11bmliYXRpb24gRUNDIjFvbnR0QTEwdjAQBgcqhkiOPQIBBgUrgQQAIGNi
AASKpW9gAwPDvTH00xecK4R1rOX9PVdu12O/5gSJko6BnOPpR27KkBLIE+Cnnfdl
dB9sELL05OnvbYUymUSxXv3MdhDYW72ixvnWQuRXdtyQwjWpS4g8EkdtXP9JTxpK

ULGjQjBAMB0GA1UdDgQWBBSGHOf+LaVKiwj+KBH6vqNm+GBZLzAOBgNVHQ8BAf8E
BAMCAQYwDwYDVR0TAQH/BAUwAwEB/zAKBggqhkiOPQQDAwNoADBlAjAVXUI9/Lbu
9zuxNuie9sRGKEkz0FhDKmMpzE2xtHqiuQ04pV1IKv3LsnNdo4gIxwwCMQDAqy0O
be0YottT6SXbVQjgUMzfRGEWgqtJsLKB7HOHeLRMsmIbEvoWTSVLY70eN9k=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDdzCCAl+gAwIBAgIBADANBgkqhkiG9w0BAQsFADBDMQswCQYDVQQGEwJKUDEI
MCMGA1UEChMcU0VDT00gVHJ1c3QgU3lzdGVtcyBDTy4sTFRELjEnMCUGA1UECXM
U2VjdXJpdHkgQ29tbXVuaWNhdGlvbiBSb290Q0EyMB4XDTA5MDUyOTA1MDAzOV
DTI5MDUyOTA1MDAzOVowXTELMaKGA1UEBhMCSlAxJTAjBgNVBAoTHFNFRQ09NIFRy
dXN0IFN5c3RlbXMgQ08uLExURC4xJzAlBgNVBAstHINiY3VyaXR5IENvbW11bmlj
YXRpb24gUm9vdENBMjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANAV
OVKxUrO6xVmCxF1SrjpDZYBLx/KWvNs2l9amZIyoXvDjChz335c9S672XewhtUGr
zbl+dp+++T42NKA7wfYxEUV0kz1XgMX5iZnK5atq1LXaQZAQwdbWQonCv/Q4EpVM
VAX3NuRFg3sUZdbcDE3R3n4MqzvEFb46VqZab3ZpUql6ucjrappdUtAtCms1FgkQ
hNBqyjoGADdH5H5XTz+L62e4iKrFvlnVspHEfbmwhRkGeC7bYRr6hfVKkaHnFtWO
ojnflLhwHyg/i/xAXmODPIMqGplrz95Zajv8bxbXH/1KEOtOghY6rCcMU/Gt1SSw
awNQwS08Ft1ENCcadfsCAwEAAaNCMEAwhQYDVR0OBByEFAqFqXdlBZh8QIH4D5cs
OPEK7DzPMA4GA1UdDwEB/wQEAwIBBjAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3
DQEBChUAA4IBAQBMOqNerLlFscEtfsgLcKlFZOoc7llsCLqJX2rKSpWeeo8HxdpF
coJxDjrSzG+ntKEju/Ykn8sX/oymzsLS28yN/HH8AynBbF0zX2S2ZTuJbxb2ePXC
okgfGT+Ok+vx+hfuzU7jBBJV1uXk3fs+BXziHV7Gp7yXT2g69ekuCkO2r1dcYmh8
t/2jioSgrGK+KwmHNBPqAbubKVY8/gA3zyNs8U6qtnRGEmyR7jTV7JqR50S+kDFy
1UkC9gLI9B/rfNmWVn/7lr5mUf/NVoCqgTLiluHcSmRvaS0eg29mvVXIwAHIRc/
SjnRBukLp7Y3gaVdjKozXoEofKd9J+sAro03
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIID3TCCAsWgAwIBAgIBADANBgkqhkiG9w0BAQsFADCBjzELMAKGA1UEBhMCMVVMx
EDAOBgNVBAgTB0FyaXpvbmExEzARBgNVBAcTCINjb3R0c2RhbGUxJTAjBgNVBAoT
HFN0YXJmaWVsZCBUZWNobm9sb2dpZXMsIEluYy4xMjAwBgNVBAMTKVN0YXJmaWVs
ZCB5b290IENlcnRpdHkgYXRlIEF1dGhvcml0eSAtIEcyMB4XDTA5MDkxMTAwMDAw
MFoXDzMTMTIzMTIzNTk1OVowY8xXzAlBgNVBAYTAiVtMRAwDgYDVQQGEwBcm16
b25hMRMwEQYDVQQHEwPTI90dHNkYWxlMSUwIwYDVQQKEwTdGFyZmlbGQgVGvJ
aG5vbG9naWVsZCBjbmMuMTIwMAZDVQQDEylTdGFyZmlbGQgUm9vdCBDZSJ0aWZp
Y2F0ZSBBDXR0b3JpdHkgLSBHMjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBAL3twQP89o/8ArFvW59I2Z154qK3A2FWGMNHttfKPTUuiUP3oWmb3ooa/RMg
nLRJdzlpVv257IzdIvpy3Cdh1+72WoTsbhm5iSzcHfVvDptrX8WJpRBSiUZV9Lh1
HOZ/5FSuS/hVclCGfgXcVnrHigHdMWdSL5stPSksPNkN3mSwOxGXn/hbVNMYq/N
Hwtjuzqd+/x5AJhhdM8mgkBj87JyahkNmcrUDnXMN/uLicFZ8WJ/X7NfZTD4p7dN
dloedl40wOiWVpmKs/B/pM293DIxfJHP4F8R+GuqSVzRmZTRouNjWwl2tVZi4Ut0
HZbUJtQIBFnQmA4O5t78w+wfkPECAwEAAaNCMEAwhQYDVR0TAQH/BAUwAwEB/zAO
BgNVHQ8BAf8EBAMCAQYwHQYDVR0OBByEFHwMMh+n2TB/xH1oo2KooC6rB1snMA0G
CSqGSIb3DQEBChUAA4IBAQAARWfolTwNvlJk7mh+ChTnUdgWUXuEok21iXQnCoKjU
sHU48TRqneSfioYmUeYs0cYtbpUgSpIB7LiKZ3sx4mcujJUDJi5DnUox9g61DLu3

4jd/IroAow57UvtruzvE03lRTs2Q9GcHGcg8RnoNAX3FWOdt5oUwF5okxBDgBPfg
8n/Uqgr/Qh037ZTlZFkSIHc40zI+OIF1lnP6aI+xy84fhez6nH7PfrHxBy22/L/K
pL/QlwVKvOoYKAKQvVR4CSFx09F9HdkWsKlhPdAKACL8x3vLCWRFCztAgfd9fDL1
mMpYjn0q7pBZc2T5NnReJaH1ZgUufzkVqSr7UluOhWn0
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIID7zCCAtegAwIBAgIBADANBgkqhkiG9w0BAQSFADCBmDELMAkGA1UEBhMCMVVMx
EDAOBgNVBAgTB0FyaXpvbmExEzARBgNVBAcTClnjb3R0c2RhbGUxJTajBgNVBAoT
HFN0YXJmaWVsZCBUZWNobm9sb2dpZXMsIEluYy4xOzA5BgNVBAMTMIN0YXJmaWVs
ZCBTZXJ2aWNlcyBSb290IENlcnRpZmljYXRlIEF1dGhvcml0eSAtIEcyMB4XDTA5
MDkwMTAwMDAwMFoXDTM3MTIzMTIzNTk1OVowgZgxZAJBgNVBAYTAIVTMRAwDgYD
VQQIEwdBcm16b25hMRMwEQYDVQQHEwpTY290dHNkYWxlMSUwIwYDVQQKExxTdGFy
ZmllbGQgVGvjaG5vbG9naWVzLCBjb250YXJmaWVsZCBZbmMuMTswOQYDVQQDEzJtdGFyZmllbGQgU2Vy
dmljZXMgUm9vdCBDZXJ0aWZpY2F0ZSBDbXR0b3JpdHkgLSBHMjCCASIdQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBANUMOsQq+U7i9b4Zl1+OiFOxHz/Lz58gE20p
OsgPfTz3a3Y4Y9k2YKibXlwAgLIvWX/2h/klQ4bnaRtSmpDhcePYLQ1Ob/bISdm2
8xpWriu2dBTrz/sm4xq6HZYuajtYlIHVv8loJNwU4PahHQUw2eeBGg6345AWh1K
Ts9DkTvnVtYAcMtS7nt9rjrnvDH5RfbCYM8TWQIrgMw0R9+53pBlbQLPLJGmpufe
hRhJfGZOozptqbXuNC66DQO4M99H67FrjSXZm86B0UVGMpZwh94CDklDhbZsc7tk
6mFBrMnUVN+HL8cisibMn1lUaJ/8viovxFUcdUBgF4UCVTmLfwUCAwEAaNCMEA
wDwYDVR0TAAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFJxfAN+q
AdcwKziIorhtSpzyEZGDMA0GCSqGSIb3DQEBCwUAA4IBAQBLLNqaEd2ndOxmFZyMI
bw5hyf2E3F/YNoHN2BtBLZ9g3ccaaNnRbobbhiCPPE95Dz+I0swSdHynVv/heyNXB
ve6SbzJ08pGCL72CQnqtKrcgfU28elUSwhXqvfdqlS5sdJ/PHLTyxQGjhdByPq1z
qwubdQxtRbeOlKyWN7Wg0I8VRw7j6IPdj/3vQQF3zCepYoUz8jcI73HPdwbeYBkd
iEDPfuYd/x7H4c7/I9vG+o1VTqkC50cRRj70/b17KSa7qWFiNyi2LSr2EIZkyXCn
0q23KXB56jzaYyWf/Wi3MOxw+3WKt21gZ7IeyLnp2KhvAotnDU0mV3HaIPzBSICN
sSi6
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFujCCA6KgAwIBAgIJALtAHEP1Xk+wMA0GCSqGSIb3DQEBBQUAMEUxCzAJBgNV
BAYTAkNIMRUwEwYDVQQKEwxTd2lzc1NpZ24gQUcxHzAdBgNVBAMTFIN3aXNzU2ln
biBHb2xkIENBIC0gRzIwHhcNMDYxMDI1MDgzMDM1WhcNMzYxMDI1MDgzMDM1WjBF
MQswCQYDVQQGEwJDSDEVMBMGA1UEChMMU3dpc3NTaWduIEFHMR8wHQYDVQQDExZT
d2lzc1NpZ24gR29sZCBBDQSAAtIEcyMIICljANBgkqhkiG9w0BAQEFAAOCAg8AMIIC
CgKCAgEAr+TufoskDhJuqVAtFkQ7kpJcyrhhdhJCEyq8ZVeCQD5XJM1QiyUqt2/8
76LQwB8CJEoTlo8jE+YoWACjR8cGp4QjK7u9lit/VcyLwVcfDmJlD909Vopz2q5+
bbqBHH5CjCA12UNNhPqE21Is8w4ndwtrvxEvcnifLtg+5hg3Wipy+dpikJKVyh+c
6bM8K8vzARO/Ws/BtQpgvd21mWRTuKcWs2/iJneRjOBIEAKfNA+k1ZlZUd6+jbqE
emA8atufK+ze3gE/bk3lUIbLtK/tREDFylqM2tfrKjuvqblCqoOpd8FUrdVxyJd
MmqXl2MT28nbeTZ7hTpKxVKJ+STnnXepgv9VHKVxaSvRAiTysybUa9oEVeXBCsdt
MDeQKuSeFDNeFhdVxVu1yzSJkvGdJo+hB9TGsnhQ2wwMC3wLjEHXuendjIj3o02y
MszYF9rNt85mndT9Xv+9lz4pded+p2JYryU0pUHHpBwNUMoDAw8IWh+Vc3hiv69y
FGkOpeUDDniOJihC8AcLYiAQZzlG+qkDzAQ4embvIIO1jEpWjpEA/I5cgt6IoMPi

aG59je883WX0XaxR7ySArqpWl2/5rX3aYT+YdzylkBYcjCbaZaIJbCHiVOO5yKxM
gI93e2CaHt+28kgeDrpOVG2Y4OGiGqJ3UM/EY5LsRxmd6+ZrzsECAwEAAaOBRDCB
qTAOBgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUWyV7
lqRIUX64OfPAeGZe6Dm8O4wHwYDVR0jBBgwFoAUWyV7lqRIUX64OfPAeGZe6Dm
8O4wRgYDVR0gBD8wPTA7BgIghXQBWQECAQEWLjAsBggrBgEFBQcCARYgaHR0cDov
L3JlcG9zaXRvcnkuc3dpc3NzaWduLmNvbS8wDQYJKoZIhvcNAQEFBQADggIBACe6
45R88a7A3hfm5djV9VSwg/S7zV4Fe0+fdWavPOhWfvxyeDgD2StiGwC5+OlgzczO
UYrHUdFu4Up+GC9pWbY9ZIEr44OE5iKHjn3g7gKZYbge9LgriBIWhMIxkziWMaa5
O1M/wySTVltpkuzFwbs4AOPsF6m43Md8AYOfMke6UiI0HTJ6CVanfCU2qT1L2sCC
bwq7EsiHSycR+R4tx5M/nttfJmtS2S6K8RTGRI0Vqbe/vd6mGu6uLftIdxf+u+yv
GPUqUfA5hJeVbG4bwvYEdGB5JbAKJ9/fXtI5z0V9QkvfsywexcZdylU6oJxpmo/a
77KwPJ+HbBIRZXAVUjEaJM9vMSNQH4xPjyPDdEFjHFWoFN0+4FFQz/EbMFYOkRCC
hdiDyyJkvC24JdVUorgG6q2SpCSgwYa1ShNqR88uC1aVVMvOmttqtKay20Elhid3
92qgQmwLOM7XdVAyKsLfKzAiSNDVQTglXaTpXZ/GIHXQRf0wl0OPkKsKx4ZzYEpp
Ld6leNcG2mqeSz53OiATiGHQv2ieY2BrNU0LbbqhPcCT4H8js1WtcIVORvnSFu+w
ZMEBnunKoGqYDs/YYPiVsbjKQuE4NRb0yG5P94FW6LqjviOvr1vA+ACozB2+htt
Qc8Bsem4yWb02ybzOqR08kkkW8mw0FfB+j564ZfJ

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFkzCCA3ugAwIBAgIUQ/oMX04bgBhE79G0TzUfRPSA7cswDQYJKoZIhvcNAQEL
BQAwUTELMAkGA1UEBhMCQ0gxFTATBgNVBAoTDFN3aXNzU2lnbiBBRzErMCKGA1UE
AxMiU3dpc3NTaWduIFJTSBUTFMgUm9vdCBDQSAyMDIyIC0gMTAeFw0yMjA2MDgx
MTA4MjJaFw00NzA2MDgxMTA4MjJaMFExCzAJBgNVBAYTAkNIMRUwEwYDVQQKEwxT
d2lzc1NpZ24gQUcxKzApBgNVBAMTIlN3aXNzU2lnbiBSU0EgVExTIFJvb3QgQ0Eg
MjAyMiAtIDEwZG93aW50GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDLKmjiC8NX
vDVjvHCIO/OMPE5Xlm7DTjak9gLKHqquuN6orx122ro10JFwB9+zBvKK8i5VUXu7
LCTLf5ImgKO0lPaCoaTo+nUdWfMHamFk4saMla+ju45vVs9xzF6BYQ1t8qsCLqSX
5XH8irCRIFucdFJtrhUnWXjyCcplDn/L9Ovn3KIMd/YrFgSVrpxxpT8q2kFC5zyE
EPthPYxr4iurR1VPuFa+Rd4iUU1OKNlFgUEGjw5NBuBwQCMBauTLE5tzeE0USJlt
/m2n+IdreXXhvhCxqohAWVTXz8TQm0SzOGlkjIHRI36qOTw7D59Ke4LKa2/KIj4x
0LDQKhySio/YGZxH5D4MucLNvkEM+KRHBdvBFzA4OmnczcNpI/2aDwLOEGrOyvi5
KaM2iYauC8BPY7kGWUleDsFpswrzd34unYyzJ5jSmY0lpx+Gs6ZUCdj8fV3oT4MM
0ZPIEuRU2j7yrTrePjxF8CgPBmnh25d7mUWe3f6VWQQvdT/TromZhqwUtKiE+shd
OxtYk8EXIFXIC+OCeYSf8wCENO7cMdWP8vpPlkwGqnj73mSiI80fPsWMvDdUDrta
clXvyFu1cvh43zcgTFeRc5JzrBh3Q4IgaezprClG5QtO+DdziZaKHG29777YtvTK
wP1H8K4LWCDfYB02rpeNUIMmJCn3nTsPBQIDAQABo2MwYTAPBgNVHRMBAf8EBTAD
AQH/MA4GA1UdDwEB/wQEAwIBBjAFBgNVHSMEGDAWgBRvjmKlK0Ow4UD2p8P98Q+4
DxU4pTAdBgNVHQ4EFgQUb45ii5NDsOFA9qfD/fEPuA8VOKUwDQYJKoZIhvcNAQEL
BQADggIBAKwsKUF9+lz1GpUYvyypiqkkVHX1uECry6gkUSsYP2OprphWKwVDIqO3
10aewCoSPY6WlkDfDDOLazeROpW7OSltwAJsipQLBwJNGD77+3v1dj2b9l4wBlgz
Hqp41eZUBDqyggmNzhYzWUu08aWjlw5DI/0LIICQ/+Mmz7hkkeUFjxOgdg3XNwwQ
iJb0Pr6VvfHdfCjw3lHC1ySFWPtUnWK50Zpy1FVCypM9fJkT6lc/2cyjUUtMoIc
gC9qkfjLvH4YoiaolQNTKIftV+Vlek4ASltOU8liNr3CjlvzrG4ngRhZi0Rjn9UM
ZfQpZX+RLOV/fuiJz48gy20HQhFRJjKKLjPHE7iNvUcNCfAWpO2Whi4Z2L6MOuhF
LhG6rlrnub+xzl/goP+4s9GFe3lmozM1O2bYQL7Pt2eLSmkZJVX8vY3PXtpOpvJp
zv1/THfQwUY1mFwjmwJFQ5Ra3bxHrSL+ul4vkSkphnsh3m5kt8sNjzdbowhq6/Td

Ao9QAwKxuDdollDruF/UKIqlIgyKhPBZLtU30WHlQnNYKoH3dtvi4k0NX/a3vgW0
rk4N3hY9A4GzJl5LuEsAz/+MF7psYC0nhzck5npgL7XTgwSqT0N1osGDsieYK7EO
gLrAhV5Cud+xYJHT6xh+cHiudoO+cVrQkOPKwRYlZ0rwtnu64ZzZ
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDwzCCAaquAwIBAgIBATANBgkqhkiG9w0BAQsFADCBgjELMAkGA1UEBhMCREUx
KzApBgNVBAoMIlQtU3lzdGVtcyBFbnRlcnByaXNlIFNlcnZpY2VzIEdtYkgxHzAd
BgNVBAsMF1QtU3lzdGVtcyBUcnVzdCBDZW50ZXIwJTAjBgNVBAMMHFQtVGVSZVNl
YyBHbG9iYWxSb290IENsYXNzIDIwHhcNMDgxMDAxMTA0MDE0WhcNMzMxMDAxMjM1
OTU5WjCBgjELMAkGA1UEBhMCREUxKzApBgNVBAoMIlQtU3lzdGVtcyBFbnRlcnBy
aXNlIFNlcnZpY2VzIEdtYkgxHzAdBgNVBAsMF1QtU3lzdGVtcyBUcnVzdCBDZW50
ZXIwJTAjBgNVBAMMHFQtVGVSZVNlYyBHbG9iYWxSb290IENsYXNzIDIwggEiMA0G
CSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCqX9obX+hzkeXaXPSi5kfl82hVYAud
AqSzm1nzHoqyNK38DcLZSBnuaY/JIPwhqgcZ7bBcrGXHX+0CfHt8LRvWurmAwhiC
FoT6ZrAlxlQjgeTNUk/9k9uN0goOA/FvudocP05l03Sx5iRUKrERLMjfTIH6VJi
1hKTXrcxlkIF+3anHqP1wvzpesVsqXFP6st4vGCvx9702cu+fjOlbpSD8DT6Iavq
jnKgP6TeMFvvhk1qlVtDRKgQFRzLAVfFmPHmBiiRqiDfT1MmUUOyCxGVWOHAD3bZ
wI18gfNycJ5v/hqO2V81xrJvNH+SE/iWjnX2J14np+GPgNeGYtEotXHAzMBAAGj
QjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVROPAQH/BAQDAgEGMB0GA1UdDgQWBBS/
WSA2AHmgoCJrjNXyYdK4LMuCSjANBgkqhkiG9w0BAQsFAAOCAQEAMQoiYQsfDohy
NsZt+U2e+iKo4YFWWz827n+qrkRk4r6p8FU3ztqONpfSO9kSpp+ghla0+AGIWIPAC
uvxhI+YzrmB6azZie60EI4RYZeLbK4rnJVM3YINfvNoBYimipid5xjoifsFvHZVw
IEoHNN/q/xWA5brXethbdXwFeilHfkCoMRN3zUA7tFFHei4R40cR3p1m0IvVVGb6
g1XqfMIpiRvpb7PO4gWEyS8+eIVibslfwXhjdFjASBgMmTnrMwatXlajRWc2BQN
9noHV8cigwUtPJsLj0Ys6lDfMjIq2SPDqO/nBudMNva0Bkuqjzx+zOAduTNRlP
BSeOE6Fuwg==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDwzCCAaquAwIBAgIBATANBgkqhkiG9w0BAQsFADCBgjELMAkGA1UEBhMCREUx
KzApBgNVBAoMIlQtU3lzdGVtcyBFbnRlcnByaXNlIFNlcnZpY2VzIEdtYkgxHzAd
BgNVBAsMF1QtU3lzdGVtcyBUcnVzdCBDZW50ZXIwJTAjBgNVBAMMHFQtVGVSZVNl
YyBHbG9iYWxSb290IENsYXNzIDMwHhcNMDgxMDAxMTA0MDE0WhcNMzMxMDAxMjM1
OTU5WjCBgjELMAkGA1UEBhMCREUxKzApBgNVBAoMIlQtU3lzdGVtcyBFbnRlcnBy
aXNlIFNlcnZpY2VzIEdtYkgxHzAdBgNVBAsMF1QtU3lzdGVtcyBUcnVzdCBDZW50
ZXIwJTAjBgNVBAMMHFQtVGVSZVNlYyBHbG9iYWxSb290IENsYXNzIDMwggEiMA0G
CSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC9dZPwYiJvJK7genasfb3ZJNW4t/zN
8ELg63iIVl6bmlQdTQyK9tPPcPRstdiTBONGhnFBSivwKixVA9ZIw+A5OO3yXDw/
RLyTPWGrTs0NvvAgJ1gORH8EGoel15YUNpDQSXuhdfsaa3Ox+M6pCSzyU9XDFES4
hqX2iys52qMzVNn6chr3IhUciJFrf2blw2qAsCTz34ZFIP0Zf3WHHx+xGwpzJFu5
ZeAsVMhg02YXP+HMVDNzkQI6pn97djmIH5a2OK61yJN0HZ65tOVgnS9W0eDrXltM
EnAMbEQgqxHY9Bn20pxSN+f6tsIxO0rUFJmtxxr1XV/6B7h8DR/Wgx6zAgMBAAGj
QjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVROPAQH/BAQDAgEGMB0GA1UdDgQWBBS1
A/d2O2GCahKqGFPrAyGUv/7OyJANBgkqhkiG9w0BAQsFAAOCAQEAVj3vINW92nOy
WL6ukK2YJ5f+AbGwUgC4TeQbIXQbfsDuXmkqJa9c1h3a0nnJ85cp4IaH3gRZD/FZ

Xtuq3VTpMCEobY5583WSjCb+3MX2w7DfRFIDo7YDKPYIMKoNM+HvnKkHIuNZW0CP
2oi3aQiotyMuRAIZN1vH4xfyIutuOVLf3lSnmMILIJXcRolftBL5hSmO68gnFSDA
S9TMfAxsNAwmmYypjyn9tnQS6Jk/zuZQXLB4HCX8SS7K8R0IrGsayIyJNN4KsDA
oS/xUgXJP+92ZuJF2A09rZXIx4kmyA+upwMu+8Ff+iDhcK2wZSA3M2Cw1a/XDBzC
kHDXShi8fgGwsOsVHkQGzaRP6AzRwyAQ4VRlnrZR0Bp2a0JaWHY06rc3Ga4udfmW
5cFZ95RXKSWNOkYrTZpB0F8mAwIDAQABo2MwYTAOBgNVHQ8BAf8EBAMCAQYwDwYD
VR0TAQH/BAUwAwEB/zAfBgNVHSMEGDAWgBSdhWEUfMFib5do5E83QOGt4A1WNzAd
BgNVHQ4EFgQUUnYVhFHZBYm+XaORPN0DhreANVjcwDQYJKoZIhvcNAQEMBAQDggIB
AGSPesRiDrWiZlJHhg6hShbNcAu3p4ULs3a2D6f/CIsLJc+o1IN1KriWiLb73y0t
tGlTITVX1olNc79pj3CjYcya2x6a4CD4bLubIp1dhDgaLlrdqHXKGNk/nZVekZn
68xDiBaiA9a5F/gZbG0jAn/xX9AKKSM70aoK7akXJlQKTcKlTfjF/biBzysseKNn
TKkHmvPfXvt89YnNdJdhEGoHK4Fa0o635yDRIG4kqIQnoVesqlVYL9zZyvpO BJ7t
RCT5dEA7IzOrg1oYJkK2bVS1FmAwBLGg+LhBoF1JSdJlBTrq/p1hvIbZv97Tujqx
f36SN17JAG7cmL3c7IAFrQI932XtCwP39xaEBDG6k5TY8hL4iuO/Qq+n1M0RFxbI
Qh0UqEL20kCGoE8jypZFVmAAGzbdVAaYBlGX+bgUJurSkquLvWL69J1bY73NxW0Qz
8ppy6rBePm6pUlvscG21h483XjyMnM7k8M4MZ0HMzvaAq07MTFb1wWFZk7Q+ptq4
NxKfKjLji7gh7MMrZQzvIt6IKTtM1/r++t+FHvpw+PoP7UV31aPcuIXcv/Fa4nzX
xeSDwWrruoBa3lwtcHb4yOWHh8qgnaHllhInD0Q9HWzq1MKLL295q39QpsQZp6F6
t5b5wR9iWqJDB0BeJsas7a5wFsWqynKKTbDPAYsDP27X

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIQTCCAymgAwIBAgICDL4wDQYJKoZIhvcNAQELBQAUTELMAkGA1UEBhMCVFcx
EjAQBgNVBAoTCVRBSVdBt1DQTEQMA4GA1UECzMUM9vdCBDQTEcMBoGA1UEAxMT
VFdDQSBHbG9iYWwgUm9vdCBDQTAeFw0xMjA2MjcwNjI4MzNaFw0zMDEyMzExNTU5
NTlaMFEExCzAJBgNVBAYTAiRlRMRiEAYDVQQKEwUQUiXQU4tQ0ExEDAOBgNVBAST
B1Jvb3QgQ0ExHDAaBgNVBAMTE1RXQ0EgR2xvYmFsIFJvb3QgQ0EwgglMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQCwBdvI64zEbooh745NnHEKH1Jw7W2CnJfF
10xORUnLQEK1EjRsGcJ0pDFfhQKX7EMzClPSnIyOt7h52yvVavKOZsTuKwEHktSz
0ALfUPZVr2YOy+BHYC8rMjk1Ujoog/h7FsYYuGLWRyWRzvAZEK2tY/XTP3VfKfCh
MBwqoJimFb3u/Rk28OKRQ4/6ytYQJ0lM793B8YVwm8rqQFpD/G2Gb3PpN0Wp8DbH
zlh1HrtsBv+baz4X7GGqcXzGHal3SekVtTzWoWH1EfcFbx39Eb7QMAfCKbAJTibc
46KokWofwPFFiFzlmLhxpRUZyXx1EcxdwE8tmx2RRP1WKKD+u4ZqyPpcC1jcxkt2
yKsi2XMPpRaAok/T54igu6idFMqPVMnaR1sjjIsZAAmY2E2TqNGtz99sy2sbZCi
laLOz9qC5wc0GZbpuCGqKX6mOL6OKUohZnkfs8O1CWfe1tQHRvMq2uYiN2DLgYP
oA/pyJV/v1WRBXrPPRXAb94JlAGD1zQbzECl8LibZ9WYkTunhHiVJqRaCPgrdLQA
BDzfuBSO6N+pjWxnkjMdwLfs7JLlvgm/LCkFbwJrnu+8vyq8W8BQj0FwcYeyTbcE
qYSjMq+u7msXi7Kx/mzhklyIqJdlzshNy/MGz19qCkKxHh53L46g5pIOBvwFItIm
4TFRfTLcDwIDAQABoyMwITAObgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUwAwEB
/zANBgkqhkiG9w0BAQsFAAOCAgEAXzSBdu+WHdXltdkCY4QWwa6gcFGn90xHNcgL
1yg9iXHZqjNB6hQbbCEAwGxCGX6faVsgQt+i0trEfJdLjbDorMjupWkEmQqSpqsn
LhpNgb+E1HAerUf+/UqdM+DyucRFCCEK2mlpc3INvjT+lIutwx4116KD7+U4x6WF
H6vPNOw/KP4M8VeGTslV9xzU2KV9Bnpv1d8Q34FOIWWxtuEXeZVFBs5fzNxGiWNo
RI2T9GRwoD2dKAXDOXC4Ynsg/eTb6QihuJ49CcdP+yz4k3ZB3lLg4VfSnQO8d57+
nile98FRYB/e2guyLXW3Q0iT5/Z5xoRdgFlgPpx4mI88k1HtQJAH32RjJMtOcQWh
15QaiDLxInQirqWm2BJpTGCjAu4r7NRjkgtevi92a6O2JryPA9gK8kxkRr05YuWW
6zRjESjMlfGt7+/cgFhI6Uu46mWs6fyAtbXIRfmswZ/ZuepiiI7E8UuDEq3mi4TW

nsLrgxifarsbJGAzcMzs9zLzXNl5fe+epP7Jl8Mk7hWSsT2RTyaGvWZzJBPqpK5j
wa19hAM8EHIGG3njxPPyBJUgriOCxLM6AGK/5jYk4Ve6xx6QddVfP5VhK8E7zeWz
aGHQRIapIVJpLesux+t3zqY6tQMzT3bR51xUAV3LePTJDL/PEo4XLSNolOer/qmy
KwbQBM0=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDezCCAmOgAwIBAgIBATANBgkqhkiG9w0BAQUFADBfMQswCQYDVQQGEwJUVzES
MBAGA1UECgwJVEFJV0FOLUNBMRAwDgYDVQQQLDAdSb290IENBMSowKAYDVQQDDCFU
V0NBIFJvb3QgQ2VydGhmaWNhdGlvbiBBdXRob3JpdHkwHhcNMDgwODI4MDcyNDMz
WhcNMzAxMjMxMTU1OTU5WjBfMQswCQYDVQQGEwJUVzESMBAGA1UECgwJVEFJV0FO
LUNBMRAwDgYDVQQQLDAdSb290IENBMSowKAYDVQQDDCFUUV0NBIFJvb3QgQ2VydGlm
aWNhdGlvbiBBdXRob3JpdHkwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
AQcWfnK4pAOU5qfeCTiRShFAh6d8WWQe7UREN3+v9XAU1bihSX0NXIP+FPQQeFE
AcK0HMMxQhZHhTMidrIKbw/IJVBPhYa+v5guEGcevhEFhgWQxFnQfHgQsIBct+HH
K3XLfJ+utdGdIzdjp9xCoi2SBBtQwXu4PhvJVgSLL1KbralW6cH/ralYhzC2gfeX
RfwZVzsrB+RH9JIF/h3x+JejiB03HFyP4HYlmlD4oFT/RJB2I9IyxsOrBr/8+7/z
rX2SYgJbKdM1o5OaQ2RgXbL6Mv87BK9NQGr5x+PvI/1ry+UPizgN7gr8/g+YnzAx
3WxSZfmLgb4i4RxYA7qRG4kHAgMBAAGjQjBAMA4GA1UdDwEB/wQEAwIBBjAPBgNV
HRMBAf8EBTADAQH/MB0GA1UdDgQWBBrQOFsmjd6LWvJPelSDGRjjCDWmujaANBgkq
hkiG9w0BAQUFAAOCAQEAPNV3PdrfibqHDAhUaiBQkr6wQT25JmSDCi/oQMCXKCeC
MErJk/9q56YAf4lCmtYR5VPOL8zy2gXE/uJQxDqGfczafhAJO5I1KlOy/usrBdls
XebQ79NqZp4VKIV66IIArB6nCWlWQtNoURI+VJq/REG6Sb4gumlc7rh3zc5sH62D
lhh9DrUUOYTxKOkt0557HnpyWoOzeW/vtPzQCqVYT0bf+215WfKEllKuD8z7fDvn
aspHYcN6+NOSBB+4IIThNIQWx0DeO4pz3N/GCUzf7Nr/1FNCocnyYh0igzyXxfkZ
YiesZSLX0zzG5Y6yU8xJzrww/nsOM5D77dIUkR8Hrw==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICQjCCAcmgAwIBAgIQNjqWjMlcsIjN0AFdxeVXADAKBggqhkiG9w0BAQUFADAzBjMQsw
CQYDVQQGEwJERTENMCUGA1UECgweRGV1dHNjaGUgVGVSZWtvcSBTZW51cml0eSBH
bWJIMSswKQYDVQQDDCJUZWxla29tIFNlY3VyaXR5IFRlUyBFQ0MgUm9vdCAyMDIw
MB4XDTIwMDgyNTA3NDgyMFOXTDTQ1MDgyNTIzNTk1OVowYzELMAkGA1UEBhMCREUx
JzAlBgNVBAoMHkRldXRzY2hlIFRlbGVrb20gU2VjdXJpdHkgR21iSDErMCKGA1UE
AwwiVGVSZWtvcSBTZW51cml0eSBUTFMgRUNDIRJvb3QgMjAyMjB2MBAGByqGSM49
AgEGBSuBBAAiA2IABM6//leov9Wq9xCazbzREaK9Z0LMkOsVGJDZos0MKiXrPk/O
tdKPD/M12kOLAOc+b1EkHQ9rK8qfwm9QMuU3ILYg/4gND21Ju9sGpleQkpT0CdDP
f8iAC8GXs7s1J8nCG6NCMEAwHQYDVRO0BBYEFONyzG6VmUex5rNhTNHLq+O6zd6f
MA8GA1UdEwEB/wQFMAMBAf8wDgYDVRO0PAQH/BAQDAgEGMAoGCCqGSM49BAMDA2cA
MGQCMHVS7i7ekEE+uShCLsoRbQuHmKjYC2qBuGT8lv9pZMo7k+5Dck2TOrbRBR2Di
z6fLHgIwN0GMZt9Ba9aDAEH9L1r3ULRn0SyocddDypwnJJGDSA3PzfdUga/sf+Rn
27iQ7t0l

