## 第3章 《并发进程》知识清单

注意：请务必准确回答。

①临界区进程同步机制准则：

空闲让进：临界区空闲时，允许一个请求进入临界区的进程立即进入

忙则等待：已有进程进入临界区时，其他请求进入临界区的必须等待

有限等待：对请求访问的进程，要保证有限时间内能进入临界区

让权等待：进程不能进入临界区时，立即释放处理器，防止进程忙等

1、分析程序执行的顺序性与并发性之间的关系。

程序内部的顺序性是指一个可独立调度的任务（进程或线程）内部的语句、指令或模块按照程序设计时规定的顺序执行，不因中断、进程切换以及进程调度的顺序而改变。程序外部的顺序性是指多个相关任务（进程或线程）按照事先确定的顺序逐个执行，完成一个作业级的任务。

程序内部的并发性是并发多线程，程序外部的并发性是并发多进程。进程的并发性是指一组进程的执行在时间上是重叠的，即一个进程执行的第一条指令是在另一个进程执行的最后一条指令完成之前开始的。

程序执行的并发性在运行上依然是顺序性的。

2、解释同步和互斥的概念，两者的区别是什么？

同步是一种合作关系，为完成某种任务而建立的多个进程或者线程之间的协调调用，次序等待，传递消息告知资源占用情况。

互斥是一种制约关系，当一个进程或者多个进程进入临界区后会进行加锁操作，此时其他进程（线程）无法进入临界区，只有当该进程（线程）使用后进行解锁其他人才可以使用，这种技术往往是通过阻塞完成。

互斥：

两个线程之间不可以同时运行，他们会相互排斥，必须等待一个线程运行完毕，另一个才能运行。

同步：

两个线程之间也不可以同时运行，但他是必须要按照某种次序来运行相应的线程（也是一种互斥）

所以说：同步是一种更为复杂的互斥，而互斥是一种特殊的同步。

3、回答临界区和临界资源的概念。

并发进程中与共享变量有关的程序段叫做临界区，共享变量代表的资源叫做临界资源。

4、请列出3种同步互斥机制，简介其原理。

互斥锁：在访问共享资源后临界区域前，对互斥锁进行加锁。在访问完成后释放互斥锁上的锁。对互斥锁加锁后，任何其他试图再加锁的线程会被阻塞，直到锁被释放。

条件变量：条件的检测是在互斥锁的保护下进行的。线程在改变条件状态之前必须首先锁住互斥量。如果一个条件为假，一个线程自动阻塞，并释放等待状态改变的互斥锁。如果另一个线程改变了条件，它发信号给关联的条件变量，唤醒一个或多个等待它的线程，重新获得互斥锁，重新评价条件。如果两进程共享可读写的内存，条件变量可以被用来实现这两进程间的线程同步。

信号量：信号量广泛用于进程或线程间的同步和互斥，信号量本质上是一个非负的整数计数器，它被用来控制对公共资源的访问。可根据操作信号量值的结果判断是否对公共资源具有访问的权限，当信号量值大于 0 时，则可以访问，否则将阻塞。

5、霍尔管程的wait过程如下，请解释每句话的含义，即加上语句注释，写在//后面。

void wait(semaphore&x\_sem, int&x\_count, InterfaceModule&IM)

{

x\_count++;

// 当前进程要在条件变量x上等待，将x上的等待计数器加1

if IM.next\_count > 0 V(IM.next);

//如果有进程正在执行signal操作（即next\_count>0），说明已经有一个被唤醒的进程正在等待管程的重

//新进入权，所以我们要把mutex的控制权交还给那个signal者进程（通过V(next)）

else V(IM.mutex);

// 否则当前没有signal者，我们释放mutex互斥锁，让其他进程能进入管程

P(x\_sem);

// 当前进程阻塞在条件变量对应的信号量x\_sem上，等待signal操作唤醒

x\_count--;

//被唤醒后，从x的等待队列中出来，将x上的等待计数器减1

}

6、霍尔管程的signal过程如下，请解释每句话的含义，即加上语句注释，写在//后面。

void signal(semaphore&x\_sem, int&x\_count, InterfaceModule&IM)

{

if x\_count > 0

{// 如果有进程正在条件变量 x 上等待

IM.next\_count++;

//增加等待 next 信号量的计数器，表示当前有一个正在执行 signal 的进程，

//它将临时放弃管程控制权，等待被唤醒后继续执行

V(x\_sem);

// 唤醒在 x\_sem 上等待的某个进程，即唤醒 wait(x\_sem,...) 中的 P(x\_sem)

P(IM.next);

