[**ps命令详解**](http://www.cnblogs.com/wangkangluo1/archive/2011/09/23/2185938.html)

原文地址：<http://apps.hi.baidu.com/share/detail/32573968>

有时候系统管理员可能只关心现在系统中运行着哪些程序，而不想知道有哪些进程在运行。由于一个应用程序可能需要启动多个进程。所以在同等情况下，进程的数 量要比程序多的多。为此从阅读方面考虑，管理员需要知道系统中运行的具体程序。要实现这个需求的话，就需要利用命令ps来帮忙。  
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。

名称：ps  
使用权限：所有使用者  
使用方式：ps [options] [--help]  
说明：显示瞬间行程 (process) 的动态  
参数：ps的参数非常多, 在此仅列出几个常用的参数并大略介绍含义  
-A    列出所有的进程  
-w    显示加宽可以显示较多的资讯  
-au    显示较详细的资讯  
-aux    显示所有包含其他使用者的行程  
  
############################################################  
  
常用参数：  
  
-A 显示所有进程（等价于-e）(utility)  
-a 显示一个终端的所有进程，除了会话引线  
-N 忽略选择。  
-d 显示所有进程，但省略所有的会话引线(utility)  
-x 显示没有控制终端的进程，同时显示各个命令的具体路径。dx不可合用。（utility）  
-p pid 进程使用cpu的时间  
-u uid or username 选择有效的用户id或者是用户名  
-g gid or groupname 显示组的所有进程。  
U username 显示该用户下的所有进程，且显示各个命令的详细路径。如:ps U zhang;(utility)  
-f 全部列出，通常和其他选项联用。如：ps -fa or ps -fx and so on.  
-l 长格式（有F,wchan,C 等字段）  
-j 作业格式  
-o 用户自定义格式。  
v 以虚拟存储器格式显示  
s 以信号格式显示  
-m 显示所有的线程  
-H 显示进程的层次(和其它的命令合用，如：ps -Ha)（utility）  
e 命令之后显示环境（如：ps -d e; ps -a e）(utility)  
h 不显示第一行  
  
############################################################  
  
ps命令常用用法（方便查看系统进程）  
  
1）ps a 显示现行终端机下的所有程序，包括其他用户的程序。  
2）ps -A 显示所有进程。  
3）ps c 列出程序时，显示每个程序真正的指令名称，而不包含路径，参数或常驻服务的标示。  
4）ps -e 此参数的效果和指定"A"参数相同。  
5）ps e 列出程序时，显示每个程序所使用的环境变量。  
6）ps f 用ASCII字符显示树状结构，表达程序间的相互关系。  
7）ps -H 显示树状结构，表示程序间的相互关系。  
8）ps -N 显示所有的程序，除了执行ps指令终端机下的程序之外。  
9）ps s 采用程序信号的格式显示程序状况。  
10）ps S 列出程序时，包括已中断的子程序资料。  
11）ps -t<终端机编号> 　指定终端机编号，并列出属于该终端机的程序的状况。  
12）ps u 　以用户为主的格式来显示程序状况。  
13）ps x 　显示所有程序，不以终端机来区分。  
最常用的方法是ps -aux,然后再利用一个管道符号导向到grep去查找特定的进程,然后再对特定的进程进行操作。  
  
############################################################  
  
运行 ps aux 的到如下信息：  
  
root:# ps aux  
USER      PID       %CPU    %MEM    VSZ    RSS    TTY    STAT    START    TIME    COMMAND  
smmsp    3521    0.0    0.7    6556    1616    ?    Ss    20:40    0:00    sendmail: Queue runner@01:00:00 f  
root    3532    0.0    0.2    2428    452    ?    Ss    20:40    0:00    gpm -m /dev/input/mice -t imps2  
htt    3563    0.0    0.0    2956    196    ?    Ss    20:41    0:00    /usr/sbin/htt -retryonerror 0  
htt    3564    0.0    1.7    29460    3704    ?    Sl    20:41    0:00    htt\_server -nodaemon  
root    3574    0.0    0.4    5236    992    ?    Ss    20:41    0:00    crond  
xfs    3617    0.0    1.3    13572    2804    ?    Ss    20:41    0:00    xfs -droppriv -daemon  
root    3627    0.0    0.2    3448    552    ?    SNs    20:41    0:00    anacron -s  
root    3636    0.0    0.1    2304    420    ?    Ss    20:41    0:00    /usr/sbin/atd  
dbus    3655    0.0    0.5    13840    1084    ?    Ssl    20:41    0:00    dbus-daemon-1 --system  
  
