进程的同步与互斥问题总结

韩芳溪
山东大学计算机科学与技术学院
hfx@sdu.edu.cn

该类问题的分析、解决过程方法：

一、对问题进行分析，看能否抽象成三类经典问题、吸烟者问题、睡眠理发师问题、前趋图，或它们的变种，是，则套用相应的解决方法；

二、若不能，或抽象有困难，或可能是几个问题的组合，则按下述思路解决：

1．不要急于定义信号量的个数及初值；（但一定要定义）

2．分析每个进程所完成的几个主要事件，分别在相应的进程中罗列出来；

3．对进程之间的这些时间进行分析，看哪些事件之间是互相制约的；

4．添加相应的 WAIT、signal 操作

（1）在各制约的事件之前添加一个 wait() 操作；－－简称为 wait 事件；

（2）在引发 wait 事件发生事件之后，或 wait 事件所等待的事件之后添加一个 signal()操作；－－简称 signal 事件；

注意：wait、signal 操作必须成对出现；

5．确定信号量的个数

信号量个数的确定按下原则进行：

先设定一个信号量，检查是否满足要求，如果不满足，再加一个，依次类推，直至满足要求为止；

注意信号量与 wait、signal 操作的配合：

相关联的 wait 事件与 signal 事件公用一个信号量（可能在同一个进程中，也可能在不同的进程中）

6．确定信号量的初值

原则：依据 wait 操作或 wait 事件确定信号量的初值；

对每一个进程 Pi，假定其它相关进程从来没有运行，进程 Pi 的第一次运行时，wait 事件是否能够执行，若不能执行，则 wait 操作对应的信号量的初值应该为 0；否则，至少为 1（可能大于 1）；

最后，算法描述：依据分析阶段确定的信号量的个数及初值，以及算法的主体，添加头尾，写出算法。

1、wait 与 signal 为什么要设计成原语？

2、一个输入进程向一个缓冲区中输入数据，另一个输出进程从缓冲区中取出数据输出。缓冲区中每次只能存放一个数。

3、三个进程共享一个缓冲区。一个计算进程送数；一个加工进程取出加工，然后将加工结果再送回缓冲区；一个输出进程将加工后的数据取出打印。缓冲区中每次只能存放一个数。

4、三个进程共享一个缓冲区。一个负责向缓冲区送数；一个取偶数输出，另一个取奇数输出。缓冲区中每次只能存放一个数。

5、四个进程共享一个缓冲区，一个送偶数，一个送奇数，一个取偶数，一个取奇数。缓冲区中每次只能存放一个数。

5’、幸福家庭问题

桌子上有一只盘子，每次只能放入一个水果。爸爸专向盘中放苹果，妈妈专项盘中放桔子，一个女儿专等吃盘中的苹果，一个儿子专等吃盘中的桔子。试用 P、V 操作写出他们能同步的程序。

6、围棋拣子问题：数量相等的黑子与白子混在一起，利用两个进程分开。一个进程拣白子，另一个进程拣黑子。要求：

（1）一个进程拣了一个子，必须让另一个进程拣子；即两个进程应交替拣子；

（2）假定先拣黑子。

7、要求下列四条语句正确执行

s1：a:=x+y;

s2：b:=z+1;

s3：c:=a-b;

s4：w:=c+1;

