



メモリのトラブルシューティング

- メモリのトラブルシューティングに関する詳細情報 (1 ページ)
- プラットフォーム メモリ使用率の一般/高レベルの評価 (2 ページ)
- プラットフォームのメモリ使用率の詳細な評価 (3 ページ)
- ユーザ プロセス (7 ページ)
- 組み込みプラットフォームのメモリモニタリング (10 ページ)
- LPSS 共有メモリ監視 (12 ページ)

メモリのトラブルシューティングに関する詳細情報

ダイナミックランダムアクセスメモリ (DRAM) は、すべてのプラットフォームで限られたリソースであり、使用率がチェックされるように制御またはモニタする必要があります。

Cisco NX-OS は、次の 3 つの方法でメモリを使用します。

- **Page cache** : 永続ストレージ (CompactFlash) からファイルにアクセスすると、カーネルはデータをページキャッシュに読み取ります。これは、将来データにアクセスするときに、ディスクストレージに関する遅いアクセス時間を回避できることを意味します。他のプロセスがメモリを必要とする場合、キャッシュされたページはカーネルによって解放されます。一部のファイルシステム (tmpfs) は、純粋にページキャッシュ内に存在します (たとえば、/dev/sh, /var/sysmgr, /var/tmp)。これは、このデータの永続的なストレージがなく、データが削除されたときを意味します。ページキャッシュからは復元できません。tmpfs-cached ファイルは、削除された場合にのみページキャッシュされたページを解放します。
- **Kernel** : カーネルには、独自のテキスト、データ、およびカーネルロード可能モジュール (KLM) を保存するためのメモリが必要です。KLM は、(個別のユーザプロセスではなく) カーネルにロードされるコードの一部です。カーネルメモリの使用例として、インバンドポートドライバがパケットを受信するためにメモリを割り当てる場合があります。
- **User processes** Cisco NX-OS : このメモリは、カーネルに統合されていない Linux プロセス (テキスト、スタック、ヒープなど) によって使用されます。

プラットフォームメモリ使用率の一般/高レベルの評価

高いメモリ使用率をトラブルシューティングする場合は、まず使用率の高いタイプ（プロセス、ページキャッシュ、またはカーネル）を判別する必要があります。使用率のタイプを特定したら、追加のトラブルシューティングコマンドを使用して、この動作の原因となっているコンポーネントを特定できます。

プラットフォームメモリ使用率の一般/高レベルの評価

次の2つの基本的なCLIコマンドを使用して、プラットフォームのメモリ使用率の全体的なレベルを評価できます。**show system resources**および**show processes memory**。



(注) これらのコマンド出力から、プラットフォームの使用率が通常/予想よりも高いことがわかりますが、どのタイプのメモリ使用率が高いかはわかりません。



(注) *show system resources* コマンドの出力に空きメモリの減少が示されている場合は、Linux カーネルキャッシングが原因である可能性があります。システムがより多くのメモリを必要とするたびに、Linux カーネルはキャッシングされたメモリを解放します。*show system internal kernel meminfo* コマンドは、システムのキャッシングメモリを表示します。

この項で説明している **show system resources** コマンドは、プラットフォームのメモリ統計情報を表示します。



(注) この出力は、/proc/meminfo の Linux メモリ統計情報から取得されます。

- **total** : プラットフォーム上の物理 RAM の量。
- **free** : 未使用または使用可能なメモリの量。
- **used** : 割り当てられた（永続的な）メモリとキャッシュされた（一時的な）メモリの量。

キャッシュとバッファは、カスタマー モニタリングには関係ありません。

この情報は、プラットフォームの使用率の一般的な表現のみを提供します。メモリ使用率が高い理由をトラブルシューティングするには、より多くの情報が必要です。

show processes memory コマンドは、プロセスごとのメモリ割り当てを表示します。

```
switch# show processes memory
Load average: 1 minute: 0.43 5 minutes: 0.30 15 minutes: 0.28
Processes : 884 total, 1 running
CPU states : 2.0% user, 1.5% kernel, 96.5% idle
PID MemAlloc MemLimit MemUsed StackBase/Ptr Process
---- -----
4662 52756480 562929945 150167552 bffffd00/bffffd970 netstack
```

