

## 3.1 例题解析

**例题 1** 进程间的互斥与同步表示了各进程间的\_\_\_\_\_。

- A. 竞争与协作
- B. 相互独立与相互制约
- C. 临界区调度原则
- D. 动态性与并发性

**分析：**答案 A。当多个进程都去访问非共享资源时就会产生竞争，需要互斥执行，通过临界区加以控制，当多个进程相互协作共同完成一个任务时，需要同步相关的信息以达到合作的目的。

**例题 2** 若执行信号量 S 操作的进程数为 3，信号量 S 初值为 2，当前值为 -1，表示有\_\_\_\_\_个等待相关临界资源的进程。

- A. 0
- B. 1
- C. 2
- D. 3

**分析：**答案 B。每当一个进程申请 S 信号量时，S 的值就减 1，当 S 的值为 0 时再申请的进程就需等待，负值的绝对值就表示在临界区等待的进程数。

**例题 3** 由于并发进程执行的随机性，一个进程对另一个进程的影响是不可预测的，甚至造成结果的不正确，\_\_\_\_\_。

- A. 造成不正确的因素与时间有关
- B. 造成不正确的因素只与进程占用的处理器有关
- C. 造成不正确的因素与执行速度无关
- D. 造成不正确的因素只与外界的影响有关

**分析：**答案 A。由于各进程的异步推进，进程之间的制约关系与时间有关，也即与进程的执行速度有关。

**例题 4** 下列机构中不能用于进程间数据通信的是\_\_\_\_\_。

- A. 消息
- B. 共享存储区
- C. 信号量
- D. 管道

**分析：**答案 C。能传送大量数据的高级通信机制可归结为三大类：共享存储器系统、消息传递系统以及管道通信系统。信号量主要用于进程的同步与互斥控制，不是为了数据通信。

**例题 5** 下面有关管程的说法，不正确的是\_\_\_\_\_。

- A. 管程是一种进程同步机制
- B. 管程是一种编程语言成分
- C. 管程是一种系统调用
- D. 管程比信号量更容易保证并行编程的正确性

**分析：**答案 C。使用信号量和 PV 操作实现进程同步时，对共享资源的管理分散于各个进程中，这样不利于系统对临界资源的管理，难以防止进程有意或无意地违反同步操作，且容易造成程序设计错误。因此提出管程的概念以解决上述问题，管程实质上是把临界区集中到抽象数据类型模板中，可作为程序设计语言的一种结构成分。

**例题 6** 什么是临界资源和临界区？一个进程进入临界区的调度原则是什么？

**答：**不允许两个或两个以上进程同时访问的资源称为临界资源。进程执行访问临界资源的程序段称为临界区、临界段或互斥段。

能支持各进程互斥地执行临界区的调度机制必须满足下列要求。

- (1) 在临界区中，每次只能允许一个进程进入。
- (2) 一个进程在非临界区中的暂停运行不能影响其他进程。
- (3) 一个进程如需要进入临界区，不能发生无限延迟的情况，即既不会死锁，也不会饥饿。
- (4) 当无进程在临界区时，必须让任何希望进入该程序段的进程无延迟地进入。
- (5) 一个进程只能在临界区内停留有限的时间。
- (6) 对于相关进程的运行速度和处理器的数量不做假设。

**例题 7** 进程之间存在哪几种制约关系？各是什么原因引起的？下列活动分别属于哪种制约关系？

- (1) 图书馆借书
- (2) 两队举行篮球赛
- (3) 流水生产线
- (4) 乐队演奏
- (5) 购买火车票

**答：**有直接制约关系(同步问题)和间接制约关系(互斥问题)；同步问题是存在关系的进程之间的相互等待所产生的制约关系，互斥问题是进程间竞争使用资源所发生的制约关系。

- (1) 属于互斥关系，因为书的个数是有限的，一本书只能借给一个同学。
- (2) 既存在互斥关系，也存在同步关系。篮球只有一个，两队都要竞争；但对于同一个队的队员之间需要相互协作才有可能取得比赛的胜利。
- (3) 属于同步关系，生产线上各道工序的开始都依赖前道工序的完成。
- (4) 属于同步关系，乐队中的每个成员需要相互协作共同完成乐曲演奏任务。
- (5) 属于互斥关系，一张火车票只能卖给一个人。

