

实验四 进程间通信

一、 实验类型

本实验为设计性实验

二、 实验目的与任务

了解 Linux 系统中的进程管道通信的基本原理，并能编写简单的管道通信的程序。

三、 预习要求

- 1) 熟悉进程通信的概念
- 2) 熟悉管道通信的基本原理
- 3) 用到的系统调用有 pipe()

四、 实验基本原理

实验利用 Linux 操作系统提供的系统调用 pipe()来实现两个进程之间的通信。输入进程从管道的一端输入数据，另一个进程从管道的另一端读数据，通过这种方式来实现进程间的通信。

五、 实验仪器与设备（或工具软件）

实验设备：计算机一台，软件环境要求：安装 Red Hat Linux 操作系统和 gcc 编译器。

六、 实验内容

编制一段程序，实现进程管道通信。使用系统调用 pipe()建立一条管道。两个子进程 P1 和 P2 分别向管道各写一句话：

Child process 1 is sending message!

Child process 2 is sending message!

而父进程则从管道中读出来自两个子进程的信息，显示在屏幕上。

参考程序程序如下

```
#include<unistd.h>
#include<signal.h>
#include<stdio.h>
int pid1,pid2;
main(){
int fd[2];
```

```

char OutPipe[100],InPipe[100];
pipe(fd);
while((pid1=fork())!=-1);
if(pid1==0){
lockf(fd[1],1,0);
sprintf(OutPipe,"\n Child process 1 is sending message!\n");
write(fd[1],OutPipe,50);
sleep(5);
lockf(fd[1],0,0);
exit(0);
}
else{
while((pid2=fork())!=-1);
if(pid2==0){
lockf(fd[1],1,0);
sprintf(OutPipe,"\n Child process 2 is sending message!\n");
write(fd[1],OutPipe,50);
sleep(5);
lockf(fd[1],0,0);
exit(0);
}
else{
wait(0);
read(fd[0],InPipe,50);
printf("%s\n",InPipe);
wait(0);
read(fd[0],InPipe,50);
printf("%s\n",InPipe);
exit(0);
}
}
}

```

七、 实验步骤

- 1) 进入 vi 编辑器
- 2) 在编译器中输入所要运行的程序代码

3) 退出编辑器, 返回命令行输入方式, 使用 `gcc` 编译器编译程序, 获得能运行的目标程序。

4) 运行目标程序, 查看运行结果。

八、 注意事项

- 1) 管道按怎样的方式传送消息? 是否能够双向传送消息?
- 2) 在管道通信方式中, 如何来实现进程同步与互斥的?
- 3) `pipe()`函数的使用可参看第一部分的介绍。

九、 实验报告要求

需要列出运行了的程序清单及相应结果, 并对结果进行分析和讨论。对结果的分析主要讨论管道通信方式的特点, 实验结果是否体现该特点。