

实验三：进程管理（一）

Hollow Man

实验目的：

- 1. 学习进程的基本概念，理解进程的基本属性
- 2. 掌握进程操作的主要命令
- 3. 掌握通过/proc 文件系统获取进程信息
- 4. 理解 Linux 系统的进程树

实验时间

6 学时

预备知识：

- 1. 有关进程的主要命令：
 - 1.1. ps：查看系统运行进程

ps 常用选项（最常用 ps -aux）：

选项	说明
x	显示机器上的所有进程，不以终端来区分
a	显示终端上的所有进程，不以用户来区分
w	提供详细的宽范围输出
u	显示面向用户的格式
f	以进程树格式列出进程

ps 运行结果的各字段含义：

字段	说明
UID	用户ID
PID	进程ID
PPID	父进程ID
TTY	控制终端ID
PRI	优先级编号（编号越低，分配给此进程的计算机时间越多）
NI (nice)	影响动态优先级调整
STAT	当前的进程状态
TIME	使用的CPU时间

COMMAND	命令的名称
---------	-------

进程状态 (STAT) 可为以下状态之一：

状态代码	说明
R（可运行）	可运行的进程
S（休眠）	正在等待外部事件的进程（如数据到达）。
D（不可中断的休眠）	类似“S”（此时不能终止此进程）
T（已跟踪或已停止）	进程已被暂停
X（不用）	进程已死
Z（僵进程）	进程已自行终止，但尚未释放

1.2. pstree：以树结构显示进程列表

pstree 常用选项：

选项	说明
-p	显示进程 PID
-u	显示用户 ID

1.3 top：实时显示正在运行的进程

top 运行结果的各字段含义：

字段	说明
PID	进程 ID
USER	用户名
PR	优先级
NI	Nice 值
VIRT	进程使用的虚拟内存总量（单位为 KB）
RES	进程持有的物理内存总量（单位为 KB）
SHR	共享内存大小（单位为 KB）
S	进程状态
%CPU	CPU 使用率
%MEM	内存使用率 (RES)
TIME	CPU 时间
COMMAND	命令名称/行

top 常用命令：

字段	说明
?	帮助
h	帮助
r	将一个新的 nice 值分配给运行中的进程
k	向某个运行中的进程发送终止信号（与 kill 或 killall 相同）
N	按进程 ID 排序
P	按 CPU 负载排序

1.4 nice 和 renice

nice: 为进程指派静态优先级

E.g. `$nice -n +5 sleep 99`

renice:更改运行进程的 nice 值

E.g. `$renice 5 1012`

注意: 1. 调度程序使用 nice 级别来确定为运行的进程提供服务的频率。

2. nice 的范围: -20~20; nice 值越低, 进城优先级越高。

3. nice 默认为 10。

4. 只有 root 用户可以负的 nice 值启动进程; 只有 root 用户可以降低运行进程的 nice 值。

1.5 前台与后台进程

`$command &` 以后台方式启动进程

`fg` 将进程切换到前台

`bg` 将进程切换到后台运行

`jobs` 列出作业控制的内容

`ctrl+Z` 挂起进程

1.6 kill 和 killall 向进程发送信号

`kill` 用 PID 向进程发送信号

E.g. `$kill - SIGTERM 1703`

`killall` 用进程的命令名称向进程发送信号

常用信号:

编号	名称	说明
1	SIGHUP	重载配置文件
2	SIGINT	通过键盘 (Ctrl+C) 中断
9	SIGKILL	停止进程。
15	SIGTERM	立即结束进程 (以受控的方法终止进程, 因此可以进行清理)
18	SIGCONT	继续由STOP停止的进程
19	STOP	停止进程

注意: `kill -l` 或 `man 7 signal` 可获取信号的完整列表。

2. 利用/proc 文件系统查看进程信息

系统中运行的每个进程在/proc 目录下有个目录, 目录名称为进程 PID。

/proc/\$PID/下包含进程的详细信息:

文件	说明
cmdline	进程的命令行

environ	进程的环境信息
fd 目录	为每个打开的文件描述符提供一个入口
mem	可通过其访问进程的内存映像
stat	包含进程的大多数信息
status	包含用户可读的进程信息
cwd	指向进程的当前工作目录
exe	指向正在被执行的文件
maps	内存映射区信息
root	进程的根目录
statm	进程对内存的使用情况

实验内容:

1. 列出你的系统中当前运行的所有进程，并解释 bash 进程的各项信息的含义（不包括 RSS 和 SHR）。

```
root@moocos-VirtualBox:/home/moocos# ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT   ST
root           1   0.0   0.1  33896   3168 ?        Ss     12
root           2   0.0   0.0     0     0 ?        S      12
root           3   0.0   0.0     0     0 ?        S      12
root           4   0.0   0.0     0     0 ?        S      12
root           5   0.0   0.0     0     0 ?        S<     12
root           7   0.0   0.0     0     0 ?        S      12
root           8   0.0   0.0     0     0 ?        R      12
root           9   0.0   0.0     0     0 ?        S      12
root          10   0.0   0.0     0     0 ?        S      12
root          11   0.0   0.0     0     0 ?        S      12
root          12   0.0   0.0     0     0 ?        S      12
root          13   0.0   0.0     0     0 ?        S<     12
root          14   0.0   0.0     0     0 ?        S      12
root          15   0.0   0.0     0     0 ?        S<     12
root          16   0.0   0.0     0     0 ?        S<     12
root          17   0.0   0.0     0     0 ?        S<     12
root          18   0.0   0.0     0     0 ?        S<     12
root          19   0.0   0.0     0     0 ?        S<     12
root          20   0.0   0.0     0     0 ?        S<     12
root          21   0.0   0.0     0     0 ?        S<     12
root          22   0.0   0.0     0     0 ?        S      12
root          23   0.0   0.0     0     0 ?        S<     12
root          24   0.0   0.0     0     0 ?        S<     12
root          27   0.0   0.0     0     0 ?        S      12
root          28   0.0   0.0     0     0 ?        S      12
```

