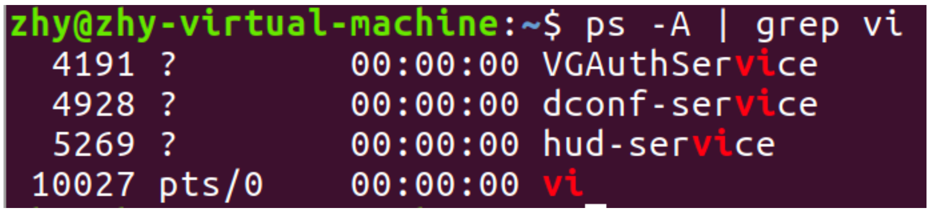
实验一 进程控制

16281025
张昊洋

1. 实验题目：

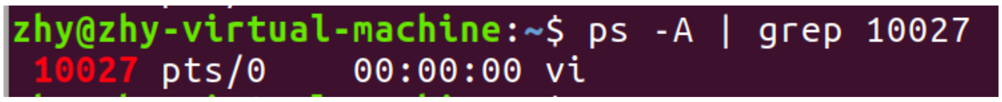
根据课堂所学内容和基础知识介绍，完成实验题目。

- 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。



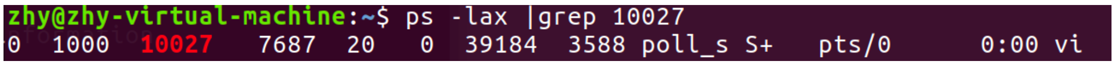
```
zhy@zhy-virtual-machine:~$ ps -A | grep vi
  4191 ?        00:00:00 VGAuthService
  4928 ?        00:00:00 dconf-service
  5269 ?        00:00:00 hud-service
 10027 pts/0    00:00:00 vi
```

若需要只显示名字为 vi 的，改用 pid 做 grep 的参数：

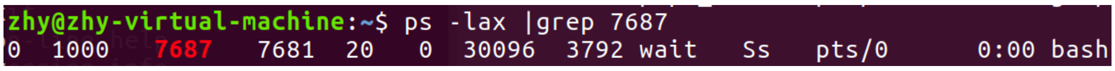


```
zhy@zhy-virtual-machine:~$ ps -A | grep 10027
 10027 pts/0    00:00:00 vi
```

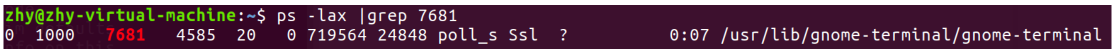
寻找父进程：



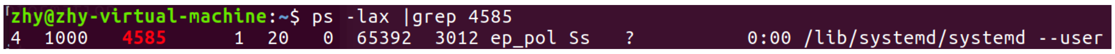
```
zhy@zhy-virtual-machine:~$ ps -lax |grep 10027
0  1000  10027   7687  20   0  39184  3588 poll_s S+   pts/0      0:00 vi
```



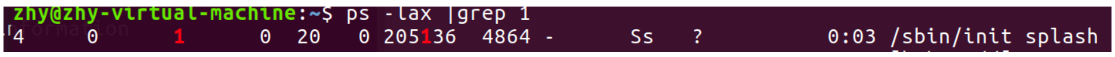
```
zhy@zhy-virtual-machine:~$ ps -lax |grep 7687
0  1000   7687   7681  20   0  30096  3792 wait   Ss   pts/0      0:00 bash
```



```
zhy@zhy-virtual-machine:~$ ps -lax |grep 7681
0  1000   7681   4585  20   0 719564 24848 poll_s Ssl  ?          0:07 /usr/lib/gnome-terminal/gnome-terminal
```