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFszCCA5ugAwIBAgIQIZxULej27HF3+k7ow3BXlZANBgqhkiG9w0BAQwFADBJ
MQswCQYDVQQGEwJERTEnMCUGA1UECgweRGV1dHNjaGUgVGVsZWtvbSBTZWN1cmI0
eSBHbWJIMSSwKQYDVQQDDCJUZWxla29tIFNlY3VyaXR5IFRmUyBSU0EgUm9vdCAy
MDIzMB4XDTIzMDMyODEyMTY0NV0XDTQ4MDMyNzIzNTk1OVowYzELMAkGA1UEBhMC
REUxJzAlBgNVBAoMHkRldXRzY2hlIFRlbGVrb20gU2VjdXJpdHkgR21iSDErMCKG
A1UEAwwiVGVsZWtvbSBTZWN1cmI0eSBUTFMgUINBIFJvb3QgMjAyMzCCAilwDQYJ
KoZihvcNAQEBBQADggIPADCCAgCggIBAO01oYGA88tKaVvC+1GDrib94W7zgRJ9
cUD/h3VCKSHtgVIs3xLBGYSJwb3FKNXVS2xE1kzbB5ZKVXrKNoIENqil/Cf2SfHV
cp6R+SPWcHu79ZvB7JPPGepIfohwoHP89v+1VmLhc2o0mD6CuKyVU/QBoCcHcqMA
U6DksquDOFcZJSfvgkdmOGjup5czQRxUX11eKvzWarE4GC+j4NSuHUaQTXtvPM6
Y+mpFEXX5ILRbtLevOP1Czvm4MS9Q2QTps70mDdsipWol8hHD/BeElvnHRz+sTug
BTNoBUGCwQmRAcnj02r6LX2zWtEtefdi+zqJbQAildNsLgYMcEWzv/9FIS3R/qy
8XDe24tsNlikfLMR0cN3f1+2JeANxdKz+bi4d9s3cXFH42AYTyS2dTd4uaNir73J
co4vzLuu2+QVUhhHM/tqty1LkCiCc/4YizWN26cEar7qwU02OxY2kTLvtkCJkUPg
8qKRCB7m8kwOFjQgrIfBLX7JZkcXFBGk8/ehJImr2BrIoVyx0/eMbcgByU/J7MT8
rFEZ0ciD0cmfHdRHNCk+y7AO+oMLKFJlKdw/fKifybYKu6boRhYPluV75Gp6SG12
mAWl3G0eQh5C2hrGUve1g8Aae3g1LDj1H/1Joy7SWWO/gLCMk3PLNaaZISJhZQNg
+y+TS/qanIA7AgMBAAGjYzBhMA4GA1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUtqeX
gj10hZv3PJ+TmPv5dVKMbUcwDwYDVR0TAQH/BAUwAwEB/zAfBgNVHSMEGDAWgBS2
p5eCPXSFm/c8n5OalXl1UoxRzANBgqhkiG9w0BAQwFAAOCAGEAqMxhpr51nhVQ
pGv7qHBFfLp+sVr8WyP6Cnf4mHGCDG3gXkaqk/QeoMPhk9tLrbKmXauw1GLLXrtm
9S3ul0A8Yute1hTWjOKWi0FpkzXmuZlrYrShF2Y0pmtjxrlO8iLpWA1WQdH6DErw
M807u20hOq6OcrXDSvvpfeWxm4bu4uB9tPcy/SKE8YXJN3nptT+/XOR0so8RYgDd
GGah2XsjX/GO1WfoVNpbOms2b/mBsTNHM3dA+VKq3dSDz4V4mZqTuXNnQkYRIer+
CqkbGmVps4+uFrb2S1ayLfmlYow7YqPt9BO1UAJpB+Y1zqlklkg5LB9zVtzaL1t
xKITDmcZu1IcFmwMmm6gJC3VRRvcxAIU/oVbZZfKTPBQCHpCNfnqwmBU+AGuHrS+
w6jv/naaoqYfRvaEf7fzbsQCzndILIyy7MMAo+wsVRjBfhnu4S/yrYObnqsZ38aK
L4x35bcF7DvB7L6Gs4a8wPfc5+pbrLmTWGS9DiP7bY+A4A7l3j941Y/8+LN+lJ
X273CXE2whJdV/LItM3z7gLfEdxquVeEHVlNjM7IDiPCtyaaEBRx/pOyiriA8A4Q
ntOoUAW3gi/q4Iqd4Sw5/7W0cwDk90imc6y/st53BIe0o82bNSQ3+pCTE4FCxpgm
dTdmQRCSu/WU48IxK63nI1bMNSWSs1A=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFODCCAYCgAwIBAgIRAJW+FqD3LkxbxzmCcvqLzZYwDQYJKoZIhvcNAQEFBQAww
NzEUMBIGA1UECgwlVGVSaWFTb25lcmExHzAdBgNVBAMMFIRlbGhU29uZXJhIFJv
b3QgQ0EgdjEwHhcNMDcxMDE4MTIwMDUwWhcNMzIxMDE4MTIwMDUwWjA3MRQwEgYD
VQQKDAUzWxpYVNvbWVyYTEfMB0GA1UEAwwVVGVSaWFTb25lcmEgUm9vdCBDQSB2
MTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgCggIBAMK+6yfwIaPzaSZVfp3F
VRaRXP3vIb9TgHot0pGMYzHw7CTww6XScnwQbfQ3t+XmfHnqjLWCi65ltqWA3GV1
7CpNX8GH9SBIK4GoRz6JI5UwFpB/6FcHSOcZrr9FZ7E3GwYq/t75rH2D+1665I+X
Z75Ljo1kB1c4VWk0Nj0TSO9P4tNmHqTPGrdeNjPUtAa9GAH9d4RQAEX1jF3oI7x+
/jXh7VB7qTCNGdMJmhnXb88lxhTuylixcepsHHltTbLaC0H2kd7OriUPEMPPCs
81Mt8Bz17Ww5OXOAFshSsCPN4D7c3TxHoLs1iuKYalu+5b9y7tL6pe0S7fyYGKkm
dtwoSxAgHNN/Fnct7W+A90m7UwW7XWjH1Mh1Fj+JWov3F0fUTPHSiXk+TT2YqGHe
Oh7S+F4D4MHJHlZtJ3TITazN19jY5szFPAAtJmtTfImMMsJu7D0hADnJoWjiUIMu
sDor8zagrC/kb2HCUQk5PotTubtn2txTuXZZNp1D5SDgPTJghSJrt8czu90VL6R4

pgd7gUY2BIbdeTXHlSw7sKMXNeVzH7RcWe/a6hBle3rQf5+ztCo3O3CLm1u5K7fs
slESl1MpWtTwEhDcTwK7EpIvYtQ/aUN8Ddb8WHUBiJ1YFkveupD/RwGJBmr2X7KQ
arMCpgKIv7NHfirZ1fpoeDVNAgMBAAGjPzA9MA8GA1UdEwEB/wQFMAMBAf8wCwYD
VR0PBAQDAgEGMB0GA1UdDgQWBbTwj1k4ALP1j5qWDNXr+nuqF+gTEjANBgkqhkiG
9w0BAQUFAAOCAgEAvuRcYk4k9AwI//DTDGjkk0kiP0Qnb7tt3oNmzqjMDfz1mgbl
dxSR651Be5kqhOX//CHBXfDkH1e3damhXwIm/9fH907eT/j3HEbAek9ALCI18Bmx
0GtnLLCo4MBANzX2hFxc469CeP6nyQ1Q6g2EdvZR74NTxnr/DlZJLo961gzmJ1Tj
TQpgcmLNdKqFwPb/ImWvtxBnmq0wROMVvMeJuScg/doAmAyYp4Db29iBT4xdwNBed
Y2gea+zDTYa4EzAvXUYNR0PVG6pZDrlcjQZIrXSHX8f8MVRBE+LHIQ6e4B4N4cB7
Q4WQxYpYxmUKefyxiMPAdkgS94P+5KFdSpcc41teyWRyu5FrgZLAMzTsVlQ2jqI
OylDRl6XK1TOU2+NSueW+r9xDkKLfP0ooNBiYtrEgUy7onOTJsirDNYmiLbAJM+7
vVvrdX3pCI6GMyx5dwlpYn8s3CQh3aP0yK7Qs69cwsGJirQmz1wHiRszYd2qReW
t88NkvuOGKmYSdGe/mBEciG5Ge3C9THxOUiIkCR1VBatzvT4aRRkOfujuLpwQMcn
HL/EVlP6Y2XQ8xwOfvVrhlhNGNTkDY6lnVuR3HYkUD/GKvvZt5y11ubQ2egZixVx
SK236thZiNSQvxaz2emsWWFUyBy6ysHK4bkgTI86k4mloMy/0/Z1pHWwVbVY=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFdDCCA1ygAwIBAgIPAWdfJ9b+euPkrL4JWwWeMA0GCSqGSIb3DQEBCwUAMEQx
CzAJBgNVBAYTAkZJMR0wGAYDVQQKDBFUFZWxpYSBGaW5sYW5kIE95ajEZMBcGA1UE
AwwQVGVSaWEgUm9vdCBDQSB2MjMjAeFw0xODExMjYxMTU1NTRaFw00MzExMjYxMTU1
NTRaMEQxCzAJBgNVBAYTAkZJMR0wGAYDVQQKDBFUFZWxpYSBGaW5sYW5kIE95ajEZ
MBcGA1UEAwwQVGVSaWEgUm9vdCBDQSB2MjMjCAAIwDQYJKoZIhvcNAQEBBQADggIP
ADCCAgogCggIBALLQPwe84nvQa5n44ndp586dpAO8gm2h/oFIH0wnrI4AuhZ76zBq
AMCzdGh+sq/H1WKzeJ9Qyow2RRCrj0jbpDIX2Q3bVTKFgcmfiKDolyzG4OIjNLh9
vVYiQJ3q9HsDrWj8soFPmNB06o3lfc1jw6P23pLCWBnglrvFxKk9pXSW/q/5iaq9
lRdU2HhE8Qx3FZLgmEKnpNaqIJLNwaCzlrI6hEKNfdWV5Nbb6WLEWLN5xYzTNTOD
n3WhUidhOPFZPY5Q4L15POdsv5e2QJltI5c0BE0312/UqeBAMN/mUWZFdUXyApT
7GPzmX3MaRKgwhfwAZ6/hLzRUssbkmbOpFPlob/E2wnW5olWK8jjfN7j/4nlNW4o
6GwLI1GpJQXrSPjdscr6bAhR77cYbETKJuFzxokGgeWKrLDiKca5JLNRBH0pUPC
TEPlcDaMtjNXepUugqD0XBCzYYP2AgWGLnwtbNwDRm41k9V6lS/eINhbfSQBGq6
WT0EBXWdN6IOLj3rwaRSg/7Qa9RmjtzG6RJOHSpXqhC8fF6CfaamyflufUXJ63R
DolUK5X6wK0dmBR4M0KGCqlzft0DbcbMBnEWg4cJ7faGND/igFuvGqHKI3t+ZI
pEYslOqodmJHixBTB0hXbOKSTbauBcvwUpej6w9GU7C7WB1K9vBykLVAgMBAAGj
YzBhMB8GA1UdIwQYMBaAFHKs5DN5qkWH9v2sHZ7Wxy+G2CQ5MB0GA1UdDgQWBbRy
rOQzeapFh/b9rB2e1scvhtgkOTAObgNVHQ8BAf8EBAMCAQYwDwYDVR0TAQH/BAUw
AwEB/zANBgkqhkiG9w0BAQsFAAOCAgEAoDtZpwmUPjaE0n4vOaWWl/oRrfxn83EJ
8rKJhGdEr7nv7ZbsnGTbMjBvZ5qsfl+yqwE2foH65IRE0qw24GtixX1LDoJt0nZi
0f6X+J8wFbJ5tFJ3gh1229MdqfDBmgC9bXXYfef6xzijnHDoRnkDry5023X4blMM
A8iZGok1GTzTyVR8qPAs5m4HeW9q4ebqkYJpCh3DflminmtGFZhb069GHWLIzoBS
SRE/yQQSwwN8PzuKlts8oB4KtItUsiRnDe+Cy748fdHif64W1lZYudogsYMVoe+K
TTJvQS8TUoKU1xrBeKJR3Stwbbca+few4GeXvt8YVMJAygcQMez2P2ccGrGKMOF
6eLtGpOg3kuYooQ+BXcBlj37tCAPnHICehIv1aO6UXivKitEZU61/Qrowc15h2Er
3oBXRb9n8ZuRXqWk7FlIEA04x7D6w0RtBPV4UBySllva9bgulvP5fBqnUsvWHMt
Ty3EHD70sz+rFQ47GUGKpMFXEmZxTPpT41frYpUJnlTd0cl8Vzy9OK2YZLe4A5pT
VmBds9hCG1xLEooc6+t9xnppxyd/pPiL8uSUZodL6ZQHCRJ5irLrdATczvREWeAW
ysUsWNc8e89ihmpQfTU2Zqf7N+cox9jQraVpII/owd8k+BsHMYeB2F326CjYSIKA

rBPuUBQemMc=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFpTCCA42gAwIBAgIUZPYOZXdhags7tOqFhLuxibhXkw8wDQYJKoZIhvcNAQEM
BQAwWjELMAkGA1UEBhMCQ04xJTAjBgNVBAoMHFRydXN0QXNpYSBUZWNobm9sb2dp
ZXMsIEluYy4xJDAiBgNVBAMMG1RydXN0QXNpYSBHbG9iYWwgUm9vdCBDQSBHMzAe
Fw0yMTA1MjAwMjEwMTlaFw00NjA1MTkwMjEwMTlaMFoxCzAJBgNVBAYTAkNOMSUw
IwYDVQQKDBxUcnVzdEFzaWEgVGvjaG5vbG9naWVzLCBjbmuMSQwIgYDVQQDDBtU
cnVzdEFzaWEgR2xvYmFsIFJvb3QgQ0EgRzMwggLiMA0GCSqGSIb3DQEBAQUAA4IC
DwAwggIKAoICAQDAMYJhkuSUGwoqZdC+BqmHO1ES6nBBruL7dOoKjbmzTNyPtxNS
T1QY4SxzlZHFZjtqz6xjbYdT8PfxObegQ2OwxANdV6nnRM7EoYNl9lA+sX4WuDqK
AtCWHwDNBSHvBm3dIzWzQ0WhxeiAysKtQGIXBsaqvPPW5vxQfmZCHzyLpnl5hkA1
nyDvP+uLRx+PjsXUjrYsyUQE49RDdT/VP68czH5GX6zfZBCK70bwkPAPLfSIC7Ep
qq+FqklYqL9joDiR5rPmd2jE+SoZhLsO4fWviejLL1AgdB4SQXMeJNnKziyhWTXA
yB1GJ2Faj/IN03J5Zh6fFZAHLf3ti1ZwA0pJPn9pMRJpXX5cynoTi+jm9WAPzJMs
hH/x/Gr8m0ed262IPfN2dTPXS6Tli/n1Q1hPy8gDVI+lhXgEGvNz8teHHUGf59gX
zhqcD0r83ERoVGjiQTz+LISGNzzNPY+i2+f3VANfWdP3kXjHi3dqFuVJhZBFcnAv
kV34PmVACxmZySYgWmjBNb9Pp1Hx2BErW+Canig7CjoKH8GB5S7wprlppYiU5msT
f9FkPz2ccEblooV7WlQn3MSAPmeamseaMQ4w7OYXQJXZRe0Blqq/DPNL0WP3E1jA
uPP6Z92bfW1K/zJmTSU7/xxnD4UiWQWRkUF3gdCFTIcQcf+eQxuulXUtgQIDAQAB
o2MwYTAPBgNVHRMBAf8EBTADAQH/MB8GA1UdIwQYMBaAFEDk5Plj7zjKsK5Xf/Ih
MBY027ySMB0GA1UdDgQWBRA5OTyI+84yrCuV3/yITAWNnu8kjAOBgNVHQ8BAf8E
BAMCAQYwDQYJKoZIhvcNAQEMBAQADggIBACY7UeFNOPMyGLS0XuFlXsSUT9SnYaP4
wM8zAQLpw6o1D/GUE3d3NZ4tVlFEbuHGLige/9rsR82XRBf34EzC4Xx8MnmpyFq2
XFNFV1pF1AWZLy4jVe5jaN/TG3inEpQGAHUNcoTpLrxaatXeL1nHo+zSh2bbt1S1
JKv0Q3jbSwTEb93mPmY+KfJLaHEih6D4sTNjduMNHXJEIU/HHzp/LgV6FL6qj6j
ITk1dImmasI5+njPtqzn59ZW/yOSLIAlqBUHM/Q4X6RJpStlcHboCoWASzY9M/eV
VHUI2qzEc4Jl6VL1XP04lQJqaTDFHApXB64ipCz5xUG3uOyfT0gA+QEEVcys+Tix
xHWVBqB/0Y0n3bOppHKH/lmLmnp0Ft0WpWIp6zqW3lunaFnT63eROfjXy9mPX1on
AX1daBli2MjN9LdyR75bl87yraKZk62Uy5P2EgmVtqvXO9A/EcswFi55gORngS1d
7XB4tmBZrOFdRWOPyN9yaFvqHbgB8X7754qz41SgOAngPN5C8sLtLpvzHzW2Ntjj
gKGLzZlkD8KqQ7HK9W+eQ42EVJmzbsASZthwEPEGNTNDqJwuuhQxzhB/Hibjj9LV
+Hfsm6vxL2PZQl/gZ4FkkfGXL/xuJvYz+NO1+MRiqzFRJQJ6+N1rZdVtTTDIZbpo
FGWsJwt0ivKH

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICVTCCAAdygAwIBAgIUtyNkuI6XY57GU4HBdk7LKnQV1tcwCgYIKoZIzj0EAwMw
WjELMAkGA1UEBhMCQ04xJTAjBgNVBAoMHFRydXN0QXNpYSBUZWNobm9sb2dpZXMs
IEluYy4xJDAiBgNVBAMMG1RydXN0QXNpYSBHbG9iYWwgUm9vdCBDQSBHNDAAEFw0y
MTA1MjAwMjEwMTlaFw00NjA1MTkwMjEwMTlaMFoxCzAJBgNVBAYTAkNOMSUwIwYD
VQQKDBxUcnVzdEFzaWEgVGvjaG5vbG9naWVzLCBjbmuMSQwIgYDVQQDDBtUcnVz
dEFzaWEgR2xvYmFsIFJvb3QgQ0EgRzMwggLiMA0GCSqGSIb3DQEBAQUAA4ICDwAw
ggIKAsICAQDASzthwEPEGNTNDqJwuuhQxzhB/Hibjj9LV+Hfsm6vxL2PZQl/gZ4Fkkf
GXL/xuJvYz+NO1+MRiqzFRJQJ6+N1rZdVtTTDIZbpoFGWsJwt0ivKH

YzBhMA8GA1UdEwEB/wQFMAMBAf8wHwYDVR0jBBgwFoAUUpbtKl86zK3+kMd6Xg1mD
pm9xy94wHQYDVR0OBBYEFKW7SpfOsytpDHl4NZg6ZvccveMA4GA1UdDwEB/wQE
AwIBBjAKBgghkjOPQDAwNnADBkAjBe8usGzEkxn0AAAbd+NvBNEU/zy4k6LHiR
UKNbwMp1JvK/kF0LgoxgKJ/GcJpo5PECMFxYDI2z21jD1xCMuo6u47xkdUfFVZDj
/bpV6wfEU6s3qe4hsiFbYI89MvHVI5TWWA==
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICMTCCAbegAwIBAgIUUnThTXxlE8msg1UloD5Sfi9QaMcwCgYIKoZlZj0EAwMw
WDELMakGA1UEBhMCQ04xJTAjBgNVBAoTHFRydXN0QXNpYSBUZWNobm9sb2dpZXMs
IEluYy4xIjAgBgNVBAMTGVRydXN0QXNpYSBUTFMgRUNDIFJvb3QgQ0EwHhcNMjQw
NTE1MDU0MTU2WhcNNDQwNTE1MDU0MTU1WjBYMQswCQYDVQQGEwJDTjEIMCMGA1UE
ChMcVHJ1c3RBc2lhIFRIY2hub2xvZ2llcywgSW5jLjEiMCAGA1UEAxMZVHJ1c3RB
c2lhIFRMUyBFQ0MgUm9vdCBDQTB2MBAGByqGSM49AgEGBSuBBAAiA2IABLh/pVs/
AT598IhtrimY4ZtcU5nb9wj/1WrgjstEpvDBjL1P1M7UiFPoXlfXTr4sP/MSpwDp
guMqWzJ8S5sUKZ74LYO1644xST0mYekdcouJtgq7nDM1D9rs3qlKH8kzsNCMEAw
DwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQULIVTu7FDzTLqnqOH/qKYqKaT6RAw
DgYDVR0PAQH/BAQDAgEGMAoGCCqGSM49BAMDA2gAMGUCMFRH18MtYYZI9HlaVQ01
L18N9mdsd0AaRuf4aFtOJx24mH1/k78ITcTaRTChD15KeAIXAKORh/IRM4PDwYqR
OkwrULG9IpRdNYlZg8WbGf60oenUoWa2AaU2+dhoYSi3dOGiMQ==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFgDCCA2igAwIBAgIUHBJYz+VTPyI1RINUJDxsR9FcSpwwDQYJKoZIhvcNAQEM
BQAwWDELMakGA1UEBhMCQ04xJTAjBgNVBAoTHFRydXN0QXNpYSBUZWNobm9sb2dp
ZXMsIEluYy4xIjAgBgNVBAMTGVRydXN0QXNpYSBUTFMgUINBIFJvb3QgQ0EwHhcN
MjQwNTE1MDU0MTU3WhcNNDQwNTE1MDU0MTU2WjBYMQswCQYDVQQGEwJDTjEIMCMG
A1UEChMcVHJ1c3RBc2lhIFRIY2hub2xvZ2llcywgSW5jLjEiMCAGA1UEAxMZVHJ1
c3RBc2lhIFRMUyBSU0EgUm9vdCBDQTCcAiIwDQYJKoZIhvcNAQEBBQADggIPADCC
AgoCggIBAMMWuBtqpERz5dZ09LnPWwvB0ZqB9WOWj0PBuwhaGnrhB3YmH49pVr7+
NmDQDIPNIOrnxS1cLwUWAp4KqC/IYCZUlvYQB2srp10Zy9U+5RjmOMmSoPGLbYJ
Q1DNDX3eRA5gEk9bNb2/mThfWza4mhzH/kxpRkQcwUqwzIZheo0qt1CHjCNP561
HmHVb70AcnKEtj+qpkLz8oYVlQwQX1Fkzv93uMltrOXVmPGZLmzjyUT5tUMnCE32
ft5EebuyjBza00tsLtbDeLdM1aTk2tyKjg7/D8OmYCYozza/+lcK7Fs/6TAWe8Tb
xNRkoDD75f0dcZLDKY9BWN4ArTr9PXwaqLEX8E40eFgl1oUh63kd0Nyrz2I8sMeX
i9bQn9P+PN7F4/w6g3CEIR0JwqH8uyghZVNgepBtljhb//HXeltt08lwSUq6HTrQ
UNoyIBnkiz/r1RYmNzz7dZ6wB3C4FGB33PYPXFIKvF1tjVEK2sUYyJtt3LCDs3+j
TnhMmCW8n4uIF6CFabW2I+s5c0yhsj55NqJ4js+k8UTav/H9xj8Z7XvGCxUq0DT
bE3txci3OE9kxJRMT6DNrqXGJyV1J23G2pyOsAWZ1SgRxSHUuPzHlqtKZFhaxP8
S8ySpg+kUb8OWJDZgoM5pl+z+m6Ss80zDoWo8SnTq1mt1tve1CuBAGMBAAGjQjBA
MA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFgHkXlcBvRG/XtZylomkadFK/hT
MA4GA1UdDwEB/wQEAwIBBjANBgkqhkiG9w0BAQwFAAOCAgEAIztqBSBdGBanEqT3
Rz/NyjuujsCCztIJXgXbODgcMTWltnZ9r96nBO7U5WS/8+S4PPFJzVXqDuiGev4
iqME3mmL5Dw8veWv0BIb5Ylrc5tvJQLkIKvQMktuppgJFqBTQUYo+IzeXoLH5Pt
7DIK9RME7I10nYEKqG/odv6LTytpEoYKNDbdgptvT+Bz3UI/KD7JO6NXBNiT2Twp
2xIQaOHEibgGIOcberyxk2GaGUARtWqFVwHxtlotJnMnlvm5P1vQiJ3koP26TpUJ
g3933FEFlJ0gcXax7PqJtZwuhfG5WyRasQmr2soaB82G39tp27RIGAAtvKLEiUUj

pQ7hRGU+isFqMB3iYPg6qocJQrmBktwliJiJ8Xw18WLK7nn4GS/+X/jbh87qqA8M
pugLoDzga5SYnH+tBuYc6kIQX+ImFTw3OffXvO645e8D7r0i+yiGNFjEWn9hongP
XvPKnbwbPKfILfanlhHKA9jnZwqKDss1jjQ52MjqjZ9k4DewbNffj8GQYSbbJIwe
SsCI3zWQzj8C9GRh3sflB5XeMhg6j6JCQCTl1jNdfK7vsU1P1FeQNWrcrgSXSyk0
ly4wBOeY99sLAZDBHwo/+ML+TvrbmNzFrwFuHnYWa8G5z9nODmxKfKuU4CkUpijy
323imttUQ/hHWKNddBWcwauwxzQ=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIF2jCCA8KgAwIBAgIMBfcOhtpJ80Y1LrqyMA0GCSqGSIb3DQEBCwUAMIGIMQsw
CQYDVQQGEwJVUzERMA8GA1UECAwISWxsaw5vaXMxEDAOBgNVBACMB0NoaWNhZ28x
ITAFBgNVBAoMGFRydXN0d2F2ZSBib2xkaW5ncywgSW5jLjExMC8GA1UEAwwOVHJ1
c3R3YXZlIEdsb2JhbCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAeFw0xNzA4MjMx
OTM0MTJhFw00MjA4MjMxOTM0MTJhMIGIMQswCQYDVQQGEwJVUzERMA8GA1UECAwI
SWxsaw5vaXMxEDAOBgNVBACMB0NoaWNhZ28xITAFBgNVBAoMGFRydXN0d2F2ZSBi
b2xkaW5ncywgSW5jLjExMC8GA1UEAwwOVHJ1c3R3YXZlIEdsb2JhbCBDZXJ0aWZp
Y2F0aW9uIEF1dGhvcml0eTCCAIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIB
ALldUShLPDeS0YLOvR29zd24q88KPUFd5dyqCblXAJ7mY2Hf8g+CY66j96xz0Xzn
swuvCAAJWX/NKSqIk4cXGIDtLK0thAfLdZfValTXdHG6wZWwiYj+rDKd/VzDBcdu
7oaJuogDnXIhhpCujwOl3J+IKMujkkkP7NAP4m1ET4BqstTnoApTAbqOl5F2brz8
1Ws25kCI1nsVXwXoLG0R8+eyvpJETNKXpP7ScoFDB5zpET71ixpZfr9oWN0EACyW
80OzfpgZdNmcc9kYvkHHNHnZ9GLCQ7mzJ7Aiy/k9UscwR7PJPrhq4ufogXBeQotP
JqX+OsIgrv4Fo7NDK0G2x2EOFYeUY+VM6AqFcJNykbmROPDMjWLBz7BegIIT1l
RtzuzWniTY+HKE40Cz7PFNm73bZQmq131BnW2hqIyE4bJ3XYsgjxroMwuREozYfw
hI0Vcnyh78zyiGG69Gm7DIwLdVcEuE4qFC49DxweMqZiNu5m4iK4BUBjECLzMx10
coos9TkpoNPNg4CELcU9402x/RpvumUHO1jsQkUm+9jaJXLE9gCxInm943xZYkq
BW89zubWR2OZxiRvchLrH+QtAuRcOi35hYQcRfO3gZPSEF9NUqjifLJS3tBEW1n
twiYTOURGa5CgNz7kAXU+FDKvuStx8KU1xad5hePrzb7AgMBAAAGjQjBAMA8GA1Ud
EwEB/wQFMAMBAf8wHQYDVR0OBBYEFjngGwCNYtt2s9o9uFvo/ULSMQ6HMA4GA1Ud
DwEB/wQEAwIBBjANBgkqhkiG9w0BAQsFAAOCAgEAMHNw4rDT7TnsTGDZqRKGFx6W
00hUKDtkLSGm+J1WE2pIPU/HPinbbViDVD2HfSMF1OQc3Og4ZYbFdada2zUFvXfe
uyk3QAUHW5RSn8pk3fEbK9xGChACMf1KaA0HZJDmHvUqoi7PF35owgLEQzxPy0Q
IG/+4jSHg9bP5Rs1bdID4bANqKCqRieCNqcVtgimQlRXtpla4gt5kNdXELE1GYhB
aCXUNxeEFfsBctyV3lImIJgm4nb1J2/6ADtKYdkNy1GTKv0WBpanI5ojSP5RvbbE
sLFUzt5sQa0WZ37b/TjNuThOssFgy50X31ieemKyJo90LzvkWx3SD92YHJtZuSPT
MaCm/zjdzyBP6VhWomfD0faZmZ26NraAL4hHT4a/RDqA5Dccprql5gR0IRiR2Qe
qu5AvzSxnI904fKSTx+O856X3vOmeWqJcU9LJxdI/uz0UA9PSX3MReO9ekDFQdxh
VicGaeVYQYHTtgGJoC86cnn+OjC/QueZHYj6RS8fZMXZC+fc8Y+wmjHMMfRod6qh8
h6jCJ3zhM0EPz8/8AKAigJ5Kp28AsEFFtyLKAEjFQqKu3R3y4G5OBVixwJAWKqQ9
EEC+j2Jjg6mcgn0tAumDMHzLJ8n9HmYAsC7TIS+OMxZsmO0QqAfWzJPP29FpHOTK
yeC2nOnOcXHebD8WpHk=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICYDCCAgegAwIBAgIMDWpfCD8oXD5Rld9dMAoGCCqGSM49BAMCMIGRMQswCQYD
VQQGEwJVUzERMA8GA1UECBMISWxsaw5vaXMxEDAOBgNVBACMB0NoaWNhZ28xITAF

BgNVBAoTGFRydXN0d2F2ZSBib2xkaW5ncywgSW5jLjE6MDgGA1UEAxMxVHJ1c3R3
YXZlIEdsb2JhbCBFQ0MgUDI1NiBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAeFw0x
NzA4MjMxOTM1MTBaFw00MjA4MjMxOTM1MTBaMIGRMQswCQYDVQQGEwJVUzERMA8G
A1UECBMISWxsaW5vaXMxEDAOBgNVBAcTB0NoaWNhZ28xITAfBgNVBAoTGFRydXN0
d2F2ZSBib2xkaW5ncywgSW5jLjE6MDgGA1UEAxMxVHJ1c3R3YXZlIEdsb2JhbCBF
Q0MgUDI1NiBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTBZMBMBGByqGSM49AgEGCCqG
SM49AwEHA0IABH77bOYj43MyCMpg5lOcuSNGLB4kFKA3TjASh3RqMyTpJcGOMoN
FWLGjgEqZZ2q3zSRL0HB5DOSMcT9CTqmP62jQzBBMA8GA1UdEwEB/wQFMAMBAf8w
DwYDVR0PAQH/BAUDAwcGADAdBgNVHQ4EFgQUo0EGrJBt0UrrdaVKEJmzsaGLSvcw
CgYIKoZlZj0EAwIDRwAwRAIgB+ZU2g6gWrKuEZ+Hxbbb/ad4lvvigtwjzRM4q3wgh
DDcCIC0mA6AFvWvR9lz4ZcyGbbOcNEhjhAnFjXca4syc4XR7

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICnTCCAiSgAwIBAgIMCL2Fl2yZJ6SAaEc7MAoGCCqGSM49BAMDMIGRMQswCQYD
VQQGEwJVUzERMA8GA1UECBMISWxsaW5vaXMxEDAOBgNVBAcTB0NoaWNhZ28xITAf
BgNVBAoTGFRydXN0d2F2ZSBib2xkaW5ncywgSW5jLjE6MDgGA1UEAxMxVHJ1c3R3
YXZlIEdsb2JhbCBFQ0MgUDM4NCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAeFw0x
NzA4MjMxOTM2NDNaFw00MjA4MjMxOTM2NDNaMIGRMQswCQYDVQQGEwJVUzERMA8G
A1UECBMISWxsaW5vaXMxEDAOBgNVBAcTB0NoaWNhZ28xITAfBgNVBAoTGFRydXN0
d2F2ZSBib2xkaW5ncywgSW5jLjE6MDgGA1UEAxMxVHJ1c3R3YXZlIEdsb2JhbCBF
Q0MgUDM4NCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTB2MBAGByqGSM49AgEGBSuB
BAAiA2IABGvaDXU1CDFHBa5FmVXxERMuSvqQMSOjfoPTfygIOiYaOs+Xgh+AtycJ
j9GOMMQKmw6sWASr9zZ9lCOkmwqKi6vr/TklZvFe/oyujUF5nQlgiip04pt89ZF
1PKYhDhloKNDMEewDwYDVR0TAQH/BAUwAwEB/zAPBgNVHQ8BAf8EBQMDBwYAMB0G
A1UdDgQWBBBRVqYSJ0sEyyRjLbKYHTsjnnb6CkDAKBggqhkJOPQQDAwNnADBKAjA3
AZKXRRJ+oPM+rRk6ct30UJMDER5E0k9BpIycnR+j9sKS50gU/k6bpZFXrsY3crsC
MGclCrEMXu6pY5Jv5ZAL/mYiykf9ijH3g/56vxC+GCsej/YpHpRZ744hN8tRmKVu
Sw==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFszCCA5ugAwIBAgIUeWLV4kBMkkaGFmddtLu7sms+/BMwDQYJKoZIhvcNAQEL
BQAwYTELMaKGA1UEBhMCVE4xNzA1BgNVBAoMLkFnZW5jZSBOYXRpb25hbGUgZGUg
Q2VydGlmaWNhdGlvb2JhbCBFQ0MgUDM4NCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTBZMBMBGByqGSM49AgEGCCqGSM49AwEHA0IABH77bOYj43MyCMpg5lOcuSNGLB4kFKA3TjASh3RqMyTpJcGOMoNFWLGjgEqZZ2q3zSRL0HB5DOSMcT9CTqmP62jQzBBMA8GA1UdEwEB/wQFMAMBAf8wDwYDVR0PAQH/BAUDAwcGADAdBgNVHQ4EFgQUo0EGrJBt0UrrdaVKEJmzsaGLSvcwCgYIKoZlZj0EAwIDRwAwRAIgB+ZU2g6gWrKuEZ+Hxbbb/ad4lvvigtwjzRM4q3wghDDcCIC0mA6AFvWvR9lz4ZcyGbbOcNEhjhAnFjXca4syc4XR7