bash 进程的各项信息的含义:

user: 用户名 **PID:** 进程 ID **%CPU:** 占有 CPU 时间的百分比

%MEM:占有主存的大小百分比**VSZ:** 进程占用虚拟内存大小**TTY:** 控制终端的编号

STAT: 当前的进程状态**TIME:** 使用的计算机时间 **COMMAND:**命令名称

2. 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。

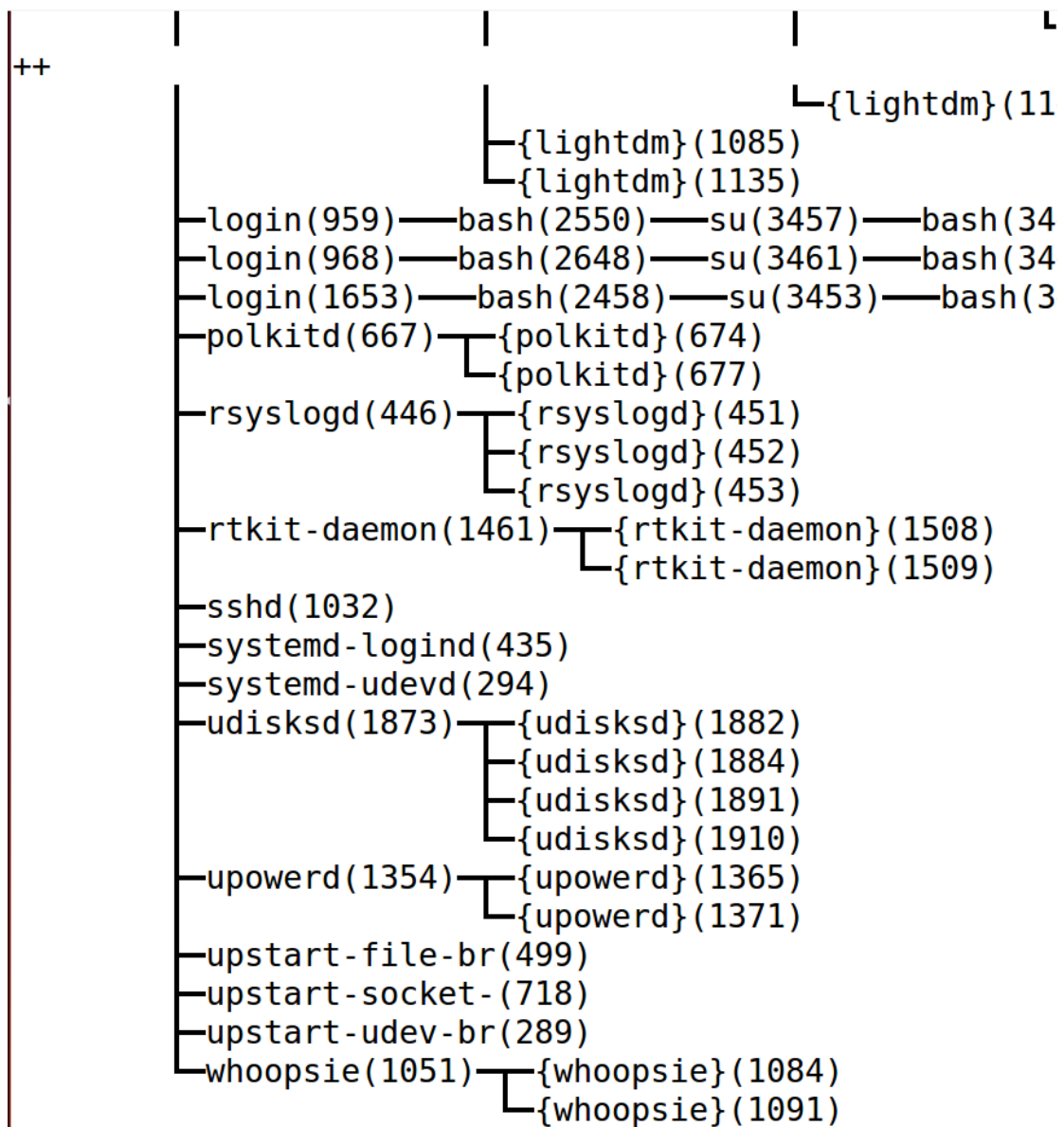
Root:

```
root@moocos-VirtualBox:/home/moocos# pstree -p
init(1)─ModemManager(635)─{ModemManager}(642)
    │   └─{ModemManager}(644)
    └─NetworkManager(660)─dhclient(710)
        │   └─dnsmasq(802)
        └─{NetworkManager}(661)
            │   └─{NetworkManager}(662)
            └─{NetworkManager}(665)
    └─VBoxClient(1216)─{VBoxClient}(1222)
        │   └─VBoxClient(1225)─{VBoxClient}(1231)
        │   └─VBoxClient(1229)─{VBoxClient}(1230)
        │   └─VBoxClient(1237)─{VBoxClient}(1238)
        │       └─{VBoxClient}(1239)
        └─VBoxService(1571)─{VBoxService}(1572)
            │   └─{VBoxService}(1573)
            │   └─{VBoxService}(1574)
            │   └─{VBoxService}(1575)
            │   └─{VBoxService}(1576)
            │   └─{VBoxService}(1577)
            └─{VBoxService}(1578)
    └─accounts-daemon(1088)─{accounts-daemon}(1088)
        └─{accounts-daemon}(1088)
    └─acpid(1045)
    └─anacron(3498)─sh(3550)─run-parts(3551)─
        │   └─avahi-daemon(472)─avahi-daemon(473)
        │   └─bluetoothd(430)
        │   └─colord(1686)─{colord}(1719)
        │       └─{colord}(1734)
        └─cron(1043)
```