プラットフォームのメモリ使用率の詳細な評価

show system internal memory-alerts-log コマンドを使用し、また **show system internal kernel** コマンドを使用して、Cisco NX-OS でメモリ使用率の詳細を表示します。

```
switch# show system internal kernel meminfo
MemTotal: 4135780 kB
MemFree: 578032 kB
Buffers: 5312 kB
Cached: 1926296 kB
RAMCached: 1803020 kB
Allowed: 1033945 Pages
Free: 144508 Pages
Available: 177993 Pages
SwapCached: 0 kB
Active: 1739400 kB
Inactive: 1637756 kB
HighTotal: 3287760 kB
HighFree: 640 kB
LowTotal: 848020 kB
LowFree: 577392 kB
SwapTotal: 0 kB
SwapFree: 0 kB
Dirty: 0 kB
Writeback: 0 kB
Mapped: 1903768 kB
Slab: 85392 kB
CommitLimit: 2067888 kB
Committed_AS: 3479912 kB
PageTables: 20860 kB
VmallocTotal: 131064 kB
VmallocUsed: 128216 kB
```

■ ページキャッシュ

VmallocChunk: 2772 kB

上記の出力で、最も重要なフィールドは次のとおりです。

- **MemTotal (kB)** : システムのメモリの総量。
- **Cached (kB)** : ページキャッシュ (tmpfsマウント内のファイルと永続ストレージ/bootflash からキャッシュされたデータを含む) で使用されるメモリの量。
- **RamCached (kB)** : 解放できないページキャッシュで使用されているメモリの量 (永続ストレージによってバックアップされていないデータ)。
- **Available (Pages)** : ページの空きメモリの量 (ページキャッシュと空きリストで使用可能にできる領域を含む)。
- **Mapped (Pages)** : ページテーブルにマッピングされたメモリ (非カーネルプロセスで使用されているデータ)。
- **Slab (Pages)** : カーネルメモリ消費量の大まかな指標。



(注) 1ページのメモリは4 kB のメモリに相当します。

この項で説明している **show system internal kernel memory global** コマンドは、ページキャッシュとカーネル/プロセスメモリのメモリ使用量を表示します。

```
switch# show system internal kernel memory global
Total memory in system : 4129600KB
Total Free memory : 1345232KB
Total memory in use : 2784368KB
Kernel/App memory : 1759856KB
RAM FS memory : 1018616KB
```



(注) Cisco NX-OS では、Linux カーネルが使用しているメモリの割合 (存在する RAM 全体に対する) をモニタし、使用率がデフォルトまたは設定されたしきい値を超えると、プラットフォームマネージャがアラートを生成します。アラートが発生した場合は、プラットフォームマネージャによってキャプチャされたログを現在の使用率と照らし合わせて確認すると便利です。

これらのコマンドの出力を確認すると、ページキャッシュ、メモリを保持しているプロセス、またはカーネルの結果として使用率が高いかどうかを判断できます。

ページキャッシュ

Cached 値または RAMCached 値が高い場合は、ファイルシステムの使用率を確認し、どの種類のファイルがページキャッシュを占有しているかを判断する必要があります。

