

5.7. プロセスとシグナル制御 -kill コマンド-

Linux では、プログラムのことをプロセス (process) と呼びます。プログラムは主にディスクに格納されており、メモリにコピーされた時にはプロセスと呼ばれます。Linux はマルチタスク OS ですから、同時に複数のプロセスがシステムで動くことになります。そのためプロセスの管理は一般ユーザにとっても、管理者にとっても重要です。

プロセスには 1~32767 までの番号 (プロセス ID→PID) が一意に割り振られます。互いの区別は PID により行われています。また、プロセスは起動した人が所有権を持ち (特殊な状態に変更することが可能です)、そのユーザレベルでしか行動することはできません。

プロセスを確認するには ps コマンドを用います。ps には多くのオプションがありますが、通常使うのは auxw というオプションです。慣例として、ps のオプションには、オプションであることを示す “-” をつけません。

```
$ ps auxw
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  19396   776 ?        Ss   Jan25   0:03 /sbin/init
root         2  0.0  0.0     0     0 ?        S    Jan25   0:00 [kthreadd]
root         3  0.0  0.0     0     0 ?        S    Jan25   0:00 [migration/0]
...
SL6        8876  0.0  0.0 108400  1812 pts/1    Ss+  16:18   0:00 bash
postfix    9067  0.0  0.1  78704  3172 ?        S    17:21   0:00 pickup -l -t fifo -u
SL6        9111  0.0  0.0 108116  1096 pts/0    R+   17:40   0:00 ps auxw
```

- USER 動作権限
- PID プロセス ID
- %CPU 単位時間内での CPU 利用率
- %MEM 単位時間内でのメモリ利用率
- VSZ プロセスが要求しているメモリ量
- RSS VSZ 中、実際に利用しているメモリ量
- TTY 利用している端末名 (端末を使っていない場合には “?” になる)
- STAT プロセス状態、主に S (Sleep=休眠) か R (Run=動作中) である
- START TIME 活動開始時間と CPU 使用時間の合計
- COMMAND 起動しているプロセス名

なんらかの理由でプロセスを終了させたい場合には、PID を ps で確認した上で、kill コマンドでプロセスを停止させることができます。(ただし自分所有のプロセスのみです)。

```
$ kill 8876
```

たとえば、上記 ps 中の PID 8876 (bash) を終了させたいという場合は、と、kill コマンドに引数として PID を渡すことで、指定した PID を持つプロセスを終了させることができます。しかし、他人が所有するプロセスを止めようとする、(root 以外は) 拒否されます。

```
$ kill 1
-bash: kill: (1) - 許可されていない操作です
```

さて、今 kill コマンドによるプロセスの終了というものを紹介しましたが、この仕組みについて少し詳しく知っておきましょう。Linux にはシグナルというものを用意されていて、このシグナルを特定のプロセスに送ることができます。シグナルは一般的なものから内部用のものまで、合わせると数十存在します。この情報はオンラインマニュアルで調べると細かい情報があります。

```
$ man 7 signal
```

とすると表示されます。

先ほど用いた kill コマンドは、このシグナルを送りつけるためのものであり、特にオプション指定しない場合は、送信されるシグナルは“SIGTERM” (15 番) となります。このシグナルは「終了して下さい」という意味で、受信したプロセスは終了に向けた処理を行うようになります。しかし、シグナルを受けた時の反応は、一部の例外を除いてプログラミング可能なため、わざとシグナルの意図に従わないという構造のプログラムも存在しています。また、そもそも受けたシグナルに応答しないものもあります。それ以外にも、暴走していてシグナルに応答できないというケースもあります。こういったプロセスはいくら kill しても終了しません。

