

进程互斥与同步

——浅谈信号量的设置与使用

赵素萍

(洛阳师范学院计算机科学系, 河南 洛阳 471002)

【摘要】 进程的同步与互斥是操作系统的重要内容, 合理运用 P、V 操作可以防止进程死锁, 本文给出分析进程同步与互斥的方法及如何编制程序。

【关键词】 进程 同步 互斥 临界资源 P 操作 V 操作

多道程序系统中进程是并发执行的, 这些进程之间存在着不同的相互制约关系, 为了协调进程之间的相互制约关系, 就需要实现进程间的同步。互斥是同步的一种特殊情况。操作系统中实现进程间同步和互斥的一种常用工具是信号量。本文就如何设置信号量以及如何对其进行 P、V 操作做一简单讨论。

一、正确区分互斥与同步

信号量是一个确定的二元组 (s, q) , 其中 s 是一个具有非负初值的整型变量, q 是一个初始状态为空的队列。整型变量 s 表示系统中某类资源的数目。设置信号量, 首先必须正确区分进程间是同步关系还是互斥关系。

互斥是两个进程不能同时进入访问同一临界资源的临界区; 同步是进程未获得合作进程发来的消息之前, 该进程等待, 消息到来之后方可继续执行。无论是同步还是互斥, 都是在执行的时间顺序上对并发进程的操作加以某种限制。对于互斥的进程, 它们各自单独执行时都是正确的; 同步的进程, 它们各自单独执行会产生错误, 必须互相配合共同推进, 也就是说, 同步条件高。同步一方面要互斥进入临界区使用临界资源, 另一方面进入临界区有先后顺序

如: 进程 P、Q 共享一台磁带机, 磁带机任一时刻只能被一个进程所使用, 而不能同时使用, P、Q 单独执行不会发生错误, 故 P、Q 是互斥进程。再如: P 进程往缓冲区送数, Q 进程从缓冲区取数, 如果 P、Q 单独执行就可能发生错误: 往满缓冲区送数或从空缓冲区取数, 故 P、Q 是同步进程。

二、用信号量实现进程互斥、同步

一个信号量的建立必须经过说明, 即应该准确说明 s 的意义和初值, 信号量的值仅能由 p、v 两条原语来改变。P 操作是申请资源, 它使信号量值减 1, 若结果非负, 该进程继续, 否则该进程被封锁; V 操作是释放资源, 它使信号量值增 1, 若结果大于零, 该进程继续, 否则从该信号量的等待队列中移出一个进程, 解除它的等待状态。

1. P、V 实现进程互斥

要实现进程互斥, 只需设公用信号量 S , 初值为 1, 用 $P(S)$ 、 $V(S)$ 框住临界区 CS。若 $S = 1$, 无进程进入 CS; $S = 0$, 有一个进程进入 CS; $S < 0$, 有 $|S|$ 个进程等待进入 CS。程序编制如下:

进程 A	进程 B
.....
P(S)	P(S)
CSA	CSB
V(S)	V(S)
.....

此时的信号量可看作进入临界区的通行证, P 操作申请通行证, 退出临界区的 V 操作是归还通行证。

2. P、V 实现进程同步

(1) 合作进程的次序关系(前趋关系)

方法一:

有几个前趋关系, 设几个公用信号量, 且初值为零, 表示前趋进程是否执行完。

程序编制方法如下:

有前趋又有后继的进程

$P(S_1)$

.....

$V(S_2)$

有前趋无后继的进程

$P(S_1)$

.....

无前趋有后继的进程

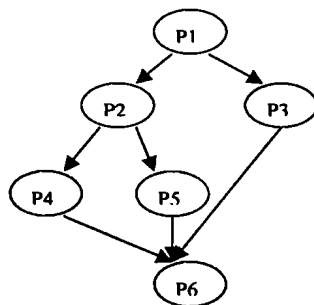
.....

