

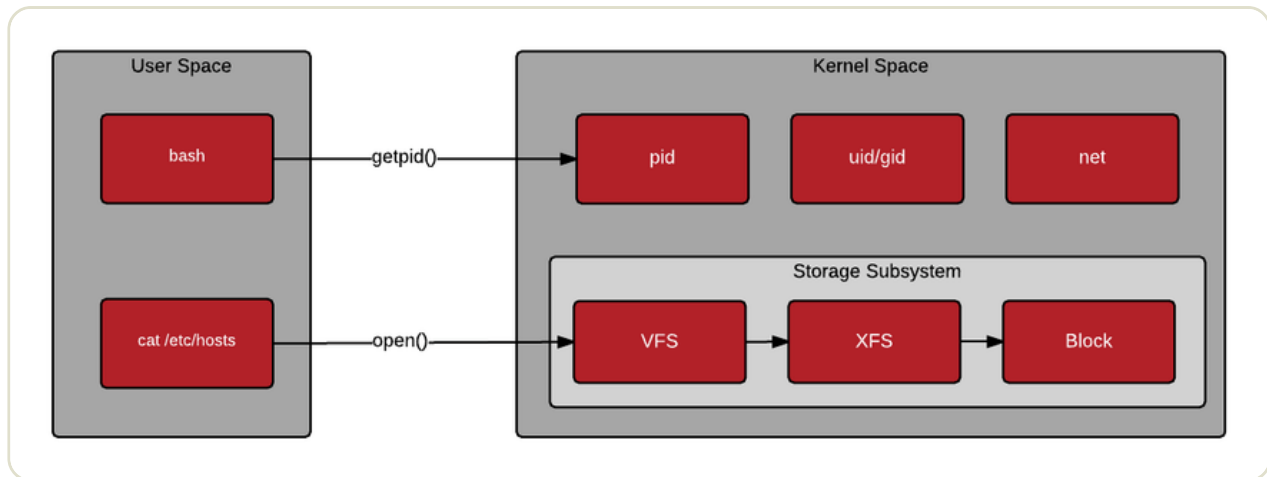
User space 与 Kernel space

作者：阮一峰

日期：2016年12月 2日

学习 Linux 时，经常可以看到两个词：User space（用户空间）和 Kernel space（内核空间）。

简单说，Kernel space 是 Linux 内核的运行空间，User space 是用户程序的运行空间。为了安全，它们是隔离的，即使用户的程序崩溃了，内核也不受影响。



Kernel space 可以执行任意命令，调用系统的一切资源；User space 只能执行简单的运算，不能直接调用系统资源，必须通过系统接口（又称 system call），才能向内核发出指令。

```
str = "my string" // 用户空间
x = x + 2
file.write(str) // 切换到内核空间

y = x + 4 // 切换回用户空间
```

上面代码中，第一行和第二行都是简单的赋值运算，在 User space 执行。第三行需要写入文件，就要切换到 Kernel space，因为用户不能直接写文件，必须通过内核安排。第四行又是赋值运算，就切换回 User space。

查看 CPU 时间在 User space 与 Kernel Space 之间的分配情况，可以使用 top 命令。它的第三行输出就是 CPU 时间分配统计。

```
top - 10:19:40 up 3 min, 3 users, load average: 2.07, 2.14, 0.96
tasks: 183 total, 1 running, 182 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.2 sy, 0.0 ni, 99.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 1545116 total, 930084 used, 615032 free, 80684 buffers
KiB Swap: 2095100 total, 0 used, 2095100 free, 360264 cached Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 1260 mysql    20   0 327548 36292 5792  S   0.7   2.3   0:04.74 mysqld
    1 root      20   0  4584   2652 1444  S   0.0   0.2   0:08.45 init
    2 root      20   0     0     0     0  S   0.0   0.0   0:00.00 kthreadd
    3 root      20   0     0     0     0  S   0.0   0.0   0:02.10 ksoftirqd/0
    4 root      20   0     0     0     0  S   0.0   0.0   0:00.00 kworker/0:0
    5 root      0 -20     0     0     0  S   0.0   0.0   0:00.00 kworker/0:0+
    6 root      20   0     0     0     0  S   0.0   0.0   0:00.16 kworker/u4+
    7 root      20   0     0     0     0  S   0.0   0.0   0:12.15 rcu_sched
    8 root      20   0     0     0     0  S   0.0   0.0   0:00.00 rcu_bh
    9 root      rt   0     0     0     0  S   0.0   0.0   0:01.07 migration/0
   10 root      rt   0     0     0     0  S   0.0   0.0   0:00.04 watchdog/0
   11 root      rt   0     0     0     0  S   0.0   0.0   0:00.26 watchdog/1
   12 root      rt   0     0     0     0  S   0.0   0.0   0:00.30 migration/1
   13 root      20   0     0     0     0  S   0.0   0.0   0:00.64 ksoftirqd/1
   14 root      20   0     0     0     0  S   0.0   0.0   0:00.00 kworker/1:0
   15 root      0 -20     0     0     0  S   0.0   0.0   0:00.00 kworker/1:0+
```