MIeSPvLfe/ZdeikZjuXIvTZxi11Mwh0/rViizz1wTaZQmCXcl/m4WEEIcb9PuISg
jwBUFFyRbVinljvrS5YnzWuioYasDXxU5mZMZl+QviGaAkYt5IPCgLnPSz7ofzwB
7I9ezX/SKEIBlYrilz0QIX32nRzFNKHsLA4KUiwSVXAkPcvCFDvDXSdOvsC9qnyW
5/yeYa1E0wCXAgMBAAGjYzBhMB0GA1UdDgQWBBQGmpsfU33x9aTI04Y+oXNZtPdE
ITAPBgNVHRMBAf8EBTADAQH/MB8GA1UdIwQYMBaAFAaamx9TffH1pMjThj6hc1m0
90QhMA4GA1UdDwEB/wQEAWIBBjANBgkqhkiG9w0BAQsFAAOCAgEAqgVutt0Vyb+z
xiD2BkewhpMl0425yAA/l/VsJ4hxyXT968pk21vvHl26v9Hr7lxpuhbI87mP0zYu
QEkHdVneixCwSQXi/5E/S7fdAo74gShczNxt18UnH1YeA32gAm56Q6XKRm4t+v4
FstVEuTGfbvE7Pi1HE4+Z7/FXxttbUcoqgRYYdZ2vyJ/0Adqp2RT8JeNnYA/u8EH
22Wv5psymsNUk8QcCMNE+3tjEUPRahphanltkE8pjkcFwRJpadbGNjHh/PqAulxP
xOu3Mqz4dWEX1xAZufHSCe96Qp1bWgvUxpVOKs7/B9dPfhgGiPEZtdmYu65xxBzn
dFIY7wyJz4sfDZMAbBSSSFCp61cpABbjNhzi+L/wM9VBD8TMPN3pM0MBkRARHtG5
Xc0yGYuPjCB31yLEQtyEFpslbei0VXF/sHyZ03FJuc9SpAQ/3D2gu68zngowYI7b
nV2UqL1g52KAdoGDDIzMMMEZJ4gzSqk/rYXHv5yJiqfdcZGyFfoxnNidF9Ql7v/YQ
CvGwjVRDjAS6oz/v4jXH+XTgbzRB0L9zZVcg+ZtnemZoJE6AZb0QmQZZ8mWvuMZH
u/2QeItBcy6vVR/cO5JyboTT0GFMDcx2V+IthSIVNg3rAZ3r2OvEhJn7wAzMMujj
d9qDRlueVSjAi1jTkD5OGwDxFa2DK5o=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFWjCCA0KGAwIBAgIQT9Irr/VkyDOeTzRYZiNwYDANBgkqhkiG9w0BAQsFAADBH
MQswCQYDVQQGEwJDTjJERMA8GA1UECgwIVW5pVHJ1c3QxJTAjBgNVBAMMHFVDQSBF
eHRLbmRlZCBWYWxpZGF0aW9uIFJvb3QwHhcNMTUwMzEzMDAwMDAwWhcNMzgXMjMx
MDAwMDAwWjBHMqswCQYDVQQGEwJDTjJERMA8GA1UECgwIVW5pVHJ1c3QxJTAjBgNV
BAMMHFVDQSBFeHRLbmRlZCBWYWxpZGF0aW9uIFJvb3QwggLiMA0GCSqGSIb3DQEB
AQUAA4ICDwAwggIKAoICAQCpCQcoEwKwmeBkqh5DFnpzsZGgdT6o+uM4AHrsiWog
D4vFsJszA1qGxliG1cGFu0/GnEBNyr7uaZa4rYewmnySBesFK5pI0Lh2PpbIILvS
sPGP2KxFRv+qZ2C0d35qHzwaUnoEPQc8hQ2E0B92CvdqFN9y4zR8V05WAT558aop
O2z6+I9tTcg1367r3CTueUWnhbYFiN6IXSV8l2RnCdM/WhUFhvMJHuxYMjMR83dk
sHYf5BA1FxyvDrFspCqjc/wJHx4yGVMR59mzLC52LqGj3n5qiAno8geK+LLNEOfi
c0CTuwjRP+H8C5SzJe98ptfRr5//lpr1kXuYC3fUfugH0mK1lTnj8/FtDw5lhIpj
VMWAuCeS31HJqcBCF3RiJ7XwzJE+oJKcmhUfzhTA8ykADNkUVkLo4KRel7sFsLz
KuZi2irbWWIQJUoqgQtHB0MGcIfS+pMRKXpITeuUx3BNr2fVUbGAIAEBtHoIppB/
TuDvB0GHR2qlXov7z1CymLsvw4m6WC31MJixNnI5fkkE/SmnTHnkBVfblLkWU41G
sx2VYvdWf6/wFlthWG82UBEL2KwrlRYaDh8lzTY0ZRBiZtWAXxQgXy0MoHgKaNYs
1+lvK9JKBZP8nm9rZ/+I8U6laUpSNwXqxhaN0sSZ0YIrO7o1dfdRUVjzyAfd5LQD
fwIDAQABo0IwQDAdbgNVHQ4EFgQU2XQ65DA9DfcS3H5aBZ8eNjr34RQwDwYDVR0T
AQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAYYwDQYJKoZIhvcNAQELBQADggIBADaN
l8xCFWQpN5smLNB7rhVpLGsaGvdftvkHTFng88nLua7Mui563MD1sC3AO6+fcAUR
ap8lTwEpcOPIDOHqWnzcSbvBHiqB9RZLcpHlojG5qtr8nR/zXUACE/xOHAbKsxSQ
VBcZEhrxH9cMaVr2cXj0lH2RC47skFSOvG+hTKv8dGT9cZr4QQehzZHkPJrgmzI5
c6sq1WnleJEmMX3ixzDx/BR4dxIOE/TdFpS/S2d7cFOFyrC78zhNLJA5wA3CXWvp
4uXVil3WLL+rG761KIcSF3Ru/H38j9CHJrAb+7lsq+KePRXBOy5nAliRn+/4Qh8s
t2j1da3Ptfb/EX3C8CSlrdP6oDyp+l3cpaDvRKS+1ujl5BOWF3sGPjLtx7dCvHaj
2GU4Kzg1USEODm8uNBNA4StnDG1KQTAYI1oyVZnJF+A83vbsea0rWBmirSwiGpWO
vpaQXUJXxPkUAzUrHC1RVwinOt4/5Mi0A3PCwSaAuwtCH60NryZy2sy+s6ODWA2C
xR9GUeOcGMyNm43sSet1UNWMKFnKdDTajAshqx7qG+XH/RU+wBeq+yNuJkBL+vmx

cmtpzyKEC2IPrNkZAJSidjzULZrtBJ4tBmIQN1IchXIbJ+XMxjHsN+xjWZsLHXbM

fjKaiJUIINIK73nZfdklJrX+9ZSCyycErdhh2n1ax

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFRjCCAY6gAwIBAgIQXd+x2lqj7V2+WmUgZQOQ7zANBgkqhkiG9w0BAQsFADA9
MQswCQYDVQQGEwJDTjERMA8GA1UECgwIVW5pVHJ1c3QxGzAZBgNVBAMMElVDQSBH
bG9iYWwgRzIgUm9vdDAeFw0xNjAzMTEwMDAwMDBaFw00MDEyMzEwMDAwMDBaMD0x
CzAJBgNVBAYTAkNOMREwDwYDVQQKDAhVbmlUcnVzdDEbMBkGA1UEAwwSVUNBIEds
b2JhbCBHMiBSb290MIIICjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAxeYr
b3zvJgUno4Ek2m/LAfmZmqkywiKHUYUGRO8vDaBsGxUypK8FnFyIdK+35KYmToni9
kmugow2ifsqTs6bRjDXVdfkX9s9FxeV67HeToI8jrg4aA3++1NDtLnurRiNb/yzm
VHqUwCoV8MmNsHo7JOHXaOIxPAYzRrZUEaalLyJUKlgNAQLx+hVRZ2zA+te2G3/R
VogvGjqNO7uCEeBHANBSh6v7hn4PJGtAnTRnvI3HLYZveT6OqTwXS3+wmeOwcWDc
C/Vkw85DvG1xudLeJ1uK6NjGruFZfc8oLTW4lVYa8bJYS7cSN8h8s+1LgOGN+jIj
tm+3SJUIsUROhYw6AlQgL9+/V087OpAh18EmNVQg7Mc/R+zvWr9LesGtOxdQXGLY
D0tK3Cv6brxzs3sx1DoQZbXqX5t2Okdj4q1uViSukqSKwxW/YDrCPBeKW4bHAyv
j5OJrdu9o54hyokZ7N+1wxrrFv54NkzWbtA+FxyQF2smuv6L78RHBgOLXMDj6DI
NaBa4kx1HXHhOThTeEDMg5PXCp6dW4+K5OXgSORIskfNTip1KnvyIvbJvgmRlId6
ilis7nCs+dwp4wwcOxJORNanTrAmyPPZGpeRaOrvjUYG0lZFWJo8DA+DuAUlwznP
O6Q0ibd5Ei9Hxeopl2n8pndntd978XplFeRhVmUCAwEAANCMEEAwDgYDVR0PAQH/
BAQDAgEGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFiHEjMz15DD/pQwIX4wV
ZyF0Ad/fMA0GCSqGSIb3DQEBCwUAA4ICAQATZSL1jiutROTL/7lo5sOASD0Ee/oj
L3rtNtqyzm325p7lX1iPyzcyochltq44PTUbPrw7tgTQvPlJ9Zv3hcU2tsu8+Mg5
1eRfB70VVJd0ysrtT7q6ZHafgbiERUIMjW+i67HM0cOU2kTC5uLqGOiiHycFutfl
1qnN3e92mI0ADs0b+gO3joBYDic/UvuUospeZcnWhNq5NXHzJsBPd+aBJ9J3O5oU
b3n09tDh05S60FdRvScFDcH9yBIw7m+NESsIndTUv4BFFJqIRNow6rSn4+7vW4LV
PtateJLbXDzz2K36uGt/xDYotgIVilQsnLAXc47QN6MUPJiVAAwpBVueSUmX8fj
y88nZY41F7dXyDDZQVu5FLbowg+UMaeUmMxq67XhJ/UQqAHojhJi6IjMtX9GIl8Cb
EGY4GjZGXyJoPd/JxhMnq1MGrKI8hgZlb7F+sSIEmqO6SWkoaY/X5V+tBIZkxbxqg
DMUIYs6Ao9Dz7GjevjpHF1t/gMRMTLGmhlrDO7gJzRSBuhjjVFc2/tsvfEehOjPI
+Vg7RE+xygKJBjYoaMVLuCaJu9YzL1DV/pqJuhgyklTGW+Cd+V7IDSKb9triyCGy
YiGqhkCyLmTTX8jjfhFnRR8F/uOi77Oos/N9j/gMHYIfLXC0uAE0djAA5SN4p1bX
UB+K+wb1whnw0A==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICjzCCAhWgAwIBAgIQXIuZxVqUxdJxVt7NiYDMJjAKBgqhkiG9w0BAQsFDEL
MAkGA1UEBhMCVVMxEzARBgNVBAGTCk5ldyBKZXJzZXkxZDASBgNVBACTC0plcnNl
eSBDaXR5MR4wHAYDVQQKEExVUaGUgVVFNFUIRSVVNUIE5ldHdvcmxsLjAsBgNVBAMT
JVVTTRVJUcnVzdCBFQ0MgQ2VydGlmawNhdGlvbiBBdXR0b3JpdHkwHhcNMTAwMjAx
MDAwMDAwWhcNMzgwMTE4MjM1OTU5WjCBiDELMAkGA1UEBhMCVVMxEzARBgNVBAGT
Ck5ldyBKZXJzZXkxZDASBgNVBACTC0plcnNleSBDaXR5MR4wHAYDVQQKEExVUaGUg
VVFNFUIRSVVNUIE5ldHdvcmxsLjAsBgNVBAMTJVVTTRVJUcnVzdCBFQ0MgQ2VydGlm
awNhdGlvbiBBdXR0b3JpdHkwZjAQBgcqhkiG9w0BAQIIBBgUrgQQAIGNiAAQarFRaqflo
I+d61SRvU8Za2EurxtW20eZzca7dnNYMYf3boIkDuAUU7FfO7l0/4iGzzvfUinng

o4N+LZfQYcTxmdwlkWOrfzCjtHDix6EznPO/LlxTsV+zfTJ/ijTjeXmjQjBAMB0G
A1UdDgQWBBQ64QmG1M8ZwpZ2dEl23OA1xmNjnmjAOBgNVHQ8BAf8EBAMCAQYwDwYD
VR0TAQH/BAUwAwEB/zAKBgqhkhjOPQQDAwNoADBlAjA2Z6EWCNzklwBBHU6+4WMB
zzuqQhFkoJ2UOQIReVx7Hfpkue4WQrO/isIJxOzksU0CMQDpKmFHjFJKS04YcPbW
RNZu9YO6bVi9JNlWSOrvxKJGgYhqOkbRqZtNyWHa0V1Xahg=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIF3jCCA8agAwIBAgIQAf1tMPyjylGoG7xkDjUDLTANBgkqhkiG9w0BAQwFADCB
iDELMAkGA1UEBhMCVVMxEzARBgNVBAGTCk5ldyBKZXJzZXkxFDASBgNVBACTC0pl
cnNleSBDaXR5MR4wHAYDVQQKEsVUaGUgVGVNVFUIRSVVNUIE5ldHdvcmsxLjAsBgNV
BAMTJVVTRVJUcnVzdCBSU0EgQ2VydGlmaWNhdGlubiBBdXR0b3JpdHkwHhcNMTAw
MjAxMDAwMDAwWhcNMzgwMTE4MjM1OTU5WjCBiDELMAkGA1UEBhMCVVMxEzARBgNV
BAGTCk5ldyBKZXJzZXkxFDASBgNVBACTC0plcnNleSBDaXR5MR4wHAYDVQQKEsVU
aGUgVGVNVFUIRSVVNUIE5ldHdvcmsxLjAsBgNVBAMTJVVTRVJUcnVzdCBSU0EgQ2Vy
dGlmaWNhdGlubiBBdXR0b3JpdHkwggliMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIK
AoICAQCAEmUXNg7D2wiz0KxXDxbtzSfTTK1Qg2HqiqBNCS1kCdzOiZ/MPans9s/B
3PHTsdZ7NygRK0faOca8Ohm0X6a9fZ2jY0K2dvKpOyuR+OJv0OwWlIAJPuLodMkY
tJHUYmTbf6MG8YgYapAiPLz+E/CHFHV25B+O1ORRxfFnRghRy4YUVD+8M/5+bJz/
Fp0YvVGONaanZshyZ9shZrHUm3gDwFA66Mzw3LyeTP6vBZY1H1dat//O+T23LLb2
VN3I5xI6Ta5MirdcmrS3ID3KfyI0rn47aGYBROcBTkZTmzNg95S+UzeQc0PzMsNT
79uq/nROacdrrjGCT3sTHDN/hMq7MkztReJVni+49Vv4M0GkPGw/zJSZrM233bkf6
c0Plfg6lZrEpFDKEY1WJxA3Bk1QwGROs0303p+tdOmw1XNtB1xLaqUkL39iAigmT
Yo61Zs8liM2EuLE/pDkP2QKe6xJmIXzzawWpXhaDzLhn4ugTncxbgtNMs+1b/97l
c6wjOy0AvzVVdAlJ2EIYGn+SNuZRkg7zJn0cTRe8yexDJtC/QV9AqURE9JnnV4ee
UB9XVKg+/XRjL7FQZQnmWEIuQxpMtPAIR1n6BB6T1CZGSICBst6+eLf8ZxXhyVeE
Hg9j1uliutZfVS7qXMYoCAQlObgOK6nyTJccBz8NUvXt7y+CDwIDAQABo0IwQDAd
BgNVHQ4EFgQUU3m/WqorSs9UgOHYm8Cd8rIDZsswDgYDVROPAQH/BAQDAgEGMA8G
A1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEMBQADggIBAFzUfA3P9wF9QZlIDHPF
Up/L+M+ZBn8b2kMVn54CVVeWFPFSPCeHlCjtHzoBN6J2/FNQwISbxmtOuowhT6KO
VWKR82kV2LyI48SqC/3vqOILVSoGIG1VeCkZ7l8wXEskEVX/JJpuXior7gtNn3/3
ATiUFJVDWBwn7YKnuHKsSjKCaXqeYalltiz8I+8jRRa8YFWSQEG9zKC7F4iRO/Fjs
8PRF/iKz6y+O0tlfYQXBl2+odnKPi4w2r78NBc5xjeambx9spnFixdjQg3IM8WcR
iQycE0xyNN+81XHfqHd4blsjDwSXWxavVcStkNr/+XeTWYRUc+ZruwXtuhxkYze
Sf7dNXGfSeUHM9h4ya7b6NnJSF5t0dCy5oGzuCr+yDZ4XUmFF0sbmZgIn/f3gZ
XHIKYC6SQK5MNyosycdiyA5d9zZbyuAlJQG03RoHnHcAP9Dc1ew91Pq7P8yF1m9/
qS3fuQL39ZeatTXaw2ewh0qpKJ4jjv9cJ2vhsE/zB+4ALtRZh8tSQZXq9EfX7mRB
VXYnWQKV3WKdwrnuWih0hKWbt5DHDAff9Yk2dDLWKMgwsAvgnEzDHNb842m1R0aB
L6KCq9NjRHDEj8tM7qtj3u1cIiuPhnPQCjY/MiQu12ZivVS5ljFH4gxQ+6IHdfG
jjxDah2nGN59PRbxYvnKkKj9

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDODCCAiCgAwIBAgIGIAyFFnACMA0GCSqGSIb3DQEBBQUAMDSxCzAJBgNVBAYT
AlJPMREwDwYDVQQKEwhjZXJ0U0lHTjEZMBcGA1UECzMqY2VydFNJR04gUk9PVCBD
QTAEFw0wNjA3MDQxNzIwMDRaFw0zMTA3MDQxNzIwMDRaMDsxCzAJBgNVBAYTAlJP

MREwDwYDVQQKEwhjZXJ0U0lHTjEZMBcGA1UECxmQY2VydFNJR04gUk9PVCBDQTCC
ASlWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALczuX7IJUqOtdu0KBuqV5D0
0SLTZLrTk+jUrIzhQGpgV2hUhE28alQCBf/fm5oqrl0Hj0rDKH/v+yv6efHHrfAQ
UySQi2bJqlirr1qjAOm+ukbuW3N7LBeCgV5iLKECZbO9xSsAfsT8AzNXDe3i+s5d
RdY4zTW2ssHQnIFKquSyAVwdj1+ZxLGt24gh65AIgoDzMKND5pCCrIUoSe1b16kQ
OA7+j0xbm0bqQfWwCHTD0IgztnzXdN/chNFDDnU5oSVAKOp4yw4sLjmdjltuFhwv
JoIQ4uNlIAoEwF73XVv4EOLQunpL+943AAAaWyjj0pxzPjKHmKHJUS/X3qws08C
AwEAAANCMEEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAcYwHQYDVR0O
BBYEFOCMm9slSbPxfIbWskKHC9BroNnkMA0GCSqGSIb3DQEBAQUAA4IBAQA+0hyJ
LjX8+HXd5n9liPRyTMks1zJO890ZeUe9jjtbkw9QSSQTaxQGcu8J06Gh40CEyecY
MnQ8SG4Pn0vU9x7Tk4ZkVJdclDVVc/6IJMCopvDI5NOFIV2oHB5bc0hH88vLbwZ
44gx+FkagQnIl6Z0x2DEW8xXjrJ1/RsCCdtZb3KTafcxQdaIOL+Hsr0Wefmq5L6I
Jd1hJyMctTEHBDa0GpC9oHRxUiltvBTjD4au8as+x6AJzKNI0eDbZOeStc+vckNw
i/nDhDwTqn6Sm1dTk/pwwpEOMfmbZ13pljheX7NzTogVZ96edhBiIL5VaZVDADIN
9u6wWk5JRFrYX0KD

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFRzCCAY+gAwIBAgIJEQA0tk7GNi02MA0GCSqGSIb3DQEBCwUAMEEExCzAJBgNV
BAYTAIJPmRQwEgYDVQQKEwtDRVJU0lHTiBTQTEcMBoGA1UECxmTY2VydFNJR04g
Uk9PVCBDQSBHMjAeFw0xNzAyMDYwOTI3MzVaFw00MjAyMDYwOTI3MzVaMEExCzAJ
BgNVBAYTAIJPmRQwEgYDVQQKEwtDRVJU0lHTiBTQTEcMBoGA1UECxmTY2VydFNJ
R04gUk9PVCBDQSBHMjCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAMDF
dRmRfUR0dIf+DjuW3NgBFszuY5HnC2/OOwppGnzC46+CjobXXo9X69MhWf05N0Iw
vldQtg+piNguLWkh59E3GE59kdUWX2tbAMI5Qw02hVK5U2UPHULj88F0+7cDBrZ
uIt4ImfkabBoxTzkbFpG583H+u/E7Eu9aqSs/cwoUe+StCmrqzWaTOTEcMYmzPhp
n+Sc8CnTXPnGFfiWeI8MgwT0PPzhAsP6CRDqWhqKa2NYOLQV07YRaXseVO6MGiKs
cpc/I1mbySKEwQdPzH/iV8oScLumZfnPdWO9lfsbl83kqK/20U6o2YpxJM02PbyW
xPFsq7lzw1uKA2wDrXKUxt4FMMgLG3/7FFXhEzn91QqhngLjYl/rNUssuHLoPj1P
rCy7Lobio3aP5ZMqz6WryFyNSwb/EkaseMsUBzXgqd+L6a8VTxaJW732jcZZroiF
DsGJ6x9nxUWO/203Nit4ZoORUSs9/1F3dmKh7Gc+PoGD4FapUB8fepmrY7+EF3fx
DTvf95xhszWYijqy7DwaNz9+j5LP2RIUZNoQAhVB/0/E6xyjyfqZ90bp4RjZsbgy
LcsUDFDYg2WD7rlc28sFWkz6GZdr1l0T08JcVLwyc6B49fFtHsufpaafItzRUZ6C
eWRgKRM+o/1Pcmqr4tTluCRVLERLiohEnMqE0yo7AgMBAAAGjQjBAMA8GA1UdEwEB
/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB0GA1UdDgQWBBSCIS1mxtteg4BXrzkWJ
d8RgnlRuAzANBgkqhkiG9w0BAQsFAAOCAGEAYN4auOfyYILVAzOBywaK8SJJ6ejq
kX/GM15oGQOGO0MBzwdw5AgeZYWR5hEit/UCI46uuR59H35s5r0l1ZUa8gWmr4UC
b6741jH/JclKyMeKqdmfS0mbEVeZkkMR3rYzpMzXjWR91M08KCyo0mpbqTfXERMQL
qiCA2CIV9+BB/AYm/7k29UMUA2Z44RGx2iBfRgB4ACGIHgAoYXhVqAEBj500mv/0
OJD7uNGzcgBJceaBxXntC6Z58hMLnPddDnskk7RI24Zf3lCGeOdA5jGokHZwYa+c
NywRtYK3qq4kNFtyDGkNzVmf9nGvnAvRCjj5BiKDUyUM/FHE5r7iOZULJK2v0ZXk
ltd0ZGtxTgl8qoXzIKNDOXZbbFD+mpwUHmUUihW9o4JFWklWatKcsWMy5WHgUyIO
pwpJ6st+H6jiYoD2EEVSmAYY3qXNL3+q1Ok+CHLsIwMCPKaq2LxndD0UF/tUSxfj
03k9bWtJySgOLnRQvwzZRjoQhsmnP+mg7H/rpXdYaXHmgwo38oZJar55CJD2AhZk
PuXaTH4MNMn5X7azKFGnpyuqSfqNZSIO42sTp5SjLVFteAxEy9/eCG/Oo2Sr05WE
1LISVHJ7liXMvGnjSG4N0MedJ5qq+BOS3R7fY581qRY27Iy4g/Q9iY/NtBde17MX
QRBDJ3NghVdJlGc=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICQDCCAeWgAwIBAgIMAVRI7yH9l1kN9QQKMAoGCCqGSM49BAMCMHExCzAJBgNV
BAYTAkhVMREwDwYDVQQHDAhCdWRhcGVzdDEWMBQGA1UECgwNTWljcm9zZWNgTHRk
LjEXMBUGA1UEYQwOVkFUSFUtMjM1ODQ0OTcxHjAcBgNVBAMMFwUtU3ppZ25vIFJv
b3QgQ0EgMjAxNzAeFw0xNzA4MjIxMjA3MDZaFw00MjA4MjIxMjA3MDZaMHExCzAJ
BgNVBAYTAkhVMREwDwYDVQQHDAhCdWRhcGVzdDEWMBQGA1UECgwNTWljcm9zZWNg
THRkLjEXMBUGA1UEYQwOVkFUSFUtMjM1ODQ0OTcxHjAcBgNVBAMMFwUtU3ppZ25v
IFJvb3QgQ0EgMjAxNzBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABJbcPYrYsHtv
xie+RJCxs1YVe45DJH0ahFnuY2iyxl6H0BVIHQIqrb1TotreOpCmYF9oMrWGQd+H
Wyx7xf58etqjYzBhMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB0G
A1UdDgQWBBSHERUI0arBeAyxr87GyZDvzvAEwDAfBgNVHSMEGDAWgBSHERUI0arB
eAyxr87GyZDvzvAEwDAKBggqhkJOPQQDAgNJADBGAiEAfVfd14pVCzbhhtT61Nlo
jbjcI4qKDdQvfezp7L9NbKgCIQDLpbQS+ue16M9+k/zzNY9vTlp8tLxOsvxyqltZ
+efcMQ==

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIFsDCCA5igAwIBAgIQFci9ZUdcr7iXAF7kBTk8nTANBgkqhkiG9w0BAQUFADBe
MQswCQYDVQQGEwJUVzEjMCEGA1UECgwaQ2h1bmdod2EgVGVsZWNVbSBDbY4sIEx0
ZC4xKjAoBgNVBAsMIWVQS0kgUm9vdCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAe
Fw0wNDEyMjAwMjM1ODQ0OTcxHjAcBgNVBAYTAiRXMSMw
IQYDVQQKDBpDaHVuZ2h3YSBUZWx1Y29tIENvLiwgTHRkLjEqMCgGA1UECwwhZVBL
SSBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MIICLjANBgkqhkiG9w0BAQEF
AAOCAg8AMIICCGKCAgEA4SUP7o3biDN1Z82tH306Tm2d0y8U82N0ywEhajfqhFAH
SyZbCUNsIZ5qyNUD9WBpj8zwIuQf5/dqIjG3LBXy4P4AakP/h2XGtRrBp0xtInAh
ijHyl3SJCRImHJ7K2RKilTza6We/CKBk49ZCt0Xvl/T29de1ShUCWH2YWEtgvM3X
DZoTM1PRYfl61dd4s5oz9wCGzh1NlDivqOx4UXCKXBCDUSH3ET00hl7ISM2XgY11
TBnsZfZrxQWh7kcT1rMhJ5QQctkO7q+RBNGMD+XPNjX12ruOzjjK9SXDk5wdJ
fzcq+Xd4z1TtW0ado4AOkUPB1ltfFLqfpo0kR0BZv3I4sjZsN/+Z0V0OWQqraffA
sgRFelQArr5T9rXn4fg8ozHSqf4hUmTFpmfwdQcGlBSBVcYn5AGPF8Fqcde+S/uU
WH1+ETOxQvdibBjWzwloPn9s9h6PYq2lY9sJpx8iQkEeb5mKPt5P0B6ebCIAZLS
nT0IFaUQAS2zMnaolQ2zepr7BxB4EW/hj8e6DyUadCrIHJhBmd8hh+iVBmoKs2pH
dmX2Os+PYhcZewoozRrSgx4hxyy/vv9haLdnG7t4TY3OZ+XkwY63I2binZB1NJip
NiuKmpS5nezMirH4JYlcWrYvjB9teSSnUmjDhDXiZo1jDiVN1Rmy5nk3pyKdVDEC
AwEAAANqMGgWHQYDVR0OBBYEFB4M97Zn8uGSJglFwFU5Lnc/QkqiMAwGA1UdEwQF
MAMBAf8wOQYEEZyoHAAQxMC8wLQIBADAJBgUrDgMCGGUAMAcGBWcQAwwAABBRFsMLH
ClZ87lt4DJX5GFPBphzYEDANBgkqhkiG9w0BAQUFAAOCAgEACbODU1kBPpVJufGB
uvl2ICO1J2B01GqZNF5sAFPZn/KmsSQHRGoqxqWOeBLor9lYGxMqXnmnbwoqZ6Y1
PwZpVnPDmZI+ymBV3QGypzqKOg4ZyYr8dW1P2WT+DZdjo2NQCCHGervJ8A9tDkP
JXtoUHRVnAxZfVo9QZQlUgRyWVMRnVvwdVxrsStZf0X4OFunHB2WyBEXYKCrC/
gpf36j36+uwtqSiUO1bd0IEursC9CBWMd1I0ltabrNMdjmEPNXubrijpC2JgQCA2
j6/7Nu4tCEoduL+bXPjqpRugc6bY+G7gMwRfaKonh+3ZwZCc7b3jajWvY9+rGNm6
5ulK6lCKD2GTHultGelwLDWSXQ62B68ZgI9HkFFLLk3dheLSCIIF5r8GrBQAuUB
o2M3IUxExJtRmREOC5wGj1QupyheRDmHVi03vYVElOEMSsyycw5KFNGHLD7ibSkNS

/jQ6fbjpKdx2qcgw+BRxgMYeNkh0IkFch4LoGHGLQYIE535YW6i4jRPpp2zDR+2z
Gp1iro2C6pSe3VkQw63d4k3jMdXH7OjysP6SHhYKGvzZ8/gntsm+HbRsZJB/9OTE
W9c3rkIO3aQab3yIVMUWbuF6aC74Or8NpDyJO3inTmODBCEIZ43ygknQW/2xzQ+D
hNQ+IIX3Sj0rnP0qCglN6oH4EZw=
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICKzCCAbGgAwIBAgIKKe3G2gla4EnycqDAKBggqhkjOPQQDAzBaMQswCQYDVQQG
EwJVUzETMBEGA1UECXMkZW1TaWduIFBLSTEUMBGA1UEChMLZU11ZGhyYSBjbmMx
IDAeBgNVBAMTF2VtU2lnbiBFQ0MgUm9vdCBDQSAiIEMzMb4XDTE4MDIxODE4MzAw
MFoXDTQzMdIxODE4MzAwMFowWjELMAkGA1UEBhMCVVMxEzARBgNVBAsTCmVtU2ln
biBQS0kxFDASBgNVBAoTC2VNdWRocmEgSW5jMSAwHgYDVQQDExdlbVNpZ24gRUND
IFJvb3QgQ0EgLSBDMzB2MBAGByqGSM49AgEGBSuBBAAiA2IABP2IYa57JhAd6bci
MK4G9IGzsUJxlTm801Ljr6/58pc1kjZGDoeVjBk5Wum739D+yAdBPLtVb4Ojavti
sIGJAnB9SMVK4+kiVCJNk7tCDK93nComfddhEc5lx/h//vXyqaNCMEAwHQYDVR0O
BBYEFtaSNCAIEDyqOkAB2kZd6fmw/TPMA4GA1UdDwEB/wQEAWIBBjAPBgNVHRMB
Af8EBTADAQH/MAoGCCqGSM49BAMDA2gAMGUQM0C02C8Cif22TGK6Q04ThHK1rt0c
3ta13FaPWEBaLd4gTCKDypOofu4SQMfWh0/434UCMBwUZOR8loMRnLDRWmFLpg9J
0wD8ofzkpf9/rdcw0Md3f76BB1UwUCAU9Vc4CqgxUQ==
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICTjCCAdOgAwIBAgIKPPYHqWhwDtqLhDAKBggqhkjOPQQDAzBrMQswCQYDVQQG
EwJJTjETMBEGA1UECXMkZW1TaWduIFBLSTEIMCMGA1UEChMcZU11ZGhyYSBUZWNp
bm9sb2dpZXMGtGtXRlZDEgMB4GA1UEAxMXZW1TaWduIEVDQyBSb290IENBIC0g
RzMwHhcnMTgwMjE4MTgzMDAwWhcNNDMwMjE4MTgzMDAwWjBrMQswCQYDVQQGEwJJ
TjETMBEGA1UECXMkZW1TaWduIFBLSTEIMCMGA1UEChMcZU11ZGhyYSBUZWNpZm9s
b2dpZXMGtGtXRlZDEgMB4GA1UEAxMXZW1TaWduIEVDQyBSb290IENBIC0gRzMw
djAQBgcqhkJOPQIBBgUrgQQAIGNiAAQjPQy4LRL1KPOxst3iAhKAnjlfSU2fySU0
WXTsuwYc58Byr+iuL+FBVlcUqEqy6HyC5ltqtdyzdc6LBtCGI79G1Y4PPwT01xyS
fvalY8L1X44uT6EYgQIrMgqCZH0Wk9GjQjBAMB0GA1UdDgQWBBr8XQKEE9TMipuB
zhccLikEnEhJqJAObgNVHQ8BAf8EBAMCAQYwDwYDVR0TAAH/BAUwAwEB/zAKBggq
hkjOPQQDAwNpADBmAjEAvvNhzwIQHWSVB7gYboiFBS+DCBeQyh+KTOgNG3qxrWB
CUfvO6wIBHxcmbHtRwfSAjEAnbpV/KIK6O3t5nYBQnvI+GDZjVGLVTv7jHvrZQnD
+JbNR6iC8hZVdyR+EhCVBCyj
-----END
CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDczCCAlugAwIBAgILAK7PALrEzzL4Q7IwDQYJKoZIhvcNAQELBQAwVjELMAkG
A1UEBhMCVVMxEzARBgNVBAsTCmVtU2lnbiBQS0kxFDASBgNVBAoTC2VNdWRocmEg
SW5jMRwwGgYDVQQDExNlbVNpZ24gUm9vdCBDQSAiIEMxMB4XDTE4MDIxODE4MzAw
MFoXDTQzMdIxODE4MzAwMFowWjELMAkGA1UEBhMCVVMxEzARBgNVBAsTCmVtU2ln
biBQS0kxFDASBgNVBAoTC2VNdWRocmEgSW5jMRwwGgYDVQQDExNlbVNpZ24gUm9v
dCBDQSAiIEMxMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAz+upufGZ
BczYKCFK83M0UYRWEPWgTywS4/oTmifQz/l5GnRfHXk5/Fv4cI7gklL35CX5VIPZ

HdPIWoU/Xse2B+4+wM6ar6xWQio5JXDWv7V7Nq2s9nPczdcdioOl+yuQFTdrHCZH
3DspVpNqs8FqOp099cGXOfgFixwR4+S0uF2FHYP+eF8LRWgYSKVGczQ7/g/IdrvH
GPMF0Ybzhe3nudkyrVWlZqa2kbBPrH4VI5b2P/AgNBbeCsbEBEV5f6f9vtKppa+c
xSMq9zwhbL2vj07FOrLzNBL834AaSaTUqZX3noleoomslMuoaJuvimUnzYnu3Yy1
aylwQ6BpC+S5DwIDAQABo0IwQDAdBgNVHQ4EFgQU/qHgcB4qAzlSWkK+XJGFehiq
TbUwDgYDVR0PAQH/BAQDAgEGMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEL
BQADggEBAMJKVvoVIXsoounlHfv4LcQ5lkFMOycsxGwYFYDGrK9HWS8mC+M2sO87
/kOXSTKZEhVb3xEp/6tT+LvBeA+snFOvV71ojD1pM/CjoCNjO2RnIkSt1XHLVip4
kqNPEjE2NuLe/gDEo2APJ62gsIq1NnpSob0n9CAnYuhNlCQT5AoE6TyrLshDCUrG
YQTISTR+08TI9Q/Aqum6VF7zYytPT1DU/r17mYw9wC68AivTxEDkigcxHpvOJpkT
+xHqmiIMERNHXhuBUDDIlhJu58tBf5E7oke3VIAb3ADMmpDqw8NQBmIMMMAVSKeo
WXzhriKi4gp6D/piq1JM4fHfyr6DDUI=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDIDCCAnygAwIBAgIKMfXkYgxsWO3W2DANBgkqhkiG9w0BAQsFADBNMQswCQYD
VQQGEwJJTjETMBEGA1UECzMkZW1TaWduIFBLSSTELMCMGA1UEChMcZU11ZGhyYSBU
ZWNobm9sb2dpZXMGtGltXRlZDEcMBoGA1UEAxMTZW1TaWduIFJvb3QgQ0EgLSBH
MTAeFw0xODAyMTgxODMwMDBaFw00MzAyMTgxODMwMDBaMGcxGzAJBgNVBAYTAklO
MRMwEQYDVQQLZWplbVNPZ24gUETJMSUwIwYDVQQKEExlTXVkaHJhIFRlY2hub2xv
Z2llcyBMAW1pdGVkMRwwGgYDVQQDEXNlbnVNPZ24gUm9vdCBDQSAiECCMIIBIjAN
BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAK0u76WaK7p1b1TST0Bsew+eeuGQz
f2N4aLTNLnf115sgxk0pvLZoYIr3IZpWNVrzdr3YzZr/k1ZLpVkJGoZM0Kd0WNHVO
8oG0x5ZOrRkVUkr+PHB1cM2vK6sVmjm8qrOLqs1D/fXqcP/tzxE7lM5OMhbTI0Aq
d7OvPAEsbO2ZLlvZTmmYsvePQbAyeGHWDV/D+qJakh1cF+ZwPjXnorfCYuKrpDhM
tTk1b+oDafo6VGiFbddyL0NVHpeNDtjVaqSW0RM8LHhQ6DqS0hdW5TUaQBw+jSzt
Od9C4INBdN+jzcKGYEho42kLVACL5HZpIQ15TjQIXhTCzLG3rdd8cIrHhQIDAQAB
o0IwQDAdBgNVHQ4EFgQU++8Nhp6w492pufEhF38+/PB3KxowDgYDVR0PAQH/BAQD
AgEGMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQELBQADggEBAFn/8oz1h31x
PaOfG1vR2vjTnGs2vZupYeveFix0PZ7mdrXuqe8QhfnPZHr5X3dPpzzx5KsbEjM
wiI/aTvFthUvozXGaCocV685743QNcMYDHsAVhzNixl03r4PEuDQqqE/AjSxcM6d
GNYIAwlg7mDgfrbESQRRfXBgvKqy/3lyeqYdPV8q+Mri/Tm3R7nrft8EI6/6nAYH
6ftjk4BAtcZsCjEozgyfz7MjNYBBjWzEN3uBL4ChQEKF6dk4jeihU80Bv2noWgby
RQuQ+q7hv53yrlc8pa6yVvSLZUDp/TGBLPQ5Cdjua6e0ph0VpZj3AYHYhX3zUVxx
iN66zB+AfkO=

-----END

CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIICDzCCAZWgAwIBAgIUbmq8WapTvpq5Z6LSa6Q75m0c1towCgYIKoZIzj0EAwMw
RzELMAkGA1UEBhMCQ04xHDAaBgNVBAoTE2lUcnVzQ2hpbmEgQ28uLEx0ZC4xGjAY
BgNVBAMTEXZUcnVzIEVDQyBSb290IENBMB4XDTE4MDczMTA3MjY0NFoXDTQzMDCz
MTA3MjY0NFowRzELMAkGA1UEBhMCQ04xHDAaBgNVBAoTE2lUcnVzQ2hpbmEgQ28u
LEx0ZC4xGjAYBgNVBAMTEXZUcnVzIEVDQyBSb290IENBMHYwEAYHKoZIzj0CAQYF
K4EEACIDYgAEZVBKrox5lkqqHAjDo6LN/llWQXf9JpRCux3NCNtzt188+cToL0
v/hhJoVs1oVbcnDS/dtitN9Ti72xRFhiQgnH+n9bEOf+QP3A2MMrMudwpremiFud
e4BdS49nTPEQo0IwQDAdBgNVHQ4EFgQUmDnNvtijPeyq+GtJK97fKHbH88wDwYD

1.42.1 Available under license :

bzip2 applet in busybox is based on lightly-modified source
of bzip2 version 1.0.4. bzip2 source is distributed
under the following conditions (copied verbatim from LICENSE file)

=====

This program, "bzip2", the associated library "libbzip2", and all
documentation, are copyright (C) 1996-2006 Julian R Seward. All
rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product
documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must
not be misrepresented
as being the original software.
4. The name of the author may not be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, Cambridge, UK.

jseward@bzip.org

bzip2/libbzip2 version 1.0.4 of 20 December 2006

--- A note on GPL versions

BusyBox is distributed under version 2 of the General Public License (included in its entirety, below). Version 2 is the only version of this license which this version of BusyBox (or modified versions derived from this one) may be distributed under.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's

software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy,

distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this

License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot

impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING

WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of
Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.43 shadow 4.17.3-r0

1.43.1 Available under license :

SPDX-License-Identifier: BSD-3-Clause

All files under this project either

1. fall under the BSD 3 clause license (by default).
2. carry an SPDX header declaring what license applies.

or

3. list a full custom license

This software is originally

* Copyright (c) 1989 - 1994, Julianne Frances Haugh

* All rights reserved.

- *
 - * Redistribution and use in source and binary forms, with or without
 - * modification, are permitted provided that the following conditions
 - * are met:
 - * 1. Redistributions of source code must retain the above copyright
 - * notice, this list of conditions and the following disclaimer.
 - * 2. Redistributions in binary form must reproduce the above copyright
 - * notice, this list of conditions and the following disclaimer in the
 - * documentation and/or other materials provided with the distribution.
 - * 3. The name of the copyright holders or contributors may not be used to
 - * endorse or promote products derived from this software without
 - * specific
 - prior written permission.
- *
 - * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 - * ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 - * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
 - * PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
 - * HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 - * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 - * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 - * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 - * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 - * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
 - * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License

All new source code committed to the shadow project is assumed to be made available under the [BSD-3-Clause](../COPYING) license unless the submitter specifies another license at that time. The shadow maintainers reserve the right to refuse a submission if the license is deemed incompatible with the goals of the project.

