



链滴

Linux - 进程管理 (一)

作者: [douniwan](#)

原文链接: <https://ld246.com/article/1573828803578>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



进程是个什么玩意？

1. 在Linux中，每一个执行的程序都称为一个进程。每一个进程都分配一个ID号。
2. 每一个进程，都会对应一个父进程，而这个父进程可以复制多个子进程。从这一点其实还可以推测，进程在Linux中是树型结构。
3. 每一个进程都可能以两种方式存在的。前台与后台，所谓前台进程就是用户目前屏幕上可以进行操作的。后台进程则是实际在操作，但由于屏幕上无法看到的进程，通常使用后台的方式执行。
4. 一般系统的服务都是后台进程的方式存在，而且会常驻在系统中，直到关机才结束。

<!--more-->

如何查看进程

一般我们会使用三条命令来查看进程，分别是`ps`，`pstree`，`top`。

其中`ps`命令可以查看各种进程信息，`pstree`查看进程树的情况，然后`top`的作用比较强大，可以查看个进程资源的占用状况。

ps

用法 `ps -选项`

选项说明

-a: 显示当前终端的所有进程信息

-u: 以用户的形式显示进程信息

-x: 显示后台进程运行的参数

-e: 显示所有进程

-f: 全格式

我们以命令 `ps -ef` 的输出的一部分例子作为分析

huny@huny-PC:~\$ ps -ef #以全格式显示当前所有的进程

```
UID      PID  PPID  C  STIME TTY      TIME CMD
root      9    2    0  17:55 ?        00:00:00 [rcu_bh]
root    5431    1    0  17:58 ?        00:00:00 /usr/lib/upower/upowerd
root   11470  3131    5  18:54 tty1    00:06:05 /usr/lib/xorg/Xorg -background none :0 -seat se
t0 -auth /var/run/lightdm/root/:0 -nolisten tcp vt1 -novtswitch
root   11544  3131    0  18:54 ?        00:00:00 lightdm --session-child 13 22
huny   11632    1    0  18:54 ?        00:00:00 /lib/systemd/systemd --user
huny   11633 11632    0  18:54 ?        00:00:00 (sd-pam)
huny   11658    1    0  18:54 ?        00:00:00 /usr/bin/dbus-daemon --fork --print-pid 5 --print-
ddress 7 --session
```

UID是用户名（执行改线程的用户）。

PID是该线程在系统中的唯一标识。

PPID是该进程的父进程的ID。

C是CPU用于计算执行优先级的因子，数值越大，表明进程是CPU密集型运算，执行优先级会降低；值越小，表明进程是I/O密集型运算，执行优先级会提高。

STIME是进程启动的时间。

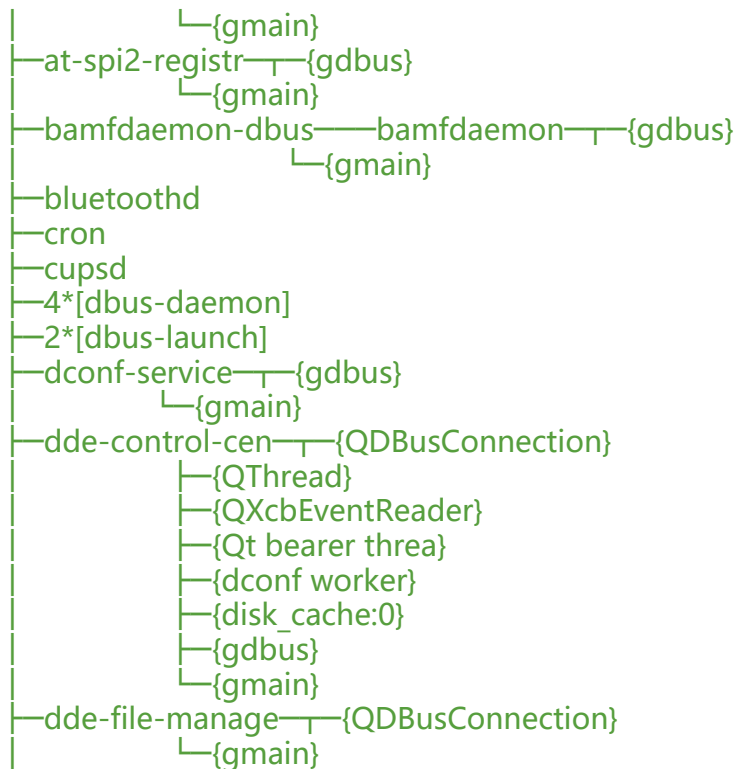
TIME是累积的 CPU 时间（user+system）

CMD是启动进程所用的命令和参数。

pstree

如果需要查看进程树的状况的时候，我们就可以使用pstree命令。下面我截取了部分我电脑上输入pstree的信息。

```
huny@huny-PC:~$ pstree
systemd├─ModemManager├─{gdbus}
│   │   └─{gmain}
│   └─NetworkManager├─dhclient
│   │   │   └─{gdbus}
│   │   └─{gmain}
│   └─QQProtect.exe──15*[{QQProtect.exe}]
│   └─TIM.exe──55*[{TIM.exe}]
│   └─accounts-daemon├─{gdbus}
│   │   └─{gmain}
│   └─acpid
│   └─at-spi-bus-laun├─dbus-daemon
│   │   │   └─{dconf worker}
│   │   └─{gdbus}
```



