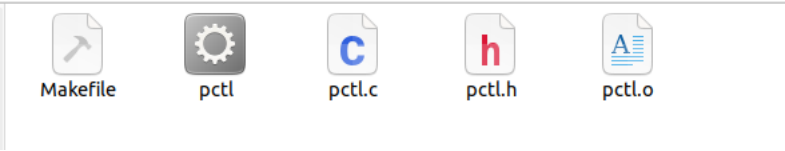
山东大学 计算机科学与技术 学院
操作系统 课程实验报告

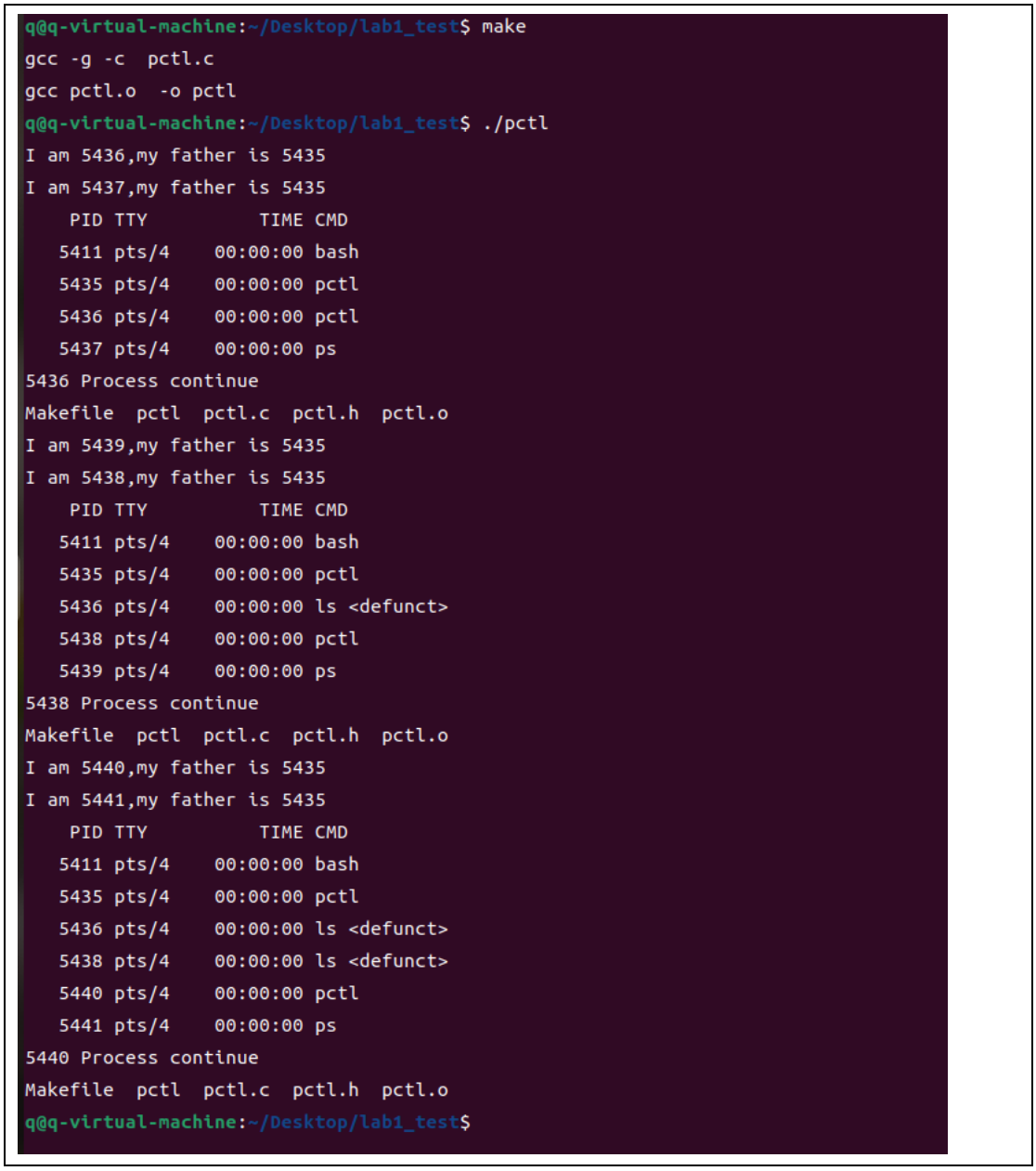
学号：202200130223	姓名：李洋永	班级：4 班
实验题目：实验 1：进程控制		
实验学时：2	实验日期：2024/10/9	
实验目的：加深对于进程并发执行概念的理解。 3 线程与管道通信 4 进程综合实验 实践并发进/线程的创建和控制方法。 观察和体验进程的动态特性。 进一步理解进程生命期期间创建、变换、撤销状态变换的过程。 掌握进程控制的方法，了解父子进程间的控制和协作关系。 练习 Linux 中有关的系统调用的编程和调试技术		
实验环境：Ubuntu		
源程序清单： 		
编译及运行结果：

```
    pid1=fork();
    if(pid1<0)
    {
            printf("Create Process fail!\n");
            exit(EXIT_FAILURE);
    }
    else if(pid1==0)
    {
            printf("I am %d,my father is %d\n",getpid(),getppid());
            pause();
            execlp("/bin/ls", "ls", NULL);//pid1 use ls
    }
    else//parent
    {
            pid2=fork();//build pid2 in parent
            if(pid2<0)
            {
                    printf("Create Process fail!\n");
                    exit(EXIT_FAILURE);
            }
            else if(pid2==0)
            {
                    printf("I am %d,my father is %d\n",getpid(),getppid());
                    status = execlp("/bin/ps", "ps", NULL);//pid2 use ps

            }
            else//wait pid2 finish , then finish pid1 ,so ps is before ls
            {
                    waitpid(pid2,&status,0);
                    kill(pid1, SIGINT);
            }

    }
```

