

带标记信号量——一种新型同步与互斥机制

Tagged Semaphores: A New Mechanism for Synchronization and Mutual Exclusion

徐宝文

(东南大学计算机科学与工程系 南京 210096)

Abstract This paper introduces a new kind of semaphores——tagged semaphores. Tagged semaphores are the generalization of standard semaphores and have more powerful capability than the latter.

Keywords Concurrent programming, Semaphore, Tagged semaphore, Mutual exclusion, Synchronization

1 问题的引出

信号量机制是 E. W. Dijkstra 最先提出来的^[1],是并发程序设计中一种基本的、重要的同步与互斥原语,也是研究并发程序行为的重要机制^[2,3]。其描述能力强、机理简单、易于掌握,能方便地描述处理绝大多数并发程序设计的同步与互斥问题(如共享变量的保护)。信号量原语自问世二十余年来,已广泛地应用于许多操作系统与程序设计语言的研究与设计中,成了它们不可分割的一部分。信号量虽然不为一些语言设计者所欣赏,但它仍然得到了很广泛的应用,是研究并发程序同步与互斥的必不可少的工具。尽管如此,也有一些问题用信号量描述起来比较困难或比较复杂。下面是一个典型的入厕问题:

设某地有一公共厕所,它有 N 个蹲位,但没有将

男女蹲位分隔开。因此,当厕所里有人在入厕时,即使还有空位异性也不能进去(可用一块牌子来标示)。但是,只要 N 个蹲坑未被全部占满,与已在入厕者同性别的人就可以进去;而当厕所中无人入厕时,任何人都可以进去。

这一问题虽然比较特殊,但也有一定的普遍性,这类例子也比较多。例如,如果一旅馆每个客房可住多个旅客,那么在为旅客安排客房时就会遇到这种互斥问题。如果一饭店比较大,可同时分批接待若干个代表团或会议就餐,那么在用餐标准相同的情况下,服务员在安排餐桌时可采用如下策略:对每一张餐桌在无人就座时,哪个人先入座就把这张餐桌安排给该人所属的代表团或会议,其他人不得入座(即使餐桌没有坐满)。只有当该桌人都就餐完毕退出以后,其他尚未就餐人员才能再按代表团或会议入座就餐。再如,一排球场的

徐宝文 教授,博士生导师,主要从事程序设计语言,软件工程,并行程序设计等方面的教学与科研工作。

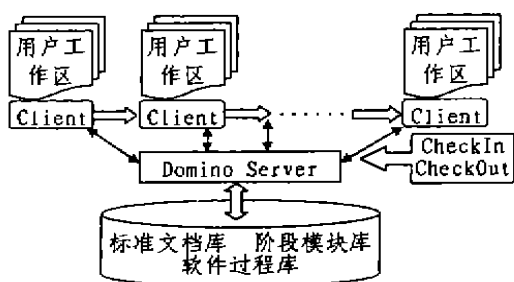


图5 软件质量保障框架总体结构图

参考文献

- 1 Paulk M C, et al. Key Practice of the Capability Maturity

Model, Version 1.1: [CMU/SEI-93-TR-24]. Feb. 1993

2 Barbacci M, et al. Quality Attributes. Dec. 1995

3 Paulk M C. the Rational Planning of (Software) Projects In: the 1st World Congress for SQ, section 4, June 1995

4 张路,李欣,梅宏,杨美清. 基于复用的软件开发过程中的配置管理. 北京大学计算机科学技术系

5 Feiler P H. Software Configuration Management: Advance in Software Development Enviroments. Mar. 1990

6 Goldenson D R, Herbsleb J D. After the Appraisal: A Systematic Survey of Process Improvement, its Benefits, and Factors that Influence Success. Aug. 1995

7 Paulk M C. Effective CMM-Based Process Improvement. In: Proc. of the 6th Intl. Conf. on SQ, Oct. 1996. 226~237

8 谢冰,张路,钟林辉. 软件配置管理系统 JBCM. 北大软件工程研究所

出租打球时制订有如下规则:所有入场人员都必须同一单位(同一班级或协会等)的且场内总人数有一定限制,如有人要进去则必须先有人出来。只有等场内所有人都出来后其它单位的人才能进去。这几个例子说明的都是同一性质的问题:同一类若干个资源在任一时刻只能分配给具同一性质(同一性别、同一代表团、同一单位等)的若干进程。下面我们先以入厕问题为例用普通信号量来解决这类问题,然后再给出一种新的信号量机制——带标记信号量。

