

1、N 个缓冲区的计算打印问题

1) 定性: 互斥+同步

2) 角色 (进程): 计算、打印

3) 信号量:

互斥信号量:

mutex=1 //临界区互斥信号量

同步信号量:

empty=n //空闲缓冲区数, 初始化为 n

full=0 //缓冲区占用数, 初始化为空

4) 主程序

mutex=1; //临界区互斥信号量

empty=n; //空闲缓冲区数, 初始化为 n

full=0; //缓冲区占用数, 初始化为空

CP() //计算进程

```
{
    while( 1 )
    {
        得到一个计算结果;
        P(empty); //空缓冲区减一
        P(mutex);
        将数据放入缓冲区;
        V(mutex);
        V(full); //缓冲区占用数加一
    }
}
```

IOP() //打印进程

```
{
    while( 1 )
    {
        P(full); //缓冲区占用数减一
        P(mutex);
        从缓冲区取出一个数据;
        V(mutex);
        V(empty); //空缓冲区加一
        打印输出;
    }
}
```

Main() //主程序

```
{
    CP();
    IOP();
}
```

2、盘子容量为 N 的吃水果问题

- 1) 定性: 互斥+同步
- 2) 角色 (进程): 父亲、儿子、女儿
- 3) 信号量:

互斥信号量:

mutex=1 //临界区盘子互斥信号量

同步信号量:

empty=n //可以放水果个数, 初始化为 n

So=0 //盘子内橘子个数, 初始化为空 0

Sa=0 //盘子内苹果个数, 初始化为 0

4) 主程序

mutex=1; //临界区盘子互斥信号量

empty=n; //可以放水果个数, 初始化为 n

So=0; //盘子内橘子个数, 初始化为空 0

Sa=0; //盘子内苹果个数, 初始化为 0

Father() //父亲进程

```
{ while(1)
{
    P(empty); //可放水果数减一
    P(mutex);
    放入水果;
    V(mutex);
    If (是橘子) V(So); //盘内橘子数加一
    else V(Sa); //盘内苹果数加一
}
```

Son() //儿子吃橘子进程

```
{ while(1)
{
    P(So); //盘内橘子数减一
    P(mutex);
    从盘子取出一个橘子;
    V(mutex);
    V(empty); //可放水果数加一
    吃橘子;
}
```

Daughter() //女儿吃苹果进程

```
{ while(1)
{
    P(Sa); //盘内苹果数减一
    P(mutex);
    从盘子取出一个苹果;
```

```
        V(mutex);
        V(empty); //可放水果数加一
        吃苹果;
    }
}
Main() //主程序
{
    Father();
    Son();
    Daughter();
}
```