

生产者与消费者问题

- PV操作是用来操作信号量的。
 - P：等待——wait——减法作用——阻塞作用
 - V：释放——signal——加法作用——唤醒作用
 - `S.value > 0`：有空闲CPU；
 - `S.value = 0`：CPU刚好用完；
 - `S.value < 0`：进程等待；

■ P操作

```
void wait(S)
{
    S.value--;
    if (S.value < 0)
    {
        加入阻塞队列;
    }
}
```

■ V操作

```
void signal(S)
{
    S.value++;
    if (S.value <= 0)
    {
        唤醒阻塞队列第一个进程;
    }
}
```

- 解题思路：

1. 画图理解题目
2. 判断题目类型
3. 分析进程数目，填写进程模板
4. 补充基本代码
5. 补充P,V代码
6. 调整代码

- 例：爸爸往桌子上每次放一个苹果，儿子每次从桌子上拿一个苹果，放苹果和拿苹果不能同时进行，桌子上最多放10个苹果，用PV操作实现同步互斥。
 - 分析：爸爸——生产者——剩余空间(empty) = 10
 - 儿子——消费者——已占用空间(full) = 0

信号量 $S = 1$ ，用来实现互斥。

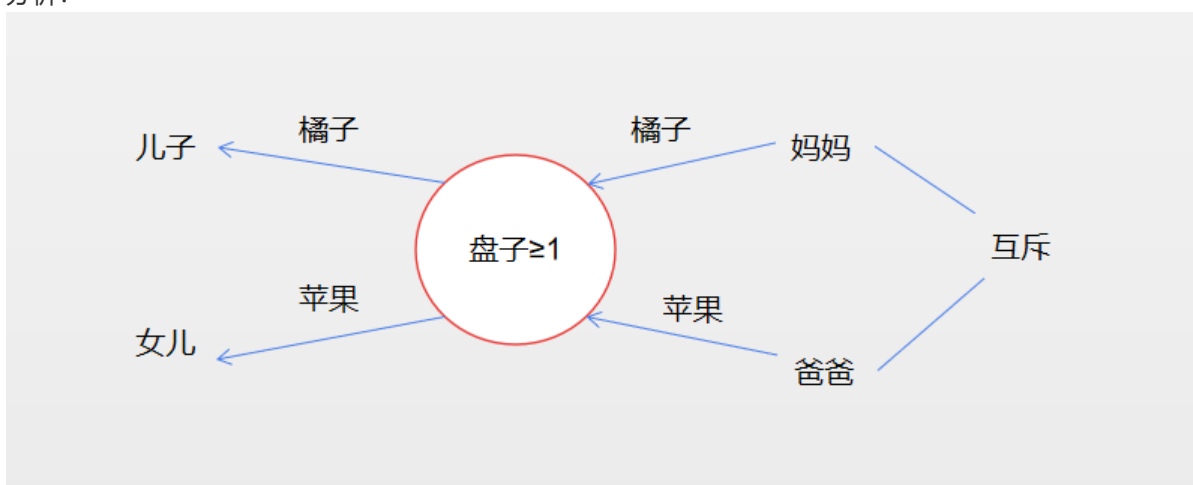
- 特征：①容器 \leq 容量
- ②生产 消费
- 伪代码

```
Dad()
{
    while(1)
    {
        P(empty); //判断盘子是否已满
        P(S);
        放苹果;
        V(S);
        V(full); //对已占用空间+1
    }
}
```

```
Son()
{
    while()
    {
        P(full); //判断盘子是否为空
        P(S);
        取苹果;
        V(S);
        V(empty); //对盘子剩余空间+1
    }
}
```

