



サーバ障害のトラブルシューティング

- プロセスのメモリ割り当ての特定（1ページ）
- プロセスのCPU使用率の特定（2ページ）
- モニタリングプロセスのコアファイル（3ページ）
- クラッシュコアファイルの処理（3ページ）
- コアのクリア（4ページ）
- コアファイルの自動コピーのイネーブル化（4ページ）

プロセスのメモリ割り当ての特定

メモリ内の各プロセスの割り当て、制限、メモリ割り当て、および使用状況を特定できます。次は **show processes memory** コマンドからの出力例です。この出力は、例を簡潔にするために省略されています。

switch# show processes memory					
PID	MemAlloc	MemLimit	MemUsed	StackBase/Ptr	Process
1	159744	0	2027520	ff808d30/ffffffff	init
2	0	0	0	0/0	kthreadd
3	0	0	0	0/0	migration/0
4	0	0	0	0/0	ksoftirqd/0
5	0	0	0	0/0	watchdog/0
6	0	0	0	0/0	migration/1
7	0	0	0	0/0	ksoftirqd/1
8	0	0	0	0/0	watchdog/1
9	0	0	0	0/0	migration/2
10	0	0	0	0/0	ksoftirqd/2
11	0	0	0	0/0	watchdog/2
12	0	0	0	0/0	migration/3
13	0	0	0	0/0	ksoftirqd/3
14	0	0	0	0/0	watchdog/3
15	0	0	0	0/0	migration/4
16	0	0	0	0/0	ksoftirqd/4
17	0	0	0	0/0	watchdog/4
18	0	0	0	0/0	migration/5
19	0	0	0	0/0	ksoftirqd/5
20	0	0	0	0/0	watchdog/5
21	0	0	0	0/0	migration/6
22	0	0	0	0/0	ksoftirqd/6
23	0	0	0	0/0	watchdog/6
24	0	0	0	0/0	migration/7

■ プロセスの CPU 使用率の特定

```

25      0 0          0          0/0  ksoftirqd/7
26      0 0          0          0/0  watchdog/7
27      0 0          0          0/0  events/0
28      0 0          0          0/0  events/1
29      0 0          0          0/0  events/2
30      0 0          0          0/0  events/3
31      0 0          0          0/0  events/4
32      0 0          0          0/0  events/5
33      0 0          0          0/0  events/6
34      0 0          0          0/0  events/7
35      0 0          0          0/0  khelper
36      0 0          0          0/0  netns
37      0 0          0          0/0  kblockd/0