将其抽象成前趋图，然后解决；

该问题也可以衍生出四个进程之间的相互制约。（*举例三个进程之间的相互制约*）

若以线段表示进程，转换成前趋图的形式。

8、有一个仓库，可以存放 X 与 Y 两种产品，仓库的存储空间足够大，但要求：

（1）每次只能存入一种产品（X 或 Y）；

（2）-N<A 产品数量 – B 产品数量<M；

其中，N 和 M 是正整数。试用“存放 A”和“存放 B”和 wait、signal 描述产品 A 与产品 B 的入库过程。

9、进程 A1、A2，……，An1 通过 m 个缓冲区向进程 B1，B2，……，Bn2 不断地发送消息。发送和接收工作遵循如下规则：

（1）每个进程发送一个消息，写入一个缓冲区，缓冲区大小与消息长度一样；

（2）对每一个消息，B1，B2，……，Bn2 都需各接收一次，读入各自的数据区中；

（3）m 个缓冲区都满时，发送进程等待，没有可读取的消息时，接收进程等待。

试用 wait 与 signal 操作组织正确的发送和接收操作。

10、有一个仓库存放两种零件 A 和 B，最大库容为各为 m 个。有一个车间不断地取 A 和 B 进行装配，每次各取一个。为避免零件锈蚀，*遵循先入库者先出库的原则*。有两组供应商分别不断地供应 A 和 B（每次一个）。为保证齐套和合理库存，当某种零件的数量比另一种的数量超过 n（n<m）个时，暂停对数量大的零件的进货，集中补充数量少的零件。试用 wait 和 signal 正确实现之。

11、某高校计算机系开设网络课并安排上机实习。假定机房共有 2m 台机器，有 2n 个学生选该课，规定：

（1）每两个学生组成一组，各占一台机器，协同完成上机实习；

（2）只有一组两个学生到齐，并且此时机房有空闲机器时，该组学生才能进入机房；

（3）上机实习由一名教师检查，检查完毕，一组学生同时离开机房。

试用 wait 和 signal 正确实现之。

12、对于读者写者问题，

（1）说明进程间的相互制约关系，应设哪些信号量？

（2）用 wait 和 signal 写出其同步算法。

（3）修改上述算法，使它对写者优先，即一旦有写者到达，后续的读者都必须等待，而无论是否有读者在读文件。

13、司机与售票员问题

在公共汽车上，司机和售票员的工作流程如下所示。为保证乘客安全，司机和售票员应密切配合协调工作。请用 wait、signal 操作来实现司机与售票员之间的同步。

司机－>（loop）{ 启动车辆－>正常行车－>到站停车 }

售票员－>（loop）{ 上乘客－>关车门－> 售票－> 开车门－>下乘客 }

14、汽车过桥问题

15、考虑一个无限长的消息队列的同步问题；

16、某数据采集与处理系统由一个数据采集进程与一个数据处理进程组成，它们共享一个缓冲区，

（1）描述两进程之间的制约关系；

（2）请利用记录型信号量机制和 wait、signal 操作解决这两个进程的同步问题，写出相应的算法描述；

17、某媒体播放器由一组循环使用的缓冲区及两个并发的播放进程与接收进程组成，其中，

(1) 8 个缓冲区构成一个循环链表，用于缓存要播放的媒体流；

(2) 接收进程负责从服务器端接收欲播放的媒体流，并依次放入缓冲区中；

(3) 播放进程依次从缓冲区中取出媒体流播放；

请利用信号量机制和 wait、signal 操作解决这两个进程的同步问题，写出相应的算法描述；

18、为某临界区设置一把锁 W，当 W＝1 时表示关锁，当 W＝0 时表示锁已经打开。试写出开锁原语与关锁原语，并利用他们实现互斥。

19、在生产者－消费者问题中，交换两个 signal 操作测次序会出现什么结果？交换两个 signal 操作呢？说明理由。

20、设有三个进程 A、B、C，其中 A 与 B 构成一对 P－C 问题，共享一个由 n 个缓冲区组成的缓冲池；B 与 C 构成一对 P－C 问题，共享一个由 m 个缓冲区组成的缓冲池。试用记录型信号量机制及 wait 与 signal 操作实现他们的同步。

21、有一阅览室，共有 100 个座位。读者进入时必须先在一张登记表上登记，该表为每一作为列一个目录，包括座号与读者姓名。读者离开时要销掉登记内容。试用记录型信号量机制及 wait 与 signal 操作描述读者之间的同步。

