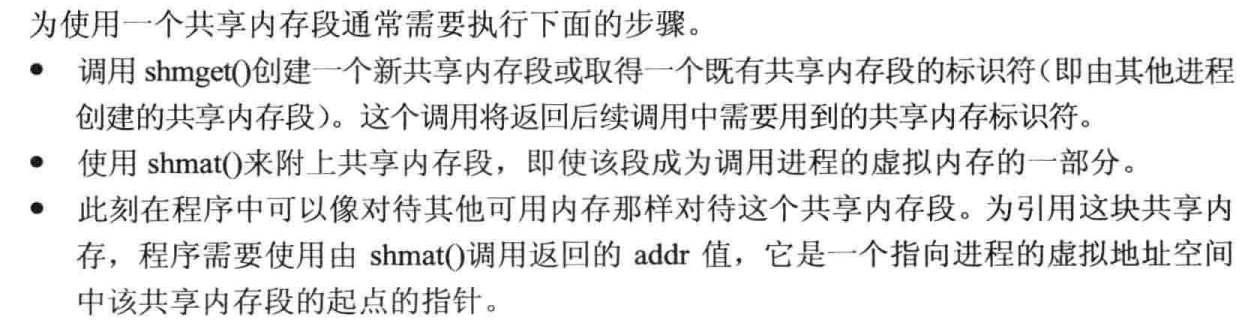
第四十八章

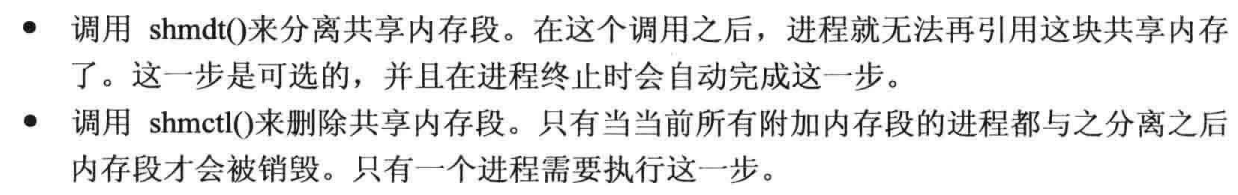
1. System V 共享内存：

共享内存段会成为一个进程用户空间的一部分，因此这种IPC机制无需内核介入，正因为如此，需要通过某种方法同步访问。

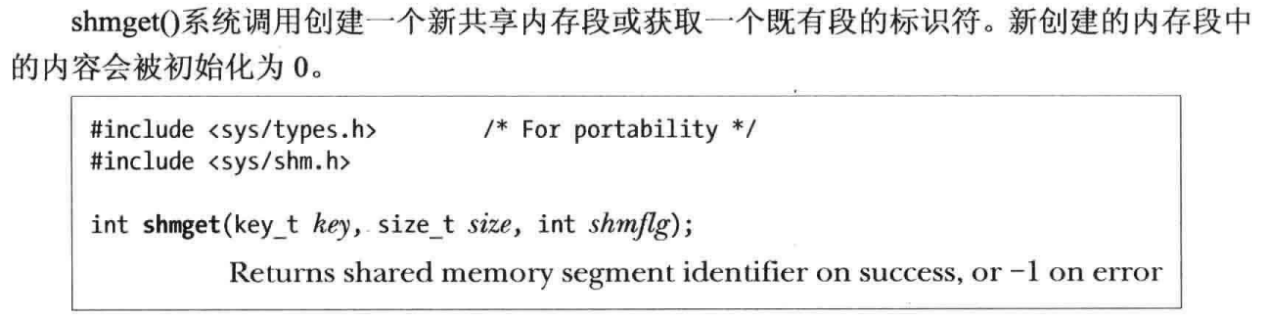
在mmap语义里，一块内存区域被映射到地址，而System V 共享内存是附加到一个地址上。

