

## 操作系统第四次作业参考解答

1、某工厂有两个生产车间和一个装配车间，两个车间分别生产 A、B 两种零件，装配车间的任务是把 A、B 两种零件组装成产品。两个生产车间每生产一个零件后都要把他们送到装配车间的货架 F1、F2 上，F1 存放零件 A，F2 存放零件 B，F1 和 F2 的容量均为可以存放 10 个零件。装配工人每次从货架上取一个 A 零件和一个 B 零件然后组装成产品，请用 PV 操作进行正确管理。

解答：

```
semaphore mutex1=1;    //互斥访问货架 F1

semaphore mutex2=1;    //互斥访问货架 F2

semaphore empty1=10;   //管理货架 F1

semaphore full1=0;     //管理货架 F1

semaphore empty2=10;   //管理货架 F2

semaphore full2=0;     //管理货架 F2

workshopA()
{
    while(true)
    {
        produce an item
        P(empty1);
        P(mutex1);
        send the item to F1;
        V(mutex1);
        V(full1);
    }
}

workshopB()
{
```

```

while(true)
{
    produce an item
    P(empty2);
    P(mutex2);
    send the item to F2;
    V(mutex2);
    V(full2);
}
}
worker()
{
    while(true)
    {
        P(full1);
        P(mutex1);
        get an item from F1;
        V(mutex1);
        V(empty1);
        P(full2);
        P(mutex2);
        get an item from F2;
        V(mutex2);
        V(empty2);
        produce a product
    }
}

```

2、系统中有多生产者进程和消费者进程，共享一个可以存 1000 个产品的缓冲区（初始为空），当缓冲区为未空时，生产者进程可以放入一件其生产的产品，否则等待；当缓冲区为未满时，消费者进程可以取走一件产品，否则等待。要求一个消费者进程从缓冲区连续取出 10 件产品后，其他消费者进程才可以取产品，请用信号量和 PV 操作实现进程间的互斥和同步，要求写出完整的过程；并指出所用信号量的含义和初值

解答：

```

semaphore mutex1=1;    //消费者互斥

semaphore mutex2=1;    //缓冲区互斥

semaphore empty=1000; //生产者消费者问题模型

semaphore full=0;      //生产者消费者问题模型

producer()
{
    while(1){
        produce an item;
        P(empty);
        P(mutex2);
        put the item into the buffer;
        V(mutex2);
        V(full)
    }
}

consumer()
{
    while(1){
        P(mutex1);
        for(int i=0;i<10;i++){
            P(full);
            P(mutex2);
            take an item from the buffer;
            V(mutex2);
            V(empty);
            consume the item;
        }
        V(mutex1)
    }
}

```

3、一个主修人类学、辅修计算机科学的学生参加了一个课题，调查是否可以教会非洲狒狒理解死锁。他找到一处很深的峡谷，在上边固定了一根横跨峡谷的绳索，这样狒狒就可以攀住绳索越过峡谷。同一时刻，只要朝着相同的方向就可以有几只狒狒通过。但如果向东和向西的狒狒同时攀在绳索上那么会产生死锁（狒

狒会被卡在中间)，由于它们无法在绳索上从另一只的背上翻过去。如果一只狒狒想越过峡谷，它必须看当前是否有别的狒狒正在逆向通行。利用信号量编写一个避免死锁的程序来解决该问题。不考虑连续东行的狒狒会使得西行的狒狒无限制的等待的情况。

解答：

```
int countWE=0;          //东行数量

int countEW=0;          //西行数量

semaphore mutexWE=1;    //保护 countWE

semaphore mutexEW=1;    //保护 countEW

semaphore mutex=1;      //互斥访问绳索

//东行
WtoE()
{
    P(mutexWE);
    if(countWE==0) P(mutex);
    countWE++;
    V(mutexWE);

    通过

    P(mutexWE);
    countWE--;
    if(countWE==0) V(mutex);
    V(mutexWE);
}

//西行
EtoW()
{
    P(mutexEW);
    if(countEW==0) P(mutex);
    countEW++;
```

```

V(mutexEW);

通过

P(mutexEW);
countEW--;
if(countEW==0) V(mutex);
V(mutexEW);
}

```

4、有一个仓库，可以存放 A 和 B 两种产品，仓库的存储空间足够大，每次仅允许一种产品入库一个，而且要求 A 和 B 产品要满足如下条件： $-N < A \text{ 产品数量} - B \text{ 产品数量} < M$ 。其中，N 和 M 是正整数。使用 P 操作和 V 操作描述产品 A 和产品 B 的入库过程。

**解答：**

```

semaphore mutex=1;    //互斥访问临界区

semaphore Sa=M-1;     //控制产品 A 数量

semaphore Sb=N-1;     //控制产品 B 数量

A()
{
    while(true)
    {
        P(Sa);
        P(mutex);
        put product A;
        V(mutex);
        V(Sb)
    }
}

B()
{
    while(true)
    {
        P(Sb);

```

```

        P(mutex);
        put product B;
        V(mutex);
        V(Sa)
    }
}

```

5、理发店里有一位理发师、一把理发椅和  $n$  把供等候理发的顾客坐的椅子。如果没有顾客，理发师便在理发椅上睡觉。一个顾客到来时，它必须叫醒理发师。如果理发师正在理发时又有顾客来到，则如果有空椅子可坐，就坐下来等待，否则就离开。试用 P、V 操作解决理发师问题。

解答：

```

int waiting=0;           //等候理发顾客数量

semaphore customers=0; //顾客是否存在

semaphore barber=0;     //理发师是否空闲

semaphore mutex=1;      //保护 waiting 变量

barber()
{
    while(true)
    {
        P(customers);      //理发师等待顾客

        P(mutex);         //进入临界区

        waiting=waiting-1; //修改 waiting 变量

        V(barber);        //理发师响应顾客等待

        V(mutex);         //退出临界区

        cut hair
    }
}

```

```
customer()
{
    P(mutex);          //进入临界区
    if(waiting<n)
    {
        waiting=waiting+1; //修改 waiting 变量

        V(customers);      //顾客响应理发师等待

        V(mutex);         //退出临界区

        P(barber);         //顾客等待理发师
        get hair cut
    }
    else
    {
        V(mutex);         //退出临界区
    }
}
```