

1.单选题

1.11

下面哪个选项不属于自调度算法的缺点()。

- a 瓶颈问题
- b 成本高
- c 低效性
- d 线程切换频繁

自调度算法的缺点：瓶颈问题，系统中只有一个必须互斥访问的公共就绪队列，在系统有多个处理器的情况下，容易形成瓶颈，故不选 A。低效性，高速缓冲的命中率较低。当进程阻塞后再重新就绪时，只能进入唯一的公共就绪队列，经过进程调度后，不一定能在阻塞前的处理器上运行。如果在每台处理器上都配有高速缓存，则这时在其中保留的该进程的数据已经失效，而在该进程新获得的处理器上又需要重新建立这些数据的备份。由于一个进程的整个生命期中，可能要多次更换处理器，因而使高速缓存的命中率很低。故不选 C。线程切换频繁，在多线程系统中，通常一个应用中的多个线程是相互合作的关系。而采用自调度时，相互合作的线程很难同时获得处理器运行，这将会使某些线程因其合作线程未获得处理器运行而阻塞，进而被切换下来。故不选 D。成本高不属于自调度算法的缺点，故选 B。

1.21

响应时间是指()。

- a 从用户提交请求开始至系统首次产生响应的时间为止
- b 从输入设备信息传送到处理机的时间
- c 处理机对请求信息进行处理的时间
- d 将所形成的响应信息回送到终端显示器的时间

响应时间是指从用户提交一个请求开始直至系统首次产生响应的时间为止的一段时间。故选 A。

它包括 3 部分时间：从输入设备信息传送到处理机的时间、处理机对请求信息进行处理的时间，以及将所形成的响应信息回送到终端显示器的时间。BCD 三个选项都是响应时间的一个组成部分，故不选 BCD。

1.31

下面关于多级队列调度算法，描述正确的是()。

- a 一个进程在运行过程中可能被插入不同的就绪队列
- b 每个就绪队列的优先权不同，调度算法相同
- c 每个就绪队列的优先权相同，调度算法不同
- d 各就绪队列调度算法和优先权都可能不相同

多级队列调度算法根据进程占用内存的大小、优先权或进程类型，将就绪队列分成多个独立队列，进程被永久地分配到一个队列，故不选 A，每个队列的优先权

相同，调度算法也相同，故不选 B 和 C，但不同队列的调度算法和优先权都可能不相同。故选 D。

1.41

()算法是在现代分时系统中广泛使用的进程调度算法。

- a 先来先服务
- b 短进程优先
- c 优先权调度
- d 时间片轮转调度

先来先服务算法从就绪队列的队首选择最先到达就绪队列的进程，为该进程分配 CPU。故不选 A。短进程优先算法从就绪队列中选择估计运行时间最短的进程，将 CPU 分配给它。故不选 B。当使用优先权调度算法进行进程调度时，系统将 CPU 分配给就绪队列中优先权值最高的进程。故不选 C。时间片轮转调度算法在现代分时系统中广泛使用。故选 D。

1.51

解除死锁的途径有()。

- a 一个
- b 两个
- c 三个
- d 四个

解除死锁的途径有两个：一是终止处于死锁状态的进程，二是抢占死锁进程占有的资源。故选 B。

1.61

产生死锁的必要条件中，不能被摒弃的条件是()。

- a 互斥条件
- b 请求和保持条件
- c 不剥夺条件
- d 环路等待条件

有些共享资源必须被定义为临界资源，对于这些资源的访问必须是互斥的。因此互斥条件不能摒弃，故选 A。摒弃请求和保持条件，系统要求所有进程执行前要一次性地申请在整个运行过程中所需要的全部资源，只要有一个资源申请不成功，其他所有资源也不分配给该进程，并阻塞该进程。故不选 B。摒弃不剥夺条件，一个已保持了某些资源的进程，当它再提出新的资源要求而不能立即得到满足时，必须释放它已经保持的所有资源。故不选 C。摒弃环路等待条件，进程必须按规定的顺序申请资源。对所有不同类型的资源排序，要求每个进程按规定的顺序申请资源。故不选 D。