$V(S_2)$

注: S_1 为前趋进程信号量, S_2 为后继进程信号量。

方法二: 在前趋图中, 有多少个没有后继的结点, 就设多少个信号量, 且初值为零, 表示进程是否完成, 此时, 一个进程如果有多个后继, 就要做多次 V 操作

例如:



解法一: 设信号量 S_{12} 、 S_{13} 、 S_{24} 、 S_{25} 、 S_{36} 、 S_{46} 、 S_{56} , 初值均为零, S_{ij} 表示第 i 个进程否完成, 第 j 个进程能否开始。程序如下:

进程 P1:	进程 P2:	进程 P3:
...	$P(S_{12})$	$P(S_{13})$
$V(S_{12})$
$V(S_{13})$	$V(S_{24})$	$V(S_{36})$
	$V(S_{25})$	

进程 P4:	进程 P5:	进程 P6:
$P(S_{24})$	$P(S_{25})$	$P(S_{36})$
...	...	$P(S_{46})$
$V(S_{46})$	$V(S_{56})$	$P(S_{56})$
		...

注解: $P(S_{ij})$ 检查第 j 个进程能否开始, $V(S_{ij})$ 通知第 j 个进程第 i 个进程已经完成。

解法二: 设信号量 s_1 、 s_2 、 s_3 、 s_4 、 s_5 , 初值均为零, 分别表示 P1、P2、P3、P4、P5 是否执行完成, 此时, 程序如下:

进程 P1:	进程 P2:	进程 P3:
...	$p(s_1)$	$p(s_1)$
$v(s_1)$
$v(s_1)$	$v(s_2)$	$v(s_3)$
	$v(s_2)$	

进程 P4:	进程 P5:	进程 P6:
$p(s_2)$	$p(s_2)$	$P(s_3)$
...	...	$P(s_4)$
$v(s_4)$	$v(s_5)$	$P(s_5)$
		...

注意:进程 P1、P2 分别有两个 $V(S_1)$ 、 $V(S_2)$ 操作,因为它们分别有两个后继。

(2)共享缓冲区

对于两个进程共享缓冲区,设两个私用信号量,此时,程序编制和信号量初值有关。若 $S_1 = 1, S_2 = 0$,则进程 1 为: $P(S_1) \dots V(S_2)$,进程 2 为 $P(S_2) \dots V(S_1)$,即申请自己的信号量,释放对方的信号量;若 $S_1 = 0, S_2 = 0$,则进程 1 为: $\dots V(S_2)$, $P(S_1)$,进程 2 为 $P(S_2) \dots V(S_1)$,即其中一个进程先执行,然后释放对方的信号量。

例:某计算进程 CP 和打印进程 LOP 共用一个单缓冲,其中 CP 负责不断地计算数据并送入缓冲区中,LOP 进程负责从缓冲区中取数。

解法一:设 $s_1 = 0$,表示缓冲区中无数可打印, $S_2 = 1$,表示缓冲区空,此时同步关系描述为:

CP 进程	LOP 进程
生产一个数据	$P(S_1)$
$P(S_2)$	从缓冲区取数
送入缓冲区	$V(S_2)$
$V(S_1)$	输出数据

解法二:设 $s_1 = 0$,表示缓冲区中无数可打印, $S_2 = 0$,表示缓冲区空,此时同步关系描述为:

CP 进程	LOP 进程
生产一个数据	$P(S_1)$
送入缓冲区	从缓冲区取数
$V(S_1)$	$V(S_2)$
$P(S_2)$	输出数据

三、应用举例

用 PV 操作实现进程同步与互斥时,应注意:(1)对每一个共享资源(含变量)都要设立信号量,互斥时对一个共享资源设一个信号量,同步时对一个共享资源可能要设两个或多个信号量,视有几个进程来使用该共享变量而定。(2)互斥时信号量的初值可大于或等于 1,同步时,至少有一个信号量的初值大于等于 1。(3)PV 操作一定成对调用(前趋关系除外),互斥时在临界区前后对同一信号量作 PV 操作,同步时对不同的信号量作 PV 操作。(4)对互斥和同步混合问题,PV 操作可能会嵌套,一般同步的 PV 操作在外,互斥的 PV 操作在内。大家可仔细阅读“生产者—消费者”问题以及“读者—写者”问题。下面再举两例。