**例题 8** 在生产者-消费者问题中，如果将两个 P 操作即生产者程序流程中的 P(buffers) 和 P(mutex) 互换位置，结果会如何？

**答：**P(buffers) 和 P(mutex) 互换位置后，因为 mutex 是生产者和消费者公用的信号量变量，生产者在执行完 P(mutex) 后，则 mutex 赋值为 0，倘若当前无空闲缓冲区，buffers 也为 0，在执行了 P(buffers) 后，buffers 为 -1，该生产者进程就会进入阻塞态，这样不仅其他的生产者进程会因 mutex 不能继续存放产品，并且消费者也因 mutex 不能取产品，从而无法释放缓冲区，使缓冲区始终为 0，这样就形成了死锁。

**例题 9** 试用 P、V 操作描述下列理发师和顾客之间的同步问题。

某个理发师当没有顾客时，去睡觉；当有顾客来理发，若理发师正在睡觉，这个顾客会

叫醒他,理发师给该顾客理发,理发期间若还有顾客到达则等待理发师依次理发,直到没有顾客到来,理发师又去睡觉。

**分析:**将此题看作  $N$  个生产者和一个消费者问题。顾客作为生产者,每到来一位,就应将计数器  $rc$  计数一次,以便让理发师理发至最后一位顾客,因此,顾客进程执行的第一个语句便是  $rc = rc + 1$ 。而第一个到来的顾客应负责唤醒理发师,理发师此时正在信号量  $wakeup$  上等待( $P(wakeup)$ );该信号量初值为 0,由第一个顾客执行  $V(wakeup)$ 。若该顾客不是第一个到达者,则在信号量  $wait$  上等待( $P(wait)$ );该信号量初值为 0,等到理发师给前一位顾客理完发后执行  $V(wait)$ ,便给该顾客理发。以上过程循环往复,理发师每处理完一个顾客,就令计数器  $rc$  值减 1,当  $rc = 0$  时,便知此时无顾客,理发师可继续睡觉,等待下一批顾客的到达。为了保证对计数器  $rc$  互斥使用,还需要设置信号量  $mutex$ (初值为 1)。

**答:**用 P、V 操作描述理发师和顾客之间的同步问题:

```
Semaphore wakeup, wait, mutex;
wakeup = 0; wait = 0; mutex = 1;
cobegin
```

顾客进程:

```
{
    P(mutex);
    rc = rc + 1;
    if (rc == 1)
        V(wakeup);
    else
        P(wait);
    V(mutex);
    理发;
}
```

理发师进程:

```
{
    P(wakeup);
    while (rc != 0)
    {
        理发;
        P(mutex);
        rc = rc - 1;
        if (rc != 0)
            V(wait);
        V(mutex);
    }
}
coend
```

