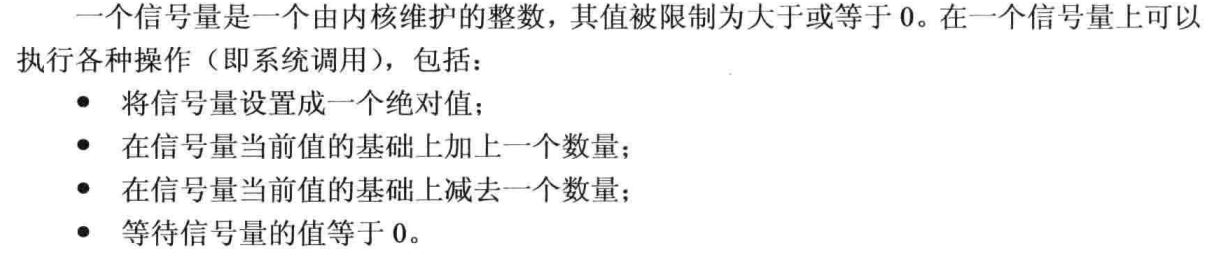
第四十七章

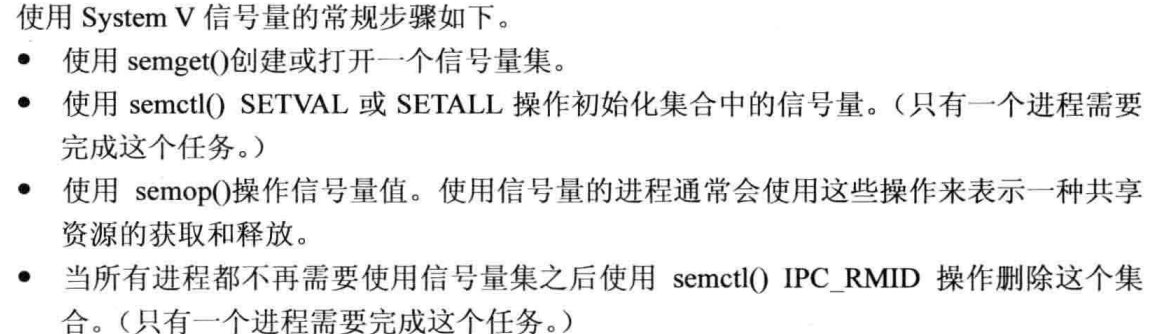
1. System V 信号量：

信号量的作用不是用来传输数据的，相反是用来同步进程动作。



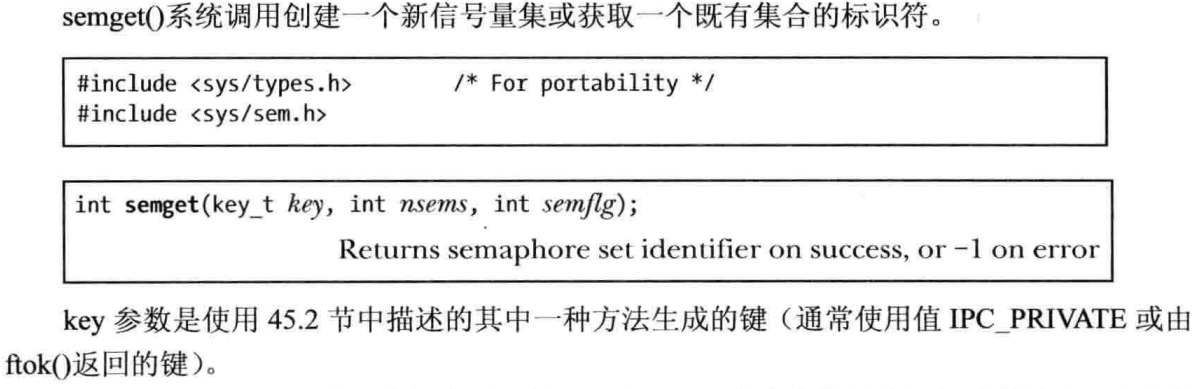
1. 概述：

使用System V 信号量的常规步骤如下：



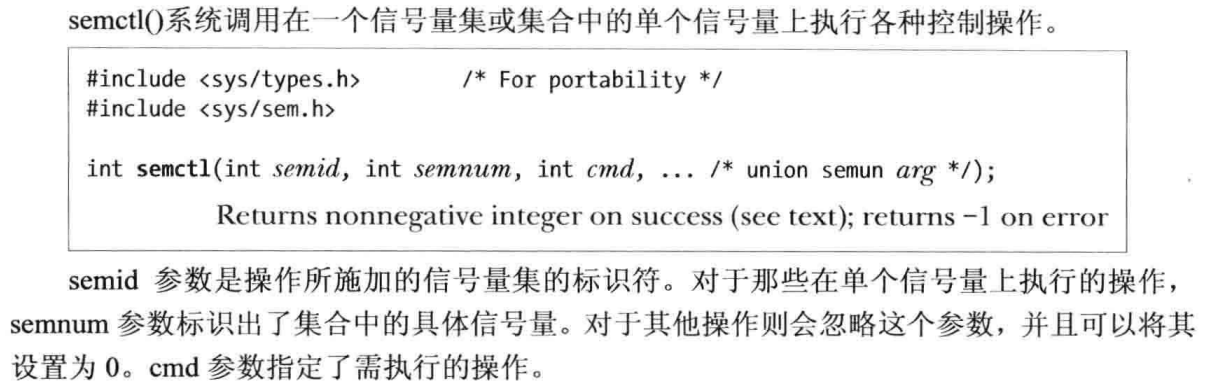
System V信号量的分配是以被称为信号量集的组为单位进行的。

1. 创建或打开一个信号量集：

