学号:031402313姓名:黄志明学院:数计学院 专业:计算机 班级:3班

《Linux 操作系统设计实践》实验四: 文件操作

实验环境: Ubuntu Kylin 14.04 LTS

实验内容:......(写出程序代码和运行结果截图，程序代码上需有注释)

1. /\*send.c\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <errno.h>

#define MSGKEY 1024

struct msgstru

{

long msgtype;

char msgtext[2048];

};

int main()

{

struct msgstru msgs;

int msg\_type;

char str[256];

int ret\_value;

int msqid;

FILE \*fin,\*fout;

fin=fopen("/home/cafe3165/input.txt","rb");

msqid=msgget(MSGKEY,IPC\_EXCL); /\*检查消息队列是否存在\*/

if(msqid < 0){

msqid = msgget(MSGKEY,IPC\_CREAT|0666);/\*创建消息队列\*/

if(msqid <0){

//printf("failed to create msq | errno=%d [%s]\n",errno,strerror(errno));

exit(-1);

}

}

while (1){

// printf("input message type(end:0):");

fscanf(fin,"%d\n",&msg\_type);

if (msg\_type == 0)

break;

//printf("input message to be sent:");

fscanf (fin,"%s",str);

msgs.msgtype = msg\_type;

strcpy(msgs.msgtext, str);

/\* 发送消息队列 \*/

ret\_value = msgsnd(msqid,&msgs,sizeof(struct msgstru),IPC\_NOWAIT);

if ( ret\_value < 0 ) {

// printf("msgsnd() write msg failed,errno=%d[%s]\n",errno,strerror(errno));

exit(-1);

}

}

msgctl(msqid,IPC\_RMID,0); //删除消息队列

return 0;

}

2. /\*receive.c \*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <errno.h>

#define MSGKEY 1024

struct msgstru

{

long msgtype;

char msgtext[2048];

};

/\*子进程，监听消息队列\*/

void childproc(FILE \*fout){

struct msgstru msgs;

int msgid,ret\_value;

char str[512];

while(1){

msgid = msgget(MSGKEY,IPC\_EXCL );/\*检查消息队列是否存在 \*/

if(msgid < 0){

printf("msq not existed! errno=%d [%s]\n",errno,strerror(errno));

sleep(2);

continue;

}

/\*接收消息队列\*/

ret\_value = msgrcv(msgid,&msgs,sizeof(struct msgstru),0,0);

printf("text=[%s] pid=[%d]\n",msgs.msgtext,getpid());

fprintf(fout,"text=[%s] pid=[%d]\n",msgs.msgtext,getpid());

break;

}

return;

}

void main()

{

int i,cpid;

FILE \*fin,\*fout;

fout=fopen("/home/cafe3165/out.txt","wb");

/\* create 5 child process \*/

for (i=0;i<5;i++){

cpid = fork();

if (cpid < 0)

printf("fork failed\n");

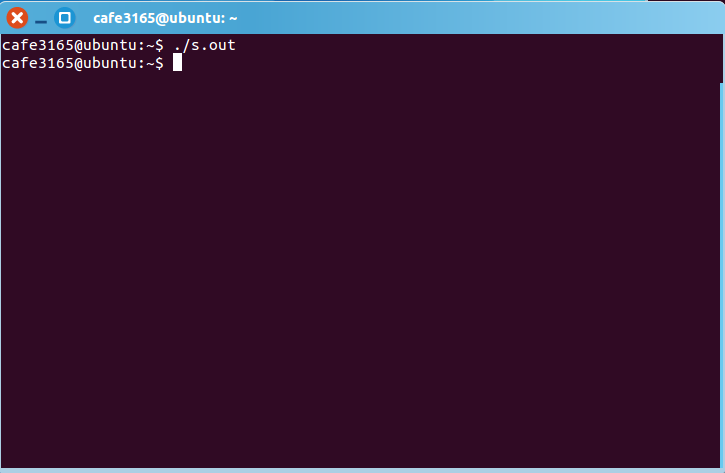
else if (cpid ==0) /\*child process\*/

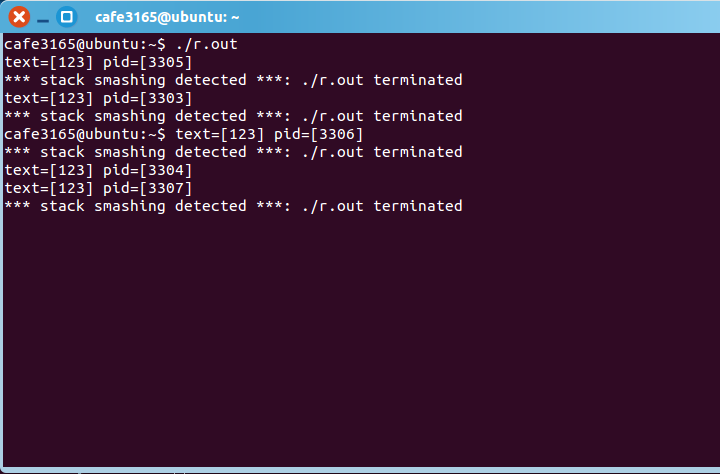
childproc(fout);

}

}









实验总结:......(解释说明所编写的程序思想;如果是基于示例程序的改进需说明 改进工作在哪里。)

基于实验二的进程通信，即通过input.txt.文件输入类型以及所要输入的 文字信息，传到消息队列，并在out.txt中打印出来，利用C中的fopen(),FILE \*指针可以轻松实现文件的读写