## 3.2 课后自测题

### 一、选择题

1. 并发性是指若干事件在\_\_\_\_\_发生。

- A. 同一时刻  
B. 同一时间间隔内  
C. 不同时刻  
D. 不同时间间隔内
2. 进程间的基本关系为\_\_\_\_\_。
- A. 相互独立  
B. 同步与互斥  
C. 信息传递与信息缓冲  
D. 并行执行与资源共享
3. 操作系统中 P、V 操作是一种\_\_\_\_\_。
- A. 系统调用  
B. 进程通信原语  
C. 控制命令  
D. 软件模块
4. 两个进程合作完成一个任务,在并发执行中,一个进程要等待其合作伙伴发来信息或者建立某个条件后再向前执行,这种关系是进程间的\_\_\_\_\_关系。
- A. 同步  
B. 互斥  
C. 竞争  
D. 合作
5. 一段不能由多个进程同时执行的代码称为\_\_\_\_\_。
- A. 临界区  
B. 临界资源  
C. 锁操作  
D. 信号量操作
6. 临界区是指并发进程中\_\_\_\_\_。
- A. 用于实现进程互斥的程序段  
B. 用于实现进程同步的程序段  
C. 用于实现进程通信的程序段  
D. 与互斥的共享资源有关的程序段
7. 不能利用\_\_\_\_\_实现父子进程间的互斥。
- A. 文件  
B. 外部变量  
C. 信号量  
D. 锁
8. 解决进程间同步与互斥问题常用方法是使用\_\_\_\_\_。
- A. 锁操作  
B. 存储管理  
C. 信号机构  
D. 信号量
9. 读者、写者是一个\_\_\_\_\_问题。
- A. 互斥  
B. 半同步  
C. 全同步  
D. 共享
10. 如果系统只有一个临界资源,同时有很多进程要竞争该资源,那么系统\_\_\_\_\_发生死锁。
- A. 一定会  
B. 一定不会  
C. 不一定会  
D. 由进程数量决定是否
11. 在操作系统中,对信号量  $s$  的 P 操作定义中,使进程进入相应等待队列的条件是\_\_\_\_\_。
- A.  $s > 0$   
B.  $s = 0$   
C.  $s < 0$   
D.  $s \leq 0$
12.  $N$  个进程访问一个临界资源,则设置的互斥信号量  $s$  的取值范围是\_\_\_\_\_。
- A.  $0 \sim N-1$   
B.  $1 \sim -(N-1)$   
C.  $1 \sim N-1$   
D.  $0 \sim -1$
13. 临界区就是指\_\_\_\_\_。
- A. 一段程序  
B. 一段数据区  
C. 一个缓冲区  
D. 一个共享资源
14.  $M$  个生产者、 $N$  个消费者共享长度为  $L$  的有界缓冲区,则对缓冲区互斥操作而设置的信号量的初值应设为\_\_\_\_\_。
- A.  $L$   
B.  $M$   
C.  $N$   
D. 1
15. 对于使用一个临界资源的两个并发进程,若互斥信号量等于 1,则表示\_\_\_\_\_。
- A. 没有进程进入临界区  
B. 有一个进程进入了临界区

- C. 有一个进程进入了临界区,另一个进程等待进入  
D. 这两个进程都在等待进入临界区
16. 若信号量  $S$  的初值为 2,当前值为 -1,则表示有 \_\_\_\_\_ 个等待进程。  
A. 0                      B. 1                      C. 2                      D. 3
17. 类似于电子邮件系统的进程间的通信方法是 \_\_\_\_\_ 通信。  
A. 管道                      B. 共享存储区                      C. 信号量                      D. 消息
18. 在进程之间要传递大量的数据,效率高而且互斥与同步控制方便的方法是采用 \_\_\_\_\_。  
A. 管道                      B. 共享存储区                      C. 全局变量                      D. 信号量
19. 信箱通信是一种 \_\_\_\_\_ 通信方式。  
A. 低级                      B. 直接                      C. 间接                      D. 中级
20. 下列 \_\_\_\_\_ 不属于管程的组成部分。  
A. 对管程内数据结构进行操作的一组过程  
B. 管程外过程调用管程内数据结构的说明  
C. 管程内共享变量的说明  
D. 共享变量初始化语句序列
21. 测试并设置指令 test-and-set 是一种 \_\_\_\_\_。  
A. 锁操作指令                      B. 互斥指令                      C. 判断指令                      D. 信号量指令
22. 关于管程与进程比较的论述中,正确的是 \_\_\_\_\_。  
A. 管程内定义的是公用数据结构,进程内定义的是私有数据结构  
B. 管程作为操作系统或编程语言成分,与进程一样也具有生命周期,由创建而产生,由撤销而消亡  
C. 管程能被系统中所有的进程调用  
D. 管程和调用它的进程能够并行工作
23. 任何进程使用管程所管理的临界资源时,需要调用特定的 \_\_\_\_\_ 才能互斥地进入管程,使用资源。  
A. 系统调用                      B. 访管指令  
C. 管程中的有关入口过程                      D. 同步操作原语

