实验二 进程控制

16281002 杜永坤 计科1601

## 实验题目：

根据课堂所学内容和基础知识介绍，完成实验题目。

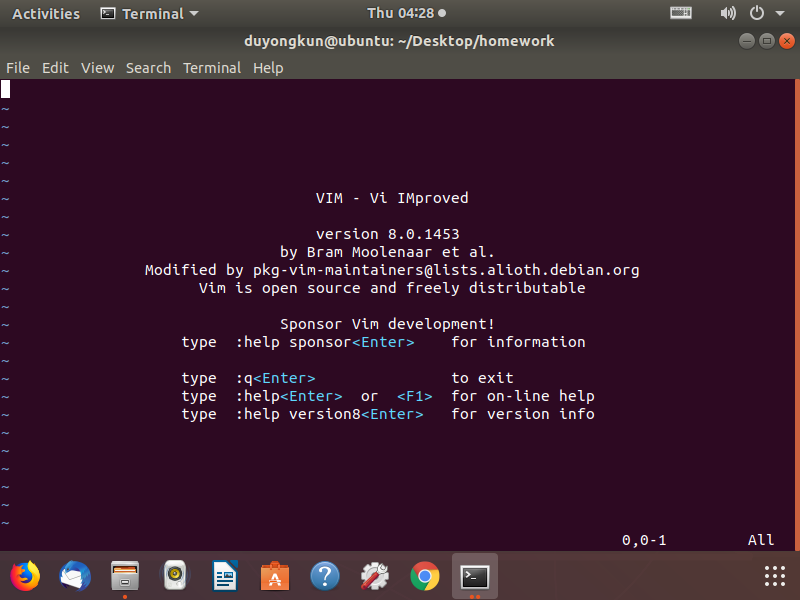
* 1、打开一个vi进程。通过ps命令以及选择合适的参数，只显示名字为vi的进程。寻找vi进程的父进程，直到init进程为止。记录过程中所有进程的ID和父进程ID。将得到的进程树和由pstree命令的得到的进程树进行比较。