1. 概述：



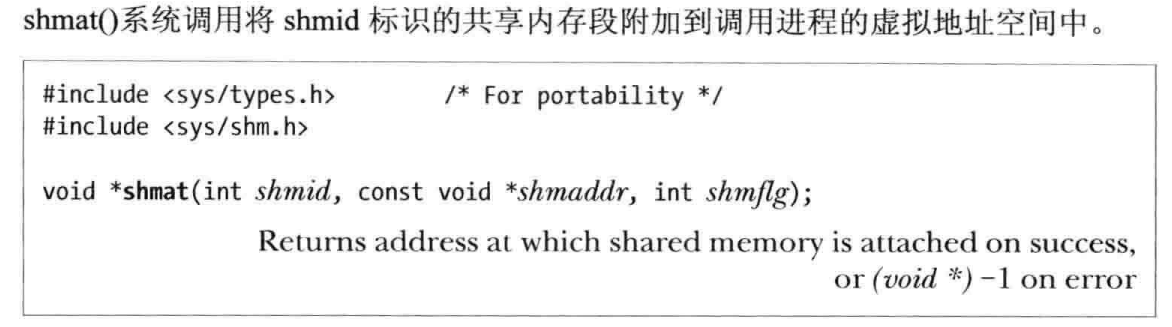


1. 创建或打开一个共享内存段：

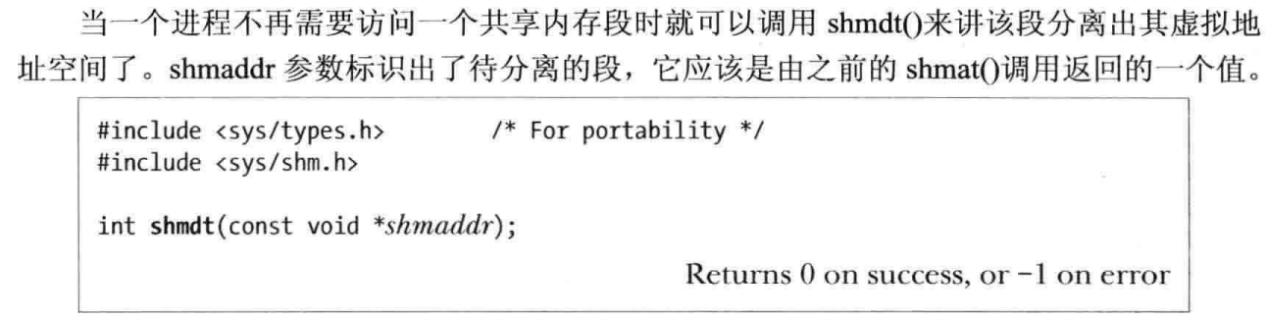


内核是以分页大小的整数倍分配内存的，因此size会被提升到最近的系统分页大小的整数倍。

1. 使用共享内存：



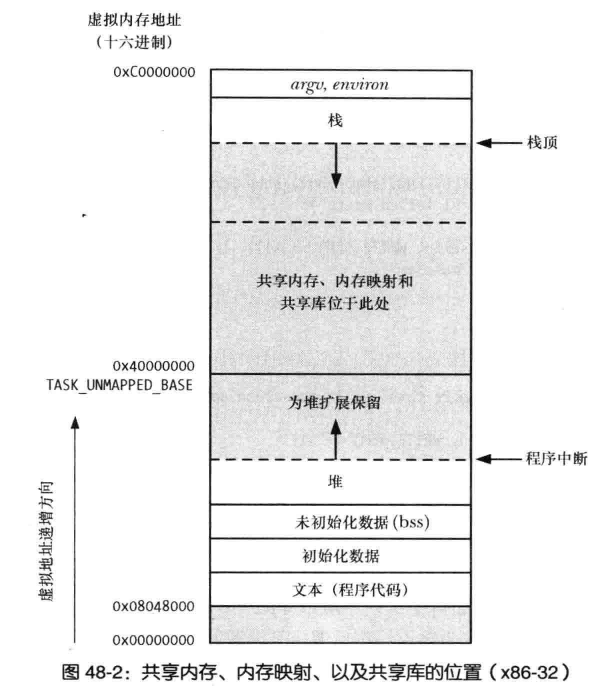
注意，shmaddr和shmflg的SHM\_RND参数设置控制着段如何被附加上去的。但是最优解是shmaddr为NULL。此外，还可以设置只读标志。具体查看手册和书本。



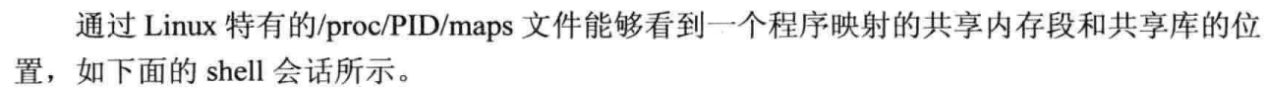
注意，通过fork创建的子进程会继承其父进程附加的共享内存段。Exec之后，所有附加的共享内存段都被分离，进程终止之后也会自动分离。