2 普通信号量解法及其存在的问题

对上述入厕问题,在处理时要作两种互斥:一是异性互斥,二是同一性别的人入厕时因蹲位限制而要作的互斥。若用普通信号量解决,则需要使用两个信号量:一个是用于控制当前可用蹲位数目的多值信号量 Sem(初值为总可用蹲位数 N),另一个是用于封锁有关操作的 P-V 操作二值信号量 Lock。(初值为真。之所以要封锁是因为这些操作是不可分的)。当有人要入厕时,首先要判别一下厕所中是否有人,若无人,则任何人都可入内,此时要置入厕者的性别标记;如有人且还有空蹲位,则只有与已在厕所同性别的人才可入厕;如不是这两种情况,则要入厕者等待。即要进行如下操作(我们用 Tag 来标记已在入厕者的性别):

```
P(Lock);
if Sem=N then
    Wait(Sem);——任何人都可入厕
    Tag:=所要入厕者的性别;
elseif (Sem>0)and(Tag=所要入厕者的性别)then
    Wait(Sem);——只有与正入厕者同性别的人才可入厕
else
    所要入厕者(挂起)排队等待;
end if;
V(Lock);
```

当有人入厕完出来时,要进行如下操作:首先要判别一下是否有人在等待,如无人等待,则释放该蹲位;否则,要分两种情况分别处理。如果该人出来时厕所中还有人,那么这时一定有 $Sem > 0$,可让正在等待的同性别的一人入厕;否则,这时可能有若干异性正在等待,可先让其中一人入内并置其性别标记,然后再让其他人入内,直到所有蹲位都已占满。该过程可描述如下:

```
P(Lock);
Signal(Sem);
if 相应等待队列非空 then
    if Sem=N then
        等待队列中选任一人进入入厕;
        Sem:=Sem-1;
        Tag:=所要入厕者的性别;
        while (Sem>0) and (等待队列中还有人的性别与
            Tag 相同) loop
            该人进入入厕;——其他与正入厕者同性别的人
            可同时入厕
            Sem:=Sem-1;
        end loop;
    elseif 等待队列中有人性别=Tag then——此时
```

```
Sem=1
该人进去入厕;——只有与正入厕者同性别的人才可
入厕
Sem:=Sem-1;
end if;
end if;
V(Lock);
```

在这两段程序中共调用了四个操作,P 与 V 操作用于完成 P-V 操作的相应功能,Wait 与 Signal 操作是多值信号操作,分别用于申请与释放资源(注意,在任何时候都有 $Sem \geq 0$,因此,按通常对 Signal 的定义,调用它的效果只是将 Sem 的值加一,即它不具有使等待队列中的待入厕者入厕(“唤醒”)的功能)。

我们之所以不允许 Sem 取负值以表示正挂在等待队列中的进程(待入厕者)的数目,是因为由于两种互斥的存在,即使还有资源(蹲位)可用也可能有异性进程处于等待状态。由于并不是在 $Sem > 0$ 时就可以用 Wait 操作来使要入厕者入厕,因此调用该操作是有条件的;同样,由于这里所用 Signal 操作不具备“唤醒”等待进程的能力,故在它释放资源后还要分情况处理等待进程。

换言之,由于在申请资源时除了要考虑有无资源可用外还要考虑申请者的身份,在释放资源时也不是只要有申请者在等待就可以把正释放的资源分配给它,因此这里没有完整地使用 Wait 与 Signal 两个原语的全部功能,而且信号变量 Sem 只反映了资源分配情况并没有反映出进程调用(包括等待)的具体情况。更为严重的是,在用于释放资源的程序段中没有通过信号原语就直接更新了信号变量 Sem 的值,这破坏了信号量操作的完整性与一致性,使得申请与释放资源的操作变得很复杂、很容易出错。

3 带标记信号量

为了处理上述这类问题,我们对信号量机制进行拓广,引入带标记信号量的概念。带标记信号量(也可称之为带身份信号量)的操作原语是 Tagged-Wait 与 Tagged-Signal,它们分别对应于普通信号量中的 Wait 与 Signal 原语,也是不可分的。带标记信号量与普通信号量的不同之处在于:普通信号量只有一个分量,当它为正时表示还未被占用的资源数,当它为负时表示正在等待使用该类资源的进程数,当它为零时表示所有资源都被占用并且没有其它进程在等待该类资源;而带标记信号量 S 有三个分量,分量 S.R 表示尚可使用的资源数(初值为所有资源数 N),分量 S.P 表示正在等待使用该资源的进程数(初值为零),另一个分量 S.T 是一个标记,标记正使用这类资源的使用者的身份(性别、单位等)。当所有资源均未被使用时,该标记不起作用(可认为它有一个特殊的“空”值)。这