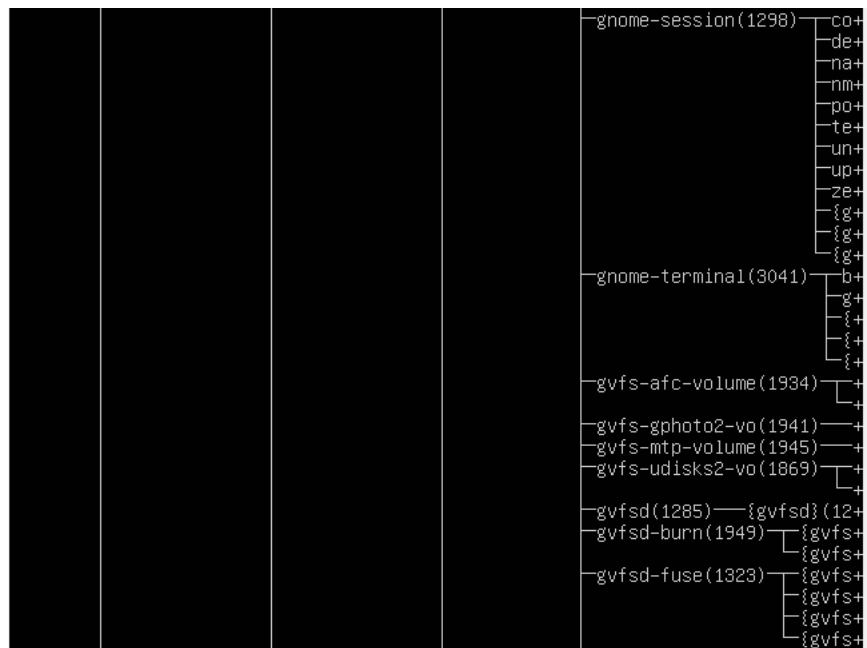
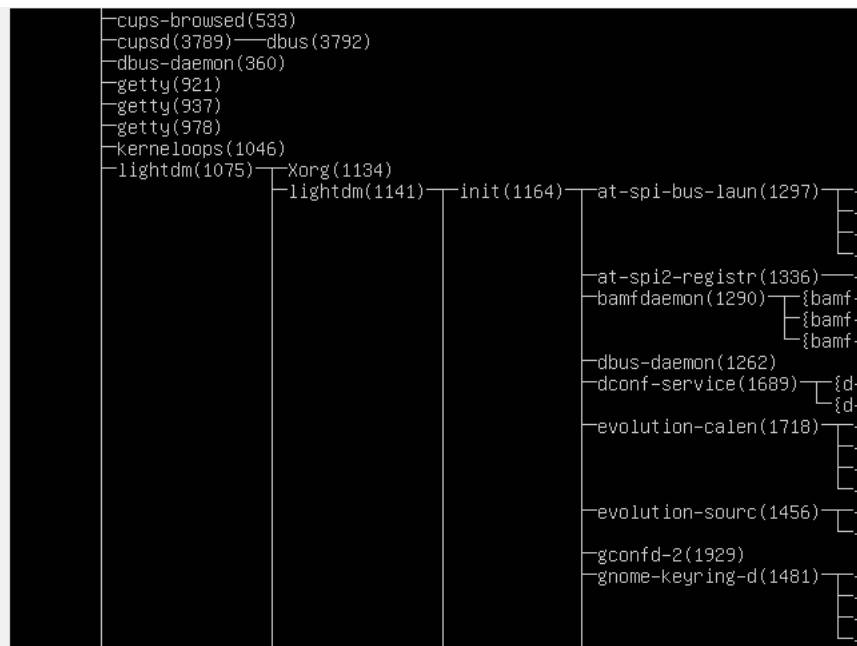
```

├─ cups - browsed(533)
├─ cupsd(470)
├─ dbus-daemon(360)
├─ getty(921)
├─ getty(937)
├─ getty(978)
├─ kerneloops(1046)
├─ lightdm(1075) ── Xorg(1134)
│                  └─ lightdm(1141) ── init(1164) ──
++
|
++
|
++
|
++
|
++
|
++
|
++
|
++
|
++
|
++
|
++
|

```

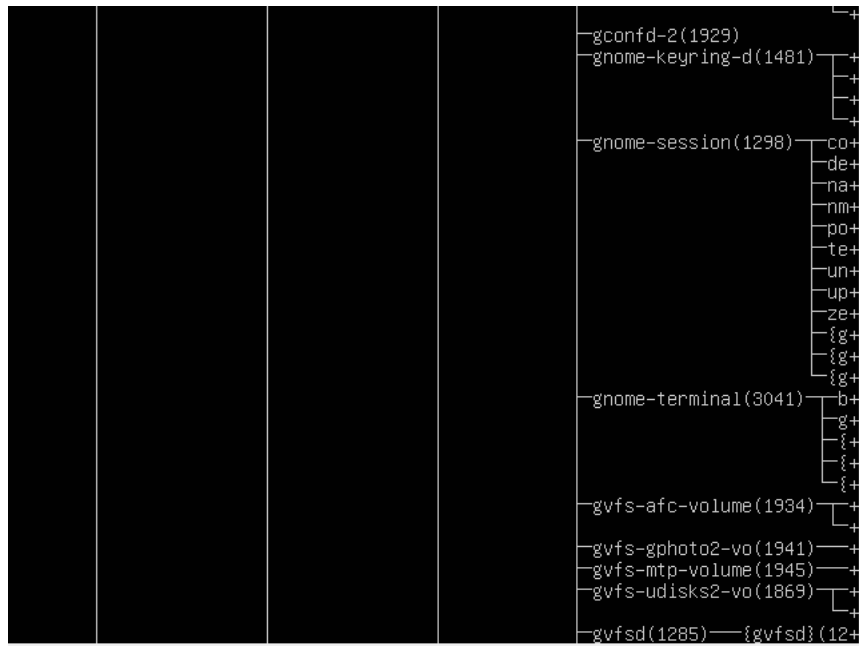
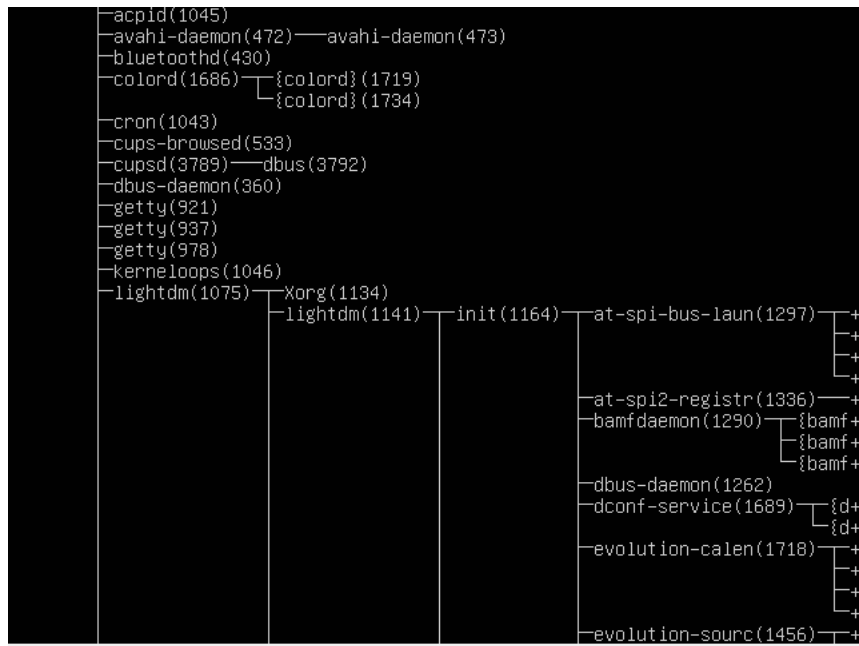


