电子科技大学信息与软件工程学院

**实 验 报 告**

学 号 2017221302006

姓 名 周玉川

（实验） 课程名称 操作系统基础

理论教师 刘瑶

实验教师 文淑华

**电子科技大学教务处制表**

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：周玉川 学号：2017221302006 指导教师：刘瑶**

**实验地点：信软楼西305 实验时间：2019.05.22**

1. **实验名称：**利用管道实现两个进程的通信、

利用消息队列实现进程间的通信

**二、实验学时：8**

**三、实验目的：**

1. 熟悉Linux下的应用程序开发
2. 熟悉Linux的进程控制原语的使用
3. 掌握Linux操作系统的进程间通信机制管道的使用。
4. 掌握Linux操作系统中父进程与子进程的同步。

**四、实验原理：**

（一）：

首先创建两个子进程，注意Linux下使用fork()函数创建进程的方法。父进程和两个子进程间需要同步，使用waitpid（）函数实现父进程等待子进程运行完毕后从管道中读取数据并打印，只有子进程将数据写入管道后，父进程才能够执行打开管道操作。

由于fork函数让子进程完整地拷贝了父进程的整个地址空间，所以子进程都有管道的读端和写端。所以在相关进程中最好关掉不用的那一端。根据要求，“父进程先接收子进程P1发来的消息，然后再接收子进程P2发来的消息。”存在两个同步问题，两个子进程和父进程之间（先子写后父读）同步、子进程1和子进程2之间（先1写，再2写）

（二）：

消息队列就是一个消息的链表。可以把消息看作一个记录，具有特定的格式以及特定的优先级。对消息队列有写权限的进程可以向中按照一定的规则添加新消息；对消息队列有读权限的进程则可以从消息队列中读走消息。

**五、实验内容：**

（一）：

在Linux系统中使用系统调用fork()创建两个子进程，使用系统调用pipe()建立一个管道，两个子进程分别向管道各写一句话：

Child process 1 is sending a message!

Child process 2 is sending a message!

而父进程则从管道中读出来自于两个子进程的信息，显示在屏幕上。然后分别结束两个子进程的运行。

要求：

（1）父进程先接收子进程P1发来的消息，然后再接收子进程P2发来的消息。

（2）在Linux平台下实现。

（二）：

编程实现两个进程通过消息队列进行通信，一个Server进程，一个client进程。Server进程向client进程发送其进程ID，client进程同时也向Server进程发送其进程ID，双方接收到消息后，将所接收到的进程ID输出到屏幕上。

**六、实验器材（设备、元器件）：**

1. 学生每人一台PC，安装Windows10操作系统。
2. 个人PC安装VMware虚拟机和Ubuntu系统。

**七、实验步骤：**

（一）

步骤1：创建一个管道。

步骤2：创建子进程1，向管道中写入“Child process 1 is sending a message!”，并做好跟父进程的同步执行。

步骤3：创建子进程2，向管道中写入“Child process 2 is sending a message!”，并做好跟父进程的同步执行。

步骤4：父进程从管道中读取数据，并打印输出。

（二）

使用“Ctrl+Alt+T”打开终端；

使用gedit或vim命令打开文本编辑器进行编码：“gedit 文件名.c”

编译程序：

gcc 文件名.c -o 可执行程序名 （如果只输入 gcc 文件名.c，默认可 执行程序名为a.out）

使用线程库时，gcc编译需要添加-lpthread

执行程序：./可执行程序名

**八、实验结果与分析（含重要数据结果分析或核心代码流程分析）**

（一）实验一：

#include <unistd.h>

#include <signal.h>

#include <stdio.h>

#include <stdlib.h>

#include<sys/types.h>

#include<sys/wait.h>

int pid1,pid2;

void main(void)

{

int fd[2];

char outstr[100],instr[100];

pipe(fd); //创建一个管道

while ((pid1=fork())==-1);//当创建不成功时，循环等待，直到创建成功为止

if(pid1==0){

//创建成功后，执行写入操作，fork()1先写

lockf(fd[1],1,0);

sprintf(outstr,"child 1 process is sending message!");

//把串放入数组outstr中

write(fd[1],outstr,50); //向管道写入长度为50字节的字符串

sleep(5); //阻塞5毫秒

lockf(fd[1],0,0);

exit(0);

}

else{

while((pid2=fork())==-1);

if(pid2==0){ //fork()2先写

lockf(fd[1],1,0);

sprintf(outstr,"child 2 process is sending message!");

write(fd[1],outstr,50);

sleep(5);

lockf(fd[1],0,0);

exit(0);

}else{

waitpid(pid1,NULL,0); //父子进程同步

read(fd[0],instr,50);//从管道中读长为50字节的串, 为fork1写入的

printf("%s\n",instr);

waitpid(pid2,NULL,0);

read(fd[0],instr,50);

printf("%s\n",instr);

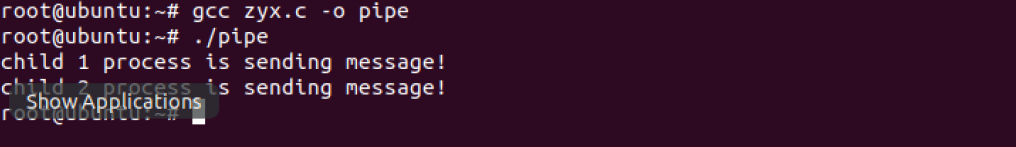
exit(0);

}

}

}

运行截图：



**九、总结及心得体会：**

（1）本次实验我学会了通过管道，来进程通信的方法，使用write和read操作可以分别向管道写入和读出数据。

（2）为了使得父进程在读出数据时有数据，我们必须使用waitpid()函数来等待子进程运行完毕，waitpid和wait是进程同步的一种方法。

（3）为了防止多个进程同时读写管道，在操作这个临界资源之前必须给他加上互斥锁，操作完以后再解锁，可见锁机制十分重要。

**十、对本实验过程及方法、手段的改进建议：**

1. 熟能生巧，一定要多实践，做起实验来才能得心应手。
2. 遇到不会的点或者不熟悉的点，先查书找不到答案再百度。
3. 实验课可以很好的检验平时有没有认真学，而且能纠正对知识理解上的错误。

**报告评分：**

**指导教师签字：**