

进程

1. 系统启动时，内核先把它的一些程序初始化为进程。然后运行 init 的程序。

2. 一个程序的运行可以触发其它程序的运行，在进程系统中这种情况被表述为父进程创建子进程，内核会保存每个进程的信息以确保任务有序进行。比如，每个进程将被分配一个称为进程 ID 的号码。进程 ID (PID) 是按递增的顺序来分配的，init 进程的 PID 始终为 1。

3. ps 命令查看进程信息

PID：进程编号

TTY：teletype 的缩写，代表了进程的控制终端。结果中的?表示这个进程没有控制终端

TIME：表示进程消耗的 CPU 时间总和

ps x 可以获得更多信息。注意 x 的前面没有-。这将告知 ps 显示所有进程，而不关心它们是哪个终端所控制的

ps x 命令中有一个新的列 STAT，是 state 的缩写，显示进程的当前状态

状态	状态的含义
R	运行状态。表示进程正在运行或准备运行
S	睡眠状态。进程不在运行，而是等待某事件发生
D	不可中断的睡眠状态。进程在等待 I/O 操作
T	暂停状态。后续还可以继续运行
Z	无效或僵尸进程。子进程被终止，但是还没被其父进程彻底释放
<	高优先级进程
N	低优先级进程
s	进程的领导者（其下面有子进程）
l	多进程的
+	位于后台的进程组

ps aux 显示属于每个用户的进程信息。使用这些选项时不带连字符前缀将使得命令以 BSD 模式运行。

标题	含义
USER	用户名，表示该进程的所有者
%CPU	CPU 使用的百分比
%MEM	内存使用百分比
VSZ	虚拟内存消耗大小。以 KB 为单位
RSS	实际使用的内存大小。以 KB 为单位
START	进程开启的时间。如果开启超过 24 小时，将用日期表示

4. top 命令动态查看进程信息

ps 命令是指该命令执行时刻的一个快照。要查看动态视图需要使用 top。默认每 3 秒更新一次。

```
top - 21:36:05 up 18:47, 1 user, load average: 0.00, 0.01, 0.05
Tasks: 146 total, 1 running, 145 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 999936 total, 386896 free, 221848 used, 391192 buff/cache
KiB Swap: 2097148 total, 2096124 free, 1024 used. 560896 avail Mem
```

行	字段	含义
1	top	程序名
	21:36:05	系统时间

	18:47	从机器启动开始正常运行的时间
	1 user	当前只有一个用户登陆
	load average	负载均值指等待运行的进程数，即共享 CPU 资源的处于可运行状态的进程数。显示三个值分别对应不同的时间段。第一个值是前 60 秒的均值，第二个值是前 5 分钟的均值，第三个值是前 15 分钟的均值。该值小于 1.0 表示机器并不忙碌
2	task	统计进程数以及各个进程的状态信息
3 %cpu(s)	0.0us	目前有 0.0% 的 CPU 时间被用户进程占用。此处指内核外进程
	0.0sy	目前有 0.0% 的 CPU 时间被系统进程（内核）占用
	0.0ni	目前有 0.0% 的 CPU 时间被低优先级进程占用
	100.0id	目前有 100% 的 CPU 资源是空闲的
	0.0wa	目前有 0.0% 彻底 CPU 时间用来等待 I/O 操作
24	Mem	内存的使用情况
5	Swap	交换分区（虚拟内存）的使用情况

q 可以退出 top 命令的执行。在 top 执行过程中按 h 可以查看帮助界面。

5. 控制进程

Ctrl+C 可以终止一个进程的执行。但是并不是所有的命令程序都可以使用这种方法来实现中断。

6. 使进程在后台运行

想要返回命令提示符，但是又不想终止程序。可以通过让程序在后台运行来实现。此时可以在命令的后面加上 & 来实现
`ping google.com > googlePing &`

7. jobs 命令可以查看从当前终端启动的所有作业

8. 使进程回到前台运行

`fg %作业编号`
 首先通过 jobs 查看作业编号，之后通过 fg 命令可以将一个后台进程回到前台。
 如 `fg %1`
 当一个进程回到前台，绝大多数情况下可以使用 Ctrl+C 将其终止

9. 停止（暂停）进程和恢复进程

Ctrl+Z 可以暂停一个进程的执行。
 当进程暂停后，我们可以使用 fg 将其在前台恢复运行。
 也可以使用 bg 命令让进程移到后台运行。
 如果用命令的方式启动了一个进程但是忘记了加 &，可以使用 bg 将其移动到后台。
 通过命令执行程序而不是图形界面可以看到更多的提示信息，找到问题所在。

10. 信号

kill 命令用于给进程发送信号。Ctrl+C 和 Ctrl+Z 就是信号，前者发送一个 INT 中断信号，后者发送 TSTP 终端暂停信号。语法：
`kill [-signal] pid signal` 表示信号编号

常用信号

信号编号	信号名	含义
1	HUP	挂起信号。当一个后台进程接收到信号时，它重启并将重新读取它的配置文件

2	INT	中断信号。执行效果和 Ctrl+C 一样，通常用来终止一个程序。
9	KILL	杀死信号。进程实际上可以忽略所有信号，但是 KILL 信号并不会真正发给进程。而是由内核处理。此时内核会立刻终止该进程。当进程以这种方式被终止时，它将没有机会对自己进行清理。考虑到这个原因，KILL 信号只能当作其它的终端信号都失败的情况下的最后选择。
15	TERM	终止信号。这是 kill 命令默认发送的信号类型。如果程序依然有 alive enough 来接收信号，那么它将终止。
18	CONT	继续运行信号，恢复之前接受了 STOP 信号的进程。
19	STOP	暂停信号。该信号不会发送给进程，而是发给内核，因此它不能忽略。

kill -l 可以列出所有信号。

除了常用信号外，还有一些信号经常被系统使用。如

信号编号	信号名	含义
3	QUIT	退出信号
11	SEGV	段错误信号。如果程序使用了非法内存空间，则系统发出该信号
20	TSTP	终端暂停信号。按下 Ctrl+Z 时发出该信号。与 STOP 不同，TSTP 由程序接收
28	WINCH	窗口改变信号。当窗口大小改变时，系统发送该信号。

我们也可以通过信号名来指定信号，其中包含带有 SIG 前缀的信号名。SIG 是 signal 的缩写。

kill -2 pid 等价于 kill INT pid 等价于 kill SIGINT pid

注意用户只能给自己的进程发送信号，root 用户可以给所有用户的进程发送信号

11. killall 命令发送信号给多个进程

语法: killall [-u user] [-signal] name...

例: killall xlogo 会终止所有的 xlogo 实例。

12. 其它进程相关的命令

- pstree 以树状的模式输出进程列表。可以显示进程间的父子关系
- vmstat 输出系统资源使用情况的快照，如内存，swap 和磁盘等。如果需要持续查看输出，可以在命令后面加上一个间隔时间，以秒为单位，如 vmstat 5
- xload 用来绘制显示系统时间负载情况图形的一种图形化界面程序
- tload 类似于 xload，但是图形在终端上控制。Ctrl+C 终止输出。