姓名：吴涵 学号：202101000720 成绩：

# 实验32消息传递通信

1、消息传递通信

源程序：22

（1）写程序msgwrite.c：

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <stdio.h>

#include <stdlib.h>

typedef struct \_msg\_buf

{                  // 消息结构体

    long type;     // 消息类型

    char buf[100]; // 消息内容

} msg\_buf;

int main()

{

    int key, qid;

    int wc = 1;

    msg\_buf buf;

    key = ftok(".", 10);

    qid = msgget(key, IPC\_CREAT | 0666);

    printf("key: %d\nqid: %d\n", key, qid);

    buf.type = 10;

    system("ipcs -q");

    printf("请输入一些消息，每条消息以回车结束。如果输入quit，则程序结束\n");

    while (1){ // 循环输入消息到队列

        printf("输入第%d条消息：", wc++);

        fgets(buf.buf, 100, stdin); // 从键盘输入不超过100个字符的消息到消息缓冲区

        if (strncmp(buf.buf, "quit", 4) == 0)

        {                                          // 如果输入"quit"，则删除消息队列并结束程序

            if ((msgctl(qid, IPC\_RMID, NULL)) < 0) /\*删除指定的消息队列\*/

            {

                exit(1);

            }

            else

            {

                system("ipcs -q");

                printf("successfully removed %d queue/n", qid); /\* 删除队列成功 \*/

                exit(0);

            }

        }

        if (msgsnd(qid, (void \*)&buf, 100, 0) < 0)

        { // 发送消息缓冲区中的信息到消息队列

            perror("msgsnd");

            exit(-1);

        }

    }

    return 0;

}

（2）读程序msgread.c：

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <stdio.h>

#include <stdlib.h>

typedef struct \_msg\_buf

{//消息结构体

long type; //消息类型

char buf[100]; //消息内容

} msg\_buf;

int main()

{

int key, qid; //消息队列键值和标识符

msg\_buf buf; //消息缓冲区

key = ftok(".", 10); //将当前目录和10转换为消息队列IPC键值

qid = msgget(key, IPC\_CREAT|0666); //创建或获得消息队列标识符

printf("key: %d\nqid: %d\n", key, qid); //输出消息队列键值和标识符

while (1)

{

if (msgrcv(qid, (void \*)&buf, 100, 0, 0) < 0)

{//循环读取队列中消息到缓冲区

perror("msgrcv");

exit(-1);

}

printf("type:%ld\nget:%s\n", buf.type, buf.buf); //输出所读信息

}

return 0;

}

编译链接命令：

gcc msgwrite.c -o msgwrite

gcc msgread.c -o msgread

运行命令：

./msgwrite

./msgread

交互与结果：

