第4章　进程同步

4.1　简答题参考答案

1. 什么是临界资源？什么是临界区？

【参考答案】①在计算机中有许多资源一次仅允许一个进程使用，我们把一次仅允许一个进程使用的资源称为临界资源，如打印机和一些共享变量等。②进程中访问临界资源的那段代码称为临界区。

2. 同步机制应遵循的准则有哪些？

【参考答案】同步机制应遵循的准则主要有4个。

（1）空闲让进。当无进程处于临界区时，表明临界资源处于空闲状态，应允许一个请求进入临界区的进程立即进入临界区，以有效地利用临界资源。

（2）忙则等待。当已有进程在临界区时，表明临界资源正在被访问，因而其他试图进入临界区的进程必须等待，以保证对临界资源的互斥访问。

（3）有限等待。对要求访问临界资源的进程，应保证其能在有限时间内进入临界区，以免陷入“死等”状态。

（4）让权等待。当进程不能进入临界区时，其应立即释放处理机，以免进程陷入“忙等”。

3. 为什么各进程对临界资源的访问必须互斥？

【参考答案】临界资源本身的特性决定了它们只能被各个进程互斥地访问，如果并发执行的多个进程同时访问临界资源，则会造成系统混乱或程序执行结果不确定。这样，进程运行结果就可能不正确或者不确定。比如，两个进程并发执行如下程序段：

```
mov ax, (counter);
inc ax;
mov (counter), ax;
```

其中，共享变量counter初值为0，对counter执行加1操作。如果允许一个进程访问counter，另一个进程也可以对其进行操作，则counter的值最终可能是正确结果2，也可能是错误结果1，即计算结果出现了不确定性。因此，各进程对临界资源的访问必须互斥地进行。

4. 如何保证各进程互斥地访问临界资源？

【参考答案】为了互斥地访问临界资源，系统必须保证进程互斥地进入临界区。为此，必须在临界区前增加一段进入区代码，以检查是否有其他进程已进入临界区而在使用临界资源。若有，则进程必须等待；否则，允许进程进入临界区，同时设置标志以表示有进程正在临界区内。同样，在临界区后必须增加一段退出区代码，用于将已有进程进入临界区访问临界资源的标志改为无进程进入临界区使用临界资源。进入区和退出区可用多种同步机制实现，如锁、信

号量机制等。

5. 何谓“忙等”？它有什么缺点？

【参考答案】①“忙等”是指“不让权”的等待，即进程因某事件的发生而无法继续执行时，它仍占用CPU，并通过不断地执行循环测试指令来等待该事件的完成。②“忙等”的主要缺点是浪费CPU时间，另外，它还可能会引起预料不到的后果。例如，考虑某个采取高优先级优先调度原则的系统，目前有两个进程A和B共享某个临界资源，A的优先级较高，B的优先级较低，且B已处于临界区内，而A欲进入自己的临界区，则A、B都不可能继续向前推进，进而即会陷入“死等”状态。

6. 试述采用 Peterson 算法实现临界区互斥的原理。

【参考答案】Peterson算法中Pi和Pj两个进程共享turn和flag两个变量。turn=i表示Pi进程可以进入临界区，flag[i]=TRUE表示Pi进程准备进入临界区。因此，可能会存在以下几种请求情况。

（1）当Pi在临界区中时，若Pj请求进入临界区，则flag[i]=TRUE，turn=i，flag[j]=TRUE，即Pj中flag[i]&&turn=i为TRUE，此时Pj会循环执行while语句——“忙等”而无法进入临界区，满足“忙则等待”。

（2）如果临界区目前空闲，但Pi请求进入，而Pj未请求，此时flag[i]=TRUE，turn=j，flag[j]=FALSE，因此Pi的while条件为FALSE，Pi可进入临界区执行。如果两个进程都要进入临界区，即flag[i]= flag[j]=TRUE，则turn只能取0或1，只能有一个进程进入临界区。当一个进程退出临界区后，另一个进程即可进入临界区，满足“有限等待”。

7. 哪些硬件方法可以解决进程互斥问题？简述它们的用法。

【参考答案】解决进程互斥问题可采用的硬件方法主要有下列3种。

（1）利用“关中断”实现，将临界区放在关中断与开中断之间，关中断后不允许当前进程被中断，因此进程在临界区执行期间都不允许被中断，不能发生进程切换；进程访问完临界区后，再执行开中断指令，此时其他进程才能获得处理机并访问临界区，进而有效保证进程互斥。

