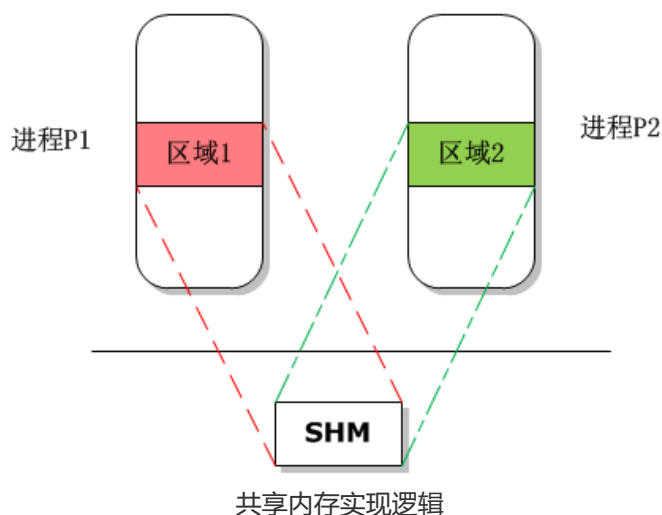


作者：曾任斯

##

1. 基本概念

共享内存，顾名思义，就是通过不同进程共享一段相同的内存来达到通信的目的，由于SHM对象不再交由内核托管，因此共享内存SHM对象是众多IPC方式最高效的一种方式，但也因为这个原因，SHM一般不能单独使用，而需要配合诸如互斥锁、信号量等协同机制使用。



2. 函数接口

使用共享内存的一般步骤是：

1. 获取共享内存对象的ID
2. 将共享内存映射至本进程虚拟内存空间的某个区域
3. 当不再使用时，解除映射关系
4. 当没有进程再需要这块共享内存时，删除它。

下面来详细介绍这些函数接口的用法。

2.1 创建或打开SHM对象

与消息队列类似，SHM对象的创建或打开也需要一个唯一的键值标识，并且需要指定内存的大小尺寸，具体接口如下：

```
#include <sys/ipc.h>
#include <sys/shm.h>

int shmget(key_t key, size_t size, int shmflg);
```

接口说明：

- 返回值：SHM对象ID
- 参数key：SHM对象键值
- 参数size：共享内存大小
- 参数shmflg：创建模式和权限

- IPC_CREAT: 如果key对应的共享内存不存在, 则创建SHM对象
- IPC_EXCL: 如果该key对应的共享内存已存在, 则报错
- 权限与文件创建open类似, 用八进制表示

示例代码:

```
int main(void)
{
    key_t key = ftok(".", 1);
    int shmid;

    // 创建或打开一个大小为1024自己的SHM对象, 获取其ID
    shmid = shmget(key, 1024, IPC_CREAT|0666);
    if(shmid < 0)
    {
        perror("创建SHM对象失败");
    }

    // ...

    return 0;
}
```

2.2 映射 / 解除映射SHM对象

有了SHM对象的ID之后, 必须先将其映射到用户进程的内存空间之后方可使用, 映射接口如下:

```
#include <sys/types.h>
#include <sys/shm.h>

void *shmat(int shmid, const void *shmaddr/*一般为NULL*/, int shmflg);
```

接口说明:

- 功能:
 - 将指定的共享内存, 映射到本进程内存空间
- 参数:
 - shmid: 指定的共享内存的ID
 - shmaddr: 指定映射后的地址, 因为是虚拟地址, 分配的原则要兼顾诸如段对齐、权限分配等问题, 因此用户进程是无法指定的, 只能由系统自动分配, 因此此参数一般为NULL, 表示交由系统来自动分配。
 - shmflg: 可选项
 - 0: 默认, 代表共享内存可读可写。
 - SHM_RDONLY: 代表共享内存只读。
- 返回值:
 - 共享内存映射后的虚拟地址入口。

正确映射之后, 命令`ipcs -m`查看SHM对象时, 可从`nattch`列中看到已映射进程个数:

```
aidevelop@aidevelop-vm:~$ ipcs -m
```

```
----- Shared Memory Segments -----
```

key	shmid	owner	perms	bytes	nattch	status
0x00000000	688132	aidevelop	600	67108864	2	dest
0x00000000	688137	aidevelop	600	524288	2	dest
0x00000000	688138	aidevelop	600	524288	2	dest
0x00000000	688139	aidevelop	600	4389528	2	dest
0x00000000	688142	aidevelop	600	524288	2	dest
0x5101374a	20	aidevelop	600	1024	1	

使用完SHM对象后，需要将其跟进程解除关联关系，即解除映射，函数接口如下：

```
#include <sys/types.h>
#include <sys/shm.h>

int shmdt(const void *shmaddr);
```

该函数接口非常简单，参数就是从 shmat() 返回的SHM对象的入口指针。

2.3 其余操作

与其他IPC对象一样，共享内存也有一个control函数，可用于设置SHM对象属性信息、获取SHM属性信息、删除SHM对象等其余操作，接口如下：

```
#include <sys/ipc.h>
#include <sys/shm.h>

int shmctl(int shmid, int cmd, struct shmid_ds *buf);
```

接口说明：

- shmid：指定的共享内存的ID
- cmd：一些命令字
 - IPC_STAT：获取共享内存的一些信息，放入shmid_ds{ }中
 - IPC_SET：将 buf 中指定的信息，设置到本共享内存中
 - IPC_RMID：删除指定的共享内存，此时第三个参数 buf 将被忽略
- buf：用来存放共享内存信息的结构体

用的较多的就是删除SHM对象，示例代码如下：

```
shmctl(shmid, IPC_RMID, NULL);
```

上述代码作用为将该块共享内存标记为“待删除状态”，当nattch为0时，则会被删除。