### 实验三：进程管理（一）

1. 分别从至少三个虚拟终端登录，以树状形式列出你的系统中当前运行的所有进程及其PID。找出你当前运行进程的所有祖先进程，并说明其各自的作用。分析Linux系统中的进程的家族关系。

答：使用pstree -p | more命令分别在tty1~3三个虚拟终端下查看进程，截图如下：

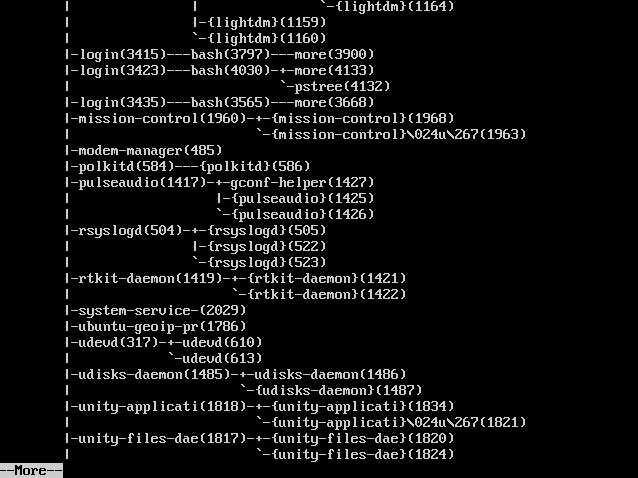
tty1(PID为3435)



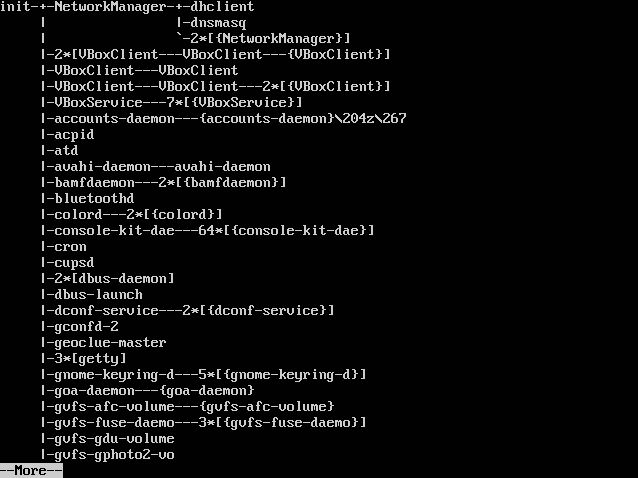
tty2(PID为3415)



tty3(PID为3423)