S1:



				-gvfsd-metadata(1977)	-{
				-gvfsd-trash(1969)	-{gvf+
					-{gvf+
					-{gvf+
					-{gvf+
				-hud-service(1296)	-{hud+
					-{hud+
					-{hud+
				-ibus-daemon(1274)	-ibus+
					-ibus+
					-ibus+
					-ibus+
					-{ibu+
					-{ibu+
				-ibus-x11(1319)	-{ibus-x+
					-{ibus-x+
					-{ibus-x+
				-indicator-appli(1425)	-
				-indicator-bluet(1399)	-
					-
				-indicator-datet(1405)	-
					-
					-
					-
				-indicator-keybo(1390)	-
					-
				-indicator-messa(1395)	-
					-

				-indicator-power(1402)	-+
				-indicator-print(1411)	-+
				-indicator-sessi(1414)	-+
					-+
				-indicator-sound(1408)	-+
					-+
					-+
				-mission-control(1987)	-+
					-+
				-notify-osd(1740)	-{noti+
					-{noti+
				-pulseaudio(1455)	-{puls+
					-{puls+
				-ssh-agent(1255)	-
				-unity-panel-ser(1307)	-+
					-+
				-unity-settings-(1289)	-+
					-+
					-+
					-
				-upstart-dbus-br(1333)	-
				-upstart-dbus-br(1335)	-
				-upstart-event-b(1270)	-
				-upstart-file-br(1330)	-
				-window-stack-br(1280)	-
				-zeitgeist-daemo(2005)	-+
				-zeitgeist-fts(2012)	-ca+



				gvfsd-burn(1949)	{gvfs+
					{gvfs+
				gvfsd-fuse(1323)	{gvfs+
					{gvfs+
					{gvfs+
					{gvfs+
				gvfsd-metadata(1977)	{+}
				gvfsd-trash(1969)	{gvf+
					{gvf+
					{gvf+
					{gvf+
				hud-service(1296)	{hud+
					{hud+
					{hud+
				ibus-daemon(1274)	ibus+
					ibus+
					ibus+
					ibus+
					{ibu+
					{ibu+
				ibus-x11(1319)	{ibus-x+
					{ibus-x+
					{ibus-x+
				indicator-appli(1425)	+
				indicator-bluet(1399)	+
					+
				indicator-datet(1405)	+
					+
					+
					+

				indicator-keybo(1390)	+
					+
					+
				indicator-messa(1395)	+
					+
					+
				indicator-power(1402)	+
					+
				indicator-print(1411)	+
					+
				indicator-sessi(1414)	+
					+
				indicator-sound(1408)	+
					+
					+
				mission-control(1987)	+
					+
				notify-osd(1740)	{noti+
					{noti+
				pulseaudio(1455)	{puls+
					{puls+
				ssh-agent(1255)	
				unity-panel-ser(1307)	+
					+
				unity-settings-(1289)	+
					+
					+
					+
				upstart-dbus-br(1333)	



s2@moocos-VirtualBox: /home/moocos\$

S3:

				gnome-session(1298)	co+ de+ na+ nm+ po+ te+ un+ up+ ze+ {g+ {g+ {g+
				gnome-terminal(3041)	b+ g+ i+ i+ i+
				gvfs-afc-volume(1934)	+ +
				gvfs-gphoto2-vo(1941)	+
				gvfs-mtp-volume(1945)	+
				gvfs-udisks2-vo(1869)	+ +
				gvfsd(1285)	{gvfsd}(12+ {gvfs+ {gvfs+ {gvfs+ {gvfs+ {gvfs+

