**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生姓名 | 黄东 | 学号 | 16281255 | 指导老师 | 何永忠 |
| 实验地点 | 九教402 | 实验时间 | 2019.3.14 | 班级 | 安全1601 |

1. **实验项目名称：**

实验二：进程控制

1. **实验目的：**
   * + 1. 加深对进程概念的理解，明确进程和程序的区别。
       2. 掌握Linux系统中的进程创建，管理和删除等操作。
       3. 熟悉使用Linux下的命令和工具，如man, find, grep, whereis, ps, pgrep, kill, ptree, top, vim, gcc，gdb等。
2. **实验环境**

Deepin15.6桌面版

1. **实验内容：**
2. 打开一个vi进程。
3. 编写程序，首先使用fork系统调用，创建子进程。
4. 使用fork系统调用，创建如下进程树，输出进程号。
5. 循环输出进程，终止p2进程后观察现象。
6. **实验过程：**
7. **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。**

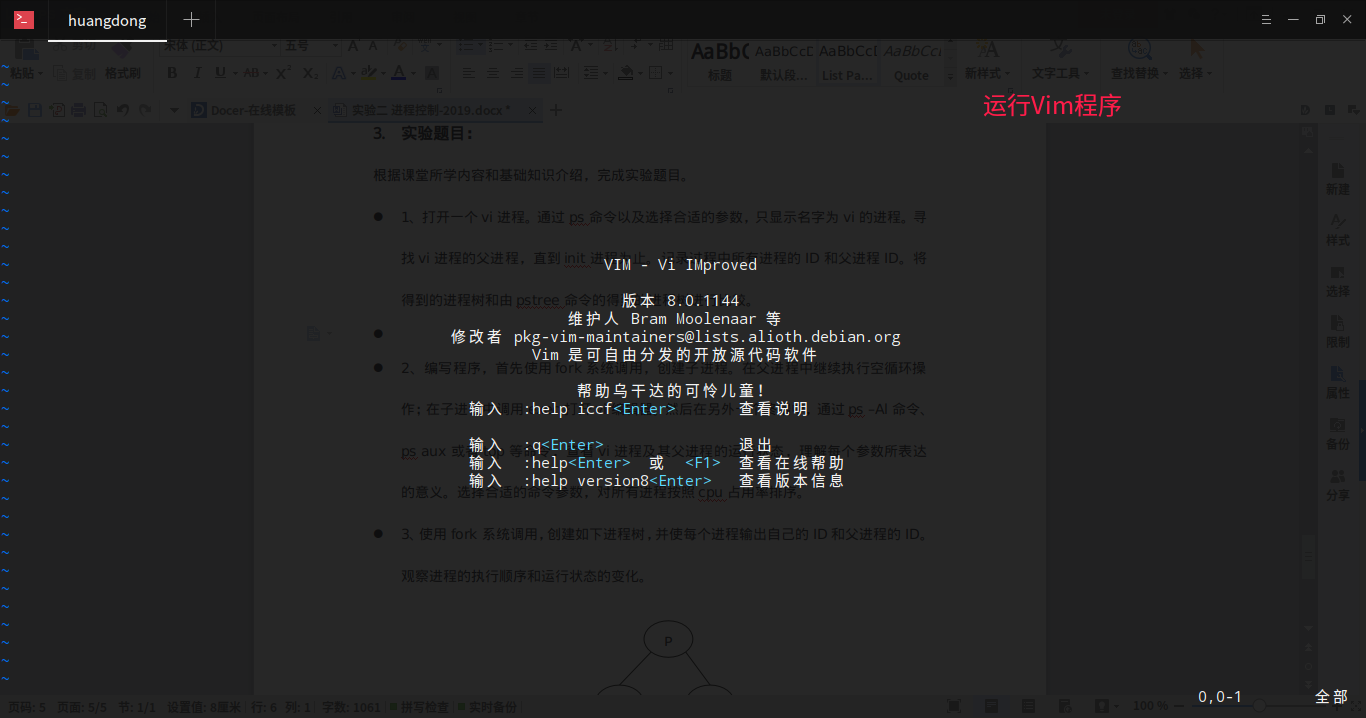


图1.1 窗口打开vi程序

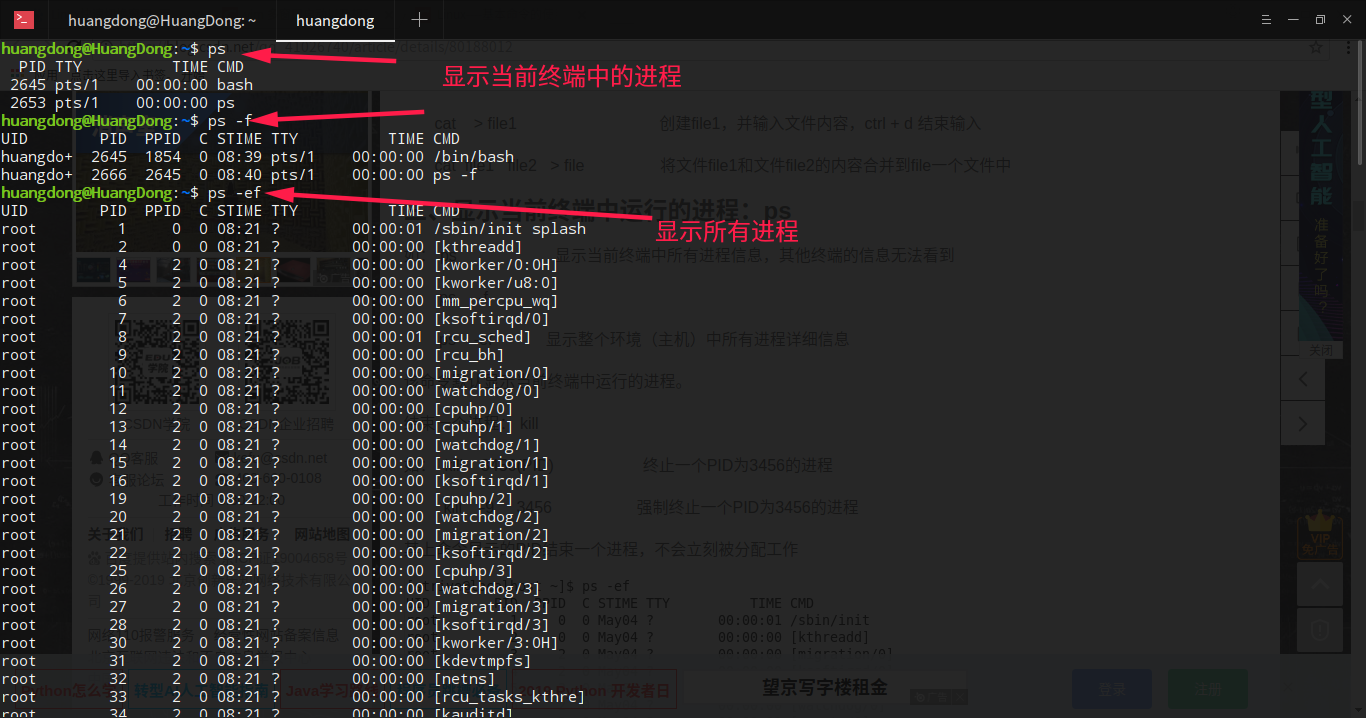


图1.2 查看所有进程

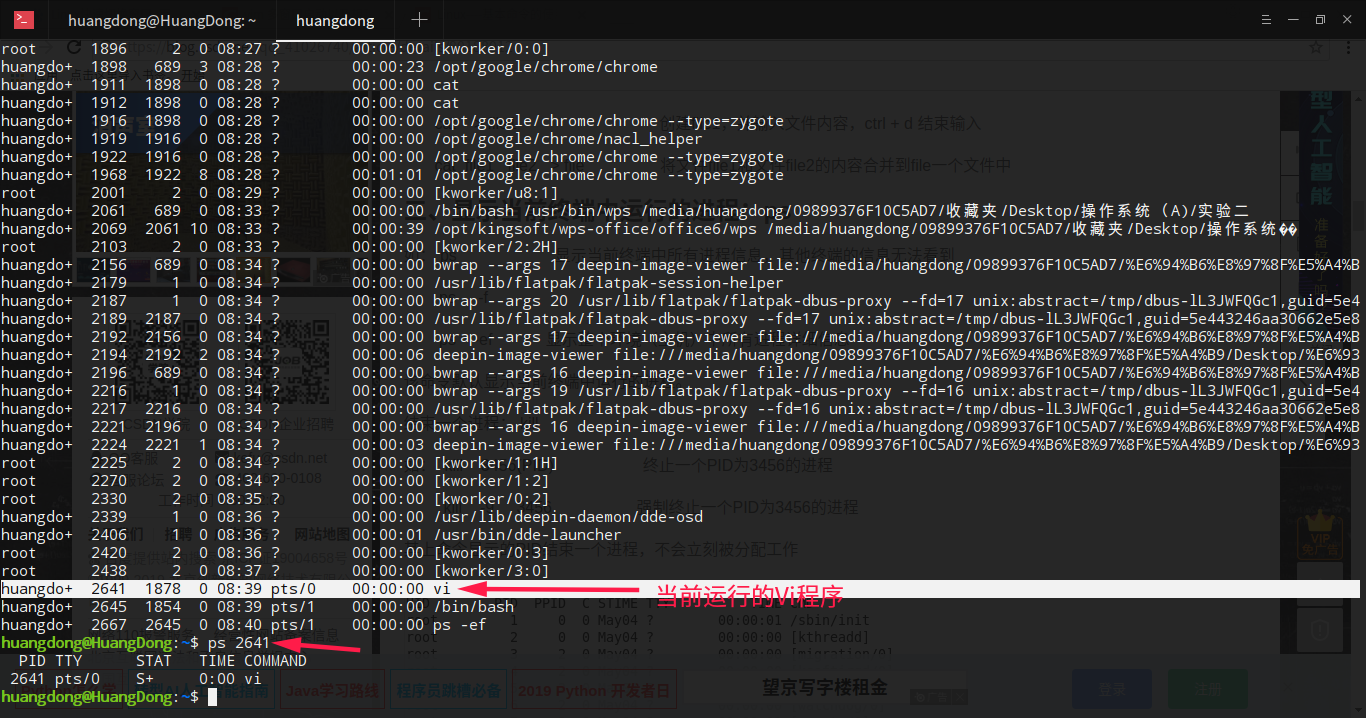


图1.3 查找VI程序和进程号

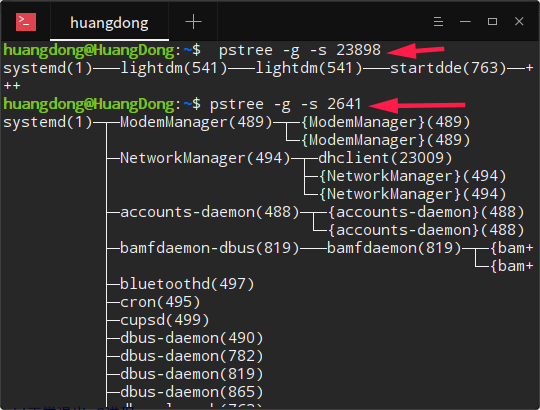


图1.4 显示进程树



图1.5 进入进程树