  
Head标头：  
  
USER    用户名  
UID    用户ID（User ID）  
PID    进程ID（Process ID）  
PPID    父进程的进程ID（Parent Process id）  
SID    会话ID（Session id）  
%CPU    进程的cpu占用率  
%MEM    进程的内存占用率  
VSZ    进程所使用的虚存的大小（Virtual Size）  
RSS    进程使用的驻留集大小或者是实际内存的大小，Kbytes字节。  
TTY    与进程关联的终端（tty）  
STAT    进程的状态：进程状态使用字符表示的（STAT的状态码）  
R 运行    Runnable (on run queue)            正在运行或在运行队列中等待。  
S 睡眠    Sleeping                休眠中, 受阻, 在等待某个条件的形成或接受到信号。  
I 空闲    Idle  
Z 僵死    Zombie（a defunct process)        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。  
D 不可中断    Uninterruptible sleep (ususally IO)    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。  
T 终止    Terminate                进程收到SIGSTOP, SIGSTP, SIGTIN, SIGTOU信号后停止运行运行。  
P 等待交换页  
W 无驻留页    has no resident pages        没有足够的记忆体分页可分配。  
X 死掉的进程  
< 高优先级进程                    高优先序的进程  
N 低优先    级进程                    低优先序的进程  
L 内存锁页    Lock                有记忆体分页分配并缩在记忆体内  
s 进程的领导者（在它之下有子进程）；  
l 多进程的（使用 CLONE\_THREAD, 类似 NPTL pthreads）  
+ 位于后台的进程组   
START    进程启动时间和日期  
TIME    进程使用的总cpu时间  
COMMAND    正在执行的命令行命令  
NI    优先级(Nice)  
PRI    进程优先级编号(Priority)  
WCHAN    进程正在睡眠的内核函数名称；该函数的名称是从/root/system.map文件中获得的。  
FLAGS    与进程相关的数字标识  
  
############################################################  
  
例子:  
查看当前系统进程的uid,pid,stat,pri, 以uid号排序.  
ps -eo pid,stat,pri,uid –sort uid  
  
查看当前系统进程的user,pid,stat,rss,args, 以rss排序.  
ps -eo user,pid,stat,rss,args –sort rss  
  
############################################################  
  
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。  
  
  
1、ps 的参数说明：  
  
ps 提供了很多的选项参数，常用的有以下几个：  
  
l 长格式输出；  
u 按用户名和启动时间的顺序来显示进程；  
j 用任务格式来显示进程；  
f 用树形格式来显示进程；  
  
a 显示所有用户的所有进程（包括其它用户）；  
x 显示无控制终端的进程；  
r 显示运行中的进程；  
ww 避免详细参数被截断；  
  
我们常用的选项是组合是 aux 或 lax，还有参数 f 的应用。  
  
  
2、ps aux 或 lax 输出的解释：  
  
USER 进程的属主；  
PID 进程的ID；  
PPID 父进程；  
%CPU 进程占用的CPU百分比；  
%MEM 占用内存的百分比；  
NI 进程的NICE值，数值大，表示较少占用CPU时间；  
VSZ 进程虚拟大小；  
RSS 驻留中页的数量；  
TTY 终端ID  
STAT 进程状态（有以下几种）  
D 无法中断的休眠状态（通常 IO 的进程）；  
R 正在运行可中在队列中可过行的；  
S 处于休眠状态；  
T 停止或被追踪；  
W 进入内存交换（从内核2.6开始无效）；  
X 死掉的进程（从来没见过）；  
Z 僵尸进程；  
  
< 优先级高的进程  
N 优先级较低的进程  
L 有些页被锁进内存；  
s 进程的领导者（在它之下有子进程）；  
l 多进程的（使用 CLONE\_THREAD, 类似 NPTL pthreads）；  
+ 位于后台的进程组；  
WCHAN 正在等待的进程资源；  
START 启动进程的时间；  
TIME 进程消耗CPU的时间；  
COMMAND 命令的名称和参数；  
  