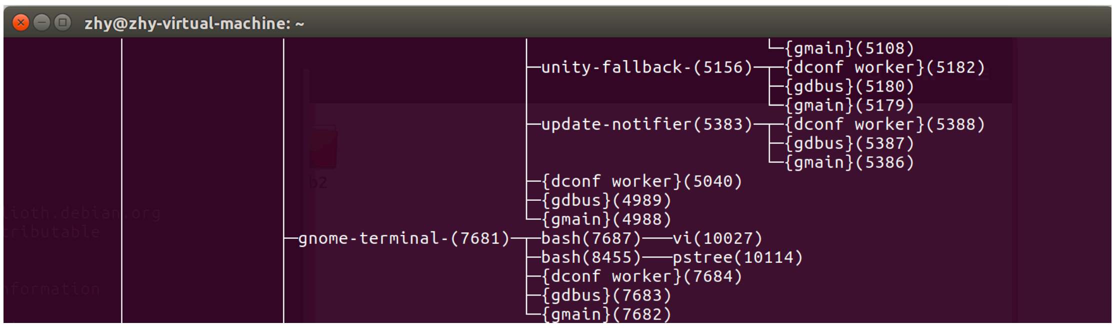
```
zhy@zhy-virtual-machine:~$ ps -lax |grep 4585
4  1000   4585      1  20   0  65392  3012 ep_pol Ss   ?          0:00 /lib/systemd/systemd --user
```



```
zhy@zhy-virtual-machine:~$ ps -lax |grep 1
4     0      1      0  20   0 205136  4864 -      Ss   ?          0:03 /sbin/init splash
```

10027>7687>7681>4585>1

使用 pstree -p:

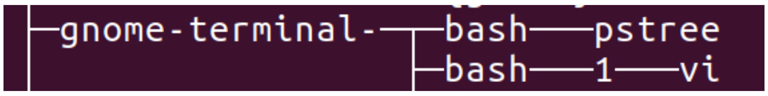


```
                                         │                 └─{gmain}(5108)
                                         ├─unity-fallback-(5156)─┬─{dconf worker}(5182)
                                         │                        ├─{gdbus}(5180)
                                         │                        └─{gmain}(5179)
                                         ├─update-notifier(5383)─┬─{dconf worker}(5388)
                                         │                        ├─{gdbus}(5387)
                                         │                        └─{gmain}(5386)
                                         ├─{dconf worker}(5040)
                                         ├─{gdbus}(4989)
                                         └─{gmain}(4988)
                   ├─gnome-terminal-(7681)─┬─bash(7687)───vi(10027)
                   │                       ├─bash(8455)───pstree(10114)
                   │                       ├─{dconf worker}(7684)
                   │                       ├─{gdbus}(7683)
                   │                       └─{gmain}(7682)
```

- 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。

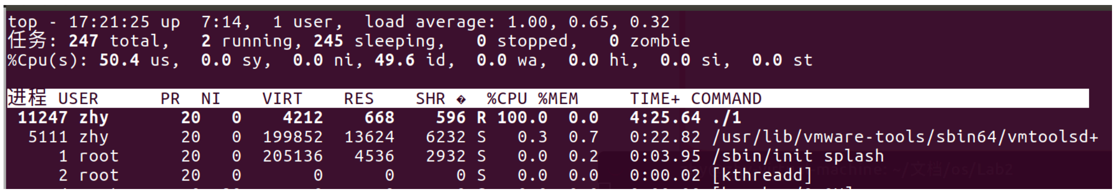
```
    #include <unistd.h>
    #include <stdio.h>
    int main ()
    {
        pid_t fpid;
        int count=0;
        fpid=fork();
        if (fpid < 0)
            printf("error in fork!");
        else if (fpid == 0) {
            execl("/usr/bin/vi","vi",NULL);
        }
        else {
            for(;;){}
        }
        return 0;
}
```



```
├gnome-terminal-┬bash──pstree
│               └bash──1──vi
```