实验过程：

先打开一个终端

输入vi 回车

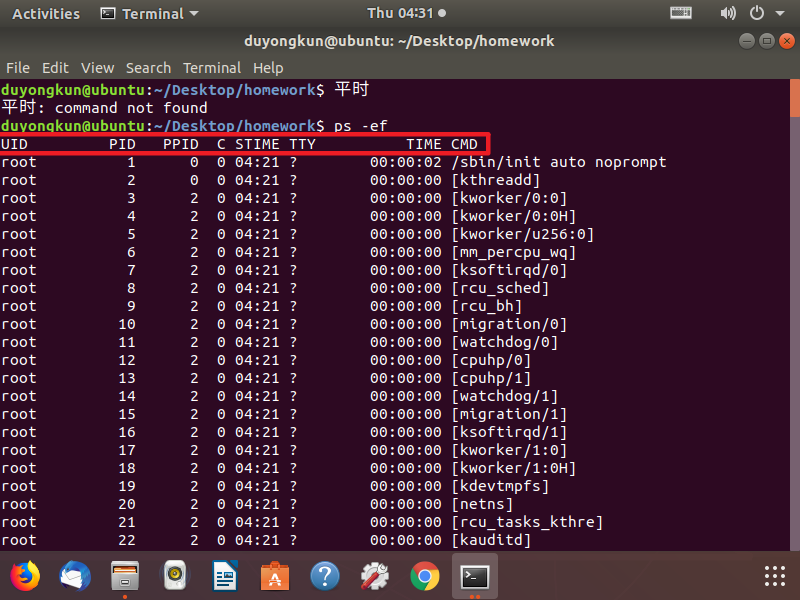
进入到如下界面：



然后打开一个新的终端

输入 ps -ef

 ps    -ef            显示整个环境（主机）中所有进程详细信息



UID ：程序被该 UID 所拥有

PID ：就是这个程序的 ID

PPID ：则是其上级父程序的ID

C ：CPU使用的资源百分比

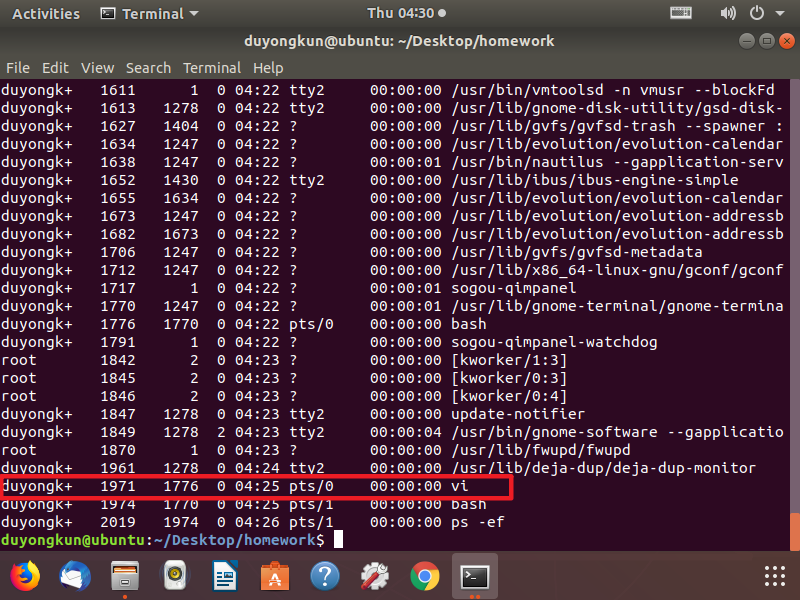
STIME ：系统启动时间

TTY ：登入者的终端机位置

TIME ：使用掉的CPU时间。

CMD ：所下达的是什么指令

然后寻找CMD为vi的进程



可以看到进程为1971 然后它的父进程为1776

可以得到

1972-1776-1770-1247-1

使用pstree -p

得到下面的进程树

可以看到所以的进程都是依附在systemd(1)这个进程下面，它的进程PID是1

然后systemd(1247)是它的一个子进程再往后依次为gnome-terminal-(1770)──bash(1776)───vi(1971)

systemd(1)─┬─ModemManager(568)─┬─{ModemManager}(655)

│ └─{ModemManager}(665)

├─NetworkManager(557)─┬─{NetworkManager}(692)

│ └─{NetworkManager}(698)

├─VGAuthService(595)

├─accounts-daemon(576)─┬─{accounts-daemon}(623)

│ └─{accounts-daemon}(658)

├─acpid(565)

├─avahi-daemon(579)───avahi-daemon(624)

├─boltd(1136)─┬─{boltd}(1141)

│ └─{boltd}(1143)

├─colord(1213)─┬─{colord}(1214)

│ └─{colord}(1217)

├─cron(555)

├─cups-browsed(646)─┬─{cups-browsed}(695)

│ └─{cups-browsed}(696)

├─cupsd(598)───dbus(693)

├─dbus-daemon(530)

├─dbus-daemon(1316)

├─fcitx(1302)───{fcitx}(1784)

├─fcitx-dbus-watc(1323)

├─fwupd(1870)─┬─{fwupd}(1874)

│ ├─{fwupd}(1876)

│ ├─{fwupd}(1877)

│ └─{fwupd}(1878)

├─gdm3(776)─┬─gdm-session-wor(797)─┬─gdm-wayland-ses(973)─┬─gnome-session-b+

│ │ │ ├─{gdm-wayland-se+

│ │ │ └─{gdm-wayland-se+

│ │ ├─{gdm-session-wor}(810)

│ │ └─{gdm-session-wor}(811)

│ ├─gdm-session-wor(1243)─┬─gdm-x-session(1265)─┬─Xorg(1267)───{X+

│ │ │ ├─gnome-session-b+

│ │ │ ├─{gdm-x-session}+

│ │ │ └─{gdm-x-session}+

│ │ ├─{gdm-session-wor}(1244)

│ │ └─{gdm-session-wor}(1245)

│ ├─{gdm3}(792)

│ └─{gdm3}(794)

├─gnome-keyring-d(1261)─┬─{gnome-keyring-d}(1262)

│ ├─{gnome-keyring-d}(1263)

│ └─{gnome-keyring-d}(1395)

├─gsd-printer(1573)─┬─{gsd-printer}(1577)

│ └─{gsd-printer}(1578)

├─ibus-x11(1126)─┬─{ibus-x11}(1202)

│ └─{ibus-x11}(1203)

├─ibus-x11(1436)─┬─{ibus-x11}(1446)

│ └─{ibus-x11}(1447)

├─irqbalance(552)───{irqbalance}(564)

├─kerneloops(1083)

├─kerneloops(1087)

├─networkd-dispat(550)───{networkd-dispat}(745)

├─packagekitd(1140)─┬─{packagekitd}(1144)

│ └─{packagekitd}(1145)

├─polkitd(666)─┬─{polkitd}(668)

│ └─{polkitd}(672)

├─pulseaudio(1420)─┬─{pulseaudio}(1421)

│ └─{pulseaudio}(1422)

├─rsyslogd(563)─┬─{rsyslogd}(586)

│ ├─{rsyslogd}(587)

│ └─{rsyslogd}(588)

├─rtkit-daemon(1107)─┬─{rtkit-daemon}(1108)

│ └─{rtkit-daemon}(1109)

├─snapd(594)─┬─{snapd}(721)

│ ├─{snapd}(722)

│ ├─{snapd}(723)

│ ├─{snapd}(724)

│ ├─{snapd}(747)

│ ├─{snapd}(748)

