**北京邮电大学软件学院**

**2017-2018学年第2学期实验报告**

**课程名称： 操作系统**

**实验名称： 实验一 进程管理 及 实验二 进程通讯**

**实验完成人：**

**姓名：**\_\_\_\_田宇\_\_\_\_**学号：**\_2016212011\_**成绩：**\_\_\_\_\_\_\_\_

**日 期： 2018 年 3 月 23 日**

1. **实验目的**

（1）理解进程的概念，明确进程和程序的区别。

（2）理解并发执行的实质。

（3）掌握进程的睡眠、同步、撤销等进程控制方法。

1. **实验内容**

2.1. 进程的创建。

① 编写一段源程序，使系统调用fork()创建两个子进程，当此程序运行时，在系统中 有一个父进程和两个子进程活动。让每一个进程在屏幕上显示一个字符：父进程显示字符 “a”；子进程分别显示字符“b”和字符“c”。试观察记录屏幕上的显示结果，并分析原 因。

② 修改已编写的程序，将每个进程输出一个字符改为每个进程输出一句话，在观察程序执行时屏幕出现的现象，并分析原因。

2.2. 进程的控制

① 用fork()创建一个进程，再调用exec()用新的程序替换该子进程的内容。 ② 利用wait()来控制进程执行顺序。

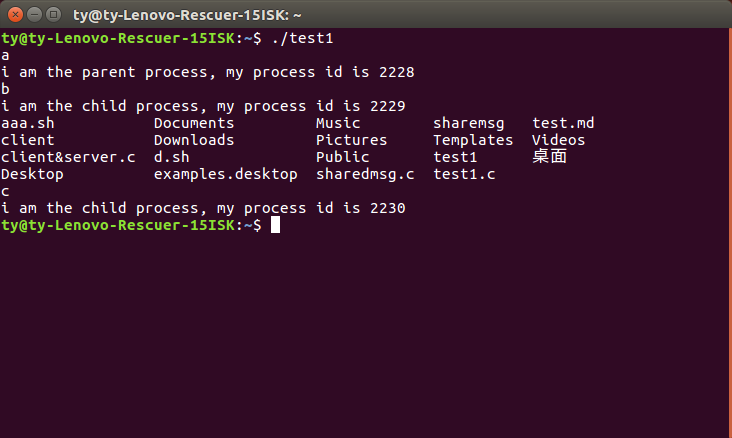
1. **实验环境**

Ubuntu 16.4

gcc

1. **实验结果**

完成



1. **附件**

**6.1 附件1：源代码**

#include <unistd.h>

#include <stdio.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("b\n");

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

execlp("/bin/ls","ls",NULL);

}

else

{

printf("a\n");

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

wait(NULL);

fpid = fork();

if (fpid < 0)

printf("error in fork!");

else if (fpid == 0)

{

printf("c\n");

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

}

}

return 0;

}