1. 共享内存在虚拟内存中的位置：

如果遵循所推介的方法，即允许内核选择在何处附加共享内存段，则内存布局如图：



注意，采用优选方式，附加共享内存段的虚拟地址从0x40000000开始。



至于maps文件的各个字段代表什么意思，具体查看书本。

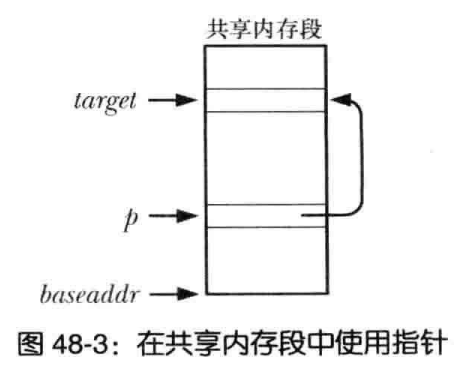
有一些概念搞不懂：

段在相应的映射文件中的偏移量。

共享内存段是被创建成不可见的tmpfs文件系统中的映射文件。

1. 在共享内存中存储指针：

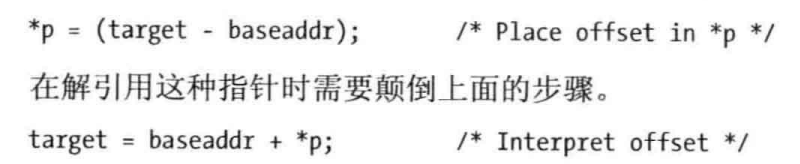
如果遵循推荐的做法，让内核选择将共享内存段附加到何处，那么一个段在各个进程中可能会被附加到不同的地址上。因此在共享内存段中存储指向段中其他地址的引用时，应该使用偏移量而不是指针。



