

五级项目 7：消息队列实现进程间通信

1 学时

- 1 学时

2 实验目的

- 理解、掌握、应用 Linux 下使用消息队列完成进程间的方法。

3 实验内容

- 本项目编写一个创建消息队列的程序，假设当前目录下有一个名为“key.msg”的文件，现在利用该文件名调用 ftok 生成消息队列资源的 key 值。编写一个向消息队列发送消息的程序，该程序在执行时需要一个命令行参数，它是要发送的消息正文。再编写一个从消息队列接收消息的程序，该程序也通过相同的文件名“key.msg”得到相同的 key 值，打开上一个程序创建的消息队列资源。最后删除消息队列。

4 实验原理

操作系统提供消息队列，有权限的进程可以把消息（message）发送到消息队列（message queue）中。消息一直存放在消息队列中直到另一个进程将其取走。消息由两部分组成：

- （1）固定大小的消息首部。
- （2）可变长度的消息正文。

消息队列的主要系统调用如表。

表 消息队列相关系统调用

名称	说明
msgget	创建、打开消息队列
msgsnd	发送消息
msgrcv	接收消息
msgctl	操作消息队列

msgget 用于创建或打开消息队列。key 可以指定为调用 ftok 生成的值，也可以指定为 IPC_PRIVATE。当 msgflg 设置为 IPC_CREAT 或 IPC_CREAT | IPC_EXCL 时，msgget 创建消息队列；如果要打开一个存在的消息队列，msgflg 指定为访问消息队列的权限。

msgget 返回消息队列的标识符。另外 3 个系统调用都需要这个标识符来访问消息队列。

表 系统调用 msgget

项目	描述
头文件	#include <sys/types.h> #include <sys/ipc.h>

	#include <sys/msg.h>
原型	int msgget(key_t key, int msgflg);
功能	创建、打开消息队列
参数	key: 消息队列键值 msgflg: 创建方式
返回值	成功返回消息队列 ID; 失败返回-1

msgsnd 用于发送消息。msqid 为目标消息队列的标识符。msgp 是指向一个类似 msgbuf 结构的指针。msgsz 指定 mtext 成员的最大字节数。msgflg 是控制函数行为的标志，若为 0 表示忽略标志位；若为 IPC_NOWAIT，如果消息队列已满，消息将不被写入队列，控制权返回调用函数的线程。如果不指定这个参数，线程将被阻塞直到消息可以被写入。

<pre> struct msgbuf { long mtype;//消息的类型 char mtext[1];//消息正文 }; </pre>

表 系统调用 msgsnd

项目	描述
头文件	#include <sys/types.h> #include <sys/ipc.h> #include <sys/msg.h>
原型	int msgsnd(int msqid, const void *msgp, size_t msgsz, int msgflg)
功能	发送消息
参数	msqid: 消息队列 ID msgp: 指向要发送消息的首地址 msgsz: 消息长度 msgflg: 发送方式
返回值	成功返回消息队列 ID; 失败返回-1

msgrcv 用于从消息队列接收消息。msqid 为目标消息队列的标识符，msgsz 指定 mtext 成员的最大字节数，msgtyp 指定要接收的消息的类型，msgflg 可以是表中三个值的按位或运算的结果。

表 系统调用 msgrcv

项目	描述
头文件	#include <sys/types.h> #include <sys/ipc.h> #include <sys/msg.h>
原型	ssize_t msgrcv(int msqid, void *msgp, size_t msgsz, long msgtyp, int msgflg);
功能	接收消息
参数	msqid: 消息队列 ID msgp: 指向保存接收消息首地址

	msgsz: 消息长度 msgtyp: 要接收消息的类型 msgflg: 接收方式
返回值	成功返回消息正文字节数；失败返回-1

表 msgflg 参数取值

项目	说明
IPC_NOWAIT	如果消息队列中没有请求类型的消息，则立刻返回。系统会设置错误码（errno）为 ENOMSG。
MSG_EXCEPT	此参数与 msgtyp（大于 0）一起使用，读出消息队列中非 msgtyp 类型的第一条消息
MSG_NOERROR	如果消息的长度大于 msgsz，则截短消息

msgctl 用于操作消息队列，参数 cmd 的取值如表所示。

表 系统调用 msgctl

项目	描述
头文件	#include <sys/types.h> #include <sys/ipc.h> #include <sys/msg.h>
原型	int msgctl(int msqid, int cmd, struct msqid_ds *buf);
功能	消息队列控制函数
参数	msqid: 消息队列 ID cmd: 控制命令
返回值	成功返回 0；失败返回-1