この項で説明している **show system internal flash** コマンドは、ファイルシステムの使用率を表示します (出力はメモリアラートログに含まれる df-hT と同様です)。

```
switch# show system internal flash
Mount-on          1K-blocks    Used   Available  Use%  Filesystem
/                  409600     43008   367616    11   /dev/root
/proc              0          0        0         0    proc
/sys               0          0        0         0    none
/isan              409600    269312   140288    66   none
/var/tmp            307200    876      306324    1    none
/var/sysmgr        1048576   999424   49152     96   none
/var/sysmgr/ftp    307200    24576    282624    8    none
/dev/shm            1048576   412672   635904    40   none
/volatile          204800    0        204800    0    none
/debug              2048      16       2032     1    none
/dev/mqueue          0          0        0         0    none
/mnt/cfg/0          76099    5674     66496     8   /dev/hda5
/mnt/cfg/1          75605    5674     66027     8   /dev/hda6
/bootflash          1796768   629784   1075712   37   /dev/hda3
/var/sysmgr/startup-cfg 409600   27536    382064    7    none
/mnt/plog            56192    3064     53128     6   /dev/mtdblock2
/dev/pts              0          0        0         0    devpts
/mnt/pss             38554    6682     29882    19   /dev/hda4
/slot0              2026608   4        2026604    1   /dev/hdc1
/logflash            7997912   219408   7372232   3    /dev/hde1
/bootflash_sup-remote 1767480   1121784   555912    67
127.1.1.6:/mnt/bootflash/
/logflash_sup-remote 7953616   554976   6994608    8   127.1.1.6:/mnt/logflash/
```



(注) この出力を確認する際、[Filesystem]列の値がnoneの場合は、tmpfsタイプであることを意味します。

この例では、/var/sysmgr（またはサブフォルダ）が多くの領域を使用しているため、使用率が高くなります。/var/sysmgrはtmpfsマウントです。つまり、ファイルはRAMにのみ存在します。パーティションを占有しているファイルのタイプと、それらのファイルの由来（コア/デバッグ/など）を判別する必要があります。ファイルを削除すると使用率は低下しますが、どのタイプのファイルが領域を占有しているか、そしてどのプロセスがそれらのファイルをtmpfsに残しているかを判断する必要があります。

次のコマンドを使用して、CLIで問題のファイルを表示し、削除します。

- **show system internal dir full directory path** コマンドは、指定されたパスのすべてのファイルとサイズをリスト表示します（隠しコマンド）。
- **filesys delete full file path** コマンドは、特定のファイルを削除します（隠しコマンド）。

カーネル

カーネルの問題はそれほど一般的ではありませんが、**show system internal kernel meminfo** コマンドの出力でスラブの使用状況を確認することで問題を特定できます。一般に、カーネルのトラブルシューティングでは、使用率が増加している理由を切り分けるために、シスコのカスタマー サポートが必要です。

時間の経過とともにスラブのメモリ使用量が増加する場合は、次のコマンドを使用して詳細情報を収集します。

- この項で説明している **show system internal kernel malloc-stats** コマンドは、現在ロードされているすべての KLM、malloc、および空きカウントを表示します。

```
switch# show system internal kernel malloc-stats
Kernel Module Memory Tracking
-----
Module      kmalloc  kcalloc  kfree   diff
klm_usd    00318846 00000000 00318825 00000021
klm_eobcmn 08366981 00000000 08366981 00000000
klm_utaker 00001306 00000000 00001306 00000000
klm_sysmgr-hb 00000054 00000000 00000049 00000005
klm_idehs   00000001 00000000 00000000 00000001
klm_sup_ctrl_mc 00209580 00000000 00209580 00000000
klm_sup_config 00000003 00000000 00000000 00000003
klm_mts     03357731 00000000 03344979 00012752
klm_kadb    00000368 00000000 00000099 00000269
klm_aipc    00850300 00000000 00850272 00000028
klm_pss     04091048 00000000 04041260 00049788
klm_rwsem   00000001 00000000 00000000 00000001
klm_vdc     00000126 00000000 00000000 00000126
klm_modlock 00000016 00000000 00000016 00000000
klm_e1000   00000024 00000000 00000006 00000018
klm_dc_srom 00000123 00000000 00000123 00000000
klm_sdwrap  00000024 00000000 00000000 00000024
klm_obfl   00000050 00000000 00000047 00000003
```

このコマンドの複数の反復を比較することで、一部の KLM が大量のメモリを割り当てているが、メモリを解放/返却していないかどうかを確認できます（差分値は通常と比較して非常に大きくなります）。

- この項で説明している **show system internal kernel skb-stats** コマンドは、SKB（KLM がパケットを送受信するために使用するバッファ）の消費量を表示します。

```
switch# show system internal kernel skb-stats
Kernel Module skbuff Tracking
-----
Module      alloc     free     diff
klm_shreth 00028632 00028625 00000007
klm_eobcmn 02798915 02798829 00000086
klm_mts    00420053 00420047 00000006
klm_aipc   00373467 00373450 00000017
klm_e1000  16055660 16051210 00004450
```