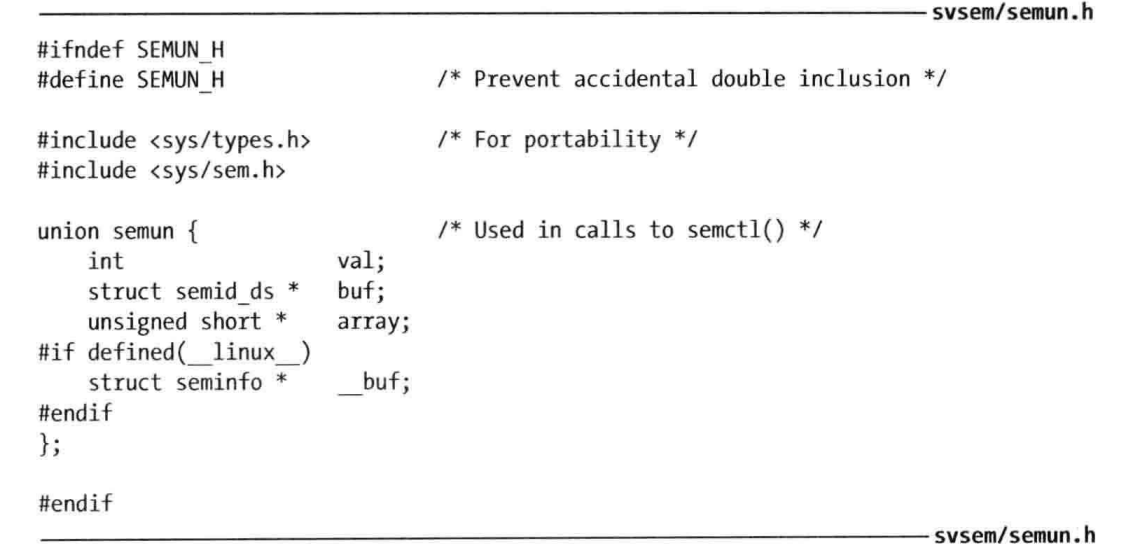


具体查看手册。

1. 信号量控制操作：



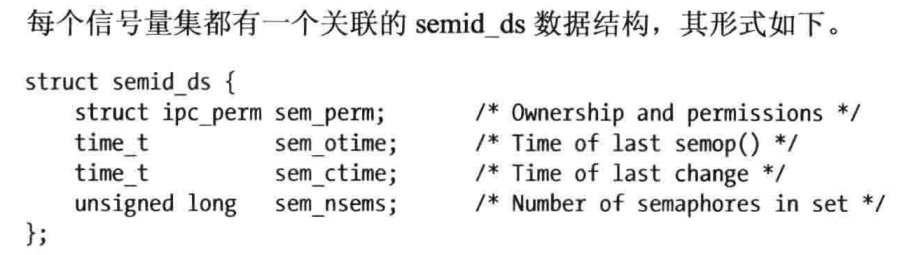
注意，一些特定的操作需要传入第四个参数，第四个参数需要程序员在程序中定义。



为了使程序完全可移植，在一些无需第四个参数的情况下也传入一个哑参数。

具体的cmd操作查看手册和书本。

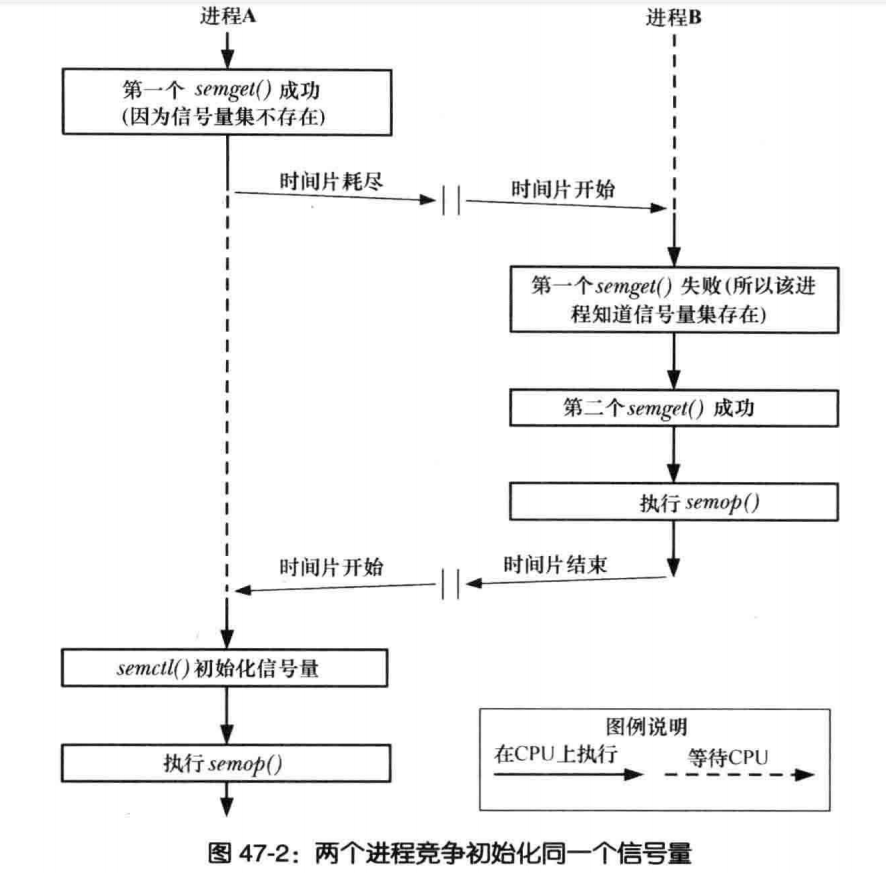
1. 信号量关联的数据结构：



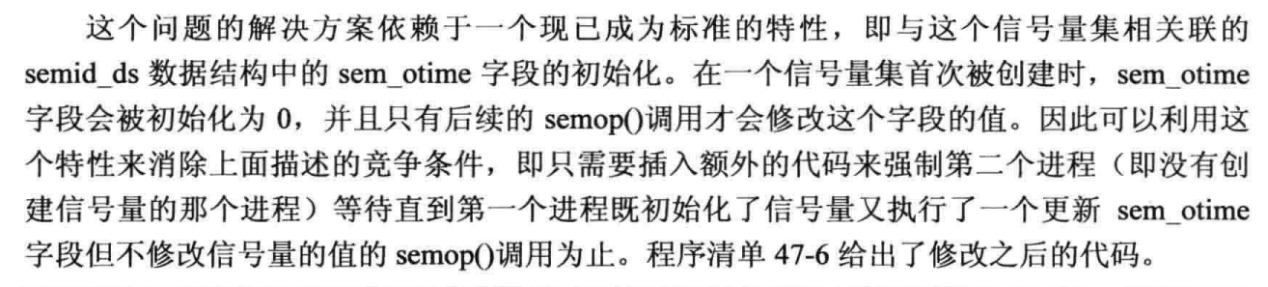