│ ├─{snapd}(853)

│ ├─{snapd}(855)

│ ├─{snapd}(856)

│ ├─{snapd}(1893)

│ ├─{snapd}(1938)

│ └─{snapd}(1939)

├─sogou-qimpanel(1717)─┬─{sogou-qimpanel}(1753)

│ ├─{sogou-qimpanel}(1754)

│ ├─{sogou-qimpanel}(1755)

│ ├─{sogou-qimpanel}(1756)

│ ├─{sogou-qimpanel}(1757)

│ ├─{sogou-qimpanel}(1758)

│ ├─{sogou-qimpanel}(1759)

│ ├─{sogou-qimpanel}(1760)

│ ├─{sogou-qimpanel}(1761)

│ └─{sogou-qimpanel}(1762)

├─sogou-qimpanel-(1791)

├─systemd(910)─┬─(sd-pam)(918)

│ ├─at-spi-bus-laun(1095)─┬─dbus-daemon(1100)

│ │ ├─{at-spi-bus-laun}(1096)

│ │ ├─{at-spi-bus-laun}(1097)

│ │ └─{at-spi-bus-laun}(1099)

│ ├─at-spi2-registr(1102)─┬─{at-spi2-registr}(1103)

│ │ └─{at-spi2-registr}(1104)

│ ├─dbus-daemon(975)

│ ├─ibus-portal(1129)─┬─{ibus-portal}(1131)

│ │ └─{ibus-portal}(1132)

│ └─pulseaudio(1106)─┬─{pulseaudio}(1110)

│ └─{pulseaudio}(1111)

├─systemd(1247)─┬─(sd-pam)(1248)

│ ├─at-spi-bus-laun(1374)─┬─dbus-daemon(1379)

│ │ ├─{at-spi-bus-laun}(1375)

│ │ ├─{at-spi-bus-laun}(1376)

│ │ └─{at-spi-bus-laun}(1378)

│ ├─at-spi2-registr(1382)─┬─{at-spi2-registr}(1383)

│ │ └─{at-spi2-registr}(1384)

│ ├─dbus-daemon(1274)

│ ├─dconf-service(1466)─┬─{dconf-service}(1472)

│ │ └─{dconf-service}(1473)

│ ├─evolution-addre(1673)─┬─evolution-addre(1682)─┬─{evolutio+

│ │ │ ├─{evolutio+

│ │ │ ├─{evolutio+

│ │ │ ├─{evolutio+

│ │ │ └─{evolutio+

│ │ ├─{evolution-addre}(1676)

│ │ ├─{evolution-addre}(1677)

│ │ ├─{evolution-addre}(1678)

│ │ └─{evolution-addre}(1679)

│ ├─evolution-calen(1634)─┬─evolution-calen(1655)─┬─{evolutio+

│ │ │ ├─{evolutio+

│ │ │ ├─{evolutio+

│ │ │ ├─{evolutio+

│ │ │ ├─{evolutio+

│ │ │ ├─{evolutio+

│ │ │ ├─{evolutio+

│ │ │ └─{evolutio+

│ │ ├─{evolution-calen}(1645)

│ │ ├─{evolution-calen}(1647)

│ │ ├─{evolution-calen}(1649)

│ │ └─{evolution-calen}(1650)

│ ├─evolution-sourc(1455)─┬─{evolution-sourc}(1456)

│ │ ├─{evolution-sourc}(1457)

│ │ └─{evolution-sourc}(1458)

│ ├─gconfd-2(1712)

│ ├─gnome-shell-cal(1449)─┬─{gnome-shell-cal}(1452)

│ │ ├─{gnome-shell-cal}(1454)

│ │ ├─{gnome-shell-cal}(1470)

│ │ ├─{gnome-shell-cal}(1471)

│ │ └─{gnome-shell-cal}(1632)

│ ├─gnome-terminal-(1770)─┬─bash(1776)───vi(1971)

│ │ ├─bash(1974)───pstree(2044)

│ │ ├─{gnome-terminal-}(1771)

│ │ ├─{gnome-terminal-}(1772)

│ │ └─{gnome-terminal-}(1773)

│ ├─goa-daemon(1463)─┬─{goa-daemon}(1482)

│ │ ├─{goa-daemon}(1484)

│ │ └─{goa-daemon}(1485)

│ ├─goa-identity-se(1488)─┬─{goa-identity-se}(1492)

│ │ ├─{goa-identity-se}(1493)

│ │ └─{goa-identity-se}(1495)

│ ├─gvfs-afc-volume(1494)─┬─{gvfs-afc-volume}(1496)

│ │ ├─{gvfs-afc-volume}(1497)

│ │ └─{gvfs-afc-volume}(1499)

│ ├─gvfs-goa-volume(1500)─┬─{gvfs-goa-volume}(1501)