（2）Test-and-Set指令，简称TS指令或TSL指令。该指令是一条硬件指令，指令执行过程中不允许被中断，即TS指令把“上锁”和“检查”操作用硬件的方式变成了一气呵成的原子操作。指令执行过程为：①为每个临界资源设置一个布尔变量lock，表示当前临界区是否加锁；②进程进入临界区前，首先用TS指令测试lock，若其值为FALSE，则表示没有进程在临界区内，while循环条件不满足，进入临界区，并将TRUE值赋给lock，即关闭临界区；③任何其他进程再利用TS指令测试lock，while都会一直循环，直到当前访问临界区的进程在退出区进行“解锁”，从而实现了进程互斥。

（3）Swap指令，该指令是用硬件实现的，执行过程不允许中断。其用法是为每个临界资源设置一个全局布尔变量lock，初值为FALSE，在每个进程中再利用一个局部布尔变量key，使用Swap指令与lock进行数值交换，循环判断lock的取值。只有当key为FALSE时，进程才可进入临界区进行操作。从逻辑上看，Swap指令和TS指令并无太大区别，都是先记录下此时临界区是否已经被上锁，再将上锁标记lock设置为TRUE，最后检查局部布尔变量key，如果key为FALSE，则说明之前没有别的进程对临界区上锁，此时可跳出循环，进入临界区。

8.（考研真题）如果用于进程同步的信号量的 P、V 操作不用原语实现，则会产生什么后果？举例说明。

【参考答案】例如：利用P、V操作实现A、B进程对临界资源的互斥使用，代码如下。

```
semaphore S=1;
    A(){                              B(){
        while(1){                         while(1){
            P(S);                             P(S);
            临界区;                           临界区;
            V(S);                             V(S);
            剩余区;                           剩余区;
        }                                 }
    }                                 }
```

若P、V操作不被设计成原语，则执行P、V操作时进程可以被中断。A、B并发执行，初始状态下，临界资源空闲，故应允许第一个申请临界资源的进程（假设为A进程）进入临界区而使用临界资源。但如果A执行到P操作的语句S.value--后（此时S.value的值为0）被B中断，B进程执行P操作，则当B进程执行语句S.value--且S.value的值变为-1时，由于S.value＜0，B会被阻塞，A进程再次获得CPU后，同样也会因为S.value＜0而被阻塞，这就出现了临界资源虽然空闲但进程申请不到的情况，即此时P、V操作无法满足同步机制中“空闲让进”的要求。同样，一个执行P操作的进程被中断后，另一个进程去执行V操作；或一个执行V操作的进程被中断后，另一个进程去执行P或V操作，都将发生混乱，难以实现进程同步。因此，P、V操作必须设计成原语的方式。

9. AND 信号量机制的基本思想是什么？它能解决什么问题？

【参考答案】①AND信号量机制的基本思想是将进程在整个运行过程中所需要的所有临界资源一次性全部分配给进程，待该进程使用完后再一起释放。只要尚有一个所需资源未能分配给该进程，则其他所有将为之分配的资源都不分配给它。亦即，对若干个临界资源的分配采取原子操作方式，要么全部分配到进程，要么一个也不分配。②它能解决的问题是防止死锁的发生，因为该方法在资源分配过程中使产生的死锁必要条件中的“请求和保持”条件不被满足。

10. 利用信号量机制实现进程互斥时，对互斥信号量的 wait() 和 signal() 操作为什么要成对出现？

【参考答案】利用信号量机制实现进程互斥时，对互斥信号量mutex的wait()和signal()操作必须成对出现，缺少wait(mutex)将会导致系统混乱，不能保证进程对临界资源的互斥访问；而缺少signal(mutex)则将会使临界资源永远不被释放，从而使因等待该资源而阻塞的进程不能被唤醒。

11. 什么是管程？它有哪些特性？

【参考答案】由代表共享资源的数据结构以及由对该共享数据结构实施操作的一组过程所组成的资源管理程序共同构成的一个OS资源管理模块，称为管程。

管程是一种程序设计语言结构成分，从语言的角度看，管程主要有以下特性：①模块化，管程是一个基本程序单位，可以单独编译；②抽象数据类型，管程中不仅有数据，而且有针对数据的操作；③信息掩蔽，管程中的数据结构只能被管程中的过程访问，这些过程在管程内部被定义，供管程外的进程调用，而管程中的数据结构以及过程（函数）的具体实现在外部不可见。

12. 试述管程中条件变量的含义和作用。

【参考答案】条件变量是管程内的一种数据结构，且只有在管程中才能被访问，它对于管程内的所有过程而言是全局变量，只能通过两个原语操作来控制它。①x.wait()原语。调用

进程阻塞并移入与条件变量x相关的队列中，释放管程，直到另一个进程在该条件变量x上执行signal()以唤醒等待进程，并将其移出条件变量x的队列。②x.signal()原语。如果存在其他进程由于对条件变量x执行wait()操作而被阻塞，则释放之；如果没有进程在等待，则信号被丢弃。

