

## 26 进程操作和系统重启

更新时间：2019-07-18 12:00:02



“

合理安排时间，就等于节约时间。

——培根

”

### 内容简介

1. 前言
2. top: 进程的动态列表
3. Ctrl + C 和 kill 命令：停止进程
4. halt 命令和 reboot 命令：停止和重启系统
5. 总结

### 1. 前言

上一课监视系统活动和查看进程，滴水不漏，我们介绍了 w 和 ps 两个命令，开启了监视系统活动和操作进程的篇章。

这一课我们将继续讲解。这一课特别简单，可以说是本系列课程里最轻松的课之一。

### 2. top: 进程的动态列表

ps 命令虽然强大，但有一个缺陷：是静态的，只能记录当下那一刻的进程列表。

但事实上，我们的系统是时刻变化的，有些进程这一秒在运行，下一秒可能就被终止了。所以 ps 命令并不能帮助我们实时监控系统的进程。

怎么才能获取一个随时间一直在变的进程列表呢？就需要用到 top 命令了。

```

top - 22:09:21 up 38 min, 1 user, load average: 0.02, 0.01, 0.00
Tasks: 172 total, 1 running, 138 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.3 us, 0.3 sy, 0.0 ni, 99.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 2041088 total, 508840 free, 687824 used, 844424 buff/cache
KiB Swap: 969960 total, 969960 free, 0 used, 1177524 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 1091 oscar     20   0 125776   2308   1928 S   0.3   0.1   0:03.89 VBoxClient
 1132 oscar     20   0 2977792 257928 107744 S   0.3  12.6   0:10.68 gnome-shell
 1802 oscar     20   0 799340   36480  27300 S   0.3   1.8   0:01.25 gnome-termina+
    1 root      20   0 159844   9092   6652 S   0.0   0.4   0:01.09 systemd
    2 root      20   0      0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    6 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-+
    8 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
    9 root      20   0      0      0      0 S   0.0   0.0   0:00.12 ksoftirqd/0
   10 root      20   0      0      0      0 I   0.0   0.0   0:00.16 rcu_sched
   11 root      20   0      0      0      0 I   0.0   0.0   0:00.00 rcu_bh
   12 root      rt    0      0      0      0 S   0.0   0.0   0:00.00 migration/0
   13 root      rt    0      0      0      0 S   0.0   0.0   0:00.00 watchdog/0
   14 root      20   0      0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   15 root      20   0      0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
   16 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 netns

```

这个列表是交互性的，而且实时更新。

第一行，你可以看到当前时间（22:09:21，就是 22 点 09 分 21 秒），系统运行时间（up 38 min，也就是 38 分钟），用户数目（1 user，就是“一个用户”），平均负载（load average: 0.02, 0.01, 0.00）。

接下来的几行，我们不过多解释了，因为这涉及到操作系统的一些概念。

在下面，[PID, USER, PR, NI, VIRT, RES, SHR, S, %CPU, %MEM, TIME+, COMMAND] 开始的那个列表就是进程列表了。

但为什么相比之前 `ps -ef` 时进程少了这么多呢？

因为 `top` 命令无法显示全部的进程。它只显示排在前面的一些进程，为了整个信息区域正好占据终端中一个页面的大小。

默认的，这些进程是按照使用处理器的比率来排序的，从大到小（按照 %CPU 那一列的数值），最消耗处理器的进程排在最前面。所以如果你觉得自己的系统变慢了，过载了，查看列表中最开始的几个进程是个好主意。

前面说过了，`top` 命令输出的信息页面不但是动态变化的，而且是可交互的。

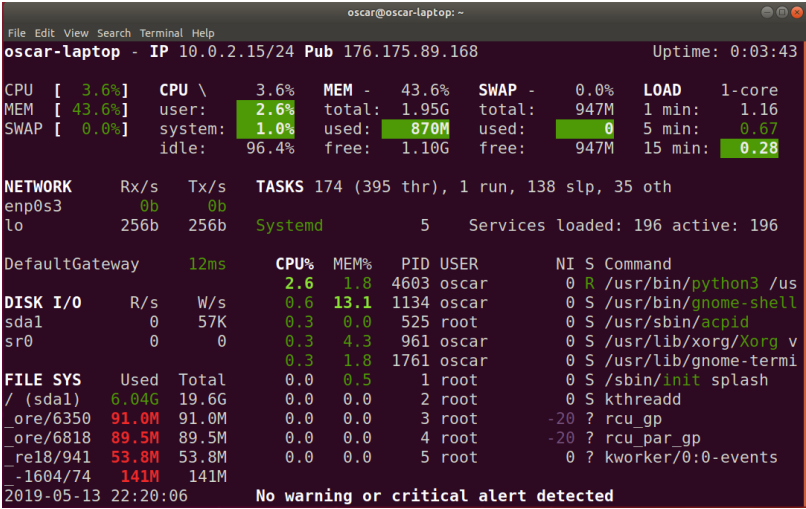
我们用一些键盘的按键可以控制 `top` 的显示结果。

- q: 退出 top。
- h: 显示帮助文档，也就是哪些按键可以使用。按下任意键返回，按 q 回到 top 命令的主界面。
- B: 大写的 B，加粗某些信息。
- f: 在进程列表中添加或删除某些列。按 q 回到 top 命令的主界面。
- F: 改变进程列表排序所参照的列。默认情况下，是按照 %CPU 那一列来排序，按 q 回到 top 命令的主界面。
- u: 依照用户来过滤显示。可以输入用户名，按回车。
- k: 结束某个进程。会让你输入要结束的进程的 PID。
- s: 改变刷新页面的时间。默认的，页面每隔 3 秒刷新一次。