1. 信号量初始化：

确切地说，应该是多个地位平等的进程由于信号量的创建和初始化是单独的系统调用，所以导致在初始化一个信号量的时候会导致竞争。

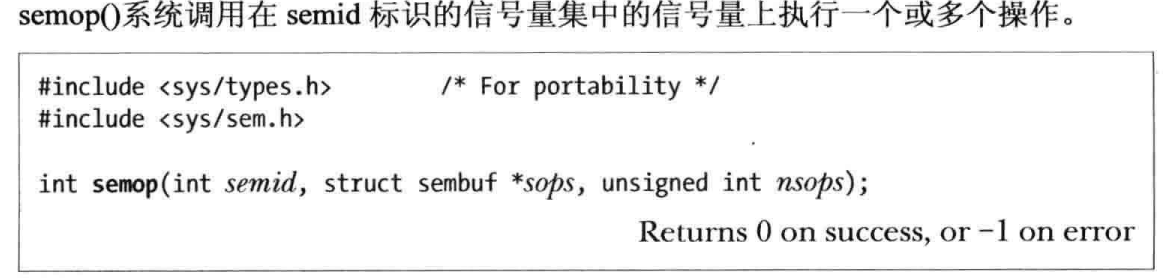
具体竞争的过程如下：

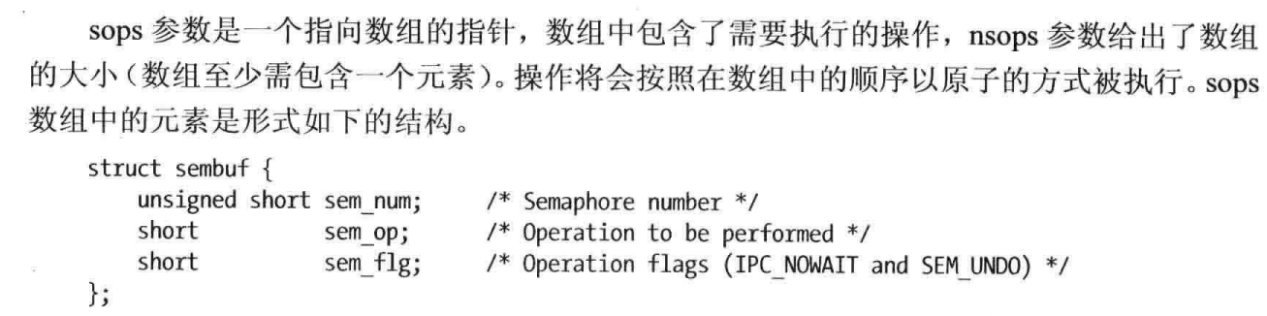


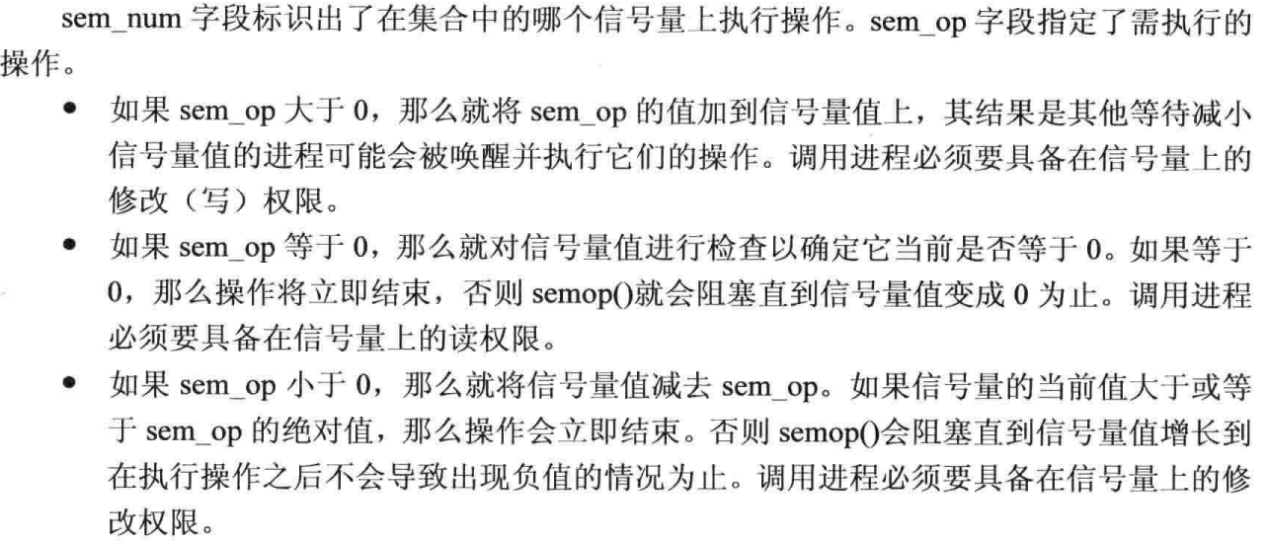
解决的方案如下：

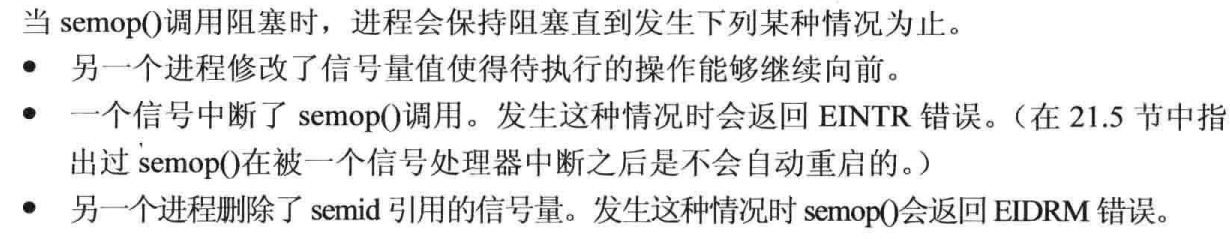


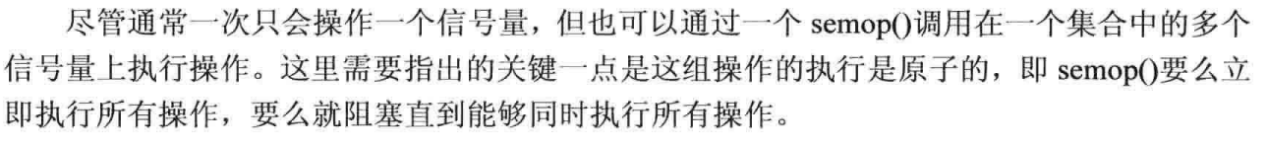
