

实验三：进程间通信

实验简介：此次实验主要是了解消息通信以及共享存储区通信。

实验目标：

- (1) 了解消息通信。
- (2) 了解共享存储区通信。

实验内容：

- (1) 消息的创建，发送和接收

使用系统调用 `msgget()`，`msgsnd()`，`msgrcv()` 及 `msgctl()` 编制一长度为 1K 的消息发送和接收的程序。

- (2) 共享存储区的创建, 附接和断接

使用系统调用 `shmget()`，`shmat()`，`shmdt()`，`shmctl()` 编制一个长度为 1K 的消息发送和接收的程序。

实验所需基础：

操作系统：Linux RHEL 6.0

实验是否需要联网：否

实训步骤：

步骤一：消息通信

(1) 为了便于操作和观察结果，用一个程序为“引子”，先后 `fork()` 两个子进程，SERVER 和 CLIENT，进行通信。

(2) SERVER 端建立一个 Key 为 75 的消息队列，等待其他进程发来的消息。当遇到类型为 1 的消息，则作为结束信号，取消该队列，并退出 SERVER。SERVER 每接收到一个消息后显示一句

“(server)received”。

(3) CLIENT 端使用 Key 为 75 的消息队列，先后发送类型从 10 到 1 的消息，然后退出。最后的一个消息，既是 SERVER 端需要的结束信号。CLIENT 每发送一条消息后显示一句 “(client)sent”。

(4) 父进程在 SERVER 和 CLIENT 均退出后结束。

〈程序〉

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/msg.h>
#include <sys/ipc.h>
#define MSGKEY 75          /*定义关键词 MSGKEY*/
struct msgform             /*消息结构*/
{
    long mtype;
    char mtext[1030];       /*文本长度*/
}msg;
int msgqid,i;

void CLIENT( )
{
    int i;
    msgqid=msgget(MSGKEY,0777|IPC_CREAT);
    for(i=50;i>=1;i--)
    {
        msg.mtype=i;
        printf("(client)sent\n");
        msgsnd(msgqid,&msg,1024,0);    /*发送消息 msg 入 msgid 消
息队列*/
    }
    exit(0);
}

void SERVER( )
{
    msgqid=msgget(MSGKEY,0777|IPC_CREAT); /*由关键字获得消息队列*/
    do
    {
        msgrcv(msgqid,&msg,1030,0,0); /*从队列 msgid 接受消息 msg*/
```

```

    printf("(server)receive\n");
}while(msg.mtype!=1);          /*消息类型为 1 时，释放队列*/
    msgctl(msgqid, IPC_RMID,0);
    exit(0);
}

void main()
{
    while ((i=fork())== -1);
        if(!i) SERVER();
    while ((i=fork())== -1);
        if(!i) CLIENT();
    wait(0);
    wait(0);
}

```

步骤二：共享存储区通信

(1) 为了便于操作 和观察结果，用一个 程序为“引子”，先后 fork() 两个子进程，SERVER 和 CLIENT，进行通信。

(2) SERVER 端建立一个 KEY 为 75 的共享区,并将第一个字节置为-1. 作为数据空的标志. 等待其他进程发来的消息. 当该字节的值发生变化时,表示收到了该消息,进行处理. 然后再次把它 的值设为-1。如果遇到的值为 0,则视为结束信号，取消该队列，并退出 SERVER. SERVER 每接收到一次数据后显示”(server)received”。

(3) CLIENT 端建立一个为 75 的共享区,当共享取得第一个字节为-1 时，Server 端空闲,可发送请求。CLIENT 随即填入 9 到 0。期间等待 Server 端再次空闲. 进行完这些操作后，CLIENT 退出。CLIENT 每发送一次数据后显示”(client)sent”。

(4) 父进程在 SERVER 和 CLIENT 均退出后结束。

<程序>

```

#include<sys/types.h>
#include<sys/msg.h>
#include<sys/ipc.h>
#define SHMKEY 75 /*定义共享区关键词*/
int shmid,i;
int *addr;

void CLIENT()
{
    int i;
    shmid=shmget(SHMKEY,1024, 0777|IPC_CREAT);
    /* 获取共享区, 长度 1024, 关键词 SHMKEY */
    addr=shmat(shmid,0,0); /*共享区起始地址为 addr*/
    for(i=9;i>=0;i--)
    {
        while(*addr!= -1);
        printf("(client)sent\n"); /*打印(client)sent*/

        *addr=i; /*把 i 赋给 addr*/
    }
    exit(0);
}

void SERVER()
{
    shmid=shmget(SHMKEY,1024,0777|IPC_CREAT); /*创建共享区*/
    addr=shmat(shmid,0,0); /*共享区起始地址为
addr*/
    do
    {
        *addr=-1;
        while(*addr == -1);
        printf("(server)received\n%d",*addr); /*服务进程使用
共享区*/
    } while(*addr);
    shmctl(shmid,IPC_RMID,0);
}

void main()
{
    while ((i=fork())!=-1);
    if(!i) SERVER();
    while ((i=fork())!=-1);
    if(!i) CLIENT();
}

```

```
wait(0);  
wait(0);  
}
```