其实可以从这个输出中（自己输出看一下整体的结构）看出，系统中的所有的进程的父进程都是systemd。

如果觉得在终端上显示观看不太方便，可以使用输出重定向将结果输出到文件。

例如 `ps tree > out.txt` 这样就可以在当前目录下查看out.txt来观察ps tree命令的结果。

top

top命令真的十分的强大，当我们需要查看系统中各个进程的资源占用情况，那么使用这个命令绝对会让我们欣喜的。

直接在终端输入ps命令进入top视图（部分输出如下）。

```

top - 21:09:13 up 3:13, 1 user, load average: 1.44, 1.60, 1.52
Tasks: 296 total, 2 running, 221 sleeping, 0 stopped, 1 zombie
%Cpu(s): 8.5 us, 5.1 sy, 0.0 ni, 86.1 id, 0.0 wa, 0.0 hi, 0.3 si, 0.0 st
KiB Mem : 16292804 total, 9736652 free, 2827132 used, 3729020 buff/cache
KiB Swap: 8388604 total, 8388604 free, 0 used. 12489240 avail Mem

  PID USER  PR  NI  VIRT  RES  SHR S %CPU %MEM  TIME+ COMMAND
14229 huny  20   0 2165636 287516 63520 S 16.3 1.8 15:45.71 TIM.exe
13905 huny  20   0 5900760 269560 164456 S 14.0 1.7 16:53.91 netease-cloud-m
14054 huny  20   0  9380  6732  1980 S 11.6 0.0  9:55.58 wineserver.real
11470 root   20   0 587440 166508 134444 S  8.5 1.0  7:22.55 Xorg
11809 huny  20   0 3468336 103444 76484 S  8.5 0.6  7:10.48 kwin_x11
14531 huny  20   0 976664 229780 135208 S  3.9 1.4  9:22.44 chrome
15352 huny  20   0 1568616 135040 89420 S  3.9 0.8  5:47.90 Typora
16832 uny    20   0 1849376 175768 82316 S  3.9 1.1  3:34.45 Typora
22383 huny  20   0  46812  3876  3180 R  0.8 0.0  0:01.00 top

```

```
1 root    20  0 204896 7168 5208 S  0.0 0.0 0:02.95 systemd
2 root    20  0    0    0    0 S  0.0 0.0 0:00.03 kthreadd
4 root    0 -20    0    0    0 I  0.0 0.0 0:00.00 kworker/0:0H
6 root    0 -20    0    0    0 I  0.0 0.0 0:00.00 mm_percpu_wq
7 root    20  0    0    0    0 S  0.0 0.0 0:00.45 ksoftirqd/0
8 root    20  0    0    0    0 I  0.0 0.0 0:22.53 rcu_sched
```

第一行

系统的一些基本信息

- 21:09:13: 当前系统时间
- 3:13: 系统已经开机3小时13分钟了
- 1 user: 当前只有一个用户登录
- load average: 后面三个数分别是1分钟、5分钟、15分钟的负载情况。
- load average数据是每隔5秒钟检查一次活跃的进程数，然后按特定算法计算出的数值。

第二行

系统的进程信息

- total表示总共的进程数
- running表示处于运行中的进程数
- sleep标识休眠中的进程数，stopped是已经停止了进程
- zombie是僵尸进程（父进程已经退出,而该进程dead之后没有进程接受）的数目。

第三行

cpu的状态信息

- us: 用户空间占用CPU的百分比
- sy: 内核空间占用CPU的百分比。
- ni: 改变过优先级的进程占用CPU的百分比
- id: 空闲CPU百分比
- wa: IO等待占用CPU的百分比
- hi: 硬中断 (Hardware IRQ) 占用CPU的百分比
- si: 软中断 (Software Interrupts) 占用CPU的百分比

什么是硬中断/软中断?

硬中断

由与系统相连的外设(比如网卡、硬盘)自动产生的。主要是用来通知操作系统系统外设状态的变化。如当网卡收到数据包的时候，就会发出一个中断。我们通常所说的中断指的是硬中断(hardirq)。

软中断

为了满足实时系统的要求，中断处理应该是越快越好。linux为了实现这个特点，当中断发生的时候，

中断处理那些短时间

就可以完成的工作，而将那些处理事件比较长的工作，放到中断之后来完成，也就是软中断(softirq)完成。

这里的CPU的使用比率貌似和windows概念不同，涉及到用户态和系统态，有空去了解了解。

第四行

内存的相关信息

- total: 物理内存总量
- used: 使用中的内存总量
- free: 空闲内存总量
- buffers: 缓存的内存量

第五行

交换分区的信息

- total: 交换区总量
- used: 使用的交换区总量
- free: 空闲交换区总量
- cached: 缓冲的交换区总量

这里需要注意的一点是：

第四行中使用中的内存总量 (used) 指的是现在系统内核控制的内存数，空闲内存总量 (free) 是内还未纳入其管控范围的数量。纳入内核管理的内存不见得都在使用中，还包括过去使用过的现在可以重复利用的内存，内核并不把这些可被重新使用的内存交还到free中去，因此在Linux上free内存会越来越少，但不用为此担心。

如果出于习惯去计算可用内存数，这里有个近似的计算公式：第四行的free + 第四行的buffers + 第五行的cached。

对于内存监控，在top里我们要时刻监控第五行swap交换分区的used，如果这个数值在不断的变化说明内核在不断进行内存和swap的数据交换，这是真正的内存不够用了。

然后下面的各行就是个进程的状态信息

其中：

USER: 进程所有者

PR: 进程优先级

NI: nice值 (-20----19) 越大表示优先级越低。

S: 进程状态

COMMAND: 进程名称 (命令/命令行)

切换观察模式

如果是多cpu的电脑，那么上面的cpu显示的是综合的cpu的信息，如果想要看单独的某个cpu的信息可以在当前按1切换模式

当然除此之外还有其他的模式可以选择，详情请参考：https://blog.csdn.net/jiangjiang_jin/article/details/84579958 Linux top命令的用法详细详解