## 二、填空题

1. 并发的实质是一个处理器在多个程序之间的 \_\_\_\_\_。
2. 通常将并发进程之间的制约关系分为两类: \_\_\_\_\_ 和 \_\_\_\_\_。
3. P、V 操作原语是对 \_\_\_\_\_ 执行的操作,其值只能由 P、V 操作改变。
4. 若一个进程已经进入临界区,其他欲进入同一临界区的进程必须 \_\_\_\_\_。
5. 一次仅允许一个进程访问的资源称为 \_\_\_\_\_。
6. 进程访问临界资源的那段代码称为 \_\_\_\_\_。
7. 在进程的同步和互斥问题中,可以用布尔变量实现 \_\_\_\_\_。
8. 在操作系统中,使用信号量可以解决进程间的 \_\_\_\_\_ 与 \_\_\_\_\_ 问题。
9. 每执行一次 Wait() 操作,信号量的数值  $S$  减 1。若 \_\_\_\_\_,则该进程继续执行,否则进入 \_\_\_\_\_ 状态。

10. 每执行一次 Signal()操作,信号量的数值 S 加 1。若 \_\_\_\_\_,则该进程继续执行;否则,从对应的 \_\_\_\_\_ 队列中移出一个进程,该进程的状态将为 \_\_\_\_\_。

11. 有  $m$  个进程共享一个同类临界资源,如果使用信号量解决进程间的互斥问题,那么信号量的取值范围为 \_\_\_\_\_。

12. 有  $m$  个进程共享  $n$  个同类临界资源,如果使用信号量解决进程间的互斥问题,那么信号量的取值范围为 \_\_\_\_\_。

13. 互斥信号量 S 的当前值为 -2 表示 \_\_\_\_\_。

14. 某一时刻系统中共有 6 个进程,每个进程要使用 1 个相关临界资源。互斥信号量 S 的初值为 3,当前值为 -2,则表示有 \_\_\_\_\_ 个进程正在访问相关临界资源,有 \_\_\_\_\_ 个访问相关临界资源的进程进入了阻塞态,有 \_\_\_\_\_ 个进程还没有申请访问相关临界资源。

15. 信号量当前值大于零时其数值表示 \_\_\_\_\_。

16. 有  $m$  个进程共享一个临界资源,若使用信号量机制实现对临界资源的访问,则信号量的初值应设为 \_\_\_\_\_,其取值范围为 \_\_\_\_\_。

17. 利用信号量实现进程的 \_\_\_\_\_,应为临界区设置一个信号量 mutex,其初值为 1,表示该资源尚未使用,临界区应置于 \_\_\_\_\_ 和 \_\_\_\_\_ 原语之间。

18. 操作系统中信号量的值与 \_\_\_\_\_ 的使用情况有关,它的值仅能由 \_\_\_\_\_ 来改变。

19. 操作系统中的一种同步与互斥机制,由共享资源的数据及其在该数据上的一组操作组成,该机制称为 \_\_\_\_\_。

20. 一个进程要向另一个进程传送大量数据,如不考虑进程间的同步,效率最高的进程通信机制为 \_\_\_\_\_。

21. 与 E-mail 类似的进程间数据通信机制是 \_\_\_\_\_。

22. 在默认的情况下,大多数信号会导致接收进程 \_\_\_\_\_。

23. 实现一个管程时,必须考虑的 3 个主要问题是互斥、\_\_\_\_\_ 和 \_\_\_\_\_。

24. 信箱通信机制通常采用 \_\_\_\_\_ 原语和 \_\_\_\_\_ 原语。

### 三、问答题

1. 使用开关中断方法实施临界区互斥的缺点是什么? 克服该缺点的改进方法是什么?