可以看到第二行 vi 是 1 的子进程。

使用 top；c 按 CPU 使用排序：

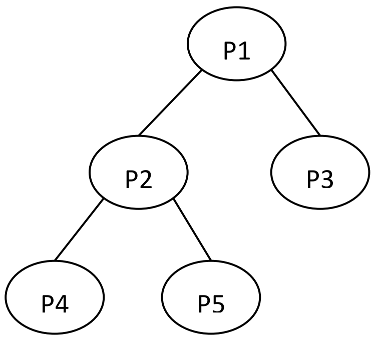


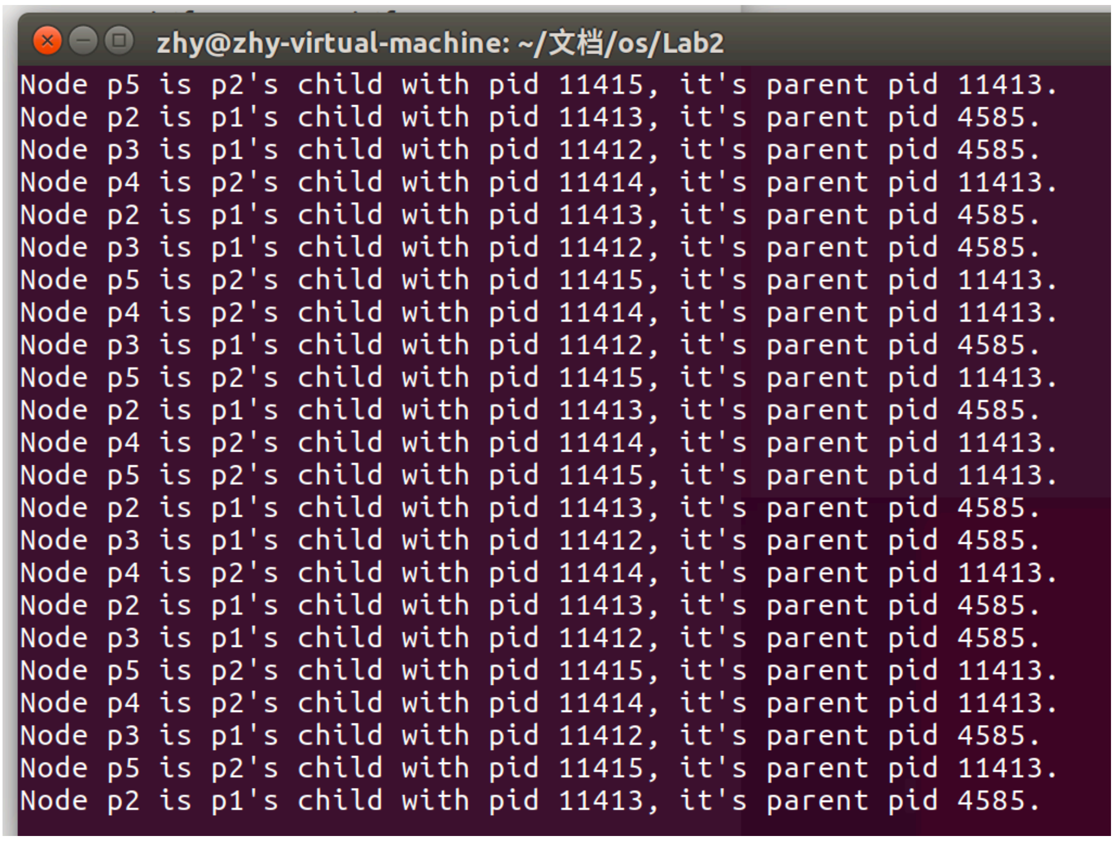
```
top - 17:21:25 up  7:14,  1 user,  load average: 1.00, 0.65, 0.32
任务: 247 total,   2 running, 245 sleeping,   0 stopped,   0 zombie
%Cpu(s): 50.4 us,  0.0 sy,  0.0 ni, 49.6 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st

进程 USER      PR  NI    VIRT    RES    SHR � %CPU %MEM     TIME+ COMMAND
 11247 zhy       20   0    4212    668    596 R 100.0  0.0   4:25.64 ./1
  5111 zhy       20   0  199852  13624   6232 S   0.3  0.7   0:22.82 /usr/lib/vmware-tools/sbin64/vmtoolsd+
     1 root      20   0  205136   4536   2932 S   0.0  0.2   0:03.95 /sbin/init splash
     2 root      20   0       0      0      0 S   0.0  0.0   0:00.02 [kthreadd]
```