│ │ └─{gvfs-goa-volume}(1502)

│ ├─gvfs-gphoto2-vo(1481)─┬─{gvfs-gphoto2-vo}(1489)

│ │ └─{gvfs-gphoto2-vo}(1491)

│ ├─gvfs-mtp-volume(1477)─┬─{gvfs-mtp-volume}(1478)

│ │ └─{gvfs-mtp-volume}(1480)

│ ├─gvfs-udisks2-vo(1469)─┬─{gvfs-udisks2-vo}(1474)

│ │ └─{gvfs-udisks2-vo}(1475)

│ ├─gvfsd(1404)─┬─gvfsd-trash(1627)─┬─{gvfsd-trash}(1630)

│ │ │ └─{gvfsd-trash}(1631)

│ │ ├─{gvfsd}(1405)

│ │ └─{gvfsd}(1406)

│ ├─gvfsd-fuse(1409)─┬─{gvfsd-fuse}(1412)

│ │ ├─{gvfsd-fuse}(1413)

│ │ ├─{gvfsd-fuse}(1414)

│ │ ├─{gvfsd-fuse}(1415)

│ │ └─{gvfsd-fuse}(1416)

│ ├─gvfsd-metadata(1706)─┬─{gvfsd-metadata}(1707)

│ │ └─{gvfsd-metadata}(1708)

│ ├─ibus-portal(1439)─┬─{ibus-portal}(1442)

│ │ └─{ibus-portal}(1443)

│ └─nautilus(1638)─┬─{nautilus}(1642)

│ ├─{nautilus}(1643)

│ └─{nautilus}(1646)

├─systemd-journal(335)

├─systemd-logind(585)

├─systemd-resolve(420)

├─systemd-timesyn(419)───{systemd-timesyn}(434)

├─systemd-udevd(346)

├─udisksd(561)─┬─{udisksd}(631)

│ ├─{udisksd}(660)

│ ├─{udisksd}(694)

│ └─{udisksd}(777)

├─unattended-upgr(727)───{unattended-upgr}(775)

├─upowerd(1012)─┬─{upowerd}(1013)

│ └─{upowerd}(1014)

├─vmtoolsd(414)───{vmtoolsd}(513)

├─vmtoolsd(1611)─┬─{vmtoolsd}(1628)

│ ├─{vmtoolsd}(1629)

│ └─{vmtoolsd}(1653)

├─vmware-vmblock-(411)─┬─{vmware-vmblock-}(412)

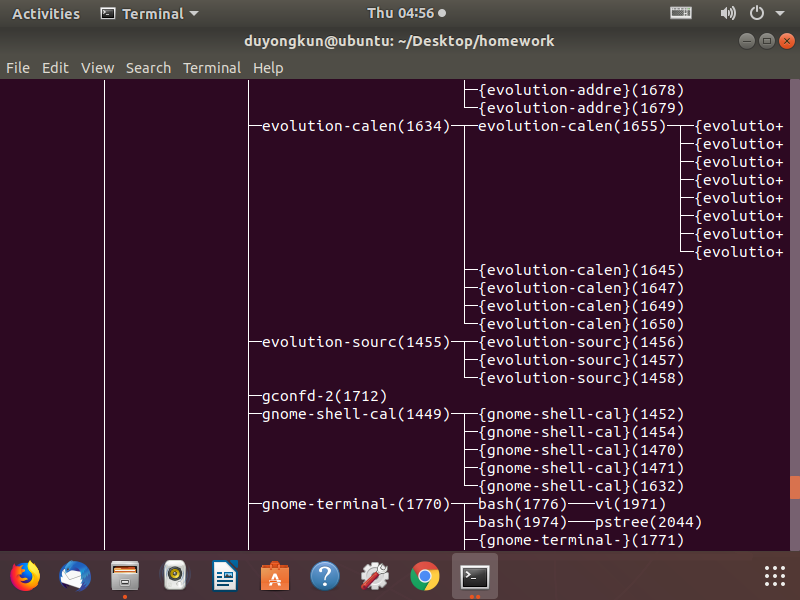
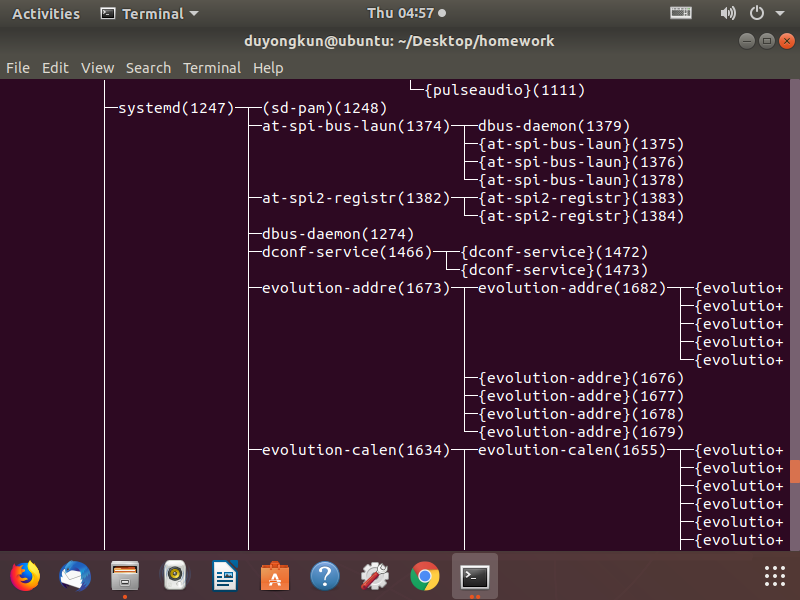
│ └─{vmware-vmblock-}(413)

