# ps

报告当前系统的进程状态

## 补充说明

**ps命令** 用于报告当前系统的进程状态。可以搭配kill指令随时中断、删除不必要的程序。ps命令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。

### 语法

ps(选项)

### 选项

-a：显示所有终端机下执行的程序，除了阶段作业领导者之外。  
a：显示现行终端机下的所有程序，包括其他用户的程序。  
-A：显示所有程序。  
-c：显示CLS和PRI栏位。  
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，选项或常驻服务的标示。  
-C<指令名称>：指定执行指令的名称，并列出该指令的程序的状况。  
-d：显示所有程序，但不包括阶段作业领导者的程序。  
-e：此选项的效果和指定"A"选项相同。  
e：列出程序时，显示每个程序所使用的环境变量。  
-f：显示UID,PPIP,C与STIME栏位。  
f：用ASCII字符显示树状结构，表达程序间的相互关系。  
-g<群组名称>：此选项的效果和指定"-G"选项相同，当亦能使用阶段作业领导者的名称来指定。  
g：显示现行终端机下的所有程序，包括群组领导者的程序。  
-G<群组识别码>：列出属于该群组的程序的状况，也可使用群组名称来指定。  
h：不显示标题列。  
-H：显示树状结构，表示程序间的相互关系。  
-j或j：采用工作控制的格式显示程序状况。  
-l或l：采用详细的格式来显示程序状况。  
L：列出栏位的相关信息。  
-m或m：显示所有的执行绪。  
n：以数字来表示USER和WCHAN栏位。  
-N：显示所有的程序，除了执行ps指令终端机下的程序之外。  
-p<程序识别码>：指定程序识别码，并列出该程序的状况。  
p<程序识别码>：此选项的效果和指定"-p"选项相同，只在列表格式方面稍有差异。  
r：只列出现行终端机正在执行中的程序。  
-s<阶段作业>：指定阶段作业的程序识别码，并列出隶属该阶段作业的程序的状况。  
s：采用程序信号的格式显示程序状况。  
S：列出程序时，包括已中断的子程序资料。  
-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。  
t<终端机编号>：此选项的效果和指定"-t"选项相同，只在列表格式方面稍有差异。  
-T：显示现行终端机下的所有程序。  
-u<用户识别码>：此选项的效果和指定"-U"选项相同。  
u：以用户为主的格式来显示程序状况。  
-U<用户识别码>：列出属于该用户的程序的状况，也可使用用户名称来指定。  
U<用户名称>：列出属于该用户的程序的状况。  
v：采用虚拟内存的格式显示程序状况。  
-V或V：显示版本信息。  
-w或w：采用宽阔的格式来显示程序状况。　  
x：显示所有程序，不以终端机来区分。  
X：采用旧式的Linux i386登陆格式显示程序状况。  
-y：配合选项"-l"使用时，不显示F(flag)栏位，并以RSS栏位取代ADDR栏位　。  
-<程序识别码>：此选项的效果和指定"p"选项相同。  
--cols<每列字符数>：设置每列的最大字符数。  
--columns<每列字符数>：此选项的效果和指定"--cols"选项相同。  
--cumulative：此选项的效果和指定"S"选项相同。  
--deselect：此选项的效果和指定"-N"选项相同。  
--forest：此选项的效果和指定"f"选项相同。  
--headers：重复显示标题列。  
--help：在线帮助。  
--info：显示排错信息。  
--lines<显示列数>：设置显示画面的列数。  
--no-headers：此选项的效果和指定"h"选项相同，只在列表格式方面稍有差异。  
--group<群组名称>：此选项的效果和指定"-G"选项相同。  
--Group<群组识别码>：此选项的效果和指定"-G"选项相同。  
--pid<程序识别码>：此选项的效果和指定"-p"选项相同。  
--rows<显示列数>：此选项的效果和指定"--lines"选项相同。  
--sid<阶段作业>：此选项的效果和指定"-s"选项相同。  
--tty<终端机编号>：此选项的效果和指定"-t"选项相同。  
--user<用户名称>：此选项的效果和指定"-U"选项相同。  
--User<用户识别码>：此选项的效果和指定"-U"选项相同。  
--version：此选项的效果和指定"-V"选项相同。  
--widty<每列字符数>：此选项的效果和指定"-cols"选项相同。