//signal进程让出管程控制权，自己阻塞在 next 信号量上，

//等待被刚才唤醒的那个进程执行完临界区后唤醒它重新获得执行权

IM.next\_count--;

// 被唤醒后恢复执行，说明临界区已经交还完毕，更新 next\_count

}

}

1. 什么是死锁？

如果在一个进程集合中的每个进程都在等待只能由该集合中的其它进程才能引发的事件，而无限期陷入僵持的局面称为死锁。

8、死锁产生的必要条件有哪些？

互斥条件：进程互斥使用资源，一旦某个资源被占用，欲使用该资源的进程必须等待

占有和等待条件（部分分配条件）：进程申请新资源得不到满足而等待时，不释放已占有资源

不剥夺条件：进程所占有的资源不能被剥夺，只能由进程自己释放

循环等待条件（环路条件）：存在一组进程循环等待资源的现象

9、死锁防止的方法有哪些？课本上生产者-消费者问题算法不会发生死锁，请核对该算法符合死锁防止的哪些方法？

破坏互斥条件：使资源可同时访问而不是互斥使用。该办法对于磁盘、CPU适用，对于磁带机、打印机等多数资源不仅不能破坏互斥使用条件，还要加以保证。

破坏占有和等待条件：静态分配可以破坏占有和等待条件。静态分配是指一个进程必须在执行前就申请它所需要的全部资源，并且直到它所需要的资源都得到满足后才开始执行。资源利用率低。

破坏不剥夺条件：即采用剥夺式调度方法。当进程申请的资源不能满足时，需要收回某些进程已经获得的资源，分配给需要该资源的进程。剥夺调度方法目前只适用于内存资源和处理器资源。

破坏循环等待条件：采用层次分配策略可以破坏循环等待条件。层次分配策略将资源被分成多个层次，进程按照由低到高的层次顺序申请和得到资源，按照由高到低的层次顺序释放资源。当进程得到某一层的一个资源后，如果需要申请该层的另一个资源，则必须先释放该层中的已占资源。

10、进程通信机制有哪些？简述其通信原理。

管道：管道是连接读写进程的一个共享文件，允许进程以先进先出（FCFS）方式写入和读出数据，并对读写操作进行同步。发送进程以字符流形式把大量数据送入管道尾部，接收进程从管道头部接收数据。

共享内存：共享内存是允许两个或多个进程共同访问的物理内存区域，是实现进程通信的一种手段。

消息传递：由信箱、发送原语（send）和接收原语（receive）组成。信箱是存放信件的存储区域，每个信箱可分成信箱头和信箱体两部分。信箱头指出信箱容量、信件格式、信件位置指针等；信箱体用来存放信件，可分成若干个区，每个区容纳一个信件。若信箱未满，则把一封信件（消息）发送到信箱，同时唤醒信件

等待者进程，否则发送者阻塞。若信箱不空，则从信箱接收一封信件（消息），同时唤醒等待发送者进程；否则接受者阻塞。

信号：信号是一种软中断信号，是对硬件中断机制的软件模拟，利用信号可以实现进程间的通信，一个进程可以向另一进程发送某一信号通知该进程某个异常事件发生。接收信号的进程被中断，对该信号代表的事件进行处理。

套接字：套接字（Socket）通信允许互联的位于不同计算机上的进程之间实现通信功能。套接字用于标识和定位特定计算机上特定进程的地址，以便数据准确传输给目标进程。发送端应用程序中的数据通过 Socket 传输到源操作系统的网络驱动程序，再经网络发送至目标操作系统的网络驱动程序，最后由接收端应用程序通过 Socket 接收数据。

11、下列算法中的SWAP为硬件对换指令，以原子操作的方式交换两个变量的值。

①bool lock=false;

②cobegin

③ Process Pi( )

④ {//i=1,2,...,n

⑤ bool keyi=true;

⑥ do

⑦ {

⑧ SWAP(keyi,lock);

⑨ }while(keyi); //上锁

⑩ {临界区};

SWAP(keyi,lock); //开锁

}

coend

进程P1、P2并发执行上述算法，P1首次执行完行语句⑧后处理器切换到了进程P2，则此时P1的keyi值是多少？P2首次执行完行语句⑧后尚未执行语句⑨时P2的keyi值是多少？P2会先于P1进入临界区吗？

1. keyi == false
2. keyi == true
3. 不会，P2 key == true 会进入自旋，无法进入临界区，需要P1访问完临界区将key改为false才能让P2跳出循环，进入临界区。

12、一个数组包含10个元素，该数组被若干进程并发访问，任何进程可以访问任何一个没有其它进程正在访问的元素。请用信号量与PV操作描述这种访问操作。