├─whoopsie(1081)─┬─{whoopsie}(1085)

│ └─{whoopsie}(1086)

└─wpa\_supplicant(574)

可以看到与1972-1776-1770-1247-1符合



* 2、编写程序，首先使用fork系统调用，创建子进程。在父进程中继续执行空循环操作；在子进程中调用exec打开vi编辑器。然后在另外一个终端中，通过ps –Al命令、ps aux或者top等命令，查看vi进程及其父进程的运行状态，理解每个参数所表达的意义。选择合适的命令参数，对所有进程按照cpu占用率排序。

根据实验要求编写代码：

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <unistd.h>

int main ()

{

pid\_t fpid; //fpid表示fork函数返回的值

fpid=fork();

if (fpid < 0)

printf("error in fork!");

else if (fpid == 0)

{

printf("i am the child process, my process id is %d\n",getpid());

int ret ;

ret = execl("/usr/bin/vi","vi","test.txt",NULL);

if(ret=-1)

perror("execl");

}

else {

while(1)

printf("i am the parent process, my process id is %d\n",getpid());

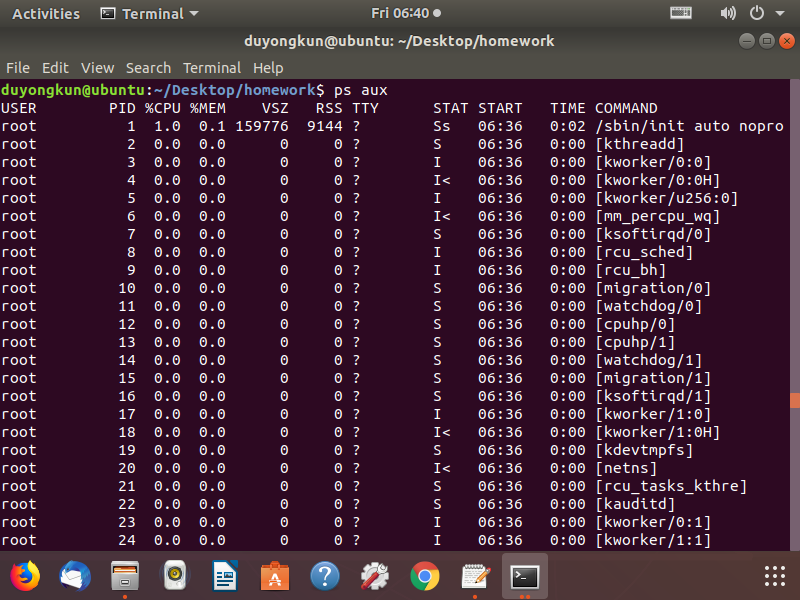
}

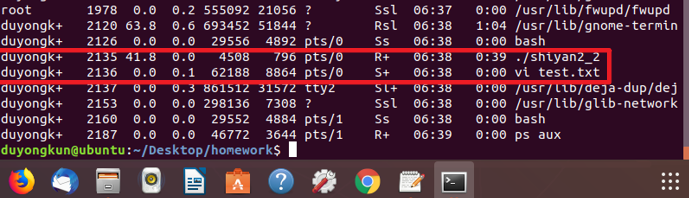
return 0;

