

1. 解:

//定义信号量

```
Semaphore  count-available, //博物馆的空余容量  
            mutex-enter,    //入口  
            mutex-leave;    //出口
```

//初始化信号量。互斥信号量的初始值总是1

```
mutex-enter=mutex-leave = 1;
```

```
count-available=700;
```

参观者进程：

```
{  
    ...  
    P(count-available);  
    P(mutex-enter);  
    进门;  
    V(mutex-enter);  
    参观;  
    P(mutex-leave);  
    出门;  
    V(mutex-leave);  
    V(count-available);  
    ...  
}
```

思考：如果出口和入口使用一个信号量会导致什么问题？

2. 解:

//Jar 缸; bucket 桶; well 井

//定义同步类信号量

semaphore empty-jar, full-jar;

//定义资源类信号量

semaphore count-bucket, mutex-well, mutex-jar;

//初始化信号量

empty-jar=10; full-jar=0;

count-bucket=5; mutex-well=1; mutex-jar=1;

入水进程

```
While(1) {
```

```
    P(empty-jar);    // Jar 缸
```

```
    P(count-bucket); // bucket桶
```

```
    P(mutex-well);   //well 井
```

```
    从井中取水;
```

```
    V(mutex-well);   //well 井
```

```
    P(mutex-jar);    // Jar 缸
```

```
    送入水缸;
```

```
    V(mutex-jar);    // Jar 缸
```

```
    V(count- bucket); // bucket 桶
```

```
    V(full-jar);      // Jar 缸
```

```
}
```

先申请同步信号量，
再申请资源信号量

取水进程

While(1)

{

P(full-jar); // Jar 缸

P(count-bucket); // bucket 桶

P(mutex-jar); // Jar 缸

从缸中取水;

V(mutex-jar); // Jar 缸

V(count- bucket); // bucket 桶

V(empty-jar); // Jar 缸

}

先申请同步信号量,
再申请资源信号量

3. 解:

//定义同步类信号量

semaphore empty, // 盘子是否为空

full-apple , // 是否可以拿苹果

full-orange; // 是否可以拿橘子

//初始化信号量

empty= 1;

full-apple=0;

full-orange=0;

爸爸

```
void father()  
{  
    while(true)  
    {  
        P(empty);  
        // 放入苹果  
        V(full-apple);  
    }  
}
```

妈妈

```
void mother()
```

```
{
```

```
    while(true)
```

```
    {
```

```
        P(empty);
```

```
        // 放入橘子
```

```
        V(full-orange);
```

```
    }
```

```
}
```


哥哥

```
void brother()  
{  
    while(true)  
    {  
        P(full-apple);  
        // 拿苹果  
        V(empty);  
    }  
}
```

妹妹

```
void sister()  
{  
    while(true)  
    {  
        P(full-orange);  
        // 拿橘子  
        V(empty);  
    }  
}
```

思考:

- (1) 与经典生产者消费者问题相比有什么变化?
- (2) 需要为盘子设置互斥信号量吗?

4. 解:

//定义同步类信号量

```
semaphore empty,    // 盘子的空闲容量
           full-apple , // 苹果的数量
           full-orange; // 橘子的数量
```

//定义互斥类信号量

```
semaphore mutex_plate; //对盘子的操作是互斥的
```

//初始化信号量

```
empty= N; full-apple=0; full-orange=0;
mutex_plate=1;
```

爸爸

```
void father()
```

```
{
```

```
    while(true)
```

```
    {
```

```
        P(empty);
```

```
        P(mutex_plate);
```

```
        // 放入苹果
```

```
        V(mutex_plate);
```

```
        V(full-apple);
```

```
    }
```

```
}
```

妈妈

```
void mother()
```

```
{
```

```
    while(true)
```

```
    {
```

```
        P(empty);
```

```
        P(mutex_plate);
```

```
        // 放入橘子
```

```
        V(mutex_plate);
```

```
        V(full-orange);
```

```
    }
```

```
}
```

哥哥

```
void brother()
```

```
{
```

```
    while(true)
```

```
    {
```

```
        P(full-apple);
```

```
        P(mutex_plate);
```

```
        // 拿苹果
```

```
        V(mutex_plate);
```

```
        V(empty);
```

```
    }
```

```
}
```

妹妹

```
void sister()
```

```
{
```

```
    while(true)
```

```
    {
```

```
        P(full-orange);
```

```
        P(mutex_plate);
```

```
        // 拿橘子
```

```
        V(mutex_plate);
```

```
        V(empty);
```

```
    }
```

```
}
```

思考：与第3题相比有什么变化？