表 cmd 参数取值

项目	说明
IPC_STAT	获取 msqid 消息队列在内核中的信息，存储在 buf 指向的 msqid_ds 对象。
IPC_SET	设置新的消息队列在内核中的信息。新的信息由 buf 指向的 msqid_ds 对象给出。
IPC_RMID	删除 msqid 消息队列。

5 预习要求和技术准备工作

- 掌握 Linux 基本操作
- 掌握 C 语言开发工具的使用
- 掌握消息队列实现进程间通信的方法

6 实验环境

- PC 机
- 在 Windows 环境中的 VMware 虚拟机上运行 Ubuntu 操作系统或者独立的 Ubuntu 操作系统
- 基于 Linux 的 vi 编辑器和 gcc 编译器

7 实验设计及操作步骤

7.1 以 root 身份登录系统，在/home 目录中创建目录 exp57

```
cd /home
```

```
mkdir exp57
```

7.2 进入刚创建的目录

```
cd exp57
```

7.3 使用 vi 编辑文件，文件名分别是 msgcreate.c，msgsendtest.c，msgrecvtest.c，msgdel.c。

7.4 编写程序，实现要求的功能

★编程思路

(1) mysmg.h

定义消息结构体。

(2) msgcreate.c

```
main(int argc,char *argv[])
```

```
{
```

① 通过 ftok 和 msgget 创建并得到消息队列 id

② 创建成功，输入消息队列 id

```
}
```

(3) msgsendtest.c

```
main(int argc,char *argv[])
```

```
{
```

① 通过 ftok 和 msgget 得到消息队列 id（不是创建）

② 定义消息结构体变量，赋值消息类型和消息正文，注意此处消息类型应是 long 类型。

③ 调用 msgsnd 将消息发送，判断成功后输出消息队列 id 和消息正文

```
}
```

(4) msgrecvtest.c

```
main(int argc,char *argv[])
```

```
{
```

① 通过 ftok 和 msgget 得到消息队列 id（不是创建）

- ② 输出消息队列 id
- ③ 调用 `msgrcv` 并按消息类型获取消息，注意此处消息类型应是 `long` 类型。
- ④ 获取成功后显示消息类型和消息正文。

```
}
```

(5) `msgdel.c`

```
main(int argc,char *argv[])
```

```
{
```

- ① 通过 `ftok` 和 `msgget` 得到消息队列 id（不是创建）
- ② 调用 `msgctl` 删除消息队列，删除成功后提示被删除的消息队列 id

```
}
```

7.5 编译可执行文件。

```
gcc -o msgcreate msgcreate.c
gcc -o msgsendtest msgsendtest.c
gcc -o msgrcvtest msgrcvtest.c
gcc -o msgdel msgdel.c
```

7.6 运行程序

```
./msgcreate
./msgsendtest abc
./msgrcvtest
./msgdel
```

参考示例如下：

```
cst@ubuntu:~/linuxsys2021/57/teacher/msg$ ./msgcreate
msg 9 created!
cst@ubuntu:~/linuxsys2021/57/teacher/msg$ ./msgsendtest abc
you send a message"abc" to msg 9
cst@ubuntu:~/linuxsys2021/57/teacher/msg$ ./msgrcvtest
receive form msg 9:
2 abc
cst@ubuntu:~/linuxsys2021/57/teacher/msg$ ./msgdel
msg 9 deleted!
```

可以在发送消息接收后，使用 `ipcs` 查看消息队列信息

```
cst@ubuntu:~/linuxsys2021/57/teacher/msg$ ipcs
```

消息队列					
键	msgqid	拥有者	权限	已用字节数	消息
0x01050004	9	cst	644	4	1

8 实验报告提交要求:

将实验操作每个步骤中的命令、源程序以及截图写入实验报告，实验报告命名为“学号姓名-实验 57.doc”，交给指定人员。

9 项目思考

调研并思考消息队列机制在内存中使用什么数据结构实现。