}

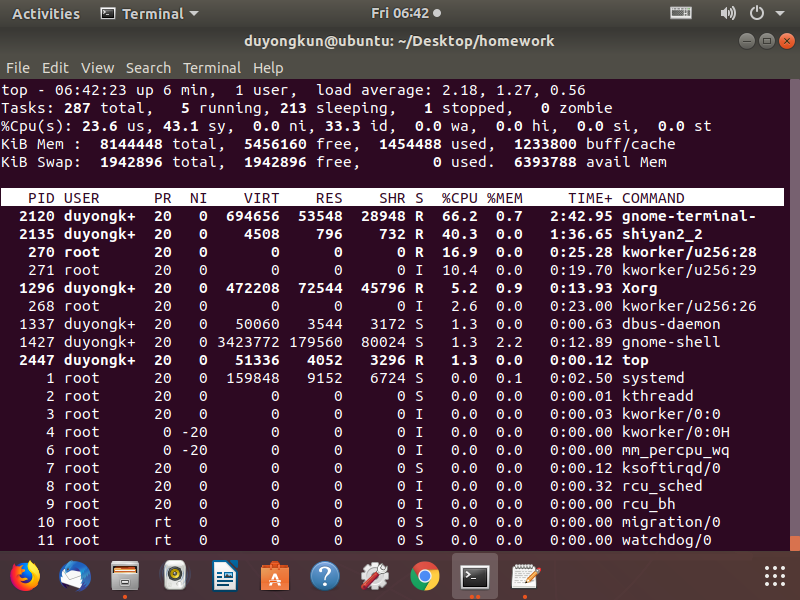
运行后在另个一个终端查看相关进程：

执行 ps aux 命令：





使用top命令查看各个进程的cpu占用率：



各个参数的意义：

• USER：该进程属于使用者账号的名字  
• PID ：该进程的进程ID号。   
• %CPU：该进程使用掉的 CPU 资源百分比；   
• %MEM：该进程所占用的物理内存百分比；   
• VSZ ：该进程使用掉的虚拟内存量 (Kbytes)   
• RSS ：该进程占用的固定的内存量 (Kbytes)   
• TTY ：该进程是在那个终端机上面运作，若与终端机无关，则显示 ?，另外， tty1-tty6 是本机上面的登入者程序，若为 pts/0 等等的，则表示为由网络连接进主机的程序。   
• STAT：该程序目前的状态，主要的状态有：   
R ：该程序目前正在运作，或者是可被运作；   
S ：该程序目前正在睡眠当中 (可说是 idle 状态啦！)，但可被某些讯号(signal) 唤醒。   
T ：该程序目前正在侦测或者是停止了；   
Z ：该程序应该已经终止，但是其父程序却无法正常的终止他，造成 zombie (疆尸) 程序的状态   
• START：该进程被触发启动的时间；   
• TIME ：该进程实际使用 CPU 运作的时间。   
• COMMAND：该程序的实际指令

* 3、使用fork系统调用，创建如下进程树，并使每个进程输出自己的ID和父进程的ID。观察进程的执行顺序和运行状态的变化。

根据实验要求编写代码：

#include "stdio.h"

#include "sys/types.h"

#include "unistd.h"

#include "stdlib.h"

#define HASPRO -10

char \*a;

int main()

{

pid\_t p1,p2,p3,p4,p5;

int cnt=0;

while((p1=fork()) == -1);

if(!p1)

{

while((p2=fork()) == -1);

if(!p2)

{

while ((p4=fork())==-1);

if (!p4)

{

while(1)

{ sleep(1);

printf(" p4 pid %d, parent p2 pid %d\n",getpid(),getppid());

}

}

else

{

while ((p5=fork())==-1);

if (!p5)

{

while(1)

{sleep(1);

printf(" p5 pid %d, parent p2 pid %d\n",getpid(),getppid());

}

}

else

{

;

}

}

while(1)

{

sleep(1);

printf("p2 pid %d, parent p1 pid %d\n",getpid(),getppid());

}

}

else

{

while ((p3=fork())==-1);

if (!p3)

{

while(1)

{sleep(1);

printf("p3 pid %d, parent p1 pid %d\n",getpid(),getppid());

}

}

else

{

;

}

}

while(1)

{sleep(1);

printf("p1 pid %d, parent pid %d\n\n",getpid(),getppid());

