**ps 命令**是最常用的监控进程的命令，通过此命令可以查看系统中所有运行进程的详细信息。  
  
ps 命令有多种不同的使用方法，这常常给初学者带来困惑。在各种 Linux 论坛上，询问 ps 命令语法的帖子屡见不鲜，而出现这样的情况，还要归咎于 UNIX 悠久的历史和庞大的派系。在不同的 Linux 发行版上，ps 命令的语法各不相同，为此，Linux 采取了一个折中的方法，即融合各种不同的风格，兼顾那些已经习惯了其它系统上使用 ps  命令的用户。  
  
ps 命令的基本格式如下：

[root@localhost ~]# ps aux  
#查看系统中所有的进程，使用 BS 操作系统格式  
[root@localhost ~]# ps -le  
#查看系统中所有的进程，使用 Linux 标准命令格式

选项：

* a：显示（当前用户）一个终端的所有进程；
* u：显示进程的归属用户及内存的使用情况；
* x：显示所有包含其他使用者的进程；
* -l：长格式显示更加详细的信息；
* -e：显示系统内所有进程；

可以看到，ps 命令有些与众不同，它的部分选项不能加入"-"，比如命令"ps aux"，其中"aux"是选项，但是前面不能带“-”。  
  
大家如果执行 "man ps" 命令，则会发现 ps 命令的帮助为了适应不同的类 UNIX 系统，可用格式非常多，不方便记忆。所以，我建议大家记忆几个固定选项即可。比如：

* "ps aux" 可以查看系统中所有的进程；
* "ps -le" 可以查看系统中所有的进程，而且还能看到进程的父进程的 PID 和进程优先级；
* "ps -l" 只能看到当前 Shell 产生的进程；

有这三个命令就足够了，下面分别来查看。  
  
【例 1】

[root@localhost ~]# ps aux

#查看系统中所有的进程

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

root 1 0.0 0.2 2872 1416 ? Ss Jun04 0:02 /sbin/init

root 2 0.0 0.0 0 0 ? S Jun04 0:00 [kthreadd]

root 3 0.0 0.0 0 0 ? S Jun04 0:00 [migration/0]

root 4 0.0 0.0 0 0 ? S Jun04 0:00 [ksoftirqd/0]

…省略部分输出…

表 1 中罗列出了以上输出信息中各列的具体含义。

|  |  |
| --- | --- |
| 表 1 ps命令输出信息含义 | |
| **表头** | **含义** |
| USER | 该进程是由哪个用户产生的。 |
| PID | 进程的 ID。 |
| %CPU | 该进程占用 CPU 资源的百分比，占用的百分比越高，进程越耗费资源。 |
| %MEM | 该进程占用物理内存的百分比，占用的百分比越高，进程越耗费资源。 |
| VSZ | 该进程占用虚拟内存的大小，单位为 KB。 |
| RSS | 该进程占用实际物理内存的大小，单位为 KB。 |
| TTY | 该进程是在哪个终端运行的。其中，tty1 ~ tty7 代表本地控制台终端（可以通过 Alt+F1 ~ F7 快捷键切换不同的终端），tty1~tty6 是本地的字符界面终端，tty7 是图形终端。pts/0 ~ 255 代表虚拟终端，一般是远程连接的终端，第一个远程连接占用 pts/0，第二个远程连接占用 pts/1，依次増长。 |
| STAT | 进程状态。常见的状态有以下几种：   1. -D：不可被唤醒的睡眠状态，通常用于 I/O 情况。 2. -R：该进程正在运行。 3. -S：该进程处于睡眠状态，可被唤醒。 4. -T：停止状态，可能是在后台暂停或进程处于除错状态。 5. -W：内存交互状态（从 2.6 内核开始无效）。 6. -X：死掉的进程（应该不会出现）。 7. -Z：僵尸进程。进程已经中止，但是部分程序还在内存当中。 8. -<：高优先级（以下状态在 BSD 格式中出现）。 9. -N：低优先级。 10. -L：被锁入内存。 11. -s：包含子进程。 12. -l：多线程（小写 L）。 13. -+：位于后台。 |
| START | 该进程的启动时间。 |
| TIME | 该进程占用 CPU 的运算时间，注意不是系统时间。 |
| COMMAND | 产生此进程的命令名。 |

【例 2】"ps aux"命令可以看到系统中所有的进程，"ps -le"命令也能看到系统中所有的进程。由于 "-l" 选项的作用，所以 "ps -le" 命令能够看到更加详细的信息，比如父进程的 PID、优先级等。但是这两个命令的基本作用是一致的，掌握其中一个就足够了。

[root@localhost ~]# ps -le

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

4 S 0 1 0 0 80 0 - 718 - ? 00:00:02 init

1 S 0 2 0 0 80 0 - 0 - ? 00:00:00 kthreadd

1 S 0 3 2 0 -40 - - 0 - ? 00:00:00 migration/0

1 S 0 4 2 0 80 0 - 0 - ? 00:00:00 ksoflirqd/0

1 S 0 5 2 0 -40 - - 0 - ? 00:00:00 migration/0

…省略部分输出…

表 2 罗列出以上输出信息中各列的含义。

|  |  |
| --- | --- |
| 表 2 ps -le 命令输出信息 | |
| **表头** | **含义** |
| F | 进程标志，说明进程的权限，常见的标志有两个:   * 1：进程可以被复制，但是不能被执行； * 4：进程使用超级用户权限； |
| S | 进程状态。具体的状态和"psaux"命令中的 STAT 状态一致； |
| UID | 运行此进程的用户的 ID； |
| PID | 进程的 ID； |
| PPID | 父进程的 ID； |
| C | 该进程的 CPU 使用率，单位是百分比； |
| PRI | 进程的优先级，数值越小，该进程的优先级越高，越早被 CPU 执行； |
| NI | 进程的优先级，数值越小，该进程越早被执行； |
| ADDR | 该进程在内存的哪个位置； |
| SZ | 该进程占用多大内存； |
| WCHAN | 该进程是否运行。"-"代表正在运行； |
| TTY | 该进程由哪个终端产生； |
| TIME | 该进程占用 CPU 的运算时间，注意不是系统时间； |
| CMD | 产生此进程的命令名； |

