# 伪文件系统

proc应该是一个虚拟文件系统，它只存在内存当中，而不占用外存空间。它以文件系统的方式为访问系统[内核](https://baike.baidu.com/item/%E5%86%85%E6%A0%B8" \t "_blank)数据的操作提供接口。用户和应用程序可以通过proc得到系统的信息，并可以改变[内核](https://baike.baidu.com/item/%E5%86%85%E6%A0%B8" \t "_blank)的某些参数。由于系统的信息，如进程，是动态改变的，所以用户或应用程序读取proc文件时，proc文件系统是动态从系统[内核](https://baike.baidu.com/item/%E5%86%85%E6%A0%B8" \t "_blank)读出所需信息并提交的。

1.proc - 进程信息伪文件系统

2.proc应该是一个虚拟文件系统, 被用作[内核](https://baike.baidu.com/item/%E5%86%85%E6%A0%B8" \t "_blank)[数据结构](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E7%BB%93%E6%9E%84)的接口, 而不仅仅是解释说明/dev/kmem

3./proc里的大多数文件都是只读的, 但也可以通过写一些文件来改变[内核](https://baike.baidu.com/item/%E5%86%85%E6%A0%B8" \t "_blank)变量.

[Linux](http://lib.csdn.net/base/linux) 内核提供了一种通过 /proc 文件系统，在运行时访问内核内部[数据结构](http://lib.csdn.net/base/datastructure" \t "_blank" \o "算法与数据结构知识库)、改变内核设置的机制。proc文件系统是一个伪文件系统，它只存在内存当中，而不占用外存空间。它以文件系统的方式为访问系统内核数据的操作提供接口。

用户和应用程序可以通过proc得到系统的信息，并可以改变内核的某些参数。由于系统的信息，如进程，是动态改变的，所以用户或应用程序读取proc文件时，proc文件系统是动态从系统内核读出所需信息并提交的。下面列出的这些文件或子文件夹，并不是都是在你的系统中存在，这取决于你的内核配置和装载的模块。另外，在/proc下还有三个很重要的目录：net，scsi和sys。 Sys目录是可写的，可以通过它来访问或修改内核的参数，而net和scsi则依赖于内核配置。例如，如果系统不支持scsi，则scsi 目录不存在。

除了以上介绍的这些，还有的是一些以数字命名的目录，它们是进程目录。系统中当前运行的每一个进程都有对应的一个目录在/proc下，以进程的 PID号为目录名，它们是读取进程信息的接口。而self目录则是读取进程本身的信息接口，是一个link。

**/proc相关指定的操作命令所获的内容信息**

/proc/buddyinfo 每个内存区中的每个order有多少块可用，和内存碎片问题有关

/proc/cmdline 启动时传递给kernel的参数信息

/proc/cpuinfo cpu的信息

/proc/crypto 内核使用的所有已安装的加密密码及细节

/proc/devices 已经加载的设备并分类

/proc/dma 已注册使用的ISA DMA频道列表

/proc/execdomains Linux内核当前支持的execution domains

/proc/fb 帧缓冲设备列表，包括数量和控制它的驱动

/proc/filesystems 内核当前支持的文件系统类型

/proc/interrupts x86[架构](http://lib.csdn.net/base/architecture)中的每个IRQ中断数

/proc/iomem 每个物理设备当前在系统内存中的映射

/proc/ioports 一个设备的输入输出所使用的注册端口范围

/proc/kcore 代表系统的物理内存，存储为核心文件格式，里边显示的是字节数，等于RAM大小加上4kb

/proc/kmsg 记录内核生成的信息，可以通过/sbin/klogd或/bin/dmesg来处理

/proc/loadavg 根据过去一段时间内CPU和IO的状态得出的负载状态，与uptime命令有关

/proc/locks 内核锁住的文件列表

/proc/mdstat 多硬盘，RAID配置信息(md=multiple disks)

/proc/meminfo RAM使用的相关信息

/proc/misc 其他的主要设备(设备号为10)上注册的驱动

/proc/modules 所有加载到内核的模块列表

/proc/mounts 系统中使用的所有挂载

/proc/mtrr 系统使用的Memory Type Range Registers (MTRRs)

/proc/partitions 分区中的块分配信息

/proc/pci 系统中的PCI设备列表

/proc/slabinfo 系统中所有活动的 slab 缓存信息

/proc/stat 所有的CPU活动信息

/proc/sysrq-trigger 使用echo命令来写这个文件的时候，远程root用户可以执行大多数的系统请求关键命令，就好像在本地终端执行一样。要写入这个文件，需要把/proc/sys/kernel/sysrq不能设置为0。这个文件对root也是不可读的

