## 信号量

定义：

**信号量（semaphore）**它是一个计数器。信号量用于实现进程间的互斥与同步，而不是用于存储进程间通信数据。

### 1、特点

1. 信号量用于进程间同步，若要在进程间传递数据需要结合共享内存。
2. 信号量基于操作系统的 PV 操作，程序对信号量的操作都是原子操作。
3. 每次对信号量的 PV 操作不仅限于对信号量值加 1 或减 1，而且可以加减任意正整数。
4. 支持信号量组。

### 2、原型

最简单的信号量是只能取 0 和 1 的变量，这也是信号量最常见的一种形式，叫做**二值信号量（Binary Semaphore）**。而可以取多个正整数的信号量被称为通用信号量。

头文件#include <sys/sem.h>

// 创建或获取一个信号量组：若成功返回信号量集ID，失败返回-1

int semget(key\_t key, int num\_sems, int sem\_flags);

// 对信号量组进行操作，改变信号量的值：成功返回0，失败返回-1

int semop(int semid, struct sembuf semoparray[], size\_t numops);

// 控制信号量的相关信息

int semctl(int semid, int sem\_num, int cmd, ...);

当semget创建新的信号量集合时，必须指定集合中信号量的个数（即num\_sems），通常为1； 如果是引用一个现有的集合，则将num\_sems指定为 0

struct sembuf

{

short sem\_num; // 信号量组中对应的序号，0～sem\_nums-1

short sem\_op; // 信号量值在一次操作中的改变量

short sem\_flg; // IPC\_NOWAIT, SEM\_UNDO

}