2. 说明互斥和同步对信号量操作方法的差异。

3. 在两个进程间的同步,如计算进程和打印进程的经典例子中,为什么对一个缓冲区要设置两个变量,是否能只设置一个变量,例如,当为 0(缓冲区没数据)时 P1 执行,为 1(缓冲区有数据)时 P2 执行,可以这样实现吗?

4. 为什么要在生产者 and 消费者的同步问题中加入互斥信号量 mutex,而在计算进程和打印进程的两个进程之间的同步问题中不需要加入互斥信号量 mutex?

5. 假如一个阅览室最多可容纳  $N$  个人,读者进入和离开阅览室时,都必须在每次只允许一个人写的登记表上做进入登记和离开登记,系统对读者进入和离开两个过程各建立一个控制进程,试用 P、V 操作实现读者进入与读者离开间的协调关系。

6. 有一座只能容下单列汽车通过的长窄桥,桥两边的汽车在对面没有汽车在桥上的情况下可以上桥并通过桥,且同一方向可以允许任意多的汽车通过。请用信号量操作实现桥

两边汽车的安全通过,两边的汽车各作为一组进程,并说明各个信号量的意义和初值。

7. 编制三个伪程序,用 P、V 操作,以实现公共汽车上司机、售票员和乘客之间的同步。只有车停了后,售票员才能开门,只有售票员开了门后,乘客才能上、下车;只有乘客上好车后,售票员才能关门;只有售票员关好门后,司机才能开车。说明各个信号量的初值及意义。假设初态时车已停稳,售票员没开门。

8. 有两个生产者 a、b 不断向仓库存放产品,由销售者 c 取走仓库中产品(仓库初态产品数为 0,仓库容量为无限大)。请写出通过 P、V 操作实现三个进程间的同步和互斥的框图或伪程序,并写出信号量的初值和意义。

9. 以下两个优先级相同的进程 PA 和 PB 在并发执行结束后,  $x$ 、 $y$  和  $z$  的值分别为多少(信号量  $S_1$  和  $S_2$  的初值均为 0)?

PA:
(1) $x=1$ ;
(2) $x=x+1$ ;
(3) $P(S_1)$ ;
(4) $z=x+1$ ;
(5) $V(S_2)$ ;
(6) $x=x+z$ ;

PB:
(1) $y=1$ ;
(2) $y=y+3$ ;
(3) $V(S_1)$ ;
(4) $z=y+1$ ;
(5) $P(S_2)$ ;
(6) $y=y+z$ ;

10. 有 3 个进程 PA、PB 和 PC 协作文件打印问题: PA 将文件记录从磁盘读入主存的缓冲区 1,每执行一次读一个记录; PB 将缓冲区 1 的内容复制到缓冲区 2,每执行一次复制一个记录; PC 将缓冲区 2 的内容打印出来,每执行一次打印一个记录。缓冲区的大小和一个记录大小一样。请用 P、V 操作来保证文件的正确打印。

11. 图书阅览室共有座位 200 个,并提供一个登记表,表中每一行内容为读者姓名,座位号、进入时间和离开时间。读者进入并找到座位后必须在登记表上登记姓名、座位号、进入时间。读者离开阅览室时也必须要在登记表中记录离开时间。试用 P、V 操作描述读者进程的并发过程。

12. 管程是什么? 管程与进程的区别是什么?

### 3.3 自测题答案及分析

#### 一、选择题

1. B 2. B 3. B 4. A 5. A 6. D 7. B 8. D 9. A 10. B

11. C 12. B 13. A 14. D 15. A 16. B 17. D 18. A 19. C 20. B

21. A 分析: 实现临界区互斥访问的机制包括: 禁止中断、比较和交换指令、exchange 指令、test-and-set 指令, test-and-set 指令与 exchange 指令的功能基本相同,都属于锁操作指令。

22. A 分析: 管程是由一个或多个过程、一个初始化序列和局部数据组成的软件模块,把相关的共享变量及其操作集中在一起统一控制和管理,避免将 P、V 操作分布在各个进程之中造成程序设计错误。管程被请求和释放资源的进程所调用,管程实质上是把临界区集中到抽象数据类型模板中,统一管理和控制。