****Note**:** old code may be made available under another license, check the license tag for each file to get additional information.

1.44 libncursesw 6.5_p20250503-r0

1.44.1 Available under license :

Upstream source <https://invisible-island.net/ncurses/Ada95.html>

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *

Copyright: 2017-2023,2024 by Thomas E. Dickey

Copyright: 1998-2016,2017 Free Software Foundation, Inc.

License: X11

Files: aclocal.m4 package

Copyright: 2010-2023,2024 by Thomas E. Dickey

License: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, distribute with modifications, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included

in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

Files: install-sh

Copyright: 1994 X Consortium

License: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except

as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

FSF changes to this file are in the public domain.

Calling this script install-sh is preferred over install.sh, to prevent `make` implicit rules from creating a file called install from it when there is no Makefile.

This script is compatible with the BSD install script, but was written from scratch. It can only install one file at a time, a restriction shared with many OS's install programs.

On Debian systems, the complete text of the GNU General Public License can be found in '/usr/share/common-licenses/GPL-2'

-- vile: txtmode file-encoding=utf-8

Upstream source <https://invisible-island.net/ncurses/ncurses-examples.html>

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *

Copyright: 2017-2023,2024 Thomas E. Dickey

Copyright: 1998-2016,2017 Free Software Foundation, Inc.

License: X11

Files: aclocal.m4 package

Copyright: 2003-2023,2024 by Thomas E. Dickey

License: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, distribute with modifications, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

Files: install-sh
Copyright: 1994 X Consortium
License: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

FSF changes to this file are in the public domain.

Calling this script install-sh is preferred over install.sh, to prevent `make` implicit rules from creating a file called install from it

when there is no Makefile.

This script is compatible with the BSD install script, but was written from scratch. It can only install one file at a time, a restriction shared with many OS's install programs.

On Debian systems, the complete text of the GNU General Public License can be found in '/usr/share/common-licenses/GPL-2'

-- vile: txtmode file-encoding=utf-8

Upstream source <https://invisible-island.net/ncurses/ncurses.html>

This package is used for testing builds of ncurses.

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *

Copyright: 2017-2023,2024 by Thomas E. Dickey

Copyright: 1998-2016,2017 Free Software Foundation, Inc.

License: X11

Files: aclocal.m4 package

Copyright: 1996-2023,2024 by Thomas E. Dickey

License: X11

Files: doc/html/NCURSES-Programming-HOWTO.html

Copyright: 2001 by Pradeep Padala

License: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, distribute with modifications, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

Files:

install-sh

Copyright: 1994 X Consortium

License: X11

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

Files: progs/tset.c ncurses/tinfo/read_termcap.c

Copyright: 1980,1991,1992,1993 The Regents of the University of California.

License: BSD

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list

of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-- vile: txtmode file-encoding=utf-8

Copyright 2018-2023,2024 Thomas E. Dickey

Copyright 1998-2017,2018 Free Software Foundation, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, distribute with modifications, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

```
-- vile:txtmode fc=72
-- $Id: COPYING,v 1.13 2024/01/05 21:13:17 tom Exp $
```

1.45 ncurses 6.5_p20250503-r0

1.45.1 Available under license :

```
# Contributor: Patrycja Rosa <alpine@ptrcnull.me>
# Maintainer: Patrycja Rosa <alpine@ptrcnull.me>
pkgname=bsd-compat-headers
pkgver=0.7.2
pkgrel=6
pkgdesc="BSD compatibility headers (cdefs, queue, tree)"
url="https://gitlab.alpinelinux.org/alpine/aports"
arch="noarch"
license="BSD-2-Clause AND BSD-3-Clause"
source="
cdefs.h
queue.h
tree.h
"
builddir="$srcdir"
options="!check" # just headers

package() {
    mkdir -p "$pkgdir"
    install -Dm644 -t "$pkgdir"/usr/include/sys \
        cdefs.h queue.h tree.h
}
sha512sums="
37c8fc73c7aea7b490f7850927e2bb91d12137c9e59e22c084146d515696dbc7973b5de92f4c987ba080dd2502ba8312
7006442c3f019b6447a620c0cae73178  cdefs.h
2f0d5e6e4dc3350285cf17009265dddcbe12431c111868eea39bc8cb038ab7c1f2acacbb21735c4e9d4a1fd106a8fc0f86
11ea33987d4faba37dde5ce6da0750  queue.h
d9ac210d81feb8ad2655bc80fb065d3fe20ae4417b32b4a1711e6738a4870140005c13373b5d1846ef3ce5ae6da45f2da
cef2092881eded0a2e94f6a07752ef3  tree.h
"

# Automatically generated by apkbuild-cpan, template 4
# Contributor: Valery Kartel <valery.kartel@gmail.com>
# Maintainer: Celeste <cielesti@protonmail.com>
maintainer="Celeste <cielesti@protonmail.com>"
pkgname=perl-bsd-resource
pkgver=1.2911
pkgrel=10
#_pkgreal is used by apkbuild-cpan to find modules at MetaCpan
_pkgreal=BSD-Resource
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="https://metacpan.org/release/BSD-Resource/"
```

```

arch="all"
license="Artistic-2.0 OR LGPL-2.0"
depends="perl"
makedepends="perl-dev"
subpackages="$pkgname-doc"
source="https://cpan.metacpan.org/authors/id/J/JH/JHI/BSD-Resource-$pkgver.tar.gz"
builddir="$srcdir/${_pkgname}-${pkgver}"

build() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    PERL_MM_USE_DEFAULT=1 perl -I. Makefile.PL \
        INSTALLDIRS=vendor \
        NO_PACKLIST=1 \
        NO_PERLLOCAL=1
    make
}

check() {
    export CFLAGS=$(perl -MConfig -E 'say $Config{ccflags}')
    make test
}

package() {
    make DESTDIR="$pkgdir" install
}

sha512sums="
d0032d41c7c0468ed1c6d8f57b885f6cb97a5039d754c8cb60b2067daedaf53bd15fb6561a3d0f828df16dfa5417f663b
8065ba65f4fea16dc9262728b3b6b85
    BSD-Resource-1.2911.tar.gz
"

# Contributor: Fabian Affolter <fabian@affolter-engineering.ch>
# Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>
pkgname=py3-flake8-copyright
_pkgname=flake8-copyright
pkgver=0.2.4
pkgrel=3
pkgdesc="Extension for flake8 which checks for copyrights"
options="!check" # No testsuite
url="https://github.com/savoirfairelinux/flake8-copyright"
arch="noarch"
license="MIT"
depends="py3-flake8 py3-setuptools"
makedepends="py3-gpep517 py3-wheel"
subpackages="$pkgname-pyc"
source="https://files.pythonhosted.org/packages/source/${_pkgname:0:1}/${_pkgname}/${_pkgname}-${pkgver}.tar.gz"
builddir="$srcdir/${_pkgname}-${pkgver}"

```

```

replaces="py-flake8-copyright"          # Backwards compatibility
provides="py-flake8-copyright=$pkgver-r$pkgrel" # Backwards compatibility

build() {
    gpep517 build-wheel \
        --wheel-dir .dist \
        --output-fd 3 3>&1 >&2
}

package() {
    gpep517 install-wheel --destdir "$pkgdir" \
        .dist/*.whl
}

sha512sums="
a6ab47e1bb715618f075c51f398e18180404871b3b0faf1c2d30701d5203db6bc23771bf22ffb5bafcd8ee856b9b64237
b316bd503ae7dceed6ca284ccd5a74d
    flake8-copyright-0.2.4.tar.gz
"

#ifndef _SYS_CDEFS_H_
#define _SYS_CDEFS_H_

#warning usage of non-standard #include <sys/cdefs.h> is deprecated

#undef __P
#undef __PMT

#define __P(args) args
#define __PMT(args) args

#define __CONCAT(x,y) x ## y
#define __STRING(x) #x

#ifdef __cplusplus
# define __BEGIN_DECLS extern "C" {
# define __END_DECLS }
#else
# define __BEGIN_DECLS
# define __END_DECLS
#endif

#if defined(__GNUC__) && !defined(__cplusplus)
# define __THROW __attribute__((__nothrow__))
# define __NTH(fct) __attribute__((__nothrow__)) fct
#else
# define __THROW
# define __NTH(fct) fct
#endif

```

```

#endif /* _SYS_CDEFS_H_ */
# Maintainer: fossdd <fossdd@pwned.life>
pkgname=cargo-license
pkgver=0.6.1
pkgrel=1
pkgdesc="Cargo subcommand to see license of dependencies"
url="https://github.com/onur/cargo-license"
arch="all"
license="MIT"
makedepends="cargo cargo-auditable"
source="$pkgname-$pkgver.tar.gz::https://crates.io/api/v1/crates/cargo-license/$pkgver/download"

prepare() {
    default_prepare

    cargo fetch --target="$CTARGET" --locked
}

build() {
    cargo auditable build --release --frozen
}

check() {
    cargo test --frozen
}

package() {
    install -Dm755 target/release/cargo-license -t "$pkgdir"/usr/bin
}

sha512sums="
36215145e79965be090ae2f5ce6f5eb208465eb67a94c26bed1fe70a131c46c6dc35695e30852759b7366883fbc3caf00
272d796ac634f0a5e1d49232b6b29d5 cargo-license-0.6.1.tar.gz
"

/* $NetBSD: tree.h,v 1.20 2013/09/14 13:20:45 joerg Exp $ */
/* $OpenBSD: tree.h,v 1.13 2011/07/09 00:19:45 pirofti Exp $ */
/* Modified by Void Linux. */
/*
* Copyright 2002 Niels Provos <provos@citi.umich.edu>
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright

```

- * notice, this list of conditions and the following disclaimer in the
- * documentation and/or other materials provided with the distribution.
- *
- * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR
- * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
- * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
- * IN NO EVENT SHALL THE
- AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
- * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
- * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
- * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
- * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
- * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
- * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
- */

```
#ifndef _SYS_TREE_H_
#define _SYS_TREE_H_
```

```
#ifdef __GNUC__
#define __GNUC_PREREQ__(x, y) \
    ((__GNUC__ == (x) && __GNUC_MINOR__ >= (y)) || \
     (__GNUC__ > (x)))
#else
#define __GNUC_PREREQ__(x, y) 0
#endif
```

```
#if __GNUC_PREREQ__(2, 7) || defined(__lint__)
#define _sys_tree_h_unused __attribute__((__unused__))
#else
#define _sys_tree_h_unused /* delete */
#endif
```

```
/*
```

- * This file defines data structures for different types of trees:
- * splay trees and red-black trees.
- *
- * A splay tree is a self-organizing
- data structure. Every operation
- * on the tree causes a splay to happen. The splay moves the requested
- * node to the root of the tree and partly rebalances it.
- *
- * This has the benefit that request locality causes faster lookups as
- * the requested nodes move to the top of the tree. On the other hand,
- * every lookup causes memory writes.
- *
- * The Balance Theorem bounds the total access time for m operations
- * and n inserts on an initially empty tree as $O((m + n) \lg n)$. The

* amortized cost for a sequence of m accesses to a splay tree is $O(\lg n)$;

*

* A red-black tree is a binary search tree with the node color as an

* extra attribute. It fulfills a set of conditions:

* - every search path from the root to a leaf consists of the

* same number of black nodes,

* - each red node (except for the root) has a black parent,

* - each leaf node is black.

*

* Every operation on a red-black tree is bounded as $O(\lg n)$.

* The maximum height of a red-black tree is $2\lg(n+1)$.

*/

```
#define SPLAY_HEAD(name, type)  \
struct name {                  \
    struct type *sph_root; /* root of the tree */ \
}
```

```
#define SPLAY_INITIALIZER(root) \
{ NULL }
```

```
#define SPLAY_INIT(root) do { \
    (root)->sph_root = NULL; \
} while (/*CONSTCOND*/ 0)
```

```
#define SPLAY_ENTRY(type) \
struct { \
    struct type *spe_left; /* left element */ \
    struct type *spe_right; /* right element */ \
}
```

```
#define SPLAY_LEFT(elm, field) (elm)->field.spe_left
#define SPLAY_RIGHT(elm, field) (elm)->field.spe_right
#define SPLAY_ROOT(head) (head)->sph_root
#define SPLAY_EMPTY(head) (SPLAY_ROOT(head) == NULL)
```

```
/* SPLAY_ROTATE_{LEFT,RIGHT} expect that tmp hold SPLAY_{RIGHT,LEFT} */
#define SPLAY_ROTATE_RIGHT(head, tmp, field) do { \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(tmp, field); \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
} while (/*CONSTCOND*/ 0)
```

```
#define SPLAY_ROTATE_LEFT(head, tmp, field) do { \
    SPLAY_RIGHT((head)->sph_root, \
    field) = SPLAY_LEFT(tmp, field); \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    (head)->sph_root = tmp; \
}
```



```

} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKLEFT(head, tmp, field) do { \
    SPLAY_LEFT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \
    (head)->sph_root = SPLAY_LEFT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_LINKRIGHT(head, tmp, field) do { \
    SPLAY_RIGHT(tmp, field) = (head)->sph_root; \
    tmp = (head)->sph_root; \
    (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field); \
} while (/*CONSTCOND*/ 0)

#define SPLAY_ASSEMBLE(head, node, left, right, field) do { \
    SPLAY_RIGHT(left, field) = SPLAY_LEFT((head)->sph_root, field); \
    SPLAY_LEFT(right, field) = SPLAY_RIGHT((head)->sph_root, field); \
    SPLAY_LEFT((head)->sph_root, field) = SPLAY_RIGHT(node, field); \
    SPLAY_RIGHT((head)->sph_root, field) = SPLAY_LEFT(node, field); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */

#define
SPLAY_PROTOTYPE(name, type, field, cmp) \
void name##_SPLAY(struct name *, struct type *); \
void name##_SPLAY_MINMAX(struct name *, int); \
struct type *name##_SPLAY_INSERT(struct name *, struct type *); \
struct type *name##_SPLAY_REMOVE(struct name *, struct type *); \
\
/* Finds the node with the same key as elm */ \
static __inline struct type * \
name##_SPLAY_FIND(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) \
        return(NULL); \
    name##_SPLAY(head, elm); \
    if ((cmp)(elm, (head)->sph_root) == 0) \
        return (head->sph_root); \
    return (NULL); \
} \
\
static __inline _sys_tree_h_unused struct type * \
name##_SPLAY_NEXT(struct name *head, struct type *elm) \
{ \
    name##_SPLAY(head, elm); \
    if (SPLAY_RIGHT(elm, field) != NULL) { \
        elm = SPLAY_RIGHT(elm, field); \
    } \
}

```

```

while (SPLAY_LEFT(elm, field) != NULL) { \
    elm = SPLAY_LEFT(elm, field); \
} \
}
else \
    elm = NULL; \
return (elm); \
} \
\

static _sys_tree_h_unused __inline struct type * \
name##_SPLAY_MIN_MAX(struct name *head, int val) \
{ \
    name##_SPLAY_MINMAX(head, val); \
    return (SPLAY_ROOT(head)); \
}

/* Main splay operation.
 * Moves node close to the key of elm to top
 */
#define SPLAY_GENERATE(name, type, field, cmp) \
struct type * \
name##_SPLAY_INSERT(struct name *head, struct type *elm) \
{ \
    if (SPLAY_EMPTY(head)) { \
        SPLAY_LEFT(elm, field) = SPLAY_RIGHT(elm, field) = NULL; \
    } else { \
        int __comp; \
        name##_SPLAY(head, elm); \
        __comp = (cmp)(elm, (head)->sph_root); \
        if (__comp < 0) { \
            SPLAY_LEFT(elm, field) = SPLAY_LEFT((head)->sph_root, field); \
            SPLAY_RIGHT(elm, field) = (head)->sph_root; \
            SPLAY_LEFT((head)->sph_root, field) = NULL; \
        } else if (__comp > 0) { \
            SPLAY_RIGHT(elm, \
field) = SPLAY_RIGHT((head)->sph_root, field); \
            SPLAY_LEFT(elm, field) = (head)->sph_root; \
            SPLAY_RIGHT((head)->sph_root, field) = NULL; \
        } else \
            return ((head)->sph_root); \
    } \
    (head)->sph_root = (elm); \
    return (NULL); \
} \
\

struct type * \
name##_SPLAY_REMOVE(struct name *head, struct type *elm) \
{ \

```

```

struct type *__tmp; \
if (SPLAY_EMPTY(head)) \
    return (NULL); \
name##_SPLAY(head, elm); \
if ((cmp)(elm, (head)->sph_root) == 0) { \
    if (SPLAY_LEFT((head)->sph_root, field) == NULL) { \
        (head)->sph_root = SPLAY_RIGHT((head)->sph_root, field);\
    } else { \
        __tmp = SPLAY_RIGHT((head)->sph_root, field);\
        (head)->sph_root = SPLAY_LEFT((head)->sph_root, field);\
        name##_SPLAY(head, elm); \
        SPLAY_RIGHT((head)->sph_root, field) = __tmp; \
    } \
    return (elm); \
} \
return (NULL); \
} \
\
void \
name##_SPLAY(struct
name *head, struct type *elm) \
{ \
    struct type __node, *__left, *__right, *__tmp; \
    int __comp; \
    \
    SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
    __left = __right = &__node; \
    \
    while ((__comp = (cmp)(elm, (head)->sph_root)) != 0) { \
        if (__comp < 0) { \
            __tmp = SPLAY_LEFT((head)->sph_root, field);\
            if (__tmp == NULL) \
                break; \
            if ((cmp)(elm, __tmp) < 0){ \
                SPLAY_ROTATE_RIGHT(head, __tmp, field);\
                if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
                    break; \
            } \
            SPLAY_LINKLEFT(head, __right, field); \
        } else if (__comp > 0) { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field);\
            if (__tmp == NULL) \
                break; \
            if ((cmp)(elm, __tmp) > 0){ \
                SPLAY_ROTATE_LEFT(head, __tmp, field);\
                if (SPLAY_RIGHT((head)->sph_root, field) == NULL)\
                    break; \
            } \
        } \
    } \
} \

```

```

    SPLAY_LINKRIGHT(head, __left, field); \
} \
} \
SPLAY_ASSEMBLE(head,
&__node, __left, __right, field); \
} \
\
/* Splay with either the minimum or the maximum element \
* Used to find minimum or maximum element in tree. \
*/ \
void name##_SPLAY_MINMAX(struct name *head, int __comp) \
{ \
    struct type __node, *__left, *__right, *__tmp; \
\
    SPLAY_LEFT(&__node, field) = SPLAY_RIGHT(&__node, field) = NULL;\
    __left = __right = &__node; \
\
    while (1) { \
        if (__comp < 0) { \
            __tmp = SPLAY_LEFT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if (__comp < 0){ \
                SPLAY_ROTATE_RIGHT(head, __tmp, field); \
                if (SPLAY_LEFT((head)->sph_root, field) == NULL)\
                    break; \
            } \
            SPLAY_LINKLEFT(head, __right, field); \
        } else if (__comp > 0) { \
            __tmp = SPLAY_RIGHT((head)->sph_root, field); \
            if (__tmp == NULL) \
                break; \
            if (__comp > 0) { \
                SPLAY_ROTATE_LEFT(head, __tmp, field); \
                if (SPLAY_RIGHT((head)->sph_root,
field) == NULL)\
                    break; \
            } \
            SPLAY_LINKRIGHT(head, __left, field); \
        } \
    } \
    SPLAY_ASSEMBLE(head, &__node, __left, __right, field); \
}

#define SPLAY_NEGINF -1
#define SPLAY_INF 1

#define SPLAY_INSERT(name, x, y) name##_SPLAY_INSERT(x, y)

```

```

#define SPLAY_REMOVE(name, x, y) name##_SPLAY_REMOVE(x, y)
#define SPLAY_FIND(name, x, y) name##_SPLAY_FIND(x, y)
#define SPLAY_NEXT(name, x, y) name##_SPLAY_NEXT(x, y)
#define SPLAY_MIN(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_NEGINF))
#define SPLAY_MAX(name, x) (SPLAY_EMPTY(x) ? NULL \
    : name##_SPLAY_MIN_MAX(x, SPLAY_INF))

#define SPLAY_FOREACH(x, name, head) \
    for ((x) = SPLAY_MIN(name, head); \
        (x) != NULL; \
        (x) = SPLAY_NEXT(name, head, x))

/* Macros that define a red-black tree */
#define RB_HEAD(name, type) \
    struct name { \
        struct type *rbh_root; /* root of the tree */ \
    }

#define RB_INITIALIZER(root) \
    { \
        NULL \
    }

#define RB_INIT(root) do { \
    (root)->rbh_root = NULL; \
} while (/*CONSTCOND*/ 0)

#define RB_BLACK 0
#define RB_RED 1
#define RB_ENTRY(type) \
    struct { \
        struct type *rbe_left; /* left element */ \
        struct type *rbe_right; /* right element */ \
        struct type *rbe_parent; /* parent element */ \
        int rbe_color; /* node color */ \
    }

#define RB_LEFT(elm, field) (elm)->field.rbe_left
#define RB_RIGHT(elm, field) (elm)->field.rbe_right
#define RB_PARENT(elm, field) (elm)->field.rbe_parent
#define RB_COLOR(elm, field) (elm)->field.rbe_color
#define RB_ROOT(head) (head)->rbh_root
#define RB_EMPTY(head) (RB_ROOT(head) == NULL)

#define RB_SET(elm, parent, field) do { \
    RB_PARENT(elm, field) = parent; \
    RB_LEFT(elm, field) = RB_RIGHT(elm, field) = NULL; \
    RB_COLOR(elm, field) = RB_RED; \
} while (0)

```

```

} while (/*CONSTCOND*/ 0)

#define RB_SET_BLACKRED(black, red, field) do { \
    RB_COLOR(black, field) = RB_BLACK; \
    RB_COLOR(red, field) \
    = RB_RED; \
} while (/*CONSTCOND*/ 0)

#ifndef RB_AUGMENT
#define RB_AUGMENT(x) do {} while (/*CONSTCOND*/ 0)
#endif

#define RB_ROTATE_LEFT(head, elm, tmp, field) do { \
    (tmp) = RB_RIGHT(elm, field); \
    if ((RB_RIGHT(elm, field) = RB_LEFT(tmp, field)) != NULL) { \
        RB_PARENT(RB_LEFT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_LEFT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
    RB_AUGMENT(tmp); \
    if ((RB_PARENT(tmp, field))) \
        RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

#define RB_ROTATE_RIGHT(head, elm, tmp, field) do { \
    (tmp) = RB_LEFT(elm, field); \
    if ((RB_LEFT(elm, field) = RB_RIGHT(tmp, field)) \
    != NULL) { \
        RB_PARENT(RB_RIGHT(tmp, field), field) = (elm); \
    } \
    RB_AUGMENT(elm); \
    if ((RB_PARENT(tmp, field) = RB_PARENT(elm, field)) != NULL) { \
        if ((elm) == RB_LEFT(RB_PARENT(elm, field), field)) \
            RB_LEFT(RB_PARENT(elm, field), field) = (tmp); \
        else \
            RB_RIGHT(RB_PARENT(elm, field), field) = (tmp); \
    } else \
        (head)->rbh_root = (tmp); \
    RB_RIGHT(tmp, field) = (elm); \
    RB_PARENT(elm, field) = (tmp); \
} while (/*CONSTCOND*/ 0)

```

```

RB_AUGMENT(tmp); \
if ((RB_PARENT(tmp, field))) \
    RB_AUGMENT(RB_PARENT(tmp, field)); \
} while (/*CONSTCOND*/ 0)

/* Generates prototypes and inline functions */
#define RB_PROTOTYPE(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp,)
#define RB_PROTOTYPE_STATIC(name, type, field, cmp) \
    RB_PROTOTYPE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_PROTOTYPE_INTERNAL(name, type, field, cmp, attr) \
    attr void name##_RB_INSERT_COLOR(struct name *, struct \
        type *); \
    attr void name##_RB_REMOVE_COLOR(struct name *, struct type *, struct type *); \
    attr struct type *name##_RB_REMOVE(struct name *, struct type *); \
    attr struct type *name##_RB_INSERT(struct name *, struct type *); \
    attr struct type *name##_RB_FIND(struct name *, struct type *); \
    attr struct type *name##_RB_NFIND(struct name *, struct type *); \
    attr struct type *name##_RB_NEXT(struct type *); \
    attr struct type *name##_RB_PREV(struct type *); \
    attr struct type *name##_RB_MINMAX(struct name *, int); \
    \

/* Main rb operation.
 * Moves node close to the key of elm to top
 */
#define RB_GENERATE(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp,)
#define RB_GENERATE_STATIC(name, type, field, cmp) \
    RB_GENERATE_INTERNAL(name, type, field, cmp, _sys_tree_h_unused static)
#define RB_GENERATE_INTERNAL(name, type, field, cmp, attr) \
    attr void \
    name##_RB_INSERT_COLOR(struct name *head, struct type *elm) \
    { \
        \
        struct type \
        *parent, *gparent, *tmp; \
        while ((parent = RB_PARENT(elm, field)) != NULL && \
            RB_COLOR(parent, field) == RB_RED) { \
            gparent = RB_PARENT(parent, field); \
            if (parent == RB_LEFT(gparent, field)) { \
                tmp = RB_RIGHT(gparent, field); \
                if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
                    RB_COLOR(tmp, field) = RB_BLACK; \
                    RB_SET_BLACKRED(parent, gparent, field); \
                    elm = gparent; \
                    continue; \
                } \
            } \
            if (RB_RIGHT(parent, field) == elm) { \

```

```

RB_ROTATE_LEFT(head, parent, tmp, field);\
tmp = parent; \
parent = elm; \
elm = tmp; \
} \
RB_SET_BLACKRED(parent, gparent, field);\
RB_ROTATE_RIGHT(head, gparent, tmp, field);\
} else { \
tmp = RB_LEFT(gparent, field); \
if (tmp && RB_COLOR(tmp, field) == RB_RED) { \
RB_COLOR(tmp, field) = RB_BLACK; \
RB_SET_BLACKRED(parent, gparent, field);\
elm = gparent; \
continue; \
} \
if (RB_LEFT(parent, field) == elm)
{ \
RB_ROTATE_RIGHT(head, parent, tmp, field);\
tmp = parent; \
parent = elm; \
elm = tmp; \
} \
RB_SET_BLACKRED(parent, gparent, field);\
RB_ROTATE_LEFT(head, gparent, tmp, field);\
} \
} \
RB_COLOR(head->rbh_root, field) = RB_BLACK; \
} \
\
attr void \
name##_RB_REMOVE_COLOR(struct name *head, struct type *parent, struct type *elm) \
{ \
struct type *tmp; \
while ((elm == NULL || RB_COLOR(elm, field) == RB_BLACK) && \
elm != RB_ROOT(head)) { \
if (RB_LEFT(parent, field) == elm) { \
tmp = RB_RIGHT(parent, field); \
if (RB_COLOR(tmp, field) == RB_RED) { \
RB_SET_BLACKRED(tmp, parent, field);\
RB_ROTATE_LEFT(head, parent, tmp, field);\
tmp = RB_RIGHT(parent, field); \
} \
if ((RB_LEFT(tmp, field) == NULL || \
RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) && \
(RB_RIGHT(tmp, field) == NULL || \
RB_COLOR(RB_RIGHT(tmp, \
field), field) == RB_BLACK)) {\
RB_COLOR(tmp, field) = RB_RED; \

```



```

elm = parent; \
parent = RB_PARENT(elm, field); \
} else { \
if (RB_RIGHT(tmp, field) == NULL || \
    RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK) {\
    struct type *oleft; \
    if ((oleft = RB_LEFT(tmp, field)) \
        != NULL) \
        RB_COLOR(oleft, field) = RB_BLACK;\
    RB_COLOR(tmp, field) = RB_RED; \
    RB_ROTATE_RIGHT(head, tmp, oleft, field);\
    tmp = RB_RIGHT(parent, field); \
} \
RB_COLOR(tmp, field) = RB_COLOR(parent, field);\
RB_COLOR(parent, field) = RB_BLACK; \
if (RB_RIGHT(tmp, field)) \
    RB_COLOR(RB_RIGHT(tmp, field), field) = RB_BLACK;\
RB_ROTATE_LEFT(head, parent, tmp, field);\
elm = RB_ROOT(head); \
break; \
} \
} else { \
tmp = RB_LEFT(parent, field); \
if (RB_COLOR(tmp, field) == RB_RED) { \
    RB_SET_BLACKRED(tmp, parent, field); \
    RB_ROTATE_RIGHT(head, parent, tmp, field);\
    tmp
= RB_LEFT(parent, field); \
} \
if ((RB_LEFT(tmp, field) == NULL || \
    RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) &&\
    (RB_RIGHT(tmp, field) == NULL || \
    RB_COLOR(RB_RIGHT(tmp, field), field) == RB_BLACK)) {\
    RB_COLOR(tmp, field) = RB_RED; \
    elm = parent; \
    parent = RB_PARENT(elm, field); \
} else { \
if (RB_LEFT(tmp, field) == NULL || \
    RB_COLOR(RB_LEFT(tmp, field), field) == RB_BLACK) {\
    struct type *oright; \
    if ((oright = RB_RIGHT(tmp, field)) \
        != NULL) \
        RB_COLOR(oright, field) = RB_BLACK;\
    RB_COLOR(tmp, field) = RB_RED; \
    RB_ROTATE_LEFT(head, tmp, oright, field);\
    tmp = RB_LEFT(parent, field); \
} \
RB_COLOR(tmp, field) = RB_COLOR(parent, field);\

```

```

RB_COLOR(parent, field) = RB_BLACK; \
if (RB_LEFT(tmp, field)) \
    RB_COLOR(RB_LEFT(tmp, field), field) = RB_BLACK;\
RB_ROTATE_RIGHT(head, parent, tmp, field);\
elm = RB_ROOT(head); \
break; \
} \
} \
} \
if
(elm) \
    RB_COLOR(elm, field) = RB_BLACK; \
} \
\
attr struct type * \
name##_RB_REMOVE(struct name *head, struct type *elm) \
{ \
    struct type *child, *parent, *old = elm; \
    int color; \
    if (RB_LEFT(elm, field) == NULL) \
        child = RB_RIGHT(elm, field); \
    else if (RB_RIGHT(elm, field) == NULL) \
        child = RB_LEFT(elm, field); \
    else { \
        struct type *left; \
        elm = RB_RIGHT(elm, field); \
        while ((left = RB_LEFT(elm, field)) != NULL) \
            elm = left; \
        child = RB_RIGHT(elm, field); \
        parent = RB_PARENT(elm, field); \
        color = RB_COLOR(elm, field); \
        if (child) \
            RB_PARENT(child, field) = parent; \
        if (parent) { \
            if (RB_LEFT(parent, field) == elm) \
                RB_LEFT(parent, field) = child; \
            else \
                RB_RIGHT(parent, field) = child; \
            RB_AUGMENT(parent); \
        } else \
            RB_ROOT(head)
    = child; \
    if (RB_PARENT(elm, field) == old) \
        parent = elm; \
    (elm)->field = (old)->field; \
    if (RB_PARENT(old, field)) { \
        if (RB_LEFT(RB_PARENT(old, field), field) == old)\
            RB_LEFT(RB_PARENT(old, field), field) = elm;\

```

```

else \
    RB_RIGHT(RB_PARENT(old, field), field) = elm;\
    RB_AUGMENT(RB_PARENT(old, field)); \
} else \
    RB_ROOT(head) = elm; \
    RB_PARENT(RB_LEFT(old, field), field) = elm; \
    if (RB_RIGHT(old, field)) \
        RB_PARENT(RB_RIGHT(old, field), field) = elm; \
    if (parent) { \
        left = parent; \
        do { \
            RB_AUGMENT(left); \
        } while ((left = RB_PARENT(left, field)) != NULL); \
    } \
    goto color; \
} \
parent = RB_PARENT(elm, field); \
color = RB_COLOR(elm, field); \
if (child) \
    RB_PARENT(child, field) = parent; \
if (parent) { \
    if (RB_LEFT(parent, field) == elm) \
        RB_LEFT(parent, field) = child; \
    else \
        RB_RIGHT(parent,
field) = child; \
    RB_AUGMENT(parent); \
} else \
    RB_ROOT(head) = child; \
color: \
if (color == RB_BLACK) \
    name##_RB_REMOVE_COLOR(head, parent, child); \
return (old); \
} \
\
/* Inserts a node into the RB tree */ \
attr struct type * \
name##_RB_INSERT(struct name *head, struct type *elm) \
{ \
    struct type *tmp; \
    struct type *parent = NULL; \
    int comp = 0; \
    tmp = RB_ROOT(head); \
    while (tmp) { \
        parent = tmp; \
        comp = (cmp)(elm, parent); \
        if (comp < 0) \
            tmp = RB_LEFT(tmp, field); \

```

```

else if (comp > 0) \
    tmp = RB_RIGHT(tmp, field); \
else \
    return (tmp); \
} \
RB_SET(elm, parent, field); \
if (parent != NULL) { \
    if (comp < 0) \
        RB_LEFT(parent, field) = elm; \
    else \
        RB_RIGHT(parent, field) = elm; \
    RB_AUGMENT(parent); \
} else \
    RB_ROOT(head) = elm; \
name##_RB_INSERT_COLOR(head,
elm); \
return (NULL); \
} \
\
/* Finds the node with the same key as elm */ \
attr struct type * \
name##_RB_FIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \
    int comp; \
    while (tmp) { \
        comp = cmp(elm, tmp); \
        if (comp < 0) \
            tmp = RB_LEFT(tmp, field); \
        else if (comp > 0) \
            tmp = RB_RIGHT(tmp, field); \
        else \
            return (tmp); \
    } \
    return (NULL); \
} \
\
/* Finds the first node greater than or equal to the search key */ \
attr struct type * \
name##_RB_NFIND(struct name *head, struct type *elm) \
{ \
    struct type *tmp = RB_ROOT(head); \
    struct type *res = NULL; \
    int comp; \
    while (tmp) { \
        comp = cmp(elm, tmp); \
        if (comp < 0) { \
            res = tmp; \

```

```

    tmp = RB_LEFT(tmp, field); \
} \
else if (comp > 0) \
    tmp =
RB_RIGHT(tmp, field); \
else \
    return (tmp); \
} \
return (res); \
} \
\
/* ARGSUSED */ \
attr struct type * \
name##_RB_NEXT(struct type *elm) \
{ \
    if (RB_RIGHT(elm, field)) { \
        elm = RB_RIGHT(elm, field); \
        while (RB_LEFT(elm, field)) \
            elm = RB_LEFT(elm, field); \
    } else { \
        if (RB_PARENT(elm, field) && \
            (elm == RB_LEFT(RB_PARENT(elm, field), field))) \
            elm = RB_PARENT(elm, field); \
        else { \
            while (RB_PARENT(elm, field) && \
                (elm == RB_RIGHT(RB_PARENT(elm, field), field))) \
                elm = RB_PARENT(elm, field); \
            elm = RB_PARENT(elm, field); \
        } \
    } \
    return (elm); \
} \
\
/* ARGSUSED */ \
attr struct type * \
name##_RB_PREV(struct type *elm) \
{ \
    if (RB_LEFT(elm, field)) { \
        elm = RB_LEFT(elm, field); \
        while (RB_RIGHT(elm, field)) \
            elm = RB_RIGHT(elm, field); \
    } \
    else { \
        if (RB_PARENT(elm, field) && \
            (elm == RB_RIGHT(RB_PARENT(elm, field), field))) \
            elm = RB_PARENT(elm, field); \
        else { \
            while (RB_PARENT(elm, field) && \

```

```

        (elm == RB_LEFT(RB_PARENT(elm, field), field)))\
    elm = RB_PARENT(elm, field); \
    elm = RB_PARENT(elm, field); \
} \
} \
return (elm); \
} \
\
attr struct type * \
name##_RB_MINMAX(struct name *head, int val) \
{ \
    struct type *tmp = RB_ROOT(head); \
    struct type *parent = NULL; \
    while (tmp) { \
        parent = tmp; \
        if (val < 0) \
            tmp = RB_LEFT(tmp, field); \
        else \
            tmp = RB_RIGHT(tmp, field); \
    } \
    return (parent); \
}

#define RB_NEGINF -1
#define RB_INF 1

#define RB_INSERT(name, x, y) name##_RB_INSERT(x, y)
#define RB_REMOVE(name, x, y) name##_RB_REMOVE(x, y)
#define RB_FIND(name, x, y) name##_RB_FIND(x, y)
#define RB_NFIND(name, x, y) name##_RB_NFIND(x, y)
#define RB_NEXT(name, x, y) name##_RB_NEXT(y)
#define RB_PREV(name, x, y) name##_RB_PREV(y)
#define RB_MIN(name, x) name##_RB_MINMAX(x, RB_NEGINF)
#define RB_MAX(name, x) name##_RB_MINMAX(x, RB_INF)

#define RB_FOREACH(x, name, head) \
    for ((x) = RB_MIN(name, head); \
        (x) != NULL; \
        (x) = name##_RB_NEXT(x))

#define RB_FOREACH_FROM(x, name, y) \
    for ((x) = (y); \
        ((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
        (x) = (y))

#define RB_FOREACH_SAFE(x, name, head, y) \
    for ((x) = RB_MIN(name, head); \

