

## 五级项目 6：共享内存实现进程间通信

### 1 学时

- 1 学时

### 2 实验目的

- 理解、掌握、应用 Linux 下使用共享内存进行进程间通信的方法。

### 3 实验内容

- 本项目编写一个创建共享内存并将数据写入共享内存的程序，再编写一个从共享内存中读取数据的程序，完成两个进程通过共享内存的通信过程，最后编写一个删除共享内存的程序，删除创建的共享内存。

### 4 实验原理

共享内存是最快的一种 IPC 方式。它使得多个进程能够访问一段指定的内存区。但是需要注意的一个问题是，进程在访问共享内存时要同步。通常使用信号量实现共享内存的同步。

表 系统调用 shmget

项目	描述
头文件	#include <sys/ipc.h> #include <sys/shm.h>
原型	int shmget (key_t key, size_t size, int shmflg);
功能	创建共享内存
参数	key: 共享内存的键值 size: 共享内存大小 shmflg: 创建标志
返回值	成功返回共享内存 ID；失败返回-1

key 可以是 ftok 的返回值，也可以是 IPC\_PRIVATE。size 指定以字节为单位的内存区大小。如果创建共享内存，则 size 必须要大于 0；若访问已存在的共享内存，则 size 为 0。shmflg 可以为 IPC\_CREAT 或 IPC\_CREAT | IPC\_EXCL，同时也可以设置共享内存的访问权限。shmget 返回共享内存的标识符，这个标识符供另外 3 个函数使用。

表 系统调用 shmat 和 shmdt

项目	描述
头文件	#include <sys/ipc.h> #include <sys/shm.h>
原型	void *shmat(int shmid, const void *shmaddr, int shmflg); int shmdt(const void *shmaddr);
功能	共享内存映射和解除映射
参数	shmid: 共享内存的 ID

	shmaddr: 映射地址 shmflg: 映射此内存的读写方式, SHM_RDLONLY 只读, 否则就是读写
返回值	shmat 成功返回实际映射地址, shmdt 成功时返回 0, 失败返回-1

shmat 把内存区附加到调用进程的地址中。shmaddr 为 NULL 时, 由系统指定进程空间的地址。这是推荐的使用方法。

shmat 返回共享内存在此调用进程的地址。

shmdt 把 shamaddr 指定的共享内存从调用进程地址空间断开。

表 系统调用 shmctl

项目	描述
头文件	#include <sys/ipc.h> #include <sys/shm.h>
原型	int shmctl(int shmid, int cmd, struct shmid_ds *buf);
功能	消息队列控制函数
参数	shmid: 共享内存 ID cmd: 控制命令 buf: 指向 shmid_ds 结构体的指针
返回值	成功返回 0; 失败返回-1

shmctl 用于操作共享内存对象。cmd 的取值如表 10.10 所示。

表 cmd 参数取值

项目	说明
IPC_STAT	获取 shmid 共享内存存在内核中的信息, 存储在 buf 指向的 shmid_ds 对象。
IPC_SET	设置新的共享内存存在内核中的信息。新的信息由 buf 指向的 shmid_ds 对象给出。
IPC_RMID	删除 shmid 共享内存。

## 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
- 掌握共享内存主要函数的使用。

## 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 exp56

```
cd /home
```

```
mkdir exp56
```

7.2 进入刚创建的目录

cd exp56

7.3 使用 vi 编辑文件，文件名分别是 shmwrite.c、shmread.c 和 shmdel.c。

7.4 编写程序，实现要求的功能。

#### ★编程思路

(1) shmwrite.c

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

```
{
```

- ① 通过 ftok 和 shmget 创建并得到共享内存 id
- ② 将共享内存映射到本进程
- ③ 向共享内存中写入字符串（来自命令行参数 argv[1]）
- ④ 解除共享内存映射

```
}
```

(2) shmread.c

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

```
{
```

- ① 通过 ftok 和 shmget 得到共享内存 id（不是创建）
- ② 将共享内存映射到本进程
- ③ 从共享内存中读字符串，并显示共享内存 id 和字符串内容
- ④ 解除共享内存映射

```
}
```

(3) shmdel.c

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

```
{
```

- ① 通过 ftok 和 shmget 得到共享内存 id（不是创建）
- ② 调用 shmctl 删除共享内存
- ③ 提示被删除共享内存的 id

```
}
```

7.5 编译可执行文件。

```
gcc -o shmwrite shmwrite.c
```

```
gcc -o shmread shmread.c
```

```
gcc -o shmdel shmdel.c
```

7.6 运行程序

```
#!/shmwrite 123456
```

```
#!/shmread
```

```
#!/shmdel
```

参考示例如下：

```
cst@ubuntu:~/linuxsys2021/56/teacher/shm$ ./shmwrite 123456  
shm id=229410  
cst@ubuntu:~/linuxsys2021/56/teacher/shm$ ./shmread  
read from shm 229410 :123456  
cst@ubuntu:~/linuxsys2021/56/teacher/shm$ ./shmdel  
shm 229410 deleted
```

#### 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 56.doc”，交给指定人员。

#### 9 项目思考

调研并思考内核中实现共享内存的机制。