**信号量**

# 一、定义

信号量本质上是一个计数器（不设置全局变量是因为进程间是相互独立的，而这不一定能看到，看到也不能保证++引用计数为原子操作）,用于多进程对共享数据对象的读取，它和管道有所不同，它不以传送数据为主要目的，它主要是用来保护共享资源（信号量也属于临界资源），使得资源在一个时刻只有一个进程独享。

由于信号量只能进行两种操作等待和发送信号，即P(sv)和V(sv),他们的行为是这样的：

（1）P(sv)：如果sv的值大于零，就给它减1；如果它的值为零，就挂起该进程的执行

（2）V(sv)：如果有其他进程因等待sv而被挂起，就让它恢复运行，如果没有进程因等待sv而挂起，就给它加1.

就犹如有一样东西（sv）和使用东西的人（进程），P（sv）可以解释为，人去使用这个东西（sv），而这个东西还有就拿去使用，并且对齐数量减一，如果没有了，那么“挂起”等待。

V（sv）可以解释为，我在使用一样东西，有人在等待使用这个东西而被挂起“，那么我使用完了就给它使用”，如果没有人要使用这个东西，那么我就放到对应区域并对其数量加一。

# 函数说明

## 1.创建信号量

**int semget(key\_t key,int nsems,int flags)**

key：是长整型（唯一非零），系统建立IPC通讯 （ 消息队列、 信号量和 共享内存） 时必须指定一个ID值。通常情况下，该id值通过ftok函数得到，由内核变成标识符，要想让两个进程看到同一个信号集，只需设置key值不变就可以。

nsem：指定信号量集中需要的信号量数目，它的值几乎总是1。

flag：是一组标志，当想要当信号量不存在时创建一个新的信号量，可以将flag设置为IPC\_CREAT与文件权限做按位或操作。

设置了IPC\_CREAT标志后，即使给出的key是一个已有信号量的key，也不会产生错误。而IPC\_CREAT | IPC\_EXCL则可以创建一个新的，唯一的信号量，如果信号量已存在，返回一个错误。一般我们会还或上一个文件权限

## 2.删除和初始化信号量

**int semctl(int semid, int semnum, int cmd, ...);**

sem\_id：是由semget返回的信号量标识符

semnum：当前信号量集的哪一个信号量

cmd：通常是下面两个值中的其中一个

SETVAL：用来把信号量初始化为一个已知的值。p 这个值通过union semun中的val成员设置，其作用是在信号量第一次使用前对它进行设置。

IPC\_RMID：用于删除一个已经无需继续使用的信号量标识符，删除的话就不需要缺省参数，只需要三个参数即可

## 3.改变信号量的值

**int semop(int semid, struct sembuf \*sops, size\_t nops);**

nsops：进行操作信号量的个数，即sops结构变量的个数，需大于或等于1。最常见设置此值等于1，只完成对一个信号量的操作

sembuf的定义如下：

