[**每天一个linux命令（41）：ps命令**](http://www.cnblogs.com/peida/archive/2012/12/19/2824418.html)

Linux中的ps命令是Process Status的缩写。ps命令用来列出系统中当前运行的那些进程。ps命令列出的是当前那些进程的快照，就是执行ps命令的那个时刻的那些进程，如果想要动态的显示进程信息，就可以使用top命令。

要对进程进行监测和控制，首先必须要了解当前进程的情况，也就是需要查看当前进程，而 ps 命令就是最基本同时也是非常强大的进程查看命令。使用该命令可以确定有哪些进程正在运行和运行的状态、进程是否结束、进程有没有僵死、哪些进程占用了过多的资源等等。总之大部分信息都是可以通过执行该命令得到的。

ps 为我们提供了进程的一次性的查看，它所提供的查看结果并不动态连续的；如果想对进程时间监控，应该用 top 工具。

kill 命令用于杀死进程。

**linux上进程有5种状态:**

1. 运行(正在运行或在运行队列中等待)

2. 中断(休眠中, 受阻, 在等待某个条件的形成或接受到信号)

3. 不可中断(收到信号不唤醒和不可运行, 进程必须等待直到有中断发生)

4. 僵死(进程已终止, 但进程描述符存在, 直到父进程调用wait4()系统调用后释放)

5. 停止(进程收到SIGSTOP, SIGSTP, SIGTIN, SIGTOU信号后停止运行运行)

**ps工具标识进程的5种状态码:**

D 不可中断 uninterruptible sleep (usually IO)

R 运行 runnable (on run queue)

S 中断 sleeping

T 停止 traced or stopped

Z 僵死 a defunct (”zombie”) process

**1．命令格式：**

ps[参数]

**2．命令功能：**

用来显示当前进程的状态

**3．命令参数：**

a  显示所有进程

-a 显示同一终端下的所有程序

-A 显示所有进程

c  显示进程的真实名称

-N 反向选择

-e 等于“-A”

e  显示环境变量

f  显示程序间的关系

-H 显示树状结构

r  显示当前终端的进程

T  显示当前终端的所有程序

u  指定用户的所有进程

-au 显示较详细的资讯

-aux 显示所有包含其他使用者的行程

-C<命令> 列出指定命令的状况

--lines<行数> 每页显示的行数

--width<字符数> 每页显示的字符数

--help 显示帮助信息

--version 显示版本显示

**4．使用实例：**

**实例1：显示所有进程信息**

**命令：**

ps -A

**输出：**

[root@localhost test6]# ps -A

  PID TTY          TIME CMD

    1 ?        00:00:00 init

    2 ?        00:00:01 migration/0

    3 ?        00:00:00 ksoftirqd/0

    4 ?        00:00:01 migration/1

    5 ?        00:00:00 ksoftirqd/1

    6 ?        00:29:57 events/0

    7 ?        00:00:00 events/1

    8 ?        00:00:00 khelper

   49 ?        00:00:00 kthread

   54 ?        00:00:00 kblockd/0

   55 ?        00:00:00 kblockd/1

   56 ?        00:00:00 kacpid

  217 ?        00:00:00 cqueue/0

  ……省略部分结果

**说明：**

**实例2：显示指定用户信息**

**命令：**

ps -u root

**输出：**

[root@localhost test6]# ps -u root

  PID TTY          TIME CMD

    1 ?        00:00:00 init

    2 ?        00:00:01 migration/0

    3 ?        00:00:00 ksoftirqd/0

    4 ?        00:00:01 migration/1

    5 ?        00:00:00 ksoftirqd/1

    6 ?        00:29:57 events/0

    7 ?        00:00:00 events/1

    8 ?        00:00:00 khelper

   49 ?        00:00:00 kthread

   54 ?        00:00:00 kblockd/0

   55 ?        00:00:00 kblockd/1

   56 ?        00:00:00 kacpid

    ……省略部分结果

**说明：**

**实例3：显示所有进程信息，连同命令行**

**命令：**

ps -ef

**输出：**

[root@localhost test6]# ps -ef

UID        PID  PPID  C STIME TTY          TIME CMD

root         1     0  0 Nov02 ?        00:00:00 init [3]