在 pid1 的父进程执行代码段上建立 pid2，pid1 需要执行 ls，pid2 执行 ps，使用 pause 阻塞 pid1，在 pid2 的父进程执行代码段上等待 pid2 执行完成后再给 pid1 传递信号，即可实现 ps 先于 ls。 | | |


q@q-virtual-machine:~/Desktop/lab1_test$ make
gcc -g -c pctl.c
gcc pctl.o -o pctl
q@q-virtual-machine:~/Desktop/lab1_test$ ./pctl
I am 5436,my father is 5435
I am 5437,my father is 5435
PID TTY TIME CMD
5411 pts/4 00:00:00 bash
5435 pts/4 00:00:00 pctl
5436 pts/4 00:00:00 pctl
5437 pts/4 00:00:00 ps
5436 Process continue
Makefile pctl pctl.c pctl.h pctl.o
I am 5439,my father is 5435
I am 5438,my father is 5435
PID TTY TIME CMD
5411 pts/4 00:00:00 bash
5435 pts/4 00:00:00 pctl
5436 pts/4 00:00:00 ls <defunct>
5438 pts/4 00:00:00 pctl
5439 pts/4 00:00:00 ps
5438 Process continue
Makefile pctl pctl.c pctl.h pctl.o
I am 5440,my father is 5435
I am 5441,my father is 5435
PID TTY TIME CMD
5411 pts/4 00:00:00 bash
5435 pts/4 00:00:00 pctl
5436 pts/4 00:00:00 ls <defunct>
5438 pts/4 00:00:00 ls <defunct>
5440 pts/4 00:00:00 pctl
5441 pts/4 00:00:00 ps
5440 Process continue
Makefile pctl pctl.c pctl.h pctl.o
q@q-virtual-machine:~/Desktop/lab1_test$

问题及收获:

```
54              printf("\nMy child exit! status = %d\n\n",status);
55          }
56          else{
57  //如果在命令行上没输入子进程要执行的命令
58  //唤醒子进程，与子进程并发执行不等待子进程执行结束
59              //sleep(5); //思考：如果去掉这条语句，可能会出现什么现象
60              if(kill(pid,SIGINT) >= 0)
61                  printf("%d Wakeup %d child.\n",getpid(),pid) ;
62              printf("%d don't Wait for child done.\n\n",getpid());
63          }
64      }
65      return EXIT_SUCCESS;
66  }
67
```

如果这里不等 5 秒的话在子进程执行 ls 前父进程就结束了，那么子进程就执行不了。

```
gcc pctl.o  -o pctl
q@q-virtual-machine:~/Desktop/lab1_example$ ./pctl
SIGINT: Success

I am Parent process 5299
5299 Wakeup 5300 child.
5299 don't Wait for child done.

5300 Process continue
I am Child process 5300
My father is 5299
q@q-virtual-machine:~/Desktop/lab1_example$
```

1.5. 实验要求

根据实验中观察和记录的信息结合示例实验和独立实验程序，说明它们反映出操作系统教材中进程及处理机管理一节讲解的进程的哪些特征和功能？在真实的操作 系统中它是怎样实现和反映出教材中讲解的进程的生命期、进程的实体和进程状态 控制的。你对于进程概念和并发概念有哪些新的理解和认识？子进程是如何创建和 执行新程序的？信号的机理是什么？怎样利用信号实现进程控制？根据实验程序、调试过程和结果分析写出实验报告。

进程的特征和功能：父子进程可以并发进行，且占有各自的资源，进程间独立。
通过 fork () 创建子进程，可以在函数中指定子进程需要执行的操作，父进程可以通过 wait（）等控制子进程状态。
进程可以创建子进程并分配其任务，父子进程可以并发执行。
子进程通过调用 exec()创建执行新程序。
信号是进程间的机制，一个进程可以向另一个进程发信号，通过信号可以控制子进程的执行。