22. 设自行车生产线上有一只箱子, 其中有 N 个位置(N≥3), 每个位置可存放一个车架或一个车轮; 又设有三个工人, 其活动分别为:

工人 1 活动：	工人 2 活动：	工人 3 活动：
do｛	do｛	do｛
加工一个车架;	加工一个车轮;	箱中取一车架;
车架放入箱中;	车轮放入箱中;	箱中取二车轮;
}while(1)	}while(1)	组装为一台车;
		}while(1)

试分别用信号量与 Wait、signal 操作实现三个工人的合作，要求解中不含死锁.

23. 一座小桥(最多只能承重两个人)横跨南北两岸，任意时刻同一方向只允许一人过桥，南侧桥段和北侧桥段较窄只能通过一人，桥中央一处宽敞，允许两个人通过或歇息. 试用信号灯和 PV 操作写出南、北两岸过桥的同步算法.

24. 某寺庙，有小和尚、老和尚若干。庙内有一水缸，由小和尚提水入缸，供老和尚饮用。水缸可容纳 30 桶水，每次入水、取水仅为 1 桶，不可同时进行。水取自同一井中，水井径窄，每次只能容纳一个水桶取水。现有水桶 5 个，供小和尚入水及老和尚取水使用。规定入水、取水后，把桶放下，使用时再重新取。试用记录型信号量和 wait、signal 操作给出老和尚和小和尚的活动。

25、Sleeping Barber Problem

26、Cigarette Smoker’s Problem

25、The Sleeping Barber Problem

常量：CHAIRS //椅子的个数

变量：int waiting=0; //记录等待服务的顾客数

信号量：

customers=0; //等待的顾客数

barber=0; //理发师是否准备好进行服务

mutex=1; //互斥信号量（实现 waiting 的互斥访问）

```
Barber:
  while (true)  {
      wait(customers);   //如果没有顾客，睡眠
      wait(mutext);      //进入临界区，获得 waiting 的访问权
      waiting=waiting-1;
      signal(barber);    //理发师准备好可以服务
      signal(mutex);     //释放 waiting 的访问权
      cut-hair;          //理发
      }

Customer:
```

```
wait(mutex);
if (waiting < CHAIRS)  {     //如果没有空闲位，离开
    waiting=waiting +1;
    signal(customers);        如果需要的话唤醒理发师
    signal(mutex);
    wait(barber);   //如果理发师没有准备好，睡眠
    get_hair cut;     //坐下，接受服务
} else                    //  理发店已满，离开
{
   signal(mutex);
   leaving;
}
```

26、The Cigarette's Problem

```
semaphore tobacco_paper          = 0 // waiting for tobacco and paper
semaphore tobacco_matches        = 0 // waiting for tobacco and matches
semaphore paper_matches          = 0 // waiting for paper and matches
```

agent:

```
while( true )
        {
        pick a random number from 1-3
          {
          if random number is 1
                {
                        // put these two    ingredient on table
                        signal( tobacco_paper )
                }
          else if random number is 2
                {
                        // put these two    ingredient on table
                        signal( tobacco_matches )    // put on table
                }
          else if random number is 3
                {
                        // put these two    ingredient on table
                        signal( paper_matches )    // put on table
                }
                    }

wait( doneSmoking )
```

```
    } /* while */

smokers:
    // the smoker that has matches
    while( true )
            {
            wait( tobacco_paper ); /* picks up tobacco and paper */

            // roll cigarette and smoke

            signal( doneSmoking );
            }

    // the smoker that has paper
    while( true )
            {
            wait(tobacco_matches ) ; /* picks up tobacco and match */

            // roll cigarette and smoke

            signal( doneSmoking );
            }

    // the smoker that has tobacco
    while( true )
            {
            wait(paper_matches ) ; /* picks up paper and match */

            // roll cigarette and smoke

            signal( doneSmoking );
            }
```