23. C 分析: 为了使进程互斥地使用临界资源, 可以把管程想象成具有一个入口点, 并保证一次只有一个进程可以进入。其他试图进入管程的进程被阻塞并加入等待管程可用的进程队列中。

## 二、填空题

- |                   |                     |                     |
|-------------------|---------------------|---------------------|
| 1. 多路复用           | 2. 直接制约 间接制约        | 3. 信号量              |
| 4. 等待             | 5. 临界资源             | 6. 临界区(互斥段)         |
| 7. 锁操作            | 8. 同步 互斥            | 9. $S \geq 0$ 阻塞    |
| 10. $S > 0$ 阻塞 就绪 | 11. $1 \sim -(m-1)$ | 12. $n \sim -(m-n)$ |

13. 有 2 个要访问相关临界资源的进程进入了阻塞态, 或有 2 个等待进入临界区的进程。

14. 3、2、1。分析: 互斥信号量  $S$  初值为 3 表示有 3 个同类资源可以被 3 个进程同时访问, 当前值为 -2 表示有 2 个要访问相关临界资源的进程进入了阻塞态, 另外有 3 个进程正在访问相关临界资源, 剩下的  $1 \times (6-5)$  个进程还没有申请访问相关临界资源。

- |  |                     |
|--|---------------------|
| 15. 可用相关资源的数目                                  | 16. $1 \sim -(m-1)$ |
| 17. 互斥 $P(\text{mutex}) \quad V(\text{mutex})$ | 18. 相应资源 $P、V$ 操作   |
| 19. 管程   | 20. 共享内存            |
| 21. 消息通信                                       | 22. 终止              |
| 23. 同步 条件变量                                    | 24. 发送 接收           |

## 三、问答题

1. 答: 用硬件锁, 即用开、关中断的方法可实现锁操作。但这种方法有以下几个不足之处。

(1) 这种方法只能用于单 CPU 系统。在多处理器系统中, 禁止中断只影响执行关中断指令的 CPU, 在其他 CPU 上并行执行的相关进程仍能不受阻碍地进入临界段。

(2) 如果临界段操作比较复杂, 执行时间较长, 那么长时间地关闭中断会降低系统对外部中断响应的速度, 影响系统处理紧迫事件的能力。

(3) 一个运行系统可以有很多的临界段, 应当允许多个进程进入不同的临界段并发地运行。采用开、关中断的硬件锁方法禁止了其他无关的进程进入不同的临界段, 这种做法显然伤害了很多的“无辜者”。

克服该缺点的改进方法是用硬件锁锁软件锁, 用软件锁锁临界段。由于软件锁的 LOCK 操作包含测试和关闭两个操作步骤, 它本身也是一种临界段, 故可以用硬件锁——开、关中断保证软件锁操作的完整性。由于软件锁是一种程序长度最短的临界段, 故用开、关中断的方法保证锁操作的完整性几乎不会影响到系统响应其他的中断请求。用软件锁保证临界段执行的独占性, 也不会影响到其他无关进程进入不同的临界段。

2. 答: 互斥和同步都是通过对信号量的 P、V 操作来实现的, 但这两种控制机制对信号量的操作策略是不同的。互斥的实现是不同的进程对同一信号量进行 P、V 操作, 一个进程在成功地对信号量执行了 P 操作后进入临界段, 并在退出临界段后, 由该进程本身对这信号量执行 V 操作, 表示没有进程处于临界段, 可让其他进程进入。同步的实现由一个进程 PA 对一个信号量进行 P 操作后, 只能由另一个进程 PB 对同一个信号量进行 V 操作, 使 PA 能继续前进, 在这种情况下, 进程 PA 要同步等待 PB。如果进程 PB 也要同步等待 PA,