```

```

((x) != NULL) && ((y) = name##_RB_NEXT(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_REVERSE(x, name, head) \
for ((x) = RB_MAX(name, head); \
(x) != NULL; \
(x) = name##_RB_PREV(x))

#define RB_FOREACH_REVERSE_FROM(x, name, y) \
for ((x) = (y); \
(x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
(x) = (y))

#define RB_FOREACH_REVERSE_SAFE(x, name, \
head, y) \
for ((x) = RB_MAX(name, head); \
(x) != NULL) && ((y) = name##_RB_PREV(x), (x) != NULL); \
(x) = (y))

#endif /* _SYS_TREE_H_ */
MIT
# MIT License

```

Copyright 2019-2021 Caleb Porzio and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2023 Leaf Corcoran

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl
pkgver=2.3.21
pkgrel=8
pkgdesc="A library for high-performance 2D graphics"
url="https://www.levien.com/libart/"
arch="all"
options="!check" # No test suite.
license="LGPL-2.0-or-later"
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart-lgpl/2.3/libart-lgpl-$pkgver.tar.bz2"
"
```

```
builddir="$srcdir"/libart-lgpl-$pkgver
```

```
prepare() {
    update_config_sub
    default_prepare
}
```

```
build() {
    ./configure \
        --build=$CBUILD \
        --host=$CHOST \
        --prefix=/usr
    make
}
```

```
package() {
    make DESTDIR="$pkgdir" install
}
```


sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"
mini_sendmail - accept email on behalf of real sendmail

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING,
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

1.46 utmps-libs 0.1.3.1-r0

1.46.1 Available under license :

Copyright (c) 2018-2025 Laurent Bercot <ska-skaware@skarnet.org>

Permission to use, copy, modify, and distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.47 iputils-ping 20240905-r0

1.47.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This

General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a

notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such

an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues),

conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For

example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any

later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively

convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into

proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

1.48 go-elasticsearch 7.17.10

1.48.1 Available under license :

```
// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
//
// Code generated from specification version 7.17.10: DO NOT EDIT
```

```
package esapi
```

```
import (
    "context"
    "io"
    "net/http"
    "strconv"
    "strings"
)
```

```
func newLicensePostFunc(t Transport) LicensePost {
    return func(o
...func(*LicensePostRequest)) (*Response, error) {
    var r = LicensePostRequest{}
    for _, f := range o {
        f(&r)
    }
    return r.Do(r.ctx, t)
}
```

```

}

// ----- API Definition -----

// LicensePost - Updates the license for the cluster.
//
// See full documentation at https://www.elastic.co/guide/en/elasticsearch/reference/master/update-license.html.
type LicensePost func(o ...func(*LicensePostRequest)) (*Response, error)

// LicensePostRequest configures the License Post API request.
type LicensePostRequest struct {
    Body io.Reader

    Acknowledge *bool

    Pretty    bool
    Human     bool
    ErrorTrace bool
    FilterPath []string

    Header http.Header

    ctx context.Context
}

// Do executes the request and returns response or error.
func (r LicensePostRequest) Do(ctx context.Context, transport Transport) (*Response, error) {
    var (
        method string
        path    strings.Builder
        params  map[string]string
    )

    method = "PUT"

    path.Grow(len("/_license"))
    path.WriteString("/_license")

    params
    = make(map[string]string)

    if r.Acknowledge != nil {
        params["acknowledge"] = strconv.FormatBool(*r.Acknowledge)
    }

    if r.Pretty {
        params["pretty"] = "true"
    }
}

```

```

if r.Human {
    params["human"] = "true"
}

if r.ErrorTrace {
    params["error_trace"] = "true"
}

if len(r.FilterPath) > 0 {
    params["filter_path"] = strings.Join(r.FilterPath, ",")
}

req, err := newRequest(method, path.String(), r.Body)
if err != nil {
    return nil, err
}

if len(params) > 0 {
    q := req.URL.Query()
    for k, v := range params {
        q.Set(k, v)
    }
    req.URL.RawQuery = q.Encode()
}

if len(r.Header) > 0 {
    if len(req.Header) == 0 {
        req.Header = r.Header
    } else {
        for k, vv := range r.Header {
            for _, v := range vv {
                req.Header.Add(k, v)
            }
        }
    }
}

if r.Body != nil && req.Header.Get(headerContentType) == "" {
    req.Header[headerContentType] = headerContentTypeJSON
}

if ctx != nil {
    req = req.WithContext(ctx)
}

res, err

```

```

:= transport.Perform(req)
if err != nil {
    return nil, err
}

response := Response{
    StatusCode: res.StatusCode,
    Body:      res.Body,
    Header:    res.Header,
}

return &response, nil
}

// WithContext sets the request context.
func (f LicensePost) WithContext(v context.Context) func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        r.ctx = v
    }
}

// WithBody - licenses to be installed.
func (f LicensePost) WithBody(v io.Reader) func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        r.Body = v
    }
}

// WithAcknowledge - whether the user has acknowledged acknowledge messages (default: false).
func (f LicensePost) WithAcknowledge(v bool) func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        r.Acknowledge = &v
    }
}

// WithPretty makes the response body pretty-printed.
func (f LicensePost) WithPretty() func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        r.Pretty = true
    }
}

// WithHuman makes statistical values human-readable.
func
(f LicensePost) WithHuman() func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        r.Human = true
    }
}

```

```

}

// WithErrorTrace includes the stack trace for errors in the response body.
func (f LicensePost) WithErrorTrace() func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        r.ErrorTrace = true
    }
}

// WithFilterPath filters the properties of the response body.
func (f LicensePost) WithFilterPath(v ...string) func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        r.FilterPath = v
    }
}

// WithHeader adds the headers to the HTTP request.
func (f LicensePost) WithHeader(h map[string]string) func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        for k, v := range h {
            r.Header.Add(k, v)
        }
    }
}

// WithOpaqueID adds the X-Opaque-Id header to the HTTP request.
func (f LicensePost) WithOpaqueID(s string) func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        {
            if r.Header == nil {
                r.Header = make(http.Header)
            }
            r.Header.Set("X-Opaque-Id", s)
        }
    }
}

// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//

```

```

// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
//
// Code generated from specification version 7.17.10: DO NOT EDIT

package esapi

import (
    "context"
    "net/http"
    "strings"
)

func newLicenseGetBasicStatusFunc(t Transport) LicenseGetBasicStatus {
    return
    func(o ...func(*LicenseGetBasicStatusRequest)) (*Response, error) {
        var r = LicenseGetBasicStatusRequest{}
        for _, f := range o {
            f(&r)
        }
        return r.Do(r.ctx, t)
    }
}

// ----- API Definition -----

// LicenseGetBasicStatus - Retrieves information about the status of the basic license.
//
// See full documentation at https://www.elastic.co/guide/en/elasticsearch/reference/master/get-basic-status.html.
type LicenseGetBasicStatus func(o ...func(*LicenseGetBasicStatusRequest)) (*Response, error)

// LicenseGetBasicStatusRequest configures the License Get Basic Status API request.
type LicenseGetBasicStatusRequest struct {
    Pretty    bool
    Human     bool
    ErrorTrace bool
    FilterPath []string

    Header http.Header

    ctx context.Context
}

// Do executes the request and returns response or error.

```

```

func (r LicenseGetBasicStatusRequest) Do(ctx context.Context, transport Transport) (*Response, error) {
    var (
        method string
        path strings.Builder
        params
        map[string]string
    )

    method = "GET"

    path.Grow(len("/_license/basic_status"))
    path.WriteString("/_license/basic_status")

    params = make(map[string]string)

    if r.Pretty {
        params["pretty"] = "true"
    }

    if r.Human {
        params["human"] = "true"
    }

    if r.ErrorTrace {
        params["error_trace"] = "true"
    }

    if len(r.FilterPath) > 0 {
        params["filter_path"] = strings.Join(r.FilterPath, ",")
    }

    req, err := newRequest(method, path.String(), nil)
    if err != nil {
        return nil, err
    }

    if len(params) > 0 {
        q := req.URL.Query()
        for k, v := range params {
            q.Set(k, v)
        }
        req.URL.RawQuery = q.Encode()
    }

    if len(r.Header) > 0 {
        if len(req.Header) == 0 {
            req.Header = r.Header
        } else {

```



```

    for k, vv := range r.Header {
        for _, v := range vv {
            req.Header.Add(k, v)
        }
    }
}

if ctx != nil {
    req = req.WithContext(ctx)
}

res, err := transport.Perform(req)
if err != nil {
    return nil, err
}

response := Response{
    StatusCode: res.StatusCode,
    Body:
        res.Body,
    Header:    res.Header,
}

return &response, nil
}

// WithContext sets the request context.
func (f LicenseGetBasicStatus) WithContext(v context.Context) func(*LicenseGetBasicStatusRequest) {
    return func(r *LicenseGetBasicStatusRequest) {
        r.ctx = v
    }
}

// WithPretty makes the response body pretty-printed.
func (f LicenseGetBasicStatus) WithPretty() func(*LicenseGetBasicStatusRequest) {
    return func(r *LicenseGetBasicStatusRequest) {
        r.Pretty = true
    }
}

// WithHuman makes statistical values human-readable.
func (f LicenseGetBasicStatus) WithHuman() func(*LicenseGetBasicStatusRequest) {
    return func(r *LicenseGetBasicStatusRequest) {
        r.Human = true
    }
}

```

```

// WithErrorTrace includes the stack trace for errors in the response body.
func (f LicenseGetBasicStatus) WithErrorTrace() func(*LicenseGetBasicStatusRequest) {
    return func(r *LicenseGetBasicStatusRequest) {
        r.ErrorTrace = true
    }
}

// WithFilterPath filters the properties of the
response body.
func (f LicenseGetBasicStatus) WithFilterPath(v ...string) func(*LicenseGetBasicStatusRequest) {
    return func(r *LicenseGetBasicStatusRequest) {
        r.FilterPath = v
    }
}

// WithHeader adds the headers to the HTTP request.
func (f LicenseGetBasicStatus) WithHeader(h map[string]string) func(*LicenseGetBasicStatusRequest) {
    return func(r *LicenseGetBasicStatusRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        for k, v := range h {
            r.Header.Add(k, v)
        }
    }
}

// WithOpaqueID adds the X-Opaque-Id header to the HTTP request.
func (f LicenseGetBasicStatus) WithOpaqueID(s string) func(*LicenseGetBasicStatusRequest) {
    return func(r *LicenseGetBasicStatusRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        r.Header.Set("X-Opaque-Id", s)
    }
}

```

name: License headers

on: [pull_request]

jobs:

build:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: Check license headers
run: |
./github/check-license-headers.sh
Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute

copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2018 Elasticsearch BV

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
//
// Code generated from specification version 7.17.10: DO NOT EDIT
```

package esapi

```
import (
    "context"
    "net/http"
    "strconv"
    "strings"
)
```

```

func newLicenseGetFunc(t Transport) LicenseGet {
    return func(o ...func(*LicenseGetRequest))
        (*Response, error) {
            var r = LicenseGetRequest{}
            for _, f := range o {
                f(&r)
            }
            return r.Do(r.ctx, t)
        }
    }

// ----- API Definition -----

// LicenseGet - Retrieves licensing information for the cluster
//
// See full documentation at https://www.elastic.co/guide/en/elasticsearch/reference/master/get-license.html.
type LicenseGet func(o ...func(*LicenseGetRequest)) (*Response, error)

// LicenseGetRequest configures the License Get API request.
type LicenseGetRequest struct {
    AcceptEnterprise *bool
    Local           *bool

    Pretty    bool
    Human     bool
    ErrorTrace bool
    FilterPath []string

    Header http.Header

    ctx context.Context
}

// Do executes the request and returns response or error.
func (r LicenseGetRequest) Do(ctx context.Context, transport Transport) (*Response, error) {
    var (
        method string
        path    strings.Builder
        params  map[string]string
    )

    method = "GET"

    path.Grow(len("/_license"))
    path.WriteString("/_license")

    params

```



```

= make(map[string]string)

if r.AcceptEnterprise != nil {
    params["accept_enterprise"] = strconv.FormatBool(*r.AcceptEnterprise)
}

if r.Local != nil {
    params["local"] = strconv.FormatBool(*r.Local)
}

if r.Pretty {
    params["pretty"] = "true"
}

if r.Human {
    params["human"] = "true"
}

if r.ErrorTrace {
    params["error_trace"] = "true"
}

if len(r.FilterPath) > 0 {
    params["filter_path"] = strings.Join(r.FilterPath, ",")
}

req, err := newRequest(method, path.String(), nil)
if err != nil {
    return nil, err
}

if len(params) > 0 {
    q := req.URL.Query()
    for k, v := range params {
        q.Set(k, v)
    }
    req.URL.RawQuery = q.Encode()
}

if len(r.Header) > 0 {
    if len(req.Header) == 0 {
        req.Header = r.Header
    } else {
        for k, vv := range r.Header {
            for _, v := range vv {
                req.Header.Add(k, v)
            }
        }
    }
}

```

```

    }
}

if ctx != nil {
    req = req.WithContext(ctx)
}

res, err := transport.Perform(req)
if
err != nil {
    return nil, err
}

response := Response{
    StatusCode: res.StatusCode,
    Body:      res.Body,
    Header:    res.Header,
}

return &response, nil
}

// WithContext sets the request context.
func (f LicenseGet) WithContext(v context.Context) func(*LicenseGetRequest) {
    return func(r *LicenseGetRequest) {
        r.ctx = v
    }
}

// WithAcceptEnterprise - if the active license is an enterprise license, return type as 'enterprise' (default: false).
func (f LicenseGet) WithAcceptEnterprise(v bool) func(*LicenseGetRequest) {
    return func(r *LicenseGetRequest) {
        r.AcceptEnterprise = &v
    }
}

// WithLocal - return local information, do not retrieve the state from master node (default: false).
func (f LicenseGet) WithLocal(v bool) func(*LicenseGetRequest) {
    return func(r *LicenseGetRequest) {
        r.Local = &v
    }
}

// WithPretty makes the response body pretty-printed.
func (f LicenseGet) WithPretty() func(*LicenseGetRequest) {
    return func(r *LicenseGetRequest) {
        r.Pretty = true
    }
}

```

```

}

//
WithHuman makes statistical values human-readable.
func (f LicenseGet) WithHuman() func(*LicenseGetRequest) {
    return func(r *LicenseGetRequest) {
        r.Human = true
    }
}

// WithErrorTrace includes the stack trace for errors in the response body.
func (f LicenseGet) WithErrorTrace() func(*LicenseGetRequest) {
    return func(r *LicenseGetRequest) {
        r.ErrorTrace = true
    }
}

// WithFilterPath filters the properties of the response body.
func (f LicenseGet) WithFilterPath(v ...string) func(*LicenseGetRequest) {
    return func(r *LicenseGetRequest) {
        r.FilterPath = v
    }
}

// WithHeader adds the headers to the HTTP request.
func (f LicenseGet) WithHeader(h map[string]string) func(*LicenseGetRequest) {
    return func(r *LicenseGetRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        for k, v := range h {
            r.Header.Add(k, v)
        }
    }
}

// WithOpaqueID adds the X-Opaque-Id header to the HTTP request.
func (f LicenseGet) WithOpaqueID(s string) func(*LicenseGetRequest) {
    {
        return func(r *LicenseGetRequest) {
            if r.Header == nil {
                r.Header = make(http.Header)
            }
            r.Header.Set("X-Opaque-Id", s)
        }
    }
}

// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with

```

```
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
//
// Code generated from specification version 7.17.10: DO NOT EDIT
```

```
package esapi
```

```
import (
    "context"
    "net/http"
    "strings"
)
```

```
func newLicenseDeleteFunc(t Transport) LicenseDelete {
    return func(o ...func(*LicenseDeleteRequest))
        (*Response, error) {
            var r = LicenseDeleteRequest{}
            for _, f := range o {
                f(&r)
            }
        }
```

```

    return r.Do(r.ctx, t)
}
}

// ----- API Definition -----

// LicenseDelete - Deletes licensing information for the cluster
//
// See full documentation at https://www.elastic.co/guide/en/elasticsearch/reference/master/delete-license.html.
type LicenseDelete func(o ...func(*LicenseDeleteRequest)) (*Response, error)

// LicenseDeleteRequest configures the License Delete API request.
type LicenseDeleteRequest struct {
    Pretty    bool
    Human     bool
    ErrorTrace bool
    FilterPath []string

    Header http.Header

    ctx context.Context
}

// Do executes the request and returns response or error.
func (r LicenseDeleteRequest) Do(ctx context.Context, transport Transport) (*Response, error) {
    var (
        method string
        path    strings.Builder
        params  map[string]string
    )

    method = "DELETE"

    path.Grow(len("/_license"))
    path.WriteString("/_license")

    params
    = make(map[string]string)

    if r.Pretty {
        params["pretty"] = "true"
    }

    if r.Human {
        params["human"] = "true"
    }

    if r.ErrorTrace {

```

```

    params["error_trace"] = "true"
}

if len(r.FilterPath) > 0 {
    params["filter_path"] = strings.Join(r.FilterPath, ",")
}

req, err := newRequest(method, path.String(), nil)
if err != nil {
    return nil, err
}

if len(params) > 0 {
    q := req.URL.Query()
    for k, v := range params {
        q.Set(k, v)
    }
    req.URL.RawQuery = q.Encode()
}

if len(r.Header) > 0 {
    if len(req.Header) == 0 {
        req.Header = r.Header
    } else {
        for k, vv := range r.Header {
            for _, v := range vv {
                req.Header.Add(k, v)
            }
        }
    }
}

if ctx != nil {
    req = req.WithContext(ctx)
}

res, err := transport.Perform(req)
if err != nil {
    return nil, err
}

response := Response{
    StatusCode: res.StatusCode,
    Body:      res.Body,
    Header:    res.Header,
}

return &response, nil

```

```

}

// WithContext sets the request context.
func
(f LicenseDelete) WithContext(v context.Context) func(*LicenseDeleteRequest) {
return func(r *LicenseDeleteRequest) {
    r.ctx = v
}
}

// WithPretty makes the response body pretty-printed.
func (f LicenseDelete) WithPretty() func(*LicenseDeleteRequest) {
return func(r *LicenseDeleteRequest) {
    r.Pretty = true
}
}

// WithHuman makes statistical values human-readable.
func (f LicenseDelete) WithHuman() func(*LicenseDeleteRequest) {
return func(r *LicenseDeleteRequest) {
    r.Human = true
}
}

// WithErrorTrace includes the stack trace for errors in the response body.
func (f LicenseDelete) WithErrorTrace() func(*LicenseDeleteRequest) {
return func(r *LicenseDeleteRequest) {
    r.ErrorTrace = true
}
}

// WithFilterPath filters the properties of the response body.
func (f LicenseDelete) WithFilterPath(v ...string) func(*LicenseDeleteRequest) {
return func(r *LicenseDeleteRequest) {
    r.FilterPath = v
}
}

// WithHeader adds the headers to the HTTP request.
func (f LicenseDelete)
WithHeader(h map[string]string) func(*LicenseDeleteRequest) {
return func(r *LicenseDeleteRequest) {
    if r.Header == nil {
        r.Header = make(http.Header)
    }
    for k, v := range h {
        r.Header.Add(k, v)
    }
}
}

```

```

}
}

// WithOpaqueID adds the X-Opaque-Id header to the HTTP request.
func (f LicenseDelete) WithOpaqueID(s string) func(*LicenseDeleteRequest) {
return func(r *LicenseDeleteRequest) {
    if r.Header == nil {
        r.Header = make(http.Header)
    }
    r.Header.Set("X-Opaque-Id", s)
}
}

// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
//
// Code generated from specification version 7.17.10: DO NOT EDIT

package esapi

import (
    "context"
    "net/http"
    "strconv"
    "strings"
)

func newLicensePostStartTrialFunc(t Transport) LicensePostStartTrial
{
return func(o ...func(*LicensePostStartTrialRequest)) (*Response, error) {
    var r = LicensePostStartTrialRequest{}
    for _, f := range o {
        f(&r)
    }
    return r.Do(r.ctx, t)
}

```



```

}
}

// ----- API Definition -----

// LicensePostStartTrial - starts a limited time trial license.
//
// See full documentation at https://www.elastic.co/guide/en/elasticsearch/reference/master/start-trial.html.
type LicensePostStartTrial func(o ...func(*LicensePostStartTrialRequest)) (*Response, error)

// LicensePostStartTrialRequest configures the License Post Start Trial API request.
type LicensePostStartTrialRequest struct {
    Acknowledge *bool
    DocumentType string

    Pretty bool
    Human bool
    ErrorTrace bool
    FilterPath []string

    Header http.Header

    ctx context.Context
}

// Do executes the request and returns response or error.
func (r LicensePostStartTrialRequest) Do(ctx context.Context, transport Transport) (*Response, error) {
    var (
        method
        string
        path strings.Builder
        params map[string]string
    )

    method = "POST"

    path.Grow(len("/_license/start_trial"))
    path.WriteString("/_license/start_trial")

    params = make(map[string]string)

    if r.Acknowledge != nil {
        params["acknowledge"] = strconv.FormatBool(*r.Acknowledge)
    }

    if r.DocumentType != "" {
        params["type"] = r.DocumentType
    }

```

```

if r.Pretty {
    params["pretty"] = "true"
}

if r.Human {
    params["human"] = "true"
}

if r.ErrorTrace {
    params["error_trace"] = "true"
}

if len(r.FilterPath) > 0 {
    params["filter_path"] = strings.Join(r.FilterPath, ",")
}

req, err := newRequest(method, path.String(), nil)
if err != nil {
    return nil, err
}

if len(params) > 0 {
    q := req.URL.Query()
    for k, v := range params {
        q.Set(k, v)
    }
    req.URL.RawQuery = q.Encode()
}

if len(r.Header) > 0 {
    if len(req.Header) == 0 {
        req.Header = r.Header
    } else {
        for k, vv := range r.Header {
            for _, v := range vv {
                req.Header.Add(k, v)
            }
        }
    }
}

if
ctx != nil {
    req = req.WithContext(ctx)
}

res, err := transport.Perform(req)

```

```

if err != nil {
    return nil, err
}

response := Response{
    StatusCode: res.StatusCode,
    Body:      res.Body,
    Header:    res.Header,
}

return &response, nil
}

// WithContext sets the request context.
func (f LicensePostStartTrial) WithContext(v context.Context) func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {
        r.ctx = v
    }
}

// WithAcknowledge - whether the user has acknowledged acknowledge messages (default: false).
func (f LicensePostStartTrial) WithAcknowledge(v bool) func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {
        r.Acknowledge = &v
    }
}

// WithDocumentType - the type of trial license to generate (default: "trial").
func (f LicensePostStartTrial) WithDocumentType(v string) func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {
        r.DocumentType = v
    }
}

// WithPretty
    makes the response body pretty-printed.
func (f LicensePostStartTrial) WithPretty() func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {
        r.Pretty = true
    }
}

// WithHuman makes statistical values human-readable.
func (f LicensePostStartTrial) WithHuman() func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {
        r.Human = true
    }
}

```

```

// WithErrorTrace includes the stack trace for errors in the response body.
func (f LicensePostStartTrial) WithErrorTrace() func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {
        r.ErrorTrace = true
    }
}

// WithFilterPath filters the properties of the response body.
func (f LicensePostStartTrial) WithFilterPath(v ...string) func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {
        r.FilterPath = v
    }
}

// WithHeader adds the headers to the HTTP request.
func (f LicensePostStartTrial) WithHeader(h map[string]string) func(*LicensePostStartTrialRequest)
{
    return func(r *LicensePostStartTrialRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        for k, v := range h {
            r.Header.Add(k, v)
        }
    }
}

// WithOpaqueID adds the X-Opaque-Id header to the HTTP request.
func (f LicensePostStartTrial) WithOpaqueID(s string) func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        r.Header.Set("X-Opaque-Id", s)
    }
}

// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,

```

```

// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
//
// Code generated from specification version 7.17.10: DO NOT EDIT

package esapi

import (
    "context"
    "net/http"
    "strings"
)

func newLicenseGetTrialStatusFunc(t Transport) LicenseGetTrialStatus {
    return
    func(o ...func(*LicenseGetTrialStatusRequest)) (*Response, error) {
        var r = LicenseGetTrialStatusRequest{}
        for _, f := range o {
            f(&r)
        }
        return r.Do(r.ctx, t)
    }
}

// ----- API Definition -----

// LicenseGetTrialStatus - Retrieves information about the status of the trial license.
//
// See full documentation at https://www.elastic.co/guide/en/elasticsearch/reference/master/get-trial-status.html.
type LicenseGetTrialStatus func(o ...func(*LicenseGetTrialStatusRequest)) (*Response, error)

// LicenseGetTrialStatusRequest configures the License Get Trial Status API request.
type LicenseGetTrialStatusRequest struct {
    Pretty    bool
    Human     bool
    ErrorTrace bool
    FilterPath []string

    Header http.Header

    ctx context.Context
}

// Do executes the request and returns response or error.
func (r LicenseGetTrialStatusRequest) Do(ctx context.Context, transport Transport) (*Response, error) {

```

```

var (
    method string
    path strings.Builder
    params
    map[string]string
)

method = "GET"

path.Grow(len("/_license/trial_status"))
path.WriteString("/_license/trial_status")

params = make(map[string]string)

if r.Pretty {
    params["pretty"] = "true"
}

if r.Human {
    params["human"] = "true"
}

if r.ErrorTrace {
    params["error_trace"] = "true"
}

if len(r.FilterPath) > 0 {
    params["filter_path"] = strings.Join(r.FilterPath, ",")
}

req, err := newRequest(method, path.String(), nil)
if err != nil {
    return nil, err
}

if len(params) > 0 {
    q := req.URL.Query()
    for k, v := range params {
        q.Set(k, v)
    }
    req.URL.RawQuery = q.Encode()
}

if len(r.Header) > 0 {
    if len(req.Header) == 0 {
        req.Header = r.Header
    } else {
        for k, vv := range r.Header {

```

```

    for _, v := range vv {
        req.Header.Add(k, v)
    }
}
}

if ctx != nil {
    req = req.WithContext(ctx)
}

res, err := transport.Perform(req)
if err != nil {
    return nil, err
}

response := Response{
    StatusCode: res.StatusCode,
    Body:
        res.Body,
    Header:    res.Header,
}

return &response, nil
}

// WithContext sets the request context.
func (f LicenseGetTrialStatus) WithContext(v context.Context) func(*LicenseGetTrialStatusRequest) {
    return func(r *LicenseGetTrialStatusRequest) {
        r.ctx = v
    }
}

// WithPretty makes the response body pretty-printed.
func (f LicenseGetTrialStatus) WithPretty() func(*LicenseGetTrialStatusRequest) {
    return func(r *LicenseGetTrialStatusRequest) {
        r.Pretty = true
    }
}

// WithHuman makes statistical values human-readable.
func (f LicenseGetTrialStatus) WithHuman() func(*LicenseGetTrialStatusRequest) {
    return func(r *LicenseGetTrialStatusRequest) {
        r.Human = true
    }
}

// WithErrorTrace includes the stack trace for errors in the response body.

```

```

func (f LicenseGetTrialStatus) WithErrorTrace() func(*LicenseGetTrialStatusRequest) {
    return func(r *LicenseGetTrialStatusRequest) {
        r.ErrorTrace = true
    }
}

// WithFilterPath filters the properties of the
// response body.
func (f LicenseGetTrialStatus) WithFilterPath(v ...string) func(*LicenseGetTrialStatusRequest) {
    return func(r *LicenseGetTrialStatusRequest) {
        r.FilterPath = v
    }
}

// WithHeader adds the headers to the HTTP request.
func (f LicenseGetTrialStatus) WithHeader(h map[string]string) func(*LicenseGetTrialStatusRequest) {
    return func(r *LicenseGetTrialStatusRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        for k, v := range h {
            r.Header.Add(k, v)
        }
    }
}

// WithOpaqueID adds the X-Opaque-Id header to the HTTP request.
func (f LicenseGetTrialStatus) WithOpaqueID(s string) func(*LicenseGetTrialStatusRequest) {
    return func(r *LicenseGetTrialStatusRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        r.Header.Set("X-Opaque-Id", s)
    }
}

// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY

```



```

// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
//
// Code generated from specification version 7.17.10: DO NOT EDIT

package esapi

import (
    "context"
    "net/http"
    "strconv"
    "strings"
)

func newLicensePostStartBasicFunc(t Transport) LicensePostStartBasic
{
    return func(o ...func(*LicensePostStartBasicRequest)) (*Response, error) {
        var r = LicensePostStartBasicRequest{}
        for _, f := range o {
            f(&r)
        }
        return r.Do(r.ctx, t)
    }
}

// ----- API Definition -----

// LicensePostStartBasic - Starts an indefinite basic license.
//
// See full documentation at https://www.elastic.co/guide/en/elasticsearch/reference/master/start-basic.html.
type LicensePostStartBasic func(o ...func(*LicensePostStartBasicRequest)) (*Response, error)

// LicensePostStartBasicRequest configures the License Post Start Basic API request.
type LicensePostStartBasicRequest struct {
    Acknowledge *bool

    Pretty    bool
    Human     bool
    ErrorTrace bool
    FilterPath []string

    Header http.Header

    ctx context.Context
}

// Do executes the request and returns response or error.

```

```

func (r LicensePostStartBasicRequest) Do(ctx context.Context, transport Transport) (*Response, error) {
    var (
        method string
        path strings.Builder
        params
        map[string]string
    )

    method = "POST"

    path.Grow(len("/_license/start_basic"))
    path.WriteString("/_license/start_basic")

    params = make(map[string]string)

    if r.Acknowledge != nil {
        params["acknowledge"] = strconv.FormatBool(*r.Acknowledge)
    }

    if r.Pretty {
        params["pretty"] = "true"
    }

    if r.Human {
        params["human"] = "true"
    }

    if r.ErrorTrace {
        params["error_trace"] = "true"
    }

    if len(r.FilterPath) > 0 {
        params["filter_path"] = strings.Join(r.FilterPath, ",")
    }

    req, err := newRequest(method, path.String(), nil)
    if err != nil {
        return nil, err
    }

    if len(params) > 0 {
        q := req.URL.Query()
        for k, v := range params {
            q.Set(k, v)
        }
        req.URL.RawQuery = q.Encode()
    }

```

```

if len(r.Header) > 0 {
    if len(req.Header) == 0 {
        req.Header = r.Header
    } else {
        for k, vv := range r.Header {
            for _, v := range vv {
                req.Header.Add(k, v)
            }
        }
    }
}

if ctx != nil {
    req = req.WithContext(ctx)
}

res, err := transport.Perform(req)
if
err != nil {
    return nil, err
}

response := Response{
    StatusCode: res.StatusCode,
    Body:      res.Body,
    Header:    res.Header,
}

return &response, nil
}

// WithContext sets the request context.
func (f LicensePostStartBasic) WithContext(v context.Context) func(*LicensePostStartBasicRequest) {
    return func(r *LicensePostStartBasicRequest) {
        r.ctx = v
    }
}

// WithAcknowledge - whether the user has acknowledged acknowledge messages (default: false).
func (f LicensePostStartBasic) WithAcknowledge(v bool) func(*LicensePostStartBasicRequest) {
    return func(r *LicensePostStartBasicRequest) {
        r.Acknowledge = &v
    }
}

// WithPretty makes the response body pretty-printed.
func (f LicensePostStartBasic) WithPretty() func(*LicensePostStartBasicRequest) {
    return func(r *LicensePostStartBasicRequest) {

```

```

    r.Pretty = true
}
}

// WithHuman makes statistical values human-readable.
func (f LicensePostStartBasic) WithHuman() func(*LicensePostStartBasicRequest) {
    return
    func(r *LicensePostStartBasicRequest) {
        r.Human = true
    }
}

// WithErrorTrace includes the stack trace for errors in the response body.
func (f LicensePostStartBasic) WithErrorTrace() func(*LicensePostStartBasicRequest) {
    return func(r *LicensePostStartBasicRequest) {
        r.ErrorTrace = true
    }
}

// WithFilterPath filters the properties of the response body.
func (f LicensePostStartBasic) WithFilterPath(v ...string) func(*LicensePostStartBasicRequest) {
    return func(r *LicensePostStartBasicRequest) {
        r.FilterPath = v
    }
}

// WithHeader adds the headers to the HTTP request.
func (f LicensePostStartBasic) WithHeader(h map[string]string) func(*LicensePostStartBasicRequest) {
    return func(r *LicensePostStartBasicRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        for k, v := range h {
            r.Header.Add(k, v)
        }
    }
}

// WithOpaqueID adds the X-Opaque-Id header to the HTTP request.
func (f LicensePostStartBasic) WithOpaqueID(s string) func(*LicensePostStartBasicRequest) {
    {
        return func(r *LicensePostStartBasicRequest) {
            if r.Header == nil {
                r.Header = make(http.Header)
            }
            r.Header.Set("X-Opaque-Id", s)
        }
    }
}

```

1.49 libcrypto3 3.5.4-r0

1.49.1 Available under license :

This software is copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

Terms of the Perl programming language system itself

- a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
- b) the "Artistic License"

--- The GNU General Public License, Version 1, February 1989 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The GNU General Public License, Version 1, February 1989

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it,

that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the

Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of

transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use

in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that

accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so,
attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (a program to direct compilers to make passes at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

That's all there is to it!

--- The Artistic License 1.0 ---

This software is Copyright (c) 2013 by Mark Jason Dominus <mjd@cpan.org>.

This is free software, licensed under:

The Artistic License 1.0

The Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
- "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
- "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
- "You" is you, if you're thinking about copying or distributing this Package.
- "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
- "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

- a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
- b) use the modified Package only within your corporation or organization.
- c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
- d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

- a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.
- b) accompany the distribution with the machine-readable source of the Package with your modifications.
- c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.
- d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output

from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Apache License
Version 2.0, January 2004
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.50 alpine-baselayout-data 3.7.0-r0

1.50.1 Available under license :

```
# Automatically generated by apkbuild-cpan, template 1
# Contributor: Francesco Colista <francesco.colista@gmail.com>
# Maintainer: Francesco Colista <francesco.colista@gmail.com>
pkgname=perl-software-license
_pkgreal=Software-License
pkgver=0.103004
pkgrel=0
pkgdesc="packages that provide templated software licenses"
url="http://search.cpan.org/dist/Software-License/"
arch="noarch"
license="GPL PerlArtistic"
cpandepends="perl-text-template perl-data-section perl-sub-install"
cpanmakedepends=" "
depends="$cpandepends"
makedepends="perl-dev $cpanmakedepends"
subpackages="$pkgname-doc"
source="http://search.cpan.org/CPAN/authors/id/R/RJ/RJBS/${_pkgreal}-${pkgver}.tar.gz"

_buildddir="$srcdir/${_pkgreal}-${pkgver}"

prepare() {
    cd "$_buildddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}`
    PERL_MM_USE_DEFAULT=1 perl Makefile.PL INSTALLDIRS=vendor
}

build() {
    cd "$_buildddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}`
    make && make test
}

package() {
    cd "$_buildddir"
```

```

make
DESTDIR="$pkgdir" install || return 1
find "$pkgdir" \( -name perllocal.pod -o -name .packlist \) -delete
}

md5sums="0c5fd53d93818566234578b27d667b35 Software-License-0.103004.tar.gz"
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl
pkgver=2.3.21
pkgrel=5
pkgdesc="A library for high-performance 2D graphics"
url="http://www.levien.com/libart/"
arch="all"
license="LGPL"
depends=
makedepends=
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart_lgpl/2.3/libart_lgpl-$pkgver.tar.bz2"
"

```

```

_buildddir="$srcdir"/libart_lgpl-$pkgver

```

```

prepare() {
    cd "$_buildddir"
    update_config_sub || return 1
}

build() {
    cd "$_buildddir"
    ./configure \
        --build=$CBUILD \
        --host=$CHOST \
        --prefix=/usr \
        || return 1
    make || return 1
}

```

```

package() {
    cd "$_buildddir"
    make DESTDIR="$pkgdir" install || return 1
}

```

```

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e7
6654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"

```

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.

Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this Agreement. If you do not wish to so agree,

do not install or use the Software.

1. COPYRIGHT NOTICE

Some or all of this work - Copyright 1999-2011, Intel Corp. All rights reserved.

2. LICENSE

2.1. This is your license from Intel Corp. under its intellectual property rights. You may have additional license terms from the party that provided you this software, covering your right to use that party's intellectual property rights.

2.2. Intel grants, free of charge, to any person ("Licensee") obtaining a copy of the source code appearing in this file ("Covered Code") an irrevocable, perpetual, worldwide license under Intel's copyrights in the base code distributed originally by Intel ("Original Intel Code") to copy, make derivatives, distribute, use and display any portion of the Covered Code in any form, with the right to sublicense such rights; and

2.3. Intel grants Licensee a non-exclusive and non-transferable patent license (with the right to sublicense), under only those claims of Intel patents that are infringed by the Original Intel Code, to make, use, sell, offer to sell, and import the Covered Code and derivative works thereof solely to the minimum extent necessary to exercise the above copyright license, and in no event shall the patent license extend to any additions to or modifications of the Original Intel Code. No other license or right is granted directly or by implication, estoppel or otherwise; The above copyright and patent license is granted only if the following conditions are met:

3. CONDITIONS

3.1. Redistribution of Source with Rights to Further Distribute Source. Redistribution of source code of any substantial portion of the Covered Code or modification with rights to further distribute source must include the above Copyright Notice, the above License, this list of Conditions, and the following Disclaimer and Export Compliance provision. In addition, Licensee must cause all Covered Code to which Licensee contributes to contain a file documenting the changes Licensee made to create that Covered Code and the date of any change. Licensee must include in that file the documentation of any changes made by any predecessor Licensee. Licensee must include a prominent statement that the modification is derived, directly or indirectly, from Original Intel Code.