このコマンドの複数の反復の出力を比較して、差分値が増加しているかどうかを確認します。

- この項で説明している **show hardware internal proc-info slabinfo** コマンドは、すべてのスラブ情報（カーネル管理に使用されるメモリ構造）をダンプします。出力は、大きくなることもあります。

ユーザ プロセス

ページキャッシュとカーネルの問題が除外されている場合は、一部のユーザプロセスが大量のメモリを使用しているか、実行中のプロセス数が多いため（使用可能な機能の数が多いため）、使用率が高くなっているという可能性があります。



(注) Cisco NX-OS は、ほとんどのプロセスのメモリ制限を定義しています（rlimit）。この rlimit を超えると、sysmgr によってプロセスがクラッシュし、通常はコアファイルが生成されます。rlimit に近いプロセスは、プラットフォームの使用率に大きな影響を与えない可能性がありますが、クラッシュが発生すると問題になる可能性があります。

大量のメモリを使用しているプロセスの特定

次のコマンドは、特定のプロセスが大量のメモリを使用しているかどうかを確認するのに役立ちます。

- The **show process memory** コマンドは、プロセスごとのメモリ割り当てを表示します。

```
switch# show processes memory
PID MemAlloc MemLimit MemUsed StackBase/Ptr Process
----- -----
4662 52756480 562929945 150167552 bffffdf00/bffffd970 netstack
```



(注) **show process memory** の出力 コマンドの出力は、現在の使用率の完全に正確な図を提供しない可能性があります（割り当てられていることを意味しません）。このコマンドは、プロセスが制限に近づいているかどうかを判断するのに役立ちます。

- この項で説明している **show system internal processes memory** コマンドは、メモリアラートログにプロセス情報を表示します（イベントが発生した場合）。

プロセスが実際に使用しているメモリ量を確認するには、Resident Set Size (RSS) を確認します。この値は、プロセスによって消費されているメモリの量 (KB 単位) の大まかな指標となります。この情報は、**show system internal processes memory** コマンドを使用してこの収集できます。

```
switch# show system internal processes memory
PID TTY STAT TIME MAJFLT TRS RSS VSZ %MEM COMMAND
4811 ? Ssl 00:00:16 0 0 49772 361588 0.3 /isan/bin/routing-sw/clis
-cli /isan/etc/routing-sw/cli
4928 ? Ssl 00:18:41 0 0 44576 769512 0.2
/isan/bin/routing-sw/netstack /isan/etc/routing-sw/pm.cfg
4897 ? Ssl 00:00:18 0 0 42604 602216 0.2 /isan/bin/routing-sw/arp
```

特定のプロセスがメモリを使用している方法の特定

```

4791 ? Ss 00:00:00 0 0 34384 318856 0.2 /isan/bin/pixm_vl
4957 ? Ssl 00:00:26 0 0 30440 592348 0.1 /isan/bin/snmpd -f -s
udp:161 udp6:161 tcp:161 tcp6:161
5097 ? Ssl 00:06:53 0 0 28052 941880 0.1 /isan/bin/routing-sw/pim
-t
5062 ? Ss 00:01:00 0 0 27300 310596 0.1 /isan/bin/diag_port_lb
5087 ? Ssl 00:03:53 0 0 24988 992756 0.1 /isan/bin/routing-sw/bgp
-t 65001
4792 ? Ss 00:00:00 0 0 24080 309024 0.1 /isan/bin/pixm_gl
5063 ? Ss 00:00:01 0 0 21940 317440 0.1 /isan/bin/ethpm
5044 ? Ss 00:00:00 0 0 21700 304032 0.1 /isan/bin/eltm
5049 ? Ss 00:00:14 0 0 20592 306156 0.1 /isan/bin/qpqosmgr
5042 ? Ssl 00:00:05 0 0 20580 672640 0.1 /isan/bin/routing-sw/igmp

5082 ? Ssl 00:00:25 0 0 19948 914088 0.1 /isan/bin/routing-sw/mrib
-m 4
5091 ? Ssl 00:01:58 0 0 19192 729500 0.1 /isan/bin/routing-sw/ospfv3
-t 8893
5092 ? Ssl 00:01:55 0 0 18988 861556 0.1 /isan/bin/routing-sw/ospf
-t 6464
5083 ? Ss 00:00:06 0 0 18876 309516 0.1 /isan/bin/mfdm
remaining output omitted