/proc/uptime 系统已经运行了多久

/proc/swaps 交换空间的使用情况

/proc/version Linux内核版本和gcc版本

/proc/bus 系统总线(Bus)信息，例如pci/usb等

/proc/driver 驱动信息

/proc/fs 文件系统信息

/proc/ide ide设备信息

/proc/irq 中断请求设备信息

/proc/net 网卡设备信息

/proc/scsi scsi设备信息

/proc/tty tty设备信息

/proc/net/dev 显示网络适配器及统计信息

/proc/vmstat 虚拟内存统计信息

/proc/vmcore 内核panic时的内存映像

/proc/diskstats 取得磁盘信息

/proc/schedstat kernel调度器的统计信息

/proc/zoneinfo 显示内存空间的统计信息，对分析虚拟内存行为很有用

以下是/proc目录中进程N的信息

/proc/N pid为N的进程信息

/proc/N/cmdline 进程启动命令

/proc/N/cwd 链接到进程当前工作目录

/proc/N/environ 进程环境变量列表

/proc/N/exe 链接到进程的执行命令文件

/proc/N/fd 包含进程相关的所有的文件描述符

/proc/N/maps 与进程相关的内存映射信息

/proc/N/mem 指代进程持有的内存，不可读

/proc/N/root 链接到进程的根目录

/proc/N/stat 进程的状态

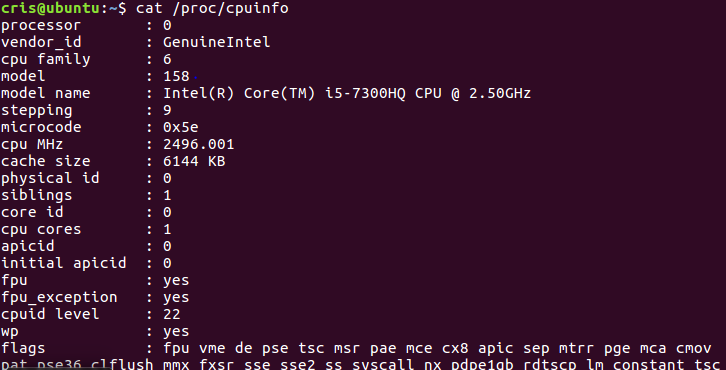
/proc/N/statm 进程使用的内存的状态

/proc/N/status 进程状态信息，比stat/statm更具可读性

/proc/self 链接到当前正在运行的进程

例如：

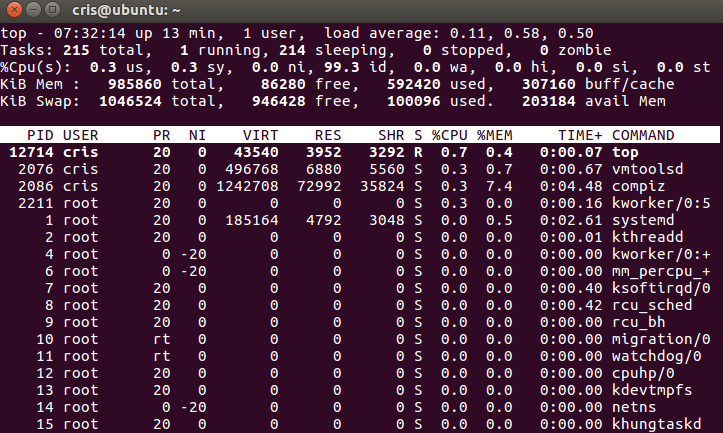
cat /proc/cpuinfo 显示Linux系统下cpu的信息



## 查看指定进程的CPU和内存使用情况操作：

1. 首先在终端输入top指定，查看所有进程的状态信息

**top命令**可以实时动态地查看系统的整体运行情况，是一个综合了多方信息监测系统性能和运行信息的实用工具。通过top命令所提供的互动式界面，用热键可以管理。



PID：进程的ID