1. **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。**



图2.1 fork函数代码

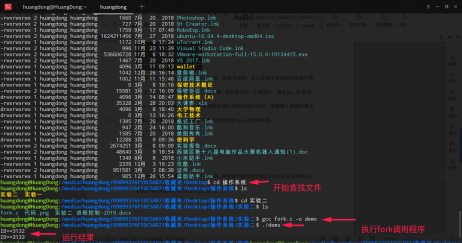


图2.2 显示进程号



图2.3 循环执行程序

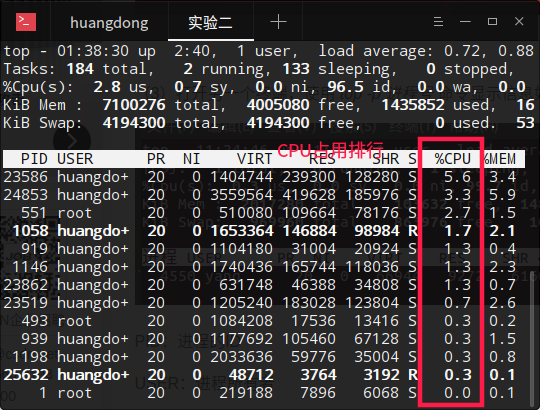


图2.4 显示CPU占用列表

实验收获:此次了解到PID：进程的ID

USER：进程所有者

PR：进程的优先级别，越小越优先被执行

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——休眠，R——正在运行，Z——僵死状态，N——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：进程启动命令名称