从下图可以看出，所有进程的祖先进程为init，其PID始终为1。



1. 执行下列命令和操作：

$sleep 180

^Z

$jobs

$sleep 240&

$sleep 300&

$sleep 330&

$sleep 630&

$jobs

$fg %4

^C

$bg %1

$kill %5

解释命令的作用和执行结果。

答：$sleep 180 #睡眠180秒

^Z #按下Ctrl+Z，挂起进程，作业将暂停执行

$jobs #列出作业控制的内容，显示所有shell中已启动的作业状态

$sleep 240& #以后台方式执行睡眠240秒

$sleep 300& #以后台方式执行睡眠300秒

$sleep 330& #以后台方式执行睡眠330秒

$sleep 630& #以后台方式执行睡眠630秒

$jobs #列出作业控制的内容

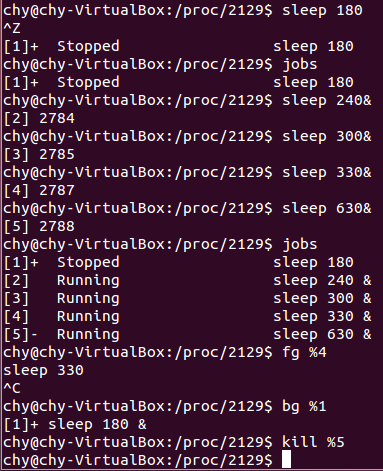
$fg %4 #将4号作业切换到前台并继续运行

^C #终止执行当前作业

$bg %1 #1号作业保持在后台并继续运行

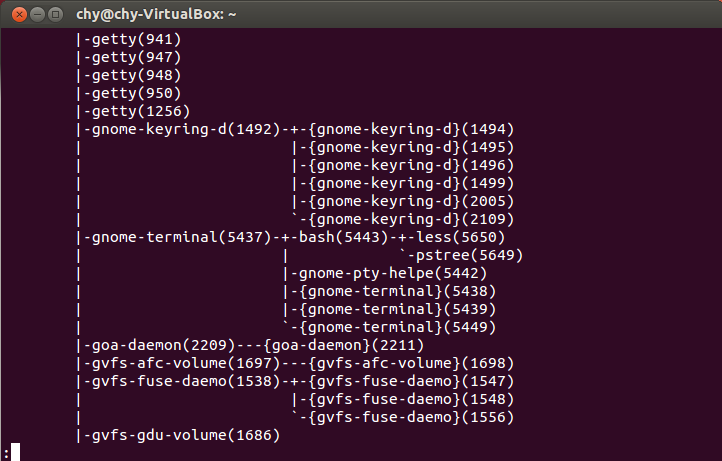
$kill %5 #结束5号作业

执行过程如下：

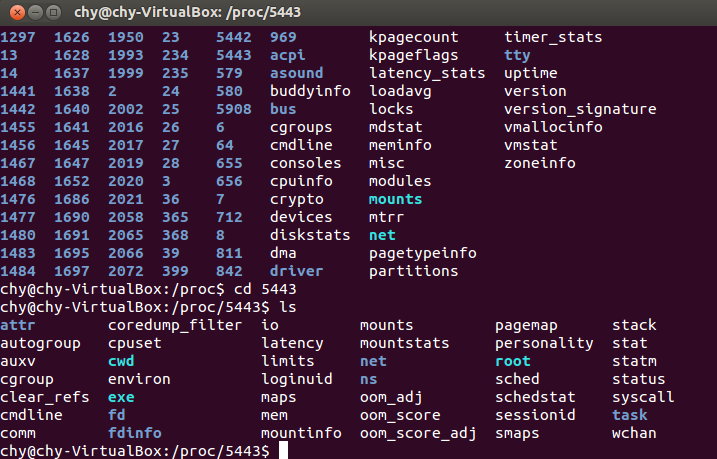


1. 利用/proc文件系统，选择一个进程，如-bash，列出并解释/proc/$PID/cmdline, /proc/$PID/stat, /proc/$PID/status文件的内容。

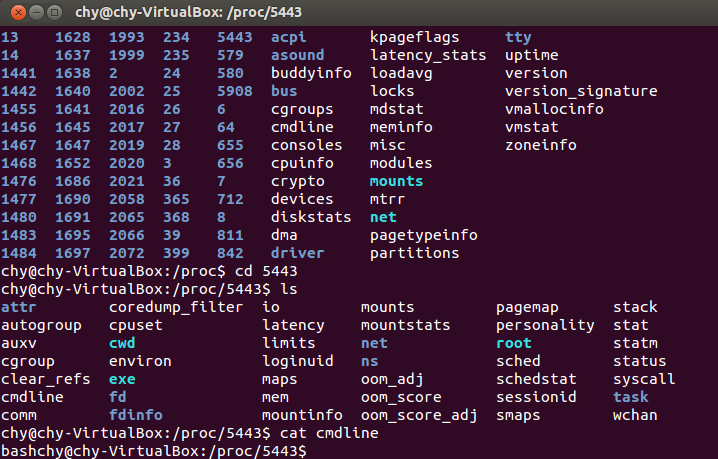
答：使用pstree -p | less如下命令查看，得到bash进程的PID为5443：