3.2. Redistribution of Source with no Rights to Further Distribute Source. Redistribution of source code of any substantial portion of the Covered Code or modification without rights to further distribute source must include the following Disclaimer and Export Compliance provision in the documentation and/or other materials provided with distribution. In addition,

Licensee may not authorize further sublicense of source of any portion of the Covered Code, and must include terms to the effect that the license from Licensee to its licensee is limited to the intellectual property embodied in the software Licensee provides to its licensee, and not to intellectual property embodied in modifications its licensee may make.

3.3. Redistribution of Executable. Redistribution in executable form of any substantial portion of the Covered Code or modification must reproduce the above Copyright Notice, and the following Disclaimer and Export Compliance provision in the documentation and/or other materials provided with the distribution.

3.4. Intel retains all right, title, and interest in and to the Original Intel Code.

3.5. Neither the name Intel nor any other trademark owned or controlled by Intel shall be used in advertising or otherwise to promote the sale, use or other dealings in products derived from or relating to the Covered Code without prior written authorization from Intel.

4. DISCLAIMER AND EXPORT COMPLIANCE

4.1. INTEL MAKES NO WARRANTY OF ANY KIND REGARDING ANY SOFTWARE PROVIDED HERE. ANY SOFTWARE ORIGINATING FROM INTEL OR DERIVED FROM INTEL SOFTWARE IS PROVIDED "AS IS," AND INTEL WILL NOT PROVIDE ANY SUPPORT, ASSISTANCE, INSTALLATION, TRAINING OR OTHER SERVICES. INTEL WILL NOT PROVIDE ANY UPDATES, ENHANCEMENTS OR EXTENSIONS. INTEL SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.

4.2. IN NO EVENT SHALL INTEL HAVE ANY LIABILITY TO LICENSEE, ITS LICENSEES OR ANY OTHER THIRD PARTY, FOR ANY LOST PROFITS, LOST DATA, LOSS OF USE OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, UNDER ANY CAUSE OF ACTION OR THEORY OF LIABILITY, AND IRRESPECTIVE OF WHETHER INTEL HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES. THESE LIMITATIONS

SHALL APPLY NOTWITHSTANDING THE
FAILURE OF THE ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.

4.3. Licensee shall not export, either directly or indirectly, any of this software or system incorporating such software without first obtaining any required license or other approval from the U. S. Department of Commerce or any other agency or department of the United States Government. In the event Licensee exports any such software from the United States or re-exports any such software from a foreign destination, Licensee shall ensure that the distribution and export/re-export of the software is in compliance with all laws, regulations, orders, or other restrictions of the U.S. Export Administration Regulations. Licensee agrees that neither it nor any of its subsidiaries will export/re-export any technical data, process, software, or service, directly or indirectly, to any country for which the United States government or any agency thereof requires an export license, other governmental approval, or letter of assurance, without first obtaining such license, approval or letter.

INTEL SOFTWARE LICENSE AGREEMENT

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.

Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this Agreement. If you do not wish to so agree, do not install or use the Software.

LICENSES: Please Note:

- If you are a network administrator, the "Site License" below shall apply to you.
- If you are an end user, the "Single User License" shall apply to you.
- If you are an original equipment manufacturer (OEM), the "OEM License" shall apply to you.

SITE LICENSE. You may copy the Software onto your organization's computers for your organization's use, and you may make a reasonable number of back-up copies of the Software, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may not sublicense or permit simultaneous use of the Software by more than one user.

5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

SINGLE USER LICENSE. You may copy the Software onto a single computer for your personal, noncommercial use, and you may make one back-up copy of the Software, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may not sublicense or permit simultaneous use of the Software by more than one user.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

OEM LICENSE: You may reproduce and distribute the Software only as an integral part of or incorporated in Your product or as a standalone Software maintenance update for existing end users of Your products, excluding any other standalone products, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may only distribute the Software to your customers pursuant to a written license agreement. Such license agreement may be a "break-the-seal" license agreement. At a minimum such license shall safeguard Intel's ownership rights to the Software.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

NO OTHER RIGHTS. No rights or licenses are granted by Intel to You, expressly or by implication, with respect to any proprietary information or patent, copyright, mask work, trademark, trade secret, or other intellectual property right owned or controlled by Intel, except as expressly provided in this Agreement.

OWNERSHIP OF SOFTWARE AND COPYRIGHTS. Title to all copies of the Software remains with Intel or its suppliers. The Software is copyrighted and

protected

by the laws of the United States and other countries, and international treaty provisions. You may not remove any copyright notices from the Software. Intel may make changes to the Software, or to items referenced therein, at any time without notice, but is not obligated to support or update the Software. Except as otherwise expressly provided, Intel grants no express or implied right under Intel patents, copyrights, trademarks, or other intellectual property rights. You may transfer the Software only if the recipient agrees to be fully bound by these terms and if you retain no copies of the Software.

LIMITED MEDIA WARRANTY. If the Software has been delivered by Intel on physical media, Intel warrants the media to be free from material physical defects for a period of ninety days after delivery by Intel. If such a defect is found, return the media to Intel for replacement or alternate delivery of the Software as Intel may select.

EXCLUSION OF OTHER WARRANTIES. EXCEPT AS PROVIDED ABOVE, THE SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. Intel does not warrant or assume responsibility for the accuracy or completeness of any information, text, graphics, links or other items contained within the Software.

LIMITATION OF LIABILITY. IN NO EVENT SHALL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, OR LOST INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS PROHIBIT EXCLUSION OR LIMITATION OF LIABILITY FOR IMPLIED WARRANTIES OR CONSEQUENTIAL OR INCIDENTAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE OTHER LEGAL RIGHTS THAT VARY FROM JURISDICTION TO JURISDICTION.

TERMINATION OF THIS AGREEMENT. Intel may terminate this Agreement at any time if you violate its terms. Upon termination, you will immediately destroy the Software or return all copies of the Software to Intel.

APPLICABLE LAWS. Claims arising under this Agreement shall be governed by the laws of California, excluding its principles of conflict of laws and the United Nations Convention on Contracts for the Sale of Goods. You may not export the Software in violation of applicable export laws and regulations. Intel is not obligated under any other agreements unless they are in writing and signed by an authorized representative of Intel.

GOVERNMENT RESTRICTED RIGHTS. The Software is provided with "RESTRICTED RIGHTS." Use, duplication, or disclosure by the Government is subject to

restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor. Use of the Software by the Government constitutes acknowledgment of Intel's proprietary rights therein. Contractor or Manufacturer is Intel
2200 Mission College Blvd., Santa Clara, CA
95052.

/*

* Copyright (c) 1983, 1988 Regents of the University of California.

* All rights reserved.

*

* Redistribution and use in source and binary forms are permitted

* provided that the above copyright notice and this paragraph are

* duplicated in all such forms and that any documentation,

* advertising materials, and other materials related to such

* distribution and use acknowledge that the software was developed

* by the University of California, Berkeley. The name of the

* University may not be used to endorse or promote products derived

* from this software without specific prior written permission.

* THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR

* IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED

* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

*/

Automatically generated by apkgbuild-cpan, template 1

Contributor: Valery Kartel <valery.kartel@gmail.com>

Maintainer: Valery Kartel <valery.kartel@gmail.com>

pkgname=perl-bsd-resource

_pkgreal=BSD-Resource

pkgver=1.2909

pkgrel=2

pkgdesc="Perl extension implements the BSD process resource limit functions"

url="http://search.cpan.org/dist/BSD-Resource/"

arch="all"

license="GPL PerlArtistic"

cpandepends=""

cpanmakedepends=""

depends="\$cpandepends"

makedepends="perl-dev \$cpanmakedepends"

subpackages="\$pkgname-doc"

source="http://search.cpan.org/CPAN/authors/id/J/JH/JHI/_pkgreal-\$pkgver.tar.gz"

_builddir="\$srcdir/_pkgreal-\$pkgver"

prepare() {

cd "\$_builddir"

export CFLAGS=`perl -MConfig -E 'say \$Config{ccflags}`

PERL_MM_USE_DEFAULT=1 perl Makefile.PL INSTALLDIRS=vendor

}

build() {


```

cd "$_builddir"
export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}'`
make && make test || return 1
}

package() {
cd "$_builddir"
make DESTDIR="$pkgdir" install || return 1
find "$pkgdir"
\ ( -name perllocal.pod -o -name .packlist ) -delete
}

md5sums="3c6dd2fa953088aa87e263f40818d012 BSD-Resource-1.2909.tar.gz"
sha256sums="c47d460e2bd00d8d049f6ff4a0f1f61957e289e8de45af5fc79e851e2e855e06 BSD-Resource-
1.2909.tar.gz"
sha512sums="5f489eaac5089e82081b7cf51d62638750e443751b49415407354da96b17771819b5e6e85529faa41ddd
85743363750157943c7f7700276cce2bd40a429fa430 BSD-Resource-1.2909.tar.gz"
SQLite Copyright
SQLite is in the
Public Domain

```

All of the deliverable code in SQLite has been dedicated to the public domain by the authors. All code authors, and representatives of the companies they work for, have signed affidavits dedicating their contributions to the public domain and originals of those signed affidavits are stored in a firesafe at the main offices of Hwaci. Anyone is free to copy, modify, publish, use, compile, sell, or distribute the original SQLite code, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

The previous paragraph applies to the deliverable code in SQLite - those parts of the SQLite library that you actually bundle and ship with a larger application. Portions of the documentation and some code used as part of the build process might fall under other licenses. The details here are unclear. We do not worry about the licensing of the documentation and build code so much because none of these things are part of the core deliverable SQLite library.

All of the deliverable code in SQLite has been written from scratch. No code has been taken from other projects or from the open internet. Every line of code can be traced back to its original author, and all of those authors have public domain dedications on file. So the SQLite code base is clean and is uncontaminated with licensed code from other projects.

Obtaining An Explicit License To Use SQLite

Even though SQLite is in the public domain and does not require a license, some users want to obtain a license anyway. Some reasons for obtaining a license include:

You are using SQLite in a jurisdiction that does not recognize the public domain.

You are using SQLite in a jurisdiction that does not recognize the right of an author to dedicate their work to the public domain.

You want to hold a tangible legal document as evidence that you have the legal right to use and distribute SQLite.

Your legal department tells

you that you have to purchase a license.

If you feel like you really have to purchase a license for SQLite, Hwaci, the company that employs the architect and principal developers of SQLite, will sell you one.

Contributed Code

In order to keep SQLite completely free and unencumbered by copyright, all new contributors to the SQLite code base are asked to dedicate their contributions to the public domain. If you want to send a patch or enhancement for possible inclusion in the SQLite source tree, please accompany the patch with the following statement:

The author or authors of this code dedicate any and all copyright interest in this code to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this code under copyright law.

We are not able to accept patches or changes to SQLite that are not accompanied by a statement such as the above. In addition, if you make changes or enhancements as an employee, then a simple statement such as the above is insufficient. You must also send by surface mail a copyright release signed by a company officer. A signed original of the copyright release should be mailed to:

Hwaci
6200 Maple Cove Lane
Charlotte, NC 28269
USA

A template copyright release is available in PDF or HTML. You can use this release to make future changes.

see <http://www.sqlite.org/copyright.html>

Copyright (c) 2016 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software source and associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, compile, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject the following terms and conditions:

All modifications to the Materials used to create a binary that is distributed to third parties shall be provided to Khronos with an unrestricted license to use for the purposes of implementing bug fixes and enhancements to the Materials;

If the binary is used as part of an OpenCL(TM) implementation, whether binary is distributed together with or separately to that implementation, then recipient must become an OpenCL Adopter and follow the published OpenCL conformance process for that implementation, details at:

<http://www.khronos.org/conformance/>;

The above copyright notice, the OpenCL trademark license, and this permission notice shall be included in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

OpenCL is a trademark of Apple Inc. used under license by Khronos.

Contributor: Fabian Affolter <fabian@affolter-engineering.ch>

Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>

pkgname=py-flake8-copyright

_pkgname=flake8-copyright

pkgver=0.2.0

pkgrel=1

pkgdesc="Extension for flake8 which checks for copyrights"

url="https://github.com/savoirfairelinux/flake8-copyright"

arch="noarch"

license="MIT"

depends="flake8"

makedepends="python3-dev"

source="https://files.pythonhosted.org/packages/source/\${_pkgname:0:1}/\${_pkgname}/\${_pkgname}-\${pkgver}.tar.gz"

builddir="\$srcdir"/_\${_pkgname}-\${pkgver}

```
build() {  
    cd "$builddir"  
    python3 setup.py build || return 1  
}
```

```
package() {  
    cd "$builddir"  
    python3 setup.py install --prefix=/usr --root="$pkgdir" || return 1  
}
```

md5sums="c6348990c6d74933bf5d5f6a55a14bdf flake8-copyright-0.2.0.tar.gz"

sha256sums="aeef26eb4d5223c9cd5b101e68175fcef6d2b353bf36da688fdde62fccfe2b73 flake8-copyright-0.2.0.tar.gz"

sha512sums="bab1bfaeed3a525ac3c7da12b1f2ec338c204cc69b72d05cd39560e528faff5d586c3b1be9570f4a977a7e2417b586e543bfba12abce256b2023c8c4993eeb90

flake8-copyright-0.2.0.tar.gz"

mini_sendmail - accept email on behalf of real sendmail

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.51 iputils-tracepath 20240905-r0

1.51.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and

(2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under

the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such

parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to

decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing
version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE,
THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,

REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this
is what you want to do, use the GNU Lesser General
Public License instead of this License.

1.52 iputils 20240905-r0

1.52.1 Available under license :

arping: GPL-2.0-or-later
clockdiff: BSD-3-Clause
ping: BSD-3-Clause
tracepath: GPL-2.0-or-later

Files containing license texts are available in Documentation directory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.

Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether

gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's

source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to

exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the

integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES

PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR

THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU

General Public License as published by

the Free Software Foundation; either version 2 of the License, or

(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute
it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

1.53 alpine-baselayout 3.7.0-r0

1.53.1 Available under license :

```
# Automatically generated by apkbuild-cpan, template 1
# Contributor: Francesco Colista <francesco.colista@gmail.com>
# Maintainer: Francesco Colista <francesco.colista@gmail.com>
pkgname=perl-software-license
_pkgreal=Software-License
pkgver=0.103004
pkgrel=0
pkgdesc="packages that provide templated software licenses"
url="http://search.cpan.org/dist/Software-License/"
arch="noarch"
```

```
license="GPL PerlArtistic"
cpandepends="perl-text-template perl-data-section perl-sub-install"
cpanmakedepends=" "
depends="$cpandepends"
makedepends="perl-dev $cpanmakedepends"
subpackages="$pkgname-doc"
source="http://search.cpan.org/CPAN/authors/id/R/RJ/RJBS/${_pkgreal}-${pkgver}.tar.gz"
```

```
_builddir="$srcdir/${_pkgreal}-${pkgver}"
```

```
prepare() {
    cd "$_builddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}`
    PERL_MM_USE_DEFAULT=1 perl Makefile.PL INSTALLDIRS=vendor
}
```

```
build() {
    cd "$_builddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}`
    make && make test
}
```

```
package() {
    cd "$_builddir"
    make
    DESTDIR="$pkgdir" install || return 1
    find "$pkgdir" \( -name perllocal.pod -o -name .packlist \) -delete
}
```

```
md5sums="0c5fd53d93818566234578b27d667b35 Software-License-0.103004.tar.gz"
# Maintainer: Natanael Copa <ncopa@alpinelinux.org>
pkgname=libart-lgpl
pkgver=2.3.21
pkgrel=5
pkgdesc="A library for high-performance 2D graphics"
url="http://www.levien.com/libart/"
arch="all"
license="LGPL"
depends=
makedepends=
subpackages="$pkgname-dev"
source="https://download.gnome.org/sources/libart-lgpl/2.3/libart-lgpl-${pkgver}.tar.bz2"
"
```

```
_builddir="$srcdir"/libart-lgpl-${pkgver}
```

```
prepare() {
    cd "$_builddir"
```

```

update_config_sub || return 1
}
build() {
cd "$_builddir"
./configure \
--build=$CBUILD \
--host=$CHOST \
--prefix=/usr \
|| return 1
make || return 1
}

```

```

package() {
cd "$_builddir"
make DESTDIR="$pkgdir" install || return 1
}

```

sha512sums="8a632a6a4da59e5e8c02ec2f5a57e36d182b325b46513765425e5f171ff9ae326af1b133725beba28f7e76654309e001aee9bace727b5b4c8589405256a3c020 libart_lgpl-2.3.21.tar.bz2"

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.

Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this Agreement. If you do not wish to so agree, do not install or use the Software.

1. COPYRIGHT NOTICE

Some or all of this work - Copyright 1999-2011, Intel Corp. All rights reserved.

2. LICENSE

2.1. This is your license from Intel Corp. under its intellectual property rights. You may have additional license terms from the party that provided you this software, covering your right to use that party's intellectual property rights.

2.2. Intel grants, free of charge, to any person ("Licensee") obtaining a copy of the source code appearing in this file ("Covered Code") an irrevocable, perpetual, worldwide license under Intel's copyrights in the base code distributed originally by Intel ("Original Intel Code") to copy, make derivatives, distribute, use and display any portion of the Covered Code in any form, with the right to sublicense such rights; and

2.3. Intel grants Licensee a non-exclusive and non-transferable patent license (with the right to sublicense), under only those claims of

Intel patents that are infringed by the Original Intel Code, to make, use, sell, offer to sell, and import the Covered Code and derivative works thereof solely to the minimum extent necessary to exercise the above copyright license, and in no event shall the patent license extend to any additions to or modifications of the Original Intel Code. No other license or right is granted directly or by implication, estoppel or otherwise; The above copyright and patent license is granted only if the following conditions are met:

3. CONDITIONS

3.1. Redistribution of Source with Rights to Further Distribute

Source. Redistribution of source code of any substantial portion of the Covered Code or modification with rights to further distribute source must include the above Copyright Notice, the above License, this list of Conditions, and the following Disclaimer and Export Compliance provision. In addition, Licensee must cause all Covered Code to which Licensee contributes to contain a file documenting the changes Licensee made to create that Covered Code and the date of any change. Licensee must include in that file the documentation of any changes made by any predecessor Licensee. Licensee must include a prominent statement that the modification is derived, directly or indirectly, from Original Intel Code.

3.2. Redistribution of Source with no Rights to Further Distribute

Source. Redistribution of source code of any substantial portion of the Covered Code or modification without rights to further distribute source must include the following Disclaimer and Export Compliance provision in the documentation and/or other materials provided with distribution. In addition,

Licensee may not authorize further sublicense of source of any portion of the Covered Code, and must include terms to the effect that the license from Licensee to its licensee is limited to the intellectual property embodied in the software Licensee provides to its licensee, and not to intellectual property embodied in modifications its licensee may make.

3.3. Redistribution of Executable. Redistribution in executable form of any substantial portion of the Covered Code or modification must reproduce the above Copyright Notice, and the following Disclaimer and Export Compliance provision in the documentation and/or other materials provided with the distribution.

3.4. Intel retains all right, title, and interest in and to the Original Intel Code.

3.5. Neither the name Intel nor any other trademark owned or

controlled by Intel shall be used in advertising or otherwise to promote the sale, use or other dealings in products derived from or relating to the Covered Code without prior written authorization from Intel.

4. DISCLAIMER AND EXPORT COMPLIANCE

4.1. INTEL MAKES NO WARRANTY OF ANY KIND REGARDING ANY SOFTWARE PROVIDED HERE. ANY SOFTWARE ORIGINATING FROM INTEL OR DERIVED FROM INTEL SOFTWARE IS PROVIDED "AS IS," AND INTEL WILL NOT PROVIDE ANY SUPPORT, ASSISTANCE, INSTALLATION, TRAINING OR OTHER SERVICES. INTEL WILL NOT PROVIDE ANY UPDATES, ENHANCEMENTS OR EXTENSIONS. INTEL SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.

4.2. IN NO EVENT SHALL INTEL HAVE ANY LIABILITY TO LICENSEE, ITS LICENSEES OR ANY OTHER THIRD PARTY, FOR ANY LOST PROFITS, LOST DATA, LOSS OF USE OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, UNDER ANY CAUSE OF ACTION OR THEORY OF LIABILITY, AND IRRESPECTIVE OF WHETHER INTEL HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES. THESE LIMITATIONS SHALL APPLY NOTWITHSTANDING THE FAILURE OF THE ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.

4.3. Licensee shall not export, either directly or indirectly, any of this software or system incorporating such software without first obtaining any required license or other approval from the U. S. Department of Commerce or any other agency or department of the United States Government. In the event Licensee exports any such software from the United States or re-exports any such software from a foreign destination, Licensee shall ensure that the distribution and export/re-export of the software is in compliance with all laws, regulations, orders, or other restrictions of the U.S. Export Administration Regulations. Licensee agrees that neither it nor any of its subsidiaries will export/re-export any technical data, process, software, or service, directly or indirectly, to any country for which the United States government or any agency thereof requires an export license, other governmental approval, or letter of assurance, without first obtaining such license, approval or letter.

INTEL SOFTWARE LICENSE AGREEMENT

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.

Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this

Agreement. If you do not wish to so agree, do not install or use the Software.

LICENSES: Please Note:

- If you are a network administrator, the "Site License" below shall apply to you.
- If you are an end user, the "Single User License" shall apply to you.
- If you are an original equipment manufacturer (OEM), the "OEM License" shall apply to you.

SITE LICENSE. You may copy the Software onto your organization's computers for your organization's use, and you may make a reasonable number of back-up copies of the Software, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may not sublicense or permit simultaneous use of the Software by more than one user.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

SINGLE USER LICENSE. You may copy the Software onto a single computer for your personal, noncommercial use, and you may make one back-up copy of the Software, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may not sublicense or permit simultaneous use of the Software by more than one user.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

OEM LICENSE: You may reproduce and distribute the Software only as an integral part of or incorporated in Your product or as a standalone Software maintenance update for existing end users of Your products, excluding any other standalone products, subject to these conditions:

1. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.
2. You may not copy, modify, rent, sell, distribute or transfer any part of the Software except as provided in this Agreement, and you agree to prevent unauthorized copying of the Software.
3. You may not reverse engineer, decompile, or disassemble the Software.
4. You may only distribute the Software to your customers pursuant to a written license agreement. Such license agreement may be a "break-the-seal" license agreement. At a minimum such license shall safeguard Intel's ownership rights to the Software.
5. The Software may include portions offered on terms in addition to those set out here, as set out in a license accompanying those portions.

NO OTHER RIGHTS. No rights or licenses are granted by Intel to You, expressly or by implication, with respect to any proprietary information or patent, copyright, mask work, trademark, trade secret, or other intellectual property right owned or controlled by Intel, except as expressly provided in this Agreement.

OWNERSHIP OF SOFTWARE AND COPYRIGHTS. Title to all copies of the Software remains with Intel or its suppliers. The Software is copyrighted and protected by the laws of the United States and other countries, and international treaty provisions. You may not remove any copyright notices from the Software. Intel may make changes to the Software, or to items referenced therein, at any time without notice, but is not obligated to support or update the Software. Except as otherwise expressly provided, Intel grants no express or implied right under Intel patents, copyrights, trademarks, or other intellectual property rights. You may transfer the Software only if the recipient agrees to be fully bound by these terms and if you retain no copies of the Software.

LIMITED MEDIA WARRANTY. If the Software has been delivered by Intel on physical media, Intel warrants the media to be free from material physical defects for a period of ninety days after delivery by Intel. If such a defect is found, return the media to Intel for replacement or alternate delivery of the Software as Intel may select.

EXCLUSION OF OTHER WARRANTIES. EXCEPT AS PROVIDED ABOVE, THE SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. Intel does not warrant or assume responsibility for the accuracy or completeness of any information, text, graphics, links or other items contained within the Software.

LIMITATION OF LIABILITY. IN NO EVENT SHALL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, OR LOST INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS PROHIBIT EXCLUSION OR LIMITATION OF LIABILITY FOR IMPLIED WARRANTIES OR CONSEQUENTIAL OR INCIDENTAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE OTHER LEGAL RIGHTS THAT VARY FROM JURISDICTION TO JURISDICTION.

TERMINATION OF THIS AGREEMENT. Intel may terminate this Agreement at any time if you violate its terms. Upon termination, you will immediately destroy the Software or return all copies of the Software to Intel.

APPLICABLE LAWS. Claims arising under this Agreement shall be governed by the laws of California, excluding its principles of conflict of laws and the United Nations Convention on Contracts for the Sale of Goods. You may not export the Software in violation of applicable export laws and regulations. Intel is not obligated under any other agreements unless they are in writing and signed by an authorized representative of Intel.

GOVERNMENT RESTRICTED RIGHTS. The Software is provided with "RESTRICTED RIGHTS." Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor. Use of the Software by the Government constitutes acknowledgment of Intel's proprietary rights therein. Contractor or Manufacturer is Intel
2200 Mission College Blvd., Santa Clara, CA
95052.

/*

- * Copyright (c) 1983, 1988 Regents of the University of California.
- * All rights reserved.
- *
- * Redistribution and use in source and binary forms are permitted
- * provided that the above copyright notice and this paragraph are
- * duplicated in all such forms and that any documentation,
- * advertising materials, and other materials related to such
- * distribution and use acknowledge that the software was developed
- * by the University of California, Berkeley. The name of the
- * University may not be used to endorse or promote products derived
- * from this software without specific prior written permission.
- * THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR
- * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
- * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

*/

Automatically generated by apkbuidl-cpan, template 1
Contributor: Valery Kartel <valery.kartel@gmail.com>
Maintainer: Valery Kartel <valery.kartel@gmail.com>
pkgname=perl-bsd-resource

```

_pkgreal=BSD-Resource
pkgver=1.2909
pkgrel=2
pkgdesc="Perl extension implements the BSD process resource limit functions"
url="http://search.cpan.org/dist/BSD-Resource/"
arch="all"
license="GPL PerlArtistic"
cpandepends=""
cpanmakedepends=" "
depends="$cpandepends"
makedepends="perl-dev $cpanmakedepends"
subpackages="$pkgname-doc"
source="http://search.cpan.org/CPAN/authors/id/J/JH/JHI/${_pkgreal}-${pkgver}.tar.gz"

_buildddir="$srcdir/${_pkgreal}-${pkgver}"

prepare() {
    cd "$_buildddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}`
    PERL_MM_USE_DEFAULT=1 perl Makefile.PL INSTALLDIRS=vendor
}

build() {
    cd "$_buildddir"
    export CFLAGS=`perl -MConfig -E 'say $Config{ccflags}`
    make && make test || return 1
}

package() {
    cd "$_buildddir"
    make DESTDIR="$pkgdir" install || return 1
    find "$pkgdir"
    \(-name perllocal.pod -o -name .packlist\) -delete
}

md5sums="3c6dd2fa953088aa87e263f40818d012 BSD-Resource-1.2909.tar.gz"
sha256sums="c47d460e2bd00d8d049f6ff4a0f1f61957e289e8de45af5fc79e851e2e855e06 BSD-Resource-1.2909.tar.gz"
sha512sums="5f489eaac5089e82081b7cf51d62638750e443751b49415407354da96b17771819b5e6e85529faa41ddd85743363750157943c7f7700276cce2bd40a429fa430 BSD-Resource-1.2909.tar.gz"
SQLite Copyright
SQLite is in the
Public Domain

```

All of the deliverable code in SQLite has been dedicated to the public domain by the authors. All code authors, and representatives of the companies they work for, have signed affidavits dedicating their contributions to the public domain and originals of those signed affidavits are stored in a firesafe at the main offices of Hwaci. Anyone is free

to copy, modify, publish, use, compile, sell, or distribute the original SQLite code, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

The previous paragraph applies to the deliverable code in SQLite - those parts of the SQLite library that you actually bundle and ship with a larger application. Portions of the documentation and some code used as part of the build process might fall under other licenses. The details here are unclear. We do not worry about the licensing of the documentation and build code so much because none of these things are part of the core deliverable SQLite library.

All of the deliverable code in SQLite has been written from scratch. No code has been taken from other projects or from the open internet. Every line of code can be traced back to its original author, and all of those authors have public domain dedications on file. So the SQLite code base is clean and is uncontaminated with licensed code from other projects.

Obtaining An Explicit License To Use SQLite

Even though SQLite is in the public domain and does not require a license, some users want to obtain a license anyway. Some reasons for obtaining a license include:

You are using SQLite in a jurisdiction that does not recognize the public domain.

You are using SQLite in a jurisdiction that does not recognize the right of an author to dedicate their work to the public domain.

You want to hold a tangible legal document as evidence that you have the legal right to use and distribute SQLite.

Your legal department tells

you that you have to purchase a license.

If you feel like you really have to purchase a license for SQLite, Hwaci, the company that employs the architect and principal developers of SQLite, will sell you one.

Contributed Code

In order to keep SQLite completely free and unencumbered by copyright, all new contributors to the SQLite code base are asked to dedicate their contributions to the public domain. If you want to send a patch or enhancement for possible inclusion in the SQLite source tree, please accompany the patch with the following statement:

The author or authors of this code dedicate any and all copyright interest in this code to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this code under copyright law.

We are not able to accept patches or changes to SQLite that are not accompanied by a statement such as the above. In addition, if you make changes or enhancements as an employee, then a simple statement such as the above is insufficient. You must also send by surface mail a copyright release signed by a company officer. A signed original of the copyright release should be mailed to:

Hwaci
6200 Maple Cove Lane
Charlotte, NC 28269
USA

A template copyright release is available in PDF or HTML. You can use this release to make future changes.

see <http://www.sqlite.org/copyright.html>

Copyright (c) 2016 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software source and associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, compile, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject the following terms and conditions:

All modifications to the Materials used to create a binary that is distributed to third parties shall be provided to Khronos with an unrestricted license to use for the purposes of implementing bug fixes and enhancements to the Materials;

If the binary is used as part of an OpenCL(TM) implementation, whether binary is distributed together with or separately to that implementation, then recipient must become an OpenCL Adopter and follow the published OpenCL conformance process for that implementation, details at:
<http://www.khronos.org/conformance/>;

The above copyright notice, the OpenCL trademark license, and this permission notice shall be included in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

OpenCL is a trademark of Apple Inc. used under license by Khronos.

Contributor: Fabian Affolter <fabian@affolter-engineering.ch>

Maintainer: Fabian Affolter <fabian@affolter-engineering.ch>

pkgname=py-flake8-copyright

_pkgname=flake8-copyright

pkgver=0.2.0

pkgrel=1

pkgdesc="Extension for flake8 which checks for copyrights"

url="https://github.com/savoirfairelinux/flake8-copyright"

arch="noarch"

license="MIT"

depends="flake8"

makedepends="python3-dev"

source="https://files.pythonhosted.org/packages/source/\${_pkgname:0:1}/\${_pkgname}/\${_pkgname}-\${pkgver}.tar.gz"

```
builddir="$srcdir"/$_pkgname-$pkgver
```

```
build() {  
  cd "$builddir"  
  python3 setup.py build || return 1  
}
```

```
package() {  
  cd "$builddir"  
  python3 setup.py install --prefix=/usr --root="$pkgdir" || return 1  
}
```

```
md5sums="c6348990c6d74933bf5d5f6a55a14bdf flake8-copyright-0.2.0.tar.gz"  
sha256sums="aeef26eb4d5223c9cd5b101e68175fcef6d2b353bf36da688fdde62fccfe2b73 flake8-copyright-  
0.2.0.tar.gz"  
sha512sums="bab1bfaeed3a525ac3c7da12b1f2ec338c204cc69b72d05cd39560e528faff5d586c3b1be9570f4a977a7e  
2417b586e543bfba12abce256b2023c8c4993eeb90  
flake8-copyright-0.2.0.tar.gz"  
mini_sendmail - accept email on behalf of real sendmail
```

Copyright 1999,2015 by Jef Poskanzer <jef@mail.acme.com>.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING,
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

1.54 alpine-release 3.22.2-r0

1.54.1 Available under license :

MIT License

Copyright 2019-2021 Caleb Porzio and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT

OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT

1.55 libbsd 0.12.2-r0

1.55.1 Available under license :

Format: <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

Files:

*

Copyright:

Copyright 2004-2024 Guillem Jover <guillem@hadrons.org>

License: BSD-3-clause

Files:

include/bsd/err.h

include/bsd/stdlib.h

include/bsd/sys/param.h

include/bsd/unistd.h

src/bsd_getopt.c

src/err.c

src/fgetln.c

src/progname.c

Copyright:

Copyright 2005, 2008-2012, 2019 Guillem Jover <guillem@hadrons.org>

Copyright 2005 Hector Garcia Alvarez

Copyright 2005 Aurelien Jarno

Copyright 2006 Robert Millan

Copyright 2018 Facebook, Inc.

License: BSD-3-clause

Files:

include/bsd/netinet/ip_icmp.h

include/bsd/sys/bitstring.h

include/bsd/sys/queue.h

include/bsd/sys/time.h

include/bsd/timeconv.h

include/bsd/vis.h

man/bitstring.3bsd

man/errc.3bsd

man/explicit_bzero.3bsd

man/fgetln.3bsd

man/fgetwln.3bsd

man/fpurge.3bsd

man/funopen.3bsd

man/getbsize.3bsd

man/heapsort.3bsd

man/nlist.3bsd

man/pwcache.3bsd

man/queue.3bsd

man/radixsort.3bsd

man/reallocarray.3bsd

man/reallocf.3bsd

man/setmode.3bsd

man/strmode.3bsd

man/strnstr.3bsd

man/strtoi.3bsd

man/strtou.3bsd

man/unvis.3bsd

man/vis.3bsd

man/wcsncpy.3bsd

src/getbsize.c

src/heapsort.c

src/merge.c

src/nlist.c

src/pwcache.c

src/radixsort.c

src/setmode.c

src/strmode.c

src/strnstr.c

src/strtoi.c

src/strtou.c

src/unvis.c

Copyright:

Copyright 1980, 1982, 1986, 1989-1994

The Regents of the University of California. All rights reserved.

Copyright 1992 Keith Muller.

Copyright 2001 Mike Barcroft <mike@FreeBSD.org>

.

Some code is derived from software contributed to Berkeley by the American National Standards Committee X3, on Information Processing Systems.

.

Some code is derived from software contributed to Berkeley by Peter McIlroy.

.

Some code is derived from software contributed to Berkeley by Ronnie Kon at Mindcraft Inc., Kevin Lew and Elmer Yglesias.

.

Some code is derived from software contributed to Berkeley by Dave Borman at Cray Research, Inc.

.

Some code is derived from software contributed to Berkeley by Paul Vixie.

.

Some code is derived from software contributed to Berkeley by Chris Torek.

.

Copyright UNIX System Laboratories, Inc.

All or some portions of this file are derived from material licensed to the University of California by American Telephone and Telegraph Co. or Unix System Laboratories, Inc. and are reproduced herein with the permission of UNIX System Laboratories, Inc.

License: BSD-3-clause-Regents

Files:

src/vis.c

Copyright:

Copyright 1989, 1993

The Regents of the University of California. All rights reserved.

.

Copyright 1999, 2005 The NetBSD Foundation, Inc.

All rights reserved.

License: BSD-3-clause-Regents and BSD-2-clause-NetBSD

Files:

include/bsd/libutil.h

Copyright:

Copyright 1996 Peter Wemm <peter@FreeBSD.org>.

All rights reserved.

Copyright 2002 Networks Associates Technology, Inc.

All rights reserved.

License: BSD-3-clause-author

Files:

man/timeradd.3bsd

Copyright:

Copyright

2009 Jukka Ruohonen <jruohonen@iki.fi>

Copyright 1999 Kelly Yancey <kbyanc@posi.net>

All rights reserved.

License: BSD-3-clause-John-Birrell

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the author nor the names of any co-contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY JOHN BIRRELL AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

man/setproctitle.3bsd

Copyright:

Copyright 1995 Peter Wemm <peter@FreeBSD.org>

All rights reserved.

License: BSD-5-clause-Peter-Wemm

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions

are met:

1. Redistributions of source code must retain the above copyright notice immediately at the beginning of the file, without modification, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. This work was done expressly for inclusion into FreeBSD. Other use is permitted provided this notation is included.
4. Absolutely no warranty of function or purpose is made by the author Peter Wemm.
5. Modifications may be freely made to this file providing the above conditions are met.

Files:

include/bsd/stringlist.h
man/arc4random.3bsd
man/fmtcheck.3bsd
man/humanize_number.3bsd
man/stringlist.3bsd
man/timeval.3bsd
src/fmtcheck.c
src/humanize_number.c
src/stringlist.c
src/strtonum.c

Copyright:

Copyright 1994, 1997-2000, 2002, 2008, 2010, 2014

The NetBSD Foundation, Inc.

Copyright 2013 John-Mark Gurney <jmg@FreeBSD.org>

All rights reserved.

.

Copyright 2014

The NetBSD Foundation, Inc.

All rights reserved.

.

Some code was derived from software contributed to The NetBSD Foundation by Taylor R. Campbell.

.

Some code was contributed to The NetBSD Foundation by Allen Briggs.

.

Some code was contributed to The NetBSD Foundation by Luke Mewburn.