1.71

下面有关时间片太长描述不正确的是()。

- a 可使多数进程在一个时间片内处理完
- b 可降低进程的周转时间
- c 可能造成交互用户的响应时间过长
- d 会增加进程切换和调度的开销

时间片太长,可使多数进程在一个时间片内处理完,可降低进程的周转时间,但可能造成交互用户的响应时间过长。故不选 ABC。时间片太短会增加进程切换和调度的开销。故选 D。

1.81

进程必须按规定的顺序申请资源,摒弃的是()条件。

- a 互斥条件
- b 请求和保持条件
- c 不剥夺条件
- d 环路等待条件

死锁的预防,通过保证至少其中一个条件不成立来达到预防发生死锁的目的。有些共享资源必须被定义为临界资源,对于这些资源的访问必须是互斥的。因此互斥条件不能摒弃,故不选 A。摒弃请求和保持条件,系统要求所有进程执行前要一次性地申请在整个运行过程中所需要的全部资源,只要有一个资源申请不成功,其他所有资源也不分配给该进程,并阻塞该进程。故不选 B。摒弃不剥夺条件,

一个已保持了某些资源的进程,当它再提出新的资源要求而不能立即得到满足时,必须释放它已经保持的所有资源。故不选 C。摒弃环路等待条件,进程必须按规定的顺序申请资源。对所有不同类型的资源排序,要求每个进程按规定的顺序申请资源。故选 D。

1.9₁

优先权调度算法的一个主要问题是无穷阻塞,或称()。

- a 无限阻塞
- b 饥饿
- c 老化
- d 死锁

优先权调度算法的一个主要问题是无穷阻塞,或称饥饿 (Starving) 问题。故选 B。无穷阻塞的解决方案之一是老化 (Aging) 技术。故不选 C。由于多个进程竞争共享资源而引起的进程不能向前推进的僵持状态称为死锁。故不选 D。

1.10₁

周转时间是指()。

- a 进程在 CPU 上执行的时间
- b 作业在外存后备队列上等待调度的时间
- c 进程在就绪队列上等待进程调度的时间
- d 作业被提交给系统开始,到作业完成为止的这段时间间隔。

周转时间是指作业被提交给系统开始，到作业完成为止的这段时间间隔。它包括 4 部分时间：作业在外存后备队列上等待调度的时间，进程在就绪队列上等待进程调度的时间，进程在 CPU 上执行的时间，以及进程等待 I/O 操作完成的时间。ABC 三个选项都只是周转时间的一个部分，故选 D。

1.11₁

死锁产生时，至少需要同时满足()个必要条件。

- a 1
- b 2
- c 3
- d 4

死锁产生时，必须同时满足 4 个必要条件：互斥条件，请求和保持条件，不剥夺条件，环路等待条件。

1.12₁

下面关于优先权类型的描述，不正确的是()。

- a 根据优先权的变化特点将优先权的类型分为静态优先权和动态优先权
- b 静态优先权在创建时确定，在进程的整个运行期间不变。
- c 动态优先权在进程创建时被赋予的优先权，随进程的推进或随其等待时间的增加而改变。
- d 静态优先权比动态优先权有更好的调度性能

根据优先权的变化特点将优先权的类型分为静态优先权和动态优先权。故不选 A。静态优先权在创建时确定，在进程的整个运行期间不变。故不选 B。动态优先权在进程创建时被赋予的优先权，随进程的推进或随其等待时间的增加而改变。故不选 C。动态优先权调度算法可以使系统获得更好的调度性能。故选 D。

1.13₁

优先权调度算法中，内部定义优先权可使用一些可测量数据以计算进程的优先权值，下面哪个选项不是内部定义优先权可使用的测量数据()。

- a 时间极限
- b 内存要求
- c 打开文件的数量
- d 进程的重要性

在使用优先权调度的系统中，每个进程都有一个与之关联的优先权。优先权可以通过内部或者外部方式来定义。内部定义优先权可使用一些可测量数据以计算进程的优先权值。例如，时间极限、内存要求、打开文件的数量，平均 I/O 服务时间与平均 CPU 服务时间之比。故不选 ABC。外部优先权是通过操作系统之外的准则来设置的，如进程的重要性、用于支付使用计算机的费用和数量等。故选 D。