当然了，Linux 下有很多优秀的软件，做得比 `top` 和 `ps` 更好，而且“颜值”也更高。例如：[glances](#) 和 [htop](#)。大家可以安装试试（`sudo apt install glances` 和 `sudo apt install htop`），我就安装了 `glances` 这个软件，`htop` 也是不错的。

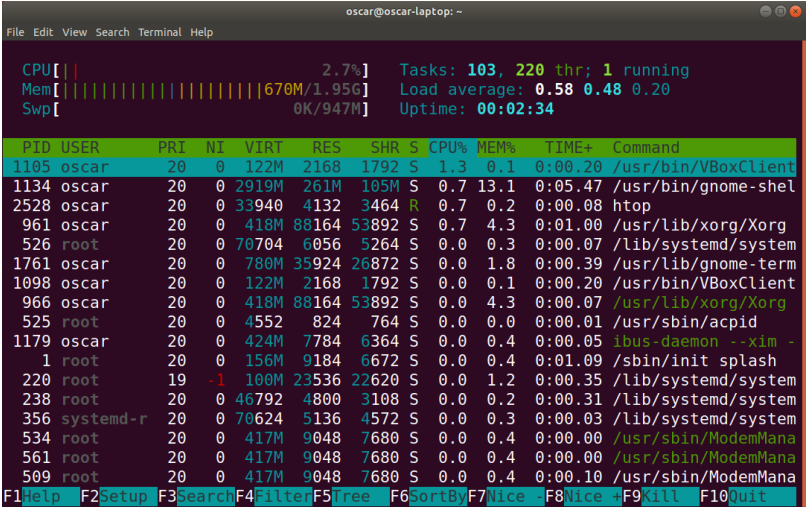
运行 glances :

```
glances
```



运行 htop 试试:

```
htop
```



### 3. Ctrl + C 和 kill 命令：停止进程

有时候，系统会突然卡住。这在 Linux 中也是会发生的。

在 Linux 中停止进程有几种方法：

#### Ctrl + C：停止终端中正在运行的进程

Ctrl + C 这个组合快捷键还是需要知道的，因为它可以比较友好地中止终端中正在运行的程序（进程），这个组合键也只在终端中会这样执行，如果在普通的图形界面中，它就是拷贝的快捷键，与 Windows 下无异。

顺便讲一个 Linux 下的小技巧，比如我们在终端下，用鼠标选中一段文字，然后按一下鼠标中间的滚轮，就会在终端里粘贴选中的文字。

而在终端中拷贝和粘贴，需要用“Ctrl + Shift + C”和“Ctrl + Shift + V”来执行。

## kill 命令：结束一个进程

Ctrl + C 这个组合键只能作用于当前终端中正在运行的程序，而 Linux 中，很多进程是你看不到的，在后台运行。

这时候就需要请出 kill 命令了。

kill 是英语“杀死”的意思，虽然有点“小暴力”，但是这个命令却很有用。

kill 命令后接需要结束的进程号，也就是之前我们看到过的 PID。是一个数字，我们可以通过 ps 命令或 top 命令来获知进程的 PID。

例如：

```
kill 8461
```

我们也可以同时用 kill 来结束好几个进程，只要用空格隔开它们的 PID：

```
kill 8461 8561 1706
```

有时候，kill 命令虽然作用于一个进程，但是它不终止，卡住了（因为 kill 命令会很友善地结束进程）。那怎么办呢？有没有比较简单粗暴的方式呢？

是有的，我们可以用 kill -9 来立即强制结束进程。比如：

```
kill -9 7291
```

立即结束 PID 是 7291 的进程，非常“野蛮粗暴”。

## killall：结束多个进程

记得我们之前提过，同一个程序运行时可能启动多个进程。如果你想一下子结束所有这些进程，怎么做呢？

当然，你可以用 kill + PID 的方式一个个地结束这些进程，但是我们有一个更快捷的命令：killall。

all 是英语“全部”的意思，所以 killall 命令就是用于结束全部要结束的进程。

不同于 kill 命令，killall 命令后接程序名，而不是 PID（进程号）。

假设现在我开了好几个图形终端，在每一个中我都运行 find 命令进行文件的查找。

这样，我用 `ps -u oscar | grep find` 就可以搜索出好多个 find 的进程。

如果我要结束所有这些 find 进程，可以这么做：

```
killall find
```

## 4. halt 命令和 reboot 命令：停止和重启系统

既然我们刚学了如何用 kill 命令来结束进程，那么现在我们该来学习一下如何关闭和重启整个 Linux 系统了吧。

虽然你肯定已经知道如何用图形界面的操作方式来关闭和重启系统了。但是用命令行，也许你还不知道怎么做吧。

## halt: 关闭系统

需要以 root 身份才能关闭系统，所以我们可以这么做：

```
sudo halt
```

## reboot: 重启系统

同样地，需要以 root 身份才能重启系统，我们可以这么做：

```
sudo reboot
```

实际上，halt 和 reboot 这两个命令都调用了 Linux 的另一个命令：shutdown

大家可以阅读 shutdown 命令的手册，也可以自行去网上搜索一下如何用 shutdown 命令来实现定时关机或重启（有很多教程）。

算是留给大家的小作业了。

其实还有一个命令：poweroff 也可以用于关机，而且非常有效率。直接运行即可关机。

```
poweroff
```

### 小结

1. top 命令是动态可交互的，会实时更新页面。
2. 组合键 Ctrl + C 结束当前终端中正在运行的程序。
3. kill 命令用于结束进程，后接进程的 PID（一个数字）。如果 kill 这个比较友好的结束命令不起作用，进程依旧卡住，那么我们可以使用 kill -9 来强制结束进程。
4. halt 命令关闭系统，reboot 命令重启系统，poweroff 命令可以实现立即关机。

今天的课就到这里，一起加油吧！