.

Some code is derived from software contributed to The NetBSD Foundation by Jason R. Thorpe of the Numerical Aerospace Simulation Facility, NASA Ames Research Center, by Luke Mewburn and by Tomas Svensson.

.

Some code is derived from software contributed to The NetBSD Foundation

by Julio M. Merino Vidal, developed as part of Google's Summer of Code 2005 program.

.

Some code is derived from software contributed to The NetBSD Foundation by Christos Zoulas.

.

Some code is derived from software contributed to The NetBSD Foundation by Jukka Ruohonen.

License: BSD-2-clause-NetBSD

Files:

include/bsd/sys/endian.h
man/byteorder.3bsd
man/closefrom.3bsd
man/expand_number.3bsd
man/flopen.3bsd
man/getpeereid.3bsd
man/pidfile.3bsd
src/expand_number.c
src/pidfile.c
src/reallocf.c
src/timeconv.c

Copyright:

Copyright 1998, M. Warner Losh <imp@freebsd.org>

All rights reserved.

.

Copyright 2001 Dima Dorfman.

All rights reserved.

.

Copyright 2001 FreeBSD Inc.

All rights reserved.

.

Copyright 2002 Thomas Moestl <tmml@FreeBSD.org>

All rights reserved.

.

Copyright 2002 Mike Barcroft <mike@FreeBSD.org>

All rights reserved.

.

Copyright 2005 Pawel Jakub Dawidek <pjd@FreeBSD.org>

All rights reserved.

.

Copyright 2005 Colin Percival

All rights reserved.

.

Copyright 2007 Eric Anderson <anderson@FreeBSD.org>

Copyright 2007 Pawel Jakub Dawidek <pjd@FreeBSD.org>

All rights reserved.

.

Copyright 2007 Dag-Erling Codan Smrgrav

All rights reserved.

.

Copyright 2009 Advanced Computing Technologies LLC

All rights reserved.

.

Copyright 2011 Guillem Jover <guillem@hadrons.org>

License: BSD-2-clause

Files:

src/flopen.c

Copyright:

Copyright 2007-2009

Dag-Erling Codan Smrgrav

All rights reserved.

License: BSD-2-clause-verbatim

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer in this position and unchanged.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:

include/bsd/sys/tree.h

man/fparsefn.3bsd

man/tree.3bsd

src/fparsefn.c

Copyright:

Copyright 1997 Christos Zoulas.

All rights reserved.

.

Copyright 2002 Niels Provos <provos@citi.umich.edu>

All rights reserved.

License: BSD-2-clause-author

Files:

include/bsd/readpassphrase.h

man/readpassphrase.3bsd

man/strncpy.3bsd

man/strtonum.3bsd

src/arc4random.c

src/arc4random_linux.h

src/arc4random_uniform.c

src/arc4random_unix.h

src/arc4random_win.h

src/closefrom.c

src/freezero.c

src/getentropy_aix.c

src/getentropy_bsd.c

src/getentropy_hpux.c

src/getentropy_hurd.c

src/getentropy_linux.c

src/getentropy_osx.c

src/getentropy_solaris.c

src/getentropy_win.c

src/readpassphrase.c

src/reallocarray.c

src/reallocarray.c

src/strlcat.c

src/strncpy.c

test/explicit_bzero.c

test/strtonum.c

Copyright:

Copyright 2004 Ted Unangst and Todd Miller

All rights reserved.

.

Copyright 1996 David Mazieres <dm@uun.org>

Copyright 1998, 2000-2002, 2004-2005, 2007, 2010, 2012-2015

Todd C. Miller <Todd.Miller@courtesan.com>

Copyright 2004 Ted Unangst

Copyright 2004 Otto Moerbeek <otto@drijf.net>

Copyright 2008 Damien Miller <djm@openbsd.org>

Copyright 2008, 2010-2011, 2016-2017 Otto Moerbeek <otto@drijf.net>

Copyright 2013 Markus Friedl <markus@openbsd.org>

Copyright 2014 Bob Beck <beck@obtuse.com>

Copyright 2014 Brent Cook <bcook@openbsd.org>

Copyright 2014 Pawel Jakub Dawidek <pjd@FreeBSD.org>

Copyright 2014 Theo de Raadt <deraadt@openbsd.org>

Copyright 2014 Google Inc.

Copyright 2015 Michael Felt <aixtools@gmail.com>

Copyright 2015, 2022 Guillem Jover <guillem@hadrons.org>

License:

ISC

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:

src/inet_net_pton.c

Copyright:

Copyright 1996 by Internet Software Consortium.

License: ISC-Original

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:

src/setproctitle.c

Copyright:

Copyright 2010 William Ahern

Copyright 2012 Guillem Jover <guillem@hadrons.org>

License: Expat

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the

following conditions:

.
The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN
NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE
USE OR OTHER DEALINGS IN THE SOFTWARE.

Files:

src/explicit_bzero.c

src/chacha_private.h

Copyright:

None

License: public-domain

Public domain.

Files:

man/mdX.3bsd

Copyright:

None

License: Beerware

"THE BEER-WARE LICENSE" (Revision 42):

<phk@login.dkuug.dk> wrote this

file. As long as you retain this notice you
can do whatever you want with this stuff. If we meet some day, and you think
this stuff is worth it, you can buy me a beer in return. Poul-Henning Kamp

License: BSD-3-clause-Regents

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors
may be used to endorse or promote products derived from this software
without specific prior written permission.

.
THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-3-clause-author

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-3-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE

AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause-NetBSD

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause-author

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.56 confluentinc-confluent-kafka-go 2.12.0

1.56.1 Available under license :

LICENSE

librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2022, Magnus Edenhill

2023, Confluent Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,

this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.cjson

For cJSON.c and cJSON.h:

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished

to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.crc32c

For src/crc32c.c copied (with modifications) from
<http://stackoverflow.com/a/17646775/1821055>

/* crc32c.c -- compute CRC-32C using the Intel crc32 instruction
* Copyright (C) 2013 Mark Adler
* Version 1.1 1 Aug 2013 Mark Adler
*/

/*
This software is provided 'as-is',
without any express or implied
warranty. In no event will the author be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Mark Adler
madler@alumni.caltech.edu
*/

LICENSE.fnv1a

parts of src/rdfnv1a.c: http://www.isthe.com/chongo/src/fnv/hash_32a.c

Please
do not copyright this code. This code is in the public domain.

LONDON CURT NOLL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO
EVENT SHALL LONDON CURT NOLL BE LIABLE FOR ANY SPECIAL, INDIRECT OR
CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF
USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR
OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

By:

chongo <Landon Curt Noll> /oo/
<http://www.isthe.com/chongo/>

Share and Enjoy! :-)

LICENSE.hdrhistogram

This license covers src/rdhdrhistogram.c which is a C port of
Coda Hale's Golang HdrHistogram <https://github.com/codahale/hdrhistogram>
at revision 3a0bb77429bd3a61596f5e8a3172445844342120

The MIT License
(MIT)

Copyright (c) 2014 Coda Hale

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH
THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE

LICENSE.lz4

src/rdxxhash.[ch] src/lz4*. [ch]: [git@github.com:lz4/lz4](https://github.com/lz4/lz4).git 5ff839680134437dbf4678f3d0c7b371d84f4964

LZ4 Library
Copyright (c) 2011-2020, Yann Collet
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,

are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.murmur2

parts of src/rdmurmur2.c: git@github.com:abandoned/murmur2.git

MurMurHash2 Library

//-----
// MurmurHash2 was written by Austin Appleby, and is placed in the public
// domain. The author hereby disclaims copyright to this source code.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE.nanopb

For

files in src/nanopb :

<https://github.com/nanopb/nanopb/blob/8ef41e0ebd45daaf19459a011f67e66224b247cd/LICENSE.txt>

Copyright (c) 2011 Petteri Aimonen <jpa at nanopb.mail.kapsi.fi>

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

LICENSE.opentelemetry

For files in src/opentelemetry: <https://github.com/open-telemetry/opentelemetry-proto/blob/81a296f9dba23e32d77f46d58c8ea4244a2157a6/LICENSE>

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution

incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

LICENSE.pycrc

The following license applies to the files rdcrc32.c and rdcrc32.h which have been generated by the pycrc tool.

=====

Copyright (c) 2006-2012, Thomas Pircher <tehpeh@gmx.net>

Permission is hereby granted,
free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

LICENSE.queue

For
sys/queue.h:

- * Copyright (c) 1991, 1993
- * The Regents of the University of California. All rights reserved.
- *
- * Redistribution and use in source and binary forms, with or without
- * modification, are permitted provided that the following conditions
- * are met:

- * 1. Redistributions of source code must retain the above copyright
- * notice, this list of conditions and the following disclaimer.
- * 2. Redistributions in binary form must reproduce the above copyright
- * notice, this list of conditions and the following disclaimer in the
- * documentation and/or other materials provided with the distribution.
- * 4. Neither the name of the University nor the names of its contributors
- * may be used to endorse or promote products derived from this software
- * without specific prior written permission.
- *
- * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND
- * ANY EXPRESS OR
- IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
- * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
- * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
- * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
- * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
- * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
- * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
- * SUCH DAMAGE.
- *
- * @(#)queue.h 8.5 (Berkeley) 8/20/94
- * \$FreeBSD\$

LICENSE.regexp

 regexp.c and regexp.h from <https://github.com/ccxvii/minilibs> sha 875c33568b5a4aa4fb3dd0c52ea98f7f0e5ca684

"

These libraries are in the public domain (or the equivalent where that is not possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.

"

LICENSE.snappy

 #####
 # LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h #
 # originally retrieved from <http://github.com/andikleen/snappy-c> #
 # git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219 #
 #####

The snappy-c code is under the same license as the original snappy source

Copyright 2011 Intel Corporation All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE.tinychread

From <https://github.com/tinychread/tinychread/README.txt> c57166cd510ffb5022dd5f127489b131b61441b9

License

Copyright (c) 2012 Marcus Geelnard
2013-2014 Evan Nemerson

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

LICENSE.wingetopt

For the files wingetopt.c wingetopt.h downloaded from <https://github.com/alex85k/wingetopt>

```
/*
 * Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>
 *
 * Permission to use, copy, modify, and distribute this software for any
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
 * WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
 * ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR
 * ANY DAMAGES
 * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
 * ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
 * OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
 *
 * Sponsored in part by the Defense Advanced Research Projects
 * Agency (DARPA) and Air Force Research Laboratory, Air Force
 * Materiel Command, USAF, under agreement number F39502-99-1-0512.
 */
/*-
 * Copyright (c) 2000 The NetBSD Foundation, Inc.
 * All rights reserved.
 *
 * This code is derived from software contributed to The NetBSD Foundation
 * by Dieter Baron and Thomas Klausner.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
```

* 2. Redistributions in binary form must reproduce the above copyright
* notice,
this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
*
* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS
* ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
* TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS
* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.
*/

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You"
(or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work

or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental,

or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.57 elastic/go-elasticsearch 7.17.10

1.57.1 Available under license :

```
// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at
```

```
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
//
// Code generated from specification version 7.17.10: DO NOT EDIT

package esapi

import (
    "context"
    "io"
    "net/http"
    "strconv"
    "strings"
)

func newLicensePostFunc(t Transport) LicensePost {
    return func(o
...func(*LicensePostRequest)) (*Response, error) {
    var r = LicensePostRequest{}
    for _, f := range o {
        f(&r)
    }
    return r.Do(r.ctx, t)
}
}

// ----- API Definition -----

// LicensePost - Updates the license for the cluster.
//
// See full documentation at https://www.elastic.co/guide/en/elasticsearch/reference/master/update-license.html.
type LicensePost func(o ...func(*LicensePostRequest)) (*Response, error)

// LicensePostRequest configures the License Post API request.
type LicensePostRequest struct {
    Body io.Reader

    Acknowledge *bool

    Pretty bool
    Human bool
}
```

```

ErrorTrace bool
FilterPath []string

Header http.Header

ctx context.Context
}

// Do executes the request and returns response or error.
func (r LicensePostRequest) Do(ctx context.Context, transport Transport) (*Response, error) {
    var (
        method string
        path    strings.Builder
        params  map[string]string
    )

    method = "PUT"

    path.Grow(len("/_license"))
    path.WriteString("/_license")

    params
    = make(map[string]string)

    if r.Acknowledge != nil {
        params["acknowledge"] = strconv.FormatBool(*r.Acknowledge)
    }

    if r.Pretty {
        params["pretty"] = "true"
    }

    if r.Human {
        params["human"] = "true"
    }

    if r.ErrorTrace {
        params["error_trace"] = "true"
    }

    if len(r.FilterPath) > 0 {
        params["filter_path"] = strings.Join(r.FilterPath, ",")
    }

    req, err := newRequest(method, path.String(), r.Body)
    if err != nil {
        return nil, err
    }

```



```

if len(params) > 0 {
    q := req.URL.Query()
    for k, v := range params {
        q.Set(k, v)
    }
    req.URL.RawQuery = q.Encode()
}

if len(r.Header) > 0 {
    if len(req.Header) == 0 {
        req.Header = r.Header
    } else {
        for k, vv := range r.Header {
            for _, v := range vv {
                req.Header.Add(k, v)
            }
        }
    }
}

if r.Body != nil && req.Header.Get(headerContentType) == "" {
    req.Header[headerContentType] = headerContentTypeJSON
}

if ctx != nil {
    req = req.WithContext(ctx)
}

res, err
:= transport.Perform(req)
if err != nil {
    return nil, err
}

response := Response{
    StatusCode: res.StatusCode,
    Body:      res.Body,
    Header:    res.Header,
}

return &response, nil
}

// WithContext sets the request context.
func (f LicensePost) WithContext(v context.Context) func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        r.ctx = v
    }
}

```

```

}
}

// WithBody - licenses to be installed.
func (f LicensePost) WithBody(v io.Reader) func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        r.Body = v
    }
}

// WithAcknowledge - whether the user has acknowledged acknowledge messages (default: false).
func (f LicensePost) WithAcknowledge(v bool) func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        r.Acknowledge = &v
    }
}

// WithPretty makes the response body pretty-printed.
func (f LicensePost) WithPretty() func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        r.Pretty = true
    }
}

// WithHuman makes statistical values human-readable.
func
(f LicensePost) WithHuman() func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        r.Human = true
    }
}

// WithErrorTrace includes the stack trace for errors in the response body.
func (f LicensePost) WithErrorTrace() func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        r.ErrorTrace = true
    }
}

// WithFilterPath filters the properties of the response body.
func (f LicensePost) WithFilterPath(v ...string) func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        r.FilterPath = v
    }
}

// WithHeader adds the headers to the HTTP request.
func (f LicensePost) WithHeader(h map[string]string) func(*LicensePostRequest) {

```

```

return func(r *LicensePostRequest) {
    if r.Header == nil {
        r.Header = make(http.Header)
    }
    for k, v := range h {
        r.Header.Add(k, v)
    }
}

// WithOpaqueID adds the X-Opaque-Id header to the HTTP request.
func (f LicensePost) WithOpaqueID(s string) func(*LicensePostRequest) {
    return func(r *LicensePostRequest) {
        {
            if r.Header == nil {
                r.Header = make(http.Header)
            }
            r.Header.Set("X-Opaque-Id", s)
        }
    }
}

// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
//
// Code generated from specification version 7.17.10: DO NOT EDIT

package esapi

import (
    "context"
    "net/http"
    "strings"
)

func newLicenseGetBasicStatusFunc(t Transport) LicenseGetBasicStatus {

```

```

return
func(o ...func(*LicenseGetBasicStatusRequest)) (*Response, error) {
    var r = LicenseGetBasicStatusRequest{}
    for _, f := range o {
        f(&r)
    }
    return r.Do(r.ctx, t)
}
}

// ----- API Definition -----

// LicenseGetBasicStatus - Retrieves information about the status of the basic license.
//
// See full documentation at https://www.elastic.co/guide/en/elasticsearch/reference/master/get-basic-status.html.
type LicenseGetBasicStatus func(o ...func(*LicenseGetBasicStatusRequest)) (*Response, error)

// LicenseGetBasicStatusRequest configures the License Get Basic Status API request.
type LicenseGetBasicStatusRequest struct {
    Pretty    bool
    Human     bool
    ErrorTrace bool
    FilterPath []string

    Header http.Header

    ctx context.Context
}

// Do executes the request and returns response or error.
func (r LicenseGetBasicStatusRequest) Do(ctx context.Context, transport Transport) (*Response, error) {
    var (
        method string
        path    strings.Builder
        params  map[string]string
    )

    method = "GET"

    path.Grow(len("/_license/basic_status"))
    path.WriteString("/_license/basic_status")

    params = make(map[string]string)

    if r.Pretty {
        params["pretty"] = "true"
    }

```

```

if r.Human {
    params["human"] = "true"
}

if r.ErrorTrace {
    params["error_trace"] = "true"
}

if len(r.FilterPath) > 0 {
    params["filter_path"] = strings.Join(r.FilterPath, ",")
}

req, err := newRequest(method, path.String(), nil)
if err != nil {
    return nil, err
}

if len(params) > 0 {
    q := req.URL.Query()
    for k, v := range params {
        q.Set(k, v)
    }
    req.URL.RawQuery = q.Encode()
}

if len(r.Header) > 0 {
    if len(req.Header) == 0 {
        req.Header = r.Header
    } else {
        for k, vv := range r.Header {
            for _, v := range vv {
                req.Header.Add(k, v)
            }
        }
    }
}

if ctx != nil {
    req = req.WithContext(ctx)
}

res, err := transport.Perform(req)
if err != nil {
    return nil, err
}

response := Response{

```

```

    StatusCode: res.StatusCode,
    Body:
        res.Body,
    Header: res.Header,
}

return &response, nil
}

// WithContext sets the request context.
func (f LicenseGetBasicStatus) WithContext(v context.Context) func(*LicenseGetBasicStatusRequest) {
    return func(r *LicenseGetBasicStatusRequest) {
        r.ctx = v
    }
}

// WithPretty makes the response body pretty-printed.
func (f LicenseGetBasicStatus) WithPretty() func(*LicenseGetBasicStatusRequest) {
    return func(r *LicenseGetBasicStatusRequest) {
        r.Pretty = true
    }
}

// WithHuman makes statistical values human-readable.
func (f LicenseGetBasicStatus) WithHuman() func(*LicenseGetBasicStatusRequest) {
    return func(r *LicenseGetBasicStatusRequest) {
        r.Human = true
    }
}

// WithErrorTrace includes the stack trace for errors in the response body.
func (f LicenseGetBasicStatus) WithErrorTrace() func(*LicenseGetBasicStatusRequest) {
    return func(r *LicenseGetBasicStatusRequest) {
        r.ErrorTrace = true
    }
}

// WithFilterPath filters the properties of the
response body.
func (f LicenseGetBasicStatus) WithFilterPath(v ...string) func(*LicenseGetBasicStatusRequest) {
    return func(r *LicenseGetBasicStatusRequest) {
        r.FilterPath = v
    }
}

// WithHeader adds the headers to the HTTP request.
func (f LicenseGetBasicStatus) WithHeader(h map[string]string) func(*LicenseGetBasicStatusRequest) {
    return func(r *LicenseGetBasicStatusRequest) {

```

```

if r.Header == nil {
    r.Header = make(http.Header)
}
for k, v := range h {
    r.Header.Add(k, v)
}
}
}

// WithOpaqueID adds the X-Opaque-Id header to the HTTP request.
func (f LicenseGetBasicStatus) WithOpaqueID(s string) func(*LicenseGetBasicStatusRequest) {
    return func(r *LicenseGetBasicStatusRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        r.Header.Set("X-Opaque-Id", s)
    }
}
name: License headers

```

on: [pull_request]

jobs:

build:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: Check license headers

run: |

./github/check-license-headers.sh

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2018 Elasticsearch BV

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

```
// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
//
// Code generated from specification version 7.17.10: DO NOT EDIT
```

package esapi

```
import (
    "context"
    "net/http"
    "strconv"
    "strings"
)
```

```
func newLicenseGetFunc(t Transport) LicenseGet {
    return func(o ...func(*LicenseGetRequest))
        (*Response, error) {
            var r = LicenseGetRequest{}
            for _, f := range o {
                f(&r)
            }
            return r.Do(r.ctx, t)
        }
}
```

// ----- API Definition -----

```
// LicenseGet - Retrieves licensing information for the cluster
//
```

```
// See full documentation at https://www.elastic.co/guide/en/elasticsearch/reference/master/get-license.html.
type LicenseGet func(o ...func(*LicenseGetRequest)) (*Response, error)
```

```

// LicenseGetRequest configures the License Get API request.
type LicenseGetRequest struct {
    AcceptEnterprise *bool
    Local           *bool

    Pretty   bool
    Human    bool
    ErrorTrace bool
    FilterPath []string

    Header http.Header

    ctx context.Context
}

// Do executes the request and returns response or error.
func (r LicenseGetRequest) Do(ctx context.Context, transport Transport) (*Response, error) {
    var (
        method string
        path    strings.Builder
        params  map[string]string
    )

    method = "GET"

    path.Grow(len("/_license"))
    path.WriteString("/_license")

    params
    = make(map[string]string)

    if r.AcceptEnterprise != nil {
        params["accept_enterprise"] = strconv.FormatBool(*r.AcceptEnterprise)
    }

    if r.Local != nil {
        params["local"] = strconv.FormatBool(*r.Local)
    }

    if r.Pretty {
        params["pretty"] = "true"
    }

    if r.Human {
        params["human"] = "true"
    }

    if r.ErrorTrace {

```

```

params["error_trace"] = "true"
}

if len(r.FilterPath) > 0 {
    params["filter_path"] = strings.Join(r.FilterPath, ",")
}

req, err := newRequest(method, path.String(), nil)
if err != nil {
    return nil, err
}

if len(params) > 0 {
    q := req.URL.Query()
    for k, v := range params {
        q.Set(k, v)
    }
    req.URL.RawQuery = q.Encode()
}

if len(r.Header) > 0 {
    if len(req.Header) == 0 {
        req.Header = r.Header
    } else {
        for k, vv := range r.Header {
            for _, v := range vv {
                req.Header.Add(k, v)
            }
        }
    }
}

if ctx != nil {
    req = req.WithContext(ctx)
}

res, err := transport.Perform(req)
if
err != nil {
    return nil, err
}

response := Response{
    StatusCode: res.StatusCode,
    Body:      res.Body,
    Header:    res.Header,
}

```

```

    return &response, nil
}

// WithContext sets the request context.
func (f LicenseGet) WithContext(v context.Context) func(*LicenseGetRequest) {
    return func(r *LicenseGetRequest) {
        r.ctx = v
    }
}

// WithAcceptEnterprise - if the active license is an enterprise license, return type as 'enterprise' (default: false).
func (f LicenseGet) WithAcceptEnterprise(v bool) func(*LicenseGetRequest) {
    return func(r *LicenseGetRequest) {
        r.AcceptEnterprise = &v
    }
}

// WithLocal - return local information, do not retrieve the state from master node (default: false).
func (f LicenseGet) WithLocal(v bool) func(*LicenseGetRequest) {
    return func(r *LicenseGetRequest) {
        r.Local = &v
    }
}

// WithPretty makes the response body pretty-printed.
func (f LicenseGet) WithPretty() func(*LicenseGetRequest) {
    return func(r *LicenseGetRequest) {
        r.Pretty = true
    }
}

//
// WithHuman makes statistical values human-readable.
func (f LicenseGet) WithHuman() func(*LicenseGetRequest) {
    return func(r *LicenseGetRequest) {
        r.Human = true
    }
}

// WithErrorTrace includes the stack trace for errors in the response body.
func (f LicenseGet) WithErrorTrace() func(*LicenseGetRequest) {
    return func(r *LicenseGetRequest) {
        r.ErrorTrace = true
    }
}

// WithFilterPath filters the properties of the response body.
func (f LicenseGet) WithFilterPath(v ...string) func(*LicenseGetRequest) {

```

```

return func(r *LicenseGetRequest) {
    r.FilterPath = v
}
}

// WithHeader adds the headers to the HTTP request.
func (f LicenseGet) WithHeader(h map[string]string) func(*LicenseGetRequest) {
    return func(r *LicenseGetRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        for k, v := range h {
            r.Header.Add(k, v)
        }
    }
}

// WithOpaqueID adds the X-Opaque-Id header to the HTTP request.
func (f LicenseGet) WithOpaqueID(s string) func(*LicenseGetRequest)
{
    return func(r *LicenseGetRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        r.Header.Set("X-Opaque-Id", s)
    }
}

// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may

```



```

// not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
//
// Code generated from specification version 7.17.10: DO NOT EDIT

package esapi

import (
    "context"
    "net/http"
    "strings"
)

func newLicenseDeleteFunc(t Transport) LicenseDelete {
    return func(o ...func(*LicenseDeleteRequest))
        (*Response, error) {
            var r = LicenseDeleteRequest{}
            for _, f := range o {
                f(&r)
            }
            return r.Do(r.ctx, t)
        }
    }

// ----- API Definition -----

// LicenseDelete - Deletes licensing information for the cluster
//
// See full documentation at https://www.elastic.co/guide/en/elasticsearch/reference/master/delete-license.html.
type LicenseDelete func(o ...func(*LicenseDeleteRequest)) (*Response, error)

// LicenseDeleteRequest configures the License Delete API request.
type LicenseDeleteRequest struct {
    Pretty    bool
    Human     bool
    ErrorTrace bool
    FilterPath []string

    Header http.Header

```

```

ctx context.Context
}

// Do executes the request and returns response or error.
func (r LicenseDeleteRequest) Do(ctx context.Context, transport Transport) (*Response, error) {
    var (
        method string
        path  strings.Builder
        params map[string]string
    )

    method = "DELETE"

    path.Grow(len("/_license"))
    path.WriteString("/_license")

    params
    = make(map[string]string)

    if r.Pretty {
        params["pretty"] = "true"
    }

    if r.Human {
        params["human"] = "true"
    }

    if r.ErrorTrace {
        params["error_trace"] = "true"
    }

    if len(r.FilterPath) > 0 {
        params["filter_path"] = strings.Join(r.FilterPath, ",")
    }

    req, err := newRequest(method, path.String(), nil)
    if err != nil {
        return nil, err
    }

    if len(params) > 0 {
        q := req.URL.Query()
        for k, v := range params {
            q.Set(k, v)
        }
        req.URL.RawQuery = q.Encode()
    }
}

```

```

if len(r.Header) > 0 {
    if len(req.Header) == 0 {
        req.Header = r.Header
    } else {
        for k, vv := range r.Header {
            for _, v := range vv {
                req.Header.Add(k, v)
            }
        }
    }
}

if ctx != nil {
    req = req.WithContext(ctx)
}

res, err := transport.Perform(req)
if err != nil {
    return nil, err
}

response := Response{
    StatusCode: res.StatusCode,
    Body:      res.Body,
    Header:   res.Header,
}

return &response, nil
}

// WithContext sets the request context.
func
(f LicenseDelete) WithContext(v context.Context) func(*LicenseDeleteRequest) {
    return func(r *LicenseDeleteRequest) {
        r.ctx = v
    }
}

// WithPretty makes the response body pretty-printed.
func (f LicenseDelete) WithPretty() func(*LicenseDeleteRequest) {
    return func(r *LicenseDeleteRequest) {
        r.Pretty = true
    }
}

// WithHuman makes statistical values human-readable.
func (f LicenseDelete) WithHuman() func(*LicenseDeleteRequest) {

```

```

return func(r *LicenseDeleteRequest) {
    r.Human = true
}
}

// WithErrorTrace includes the stack trace for errors in the response body.
func (f LicenseDelete) WithErrorTrace() func(*LicenseDeleteRequest) {
    return func(r *LicenseDeleteRequest) {
        r.ErrorTrace = true
    }
}

// WithFilterPath filters the properties of the response body.
func (f LicenseDelete) WithFilterPath(v ...string) func(*LicenseDeleteRequest) {
    return func(r *LicenseDeleteRequest) {
        r.FilterPath = v
    }
}

// WithHeader adds the headers to the HTTP request.
func (f LicenseDelete)
    WithHeader(h map[string]string) func(*LicenseDeleteRequest) {
    return func(r *LicenseDeleteRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        for k, v := range h {
            r.Header.Add(k, v)
        }
    }
}

// WithOpaqueID adds the X-Opaque-Id header to the HTTP request.
func (f LicenseDelete) WithOpaqueID(s string) func(*LicenseDeleteRequest) {
    return func(r *LicenseDeleteRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        r.Header.Set("X-Opaque-Id", s)
    }
}

// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at

```

```
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
//
// Code generated from specification version 7.17.10: DO NOT EDIT

package esapi

import (
    "context"
    "net/http"
    "strconv"
    "strings"
)

func newLicensePostStartTrialFunc(t Transport) LicensePostStartTrial
{
    return func(o ...func(*LicensePostStartTrialRequest)) (*Response, error) {
        var r = LicensePostStartTrialRequest{}
        for _, f := range o {
            f(&r)
        }
        return r.Do(r.ctx, t)
    }
}

// ----- API Definition -----

// LicensePostStartTrial - starts a limited time trial license.
//
// See full documentation at https://www.elastic.co/guide/en/elasticsearch/reference/master/start-trial.html.
type LicensePostStartTrial func(o ...func(*LicensePostStartTrialRequest)) (*Response, error)

// LicensePostStartTrialRequest configures the License Post Start Trial API request.
type LicensePostStartTrialRequest struct {
    Acknowledge *bool
    DocumentType string

    Pretty bool
    Human bool
    ErrorTrace bool
    FilterPath []string
}
```

```

Header http.Header

ctx context.Context
}

// Do executes the request and returns response or error.
func (r LicensePostStartTrialRequest) Do(ctx context.Context, transport Transport) (*Response, error) {
    var (
        method
        string
        path strings.Builder
        params map[string]string
    )

    method = "POST"

    path.Grow(len("/_license/start_trial"))
    path.WriteString("/_license/start_trial")

    params = make(map[string]string)

    if r.Acknowledge != nil {
        params["acknowledge"] = strconv.FormatBool(*r.Acknowledge)
    }

    if r.DocumentType != "" {
        params["type"] = r.DocumentType
    }

    if r.Pretty {
        params["pretty"] = "true"
    }

    if r.Human {
        params["human"] = "true"
    }

    if r.ErrorTrace {
        params["error_trace"] = "true"
    }

    if len(r.FilterPath) > 0 {
        params["filter_path"] = strings.Join(r.FilterPath, ",")
    }

    req, err := newRequest(method, path.String(), nil)
    if err != nil {

```

```

    return nil, err
}

if len(params) > 0 {
    q := req.URL.Query()
    for k, v := range params {
        q.Set(k, v)
    }
    req.URL.RawQuery = q.Encode()
}

if len(r.Header) > 0 {
    if len(req.Header) == 0 {
        req.Header = r.Header
    } else {
        for k, vv := range r.Header {
            for _, v := range vv {
                req.Header.Add(k, v)
            }
        }
    }
}

if
ctx != nil {
    req = req.WithContext(ctx)
}

res, err := transport.Perform(req)
if err != nil {
    return nil, err
}

response := Response{
    StatusCode: res.StatusCode,
    Body:      res.Body,
    Header:   res.Header,
}

return &response, nil
}

// WithContext sets the request context.
func (f LicensePostStartTrial) WithContext(v context.Context) func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {
        r.ctx = v
    }
}

```

```

// WithAcknowledge - whether the user has acknowledged acknowledge messages (default: false).
func (f LicensePostStartTrial) WithAcknowledge(v bool) func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {
        r.Acknowledge = &v
    }
}

// WithDocumentType - the type of trial license to generate (default: "trial").
func (f LicensePostStartTrial) WithDocumentType(v string) func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {
        r.DocumentType = v
    }
}

// WithPretty
// makes the response body pretty-printed.
func (f LicensePostStartTrial) WithPretty() func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {
        r.Pretty = true
    }
}

// WithHuman makes statistical values human-readable.
func (f LicensePostStartTrial) WithHuman() func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {
        r.Human = true
    }
}

// WithErrorTrace includes the stack trace for errors in the response body.
func (f LicensePostStartTrial) WithErrorTrace() func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {
        r.ErrorTrace = true
    }
}

// WithFilterPath filters the properties of the response body.
func (f LicensePostStartTrial) WithFilterPath(v ...string) func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {
        r.FilterPath = v
    }
}

// WithHeader adds the headers to the HTTP request.
func (f LicensePostStartTrial) WithHeader(h map[string]string) func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {

```



```

if r.Header == nil {
    r.Header = make(http.Header)
}
for k, v := range h {
    r.Header.Add(k, v)
}
}
}

// WithOpaqueID adds the X-Opaque-Id header to the HTTP request.
func (f LicensePostStartTrial) WithOpaqueID(s string) func(*LicensePostStartTrialRequest) {
    return func(r *LicensePostStartTrialRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        r.Header.Set("X-Opaque-Id", s)
    }
}

// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
//
// Code generated from specification version 7.17.10: DO NOT EDIT

package esapi

import (
    "context"
    "net/http"
    "strings"
)

func newLicenseGetTrialStatusFunc(t Transport) LicenseGetTrialStatus {
    return

```

```

func(o ...func(*LicenseGetTrialStatusRequest)) (*Response, error) {
    var r = LicenseGetTrialStatusRequest{}
    for _, f := range o {
        f(&r)
    }
    return r.Do(r.ctx, t)
}
}

// ----- API Definition -----

// LicenseGetTrialStatus - Retrieves information about the status of the trial license.
//
// See full documentation at https://www.elastic.co/guide/en/elasticsearch/reference/master/get-trial-status.html.
type LicenseGetTrialStatus func(o ...func(*LicenseGetTrialStatusRequest)) (*Response, error)

// LicenseGetTrialStatusRequest configures the License Get Trial Status API request.
type LicenseGetTrialStatusRequest struct {
    Pretty    bool
    Human     bool
    ErrorTrace bool
    FilterPath []string

    Header http.Header

    ctx context.Context
}

// Do executes the request and returns response or error.
func (r LicenseGetTrialStatusRequest) Do(ctx context.Context, transport Transport) (*Response, error) {
    var (
        method string
        path    strings.Builder
        params  map[string]string
    )

    method = "GET"

    path.Grow(len("/_license/trial_status"))
    path.WriteString("/_license/trial_status")

    params = make(map[string]string)

    if r.Pretty {
        params["pretty"] = "true"
    }
}

```

```

if r.Human {
    params["human"] = "true"
}

if r.ErrorTrace {
    params["error_trace"] = "true"
}

if len(r.FilterPath) > 0 {
    params["filter_path"] = strings.Join(r.FilterPath, ",")
}

req, err := newRequest(method, path.String(), nil)
if err != nil {
    return nil, err
}

if len(params) > 0 {
    q := req.URL.Query()
    for k, v := range params {
        q.Set(k, v)
    }
    req.URL.RawQuery = q.Encode()
}

if len(r.Header) > 0 {
    if len(req.Header) == 0 {
        req.Header = r.Header
    } else {
        for k, vv := range r.Header {
            for _, v := range vv {
                req.Header.Add(k, v)
            }
        }
    }
}

if ctx != nil {
    req = req.WithContext(ctx)
}

res, err := transport.Perform(req)
if err != nil {
    return nil, err
}

response := Response{
    StatusCode: res.StatusCode,

```

```

    Body:
        res.Body,
    Header: res.Header,
}

return &response, nil
}

// WithContext sets the request context.
func (f LicenseGetTrialStatus) WithContext(v context.Context) func(*LicenseGetTrialStatusRequest) {
    return func(r *LicenseGetTrialStatusRequest) {
        r.ctx = v
    }
}

// WithPretty makes the response body pretty-printed.
func (f LicenseGetTrialStatus) WithPretty() func(*LicenseGetTrialStatusRequest) {
    return func(r *LicenseGetTrialStatusRequest) {
        r.Pretty = true
    }
}

// WithHuman makes statistical values human-readable.
func (f LicenseGetTrialStatus) WithHuman() func(*LicenseGetTrialStatusRequest) {
    return func(r *LicenseGetTrialStatusRequest) {
        r.Human = true
    }
}

// WithErrorTrace includes the stack trace for errors in the response body.
func (f LicenseGetTrialStatus) WithErrorTrace() func(*LicenseGetTrialStatusRequest) {
    return func(r *LicenseGetTrialStatusRequest) {
        r.ErrorTrace = true
    }
}

// WithFilterPath filters the properties of the
response body.
func (f LicenseGetTrialStatus) WithFilterPath(v ...string) func(*LicenseGetTrialStatusRequest) {
    return func(r *LicenseGetTrialStatusRequest) {
        r.FilterPath = v
    }
}

// WithHeader adds the headers to the HTTP request.
func (f LicenseGetTrialStatus) WithHeader(h map[string]string) func(*LicenseGetTrialStatusRequest) {
    return func(r *LicenseGetTrialStatusRequest) {
        if r.Header == nil {

```

```

    r.Header = make(http.Header)
}
for k, v := range h {
    r.Header.Add(k, v)
}
}
}

// WithOpaqueID adds the X-Opaque-Id header to the HTTP request.
func (f LicenseGetTrialStatus) WithOpaqueID(s string) func(*LicenseGetTrialStatusRequest) {
    return func(r *LicenseGetTrialStatusRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        r.Header.Set("X-Opaque-Id", s)
    }
}

// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
//
// Code generated from specification version 7.17.10: DO NOT EDIT

package esapi

import (
    "context"
    "net/http"
    "strconv"
    "strings"
)

func newLicensePostStartBasicFunc(t Transport) LicensePostStartBasic
{
    return func(o ...func(*LicensePostStartBasicRequest)) (*Response, error) {

```

```

var r = LicensePostStartBasicRequest{}
for _, f := range o {
    f(&r)
}
return r.Do(r.ctx, t)
}
}

// ----- API Definition -----

// LicensePostStartBasic - Starts an indefinite basic license.
//
// See full documentation at https://www.elastic.co/guide/en/elasticsearch/reference/master/start-basic.html.
type LicensePostStartBasic func(o ...func(*LicensePostStartBasicRequest)) (*Response, error)

// LicensePostStartBasicRequest configures the License Post Start Basic API request.
type LicensePostStartBasicRequest struct {
    Acknowledge *bool

    Pretty    bool
    Human     bool
    ErrorTrace bool
    FilterPath []string

    Header http.Header

    ctx context.Context
}

// Do executes the request and returns response or error.
func (r LicensePostStartBasicRequest) Do(ctx context.Context, transport Transport) (*Response, error) {
    var (
        method string
        path    strings.Builder
        params  map[string]string
    )

    method = "POST"

    path.Grow(len("/_license/start_basic"))
    path.WriteString("/_license/start_basic")

    params = make(map[string]string)

    if r.Acknowledge != nil {
        params["acknowledge"] = strconv.FormatBool(*r.Acknowledge)
    }

```

```

if r.Pretty {
    params["pretty"] = "true"
}

if r.Human {
    params["human"] = "true"
}

if r.ErrorTrace {
    params["error_trace"] = "true"
}

if len(r.FilterPath) > 0 {
    params["filter_path"] = strings.Join(r.FilterPath, ",")
}

req, err := newRequest(method, path.String(), nil)
if err != nil {
    return nil, err
}

if len(params) > 0 {
    q := req.URL.Query()
    for k, v := range params {
        q.Set(k, v)
    }
    req.URL.RawQuery = q.Encode()
}

if len(r.Header) > 0 {
    if len(req.Header) == 0 {
        req.Header = r.Header
    } else {
        for k, vv := range r.Header {
            for _, v := range vv {
                req.Header.Add(k, v)
            }
        }
    }
}

if ctx != nil {
    req = req.WithContext(ctx)
}

res, err := transport.Perform(req)
if

```

```

err != nil {
    return nil, err
}

response := Response{
    StatusCode: res.StatusCode,
    Body:      res.Body,
    Header:    res.Header,
}

return &response, nil
}

// WithContext sets the request context.
func (f LicensePostStartBasic) WithContext(v context.Context) func(*LicensePostStartBasicRequest) {
    return func(r *LicensePostStartBasicRequest) {
        r.ctx = v
    }
}

// WithAcknowledge - whether the user has acknowledged acknowledge messages (default: false).
func (f LicensePostStartBasic) WithAcknowledge(v bool) func(*LicensePostStartBasicRequest) {
    return func(r *LicensePostStartBasicRequest) {
        r.Acknowledge = &v
    }
}

// WithPretty makes the response body pretty-printed.
func (f LicensePostStartBasic) WithPretty() func(*LicensePostStartBasicRequest) {
    return func(r *LicensePostStartBasicRequest) {
        r.Pretty = true
    }
}

// WithHuman makes statistical values human-readable.
func (f LicensePostStartBasic) WithHuman() func(*LicensePostStartBasicRequest) {
    return
    func(r *LicensePostStartBasicRequest) {
        r.Human = true
    }
}

// WithErrorTrace includes the stack trace for errors in the response body.
func (f LicensePostStartBasic) WithErrorTrace() func(*LicensePostStartBasicRequest) {
    return func(r *LicensePostStartBasicRequest) {
        r.ErrorTrace = true
    }
}

```



```

// WithFilterPath filters the properties of the response body.
func (f LicensePostStartBasic) WithFilterPath(v ...string) func(*LicensePostStartBasicRequest) {
    return func(r *LicensePostStartBasicRequest) {
        r.FilterPath = v
    }
}

// WithHeader adds the headers to the HTTP request.
func (f LicensePostStartBasic) WithHeader(h map[string]string) func(*LicensePostStartBasicRequest) {
    return func(r *LicensePostStartBasicRequest) {
        if r.Header == nil {
            r.Header = make(http.Header)
        }
        for k, v := range h {
            r.Header.Add(k, v)
        }
    }
}

// WithOpaqueID adds the X-Opaque-Id header to the HTTP request.
func (f LicensePostStartBasic) WithOpaqueID(s string) func(*LicensePostStartBasicRequest) {
    {
        return func(r *LicensePostStartBasicRequest) {
            if r.Header == nil {
                r.Header = make(http.Header)
            }
            r.Header.Set("X-Opaque-Id", s)
        }
    }
}

```

1.58 ssl-client 1.37.0-r19

1.58.1 Available under license :

No license file was found, but licenses were detected in source scan.

