实验四 进程间通信——有名管道

**一、实验目的**

1、了解有名管道通信的原理。

2、掌握有名管道的创建及使用方法。

**二、实验原理**

1、有名管道可用于无亲缘关系的进程间的通信，又称为FIFO。

2、有名管道使用mkfifo()创建。

3、管道打开规则：

对于读进程：

（1）阻塞打开：阻塞直到有相应进程为写而打开该FIFO。

（2）非阻塞打开：立刻返回成功。

对于写进程：

（1）阻塞打开：阻塞直到有相应进程为读而打开该FIFO。

（2）非阻塞打开：立刻返回失败，错误码为ENXIO。

**三、实验内容**

1、使用mkfifo函数创建一个有名管道。以阻塞方式打开管道，将磁盘文件test1的内容都读取到管道中。

2、以阻塞方式打开管道，将管道中的数据写入文件test2，从而在test1与test2之间完成拷贝文件的功能。

3、编译并分别在两个终端里运行读端和写端程序，查看结果。

四. 代码

Write\_fifo.c

#include<stdio.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<unistd.h>

#include<fcntl.h>

#include<assert.h>

int main(int argc, char \*argv[]){

const char \*fifo = "/home/lambdafate/fifo";

if(access(fifo, F\_OK) == -1){

assert(mkfifo(fifo, 0777) != -1);

}

int test1\_fd = open("./test1", O\_RDONLY);

assert(test1\_fd != -1);

int fifo\_fd = open(fifo, O\_WRONLY);

assert(fifo\_fd);

char buffer[128];

int bytes = 0;

while((bytes = read(test1\_fd, buffer, 128)) > 0){

write(fifo\_fd, buffer, bytes);

}

return 0;

}

Read\_fifo.c

#include<stdio.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<unistd.h>

#include<fcntl.h>

#include<assert.h>

int main(int argc, char \*argv[]){

const char \*fifo = "/home/lambdafate/fifo";

if(access(fifo, F\_OK) == -1){

assert(mkfifo(fifo, 0777) != -1);

}

int test2\_fd = open("./test2", O\_CREAT | O\_WRONLY, 0777);

assert(test2\_fd != -1);

int fifo\_fd = open(fifo, O\_RDONLY);

assert(fifo\_fd != -1);

char buffer[128];

int bytes = 0;

while ((bytes = read(fifo\_fd, buffer, 128)) > 0){

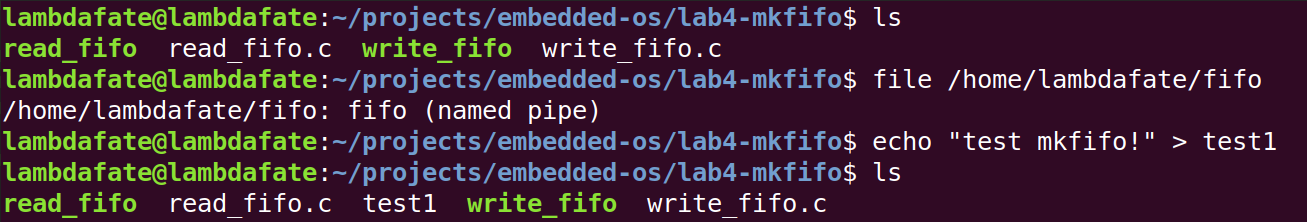
write(test2\_fd, buffer, bytes);

}

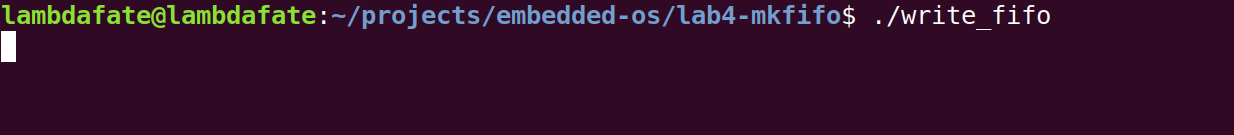
return 0;

}

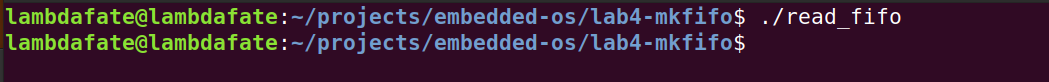
五，实验过程



先在一个终端中执行中 write\_fifo.c，将磁盘文件test1的内容都读取到管道中。



在另一个终端中 执行 read\_fifo.c ， 将管道中的数据写入文件test2



查看 test2中的内容， 和test1一样