1.14₁

采用多级队列调度，下面说法不正确的是()。

- a 进程在被撤销前可在不同队列之间移动

- b 降低了进程调度的开销
- c 每个队列有自己的调度算法
- d 对低优先权进程会存在无穷阻塞问题

采用多级队列调度，一旦进程进入系统，就被固定地分配到一个就绪队列中，进程在被撤销前不会在不同队列之间移动，故选 A。每个队列有自己的调度算法，故不选 C。降低了进程调度的开销，故不选 B。但对低优先权进程会存在无穷阻塞（饥饿）的问题。故不选 D。

1.151

进程调度算法是指从()中选择一个进程为其分配 CPU。

- a 执行态进程
- b 就绪态进程
- c 阻塞态进程
- d 以上均有可能

进程调度算法是指从就绪态进程中选择一个进程为其分配 CPU，使其进入执行态的算法。故选 B。

1.161

时间片是一个较小的时间单位，通常为()。

- a 10~100 秒
- b 10~100 毫秒

- c 10~100 微秒
- d 10~100 纳秒

时间片是一个较小的时间单位，通常为 10~100ms。故选 B。

1.17₁

以下关于最低松弛度优先 LLF 算法，描述不正确的是()。

- a 松弛度表示一个实时进程的紧迫程度
- b 调度程序在调度时，每次选择松弛度最小的进程，把 CPU 分配给该进程。
- c 进程按松弛度排序
- d 让松弛度最小的进程处于就绪队列队尾

松弛度表示一个实时进程的紧迫程度，故不选 A。调度程序在调度时机到来时，每次选择松弛度最小的进程，把 CPU 分配给该进程。故不选 B。把进程按松弛度排序，故不选 C。让松弛度最小的进程处于就绪队列队首。故选 D。

1.18₁

若某系统有 5 个并发进程，每个进程需要同类资源 3 个，该系统不会发生死锁需要的最少资源数是()。

- a 9
- b 10
- c 11

- d 12

若某系统中有 n (n 为大于 1 的整数) 个并发进程, 每个进程需要同类资源 m 个 (m 为大于 0 的整数), 则只要分配 $n*(m-1)+1$ 个资源, 系统就不会发生死锁。每个进程都申请到 $m-1$ 个资源后, 还有 1 个资源, 无论哪个进程获得这个资源都可以继续向前推进, 当获得这个资源的进程执行完后, 可以释放 m 个资源再分配给其他进程, 因此不会发生死锁。 $5*(3-1)+1=11$, 故选 C。

1.19₁

下面有关截止时间的描述不正确的是()。

- a 指某个任务必须开始执行的最迟时间
- b 指某个任务必须完成的最迟时间
- c 指某个任务必须完成的最早时间
- d 是评价实时系统性能的重要指标

截止时间是指某个任务必须开始执行的最迟时间, 或必须完成的最迟时间。故不选 AB, 选 C。是评价实时系统性能的重要指标。故不选 D。

1.20₁

下面有关系统吞吐量的描述不正确的是()。

- a 吞吐量是指单位时间内完成的作业数
- b 是评价系统性能的重要指标之一

- c 调度算法影响系统的吞吐量
- d 调度算法不影响系统的吞吐量

吞吐量是指单位时间内完成的作业数。故不选 A。是评价系统性能的重要指标之一。故不选 B。调度算法影响系统的吞吐量。故不选 C，选 D。

1.211

下面关于短进程优先调度算法，说法不正确的是()。

- a 从就绪队列中选择估计运行时间最短的进程
- b 对长进程不利
- c 能保证紧迫进程的及时处理
- d 进程的长短根据用户的估计而定

从就绪队列中选择估计运行时间最短的进程，将 CPU 分配给它，使它立即执行并一直执行完成，或发生某事件而被阻塞放弃处理机时，再重新调度。故不选 A。对长进程不利。如果系统中不断有短进程到来，长进程可能长时间得不到调度。故不选 B。不能保证紧迫进程的及时处理，因本算法不考虑进程的紧迫程度。故选 C。进程的长度由用户估计而定，可能有偏差。故不选 D。