```

特定のプロセスの使用率が時間の経過とともに増加する場合は、プロセスの使用率に関する追加情報を収集する必要があります。

特定のプロセスがメモリを使用している方法の特定

プロセスが予想よりも多くのメモリを使用していると判断した場合は、プロセスがメモリをどのように使用しているかを調査すると役立ちます。

- **show system internal sysmgr service pid PID-in-decimal** コマンドを使用して、指定された PID を実行しているサービス情報をダンプします。

```

switch# show system internal sysmgr service pid 4727
Service "pixm" ("pixm", 109):
UUID = 0x133, PID = 4727, SAP = 176
State: SRV_STATE_HANDSHAKED (entered at time Fri May 10 01:42:01 2013).
Restart count: 1
Time of last restart: Fri May 10 01:41:11 2013.
The service never crashed since the last reboot.
Tag = N/A
Plugin ID: 1

```

上記の出力の UUID を 10 進数に変換し、次のコマンドで使用します。



(注)

ラボでトラブルシューティングを行う場合は、次の隠しコマンドを使用してCisco NX-OS 16 進数/10 進数変換を使用できます。

- **hex<decimal to convert>**
- **dec<hexadecimal to convert>**

- **show system internal kernel memory uuid *uuid-in-decimal*** コマンドを使用して、システム内の特定の UUID のライブラリを含む詳細なプロセス メモリ使用量を表示します（sysmgr サービスの出力から UUID を変換します）。

```
switch# show system internal kernel memory uuid 307
Note: output values in KiloBytes
      Name          rss   shrd    drt   map   heap   ro   dat   bss   stk
  misc
  ----
  ----
  /isan/bin/pixm_0           7816  5052   2764   1     0     0     0     0   52
  /isan/plugin/1/isan/bin/pixm_24        115472   0  115472   0  109176  752   28  6268   0
  /lib/ld-2.3.3.so_8           84    76     8   2     0    76     0     0   0
  /usr/lib/libz.so.1.2.1.1_0       16    12     4   1     0    12     4     0   0
  /usr/lib/libstdc++.so.6.0.3_0     296   272    24   1     0   272   20     4   0
  /lib/libgcc_s.so.1_0           1824   12  1812   1  1808   12     4     0   0
  /isan/plugin/1/isan/lib/libtmifdb.so.0_0        12     8     4   1     0    8     4     0   0
  /isan/plugin/0/isan/lib/libtmifdb_stub_0        12     8     4   1     0    8     4     0   0
  /dev/mts_0                     0     0     0   0     1    0     0     0   0
  /isan/plugin/1/isan/lib/libpcm_sdb.so._0         16    12     4   1     0   12     4     0   0
  /isan/plugin/1/isan/lib/libethpm.so.0._0         76    60    16   1     0   60   16     0   0
  /isan/plugin/1/isan/lib/libsvifdb.so._0           20     4    16   1    12    4     4     0   0
  /usr/lib/libcrypto.so.0.9.7_0         272   192    80   1     0   192   76     4   0
  /isan/plugin/0/isan/lib/libeureka_hash_0           8     4     4   1     0    4     4     0   0
  remaining output omitted
```