条件变量是一种信号量，起到了维护等待进程队列的作用。当管程中的进程被阻塞或挂起而不能运行时，如果该进程不释放管程，则其他进程就无法进入管程，此时就需要条件变量来进行控制。

4.2 计算题参考答案

13. 若信号量的初值为 2，当前值为 –1，则表示有多少个等待进程？请分析。

【参考答案】信号量的初值表示系统中资源的数目，每次的P操作表示进程请求一个单位的资源，信号量进行减1操作，当信号量小于0时，表示资源已分配完毕，进程自我阻塞。如果信号量小于0，那么信号量的绝对值表示当前阻塞队列中进程的个数。因此，当前值为-1，表示有1个等待进程。

14. 有 m 个进程共享同一临界资源，若使用信号量机制实现对某个临界资源的互斥访问，请求出信号量的变化范围。

【参考答案】某个临界资源的信号量初值为1，其是信号量的最大值。m个进程分别对临界资源发出1次请求，信号量均要执行减1操作，因此，最多可允许m个进程同时申请，此时信号量的值是$1-m$，为最小值。因此，信号量值的范围是$1-m$至1。

15. 若有 4 个进程共享同一程序段，而且每次最多允许 3 个进程进入该程序段，则信号量值的变化范围是什么？

【参考答案】程序段作为共享资源，最多允许3个进程进入其中，因此设置信号量初值为3。当4个进程共享该程序段时，在每个进程申请进入时，信号量都会执行减1操作。当第1个进程申请进入时，信号量值变为2；第2个进程申请进入时，信号量值变为1；第3个进程申请进入时，信号量值变为0，第4个进程申请进入时，信号量值变为-1。因此，信号量的变化范围是3，2，1，0，-1。

4.3 综合应用题参考答案

16.（考研真题）3 个进程 P_1、P_2、P_3 互斥地使用一个包含 N（$N>0$）个单元的缓冲区。P_1 每次用 produce() 生成一个正整数，并用 put() 将其送入缓冲区的某一空单元中；P_2 每次用 getodd() 从该缓冲区中取出一个奇数，并用 countodd() 统计奇数的个数；P_3 每次用 geteven() 从该缓冲区中取出一个偶数，并用 counteven() 统计偶数的个数。请用信号量机制实现这 3 个进程的同步与互斥活动，并说明所定义的信号量的含义。要求用伪代码描述。

【参考答案】定义资源信号量empty、odd、even，用于控制生产者与消费者之间的同步，其中，empty表示空缓冲区的数目，odd表示缓冲区中奇数的个数，even表示缓冲区中偶数的个数；定义互斥信号量mutex，用于实现进程对缓冲区的互斥访问。伪代码描述如下：

```
semaphore empty=N,even=0,odd=0,mutex=1;
P1:                          P2:                          P3:
while(1){                    while(1){                    while(1){
    x=produce( );                P(odd);                      P(even);
    P(empty);                    P(mutex);                    P(mutex);
    P(mutex);                    getodd( );                   geteven( );
    put(x);                      countodd( );                 counteven( );
    V(mutex);                    V(mutex);                    V(mutex);
    if (x%2==0)                  V(empty);                    V(empty);
        V(even);                 }                            }
    else
        V(odd);
}
```

17.（考研真题）某银行提供了 1 个服务窗口和 10 个供顾客等待时使用的座位。顾客到达银行时，若有空座位，则到取号机上领取一个号，等待叫号。取号机每次仅允许一位顾客使用。当营业员空闲时，通过叫号选取一位顾客，并为其服务。顾客和营业员的活动过程描述如下。

```
cobegin {
    process 顾客 {
        从取号机上获得一个号码;
        等待叫号;
        获得服务;
    }
    process 营业员 {
        while (TRUE) {
            叫号;
            为顾客服务;
        }
    }
} coend
```

请添加必要的信号量和 P、V 操作或 wait()、signal() 操作，实现上述过程中的互斥与同步。要求写出完整的过程，说明信号量的含义并赋初值。

【参考答案】

```
semaphore numget=1, seats=10, custom=0; //numget 是关于取号机互斥的信号量；信号量
                                        seats 是座位的个数；信号量 custom 是顾客的个数
```

```
process 顾客 {
    P(seats);      //看有没有空座位
    P(numget);  //取号
    取号;
    V(numget); //取完号后释放取号机
    V(custom);
    等待叫号;
    V(seats);
    接受服务;
}
```

```
process 营业员 {
    P(custom);
    叫号;
    为顾客服务;
}
```

18. 如图 1-4-1 所示，有 1 个计算进程和 1 个打印进程，它们共享一个单缓冲区，计算进程不断计算出一个整型结果，并将它放入单缓冲区中；打印进程则负责从单缓冲区中取出每个结果并进行打印。请用信号量机制来实现它们的同步关系。

