

## 作业4

1. 读者写者问题（写者优先）：1) 共享读; 2) 互斥写、读写互斥; 3) 写者优先于读者（一旦有写者，则后续读者必须等待，唤醒时优先考虑写者）

```
Writer:
P(mutex)
write()
V(mutex)
```

```
Reader:
P(rmutex)
if readCount == 0:
    P(mutex)
readCount += 1
V(rmutex)
read()
P(rmutex)
readCount -= 1
if readCount == 0:
    V(mutex)
V(rmutex)
```

2. 寿司店问题。假设一个寿司店有 5 个座位，如果你到达的时候有一个空座位，你可以立刻就坐。但是如果你到达的时候5个座位都是满的有人已经就坐，这就意味着这些人都是一起来吃饭的，那么你需要等待所有的人一起离开才能就坐。编写同步原语，实现这个场景的约束。

```
Semaphore mutex = 1;
Semaphore empty = 5;
Semaphore barrier = 0;
int full = 0;

P(empty);
P(mutex);
full++;
if (full == 5) {
    V(mutex);
    V(barrier);
} else {
    V(mutex);
}

P(barrier);
eat();
P(mutex);
full--;
if (full == 0) {
    V(barrier);
}
V(mutex)
V(empty)
```

3. 进门问题。(1) 请给出P、V操作和信号量的物理意义。(2) 一个软件公司有5名员工，每人刷卡上班。员工刷卡后需要等待，直到所有员工都刷卡后才能进入公司。为了避免拥挤，公司要求员工一个一个通过大门。所有员工都进入后，最后进入的员工负责关门。请用P、V操作实现员工之间的同步关系。

(1)

P操作用于请求资源，当一个线程或进程执行P操作时，它试图获取一个资源的使用权。如果该资源可用，那么该线程将继续执行，而资源被标记为已被使用。如果资源当前不可用，那么该线程将被阻塞，直到资源变为可用。

V操作用于释放资源，当一个线程或进程执行V操作时，它释放先前使用的资源。如果有其他线程或进程正在等待该资源，那么其中一个将被唤醒，以便它可以继续执行。

信号量可以视为一个计数器，表示资源的可用数量。P操作就像从这个计数器中减去一个单位，表示使用了一个资源；V操作就像向计数器中添加一个单位，表示释放了一个资源。如果计数器的值为正数，表示有可用的资源；如果为零或负数，表示资源已被使用，需要等待其他线程或进程释放资源。

(2)

```
Semaphore enter = 1;
Semaphore door = 1;
Semaphore count = 0;
```

```
P(enter);
V(count);
if (count == 5) {
    P(door);
}
V(enter);
```

```
P(enter);
P(count);
if (count == 0) {
    V(door);
}
V(enter);
```

4. 搜索-插入-删除问题。三个线程对一个单链表进行并发的访问，分别进行搜索、插入和删除。搜索线程仅仅读取链表，因此多个搜索线程可以并发。插入线程把数据项插入到链表最后的位置；多个插入线程必须互斥防止同时执行插入操作。但是，一个插入线程可以和多个搜索线程并发执行。最后，删除线程可以从链表中任何一个位置删除数据。一次只能有一个删除线程执行；删除线程之间，删除线程和搜索线程，删除线程和插入线程都不能同时执行。请编写三类线程的同步互斥代码，描述这种三路分类互斥问题。

```
Semaphore insert = 1;
Semaphore search = 1;
Semaphore delete = 1;
```

```
searchThread() {
    while(true) {
        P(search);
        search();
        V(search);
    }
}
```

```
insertThread() {  
    while(true) {  
        P(insert);  
        insert();  
        V(insert);  
    }  
}  
  
deleteThread() {  
    while(true) {  
        P(delete);  
        delete();  
        V(delete);  
    }  
}
```