この出力は、プロセスが特定のライブラリにメモリを保持しているかどうかを判断するのに役立ち、メモリリークの識別に役立ちます。

- **show system internal service mem-stats detail** コマンドを使用して、特定のサービスのライブラリを含む詳細なメモリ使用率を表示します。

```
switch# show system internal pixm mem-stats detail
Private Mem stats for UUID : Malloc track Library(103) Max types: 5
-----
TYPE NAME          ALLOCS          BYTES
          Curr    Max    Curr    Max
  2 MT_MEM_mtrack_hdl      35     35  132132  149940
  3 MT_MEM_mtrack_info     598    866   9568   13856
  4 MT_MEM_mtrack_lib_name 598    866  15860   22970
```

組み込みプラットフォームのメモリモニタリング

```
Total bytes: 157560 (153k)
-----
Private Mem stats for UUID : Non mtrack users(0) Max types: 157
-----
  TYPE NAME                                ALLOCS          BYTES
                                         CURR      MAX    CURR      MAX
  1 [0x41000000]ld-2.15.so        283      283   48255   48256
  2 [0x41024000]libc-2.15.so       142      144   4979    5587
  8 [0x41241000]libglib-2.0.so.0.3200.3  500      771  10108  15588
 39 [0xf68af000]libbindxobj.so       7       7    596    596
 45 [0xf68ca000]libavl.so         73       73  1440    1440
 67 [0xf71b3000]libsdb.so        56       58  3670  73278
 75 [0xf7313000]libmpmpts.so      35       37   280    380
 86 [0xf7441000]libutils.so       23       28  3283   5766
 89 [0xf74bf000]libpss.so        59       60  8564  483642
 90 [0xf750b000]libmtts.so       7       8   816    828
 92 [0xf754c000]libacfg.so       0       4     0   51337
-----
Total bytes: 82817 (80k)
-----
remaining output omitted
```

これらの出力は通常、プロセスまたはそのライブラリの潜在的なメモリリークを調査する際に、シスコ カスタマー サポート担当者によって要求されます。

組み込みプラットフォームのメモリモニタリング

Cisco NX-OS には、システムのハング、プロセスのクラッシュ、およびその他の望ましくない動作を回避するために、カーネルによる、メモリ使用量のモニタリング機構が組み込まれています。プラットフォームマネージャは、（搭載されている RAM の総量を基準とする）メモリの使用率を定期的にチェックし、使用率が設定されたしきい値を超えると、自動的にアラートイベントを生成します。アラートレベルに達すると、カーネルは不要になったページ（たとえば、アクセスされなくなった永続ファイルのページキャッシュ）を解放することでメモリを解放しようとします。または、クリティカルレベルに達すると、カーネルは、メモリ使用率が最も高いプロセスを強制終了します。Cisco NX-OS の他のコンポーネントには、ボーダーゲートウェイ プロトコル (BGP) のグレースフルローメモリハンドリングなどのメモリアラート処理が導入されており、プロセスがそれ自身の動作を調整してメモリの使用率を制御できるようになっています。

メモリしきい値

多くの機能が展開されている場合、ベースラインのメモリでは、次のしきい値が必要です。

- MINOR
- SEVERE
- CRITICAL

デフォルトのしきい値は DRAM サイズに応じて起動時に計算されるため、その値はプラットフォームで使用されている DRAM サイズによって異なります。しきい値は、**system memory-thresholds minor** パーセンテージ **severe** パーセンテージ **critical** パーセンテージを使用して設定できます。コマンドを使用する必要があります。

Cisco NX-OS リリース 10.2(4)M 以降、デフォルトのシステムメモリしきい値は次のとおりです。

Cisco NX-OS リリース 10.3(1)F 以降、デフォルトのシステムメモリのしきい値は次のとおりです。

- クリティカル : 91
- 重大 : 89
- マイナー : 88

この項で説明している **show system internal memory-status** コマンドを使用すると、現在のメモリアラートステータスを確認できます。

```
switch# show system internal memory-status
MemStatus: OK
```

拡張 BGP EVPN VxLAN VNI（サポートされている拡張については、Cisco Nexus 9000 シリーズ NX-OS 検証済み拡張性ガイドを参照）を含む拡張導入を実行しているスイッチでは、デフォルトのシステムメモリしきい値が Cisco NX-OS リリース 10.3 (3) F リリースでサポートされています。システムがクリティカルメモリアラートに反応しないようにするには、アップグレードする前に、システムメモリのしきい値をより高い値に構成します。たとえば、システムメモリのしきい値をマイナーの場合は 90、重大な場合は 94、クリティカルの場合は 95 に設定します。

