

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
{
    long type;    // 消息结构体
    char buf[100]; // 消息类型
} 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
            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

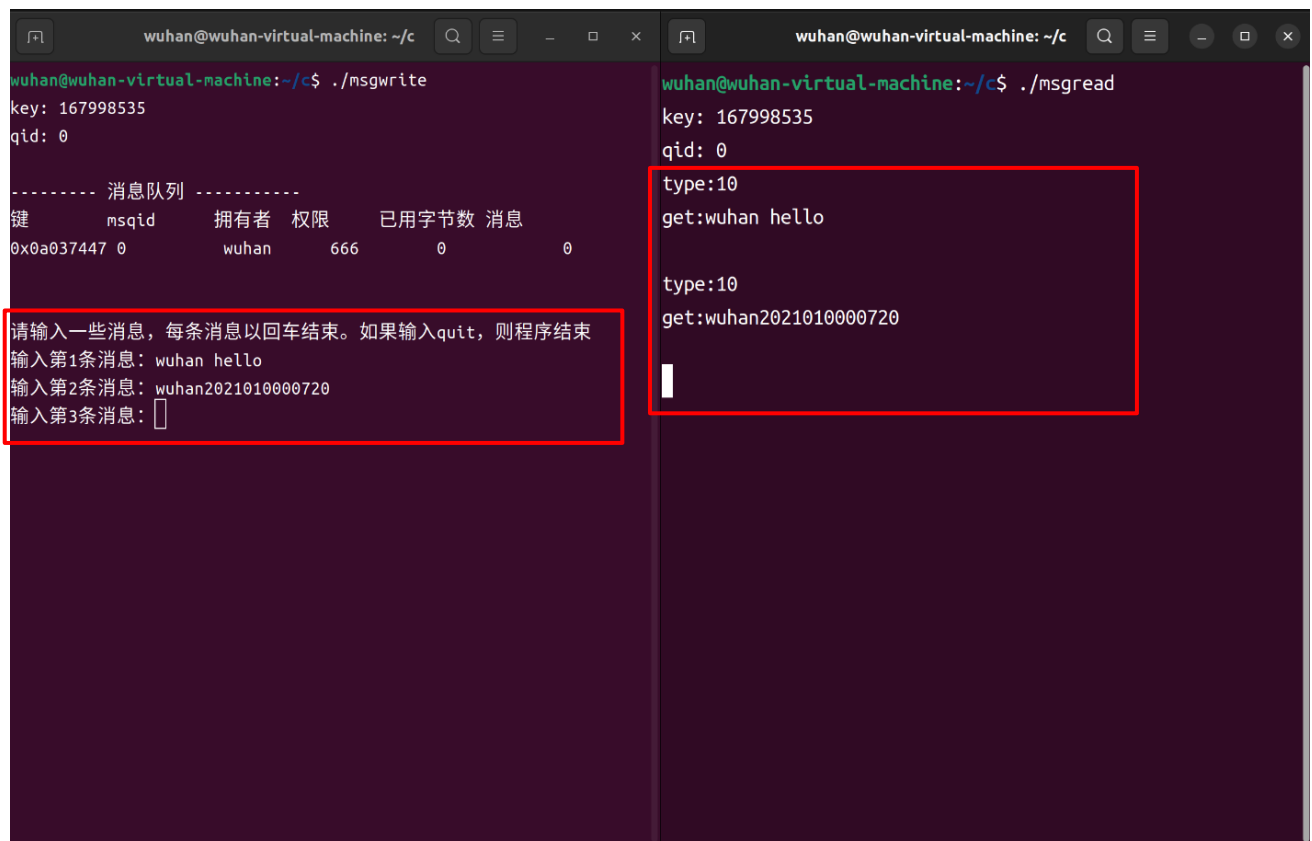
gcc msgread.c -o msgread

运行命令:

./msgwrite

./msgread

交互与结果:



The image shows two terminal windows side-by-side, both with the title bar 'wuhan@wuhan-virtual-machine: ~/c'. The left window is running the './msgwrite' command. It displays the key '167998535' and qid '0'. Below this, it shows a table header for a message queue and a table with one row of data. The table header is: '----- 消息队列 -----' followed by columns '键', 'msgid', '拥有者', '权限', '已用字节数', and '消息'. The table row is: '0x0a037447 0', 'wuhan', '666', '0', and '0'. Below the table, there is a prompt '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' and three lines of input: '输入第1条消息: wuhan hello', '输入第2条消息: wuhan2021010000720', and '输入第3条消息: ' (with an empty box). The right window is running the './msgread' command. It displays the same key '167998535' and qid '0'. Below this, it shows two lines of output: 'type:10' followed by 'get:wuhan hello', and 'type:10' followed by 'get:wuhan2021010000720'. There is a cursor on the line following the second output.

```
wuhan@wuhan-virtual-machine: ~/c$ ./msgwrite
key: 167998535
qid: 0

----- 消息队列 -----
键      msgid    拥有者  权限  已用字节数  消息
0x0a037447 0      wuhan   666    0          0

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
输入第1条消息: wuhan hello
输入第2条消息: wuhan2021010000720
输入第3条消息: 
```

```
wuhan@wuhan-virtual-machine: ~/c$ ./msgread
key: 167998535
qid: 0

type:10
get:wuhan hello

type:10
get:wuhan2021010000720

```