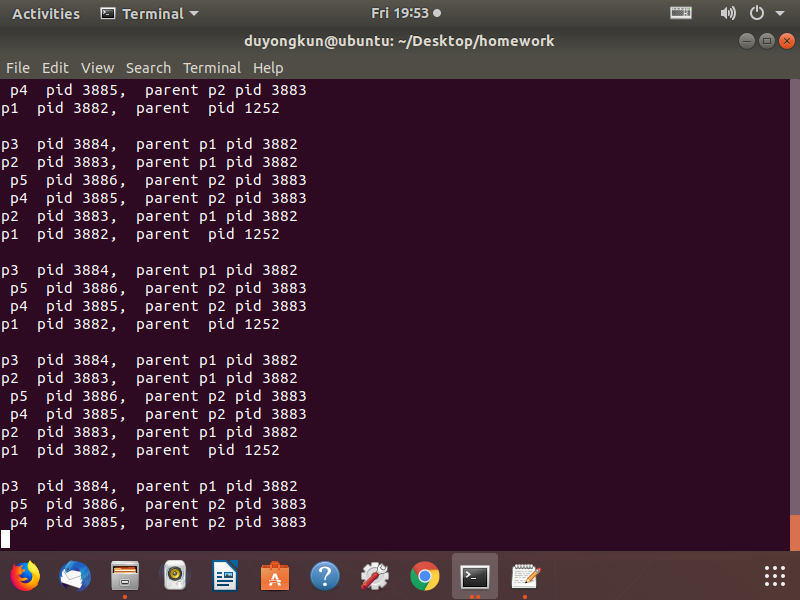
}

}

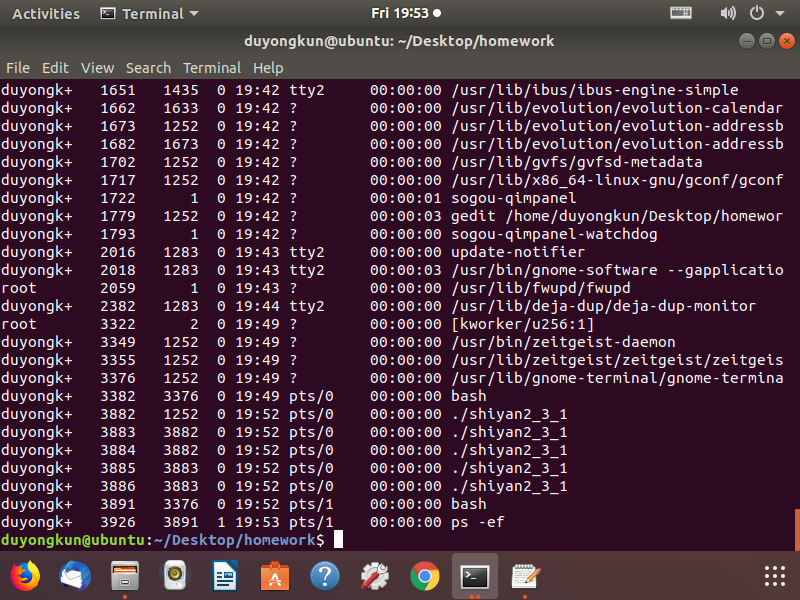
return 0;

}

编译执行代码后：



通过ps -ef查看进程：



可以看到p1：3882，p2：3883，p3:3884，p4:3885，p5：3886。

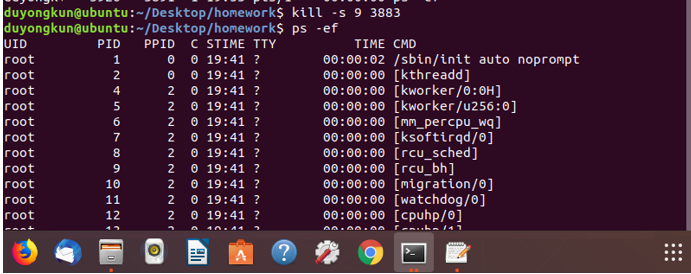
运行时可以看到这五个进程输出顺序并不确定，是因为系统中运行着基本完全相同的几个进程，这几个进程执行没有固定的先后顺序，哪个进程先执行要看系统的进程调度策略。

* 4、修改上述进程树中的进程，使得所有进程都循环输出自己的ID和父进程的ID。然后终止p2进程(分别采用kill -9 、自己正常退出exit()、段错误退出)，观察p1、p3、p4、p5进程的运行状态和其他相关参数有何改变。

1. 使用kill结束进程：

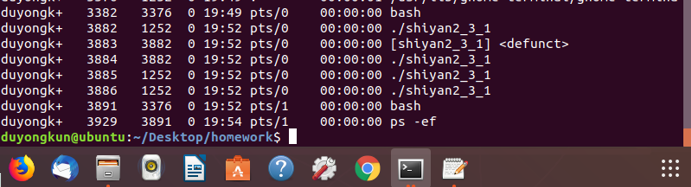
通过第三题可以看到p2的pid为3883：

执行：kill -s 9 3883



然后执行ps -ef 查看进程执行情况：

可以看到3883进程变成了僵尸进程，p4、p5的父进程变成了和p1相同的父进程。



1. 正常退出exit(0):

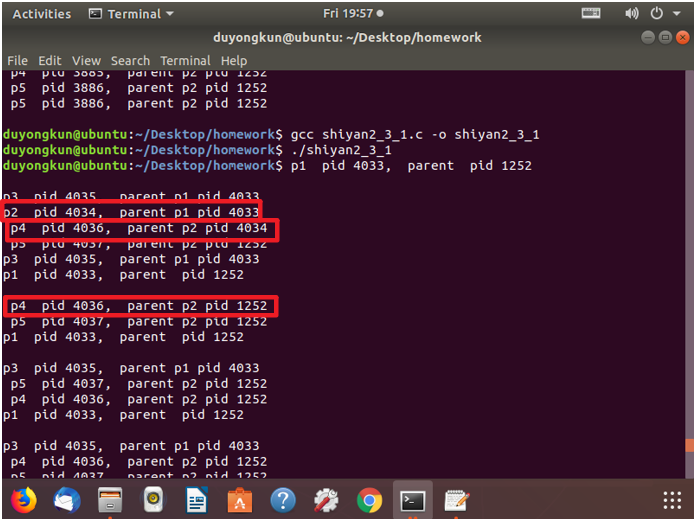
修改代码：

sleep(1);

printf("p2 pid %d, parent p1 pid %d\n",getpid(),getppid());