root         2     1  0 Nov02 ?        00:00:01 [migration/0]

root         3     1  0 Nov02 ?        00:00:00 [ksoftirqd/0]

root         4     1  0 Nov02 ?        00:00:01 [migration/1]

root         5     1  0 Nov02 ?        00:00:00 [ksoftirqd/1]

root         6     1  0 Nov02 ?        00:29:57 [events/0]

root         7     1  0 Nov02 ?        00:00:00 [events/1]

root         8     1  0 Nov02 ?        00:00:00 [khelper]

root        49     1  0 Nov02 ?        00:00:00 [kthread]

root        54    49  0 Nov02 ?        00:00:00 [kblockd/0]

root        55    49  0 Nov02 ?        00:00:00 [kblockd/1]

root        56    49  0 Nov02 ?        00:00:00 [kacpid]

……省略部分结果

**说明：**

**实例4：** **ps 与grep 常用组合用法，查找特定进程**

**命令：**

ps -ef|grep ssh

**输出：**

[root@localhost test6]# ps -ef|grep ssh

root      2720     1  0 Nov02 ?        00:00:00 /usr/sbin/sshd

root     17394  2720  0 14:58 ?        00:00:00 sshd: root@pts/0

root     17465 17398  0 15:57 pts/0    00:00:00 grep ssh

**说明：**

**实例5：将目前属于您自己这次登入的 PID 与相关信息列示出来**

**命令：**

ps -l

**输出：**

[root@localhost test6]# ps -l

F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD

4 S     0 17398 17394  0  75   0 - 16543 wait   pts/0    00:00:00 bash

4 R     0 17469 17398  0  77   0 - 15877 -      pts/0    00:00:00 ps

**说明：**

各相关信息的意义：

F 代表这个程序的旗标 (flag)， 4 代表使用者为 super user

S 代表这个程序的状态 (STAT)，关于各 STAT 的意义将在内文介绍

UID 程序被该 UID 所拥有

PID 就是这个程序的 ID ！

PPID 则是其上级父程序的ID

C CPU 使用的资源百分比

PRI 这个是 Priority (优先执行序) 的缩写，详细后面介绍

NI 这个是 Nice 值，在下一小节我们会持续介绍

ADDR 这个是 kernel function，指出该程序在内存的那个部分。如果是个 running的程序，一般就是 "-"

SZ 使用掉的内存大小

WCHAN 目前这个程序是否正在运作当中，若为 - 表示正在运作

TTY 登入者的终端机位置

TIME 使用掉的 CPU 时间。

CMD 所下达的指令为何

在预设的情况下， ps 仅会列出与目前所在的 bash shell 有关的 PID 而已，所以， 当我使用 ps -l 的时候，只有三个 PID。

**实例6：列出目前所有的正在内存当中的程序**

**命令：**

ps aux

**输出：**

[root@localhost test6]# ps aux

USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND

root         1  0.0  0.0  10368   676 ?        Ss   Nov02   0:00 init [3]

root         2  0.0  0.0      0     0 ?        S<   Nov02   0:01 [migration/0]

root         3  0.0  0.0      0     0 ?        SN   Nov02   0:00 [ksoftirqd/0]

root         4  0.0  0.0      0     0 ?        S<   Nov02   0:01 [migration/1]

root         5  0.0  0.0      0     0 ?        SN   Nov02   0:00 [ksoftirqd/1]

root         6  0.0  0.0      0     0 ?        S<   Nov02  29:57 [events/0]

root         7  0.0  0.0      0     0 ?        S<   Nov02   0:00 [events/1]

root         8  0.0  0.0      0     0 ?        S<   Nov02   0:00 [khelper]

root        49  0.0  0.0      0     0 ?        S<   Nov02   0:00 [kthread]

root        54  0.0  0.0      0     0 ?        S<   Nov02   0:00 [kblockd/0]

root        55  0.0  0.0      0     0 ?        S<   Nov02   0:00 [kblockd/1]

root        56  0.0  0.0      0     0 ?        S<   Nov02   0:00 [kacpid]