  
3、应用举例：  
  
[root@localhost ~]# ps -aux |more  
可以用 | 管道和 more 连接起来分页查看。  
[root@localhost ~]# ps -aux > ps001.txt  
把结果输出到ps001.txt文本中并保存。  
[root@localhost ~]# more ps001.txt  
这里是把所有进程显示出来，并输出到ps001.txt文件，然后再通过more 来分页查看。  
  
  
4、kill 终止（杀死）进程，有十几种控制进程的方法，下面是一些常用的方法:  
[root@localhost ~]#kill -STOP [pid]  
发送SIGSTOP (17,19,23)停止一个进程，而并不消灭这个进程。  
[root@localhost ~]#kill -CONT [pid]  
发送SIGCONT (19,18,25)重新开始一个停止的进程。  
[root@localhost ~]#kill -KILL [pid]  
发送SIGKILL (9)强迫进程立即停止，并且不实施清理操作。  
[root@localhost ~]#kill -9 -1  
终止你拥有的全部进程。  
SIGKILL 和 SIGSTOP 信号不能被捕捉、封锁或者忽略，但是，其它的信号可以。所以这是你的终极武器。  
  
############################################################  
  
利用ps命令向管理员报告执行中的程序：  
  
有时候系统管理员可能只关心现在系统中运行着哪些程序，而不想知道有哪些进程在运行。由于一个应用程序可能需要启动多个进程。所在在同等情况下，进程的数 量要比程序多的多。为此从阅读方面考虑，管理员需要知道系统中运行的具体程序。要实现这个需求的话，就需要利用命令ps来帮忙。  
  
一、ps命令显示结果的含义。  
  
当需要查看系统中执行的程序时，虽然ps 命令不是唯一的命令，但绝对是使用的最频繁的命令。如下图所示，就是执行ps命令后显示的结果。  
  
在命令行中输入命令ps，就可以显示系统中当前运行的所有应用程序。如上图所示，如果输入ps命令，其显示结果主要有四部分内容。首先是PID，这是程序 的ID号。才作系统就是利用这个ID号来唯一的标识应用程序，而不是利用命令来辨认。当需要强制关闭应用程序时，就需要用到这个PID号码。其次是 TTY，这个字段表示用户使用的终端代码。Pts表示用户是采用远程登录的。第三个参数TIME表示这个程序所消耗的CPU时间，注意这个时间不是程序开 始运行的时间。最后一个参数CMD就表示程序的名字。  
  
二、让系统报告详细的信息。  
  
在使用ps命令时，如果不采用任何的可选项，则其显示的信息是非常有限的，而且往往只显示当前用户所运行的程序。当系统管理员需要知道应用程序更加详细的 运行信息时，如想要知道这个应用程序内存、CPU的占用率情况时，那么就需要加入一些可选项。如系统管理员需要一并查看其它用户所执行的应用程序时，就需 要在这个命令后面采用可选项-al。如此的话，系统会列出系统中所有用户运行的所有程序。如想要知道某个程序CPU与内存的使用情况，而不是只简单的现实 其CPU的使用时间，那么就需要在这个命令后面加入参数-l ，即使用ps –l命令可以让系统显示出应用程序的详细运行信息。关于上面各个字段所代表的含义，大家若有兴趣的话可以查看相关的帮助。一般来说，系统管理员关心的只是 程序的PID号码、内存与CPU的使用率、命令的名字、用户采用的终端等等。其它的信息对于系统管理员来说价值不是很大。  
  
三、查看后台运行的程序。  
  
默认情况下，ps命令只显示前台运行的程序，而不会显示后台运行的程序。但是并非所有的程序都是在前台运行。正常情况下，隐藏在后台运行的程序数量要比前 台运行的程序多的多。如随着操作系统启动而启动的不少系统自带程序，其运行的方式都是后台运行。而且有时候，系统出现问题往往是由于后台程序所造成的。如 常见的木马等程序都是在后台所运行的。为此系统管理员相对来说，更加想要知道在后台运行着哪些程序。  
  