【例 3】如果不想看到所有的进程，只想查看一下当前登录产生了哪些进程，那只需使用 "ps -l" 命令就足够了：

[root@localhost ~]# ps -l

#查看当前登录产生的进程

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

4 S 0 18618 18614 0 80 0 - 1681 - pts/1 00:00:00 bash

4 R 0 18683 18618 4 80 0 - 1619 - pts/1 00:00:00 ps

可以看到，这次从 pts/1 虚拟终端登录，只产生了两个进程：一个是登录之后生成的 Shell，也就是 bash；另一个是正在执行的 ps 命令。

# Linux nice和renice命令：改变进程优先级

当 Linux 内核尝试决定哪些运行中的进程可以访问 CPU 时，其中一个需要考虑的因素就是进程优先级的值（也称为 nice 值）。每个进程都有一个介于 -20 到 19 之间的 nice 值。默认情况下，进程的 nice 值为 0。  
  
进程的 nice 值，可以通过 nice 命令和 renice 命令修改，进而调整进程的运行顺序。

## nice命令

在linux中，nice命令用于改变进程的优先级。

nice命令？什么鬼？哪有命令直接说自己“nice”的？咋一看，这个名字确实很“nice”，只是这里的“nice”是指“niceness”，即友善度、谦让度。用于进程中，表示进程的优先级，也即进程的友善度。niceness值为负时，表示高优先级，能提前执行和获得更多的资源，对应低友善度；反之，则表示低优先级，高友善度。

nice命令格式：nice [OPTION] [COMMAND [ARG]...]。

当nice没有选项时，输出值表示系统进程缺省的niceness值，一般为0。

xie@xie-VirtualBox:~$ nice

0

当nice命令中没有给出具体的niceness值时，默认为10。如nice vi设置vi进程的niceness为10。

**-n 选项**

-n选项后面跟具体的niceness值。niceness值的范围-20~19，小于-20或大于19的值分别记为-20和19。

xie@xie-VirtualBox:~$ nice -n 10 vi&

该命令设置vi进程的niceness值为10，也就是较低的优先级。

如果设置的niceness值为负，那么必须要有管理员权限。这也不难理解，当niceness为负时，意味着该进程要抢占其他进程的资源，必须要有权限才行；如果niceness为正，即表示谦让度高，这对其他进程来说是喜闻乐见的，也当然就不用权限认证了。

可以通过ps -l查看进程的niceness值。

xie@xie-VirtualBox:~$ ps -l

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

0 S 1000 1635 1634 0 80 0 - 2178 wait pts/13 00:00:00 bash

0 T 1000 1677 1635 0 90 10 - 1767 signal pts/13 00:00:00 vi

0 R 1000 1678 1635 0 80 0 - 1606 - pts/13 00:00:00 ps

NI列即表示进程的niceness值。vi进程对应的NI值正好为刚设置的10。那PRI列又是什么呢？PRI表示进程当前的总优先级，值越小表示优先级越高，由进程默认的PRI加上NI得到，即PRI(new) = PRI(old) + NI。由上程序，进程默认的PRI是80，所以加上值为10的NI后，vi进程的PRI为90。  
所以，需要注意的是，NI即niceness的值只是进程优先级的一部分，不能完全决定进程的优先级，但niceness值的绝对值越大，效果越显著。

**--adjustment=N 选项**

--adjustment选项和-n选项的效果是一样的，在等号右边设置对应的niceness值即可。

xie@xie-VirtualBox:~$ nice --adjustment=16 vi&

[1] 1681

xie@xie-VirtualBox:~$ ps -l

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

0 S 1000 1635 1634 0 80 0 - 2178 wait pts/13 00:00:00 bas

0 T 1000 1681 1635 0 96 16 - 1767 signal pts/13 00:00:00 vi

0 R 1000 1682 1635 0 80 0 - 1606 - pts/13 00:00:00 ps

注意，直接使用“-N”也可以设置niceness值。比如，nice -12 vi& 将vi的niceness值设置为12，如果是nice --12 vi&的话，则设置niceness的值为-12。这很容易混淆，建议使用-n或--adjustment选项，不易出错。

## renice 命令

同 nice 命令恰恰相反，renice 命令可以在进程运行时修改其 NI 值，从而调整优先级。  
  
renice，字面意思即重新设置niceness值，进程启动时默认的niceness值为0，可以用renice更新。

renice语法：renice [优先等级][-g<程序群组名称>...][-p<程序识别码>...][-u <用户名称>...]

如：

renice -5 -p 5200 #将PID为5200的进程的niceness设为-5

renice -5 -u xie #将属于用户xie的进程的niceness设为-5

renice -5 -g group1 #将属于group1组的程序的niceness设为5

如何合理地设置进程优先级，曾经是一件让系统管理员非常费神的事情。但现在已经不是了，如何地 CPU 足够强大，能够合理地对进程进行调整，输入输出设备也远远跟不上 CPU 地脚步，反而在更多的情况下，CPU 总是在等待哪些缓慢的 I/O（输入/输出）设备完成数据的读写和传输任务。  
  
然而，手动设置进程的优先级并不能影响 I/O 设备对它的处理，这就意味着，哪些有着低优先级的进程常常不合理地占据着本就低效地 I/O 资源。