循环中的 1 程序 CPU 占用最多。

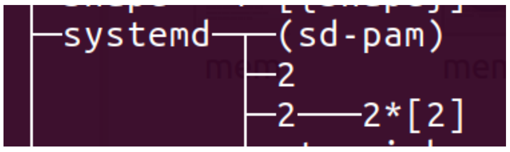
- 3、使用 fork 系统调用，创建如下进程树，并使每个进程输出自己的 ID 和父进程的 ID。观察进程的执行顺序和运行状态的变化。





```
Node p5 is p2's child with pid 11415, it's parent pid 11413.
Node p2 is p1's child with pid 11413, it's parent pid 4585.
Node p3 is p1's child with pid 11412, it's parent pid 4585.
Node p4 is p2's child with pid 11414, it's parent pid 11413.
Node p2 is p1's child with pid 11413, it's parent pid 4585.
Node p3 is p1's child with pid 11412, it's parent pid 4585.
Node p5 is p2's child with pid 11415, it's parent pid 11413.
Node p4 is p2's child with pid 11414, it's parent pid 11413.
Node p3 is p1's child with pid 11412, it's parent pid 4585.
Node p5 is p2's child with pid 11415, it's parent pid 11413.
Node p2 is p1's child with pid 11413, it's parent pid 4585.
Node p4 is p2's child with pid 11414, it's parent pid 11413.
Node p5 is p2's child with pid 11415, it's parent pid 11413.
Node p2 is p1's child with pid 11413, it's parent pid 4585.
Node p3 is p1's child with pid 11412, it's parent pid 4585.
Node p4 is p2's child with pid 11414, it's parent pid 11413.
Node p2 is p1's child with pid 11413, it's parent pid 4585.
Node p3 is p1's child with pid 11412, it's parent pid 4585.
Node p5 is p2's child with pid 11415, it's parent pid 11413.
Node p4 is p2's child with pid 11414, it's parent pid 11413.
Node p3 is p1's child with pid 11412, it's parent pid 4585.
Node p5 is p2's child with pid 11415, it's parent pid 11413.
Node p2 is p1's child with pid 11413, it's parent pid 4585.
```

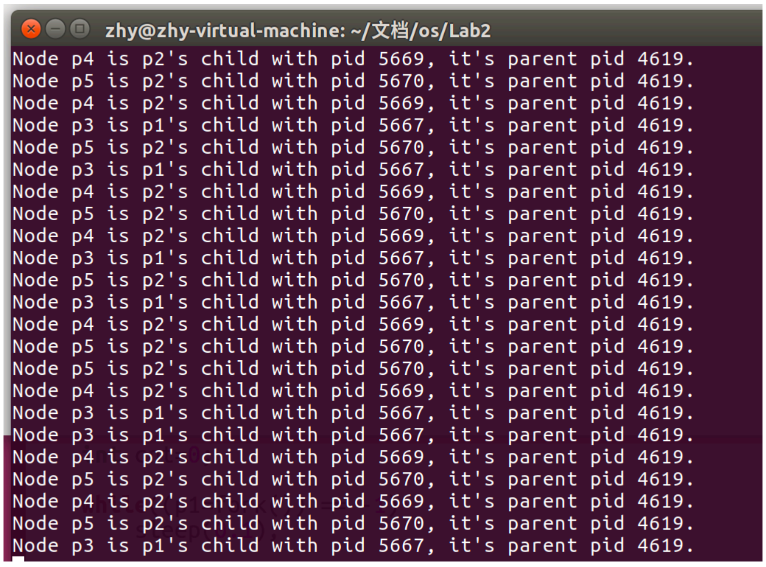
Pstree：



```
├─systemd─┬─(sd-pam)
│         ├─2
│         ├─2───2*[2]
```