1.221

就绪时间是一个实时任务成为就绪态的()时间。

- a 起始

- b 等待
- c 延迟
- d 截止

就绪时间是一个实时任务成为就绪态的起始时间。故选 A。

1.23₁

当设定了系统的最长响应时间值后，时间片的大小与系统允许的最大进程数()。

- a 无相关性
- b 成正比
- c 成反比
- d 不成比例

设系统的最长响应时间为 T ，最大进程数目为 N ，时间片值为 q ，则有 $T = Nq$ 。

在 T 值一定的情况下， q 与 N 成反比。即当设定了系统的最长响应时间值后，时间片的大小与系统允许的最大进程数成反比。故选 C。

1.24₁

如果时间片轮转调度算法的时间片很大，且就绪队列按先进先出对进程排序，那么时间片轮转调度算法与()算法一样。

- a 先来先服务
- b 短进程优先

- c 优先权调度
- d 多级队列调度

如果时间片轮转调度算法的时间片很大，且就绪队列按先进先出对进程排序，那么时间片轮转调度算法与先来先服务算法一样。因为时间片如果很大，和不划分时间片的情况相同。故选 A。

1.251

在系统允许的最大进程数一定的情况下，时间片的长短取决于()。

- a 系统要求的响应时间
- b 就绪队列中进程的数目
- c 系统的处理能力
- d 阻塞队列中进程的数目

设系统响应时间为 T ，进程数目为 N ，时间片为 q ，则有 $T=Nq$ ，即在 N 一定的情况下， T 的大小取决于 q 。即时间片的长短取决于系统要求的响应时间。故选 A。

1.261

银行家算法主要应用了()方法。

- a 死锁的预防
- b 死锁的避免

- c 检测并解除死锁
- d 忽略死锁

银行家算法的说明：银行家算法分为两个过程，一是进行资源试分配的过程；二是对试分配后系统的状态做安全性检测的过程。经安全性检测，若试分配后系统状态是安全的，则分配资源。若不安全，则阻塞申请资源的进程，暂不为它分配资源。而死锁的避免，把系统的资源分配状态分为安全状态和不安全状态，只要资源分配使系统资源分配状态处于安全状态，死锁就不会发生，故选 B。

1.271

下面有关时间片太长描述正确的是()。

- a 一个进程需经过多次调度才能执行完
- b 可使多数进程在一个时间片内处理完
- c 进程切换和进程调度的开销会增加
- d 系统的平均周转时间较长

时间片太长，可使多数进程在一个时间片内处理完。故选 B。时间片太短，一个进程需经过多次调度才能执行完，会增加进程切换和调度的开销，系统的平均周转时间也较长。故不选 ACD。

1.281

下面有关先来先服务调度算法，描述不正确的是()。

- a 适合短进程，不利于长进程
- b 使短进程的周转时间过长
- c 有利于 CPU 繁忙型进程
- d 从就绪队列的队首选择最先到达的进程

先来先服务调度算法适合长进程，不利于短进程。故选 A。使短进程的周转时间过长，系统的平均周转时间也比较长，故不选 B。有利于 CPU 繁忙型进程，不利于 I/O 繁忙型进程，故不选 C。从就绪队列的队首选择最先到达就绪队列的进程，为该进程分配 CPU，故不选 D。

1.29₁

下面关于死锁的避免，描述不正确的是()。

- a 系统的资源分配状态分为安全状态和不安全状态
- b 只要系统资源分配状态处于安全状态，死锁就不会发生。
- c 不安全状态一定是死锁状态
- d 避免进程死锁的实质在于使系统处于安全状态

避免死锁的方法是把系统的资源分配状态分为安全状态和不安全状态，故不选 A。只要资源分配使系统资源分配状态处于安全状态，死锁就不会发生，故不选 B。不安全状态不一定是死锁状态，但当系统进入不安全状态之后，便可能进入死锁状态，故选 C。因此，避免进程死锁的实质在于使系统处于安全状态。故不选 D。