				gvfsd-metadata(1977)	{+ {gvf+ {gvf+ {gvf+ {gvf+
				gvfsd-trash(1969)	{gvf+ {gvf+ {gvf+ {gvf+
				ibus-daemon(1274)	ibus+ ibus+ ibus+ ibus+ {ibu+ {ibu+
				ibus-x11(1319)	{ibus-x+ {ibus-x+ {ibus-x+
				indicator-appli(1425)	+
				indicator-bluet(1399)	+ +
				indicator-datet(1405)	+ + + + +
				indicator-keybo(1390)	+ + +
				indicator-messa(1395)	+ + +


```

indicator-power(1402)
indicator-print(1411)
indicator-sessi(1414)
indicator-sound(1408)
mission-control(1987)
notify-osd(1740)
pulseaudio(1455)
ssh-agent(1255)
unity-panel-ser(1307)
unity-settings-(1289)
upstart-dbus-br(1333)
upstart-dbus-br(1335)
upstart-event-b(1270)
upstart-file-br(1330)
window-stack-br(1280)
zeitgeist-daemo(2005)
zeitgeist-fts(2012)

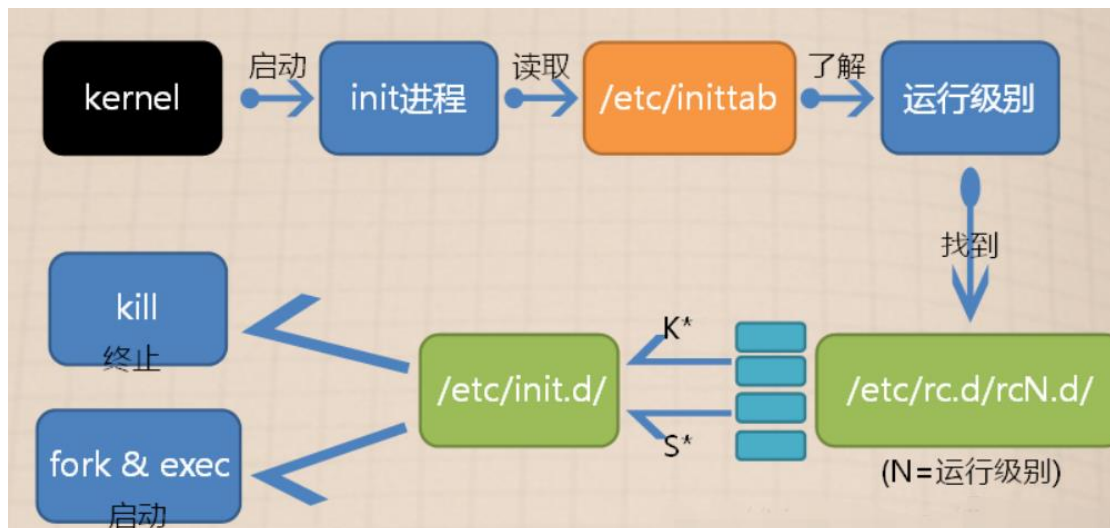
```

```

{lightdm}(1146)
{lightdm}(1085)
{lightdm}(1135)
login(959) bash(2550) su(3457) bash(3458)
login(968) bash(2648) su(3461) bash(3462) pstree(3966)
login(1653) bash(2458) su(3453) bash(3454)
polkitd(667) {polkitd}(674)
{polkitd}(677)
rsyslogd(446) {rsyslogd}(451)
{rsyslogd}(452)
{rsyslogd}(453)
rtkit-daemon(1461) {rtkit-daemon}(1508)
{rtkit-daemon}(1509)
sshd(1032)
systemd-logind(435)
systemd-udev(294)
udisksd(1873) {udisksd}(1882)
{udisksd}(1884)
{udisksd}(1891)
{udisksd}(1910)
upowerd(1354) {upowerd}(1365)
{upowerd}(1371)
upstart-file-br(499)
upstart-socket-(718)
upstart-udev-br(289)
whoopsie(1051) {whoopsie}(1084)
{whoopsie}(1091)
s3@moocos-VirtualBox:/home/moocos$ ps

```

从上图可以看出，所有进程的祖先进程为 init，其 PID 始终为 1。Init 用来守护进程。



3. 试验有几种方法可以列出系统中的所有运行进程？

PS -A -a 显示所有进程 a 显示终端中包括其他用户的所有进程

U f 树状方式 auf axfaxu 综合表示

root@moocos-VirtualBox:/home/moocos# ps -A

PID	TTY	TIME	CMD
1	?	00:00:02	init
2	?	00:00:00	kthreadd
3	?	00:00:00	ksoftirqd/0
4	?	00:00:00	kworker/0:0
5	?	00:00:00	kworker/0:0H
7	?	00:00:01	rcu_sched
8	?	00:00:00	rcuos/0
9	?	00:00:00	rcu_bh
10	?	00:00:00	rcuob/0
11	?	00:00:00	migration/0
12	?	00:00:00	watchdog/0
13	?	00:00:00	khelper
14	?	00:00:00	kdevtmpfs
15	?	00:00:00	netns
16	?	00:00:00	writeback
17	?	00:00:00	kintegrityd
18	?	00:00:00	bioaset
19	?	00:00:00	kworker/u3:0
20	?	00:00:00	kblockd
21	?	00:00:00	ata_sff
22	?	00:00:00	khubd
23	?	00:00:00	md
24	?	00:00:00	devfreq_wq

4. 执行下列命令和操作：

```
$sleep180
^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。