USER：进程的所属者

PR：进程的优先级，越小越优先被执行

NI：NInice值

VIRT：进程占用的虚拟内存

RES：进程占用的物理内存

SHR：进程使用的共享内存

S：进程的状态 S表示休眠，R表示正在运行，Z表示僵死状态，N表示该进程优先值为负数

%CPU：进程占用CPU的使用率

%MEM：进程使用的物理内存和总内存的百分比

TIME+：该进程启动后占用的总的CPU时间，即占用CPU使用时间的累加值

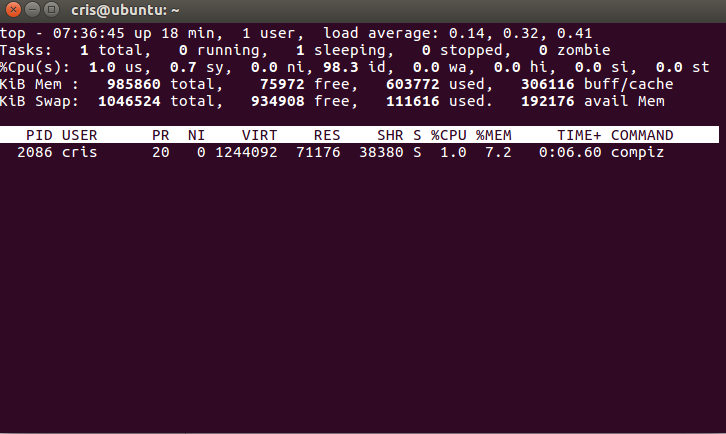
COMMAND:进程启动命令的名称

top命令是Linux下常用的性能分析工具，有点类似于windows的任务管理器，这里面包括了所有进程的信息，包括占用内存的信息，进程状态，CPU使用率，运行时间等重要信息。并且这些信息是动态刷新的

2.查看指定进程的CPU和内存等信息的占用情况

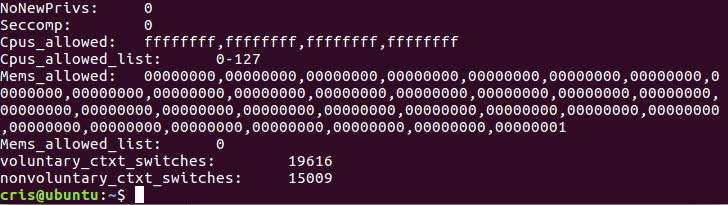
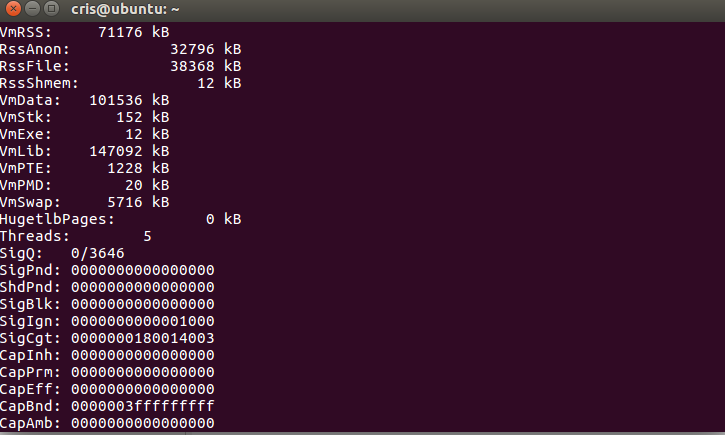
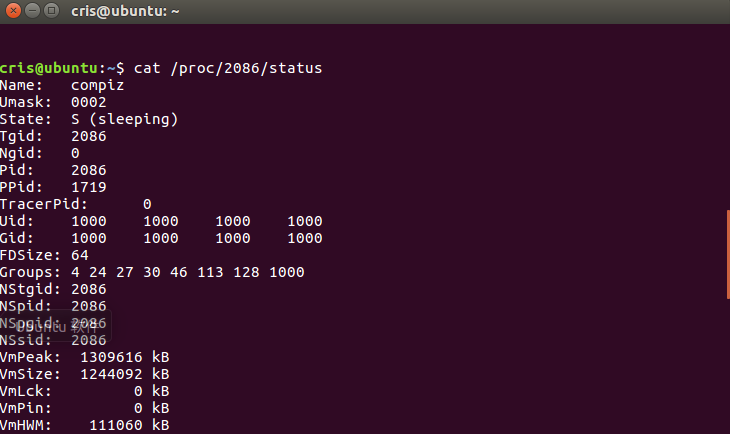
输入指令top -p 加上指定进程的PID值

选取指定的IP地址，例如PID为2086



1. 查看更为详细的内存占比

输入指令 cat /proc/指定进程的PID值/status



Linux系统终端下输入的ps命令：

用来显示当前进程的状态

ps -A:显示所有的进程

Linux系统的top命令功能和ps命令功能相似，同样都是查看所有进程的信息，如内存和CPU等信息，但是不同点在于top命令是反馈实时的进程信息，这个信息是动态的，刷新时间大概是1s，而ps命令是反馈静态的进程信息，是不可变化的。

问题：我也不知道学长要求我完成的任务是 写一个.py文件的demo还是一个终端指令操作的demo。我目前的理解是通过top指令的功能实现，去完成学长要求我做的查看指定进程的CPU，内存使用等信息，我将每一步的操作通过截图发给学长看一看，不知道完成的对不对。希望学长将此次我对任务的理解以及操作的问题可以给我指出来，也希望学长能够将接下来的任务发送给我，麻烦学长啦！

今天是除夕夜，知道学长挺忙的，还要忙着和家人一起办置过年的事情，也就不和学长单独开会耽误学长的时间啦，祝学长您新年快乐，工作顺利，身体健康！