例 1:(北京大学 1991 年考研试题)

有一个仓库,可以存放 A 和 B 两种产品,但要求:(1)每次只能存入一种产品(A 或 B);(2) $N - A$ 产品数量 $- B$ 产品数量 $< M$ 。其中, N 和 M 是正整数。试用 P、V 操作描述产品 A 与产品 B 的入库过程。

【分析】将放 A 产品看成是一个进程,放 B 产品看成另一个进程,这两个进程单独执行会产生错误:产品数量不符合条件(2),

故两个进程是同步的。由第二个条件知:B 产品的数量不能比 A 产品的数量多 N 个以上, A 产品的数量不能比 B 产品的数量多 M 个以上。即:每次放 A 产品时先检查 A 产品的数量是否比 B 产品的数量多 M 个以上,若是,不允许放,放 B 产品时也做检查。因此设两个同步信号量,分别表示 A 产品比产品多入库的数量和 B 产品比 A 产品多入库的数量,有条件可知,初值分别为 $M - 1$ 和 $N - 1$ 。

本题给出的第一个条件是临界资源的访问控制,设一个互斥信号量解决这个问题。

解:设互斥信号量 $mutex = 1$
同步信号量 $sa = M - 1$
 $sb = N - 1$

A 进程:	B 进程
$P(sa)$	$P(sb)$
$P(mutex)$	$P(mutex)$
将产品入库	将产品入库
$V(mutex)$	$V(mutex)$
$V(sb)$	$V(sa)$

注:A 进程中同步的 $P(sa)$ 、 $V(sb)$ 在外,互斥的 $P(mutex)$ 、 $V(mutex)$ 在内。B 进程也如此。

例 2:一个海底隧道中只有一个车道,规定同一方向的可以连续过隧道;某方向有列车过隧道时,另一个方向的列车就要等待,现在东岸和西岸都有列车要过隧道,如果把每个过隧道的列车看作一个进程,为保证安全,请用 PV 操作实现正确管理。

【分析】将从东岸向西岸行驶看成一个进程,从西岸向东岸行驶看成另一个进程,这两个进程单独执行都是正确的,故属于互斥进程,因此应定义一个互斥信号量,初值为 1。由于同方向的列车可以连续过隧道,因此该方向的第一个列车应该去争取过隧道权,其它同方向列车直接过隧道,直到该方向没有列车经过隧道才释放共享资源—隧道权的使用,为此,应该设两个共享变量,分别记录同方向的过隧道的列车数,还应该设两个互斥信号量,用于对共享变量的互斥访问。

部分程序如下:

设两个计数器 rc_1 、 rc_2 ,分别记录由东向西和由西向东的列车数,初值为零;设互斥信号量 S 、 S_1 、 S_2 ,初值均为 1,分别表示对隧道的互斥使用和两个计数器的互斥访问。

```

东岸到西岸
P(S1); /* 申请对临界资源 rc1 的访问 */
rc1 = rc1 + 1;
if rc1 == 1 P(S); /* 申请过隧道权 */
V(S1); /* 释放对临界资源的使用权 */
过隧道;
P(S1);
rc1 = rc1 - 1;
if rc1 == 0 V(S); /* 释放隧道使用权 */
V(S1);

```

西岸到东岸程序类似。

参考资料

- [1] 汤子瀛 哲风屏 汤小丹. 计算机操作系统. 西安电子科技大学出版社, 2001.
- [2] 张尧学 史美林. 计算机操作系统教程. 清华大学出版社, 1997.
- [3] 孙钟秀等. 操作系统教程. 第二版. 高等教育出版社, 2000.
- [4] 曹平等. 操作系统—习题与解析. 清华大学出版社, 2001.