同步互斥举例

超市购物问题

超市可以容纳500人同时购物，有6扇可供出入的门，既可以进又可以出，每扇门只允许一个人通过，使用信号量解决一下问题：

a) 描述购物过程；

b) 如果增加一个限制条件：顾客进出必须走同一个门，这个过程又是怎么样的；

问题1分析：

问题定性：一个混合问题（同步和互斥）。

同步：超市里面维持500个顾客的规模；

互斥：每道门都是临界资源；

进程的类别：两类代表进程，进入超市，离开超市。

初始化：

信号量需要7个，同步信号量一个，定义为s=500,

互斥量6个，定义为s1=s2=s3=s4=s5=s6=1

写代码

```
void enter()
{
        semwait(s);
        choose the door i
        semwait(si);
        cross the door  i;
        semsignal(si)
        buy something;

}
```

```
void leave()
{
        choose door i;
        semwait(si);
        cross the door;
        semsignal(si)
        semsignal(s)

}
```

问题2 分析

- 关系和问题1完全形同，只不过需要在进程间定义一个实例变量用来记录进程进门的状态；

写代码

```
void enter()
{
        semwait(s);
        choose the door i
        self.door=i;
        semwait(si);
        buy something;
        semsignal(si)

}
```

```
void leave()
{
        i=self.door;
        semwait(si);
        cross the door;
        semsignal(si)
        semsignal(s)

}
```

理发师问题

一个理发店有1个理发师，n张椅子，1张理发椅。若没有要理发的顾客，则理发师就去睡觉；若有顾客走进理发店且所有的椅子都被占用了，则该顾客就离开理发店；若理发师正在为人理发，则该顾客就找一张空椅子坐下等待；若理发师在睡觉，则顾客就唤醒他，请用信号量描述这个过程。

问题分析

- 通过问题描述，我们发现这个问题有两类进程：理发师与顾客进程。进程之间的关系是复合关系，既有同步关系（唤醒和睡觉），也有进程间的互斥（顾客等候人数）；

- 初始化：定义两个信号量：mutex=1，用来实现对顾客人数的互斥访问， barbers=0（初始时，理发室在睡觉）用来描述理发师的状态。定义一个共享变量count=0，用来记录客人的数目。

```
void baber()
{
        while(true)
        {
                semwait(baber)#唤醒

                cut the hair；
                semwait(mutex)
                        count-=1;
                    if (count!=0)
                    semsignal(barber)#保持清醒
                semsignal(mutex)
        }
}
```

```
void customer()
{
        semwait(mutex);
            if (count<n+1)
            {
                        count+=1
                        if (count==1)
                                semsignal(barber)
                        else:
                                sit  on the  chair


            }
            else
            {
                        leave the baber's
            }
        semsignal(mutex)
}
```

过桥问题

- 有一座桥，东西走向，汽车可以从东往西走，也可以西往东走，桥上每次只允许朝一个方向走，请用信号量描述下列过程：
- 1）如果某一个方向的车占有桥，让这个方向的车优先过桥；
- 2）让两个方向的车公平的过桥。

问题分析

1. 问题1的模型和读者写者问题中的读者优先问题类似，在问题中存在两类进程，从东往西走的车，和从西往东走的车，两者存在竞争关系，竞争争夺桥的使用权，这是互斥关系，同时对于两类进程之间有存在两个竞争关系，对进程的数目进行统计。所以这个问题是一个典型的互斥关系；

2. 初始化：定义三个信号量：mutex=1,emutex=1,wmutex=1;
   定义两个全局变量ecount 和wcount 用来统计两边过桥的汽车数目

```
void  e2w()
{
        semwait(emutex)
                ecount+=1
                if (ecount==1) semwait(mutex)
        semsignal(emutext)
                cross the bridge from the east to west
        semwait(emutex)
                ecount-=1
                if (ecount==0) semsignal(mutex)
        semsignal(emutex)
}
```

```
void  w2e()
{
        semwait(wmutex)
                wcount+=1
                if (wcount==1) semwait(mutex)
        semsignal(wmutext)
                cross the bridge from the west to east
        semwait(wmutex)
                wcount-=1
                if (wcount==0) semsignal(mutex)
        semsignal(wmutex)
}
```

问题分析

1. 第一个问题带来一个不公平的问题就是，一旦一个方向占有的桥的使用权，就会长期霸占桥的使用权，所以引入第二个问题。思路就是让两边新加入的进程按照先后顺序抢夺桥的使用权,只有一个方向有收到请求，就不允许占有桥使用权的车继续增加请求了。

2. 初始化：定义四个信号量：mutex=1,emutex=1,wmutex=1, queue=1;
   定义两个全局变量ecount 和wcount 用来统计两边过桥的汽车数目

```
void  e2w()
{
        semwait(queue);
        semwait(emutex)
                ecount+=1
                if (ecount==1) semwait(mutex)
                semsignal(queue)
        semsignal(emutext)
                cross the bridge from the east to west

        semwait(emutex)
                ecount-=1
                if (ecount==0) semsignal(mutex)
        semsignal(emutex)
}
```

```
void  w2e()
{
        semwait(queue)
        semwait(wmutex)
                wcount+=1
                if (wcount==1) semwait(mutex);
                semsignal(queue)
        semsignal(wmutext)
                cross the bridge from the west to east
                semwait(wmutex)
                wcount-=1
                if (wcount==0) semsignal(mutex)
        semsignal(wmutex)
}
```