

进程同步的算法题解法

1、明确三个基本概念

临界资源

同步

互斥

2、熟记利用信号量实现同步、互斥和控制资源访问的经典范式

3、熟悉课本上的经典同步问题的算法实现

4、分析进程同步和互斥问题的方法步骤

1) 关系分析。找出问题中的进程数，并且分析他们之间的同步和互斥关系。同步、互斥关系直接按照经典范式改写

2) 整理思路。找出解决问题的关键点，并且根据课本上的经典题目找出解决思路。根据进程的操作流程确定 **wait** 操作，**signal** 操作的大致顺序。

3) 设置信号量。根据上面两步，设置需要的信号量，确定初值，完善整理。

例 1

问题描述：桌子上有一只盘子，每次只能向其中放入一个水果。爸爸专向盘子中放苹果，妈妈专向盘子中放橘子，儿子专等吃盘子中的橘子，女儿专等吃盘子中的苹果。只有盘子为空时，爸爸或妈妈可向盘子中放一个水果；仅当盘子中有自己需要的水果时，儿子或女儿可以从盘子中取出水果来吃。

<pre>Semaphore plate=1,apple=0,orange=0; Dad(){ While(1){ Prepare an apple; Wait(plate); Put the apple on the plate; Signal(apple); } } Mom(){ While(1){ Prepare an orange; Wait(plate); Put the orange on the plate; Signal(orange); } }</pre>	<pre>Son(){ While(1){ Wait(orange); Take an orange from the plate; Signal(plate); Eat the orange; } } daughter(){ While(1){ Wait(apple); Take an apple from the plate; Signal(plate); Eat the apple; } }</pre>
---	--

例 2

问题描述：假设有三个抽烟者和一个供应者。每个抽烟者不停地卷烟并抽掉它，但是要卷起并抽掉一支烟，抽烟者需要三种材料：烟草、纸和胶水。三个抽烟者中，第一个拥有烟草、第二个拥有纸、第三个拥有胶水。供应者无限地提供这三种材料，每次将两种不同的材料放到桌子上，拥有剩下那种材料的抽烟者将会卷一根烟并抽掉它，并给供应者一个信号告诉完成了，供应者就会继续放两种材料在桌上，以让一位抽烟者抽烟，这种过程一直重复下去。

```

Int random;    //存储随机数
Semaphore offer1=0;    //offer1 对应烟草和纸组合的资源
Semaphore offer2=0;    //offer2 对应烟草和胶水组合的资源
Semaphore offer3=0;    //offer3 对应纸和胶水组合的资源
Semaphore finish=0;    // 表示抽烟是否完成

```

<pre> Offer(){ //提供者 While(1){ Random=随机生成的 一个整数; Random=random%3; If(random==0) Signal(offer1); If(random==1) Signal(offer2); If(random==2) Signal(offer3); Wait(finish); } } </pre>	<pre> tobacco (){ //拥有烟草的吸烟者 while(1){ wait(offer3); 拿纸和胶水, 卷 成烟, 抽掉; Signal(finish); } } paper (){ //拥有纸的吸烟者 while(1){ wait(offer2); 拿烟草和胶水, 卷成烟, 抽掉; Signal(finish); } } </pre>	<pre> glue (){ //拥有胶水的吸烟者 while(1){ wait(offer1); 拿纸和烟草, 卷 成烟, 抽掉; Signal(finish); } } </pre>
--	--	---

某寺庙，有小和尚，老和尚若干，有一水缸，由小和尚提入水缸供老和尚饮用，。水缸可容纳 10 桶水，水取自同一井中。水井窄，每次只能容一个桶取水，水桶总数为 3 个，每次入缸取水仅为 1 桶，且不可同时进行，试着给出有关从缸取水，入水的算法描述。

解答：

从题中可以看出水井和水缸都是临界资源，引入 **well**, **vat**；三个水桶无论从井中取水还是放入水缸中都只能是一次一个，给它们一个信号量 **pail** 表示空的水桶的数量，抢不到水桶的继续等待，水缸满时，不可以继续放水，设置一个信号量 **empty** 表示水缸现在还能容纳多少水量，水缸空时，不可以继续取水，设置一个信号量 **full** 表示水缸现在的容量。

本题设置 5 个信号量：

```

Semaphore well=1; //互斥信号量，表示水井互斥使用
Semaphore vat = 1; //互斥信号量，表示水缸互斥使用
Semaphore pail = 3; //表示能用的水桶数量，初始值为 3
Semaphore empty = 10; //表示水缸现在还能容纳多少水，初始值为 10，水缸初始值为空
Semaphore full = 0; //表示水缸现在的容量，初始值为 0，水缸初始值为空

```

<pre>//老和尚 While(1){ P(full) //申请空水缸里的水操作，有则继续， 没有等待 P(pail) //申请空桶操作，有则继续，没有等 待 P(vat); //占用水缸互斥资源 从水缸中取一桶水； V(vat); //取水后，放弃水缸这个互斥资源， V(empty); //取水后，水缸的剩余空间+1 喝水； V(pail); //用完桶后，空桶资源+1 }</pre>	<pre>//小和尚 While(1){ P(empty); //观察水缸是否有剩余空间，有的 话，才能去打水 P(pail); //申请空桶资源 P(well); //打水的时候，水井互斥使用 打水； V(well); //释放水井互斥资源 P(vat); //把水放进水缸，申请，水缸互斥资 源 倒水； V(vat); //倒完水后，释放水缸互斥资源 V(full); //倒完水后，水缸先有的水容量+1 V(pail); //空桶资源+1 }</pre>
--	---