如果要查看后台运行的程序，那就比较复杂一点。因为在不同版本的Linux操作系统中，要显示后台进程其所采用的可选项是不同的。如在红帽子 Linux操作系统中，其实采用参数的形式而不是可选项。即采用ps aux命令可以显示出所有的应用程序(包括前台与后台的)。参数与可选项的差异主要在前面又没有这个-符号。如果带有这个符号的就表示这是一个可选项。而 如果不带的，就表示这事一个参数。这个符号一般情况下可不能够省。在该写的地方没写，或者不需要些的地方偏偏加上了，则系统都会提示错误信息，说找不到这 个命令。而在其它的Linux系统版本中，可能不能够识别这个aux参数。如在一些Linux操作系统版本中，需要采用-a可选项来完成这个任务。由于系 统版本之间的差异，给系统管理员带来了不少的麻烦。不过值得庆幸的是，各个操作系统版本中都有在线的帮助。如果系统管理员在使用一个新版本的操作系统时， 不知道要显示全部进程该使用哪个可选项时，可以利用ps --heip等命令来查看系统帮助。不过美中不足的是，系统在线帮助都是英文的，对系统管理员的英文水平是一个不小的考验。不过如果要作Linux系统管 理员，这点英文底子还是要有的。因为最新的Linux技术基本上都是先出来英文文档的。其实要掌握最先进的操作系统书籍，大部分的计算机书籍都是英文的。  
  
四、对程序列表进行排序。  
  
当运行的应用程序比较多时，系统管理员需要对应用程序进行排序。Ps命令的排序功能是比较强的。主要是因为这个命令有一个--sort参数(注意在这个参 数前面采用的是两个小横杆符号，各位读者不要以为是笔者写错了)。在这个参数后面加上系统管理员想要的排序字段，就可以进行排序了。如这个命令ps –A --sort cmd，就表示显示系统所有的应用程序，并根据程序命令来进行排序。在Linux操作系统参数中，还有一个比较麻烦的事情，就是参数大小写不同往往代表着 不同的含义。如上面这个命令，将大写字母A换成是小写字符a，则结果就完全两样了。大写字符A表示所有的应用程序，而小写字符a则表示“all w/ tty except session leaders”。两者有本质的区别。通过这个差异可以用来过滤不同终端登陆帐户所运行的应用程序。  
  
在ps命令中，不少参数都有这种情况。运行ps --help查看这个命令的所有参数，就可以看到类似的大小写不同代表不同含义的情况还有很多。如大写字符O与小写字符o、大写字符U与小写字符u等等。 这些差异无形之中增加了系统管理员维护操作系统的难度。系统管理员要掌握这么多的参数以及参数大小写之间的差异，往往是不可能的。系统管理员除了平时要多 使用使用常用的参数来增加值观的印象，这个系统命令的在线帮助确实也是少不了的。这个系统帮助可以给管理员使用系统命令提供在线的指导。  
  
五、报告特定程序的运行情况。  
  
当系统中运行的程序比较多时，通过对程序名字排序可以帮助管理员找到自己所关心的程序。但是这仍然不是最简便的方式。如现在系统管理员在其它操作系统中发 现有一个叫做threadx的木马程序在系统后台运行。为此管理员需要在其它电脑上查看是否也有这个木马程序在运行。此时该如何处理呢?利用排序功能，对 程序的名字进行排序(注意不是对程序的PID进行排序，因为即使程序相同，启动的时间不同或者操作系统中已经启动程序的数量不同，这个PID号码也就不 同。也就是说这个PID号码是自动生成的)。这在一定程度上可以帮助管理员加快程序查找的速度。不过笔者认为，如果系统管理员能够使用管道符与 grep等查询命令可能会更快的找到自己所需要的应用程序信息。  
  
如现在系统管理员若使用这个命令ps aux |grep “threadx”，会出现什么情况呢?首先系统会列出当前系统中所有运行的应用程序(包括前台运行与后台运行的)。然后将输出的结果通过管道符号|传递 给命令grep。然后命令grep就会在ps命令输出的结果中查找是否有threadx这个程序运行的信息。如果有的话，则会在窗口中显示这个程序的信 息，而过滤掉其它应用程序的运行信息。可见采用管道符跟其它查询命令结合，可以帮助系统管理员在最短的时间内找到自己想了解的应用程序的信息。  
  
跟一些特殊符号结合还可以实现一些更加负责的功能。如跟cat命令结合使用可以统计出符合条件的程序数量。如在命令后面加入>符号，可以将输出的结 果定位到一个文件中。如跟通配符结合使用，可以查询出满足特定条件的一类程序。如还可以查询出某个程序父程序或者其子程序的运行信息等等。