1.30₁

下面哪种操作系统不是采用基于时间片轮转、支持优先权和抢占式调度的混合式进程调度算法()。

- a MS-DOS
- b UNIX
- c Linux
- d Windows

MS-DOS 是早期的一种操作系统，故选 A。时间片轮转调度算法在现代分时系统中广泛使用，UNIX、Linux、Windows 操作系统都采用基于时间片轮转、支持优先权和抢占式调度的混合式进程调度算法。故不选 BCD。

1.31₁

摒弃产生死锁的必要条件的方法是()。

- a 死锁的预防
- b 死锁的避免
- c 检测并解除死锁
- d 忽略死锁

死锁的预防，通过保证至少其中一个必要条件不成立来达到预防发生死锁的目的，故选 A。死锁的避免，把系统的资源分配状态分为安全状态和不安全状态，只要资源分配使系统资源分配状态处于安全状态，死锁就不会发生，故不选 B。死锁

的检测和解除，操作系统可以不采取事先预防和避免的方法来解决死锁问题，而是检测是否有死锁发生。如果检测到系统中有死锁的进程，则解除死锁，故不选 C。忽略死锁问题，即假定死锁不可能在系统内发生而忽略死锁，故不选 D。

1.321

多级反馈队列算法的设计要考虑()个方面的问题。

- a 3
- b 4
- c 5
- d 6

多级反馈队列调度算法的设计要考虑以下 5 方面的问题。(1) 就绪队列的数量。(2) 根据进程优先权确定进程应该进入哪个就绪队列的算法。(3) 用以确定进程何时转移到较高优先权队列的算法。(4) 用以确定进程何时转移到较低优先权队列的算法。(5) 用以确定进程在需要服务时应该进入哪个队列的算法。故选 C。

1.331

进程已经保持了至少一个资源，又提出了新的资源要求，属于产生死锁必要条件的()。

- a 互斥条件
- b 请求和保持条件

- c 不剥夺条件
- d 环路等待条件

互斥条件，指一个进程在访问资源的过程中，其他进程不能访问该资源，故不选 A。请求和保持条件，进程已经保持了至少一个资源，又提出了新的资源要求，而新请求的资源已经被其他进程占有，此时进程阻塞，但又对已经获得的资源保持不放，使得其他进程无法使用被保持的资源。故选 B。不剥夺条件，进程已经获得的资源不能被剥夺，只能由进程自己释放。故不选 C。环路等待条件，在发生死锁时，必然存在一个进程申请资源的环形链。故不选 D。

1.341

如果一个进程的完成截止时间为 T ，当前时间为 T_c ，处理完该任务还需要的时间为 T_s ，则松弛度 L 的计算式表示为()。

- a $L=T$
- b $L=T-T_c$
- c $L=T-T_s$
- d $L=T-T_c-T_s$

松弛度用来表示一个实时进程的紧迫程度。如果一个进程的完成截止时间为 T ，当前时间为 T_c ，处理完该任务还需要的时间为 T_s ，则松弛度 L 的计算式表示为： $L=T-T_c-T_s$ 。故选 D。

1.35₁

处理死锁的基本方法有()种。

- a 3
- b 4
- c 5
- d 6

处理死锁的基本方法有预防死锁、避免死锁、检测并解除死锁和忽略死锁问题，共 4 种。故选 B。

1.36₁

响应时间是分时系统性能的重要指标，响应时间越短，时间片取值应该()。

- a 越大
- b 越小
- c 大小固定
- d 随机

时间片的长短取决于系统要求的响应时间。响应时间越短，时间片取值应该越小。故选 B。

1.37₁

()年 Dijkstra (迪杰斯特拉) 提出了一种能够避免死锁的资源分配算法。

- a 1963
- b 1964
- c 1965
- d 1966

1965 年 Dijkstra (迪杰斯特拉) 提出了一种能够避免死锁的资源分配算法 , 故选 C。其基本思想是一个进程提出资源请求后 , 系统先进行资源的试分配。然后检测本次的试分配是否使系统处于安全状态 , 若安全则按试分配方案分配资源 , 否则不分配资源。