+ * Licensed under GPLv2 or later, see file LICENSE in this source tree.

Found in path(s):

* /aports-3-22-stable-main-busybox-zip/aports-3.22-stable-main-busybox/main/busybox/0024-umount-Implement-O-option-to-unmount-by-mount-option.patch

No license file was found, but licenses were detected in source scan.

/*

* depmod - generate modules.dep

* Copyright (c) 2008 Bernhard Reutner-Fischer

- * Copyrihgt (c) 2008 Timo Teras <timo.teras@iki.fi>

```

+ * Copyrihgt (c) 2008-2015 Timo Teras <timo.teras@iki.fi>
+ * Copyright (c) 2008 Vladimir Dronnikov
+
+ * Licensed under GPLv2 or later, see file LICENSE in this source tree.
@@ -25,6 +25,24 @@
#include "modutils.h"
#include <sys/utsname.h> /* uname() */

#define INDEX_MINCHAR 32
#define INDEX_MAXCHAR 128
+
+typedef struct index_node {
+ char *prefix;
+ llist_t *values;
+ struct index_node *children[INDEX_MAXCHAR-INDEX_MINCHAR];
+} index_node;
+
+struct globals {
+ module_db db;
+ index_node *root_node;
+} FIX_ALIASING;
#define G (*ptr_to_globals)
#define INIT_G() do { \
+ SET_PTR_TO_GLOBALS(xzalloc(sizeof(G))); \
+} while (0)
+
+/*
+ * Theory of operation:
+ * - iterate over all modules and record their full path
@@ -53,18 +71,12 @@ static int FAST_FUNC parse_module(struct recursive_state
+ *state,

for (ptr = image; ptr < image + len - 10; ptr++) {
if (is_prefixed_with(ptr, "depends=")) {
- char *u;
-
ptr += 8;
- for (u = ptr; *u; u++)
- if (*u == '-')
- *u = '_';
- ptr += string_to_llist(ptr, &e->deps, ",");
+ string_to_llist(replace_underscores(ptr), &e->deps, ",");
} else if (ENABLE_FEATURE_MODUTILS_ALIAS
&& is_prefixed_with(ptr, "alias=")
) {
- llist_add_to(&e->aliases, xstrdup(ptr + 6));
- ptr += strlen(ptr);
+ llist_add_to(&e->aliases, replace_underscores(xstrdup(ptr + 6)));

```

```

} else if (ENABLE_FEATURE_MODUTILS_SYMBOLS
&& is_prefixed_with(ptr, "__ksymtab_")
) {
@@ -74,9 +86,10 @@ static int FAST_FUNC parse_module(struct recursive_state *state,
) {
continue;
}
- llist_add_to(&e->symbols, xstrdup(ptr));
- ptr += strlen(ptr);
- }
+ llist_add_to(&e->symbols, xasprintf("symbol:%s", ptr));
+ } else
+ continue;
+ ptr += strlen(ptr);
}
free(image);

@@ -108,12 +121,6 @@ static void order_dep_list(module_db *modules, module_entry *start, llist_t
*add
}
}

```

```

-static void xfreopen_write(const char *file, FILE *f)
-{
- if (freopen(file, "w", f) == NULL)
- bb_perror_msg_and_die("can't open '%s'", file);
-}
-
//usage:#if !ENABLE_MODPROBE_SMALL
//usage:#define depmod_trivial_usage "[-n] [-b BASE] [VERSION] [MODFILES]..."
//usage:#define depmod_full_usage "\n\n"
@@ -167,6 +174,169 @@ enum {
OPT_C = (1 << 9), /* -C,--config etc_modules_conf: ignored */

```

Found in path(s):

* /aports-3-22-stable-main-busybox-zip/aports-3.22-stable-main-busybox/main/busybox/0018-depmod-support-generating-kmod-binary-index-files.patch

No license file was found, but licenses were detected in source scan.

```

/*
* Copyright (C) 2008 Natanael Copa <natanael.copa@gmail.com>
* All rights reserved.
*
* This program is free software; you can redistribute it and/or modify it
* under the terms of the GNU General Public License version 2 as published
* by the Free Software Foundation. See http://www.gnu.org/ for details.
*
*/

```

Found in path(s):

* /aports-3-22-stable-main-busybox-zip/aports-3.22-stable-main-busybox/main/busybox/bbsuid.c

1.59 libapk2 2.14.9-r3

1.59.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom

to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for

each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you

may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works.

But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section

2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it

with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all

the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program

(or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions.

You may

not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of

this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain

countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this

License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE

THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and an idea of what it does.>

Copyright (C)< yyyy> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon >, 1 April 1989 Ty Coon, President of Vice This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

1.60 clientmodel 0.6.2

1.60.1 Available under license :

Data model artifacts for Prometheus.

Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.61 skalibs-libs 2.14.4.0-r0

1.61.1 Available under license :

Copyright (c) 2011-2025 Laurent Bercot <ska-skaware@skarnet.org>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.62 musl-utils 1.2.5-r10

1.62.1 Available under license :

musl as a whole is licensed under the following standard MIT license:

Copyright 2005-2020 Rich Felker, et al.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Authors/contributors include:

A. Wilcox
Ada Worcester
Alex Dowad
Alex Suykov
Alexander Monakov
Andre McCurdy
Andrew Kelley
Anthony G. Basile

Aric Belsito
Arvid Picciani
Bartosz Brachaczek
Benjamin Peterson
Bobby Bingham
Boris Brezillon
Brent Cook
Chris Spiegel
Clment Vasseur
Daniel Micay
Daniel Sabogal
Daurnimator
David Carlier
David Edelsohn
Denys Vlasenko
Dmitry Ivanov
Dmitry V. Levin
Drew DeVault
Emil Renner Berthing
Fangrui Song
Felix Fietkau
Felix Janda
Gianluca Anzolin
Hauke Mehrrens
He X
Hiltjo Posthuma
Isaac Dunham
Jaydeep Patil
Jens Gustedt
Jeremy Huntwork
Jo-Philipp Wich
Joakim Sindholt
John Spencer
Julien Ramseier
Justin Cormack
Kaarle Ritvanen
Khem Raj
Kylie McClain
Leah Neukirchen
Luca Barbato
Luka
Perkov
M Farkas-Dyck (Strake)
Mahesh Bodapati
Markus Wichmann
Masanori Ogino
Michael Clark
Michael Forney

Mikhail Kremnyov
Natanael Copa
Nicholas J. Kain
orc
Pascal Cuoq
Patrick Oppenlander
Petr Hosek
Petr Skocik
Pierre Carrier
Reini Urban
Rich Felker
Richard Pennington
Ryan Fairfax
Samuel Holland
Segev Finer
Shiz
sin
Solar Designer
Stefan Kristiansson
Stefan O'Rear
Szabolcs Nagy
Timo Ters
Trutz Behn
Valentin Ochs
Will Dietz
William Haddon
William Pitcock

Portions of this software are derived from third-party works licensed under terms compatible with the above MIT license:

The TRE regular expression implementation (`src/regex/reg*` and `src/regex/tre*`) is Copyright 2001-2008 Ville Laurikari and licensed under a 2-clause BSD license (license text in the source files). The included version has been heavily modified by Rich Felker in 2012, in the interests of size, simplicity, and namespace cleanliness.

Much of the math library code (`src/math/*` and `src/complex/*`) is Copyright 1993,2004 Sun Microsystems or Copyright 2003-2011 David Schultz or Copyright 2003-2009 Steven G. Kargl or Copyright 2003-2009 Bruce D. Evans or Copyright 2008 Stephen L. Moshier or Copyright 2017-2018 Arm Limited and labelled as such in comments in the individual source files. All have been licensed under extremely permissive terms.

The ARM memcpy code (src/string/arm/memcpy.S) is Copyright 2008 The Android Open Source Project and is licensed under a two-clause BSD license. It was taken from Bionic libc, used on Android.

The AArch64 memcpy and memset code (src/string/aarch64/*) are Copyright 1999-2019, Arm Limited.

The implementation of DES for crypt (src/crypt/crypt_des.c) is Copyright 1994 David Burren. It is licensed under a BSD license.

The implementation of blowfish crypt (src/crypt/crypt_blowfish.c) was originally written by Solar Designer and placed into the public domain. The code also comes with a fallback permissive license for use in jurisdictions that may not recognize the public domain.

The smoothsort implementation (src/stdlib/qsort.c) is Copyright 2011 Valentin Ochs and is licensed under an MIT-style license.

The x86_64 port was written by Nicholas J. Kain and is licensed under the standard MIT terms.

The mips and microblaze ports were originally written by Richard Pennington for use in the elcc project. The original code was adapted by Rich Felker for build system and code conventions during upstream integration. It is licensed under the standard MIT terms.

The mips64 port was contributed by Imagination Technologies and is licensed under the standard MIT terms.

The powerpc port was also originally written by Richard Pennington, and later supplemented and integrated by John Spencer. It is licensed under the standard MIT terms.

All other files which have no copyright comments are original works produced specifically for use as part of this library, written either by Rich Felker, the main author of the library, or by one or more contributors listed above. Details on authorship of individual files can be found in the git version control history of the project. The omission of copyright and license comments in each file is in the interest of source tree size.

In addition, permission is hereby granted for all public header files (include/* and arch/*/bits/*) and crt files intended to be linked into applications (crt/*, ldso/dlstart.c, and arch/*/crt_arch.h) to omit the copyright notice and permission notice otherwise required by the license, and to use these files without any requirement of

attribution. These files include substantial contributions from:

Bobby Bingham
John Spencer
Nicholas J. Kain
Rich Felker
Richard Pennington
Stefan Kristiansson
Szabolcs Nagy

all of whom have explicitly granted such permission.

This file previously contained text expressing a belief that most of the files covered by the above exception were sufficiently trivial not to be subject to copyright, resulting in confusion over whether it negated the permissions granted in the license. In the spirit of permissive licensing, and of not having licensing issues being an obstacle to adoption, that text has been removed.

1.63 github.com/go-logr/stdr 1.2.2

1.63.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.64 sirupsen-logrus 1.9.3

1.64.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Simon Eskildsen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.65 stdr 1.2.2

1.65.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.66 cJSON 1.7.19-r0

1.66.1 Notifications :

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit
(<http://www.openssl.org/>)

This product includes cryptographic software written by Eric Young (ey@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

1.66.2 Available under license :

Apache License 2.0

Contributors

=====

Original Author:

- [Dave Gamble](<https://github.com/DaveGamble>)

Current Maintainer:

- [Max Bruckner](<https://github.com/FSMaxB>)

- [Alan Wang](<https://github.com/Alanscut>)

Contributors:

- * [Ajay Bhargav](<https://github.com/ajaybhargav>)
- * [Alexander Vasiljev](<https://github.com/AlexanderVasiljev>)
- * [Alper Akcan](<https://github.com/alperakcan>)
- * [Andrew Tang](<https://github.com/singku>)
- * [Andy](<https://github.com/mlh0101>)
- * [Anton Sergeev](<https://github.com/anton-sergeev>)
- * [Benbuck Nason](<https://github.com/bnason-nf>)
- * [Bernt Johan Damslara](<https://github.com/bjda>)
- * [Bob Kocisko](<https://github.com/bobkocisko>)
- * [Christian Schulze](<https://github.com/ChristianSch>)
- * [Casperinous](<https://github.com/Casperinous>)
- * [ChenYuan](<https://github.com/zjuchenyuan>)
- * [Debora Grosse](<https://github.com/DeboraG>)
- * [dieyushi](<https://github.com/dieyushi>)
- * [Dngwn Hung ()](<https://github.com/DongwenHuang>)
- * [Dominik](<https://github.com/DL6ER>)
- * [Donough Liu](<https://github.com/ldm0>)
- *
- [Erez Oxman](<https://github.com/erez-o>)
- * Eswar Yaganti
- * [Evan Todd](<https://github.com/etodd>)
- * [Fabrice Fontaine](<https://github.com/ffontaine>)
- * Ian Mobley
- * Irwan Djadjadi
- * [hopper-vul](<https://github.com/hopper-vul>)
- * [HuKeping](<https://github.com/HuKeping>)
- * [IvanVoid](<https://github.com/npi3pak>)

- * [Jakub Wilk](https://github.com/jwilk)
- * [Jiri Zouhar](https://github.com/loigu)
- * [Jonathan Fether](https://github.com/jfether)
- * [Joshua Arulsamy](https://github.com/jarulsamy)
- * [Julian Ste](https://github.com/julian-st)
- * [Julin Vsquez](https://github.com/juvasquezg)
- * [Junbo Zheng](https://github.com/Junbo-Zheng)
- * [Kevin Branigan](https://github.com/kbranigan)
- * [Kevin Sapper](https://github.com/sappo)
- * [Kyle Chisholm](https://github.com/ChisholmKyle)
- * [Linus Wallgren](https://github.com/ecksun)
- * [Luo Jin](https://github.com/Up-wind)
- * [Max](https://github.com/maebex)
- * [MaxBrandtner](https://github.com/MaxBrandtner)
- * [Mateusz Szafoni](https://github.com/raiden00pl)
- *
- Mike Pontillo
- * [miaoerduo](https://github.com/miaoerduo)
- * [mohawk2](https://github.com/mohawk2)
- * [Mike Jerris](https://github.com/mjerris)
- * [Mike Robinson](https://github.com/mhrobinson)
- * [Moorthy](https://github.com/moorthy-bs)
- * [myd7349](https://github.com/myd7349)
- * [NancyLi1013](https://github.com/NancyLi1013)
- * [Orri](https://github.com/sbvoxel)
- * Paulo Antonio Alvarez
- * [Pawe Malowany](https://github.com/PawelMalowany)
- * [Pawel Winogrodzki](https://github.com/PawelWMS)
- * [prefetchnta](https://github.com/prefetchnta)
- * [Rafael Leal Dias](https://github.com/rafaeldias)
- * [Randy](https://github.com/randy408)
- * [raiden00pl](https://github.com/raiden00pl)
- * [Robin Mallinson](https://github.com/rmallins)
- * [Rod Vagg](https://github.com/rvagg)
- * [Roland Meertens](https://github.com/rmeertens)
- * [Romain Porte](https://github.com/MicroJoe)
- * [SANJEEV BA](https://github.com/basanjeev)
- * [Sang-Heon Jeon](https://github.com/Intuition)
- * [Sayan Bandyopadhyay](https://github.com/saynb)
- *
- [Simon Sobisch](https://github.com/GitMensch)
- * [Simon Ricaldone](https://github.com/simon-p-r)
- * [Stoian Ivanov](https://github.com/sdrshr)
- * [SuperH-0630](https://github.com/SuperH-0630)
- * [Square789](https://github.com/Square789)
- * [Stephan Gatzka](https://github.com/gatzka)
- * [Tony Langhammer](https://github.com/BigBrainAFK)
- * [Vemake](https://github.com/vemakereporter)

- * [vwvw](https://github.com/vwvw)
- * [warmsocks](https://github.com/warmsocks)
- * [Wei Tan](https://github.com/tan-wei)
- * [Weston Schmidt](https://github.com/schmidtw)
- * [xiaomianhehe](https://github.com/xiaomianhehe)
- * [yangfl](https://github.com/yangfl)
- * [yuta-oxo](https://github.com/yuta-oxo)
- * [Zach Hindes](https://github.com/zhindes)
- * [Zhao Zhixu](https://github.com/zhaozhixu)
- * [10km](https://github.com/10km)

And probably more people on [SourceForge](https://sourceforge.net/p/cjson/bugs/search/?q=status%3Aclosed-rejected+or+status%3Aclosed-out-of-date+or+status%3Awont-fix+or+status%3Aclosed-fixed+or+status%3Aclosed&page=0)

Also

thanks to all the people who reported bugs and suggested new features.

Copyright (c) 2009-2017 Dave Gamble and cJSON contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF

OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) <year> 2007-14 Mike Karlesky, Mark VanderVoord, Greg Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.67 logr 1.4.3

1.67.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.68 prometheus-client-model 0.6.2

1.68.1 Available under license :

Data model artifacts for Prometheus.

Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices

stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.69 prometheus-client 1.23.2

1.69.1 Available under license :

Copyright (c) 2013 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
- * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Prometheus instrumentation library for Go applications

Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

The following components are included in this product:

perks - a fork of <https://github.com/bmizerany/perks>

<https://github.com/beorn7/perks>

Copyright 2013-2015 Blake Mizerany, Bjrn Rabenstein

See <https://github.com/beorn7/perks/blob/master/README.md> for license details.

Go support for Protocol Buffers - Google's data interchange format

<http://github.com/golang/protobuf/>

Copyright 2010 The Go Authors

See source code for license details.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.70 xxhash 2.3.0

1.70.1 Available under license :

Copyright (c) 2016 Caleb Spare

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.71 procs 0.19.2

1.71.1 Available under license :

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

procfs provides functions to retrieve system, kernel and process metrics from the pseudo-filesystem proc.

Copyright 2014-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

1.72 stdlib go1.25.4

1.72.1 Available under license :

```
// Copyright 2024 The Go Authors. All rights reserved.  
// Use of this source code is governed by a BSD-style  
// license that can be found in the LICENSE file.
```

```
package copyright
```

```
import (  
    "bytes"  
    "internal/testenv"  
    "io"  
    "io/fs"  
    "os"  
    "path/filepath"  
    "testing"  
)
```

```
var copyright = []byte("Copyright")
```

```
var permitted = [][]byte{  
    []byte("// Code generated by "),  
    []byte("// Code generated from "),  
    []byte("// Created by cgo -cdefs"),  
    []byte("// DO NOT EDIT\n// generated by:"),  
    []byte("// Empty assembly file"),  
    []byte("// Generated using cgo"),  
    []byte("// Original source:\n\nhttp://www.zorinaq.com/papers/md5-amd64.html"), // public domain crypto/md5  
    []byte("// created by cgo -cdefs"),  
    []byte("// go run mkasm.go"),  
    []byte("// mkerrors"),  
    []byte("// mksys"),  
    []byte("// run\n// Code generated by"), // cmd/compile/internal/test/constFold_test.go  
}
```

```
func TestCopyright(t *testing.T) {  
    buf := make([]byte, 2048)  
    filepath.WalkDir(filepath.Join(testenv.GOROOT(t), "src"), func(path string,  
        d fs.DirEntry, err error) error {  
        if d.IsDir() && (d.Name() == "testdata" || d.Name() == "vendor") {  
            return filepath.SkipDir  
        }  
        switch filepath.Ext(d.Name()) {  
        default:  
            return nil  
        case ".s", ".go":  
            // check  
        }  
    })  
}
```

```

f, err := os.Open(path)
if err != nil {
    t.Error(err)
    return nil
}
defer f.Close()
n, err := f.Read(buf)
if err != nil && err != io.EOF {
    t.Error(err)
    return nil
}
b := buf[:n]
if bytes.Contains(b, copyright) {
    return nil
}
for _, ok := range permitted {
    if bytes.HasPrefix(b, ok) {
        return nil
    }
}
t.Errorf("%s: missing copyright notice", path)
return nil
}))
}

```

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The Go source code and supporting files in this directory are covered by the usual Go license (see ../../../../LICENSE).

When building with GOEXPERIMENT=boringcrypto, the following applies.

The goboringcrypto_linux_amd64.syso object file is built from BoringSSL source code by build/build.sh and is covered by the BoringSSL license reproduced below and also at <https://boringssl.googlesource.com/boringssl/+fips-20220613/LICENSE>.

BoringSSL is a fork of OpenSSL. As such, large parts of it fall under OpenSSL licensing. Files that are completely new have a Google copyright and an ISC license. This license is reproduced at the bottom of this file.

Contributors to BoringSSL are required to follow the CLA rules for Chromium: <https://cla.developers.google.com/clas>

Some files from Intel are under yet another license, which is also included underneath.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit.

See below

for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

The following are Google-internal bug numbers where explicit permission from some authors is recorded for use of their work. (This is purely for our own record keeping.)

27287199

27287880

27287883

OpenSSL License

/* =====

* Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

*

- * 1. Redistributions of source code must retain the above copyright
- * notice, this list of conditions and the following disclaimer.
- *
- * 2. Redistributions in binary form must reproduce the above copyright
- * notice, this list of conditions and the following
- disclaimer in
- * the documentation and/or other materials provided with the
- * distribution.
- *
- * 3. All advertising materials mentioning features or use of this
- * software must display the following acknowledgment:
- * "This product includes software developed by the OpenSSL Project
- * for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
- *
- * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
- * endorse or promote products derived from this software without
- * prior written permission. For written permission, please contact
- * openssl-core@openssl.org.
- *
- * 5. Products derived from this software may not be called "OpenSSL"
- * nor may "OpenSSL" appear in their names without prior written
- * permission of the OpenSSL Project.
- *
- * 6. Redistributions of any form whatsoever must retain the following
- * acknowledgment:
- * "This product includes software developed by the OpenSSL Project
- * for use in the OpenSSL Toolkit
- (<http://www.openssl.org/>)"
- *
- * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY
- * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
- * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
- * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
- * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
- * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
- * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
- * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
- * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
- * OF THE POSSIBILITY OF SUCH DAMAGE.
- * =====
- *
- * This product includes cryptographic software written by Eric Young
- * (ey@cryptsoft.com). This product
- includes software written by Tim
- * Hudson (tjh@cryptsoft.com).
- *

*/

Original SSLeay License

```
/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of
 * the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 * must display the following acknowledgement:
 * "This product includes cryptographic software written by
 * Eric Young (eay@cryptsoft.com)"
 * The word 'cryptographic' can be left out if the rouines from the library
 * being used are
 * not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 * the apps directory (application code) you must include an acknowledgement:
 * "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
```

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed. i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 */

ISC license used for completely new code in BoringSSL:

/* Copyright (c) 2015, Google Inc.
 *
 * Permission to use, copy, modify, and/or distribute this software for any
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
 * WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
 * SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
 * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
 * OF CONTRACT, NEGLIGENCE
 OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
 * CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. */

Some files from Intel carry the following license:

Copyright (c) 2012, Intel Corporation
 #
 # All rights reserved.
 #
 # Redistribution and use in source and binary forms, with or without
 # modification, are permitted provided that the following conditions are
 # met:
 #
 # * Redistributions of source code must retain the above copyright
 # notice, this list of conditions and the following disclaimer.

```
#
# * Redistributions in binary form must reproduce the above copyright
# notice, this list of conditions and the following disclaimer in the
# documentation and/or other materials provided with the
# distribution.
#
# * Neither the name of the Intel Corporation nor the names of its
# contributors may be used to endorse or promote products derived from
# this software without specific prior written permission.
#
#
# THIS SOFTWARE IS PROVIDED BY INTEL CORPORATION
# ""AS IS"" AND ANY
# EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
# IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
# PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INTEL CORPORATION OR
# CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
# EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
# PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
# PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
# People who have agreed to one of the CLAs and can contribute patches.
# The AUTHORS file lists the copyright holders; this file
# lists people. For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
# https://developers.google.com/open-source/cla/individual
# https://developers.google.com/open-source/cla/corporate
#
# Names should be added to this file as:
# Name <email address>
Raul Silvera <rsilvera@google.com>
Tipp Moseley <tipp@google.com>
Hyoun Kyu Cho <netforce@google.com>
Martin Spier <spiermar@gmail.com>
Taco de Wolff <tacodewolff@gmail.com>
Andrew Hunter <andrewhhunter@gmail.com>
Copyright (c) 2009 The Go Authors. All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above

copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2015 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the

distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2015 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2009-2017 Andrea Leofreddi <a.leofreddi@vleo.net>. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are

permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of Andrea Leofreddi.

1.73 golang 1.25.4

1.73.1 Available under license :

```
// Copyright 2024 The Go Authors. All rights reserved.  
// Use of this source code is governed by a BSD-style  
// license that can be found in the LICENSE file.
```

```
package copyright
```

```
import (  
    "bytes"  
    "internal/testenv"  
    "io"  
    "io/fs"  
    "os"  
    "path/filepath"
```

```

"testing"
)

var copyright = []byte("Copyright")

var permitted = [][]byte{
    []byte("// Code generated by "),
    []byte("// Code generated from "),
    []byte("// Created by cgo -cdefs"),
    []byte("// DO NOT EDIT\n// generated by:"),
    []byte("// Empty assembly file"),
    []byte("// Generated using cgo"),
    []byte("// Original source:\n/\thttp://www.zorinaq.com/papers/md5-amd64.html"), // public domain crypto/md5
    []byte("// created by cgo -cdefs"),
    []byte("// go run mkasm.go"),
    []byte("// mkerrors"),
    []byte("// mksys"),
    []byte("// run\n// Code generated by"), // cmd/compile/internal/test/constFold_test.go
}

func TestCopyright(t *testing.T) {
    buf := make([]byte, 2048)
    filepath.WalkDir(filepath.Join(testenv.GOROOT(t), "src"), func(path string,
    d fs.DirEntry, err error) error {
        if d.IsDir() && (d.Name() == "testdata" || d.Name() == "vendor") {
            return filepath.SkipDir
        }
        switch filepath.Ext(d.Name()) {
        default:
            return nil
        case ".s", ".go":
            // check
        }

        f, err := os.Open(path)
        if err != nil {
            t.Error(err)
            return nil
        }
        defer f.Close()
        n, err := f.Read(buf)
        if err != nil && err != io.EOF {
            t.Error(err)
            return nil
        }
        b := buf[:n]
        if bytes.Contains(b, copyright) {
            return nil
        }
    }
}

```

```

}
for _, ok := range permitted {
    if bytes.HasPrefix(b, ok) {
        return nil
    }
}
t.Errorf("%s: missing copyright notice", path)
return nil
})
}

```

Copyright 2009 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The Go source code and supporting files in this directory are covered by the usual Go license (see ../../../../LICENSE).

When building with GOEXPERIMENT=boringcrypto, the following applies.

The goboringcrypto_linux_amd64.syso object file is built from BoringSSL source code by build/build.sh and is covered by the BoringSSL license reproduced below and also at <https://boringssl.googlesource.com/boringssl/+fips-20220613/LICENSE>.

BoringSSL is a fork of OpenSSL. As such, large parts of it fall under OpenSSL licensing. Files that are completely new have a Google copyright and an ISC license. This license is reproduced at the bottom of this file.

Contributors to BoringSSL are required to follow the CLA rules for Chromium:
<https://cla.developers.google.com/clas>

Some files from Intel are under yet another license, which is also included underneath.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit.

See below

for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

The following are Google-internal bug numbers where explicit permission from some authors is recorded for use of their work. (This is purely for our own record keeping.)

27287199

27287880

27287883

OpenSSL License

```
/* =====
* Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following
disclaimer in
* the documentation and/or other materials provided with the
* distribution.
*
* 3. All advertising materials mentioning features or use of this
* software must display the following acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
```

- * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
- * endorse or promote products derived from this software without
- * prior written permission. For written permission, please contact
- * openssl-core@openssl.org.
- *
- * 5. Products derived from this software may not be called "OpenSSL"
- * nor may "OpenSSL" appear in their names without prior written
- * permission of the OpenSSL Project.
- *
- * 6. Redistributions of any form whatsoever must retain the following
- * acknowledgment:
- * "This product includes software developed by the OpenSSL Project
- * for use in the OpenSSL Toolkit
- (<http://www.openssl.org/>)"
- *
- * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY
- * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
- * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
- * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
- * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
- * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
- * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
- * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
- * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
- * OF THE POSSIBILITY OF SUCH DAMAGE.
- * =====
- *
- * This product includes cryptographic software written by Eric Young
- * (eay@cryptsoft.com). This product
- includes software written by Tim
- * Hudson (tjh@cryptsoft.com).
- *
- */

Original SSLeay License

```
/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
```

- * apply to all code found in this distribution, be it the RC4, RSA,
- * lhash, DES, etc., code; not just the SSL code. The SSL documentation
- * included with this distribution is covered by the same copyright terms
- * except that the holder is Tim Hudson (tjh@cryptsoft.com).
- *
- * Copyright remains Eric Young's, and as such any Copyright notices in
- * the code are not to be removed.
- * If this package is used in a product, Eric Young should be given attribution
- * as the author of
- the parts of the library used.
- * This can be in the form of a textual message at program startup or
- * in documentation (online or textual) provided with the package.
- *
- * Redistribution and use in source and binary forms, with or without
- * modification, are permitted provided that the following conditions
- * are met:
- * 1. Redistributions of source code must retain the copyright
- * notice, this list of conditions and the following disclaimer.
- * 2. Redistributions in binary form must reproduce the above copyright
- * notice, this list of conditions and the following disclaimer in the
- * documentation and/or other materials provided with the distribution.
- * 3. All advertising materials mentioning features or use of this software
- * must display the following acknowledgement:
- * "This product includes cryptographic software written by
- * Eric Young (eay@cryptsoft.com)"
- * The word 'cryptographic' can be left out if the routines from the library
- * being used are
- not cryptographic related :-).
- * 4. If you include any Windows specific code (or a derivative thereof) from
- * the apps directory (application code) you must include an acknowledgement:
- * "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
- *
- * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND
- * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
- * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
- * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
- * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
- * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
- * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
- * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
- * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
- * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
- POSSIBILITY OF
- * SUCH DAMAGE.
- *
- * The licence and distribution terms for any publically available version or
- * derivative of this code cannot be changed. i.e. this code cannot simply be
- * copied and put under another distribution licence

* [including the GNU Public Licence.]

*/

ISC license used for completely new code in BoringSSL:

/* Copyright (c) 2015, Google Inc.

*

* Permission to use, copy, modify, and/or distribute this software for any

* purpose with or without fee is hereby granted, provided that the above

* copyright notice and this permission notice appear in all copies.

*

* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES

* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF

* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY

* SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES

* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION

* OF CONTRACT, NEGLIGENCE

OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN

* CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. */

Some files from Intel carry the following license:

Copyright (c) 2012, Intel Corporation

#

All rights reserved.

#

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are

met:

#

* Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.

#

* Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the

distribution.

#

* Neither the name of the Intel Corporation nor the names of its

contributors may be used to endorse or promote products derived from

this software without specific prior written permission.

#

#

THIS SOFTWARE IS PROVIDED BY INTEL CORPORATION

""AS IS"" AND ANY

EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INTEL CORPORATION OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
People who have agreed to one of the CLAs and can contribute patches.
The AUTHORS file lists the copyright holders; this file
lists people. For example, Google employees are listed here
but not in AUTHORS, because Google holds the copyright.

<https://developers.google.com/open-source/cla/individual>
<https://developers.google.com/open-source/cla/corporate>

Names should be added to this file as:
Name <email address>
Raul Silvera <rsilvera@google.com>
Tipp Moseley <tipp@google.com>
Hyouun Kyu Cho <netforce@google.com>
Martin Spier <spiermar@gmail.com>
Taco de Wolff <tacodewolff@gmail.com>
Andrew Hunter <andrewhhunter@gmail.com>
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
- * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2015 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2015 The Go Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2009-2017 Andrea Leofreddi <a.leofreddi@vleo.net>. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY
AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF
ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the
authors and should not be interpreted as representing official policies, either expressed
or implied, of Andrea Leofreddi.

1.74 x-sys 0.38.0

1.74.1 Available under license :

This source code was written by the Go contributors.
The master list of contributors is in the main Go distribution,
visible at <http://tip.golang.org/CONTRIBUTORS>.
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
- * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.75 common 0.67.3

1.75.1 Available under license :

Common libraries shared by Prometheus Go components.
Copyright 2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.76 prometheus-common 0.67.3

1.76.1 Available under license :

Common libraries shared by Prometheus Go components.
Copyright 2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (<http://soundcloud.com/>).

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
- You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.77 github.com/elastic/elastic-transport-go/v8 8.8.0

1.77.1 Available under license :

```
name: License headers
on: [pull_request]
jobs:
  license-check:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Check license headers
        run: ./github/check-license-headers.sh
// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
//  http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
Elastic Transport Go
Copyright 2014-2021 Elasticsearch BV
```

This product includes software developed by The Apache Software
Foundation (<http://www.apache.org/>).

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices
stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.
- You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a

file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.78 elastic-transport-go 8.8.0

1.78.1 Available under license :

```
name: License headers
on: [pull_request]
jobs:
  license-check:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Check license headers
        run: ./github/check-license-headers.sh
// Licensed to Elasticsearch B.V. under one or more contributor
// license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright
// ownership. Elasticsearch B.V. licenses this file to you under
// the Apache License, Version 2.0 (the "License"); you may
// not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing,
// software distributed under the License is distributed on an
// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
// KIND, either express or implied. See the License for the
// specific language governing permissions and limitations
// under the License.
```

This product includes software developed by The Apache Software Foundation (<http://www.apache.org/>).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not

pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

©2025 Cisco Systems, Inc. All rights reserved.