- 例：桌子上有个盘子，每次只能放一个水果，妈妈放橘子，爸爸放苹果，儿子吃橘子，女儿吃苹果。盘子为空，爸爸或妈妈才能放水果，盘子有水果时，儿子和女儿才能取水果。

- 分析：



- 初始值：orange = 0
apple = 0

plate = 1

S = 1

o 伪代码

```
Mom()
{
    while(1)
    {
        P(plate);
        P(S);
        放橘子;
        V(S);
        V(orange);
    }
}
```

```
Dad()
{
    while(1)
    {
        P(plate);
        P(S);
        放苹果;
        V(S);
        V(apple);
    }
}
```

```
Son()
{
    while(1)
    {
        P(orange);
        P(S);
        取橘子;
        V(S);
        V(plate);
    }
}
```

```

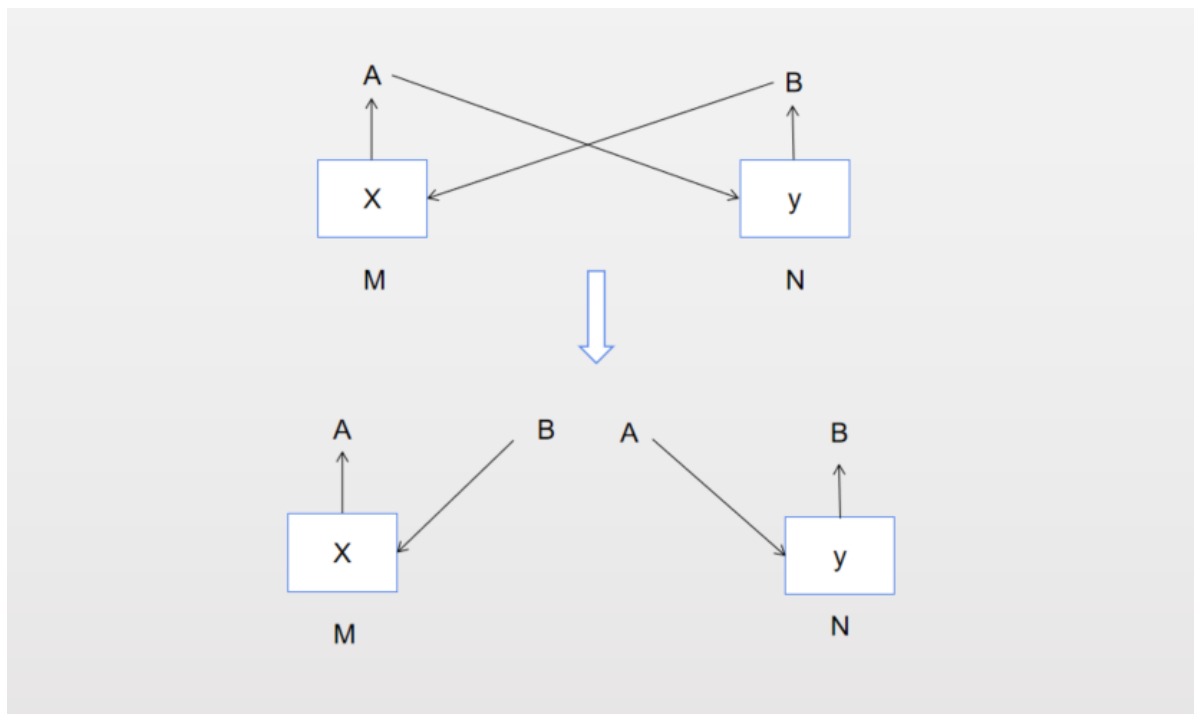
Daughter()
{
    while(1)
    {
        P(apple);
        P(S);
        取苹果;
        V(S);
        V(plate);
    }
}

```

- 操作同一个对象的时候，使用信号量S来实现互斥。

• [2015—408统考]

- 有A,B两个人通过信箱辩论，每个人都从自己的信箱取得对方的问题，将答案和新问题组成一个邮件放入对方的信箱中。假设A的信箱可装M封邮件，B的信箱可装N封邮件。初始时A信箱有X封，B信箱有Y封，，辩论者每次只取一封邮件。请用P,V操作实现，并解释信号量的初始值和含义。