由于ps命令能够支持的系统类型相当的多，所以选项多的离谱！

### 实例

ps axo pid,comm,pcpu # 查看进程的PID、名称以及CPU 占用率  
ps aux | sort -rnk 4 # 按内存资源的使用量对进程进行排序  
ps aux | sort -nk 3 # 按 CPU 资源的使用量对进程进行排序  
ps -A # 显示所有进程信息  
ps -u root # 显示指定用户信息  
ps -efL # 查看线程数  
ps -e -o "%C : %p :%z : %a"|sort -k5 -nr # 查看进程并按内存使用大小排列  
ps -ef # 显示所有进程信息，连同命令行  
ps -ef | grep ssh # ps 与grep 常用组合用法，查找特定进程  
ps -C nginx # 通过名字或命令搜索进程  
ps aux --sort=-pcpu,+pmem # CPU或者内存进行排序,-降序，+升序  
ps -f --forest -C nginx # 用树的风格显示进程的层次关系  
ps -o pid,uname,comm -C nginx # 显示一个父进程的子进程  
ps -e -o pid,uname=USERNAME,pcpu=CPU\_USAGE,pmem,comm # 重定义标签  
ps -e -o pid,comm,etime # 显示进程运行的时间  
ps -aux | grep named # 查看named进程详细信息  
ps -o command -p 91730 | sed -n 2p # 通过进程id获取服务名称

将目前属于您自己这次登入的 PID 与相关信息列示出来

ps -l  
# UID PID PPID F CPU PRI NI SZ RSS WCHAN S ADDR TTY TIME CMD  
# 501 566 559 4006 0 31 0 4317620 228 - Ss 0 ttys001 0:00.05 /App...cOS/iTerm2 --server /usr/bin/login -fpl kenny /Ap...s/MacOS/iTerm2 --launch\_shel  
# 501 592 577 4006 0 31 0 4297048 52 - S 0 ttys001 0:00.63 -zsh

* 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。

列出目前所有的正在内存当中的程序

ps aux  
  
# USER PID %CPU %MEM VSZ RSS TT STAT STARTED TIME COMMAND  
# kenny 6155 21.3 1.7 7969944 284912 ?? S 二03下午 199:14.14 /Appl...OS/WeChat  
# kenny 559 20.4 0.8 4963740 138176 ?? S 二03下午 33:28.27 /Appl...S/iTerm2  
# \_windowserver 187 18.0 0.6 7005748 95884 ?? Ss 二03下午 288:44.97 /Syst...Light.WindowServer -daemon  
# kenny 1408 10.7 2.1 5838592 347348 ?? S 二03下午 138:51.63 /Appl...nts/MacOS/Google Chrome  
# kenny 327 5.8 0.5 5771984 79452 ?? S 二03下午 2:51.58 /Syst...pp/Contents/MacOS/Finder

* 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

列出类似程序树的程序显示

ps -axjf  
  
# USER PID PPID PGID SESS JOBC STAT TT TIME COMMAND UID C STIME TTY  
# root 1 0 1 0 0 Ss ?? 10:51.90 /sbin/launchd 0 0 二03下午 ??  
# root 50 1 50 0 0 Ss ?? 0:10.07 /usr/sbin/syslog 0 0 二03下午 ??  
# root 51 1 51 0 0 Ss ?? 0:29.90 /usr/libexec/Use 0 0 二03下午 ??

找出与 cron 与 syslog 这两个服务有关的 PID 号码

ps aux | egrep '(cron|syslog)'  
  
# root 50 0.0 0.0 4305532 1284 ?? Ss 二03下午 0:10.08 /usr/sbin/syslogd  
# kenny 90167 0.0 0.0 4258468 184 s007 R+ 9:23下午 0:00.00 egrep (cron|syslog)

把所有进程显示出来，并输出到ps001.txt文件

ps -aux > ps001.txt

输出指定的字段