```
root@moocos-VirtualBox:/home/moocos# sleep 180
^Z
[1]+  Stopped                  sleep 180
root@moocos-VirtualBox:/home/moocos# jobs
[1]+  Stopped                  sleep 180
root@moocos-VirtualBox:/home/moocos# sleep 240&
[2] 3980
root@moocos-VirtualBox:/home/moocos# sleep 300&
[3] 3981
root@moocos-VirtualBox:/home/moocos# sleep 330&
[4] 3982
root@moocos-VirtualBox:/home/moocos# sleep 630&
[5] 3983
root@moocos-VirtualBox:/home/moocos# jobs
[1]+  Stopped                  sleep 180
[2]   Running                  sleep 240 &
[3]   Running                  sleep 300 &
[4]   Running                  sleep 330 &
[5]-  Running                  sleep 630 &
root@moocos-VirtualBox:/home/moocos# fg 4
sleep 330
^C
root@moocos-VirtualBox:/home/moocos# bg 1
[1]+ sleep 180 &
root@moocos-VirtualBox:/home/moocos# kill 5
root@moocos-VirtualBox:/home/moocos#
```

解释：

`$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

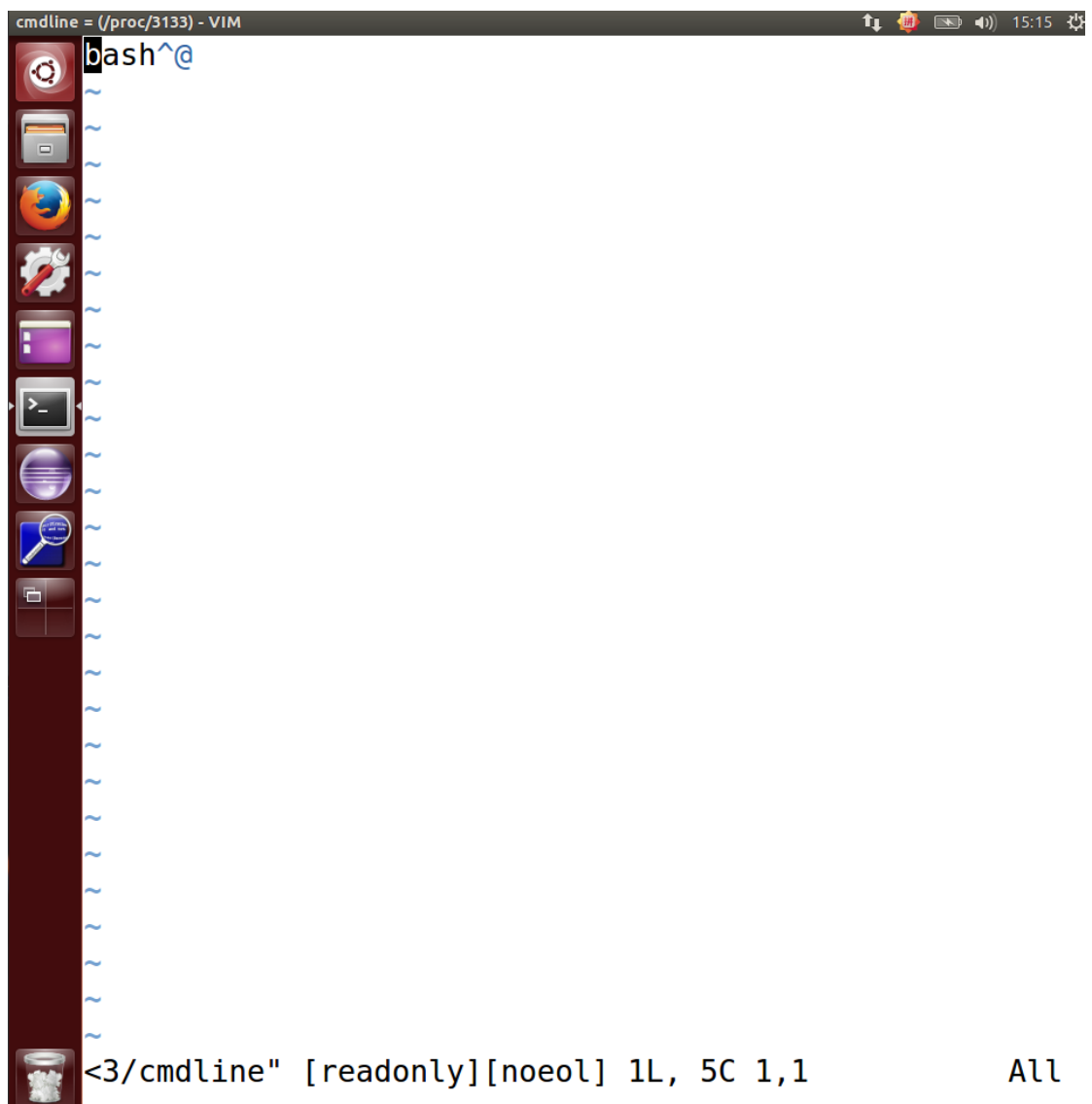
`$kill %5` #结束 5 号作业

5. 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/$psPID/status` 文件的内容。

下面为各文件内容：

1. `Cmdline` 内容：

进程的命令行参数 `bash^@`



2. Stat 内容:

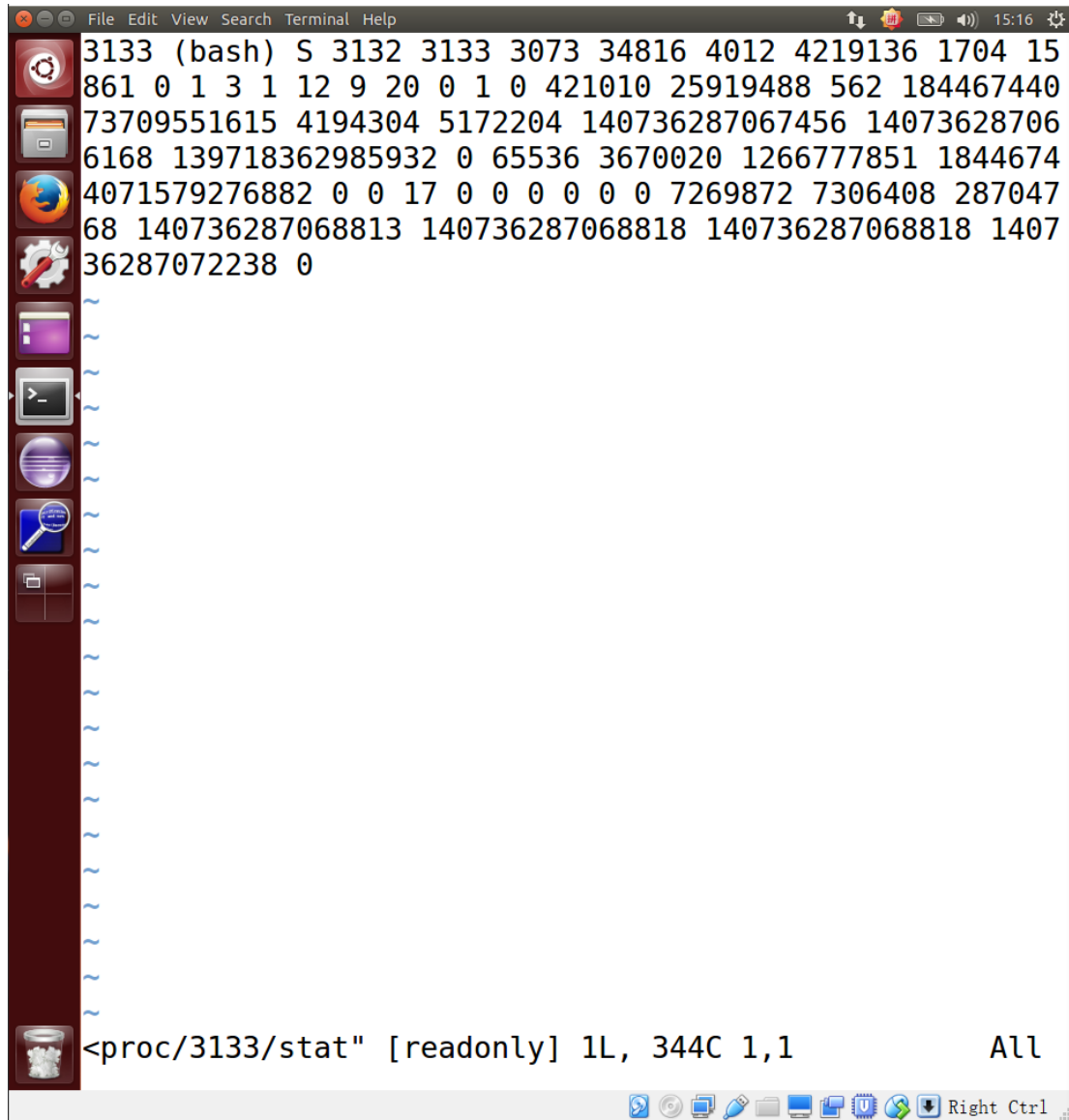
包含了进程状态信息，如 PID=3133

应用进程名为 bash

状态为 S(sleep)

父进程为 3132

进程组号为 3133 等