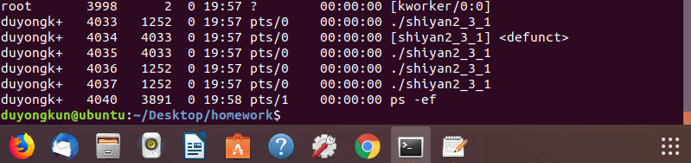
exit(0);//在p2中添加进程终止函数。

然后编译执行：



可以看到p2正常执行后执行exit()函数结束，p4在p2退出前执行结束，可以看到p4的父进程为p2:4034，p2进程结束后，后面就没有看到进程p2输出结果，p4、p5的父进程变成了：1252。

然后执行ps -ef 查看各个进程的执行情况。



可以看到p2:4034执行结束变成了僵尸进程，p4、p5的父进程变成了：1252。

1. 段错误退出进程：

修改代码：

char \*a;//定义全局变量

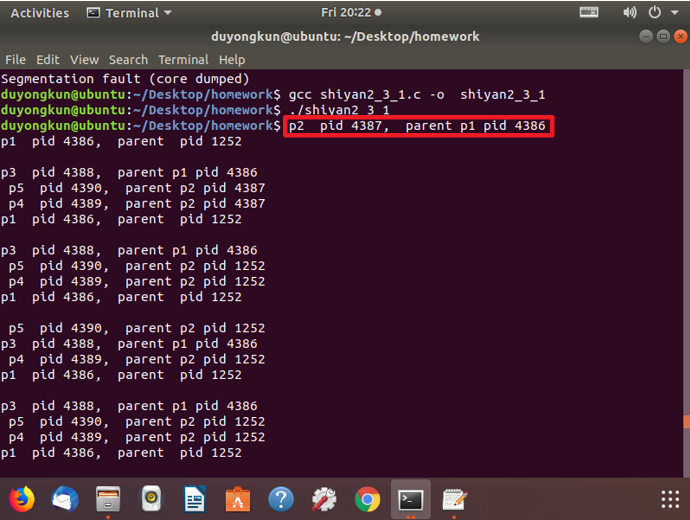
sleep(1);

printf("p2 pid %d, parent p1 pid %d\n",getpid(),getppid());

//exit(0);

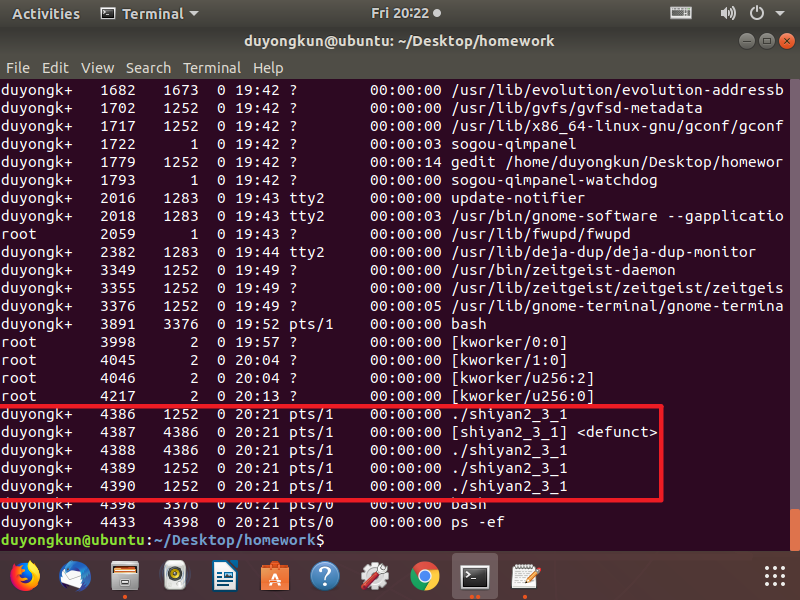
printf("定义未初始化指针\*a，制造段错误（数组中没有a[11]）：a[11]:%d\n",a[11]);

编译执行：



可以看到p2执行一次后由于发生段错误结束了进程。p4、p5的父进程变成了：1252。

执行ps -ef 查看进程执行情况：



可以看到p2：4387执行结束成为僵尸进程，p4：4389，p5：：4390的父进程变成了1252.