1. **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。**

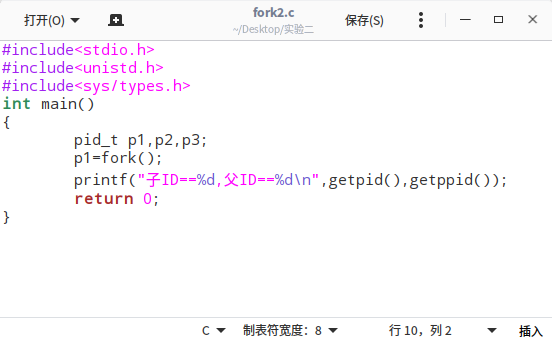


图3.1 创建进程树

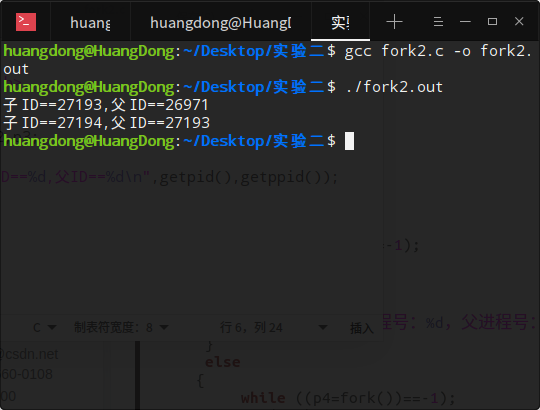


图3.2 执行进程树程序

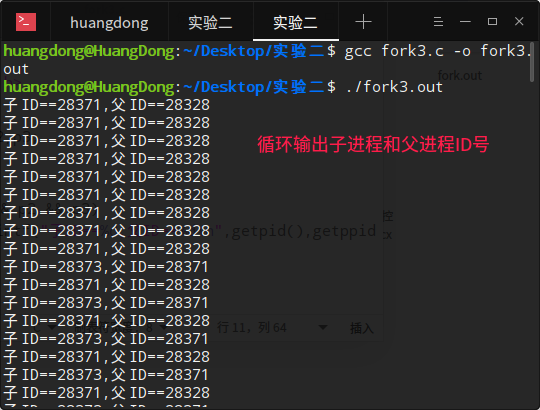


图3.3 执行结果

**实验总结：**开始没有没有灵活学习本次实验所需代码,导致无法编写出5个进程的程序实现本实验.后续参考学习后处理下列代码,进行该次实验的进程的执行顺序和进程状态判断.

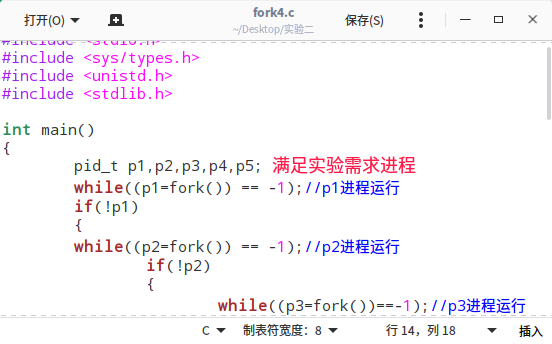


图3.4 创建五个进程

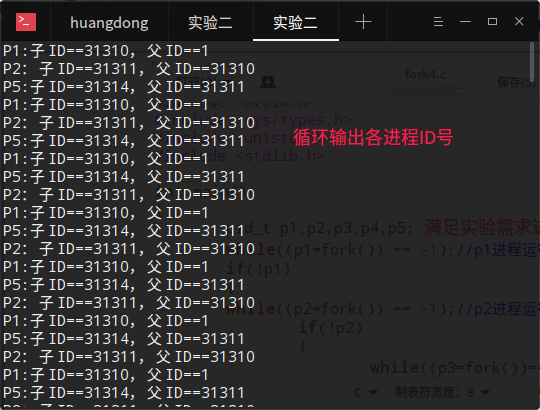


图3.5 执行结果

1. **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。**



图4.1 进行kill终止

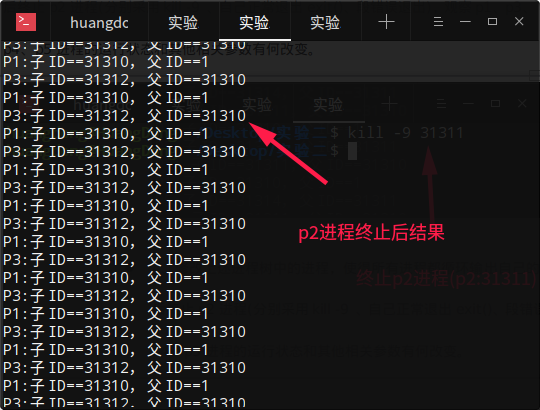


图4.2 kill终止p2进程



图4.3 p2终止语句

1. **总结反思：**



本次实验初始阶段没有充分了解系统中进程的控制，导致在实验时没有按照要求更改编写函数代码完成所需的实验结果，后面学习参考其他同学的学习结果，明白了自己需要重新掌握的知识和进程控制关系。也学习到实验前进行基础知识学习的重要性，不应该只是简单的进行浏览后就开始实验步骤。

下面是本次实验博客地址和GitHub文件地址：