```
3133 (bash) S 3132 3133 3073 34816 4012 4219136 1704 15861 0 1 3 1 12 9 20 0 1 0 421010 25919488 562 184467440 73709551615 4194304 5172204 140736287067456 140736287066168 139718362985932 0 65536 3670020 1266777851 18446744071579276882 0 0 17 0 0 0 0 0 0 7269872 7306408 28704768 140736287068813 140736287068818 140736287068818 140736287072238 0
```

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

<code>session</code>	进程会话组 ID
<code>tty_nr</code>	当前进程的 <code>tty</code> 终点设备号
<code>tpgid</code>	控制进程终端的前台进程号
<code>flags</code>	进程标识位, 定义在 <code>include/linux/sched.h</code> 中的 <code>PF_*</code>
<code>minflt</code>	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
<code>cminflt</code>	当前进程等待子进程的 <code>minflt</code>
<code>majflt</code>	主要缺页中断的次数, 需要从磁盘加载内存页. 比如 <code>map</code> 文件
<code>majflt</code>	当前进程等待子进程的 <code>majflt</code>
<code>utime</code>	该进程处于用户态的时间, 单位 <code>jiffies</code>
<code>stime</code>	该进程处于内核态的时间, 单位 <code>jiffies</code>
<code>cutime</code>	当前进程等待子进程的 <code>utime</code>
<code>cstime</code>	当前进程等待子进程的 <code>utime</code>
<code>priorit</code>	进程优先级
<code>nice</code>	<code>nice</code> 值, 取值范围[19, -20]
<code>num_threads</code>	线程个数
<code>itrealvalue</code>	该字段已废弃, 恒等于 0
<code>starttime</code>	自系统启动后的进程创建时间, 单位 <code>jiffies</code>
<code>vsize</code>	进程的虚拟内存大小, 单位为 <code>bytes</code>
<code>rss</code>	进程独占内存+共享库, 单位 <code>pages</code>
<code>rsslim</code>	<code>rss</code> 大小上限

3. Status 内容:

包含了进程名 `name`

状态 `state`

线程组内第一个线程 id `tgid`

父进程 `PPID`

跟踪进程 id: `Tracepid`

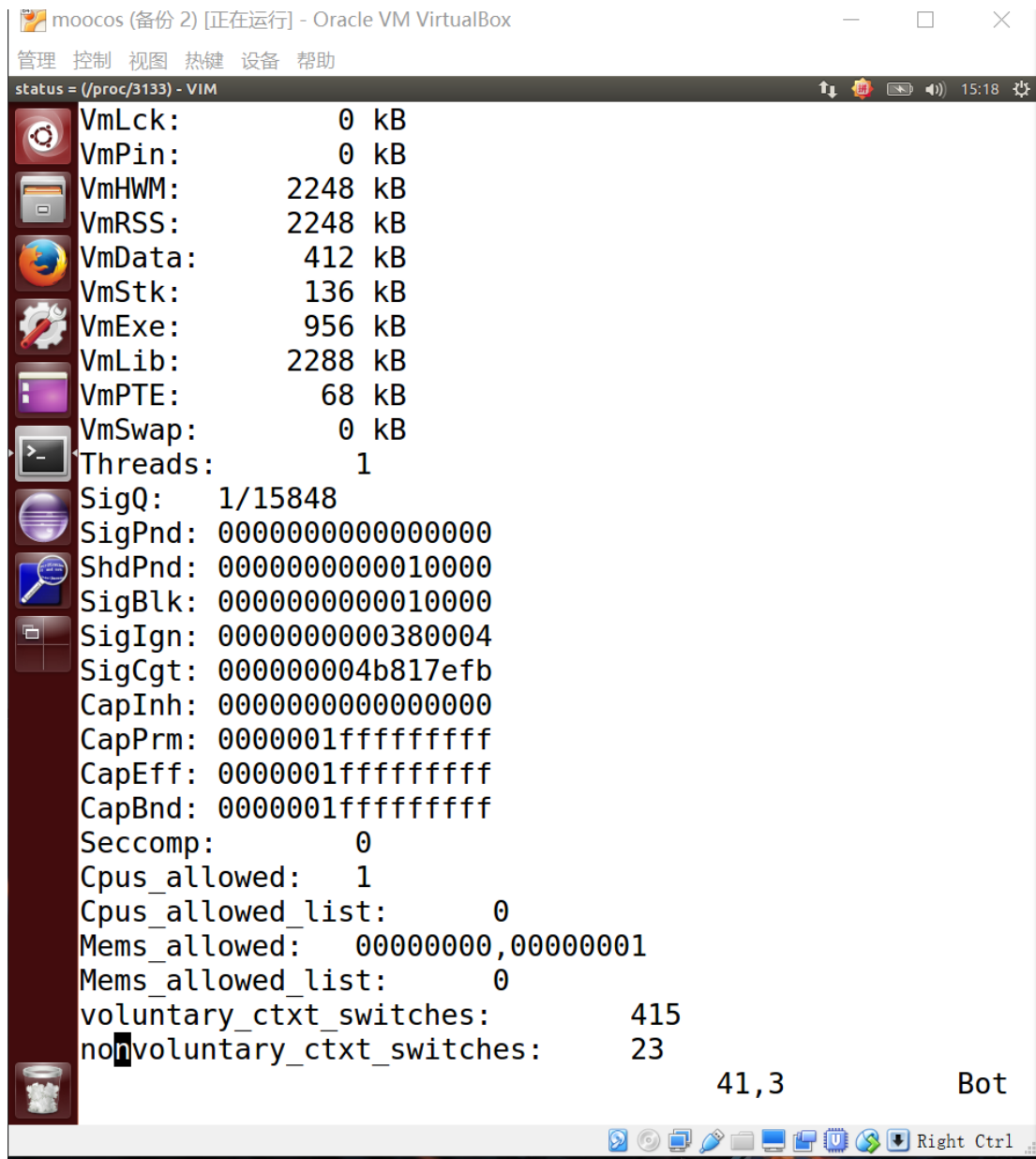
进程对应的 `uid`, `Gid`

`FDSize` 进程使用的文件句柄大小

`Groups` 组信息

以及进程所使用的虚拟内存以及实际内存信息等。。