1. 信号量操作：

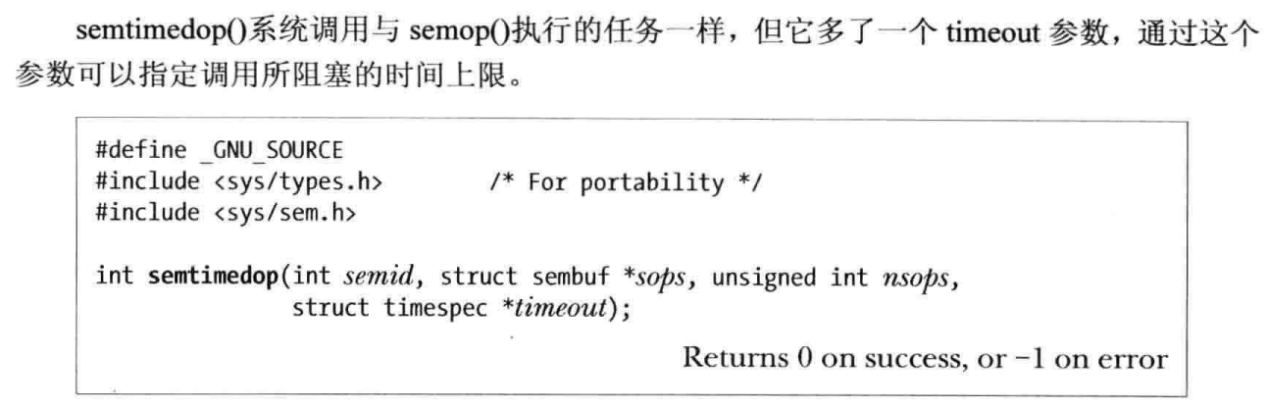












1. 多个阻塞信号量操作的处理：
2. 如果减去一个相同的信号量值导致阻塞，当条件允许时首先被执行的操作时不确定的。
3. 按照先满足条件先服务的顺序。
4. 注意防止饿死的情况发生。
5. 信号量撤销值：

通过semop修改一个信号量的值使用SEM\_UNDO的时候，内核会记录信号量的操作的结果，然后在进程终止的时候撤销这个操作。原理是，内核为每个进程在每个信号量上使用SEM\_UNDO标志维护一个信号量调整semadj的整数，它是使用SEM\_UNDO操作所做的调整总和。

注意，使用semctl的SETVAL和SETALL的时候，semadj会清空。

Fork创建的子进程不会继承父进程的semadj。在exec中这个值会得到保留。

SEM\_UNDO的限制：

1. 使用这个一般是与某些共享资源联系，信号量可以撤销，而资源难以一致。
2. 某些情况导致进程无法对信号量进行调整。
3. 可能会导致超过规定的最大值。
4. 实现一个二元信号量协议：

一般的应用程序不需要使用System V信号量以几何为单位分配和操作的情况。

二元信号量常用的操作也就是平时所说的P，V操作。

1. 信号量的限制：

具体查看书本。

1. System V信号量的缺点：