程序名为 2，可以看到进程树呈上图相同结构。

- 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。

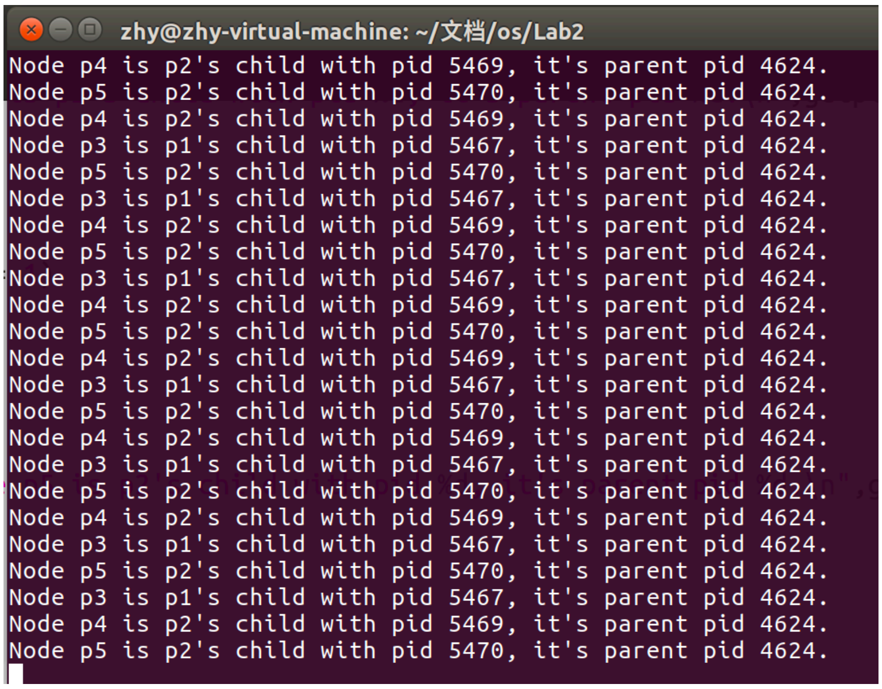


```
zhy@zhy-virtual-machine: ~/文档/os/Lab2
Node p4 is p2's child with pid 5669, it's parent pid 4619.
Node p5 is p2's child with pid 5670, it's parent pid 4619.
Node p4 is p2's child with pid 5669, it's parent pid 4619.
Node p3 is p1's child with pid 5667, it's parent pid 4619.
Node p5 is p2's child with pid 5670, it's parent pid 4619.
Node p3 is p1's child with pid 5667, it's parent pid 4619.
Node p4 is p2's child with pid 5669, it's parent pid 4619.
Node p5 is p2's child with pid 5670, it's parent pid 4619.
Node p4 is p2's child with pid 5669, it's parent pid 4619.
Node p3 is p1's child with pid 5667, it's parent pid 4619.
Node p5 is p2's child with pid 5670, it's parent pid 4619.
Node p3 is p1's child with pid 5667, it's parent pid 4619.
Node p4 is p2's child with pid 5669, it's parent pid 4619.
Node p5 is p2's child with pid 5670, it's parent pid 4619.
Node p5 is p2's child with pid 5670, it's parent pid 4619.
Node p4 is p2's child with pid 5669, it's parent pid 4619.
Node p3 is p1's child with pid 5667, it's parent pid 4619.
Node p3 is p1's child with pid 5667, it's parent pid 4619.
Node p4 is p2's child with pid 5669, it's parent pid 4619.
Node p5 is p2's child with pid 5670, it's parent pid 4619.
Node p4 is p2's child with pid 5669, it's parent pid 4619.
Node p5 is p2's child with pid 5670, it's parent pid 4619.
Node p3 is p1's child with pid 5667, it's parent pid 4619.
```

Kill -9 p2：

子进 **p4,p5** 继续运行，父进程变为 **p1**，**p3** 与其无关，继续运行。

修改程序，使 **p2 exit(0)**：



```
zhy@zhy-virtual-machine: ~/文档/os/Lab2
Node p4 is p2's child with pid 5469, it's parent pid 4624.
Node p5 is p2's child with pid 5470, it's parent pid 4624.
Node p4 is p2's child with pid 5469, it's parent pid 4624.
Node p3 is p1's child with pid 5467, it's parent pid 4624.
Node p5 is p2's child with pid 5470, it's parent pid 4624.
Node p3 is p1's child with pid 5467, it's parent pid 4624.
Node p4 is p2's child with pid 5469, it's parent pid 4624.
Node p5 is p2's child with pid 5470, it's parent pid 4624.
Node p3 is p1's child with pid 5467, it's parent pid 4624.
Node p4 is p2's child with pid 5469, it's parent pid 4624.
Node p5 is p2's child with pid 5470, it's parent pid 4624.
Node p4 is p2's child with pid 5469, it's parent pid 4624.
Node p3 is p1's child with pid 5467, it's parent pid 4624.
Node p5 is p2's child with pid 5470, it's parent pid 4624.
Node p4 is p2's child with pid 5469, it's parent pid 4624.
Node p3 is p1's child with pid 5467, it's parent pid 4624.
Node p5 is p2's child with pid 5470, it's parent pid 4624.
Node p4 is p2's child with pid 5469, it's parent pid 4624.
Node p3 is p1's child with pid 5467, it's parent pid 4624.
Node p5 is p2's child with pid 5470, it's parent pid 4624.
Node p3 is p1's child with pid 5467, it's parent pid 4624.
Node p4 is p2's child with pid 5469, it's parent pid 4624.
Node p5 is p2's child with pid 5470, it's parent pid 4624.
```

父进程变为 **p1**，同样继续运行。

段错误同样。