1.38₁

每当进程在 CPU 上连续运行的时间()一个时间片长度时 , 操作系统在时钟中断处理过程中会抢占 CPU , 进行进程切换。

- a 小于
- b 小于等于
- c 等于
- d 大于

每当进程在 CPU 上连续运行的时间等于一个时间片长度时 , 操作系统在时钟中断处理过程中会抢占 CPU , 进行进程切换 , 用新的就绪进程替代当前进程 , 被替换的当前进程重新回到就绪队列中。故选 C。

1.39₁

可分为抢占式和非抢占式两种类型的调度算法是()。

- a 先来先服务调度算法
- b 短进程优先调度算法
- c 优先权调度算法
- d 时间片轮转调度算法

用于进程调度的优先权调度算法可以分为非抢占式优先权调度算法和抢占式优先权调度算法两种类型。故选 C。

1.40₁

关于对称多处理器系统，描述不正确的是()。

- a 对称多处理器系统属于同构的多处理器系统
- b 进程到处理器的分配可以采用静态分配和动态分配
- c 静态分配方式的进程调度开销小
- d 静态分配能动态地平衡各处理器的负载

对称多处理器系统属于同构的多处理器系统，其中所包含的各处理单元，在功能和结构上都是相同的，故不选 A。进程到处理器的分配可以采用静态分配和动态分配两种方式，故不选 B。静态分配方式的优点是进程调度的开销小，故不选 C。静态分配的缺点是不能动态地平衡各处理器的负载，使系统存在各处理器忙闲不

均的情况。而动态分配的优点是可以在每次调度时考虑处理器的负载平衡问题，总是把进程分配给当前空闲的处理器，故选 D。

2.多选题

2.12

实现实时调度的基本条件包括()。

- a 提供必要的调度信息
- b 系统处理能力强
- c 采用非抢占式调度机制
- d 采用抢占式调度机制
- e 具有快速切换机制

实现实时调度的基本条件有 4 个，提供必要的调度信息，系统处理能力强，采用抢占式调度机制，具有快速切换机制。故选 ABDE。

2.22

以下关于时间片太短的描述，正确的是()。

- a 一个进程需经过多次调度才能执行完
- b 可使多数进程在一个时间片内处理完
- c 进程切换和进程调度的开销会增加
- d 系统的平均周转时间较长

- e 可能造成交互用户的响应时间过长

时间片太长,可使多数进程在一个时间片内处理完,可降低进程的周转时间,但可能造成交互用户的响应时间过长。故不选 BE。时间片太短,一个进程需经过多次调度才能执行完,会增加进程切换和调度的开销,系统的平均周转时间也较长。故选 ACD。

2.32

下面哪些是自调度算法的优点()。

- a 易移植
- b 减少线程切换
- c 改善系统性能
- d 减少调度开销
- e 有利于提高 CPU 的利用率

自调度,采用自调度的系统中设置有一个公共的就绪队列,任何一个空闲的处理器都可以自行从该就绪队列中选取一个进程或者一个线程运行。自调度算法的优点:易移植,容易将单处理器环境下的调度机制移植到多处理器系统中,故选 A;有利于提高 CPU 的利用率,只要公共就绪队列不为空,就不会出现处理器空闲的情况,故选 E。成组调度,系统将一组相互合作的进程或线程同时分配到一组处理器上运行,进程或线程与处理器一一对应。优点:减少线程切换,改善系统性能,故不选 BC;减少调度开销,一次调度一组,减少调度频率,故不选 D。

2.42

优先权调度算法中，外部优先权是通过操作系统之外的准则来设置的，下面哪些选项可用于设置外部优先权()。

- a 时间极限
- b 进程的重要性
- c 内存要求
- d 用于支付使用计算机的费用
- e 打开文件的数量

在使用优先权调度的系统中，每个进程都有一个与之关联的优先权。优先权可以通过内部或者外部方式来定义。内部定义优先权可使用一些可测量数据以计算进程的优先权值。例如，时间极限、内存要求、打开文件的数量，平均 I/O 服务时间与平均 CPU 服务时间之比。故不选 ACE。外部优先权是通过操作系统之外的准则来设置的，如进程的重要性、用于支付使用计算机的费用和数量等。故选 BD。