三个分量的值均不得在除 Tagged-Wait 与 Tagged-Signal 这两个操作原语之外的其它地方更新。

Tagged-Wait 原语用于申请资源,有两个参数:一个是值参数 Tag,表示调用进程的调用标记(不是信号量标记);另一个是变量参数 S,是所要作用的带标记信号变量,Tagged-Wait 原语的功能可描述如下

```
Tagged-Wait(Tag,S)::=
if S.R=N then
  S.R:=S.R-1;
  当前调用进程继续执行;
  S.T:=Tag;
elseif (S.R>0) and (S.T=Tag) then
  S.R:=S.R-1;
  调用进程继续执行;
else
  挂起当前调用进程到相应队列中(并相应保存 Tag);
end if;
```

Tagged-Signal 原语用于释放资源,它只有一个变量参数 S,为所要作用的带标记信号变量,该原语的功能可形式描述如下:

```
Tagged-Signal(S)::=
if S.P>0 then
  if S.R=N-1 then——该类资源只被当前进程占用了
    一个,释放完后该类资源暂无进程占用从等待队列中
    取一进程执行;
    S.T:=Tag;——将 S.T 置为刚从等待队列中取出的
    进程的 Tag 值
    while (S.R>0) and (等待队列中还有进程的 Tag 值等
      于 S.T) loop
      从等待队列中取该进程执行;
      S.R:=S.R-1;
    end loop;
    elseif (S.R>0) and (等待队列中有进程的 Tag 值等于
      S.T) then
      从等待队列中取该进程执行;
    end if;
  else
    S.R:=S.R+1;
  end if;
```

4 应用实例

利用带标记信号量可以很容易解决入厕问题:当要入厕时只要调用 Tagged-Wait(Sex, Sem) 原语,这里 Sex 是要入厕者的性别, Sem 是表示厕所蹲位占用情况的带标记信号量;当入厕完时只要调用 Tagged-Signal(Sem) 原语即可。

下面以著名的生产者-消费者进程的例子来说明带标记信号量的用途与用法。设某厂只有一仓库供各车间存放产品,假定各车间所生产的产品有多个品种(有些车间生产的产品的有些品种可能相同),要求仓库中不得同时存放两种产品但可同时存放两个车间的同一种产品,并且每次只能往仓库中存一件产品。这样,当仓库里存放有某种产品时,只要仓库不满,生产同一种产品的车间就可以往仓库中存放产品,而生产其它产品的车间就得等待仓库中的产品清空。这样,一个典型的车间在生产某一产品时的典型运作过程(用 Producer 表示)就如下所示:

```
process Producer is
begin
  loop
    生产一件产品 Name;
    Tagged-Wait(Name, Null-NO);
    P(Mut-Ex);
    将该件产品入库;
    V(Mut-Ex);
    Signal(Full-NO);
  end loop;
end Producer.
```

其中, Name 是产品名称(标记), Mut-Ex 是一用于互斥的二值信号变量(限定同一时刻最多只能有一进程在“访问”仓库), Full-NO 是一普通多值信号变量(表示仓库中已存放产品数), Null-NO 是一多值带标记信号量(表示仓库中还可存放的产品数),这三个信号变量的初值分别为真,零与仓库的容量。

再假定有若干“消费者”在不停地从仓库中取出产品进行“消费”(包括转运、销售),他们是仓库里有什么产品就取什么产品,每当取走一件产品时就发一信号通知“管理者”仓库中又空出一个位置。其运作过程如下:

```
process Consumer is
begin
  loop
    Wait(Full-NO);
    P(Mut-EX);
    从仓库中取出一件产品 Product;
    V(Mut-Ex);
    Tagged-Signal(Null-NO);
    “消费”这件产品 Product;
  end loop;
end Consumer;
```

结论 本文尽管是以资源分配为例来引入并讨论带标记信号量,但这种信号量也可用于进程的同步与互斥。当 S.R 的初值为 1 时,带标记信号量就退化成了二值互斥信号量(这时其标记分量不起作用);对一带标记信号量,当对它的使用都用一特定“空”标记值调用 Tagged-Wait 原语时,该信号量就退化成普通多值信号量;当我们要求只有具同一“性质”的进程才能同时并行执行时,也可以使用带标记信号量(这时 S.R 表示的不再是可用资源数,而是还可加入并行执行的进程数)。由此可见,带标记信号量是普通信号量的一般化,普通信号量则是带标记信号量的特例,而带标记信号量的描述能力则比普通信号量强得多。如果把带标记信号量的思想运用到管程、消息传送等机制中,可大大提高它们的功能与描述能力。

参考文献

- 1 Dijkstra E W. The structure of the "THE" multiprogramming system. Comm of ACM, 1968, 11(5): 341~346
- 2 Andrews G R. Concurrent Programming: Principles and Practices. The Benjamin/Cummings Publishing Company, Inc., 1991
- 3 徐宝文. 高级程序设计语言原理. 北京. 航空工业出版社, 1992