图 1-4-1　共享单缓冲区的计算进程和打印进程

【参考答案】由题意可知，本题中计算进程和打印进程为合作的同步关系。计算进程需要向空闲缓冲区中放入计算好的数据，因此要设置它所需要的empty信号量，由于开始时缓冲区为空，因此empty初值为1；打印进程需要输出已放入缓冲区中的打印结果，因此需要设置它所需要的信号量full，初始状态下缓冲区中无结果可供打印，故full的初值为0。

```
semaphore full=0, empty=l;
int buffer;
cp( ) {
    int nextc;
    while(1) {
        compute the next number nextc;
        P(empty);
        buffer=nextc;
        V(full);
    }
}
main( ) {
    cobegin
    cp( );
    pp( );
    coend
}
```

```
pp( ){
    int nextp;
    while(1) {
        P(full);
        nextp=buffer;
        V(empty);
        print the number in nextp;
    }
}
```

19. 有3个进程P_1、P_2、P_3协作解决文件打印问题。P_1将文件记录从磁盘读入内存的缓冲区1，每执行一次读一个记录；P_2将缓冲区1中的内容复制到缓冲区2中，每执行一次复制一个记录；P_3将缓冲区2中的内容打印出来，每执行一次打印一个记录。缓冲区的大小与记录大小一样。请用信号量机制来保证文件的正确打印。

【参考答案】对缓冲区1来说，P_1是生产者，P_2是消费者；对缓冲区2来说，P_2是生产者，P_3是消费者。缓冲区1和缓冲区2都只能存放一个记录，它们都是临界资源，但无须使用信号量来实现互斥。P_2对于缓冲区1是消费者，对于缓冲区2是生产者，因此要对P_2设置两个信号量来分别控制其对不同缓冲区的不同操作。该文件打印过程的同步算法可描述为：

```
semaphore empty1=1,full1=0,empty2=1,full2=0;
P1( ) {
    while(1) {
        从磁盘读一个记录;
        P(empty1);
        将记录存放到缓冲区1中;
        V(full1);
    }
}

P2( ){
    while(1) {
        P(full1);
        P(empty2);
        从缓冲区1中取一个记录;
        将记录复制到缓冲区2中;
        V(empty1);
        V(full2);
    }
}

P3( ) {
    while(1){
        P(full2);
        从缓冲区2中取一个记录;
        V(empty2);
        将取出的记录打印出来;
    }
}

main( ) {
    cobegin
      P1( );
      P2( );
      P3( );
    coend
}
```

20. 桌上有一个能盛得下5个水果的空盘子。爸爸不停地向盘中放苹果或橘子，儿子不停地从盘中取出橘子享用，女儿不停地从盘中取出苹果享用。规定3人不能同时向（从）盘子中放（取）水果。试用信号量来实现爸爸、儿子和女儿这3个“循环进程”之间的同步。

【参考答案】分析：本题是生产者-消费者问题的变形，相当于一个能生产两种产品的生产者（爸爸）向两个消费者（儿子和女儿）提供产品的同步问题，因此，须设置两个不同的full信号量apple和orange，它们的初值均为0。为了描述上述同步问题，可定义如下信号量：

```
semaphore empty=5, orange=0, apple=0, mutex=1;
```

爸爸、儿子、女儿的算法可描述为：

```
Dad( ) {
    while(1) {
        P(empty);
        P(mutex);
        将水果放入盘中;
        V(mutex);
        if(放入的是橘子)
         V(orange);
    else
         V(apple);
    }
}
```

```
Son( ) {
    while(1) {
        P(orange);
        P(mutex);
        从盘中取一个橘子;
        V(mutex);
        V(empty);
        享用橘子;
    }
}
```

```
Daughter( ) {
    while(1) {
        P(apple);
        P(mutex);
        从盘中取一个苹果;
        V(mutex);
        V(empty);
        享用苹果;
    }
}
```

21. 试用记录型信号量写出一个不会死锁的哲学家进餐问题的算法。

【参考答案】此题有多种解法。其中之一是只允许4个哲学家同时进餐，以保证至少有1个哲学家可以进餐，最终才可能由他释放出其所用过的两根筷子，从而使更多的哲学家可以进餐。为此，须设置一个信号量Sm来限制同时进餐的哲学家数目，使它不超过4，因此可将Sm的初值设置为4。

除了为每根筷子设置一个初值为1的信号量chopstick[i]（i=0，…，4）外，还须再设置一个初值为4的信号量Sm。第i个哲学家的活动可描述为：

```
Pi( ) {
    while(1) {
        P(Sm);
        P(chopstick[ i ]);
        P(chopstick[ (i+1)% 5 ]);
        eat;
        V(chopstick[ i ]);
        V(chopstick[ (i+1)% 5 ]);
        V(Sm);
        think;
    }
}
```