2.52

以下哪些情况，系统一定会通过执行进程调度程序重新进行进程调度()。

- a 进程运行结束
- b 进程阻塞
- c 中断返回
- d 优先级更高的进程到来

- e 当前运行进程的时间片用完

当一个进程运行结束、进程阻塞、中断返回、在支持抢占式调度的系统中有比当前运行进程优先级更高的进程到来、当前运行进程的时间片用完，系统都会通过执行进程调度程序重新进行进程调度。故选 ABCE。只有在支持抢占式调度的系统中有比当前运行进程优先级更高的进程到来，系统才会通过执行进程调度程序重新进行进程调度。故不选 D。

3.填空题

3.1₂

银行家算法是用来_____的算法

答案：避免死锁

3.2₂

_____是进程因长时间不能获得所需要的资源而无限等待的状态。

答案：饥饿

3.3₂

抢占式调度算法根据抢占 CPU 的时机不同，可以分为_____和立即抢占。

答案：基于时钟中断的抢占

3.4₂

根据处理器的耦合程度，可以把多处理器系统分为紧密耦合多处理器系统和

_____多处理器系统。

答案：松弛耦合

3.5₂

S 为死锁状态的充分条件是当且仅当 S 状态的资源分配图是_____

答案：不可完全简化的

3.6₂

根据处理器结构是否相同，可以把多处理器系统分为对称多处理器系统和_____多处理器系统。

答案：非对称

3.7₂

进程调度功能由操作系统内核的_____完成。

答案：进程调度程序

4.简答题

4.1₆

简述进程调度的功能。

答案：进程调度功能由操作系统内核的进程调度程序完成。进程调度的功能是按照某种策略和算法从就绪态进程中为当前空闲的 CPU 选择在其上运行的新进程。

4.2₆

请解释什么是死锁？引起死锁的原因是什么？

答案：由于多个进程竞争共享资源而引起的进程不能向前推进的僵持状态称为死锁。

产生死锁的原因：竞争共享资源且分配资源的顺序不当。

4.3₆

简述时间片轮转调度算法。

答案 :时间片轮转调度算法。时间片是一个较小的时间单位 ,通常为 10~100ms。

每当进程在 CPU 上连续运行的时间等于一个时间片长度时 ,操作系统在时钟中断处理过程中会抢占 CPU ,进行进程切换 ,用新的就绪进程替代当前进程 ,被替换的当前进程重新回到就绪队列中。

4.46

简述多级队列调度算法。

答案 :多级队列调度算法。根据进程占用内存的大小、优先权或进程类型 ,将就绪队列分成多个独立队列 ,进程被永久地分配到一个队列 ,每个队列有自己的调度算法。

5.综合题

5.112

下表是一个系统在某时刻的状态 :

用银行家算法回答下列问题 :

(1) 每个进程还需要各类资源的数量多少个 ?

(2) 此时系统是否处于安全状态 ?

答案 :

(1) $need = max - allocation$

(2) 当前可用资源数量 (A B C D) 为 (1 5 2 0) ,可满足 P0 要求 ,先执行进程 P0 ; P0 执行完后 , P0 的已分配资源释放 ,可用资源数量 (A B C D) 变为 (1 5 3 2) ,不能满足 P1 要求 ,可满足 P2 要求 ,所以先执行 P2 ; P2 执行完后 ,可用资源数量 (A B C D) 变为 (2 8 8 6) ,可满足 P3 要求 ,接着执行 P3 ; P3 执行完后 ,可用资源数量 (A B C D) 变为 (2 14 11 8) ,可满足 P4 要求 ,接着执行 P4 ; P4 执行完后 ,可用资源数量 (A B C D)

变为 (2 14 12 12) , 可满足 P1 要求 , 接着执行 P1 ; P1 执行完后 , 可用资源数量 (A B C D) 变为 (3 14 12 12) 。找到可执行序列 (即安全序列) 为 <P0 , P2 , P3 , P4 , P1> , 所以系统处于安全状态。

说明 : 本题答案所用方法有助于理解 , 仅供大家参考 , 考试时建议大家还是使用教材中的方法。