这一行有 8 项统计指标。

```
%Cpu(s): 24.8 us, 0.5 sy, 0.0 ni, 73.6 id, 0.4 wa, 0.0 hi, 0.2 si, 0.0 st
```

其中，第一项 24.8 us（user 的缩写）就是 CPU 消耗在 User space 的时间百分比，第二项 0.5 sy（system 的缩写）是消耗在 Kernel space 的时间百分比。

随便也说一下其他 6 个指标的含义。

- ni: niceness 的缩写，CPU 消耗在 nice 进程（低优先级）的时间百分比
- id: idle 的缩写，CPU 消耗在闲置进程的时间百分比，这个值越低，表示 CPU 越忙
- wa: wait 的缩写，CPU 等待外部 I/O 的时间百分比，这段时间 CPU 不能干其他事，但是也没有执行运算，这个值太高就说明外部设备有问题
- hi: hardware interrupt 的缩写，CPU 响应硬件中断请求的时间百分比
- si: software interrupt 的缩写，CPU 响应软件中断请求的时间百分比
- st: stole time 的缩写，该项指标只对虚拟机有效，表示分配给当前虚拟机的 CPU 时间之中，被同一台物理机上的其他虚拟机偷走的时间百分比

如果想查看单个程序的耗时，一般使用 time 命令。

```
File Edit View Terminal Help
roger@roger-desktop:~$ time ./perl-timeout-example 100.100.100.100

We could not ping the desired address!

real    0m5.013s
user    0m0.004s
sys     0m0.008s
roger@roger-desktop:~$
```

程序名之前加上 time 命令，会在程序执行完毕以后，默认显示三行统计。

- real: 程序从开始运行到结束的全部时间，这是用户能感知到的时间，包括 CPU 切换去执行其他任务的时间。
- user: 程序在 User space 执行的时间
- sys: 程序在 Kernel space 执行的时间

user 和 sys 之和，一般情况下，应该小于 real 。但如果是多核 CPU，这两个指标反映的是所有 CPU 的总耗时，所以它们之和可能大于 real 。

[参考链接]

- [User space vs kernel space](#)
- [Using the Linux Top Command](#)
- [Understanding Linux CPU stats](#)
- [What do 'real', 'user' and 'sys' mean in the output of time\(1\)?](#)

(完)

文档信息

- 版权声明：自由转载-非商用-非衍生-保持署名（创意共享3.0许可证）
- 发表日期：2016年12月 2日
- 更多内容：档案 » 理解计算机
- 博客文集：《寻找思想之路》，《未来世界的幸存者》
- 社交媒体：[twitter](#), [weibo](#)
- Feed订阅：[RSS](#)

打造中国最权威的《前端-全栈-工程化课程》

八年专注前端，珠峰培训让你高薪就业

快戳我！了解详情 

Google × 滴滴

零基础到进阶 | 一对一指导 | 滴滴面试

Android 开发工程师(入门)认证课程

立即加入 

报名仅剩 3 天！

相关文章

▪ 2016.11.22: 理解字节序

1. 计算机硬件有两种储存数据的方式：大端字节序（big endian）和小端字节序（little endian）。

▪ 2016.08.05: 布尔代数入门

布尔代数是计算机的基础。没有它，就不会有计算机。