

Chap 6 进程同步习题





进程同步问题

- 例1：桌上有一空盘，允许存放一个水果。爸爸可向盘中放苹果(Apple)，也可向盘中放橙子(Orange)，儿子专等吃盘中的橙子，女儿专等吃盘中的苹果。规定当盘空时一次只能放一个水果供儿子或女儿取用，请用P、V原语实现爸爸、儿子、女儿3个并发进程的同步。
- 分析：
 - 爸爸、儿子、女儿共用一个盘子，且盘中一次只能放一个水果
 - 当盘子为空时，爸爸可将一个水果放入盘中。若放入盘中的是橙子，则允许儿子吃，女儿必须等待；若放入盘中的是苹果，则允许女儿吃，儿子必须等待。
 - 生产者——消费者问题的一种变形。这里，生产者放入缓冲区的产品有两类，消费者也有两类，每类消费者只消费其中固定的一类产品。





```
int S=1; int SA=0; int SO=0;
main()
{
    cobegin
        father();
        son();
        daughter();
    coend
}
```

```
father()
{
    while(1)
    {
        wait(S); // 盘子是否空
        将水果放入盘中;
        if(放入的是橙子)
            signal(SO);
        else
            signal(SA);
    }
}
```

```
son()
{
    while(1)
    {
        wait(SO); // 盘子中有无橙子
        从盘中取出橙子;
        signal(S);
        吃橙子;
    }
}
```

```
daughter()
{
    while(1)
    {
        wait(SA); // 盘子中有无苹果
        从盘中取出苹果;
        signal(S);
        吃苹果;
    }
}
```





进程同步问题

例2： 桌子上有一个盘子, 每次只能放一个水果. 爸爸专门向盘子里放苹果, 妈妈专门向盘子里放橘子, 一个儿子专门等吃盘子中的橘子, 一个女儿专门等吃盘子中的苹果. 用信号量实现他们间的同步机制.

分析:

1: 盘子是临界资源. 要实施互斥. 用信号量 f 表示.

2: 苹果用 a 表示. 橘子用 o 表示.

3: 盘子的状态(空, 或者满) 用 e 表示.

semaphore: $f=1, e=1, a=0, o=0$;

```
ProcessDad() {  
    wait(e); // 盘子中是否空  
    wait(f); // 是否有人在用盘子, f 保证盘子被独占。  
    setApple();  
    signal(a); // 通知放了一个苹果  
    signal(f);  
}
```

```
ProcessMom() {  
    wait(e);  
    wait(f);  
    setOrange();  
    signal(o);  
    signal(f);  
}
```

```
ProcessSon() {  
    wait(o); // 是否有橘子  
    wait(f); // 是否有人在用盘子  
    eatOrange();  
    signal(e); // 通知盘子为空  
    signal(f);  
}
```

```
ProcessDaughter() {  
    wait(a);  
    wait(f);  
    eatApple();  
    signal(e);  
    signal(f);  
}
```





进程同步问题

- 例3: 桌上有一空盘, 允许存放一个水果。爸爸可向盘中放苹果(Apple)。儿子和女儿各吃一半(不能一人吃全部)。请用P、V原语实现爸爸、儿子、女儿3个并发进程的同步。
- 分析:
 - 1: 盘子是临界资源. 要实施互斥. 用信号量 f 表示.
 - 2: 苹果用 A 表示. AL 左半个, AR 右半个
 - 3: 盘子的状态(空, 或者满) 用 e 表示.





进程同步问题

■ 分析:

- 1: 盘子是临界资源. 要实施互斥. 用信号量 f 表示.
- 2: 苹果用 A 表示. AL 左半个, AR 右半个
- 3: 盘子的状态(空, 或者满) 用 e 表示.

```
f=1; e=1; AL=0; AR=0;
father()
{
    while(1)
    {
        wait(e); // 盘子是否空
        wait(f); // 是否有人在用盘子
        setApple();
        signal(AL);
        signal(AR);
        signal(f);
    }
}
```

```
Son() {
    wait(AL); // 是否有左半个苹果
    wait(f);   // 是否有人在用盘子
    eatApple();
    if(AL==0 and AR==0)
        signal(e); // 通知盘子为空
        signal(f);
}

Daughter() {
    wait(AR);
    wait(f);
    eatApple();
    if(AL==0 and AR==0)
        signal(e);
    signal(f);
}
```





进程同步问题

- 桌上有一空盘，允许存放2个不同水果（不允许存放2个相同水果）。爸爸可向盘中放苹果(Apple)，妈妈可向盘中放橙子(Orange)。等盘子满后，儿子吃盘中的半个橙子和半个苹果，女儿吃盘中的半个橙子和半个苹果。不允许儿子或女儿一人吃掉全部水果。用P、V原语实现爸爸、妈妈、儿子、女儿4个并发进程的同步。