则要设置另一个信号量。

3. 答: 要采用这个方法, 该变量一定要是共享变量, 如通过共享内存机制分配, 对该变量要互斥访问, 如果用纯软件实现将比较复杂。另外还要专门设计分别针对这两个用户进程的阻塞和唤醒操作, 这要求这两个独立进程要是互相可见的(要有权限, 至少要知道对方的标识数), 而不能采用轮询的耗费处理器时间的方法, 这样做还不如使用两个信号量实现两个进程间的同步。

4. 答: 由于在生产者和消费者问题中的两个信号量 buffers 和 products 的值都可以大于 1, 因此就可能发生有多个生产者进程和消费者进程同时通过 P(buffers) 和 V(products) 操作, 进入缓冲区存取产品的情况。由于存放产品的缓冲区是一种数据结构, 本身也是临界资源, 故对该部分的操作是一个临界段, 各个进程也要互斥地执行。

在计算进程和打印进程的两个进程之间的同步问题中, 由于受对方的制约, 两个进程不可能同时访问缓冲区, 故这种同步中就隐含了互斥。如果像生产者和消费者问题一样, 也加入互斥信号量 mutex, 尽管没有问题, 但这是没有必要的。

5. 答: 信号量的含义及初值如下。

chair: 阅览室椅子数, 即最多可容纳人数, 初值为  $N$ 。

register: 进入登记和离开登记的互斥信号量, 初值为 1。

读者进入	读者离开
P(chair);	P(register);
P(register);	离开登记
进入登记;	V(register);
V(register);	V(chair);
阅读;	离开阅览室;

6. 答: 这个问题类似于读者写者问题中的读者, 区别是桥两边各是一组独立的读者, 这两者之间需要互斥。

int count1, count2: 桥两边汽车上桥的计数器变量, 初值为 0。

mutex1, mutex2: 计数器变量加减的互斥信号量, 初值为 1。

first: 两边允许第一辆汽车上桥的互斥信号量, 初值为 1。

一边的汽车:	另一边的汽车:
while(1) {	while(1) {
P(mutex1);	P(mutex2);
if ( + + count1 = = 1)	if ( + + count2 = = 1)
P(first);	P(first);
V(mutex1);	V(mutex2);
上桥, 通过;	上桥, 通过;
P(mutex1);	P(mutex2);
if ( - - count1 = = 0)	if ( - - count2 = = 0)
V(first);	V(first);
V(mutex1);	V(mutex2);
}	}

7. 答: 信号量的意义及初值如下。

close: 售票员是否关好门, 初值为 1, 已关好门。

stop: 司机是否已停好车, 初值为 0, 没停好车。

open: 售票员是否打开门, 初值为 0, 没开门。

go-up: 乘客是否已上、下好车,初值为 0,没上、下好车。

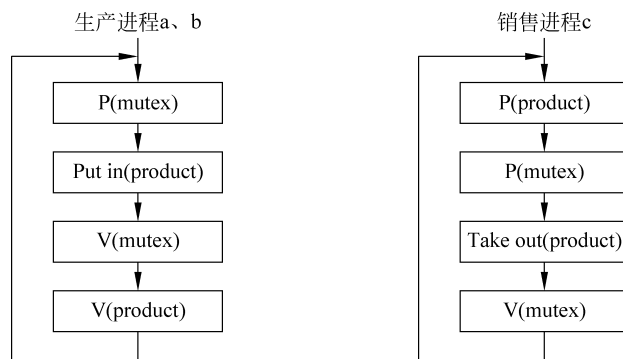
<pre> 司机 while (1) {     P(close);     开车;     停车;     V(stop); }         </pre>	<pre> 售票员 while (1) {     P(stop);     开车门;     V(open);     P(go-up);     关车门;     V(close); }         </pre>	<pre> 乘客 while (1) {     P(open);     上、下车;     V(go-up); }         </pre>
--	--	--

