消息队列实验

**实验目的**：

了解和熟悉linux支持的消息通信机制。

**实验任务：**

使用linux系统提供的系统调用msgget(),msgrev(),msgctl()编制一个长度为1K的消息发送和接受的程序。

**实验要求：**

1. 用一个程序作为“引子”，先后fork()两个子进程，父进程执行服务端SERVER程序，两个子进程分别执行客户端CLIENT1程序和客户端CLIENT2程序，进行通信。
2. 由SERVER端创建一个Key为75的消息队列，等待CLIENT1端进程和CLIENT2端进程发来的消息。当收到CLIENT1端消息编号为1的消息，以及收到CLIENT2端消息编号为1的消息后，作为结束信号，删除该队列，并结束SERVER。其中，SERVER每接受到一个消息后显示一句“(SERVER *SERVER端pid*) received message *消息的编号* from CLIENT *CLIENT端pid*”,然后发送一个返回消息给对应CLIENT端，显示一句“(SERVER *SERVER端pid*) sent return message *消息的编号* of CLIENT *CLIENT端pid*”。
3. LIENT1端使用key为75的消息队列，先后发送消息编号从10到1的消息，然后退出。CLIENT1每发送一条消息后显示一句“(CLIENT1 *CLIENT1端pid*) sent message *消息的编号*”,然后等待接受SERVER端返回给自己的消息后，显示一句“(CLIENT1 *CLIENT1端pid*) received return message *消息的编号* from *SERVER端pid*”，再发送下一条消息。
4. CLIENT2端使用key为75的消息队列，先后发送消息编号从10到1的消息，然后退出。CLIENT2每发送一条消息后显示一句“(CLIENT2 *CLIENT2端pid*) sent message *消息的编号*”,然后等待接受SERVER端返回给自己的消息后，显示一句“(CLIENT2 *CLIENT2端pid*) received return message *消息的编号* from *SERVER端pid*”，再发送下一条消息。

**注意：**上述红色斜体部分表示变量

**思考**：

1. 如何保证先由SERVER端创建一个Key为75的消息队列后，CLIENT端才能使用该消息队列；
2. 请设计消息结构struct msgform，每个消息至少包含消息类型、消息编号、进程pid;
3. 请设计消息类型，保证：服务端可以接收所有客户端发给服务端的消息；每个客户端只能接收服务端发给自己的返回消息，而不会接收发给其他客户端的消息；服务端和客户端都不能接收自己发出去的消息。

**参考程序：**

1. 消息队列通信（[客户端程序](例子/msg_client.c)，[服务端程序](例子/msg_server.c)）；
2. [管道通信](例子/pipe.c)；
3. [共享内存通信](例子/shm.c)。

**提示：**

在linux中，打开gedit编辑器（在命令方式直接敲入gedit即可；将写好的程序输入并保存（假设保存为a.c）。然后在命令状态下敲入命令“gcc a.c –o a.exe”即生成一执行文件；你敲入“./a.exe”即可运行该执行文件了。

**提供的相关资料：**

消息队列的相关函数；管道通信的相关函数；共享内存的相关函数；其他常用的一些函数。

**消息队列相关命令：**

ipcs ：显示所有的消息队列；

ipcrm：删除一个或更多的消息队列、信号量集或者共享内存标识。

**上交要求：**

* 上交一份实验报告和源程序，实验报告包括：程序代码、运行结果、实验总结。
* 实验报告命名为：学号\_姓名\_msg.docx，源程序命名为：学号\_姓名\_msg.c或.cpp，将源程序和实验报告打包为：学号\_姓名\_msg.rar，然后将打包文件发送到邮箱：os\_experiment@126.com
* 上交截止时间：10月10日24：00