```

この項で説明している **show processes memory** コマンドには、次のキーワードが含まれます。

キーワード	説明
>	出力をファイルにリダイレクトします。
>>	出力が既存のファイルに追加されます。
shared	共有メモリ情報を表示します。

プロセスの CPU 使用率の特定

メモリ内で実行中のプロセスの CPU 使用率を特定できます。次は **show processes cpu** コマンドからの出力例です。この出力は、例を簡潔にするために省略されています。

```

switch# show processes cpu

CPU utilization for five seconds: 0%/0%; one minute: 1%; five minutes: 2%

  PID  Runtime(ms) Invoked   uSecs  5Sec    1Min   5Min   TTY Process
  ---  -----  -----  -----  -----  -----  -----  -----
  1    28660    405831     70  0.00%  0.00%  0.00%  -  init
  2      21      1185      18  0.00%  0.00%  0.00%  -  kthreadd
  3      468      36439     12  0.00%  0.00%  0.00%  -  migration/0
  4    79725    8804385      9  0.00%  0.00%  0.00%  -  ksoftirqd/0
  5      0       4        65  0.00%  0.00%  0.00%  -  watchdog/0
  6      472      35942     13  0.00%  0.00%  0.00%  -  migration/1
  7    33967    953376     35  0.00%  0.00%  0.00%  -  ksoftirqd/1
  8      0       11        3  0.00%  0.00%  0.00%  -  watchdog/1
  9      424      35558     11  0.00%  0.00%  0.00%  -  migration/2
 10     58084    7683251      7  0.00%  0.00%  0.00%  -  ksoftirqd/2
 11      0       3        1  0.00%  0.00%  0.00%  -  watchdog/2
 12     381      29760     12  0.00%  0.00%  0.00%  -  migration/3
 13     17258    265884     64  0.00%  0.00%  0.00%  -  ksoftirqd/3
 14      0       2        0  0.00%  0.00%  0.00%  -  watchdog/3
 15     46558    1300598     35  0.00%  0.00%  0.00%  -  migration/4
 16   1332913    4354439    306  0.00%  0.00%  0.00%  -  ksoftirqd/4
 17      0       6        2  0.00%  0.00%  0.00%  -  watchdog/4
 18     45808    1283581     35  0.00%  0.00%  0.00%  -  migration/5
 19    981030    1973423    497  0.00%  0.00%  0.00%  -  ksoftirqd/5
 20      0       16        3  0.00%  0.00%  0.00%  -  watchdog/5
 21    48019    1334683     35  0.00%  0.00%  0.00%  -  migration/6

```

22	1084448	2520990	430	0.00%	0.00%	0.00%	-	ksoftirqd/6		
23	0	31	3	0.00%	0.00%	0.00%	-	watchdog/6		
24	46490	1306203	35	0.00%	0.00%	0.00%	-	migration/7		
25	1187547	2867126	414	0.00%	0.00%	0.00%	-	ksoftirqd/7		
26	0	16	3	0.00%	0.00%	0.00%	-	watchdog/7		
27	21249	2024626	10	0.00%	0.00%	0.00%	-	events/0		
28	8503	1990090	4	0.00%	0.00%	0.00%	-	events/1		
29	11675	1993684	5	0.00%	0.00%	0.00%	-	events/2		
30	9090	1973913	4	0.00%	0.00%	0.00%	-	events/3		
31	74118	2956999	25	0.00%	0.00%	0.00%	-	events/4		
32	76281	2837641	26	0.00%	0.00%	0.00%	-	events/5		
33	129651	3874436	33	0.00%	0.00%	0.00%	-	events/6		
34	8864	2077714	4	0.00%	0.00%	0.00%	-	events/7		
35	0	8	23	0.00%	0.00%	0.00%	-	khelper		
36	234	34	6884	0.00%	0.00%	0.00%	-	netns		

show processes cpu コマンドには、次のキーワードが含まれています。

キーワード	説明
>	出力をファイルにリダイレクトします。
>>	出力が既存のファイルに追加されます。
history	CPU の使用状況に関する情報を表示します。
sort	メモリ使用量に基づいてリストをソートします。

モニタリング プロセスのコア ファイル

show cores を使用してプロセス コア ファイルをモニタできます。 コマンドを使用する必要があります。

```
switch# show cores
Module  Instance  Process-name      PID      Date (Year-Month-Day Time)
-----  -----  -----
28       1          bgp-64551      5179     2013-11-08 23:51:26
```

出力には、現在アクティブなスーパーバイザからアップロードできるすべてのコアが表示されます。

クラッシュ コア ファイルの処理

クラッシュ コア ファイルを処理するには、**show processes log** コマンドを使用します。

```
switch# show process log
Process      PID      Normal-exit  Stack-trace  Core      Log-create-time
-----  -----
ntp          919      N            N            N        Jun 27 04:08
snsm         972      N            Y            N        Jun 24 20:50
```

コアのクリア

clear cores を使用してコアをクリアできます。 コマンドを使用します。

```
switch# clear cores
```

コア ファイルの自動コピーのイネーブル化

システム コアを入力できます。 コマンドを使用して、TFTP サーバ、フラッシュ ドライブ、またはファイルへのコア ファイルの自動コピーを有効にします。

```
switch(config)# system cores tftp://10.1.1.1/cores
```

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。