- 分析：生产者A——B的剩余空间

生产者B——A的剩余空间

消费者A——自己邮箱还有多少封信

消费者B——自己邮箱还有多少封信

full-A：A信箱有多少封邮件

empty-A：A信箱的容量

full-B：B信箱有多少封邮件

empty-B：B信箱的容量

mutex-A：A信箱的信号量

mutex-B：B信箱的信号量

■ 初始值: `mutex-A = 1`, `mutex-B = 1`

`full-A = x`, `empty-A = M - X`

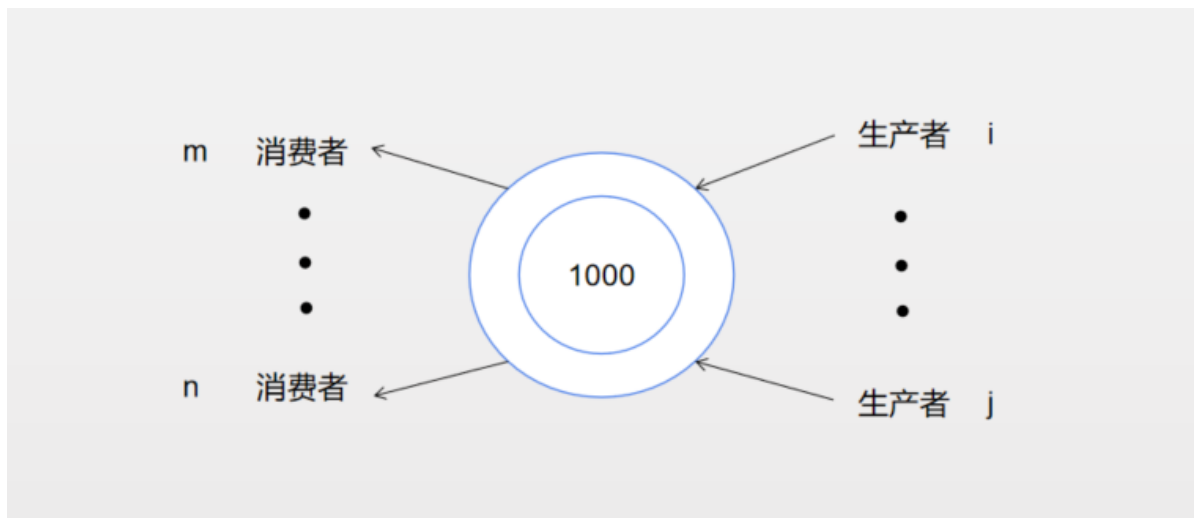
`full-B = y`, `empty-B = N - y`

```
A()
{
    while(1)
        P(full-A);
        P(mutex-A);
        从自己邮箱取信;
        V(mutex-A);
        V(empty-A);
        读信;
        P(empty-B);
        P(mutex-B);
        向B邮箱投信;
        V(mutex-B);
        V(full-B);
}
```

```
B()
{
    while(1)
        P(full-B);
        P(mutex-B);
        从自己邮箱取信;
        V(mutex-B);
        V(empty-B);
        读信;
        P(empty-A);
        P(mutex-A);
        向A邮箱投信;
        V(mutex-A);
        V(full-A);
}
```

• [2014—408统考]

- 例: 系统中有多生产者与消费者, 共享一个能存放1000件产品的环形缓冲区(初始为空)。当缓冲区未
满时生产者可放入生产的一个产品, 否则等待。当缓冲区未空时, 消费者进程可取走一件产品, 否则等
待。要求一个消费者从缓冲区连续取10件产品后, 其他消费者才能取走产品。请用P, V操作实现该流程
并解释信号量含义。



- 分析: 生产者 (j) — 剩余空间

生产者 (i) — 剩余空间

消费者 (m) — 物件数量

消费者 (n) — 物件数量

- 初始值: `empty = 1000;`

`full = 0;`

`mutex = 1;`

`mutex1 = 1;`

```
j()
{
    while(1)
    {
        P(empty);
        P(mutex);
        放一件产品;
        V(mutex);
        V(full);
    }
}
```

```
i()
{
    while(1)
    {
        P(empty);
        P(mutex);
        放一件产品;
        V(mutex);
        V(full);
    }
}
```

```

m()
{
    while(1)
    {
        P(mutex1);
        for(int i = 0; i < 10; i++)
        {
            P(full);
            P(mutex);
            取一件物品;
            V(mutex);
            V(empty);
        }
        V(mutex1);
    }
}

```

```

n()
{
    while()
    {
        P(mutex1);
        for(int i = 0; i < 10; i++)
        {
            P(full);
            P(mutex);
            取一件物品;
            V(mutex);
            V(empty);
        }
        V(mutex1);
    }
}

```

- 上述代码中，`m` 和 `n`，`i` 和 `j` 的代码一样，所以合并 `m` 和 `n`，`i` 和 `j`。