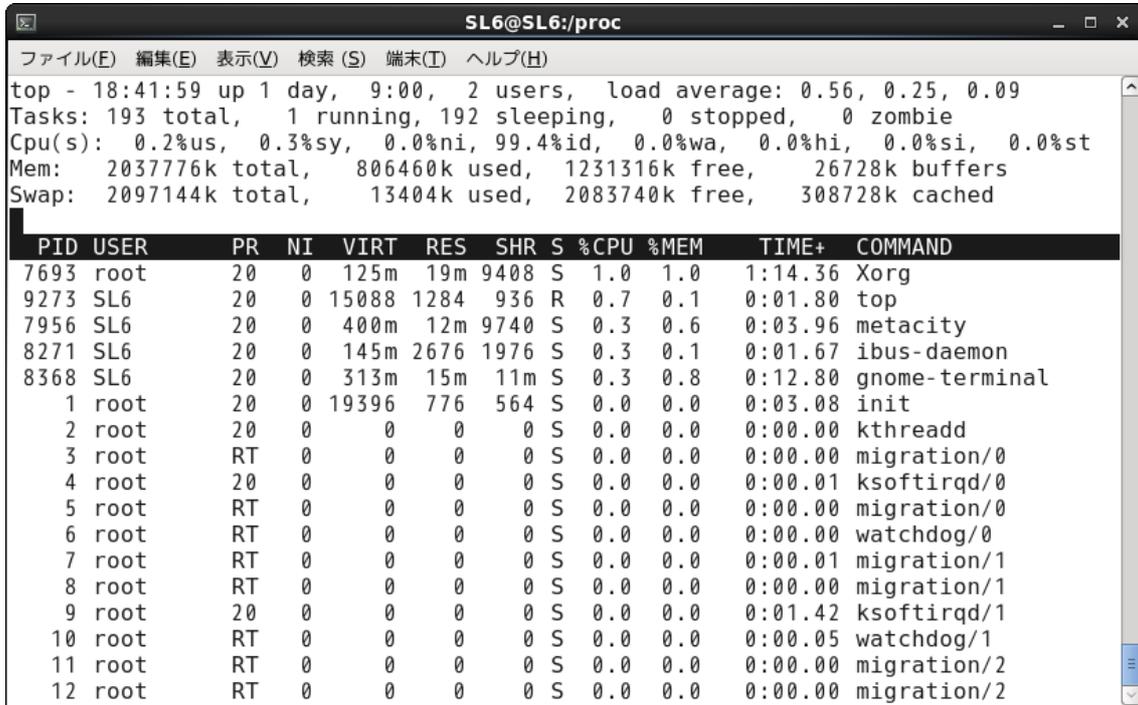
このような暴走プロセスに対して効果があるのが“SIGKILL” (9 番) であり、強制終了させることができます。SIGKILL を送るためには、

```
$ kill -9 プロセス ID
```

としてみましょう。“-番号”でシグナル番号を指定することもできます。

応用例として、killall コマンドがあります。こちらはプロセス ID ではなくプロセス名を引数とすることにより、該当したプロセス全てにシグナルを送ることが可能という便利なものですが、意図していないプロセスにまでシグナルを送ることがあるため (他人のプロセスや同名で異なるプロセス等)、root での使用はおすすめしません。

プロセスを確認するために有効なツールとしては、top コマンドがあります。このコマンドを使うと、端末全体を用いて、現在のプロセス状態を眺めることができます。



```
SL6@SL6:/proc
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
top - 18:41:59 up 1 day, 9:00, 2 users, load average: 0.56, 0.25, 0.09
Tasks: 193 total, 1 running, 192 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.2%us, 0.3%sy, 0.0%ni, 99.4%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 2037776k total, 806460k used, 1231316k free, 26728k buffers
Swap: 2097144k total, 13404k used, 2083740k free, 308728k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 7693 root        20   0  125m  19m 9408 S   1.0   1.0   1:14.36 Xorg
 9273 SL6         20   0 15088 1284  936 R   0.7   0.1   0:01.80 top
 7956 SL6         20   0   400m  12m 9740 S   0.3   0.6   0:03.96 metacity
 8271 SL6         20   0   145m 2676 1976 S   0.3   0.1   0:01.67 ibus-daemon
 8368 SL6         20   0   313m  15m 11m S   0.3   0.8   0:12.80 gnome-terminal
    1 root        20   0 19396  776  564 S   0.0   0.0   0:03.08 init
    2 root        20   0     0     0     0 S   0.0   0.0   0:00.00 kthreadd
    3 root        RT   0     0     0     0 S   0.0   0.0   0:00.00 migration/0
    4 root        20   0     0     0     0 S   0.0   0.0   0:00.01 ksoftirqd/0
    5 root        RT   0     0     0     0 S   0.0   0.0   0:00.00 migration/0
    6 root        RT   0     0     0     0 S   0.0   0.0   0:00.00 watchdog/0
    7 root        RT   0     0     0     0 S   0.0   0.0   0:00.01 migration/1
    8 root        RT   0     0     0     0 S   0.0   0.0   0:00.00 migration/1
    9 root        20   0     0     0     0 S   0.0   0.0   0:01.42 ksoftirqd/1
  10 root        RT   0     0     0     0 S   0.0   0.0   0:00.05 watchdog/1
  11 root        RT   0     0     0     0 S   0.0   0.0   0:00.00 migration/2
  12 root        RT   0     0     0     0 S   0.0   0.0   0:00.00 migration/2
```

top 出力

top コマンドは一定時間（標準設定で5秒）ごとに、最新の統計を表示するようになっており、継続的に監視をする時には非常に便利です。また、表示する際のプロセスの並べ方を CPU の利用率順や稼働時間順、メモリ使用順に変更、特定のプロセスにシグナルを送信する等の機能が用意されています。表示中に“?”を入力することでキー操作を確認できますのでチェックしてみてください。また、終了は“q”となっています。