メモリアラート

メモリしきい値が渡されると ([OK]-> [MINOR]、[MINOR]-> [SEVERE]、[SEVERE]-> [CRITICAL])、Cisco NX-OS プラットフォームマネージャはメモリ使用率のスナップショットをキャプチャし、アラートを syslog に記録します。このスナップショットは、メモリ使用率が高い理由（プロセス、ページキャッシング、またはカーネル）を判断するのに役立ちます。ログは Linux ルートパス (/) で生成され、可能な場合はコピーが OBFL (/mnt/plog) に移動されます。このログは、ページキャッシング、カーネル、または Cisco NX-OS ユーザプロセスによって消費されたメモリが原因でメモリ使用率が高いかどうかを判断するのに非常に役立ちます。

この項で説明している **show system internal memory-alerts-log** コマンドにより、メモリアラートログを表示します。

メモリアラートログは、次の出力で構成されます。

コマンド	説明
cat /proc/memory_events	メモリアラートが発生したときのタイムスタンプのログを提供します。

コマンド	説明
cat /proc/meminfo	RAM の合計、ページキャッシュによって消費されたメモリ、スラブ（カーネルヒープ）、マッピングされたメモリ、使用可能な空きメモリなどの全体的なメモリ統計情報を表示します。
cat /proc/memtrack	KLM（カーネルメモリで実行中の Cisco NX-OS プロセス）の割り当て/割り当て解除カウントを表示します。
df -hT	ファイルシステム使用率情報を（タイプとともに）表示します。
du --si -La /tmp	/tmp にあるすべてのファイル情報を表示します（/var/tmp へのシンボリックリンク）。
cat /proc/memory_events	データ収集中に使用率が変更されたかどうかを判断するため、2回目にダンプされます。
cat /proc/meminfo	データ収集中に使用率が変更されたかどうかを判断するため、2回目にダンプされます。

LPSS 共有メモリ監視

Cisco NX-OS 10.5 (1) F 以降、LPSS (Lightweight Persistent Storage Service) 共有メモリ監視機能が導入されました。ユーザーはこの機能を使用して、LPSS による共有メモリの使用状況をモニターできます。この機能は自動的にイネーブルになります。この機能は、すべての Nexus および 3000 シリーズ スイッチでサポートされています。

LPSS 共有メモリ監視の無効化

手順の概要

1. **configure**
2. (オプション) **system lpss monitor**
3. (オプション) **frequency frequency**
4. (オプション) **threshold threshold**
5. **no system lpss monitor**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 :	構成モードになります。

	コマンドまたはアクション	目的
	switch# configure switch(config)#	
ステップ2	(オプション) system lpss monitor 例： switch(config)# system lpss monitor	LPSS モニタリングの構成
ステップ3	(オプション) frequency frequency 例： switch(config-lpss)# frequency 8	モニタリング頻度の構成指定した頻度に達すると、 syslog が生成されます。 頻度のデフォルト : 10 (500 ミリ秒の倍数) 周波数の範囲 : 1-10
ステップ4	(オプション) threshold threshold 例： switch(config-lpss)# threshold 80	モニタリングしきい値を構成します。しきい値に達 すると、syslog が生成されます。 しきい値のデフォルト : 100 (パーセント単位) しきい値範囲 : 共有メモリの 70 ~ 100
ステップ5	no system lpss monitor 例： switch(config-lpss)# system lpss monitor	この機能をディセーブルにします。 この機能は自動的にイネーブルになります。

LPSS 共有メモリ監視構成の確認

LPSS の使用状況の詳細を表示するには、次のコマンドを使用します。

```
switch# show system lpss monitor usage
Total SHM size: 6400 MB
Total LPSS Shared memory usage: 756 MB (11%)
Monitoring Frequency: 8
Total Threshold: 80
switch#
```

■ LPSS 共有メモリ監視構成の確認

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。