错误做法：

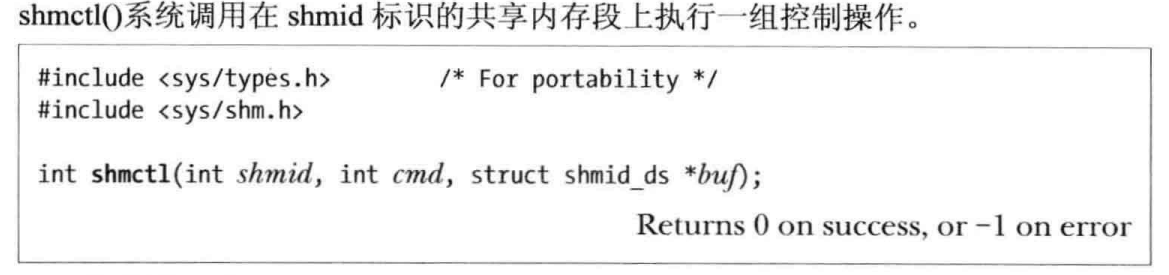


正确做法：

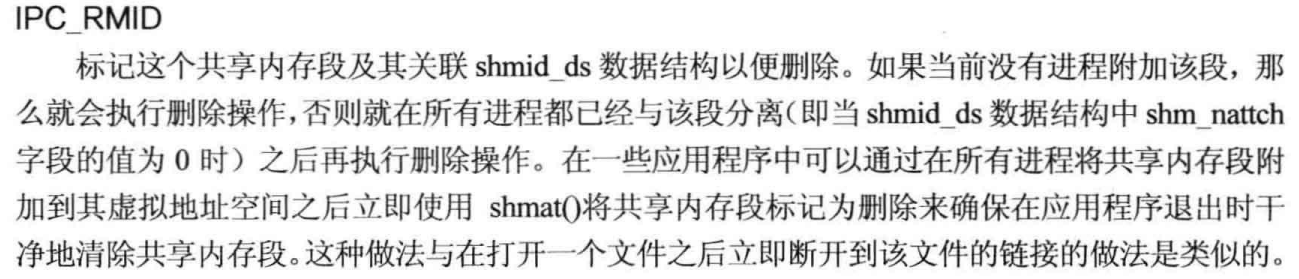


或者如果是一组固定大小的结构连接起来，则强制转换成数组。

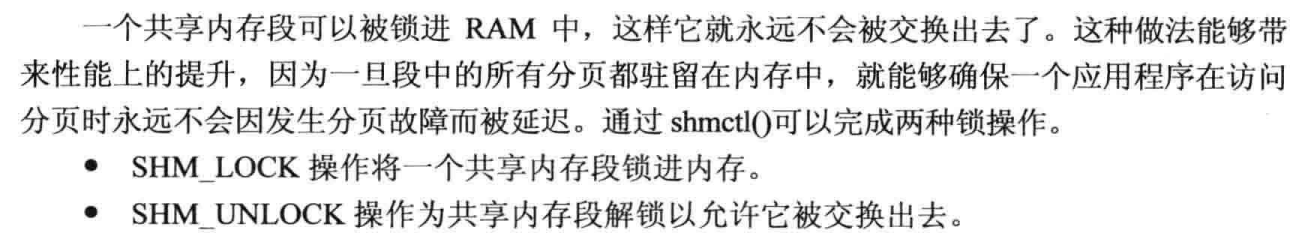
1. 共享内存控制操作：

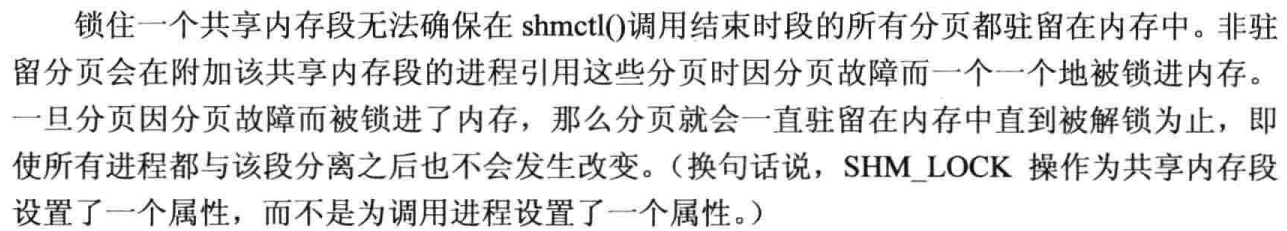


注意cmd为IPC\_RMID的时候：

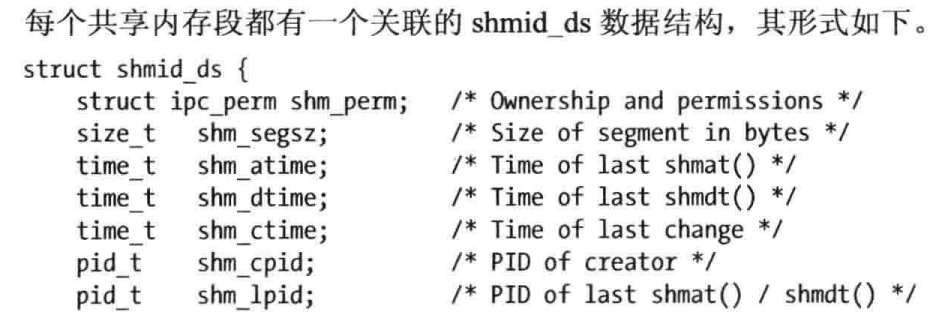


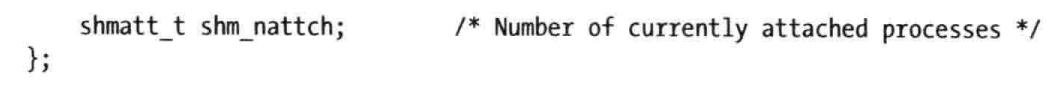
此外，加锁和解锁内存的控制操作：





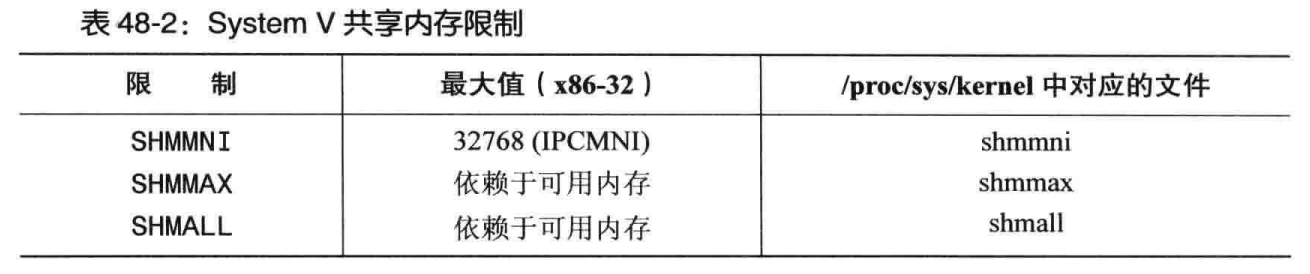
1. 共享内存关联数据结构：





需要注意的是，shm\_perm里的mode子字段除了常规的权限位，还有两个只读位掩码标记。分别是SHM\_DEST和SHM\_LOCKED。其余字段查看手册和书本。

1. 共享内存的限制：



具体的查看书本和手册。