进入/proc/5443文件夹，查看目录下的文件如下：

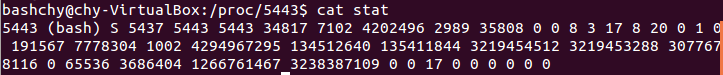


列出/proc/5443/cmdline的内容如下：



发现里面的内容仅有一行bash(末尾没有换行)，这个文件保存了进程执行时的命令行。

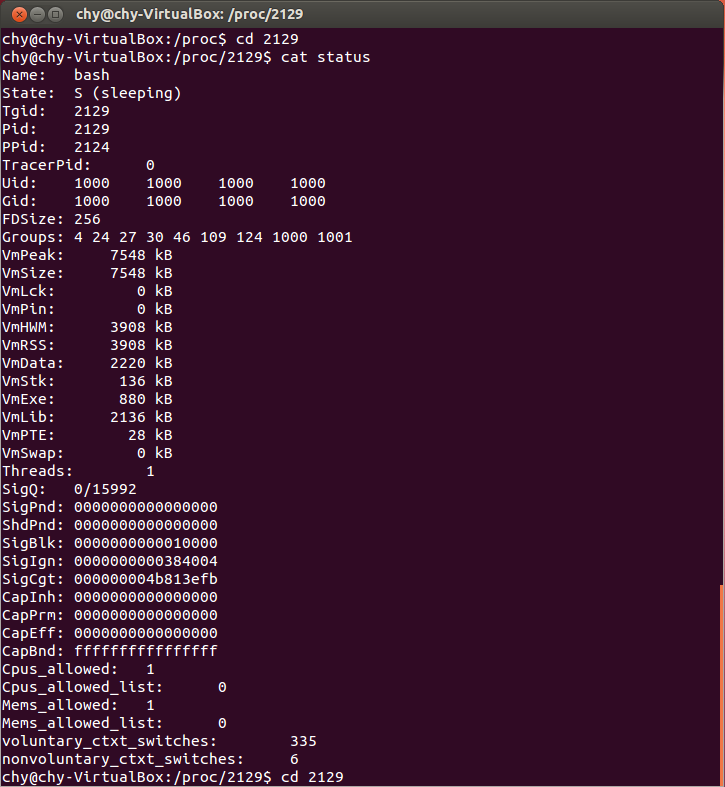
列出/proc/5443/stat的内容如下：



部分字段解释如下：

|  |  |
| --- | --- |
| 字段 | 含义 |
| pid | 进程ID. |
| comm | task\_struct结构体的进程名 |
| state | 进程状态, 此处为S |
| ppid | 父进程ID （父进程是指通过fork方式，通过clone并非父进程） |
| pgrp | 进程组ID |
| session | 进程会话组ID |
| tty\_nr | 当前进程的tty终点设备号 |
| tpgid | 控制进程终端的前台进程号 |
| flags | 进程标识位，定义在include/linux/sched.h中的PF\_\* |
| minflt | 次要缺页中断的次数，即无需从磁盘加载内存页. 比如COW和匿名页 |
| cminflt | 当前进程等待子进程的minflt |
| majflt | 主要缺页中断的次数，需要从磁盘加载内存页. 比如map文件 |
| majflt | 当前进程等待子进程的majflt |
| utime | 该进程处于用户态的时间，单位jiffies |
| stime | 该进程处于内核态的时间，单位jiffies |
| cutime | 当前进程等待子进程的utime |
| cstime | 当前进程等待子进程的utime |
| priorit | 进程优先级 |
| nice | nice值，取值范围[19, -20] |
| num\_threads | 线程个数 |
| itrealvalue | 该字段已废弃，恒等于0 |
| starttime | 自系统启动后的进程创建时间，单位jiffies |
| vsize | 进程的虚拟内存大小，单位为bytes |
| rss | 进程独占内存+共享库，单位pages |
| rsslim | rss大小上限 |

列出/proc/2129/status的内容如下：

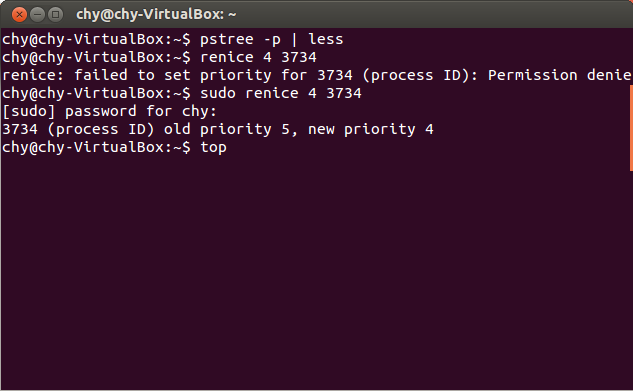


部分参数解释如下：

|  |  |
| --- | --- |
| 参数 | 解释 |
| Name | 应用程序或命令的名字 |
| State | 任务的状态，运行/睡眠/僵死/ |
| Tgid | 线程组号 |
| Pid | 任务ID |
| PPid | 父进程ID |
| TracerPid | 接收跟踪该进程信息的进程的ID号 |
| Uid | Uid |
| Gid | Gid |
| FDSize | 文件描述符的最大个数，file->fds |
| Groups |  |
| VmPeak |  |
| VmSize | 任务虚拟地址空间的大小 |
| VmLck | 任务已经锁住的物理内存的大小。锁住的物理内存不能交换到硬盘 |
| VmPin |  |
| VmHWM |  |
| VmRSS | 应用程序正在使用的物理内存的大小，就是用ps命令的参数rss的值 |
| VmData | 程序数据段的大小（所占虚拟内存的大小），存放初始化了的数据； |
| VmStk | 任务在用户态的栈的大小 |
| VmExe(KB) | 程序所拥有的可执行虚拟内存的大小，代码段，不包括任务使用的库 |
| VmLib(KB) | 被映像到任务的虚拟内存空间的库的大小 |
| VmPTE | 该进程的所有页表的大小，单位：kb |
| Threads | 共享使用该信号描述符的任务的个数，在POSIX多线程序应用程序中，线程组中的所有线程使用同一个信号描述符。 |
| SigQ | 待处理信号的个数 |
| SigPnd | 屏蔽位，存储了该线程的待处理信号 |
| ShdPnd | 屏蔽位，存储了该线程组的待处理信号 |
| SigBlk | 存放被阻塞的信号 |
| SigIgn | 存放被忽略的信号 |
| SigCgt | 存放被俘获到的信号 |
| CapInh | Inheritable，能被当前进程执行的程序的继承的能力 |
| CapPrm | Permitted，进程能够使用的能力，可以包含CapEff中没有的能力，这些能力是被进程自己临时放弃的，CapEff是CapPrm的一个子集，进程放弃没有必要的能力有利于提高安全性 |
| CapEff | Effective，进程的有效能力 |
| CapBnd |  |
| Cpus\_allowed |  |
| Cpus\_allowed\_list |  |
| Mems\_allowed |  |
| Mems\_allowed\_list |  |
| voluntary\_ctxt\_switches | 进程主动切换的次数 |
| nonvoluntary\_ctxt\_switches | 表示进程被动切换的次数 |

1. 分别以root和普通用户身份运行一个进程，使用nice、renice和top设置或者改变其优先级，并验证是否成功。

答：经测试，使用nice设置静态优先级不论在root还是普通身份下都能执行成功，而使用renice设置PID为3737的进程(bash)的nice值如下，只在root权限(sudo)下执行成功：



使用普通身份用top改变bash进程的nice值失败，截图如下：