```
status = (/proc/3133) - VIM
Name:  bash
State:  S (sleeping)
Tgid:  3133
Ngid:  0
Pid:   3133
PPid:  3132
TracerPid: 0
Uid:   0      0      0      0
Gid:   0      0      0      0
FDSize: 256
Groups: 0
VmPeak:  25316 kB
VmSize:  25312 kB
VmLck:    0 kB
VmPin:    0 kB
VmHWM:    2248 kB
VmRSS:    2248 kB
VmData:   412 kB
VmStk:    136 kB
VmExe:    956 kB
VmLib:    2288 kB
VmPTE:     68 kB
VmSwap:    0 kB
Threads:  1
SigQ:    1/15848
SigPnd:  0000000000000000
ShdPnd:  00000000000010000
SigBlk:  00000000000010000
<c/3133/status" [readonly] 41L, 776C 1,1                                Top
```

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

6. 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。

Nice 与 renice 命令：

用户状态可以降低进程优先级，只有在 root 下才能提高优先级。

```

root@moocos-VirtualBox:/home/moocos# nice --5 sleep 500
0&
[4] 4683
root@moocos-VirtualBox:/home/moocos# ps
  PID TTY          TIME CMD
 4451 pts/0        00:00:00 su
 4453 pts/0        00:00:00 bash
 4502 pts/0        00:00:00 ftp
 4635 pts/0        00:00:00 sleep
 4637 pts/0        00:00:00 sleep
 4683 pts/0        00:00:00 sleep
 4684 pts/0        00:00:00 ps
root@moocos-VirtualBox:/home/moocos# renice 10 4683
4683 (process ID) old priority -5, new priority 10
root@moocos-VirtualBox:/home/moocos#

```

```

moocos-> nice -10 sleep 3000&
[1] 4704
[~]
moocos-> ps
  PID TTY          TIME CMD
 4339 pts/0        00:00:00 bash
 4429 pts/0        00:00:00 ftp
 4434 pts/0        00:00:00 top
 4436 pts/0        00:00:00 top
 4437 pts/0        00:00:00 grep
 4441 pts/0        00:00:00 ftp
 4688 pts/0        00:00:00 bash
 4704 pts/0        00:00:00 sleep
 4707 pts/0        00:00:00 ps
[~]
moocos-> renice -10 4704
renice: failed to set priority for 4704 (process ID):
permission denied
[~]
moocos->

```

使用 top 命令修改优先级:

1. 进入 top 后按 r 后输入进程 pid
2. 输入需要的优先级 (-20, 19)
3. 在 root 下可以提升优先级, 普通用户只能下降优先级。

```

top - 16:20:08 up 3:20, 6 users, load average: 0.17,
Tasks: 180 total, 3 running, 170 sleeping, 7 stoppe
%Cpu(s): 5.8 us, 0.9 sy, 0.4 ni, 92.8 id, 0.1 wa,
KiB Mem: 2050088 total, 1494080 used, 556008 free,
KiB Swap: 2095100 total, 0 used, 2095100 free.
PID to renice [default pid = 1724] 4704

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU
1724	moocos	20	0	1424192	269032	39552	S	69.9
1134	root	20	0	304968	101576	10420	R	12.7
1	root	20	0	38012	7392	1508	S	0.0
2	root	20	0	0	0	0	S	0.0
3	root	20	0	0	0	0	S	0.0
4	root	20	0	0	0	0	S	0.0
5	root	0	-20	0	0	0	S	0.0
7	root	20	0	0	0	0	S	0.0
8	root	20	0	0	0	0	R	0.0
9	root	20	0	0	0	0	S	0.0
10	root	39	19	0	0	0	S	0.0
11	root	rt	0	0	0	0	S	0.0
12	root	rt	0	0	0	0	S	0.0
13	root	0	-20	0	0	0	S	0.0