在本程序中,售票员既与司机之间通过信号量 close 和 stop 进行同步,也与乘客之间通过信号量 open 和 go-up 进行同步。

8. 答: 信号量的初值及意义如下。

product: 初值为 0,仓库中已存放的产品个数,同步信号量。

mutex: 初值为 1,向仓库存放产品和从仓库取走产品的互斥信号量。



9. 答: 将 PA 和 PB 进程分解为以下 6 个程序段:

```

SA1 : x = 1;
      x = x + 1;
SA2 : z = x + 1;
SA3 : x = x + z;
SB1 : y = 1;
      y = y + 3;
SB2 : z = y + 1;
SB3 : y = y + z;
    
```

根据 Bernstein 条件,SA1 与 SB1 可以并发执行,SA3 与 SB3 可以并发执行。SA2 与 SB2 因变量交集不为空,而不能并发执行。因此 SA2 与 SB2 的执行顺序不同,可能会有不同的结果。

如果先执行 SA2,则  $x=7, y=9, z=5$ ; 如果先执行 SB2,则  $x=5, y=7, z=3$ 。

10. 答: mutex1 和 mutex2 是两个公用信号量,用于控制进程对缓冲区 1 和缓冲区 2 这两个临界资源访问的互斥。avail1、full1、avail2、full2 分别对应两个缓冲区,其中 avail 的初始值为 1,表示可以利用的缓冲区数目为 1; full 的初值为 0,表示存在于缓冲区内的数据个数为 0。通过对这两组私有信号量的 P、V 操作,就实现了进程间的同步。

三个进程之间的同步描述如下。

```

semaphore mutex1 = 1;
semaphore mutex2 = 1;
semaphore avail1 = 1;
semaphore avail2 = 1;
semaphore full1 = 0;
semaphore full2 = 0;
main()
{
    cobegin
        PA()
        PB()
        PC()
    coend
}

```

```

PB()
{
    While(1) {
        P(full1);
        P(mutex1);
        get from buffer1;
        V(avail1);
        V(mutex1);
        P(avail2);
        P(mutex2);
        put to buffer2;
        V(full2);
        V(mutex2);
    }
}

```

```

PA()
{
    While(1) {
        read from disk;
        P(avail1);
        P(mutex1);
        put to buffer1;
        V(full1);
        V(mutex1);
    }
}

```

```

PC()
{
    While(1) {
        P(full2);
        P(mutex2);
        get from buffer2;
        V(avail2);
        V(mutex2);
        print record;
    }
}

```

11. 答: 这是一个多进程互斥问题。图书阅览室有 200 个座位, 需要设置互斥信号量  $S_1$ , 控制读者进程对座位的竞争; 另外读者进入还是离开都必须在登记表中登记, 而登记表只有一个, 需要互斥访问, 因此需要设置互斥信号量  $S_2$ , 控制读者进程对登记表的使用。设置两个互斥信号量的初值为  $S_1 = 200, S_2 = 1$ 。

读者进程的并发过程描述如下。

```

semaphore S1, S2;
S1 = 200;
S2 = 1;
cobegin
process readeri() // i = 1, 2, ..., 200
{
    P(S1);
    P(S2);
    在表中登记读者姓名、座位号、进入时间;
    V(S2);
    ...
    阅览图书;
    ...
    P(S2);
    在表中登记离开时间;
    V(S2);
    V(S1);
}
coend

```

12. 答：管程(Monitor)就是为了解决信号量机制而提出的一种新的进程间同步互斥机制。管程引入了面向对象的思想。管程是把共享资源的数据结构及一组对该资源的操作和其他相关操作封装在一起所构成的软件模块。进程只能用管程定义的接口进入管程,访问共享资源。在管程的实现中,为了保护管程共享数据结构的数据完整性,需要保证进程互斥地进入,故在管程中定义了阻塞及唤醒操作,设置了进程等待队列。

管程与进程的区别是：进程是活动主体,是动态的,进程能创建和撤销。在操作系统中设置进程的目的在于记录和管理程序的动态执行过程。

管程与操作系统中的共享资源相关,是被动的、静态的,没有创建和撤销。设置管程是为了协调进程的同步与互斥和对共享资源的访问,管程可被进程调用。