……省略部分结果

**说明：**

USER：该 process 属于那个使用者账号的

PID ：该 process 的号码

%CPU：该 process 使用掉的 CPU 资源百分比

%MEM：该 process 所占用的物理内存百分比

VSZ ：该 process 使用掉的虚拟内存量 (Kbytes)

RSS ：该 process 占用的固定的内存量 (Kbytes)

TTY ：该 process 是在那个终端机上面运作，若与终端机无关，则显示 ?，另外， tty1-tty6 是本机上面的登入者程序，若为 pts/0 等等的，则表示为由网络连接进主机的程序。

STAT：该程序目前的状态，主要的状态有

R ：该程序目前正在运作，或者是可被运作

S ：该程序目前正在睡眠当中 (可说是 idle 状态)，但可被某些讯号 (signal) 唤醒。

T ：该程序目前正在侦测或者是停止了

Z ：该程序应该已经终止，但是其父程序却无法正常的终止他，造成 zombie (疆尸) 程序的状态

START：该 process 被触发启动的时间

TIME ：该 process 实际使用 CPU 运作的时间

COMMAND：该程序的实际指令

**实例7：列出类似程序树的程序显示**

**命令：**

ps -axjf

**输出：**

[root@localhost test6]# ps -axjf

Warning: bad syntax, perhaps a bogus '-'? See /usr/share/doc/procps-3.2.7/FAQ

 PPID   PID  PGID   SID TTY      TPGID STAT   UID   TIME COMMAND

    0     1     1     1 ?           -1 Ss       0   0:00 init [3]

    1     2     1     1 ?           -1 S<       0   0:01 [migration/0]

    1     3     1     1 ?           -1 SN       0   0:00 [ksoftirqd/0]

    1     4     1     1 ?           -1 S<       0   0:01 [migration/1]

    1     5     1     1 ?           -1 SN       0   0:00 [ksoftirqd/1]

    1     6     1     1 ?           -1 S<       0  29:58 [events/0]

    1     7     1     1 ?           -1 S<       0   0:00 [events/1]

    1     8     1     1 ?           -1 S<       0   0:00 [khelper]

    1    49     1     1 ?           -1 S<       0   0:00 [kthread]

   49    54     1     1 ?           -1 S<       0   0:00  \\_ [kblockd/0]

   49    55     1     1 ?           -1 S<       0   0:00  \\_ [kblockd/1]

   49    56     1     1 ?           -1 S<       0   0:00  \\_ [kacpid]

**说明：**

**实例8：找出与 cron 与 syslog 这两个服务有关的 PID 号码**

**命令：**

**输出：**

[root@localhost test6]# ps aux | egrep '(cron|syslog)'

root      2682  0.0  0.0  83384  2000 ?        Sl   Nov02   0:00 /sbin/rsyslogd -i /var/run/syslogd.pid -c 5

root      2735  0.0  0.0  74812  1140 ?        Ss   Nov02   0:00 crond

root     17475  0.0  0.0  61180   832 pts/0    S+   16:27   0:00 egrep (cron|syslog)

[root@localhost test6]#

**说明：**

**其他实例：**

1. 可以用 | 管道和 more 连接起来分页查看

命令：

ps -aux |more

2. 把所有进程显示出来，并输出到ps001.txt文件

命令：

ps -aux > ps001.txt

3. 输出指定的字段

命令：

 ps -o pid,ppid,pgrp,session,tpgid,comm

输出：

[root@localhost test6]# ps -o pid,ppid,pgrp,session,tpgid,comm

  PID  PPID  PGRP  SESS TPGID COMMAND

17398 17394 17398 17398 17478 bash

17478 17398 17478 17398 17478 ps

[root@localhost test6]#