

## 第一章

1. 设计现代OS的主要目标是什么？

答：(1) 有效性 (2) 方便性 (3) 可扩展性 (4) 开放性

2. OS的作用可表现在哪几个方面？

答：(1) OS作为用户与计算机硬件系统之间的接口  
(2) OS作为计算机系统资源的管理者  
(3) OS实现了对计算机资源的抽象

3. 为什么说OS实现了对计算机资源的抽象？

答：OS首先在裸机上覆盖一层I/O设备管理软件，实现了对计算机硬件操作的第一层次抽象；在第一层软件上再覆盖文件管理软件，实现了对硬件资源操作的第二层次抽象。OS通过在计算机硬件上安装多层系统软件，增强了系统功能，隐藏了对硬件操作的细节，由它们共同实现了对计算机资源的抽象。

4. 试说明推动多道批处理系统形成和发展的主要动力是什么？

答：主要动力来源于四个方面的社会需求与技术发展：

- (1) 不断提高计算机资源的利用率；
- (2) 方便用户；
- (3) 器件的不断更新换代；
- (4) 计算机体系结构的不断发展。

5. 何谓脱机I/O和联机I/O？

答：脱机I/O是指事先将装有用户程序和数据的纸带或卡片装入纸带输入机或卡片机，在外围机的控制下，把纸带或卡片上的数据或程序输入到磁带上。该方式下的输入输出由外围机控制完成，是在脱离主机的情况下进行的。而联机I/O方式是指程序和数据的输入输出都是在主机的直接控制下进行的。

6. 试说明推动分时系统形成和发展的主要动力是什么？

答：推动分时系统形成和发展的主要动力是更好地满足用户的需求。主要表现在：CPU的分时使用缩短了作业的平均周转时间；人机交互能力使用户能直接控制自己的作业；主机的共享使多用户能同时使用同一台计算机，独立地处理自己的作业。

7. 实现分时系统的关键问题是什么？应如何解决？

答：关键问题是当用户在自己的终端上键入命令时，系统能及时接收并及时处理该命令，在用户能接受的时延内将结果返回给用户。解决方法：针对及时接收问题，可以在系统中设有多路卡，使主机能同时接收用户从各个终端上输入的数据；为每个终端配备缓冲区，暂存用户键入的命令或数据。针对及时处理问题，应使所有的用户作业都直接进入内存，并且为每个作业分配一个时间片，允许作业只在自己的时间片内运行，这样在不长的时间内，能使每个作业都运行一次。

8. 为什么要引入实时OS？

答：实时操作系统是指系统能及时响应外部事件的请求，在规定的时间内完成对

该事件的处理，并控制所有实时任务协调一致地运行。引入实时 OS 是为了满足应用的需求，更好地满足实时控制领域和实时信息处理领域的需要。

9. 什么是硬实时任务和软实时任务？试举例说明。

答：硬实时任务是指系统必须满足任务对截止时间的要求，否则可能出现难以预测的结果。举例来说，运载火箭的控制等。软实时任务是指它的截止时间并不严格，偶尔错过了任务的截止时间，对系统产生的影响不大。举例：网页内容的更新、火车售票系统。

10. 试从交互性、及时性以及可靠性方面，将分时系统与实时系统进行比较。

答：(1)及时性：实时信息处理系统对实时性的要求与分时系统类似，都是以人所能接受的等待时间来确定；而实时控制系统的及时性，是以控制对象所要求的开始截止时间或完成截止时间来确定的，一般为秒级到毫秒级，甚至有的要低于100微妙。

(2)交互性：实时信息处理系统具有交互性，但人与系统的交互仅限于访问系统中某些特定的专用服务程序。不像分时系统那样能向终端用户提供数据和资源共享等服务。

(3)可靠性：分时系统也要求系统可靠，但相比之下，实时系统则要求系统具有高度的可靠性。因为任何差错都可能带来巨大的经济损失，甚至是灾难性后果，所以在实时系统中，往往都采取了多级容错措施保障系统的安全性及数据的安全性。

11. OS有哪几大特征？其基本的特征是什么？

答：并发性、共享性、虚拟性和异步性四个基本特征；最基本的特征是并发性。

12. 在多道程序技术的OS环境下的资源共享与一般情况下的资源共享有何不同？对独占资源应采取何种共享方式？

一般情况下的共享与操作系统环境下的共享其含义并不完全相同。前者只是说明某种资源能被大家使用，如图书馆中的图书能提供给大家借阅，但并未限定借阅者必须在同一时间（间隔）和同一地点阅读。又如，学校中的计算机机房供全校学生上机，或者说，全校学生共享该机房中的计算机设备，虽然所有班级的上机地点是相同的，但各班的上机时间并不相同。对于这样的资源共享方式，只要通过适当的安排，用户之间并不会产生对资源的竞争，因此资源管理是比较简单的。

而在OS环境下的资源共享或称为资源复用，是指系统中的资源可供内存中多个并发执行的进程共同使用。这里在宏观上既限定了时间（进程在内存期间），也限定了地点（内存）。对于这种资源共享方式，其管理就要复杂得多，因为系统中的资源少于多道程序要求的总和，会形成它们对共享资源的争夺。所以，系统必须对资源共享进行严密管理。

对独占资源采用互斥共享方式。

13. 什么是时分复用技术？举例说明它能提高资源利用率的根本原因是什么？

答：时分复用技术：将资源在不同的时间片内分配给各进程以使该资源被重复利用，从而提高资源的利用率。如采用时分复用技术的虚拟处理机，能够在不同

的时间片内处理多个用户的请求，从而使得用户感觉自己独占主机，而处理机在这期间也被充分的利用。

14. 是什么原因使操作系统具有异步性特征？

答：操作系统的异步性体现在三个方面：一是进程的异步性，进程以人们不可预知的速度向前推进，二是程序的不可再现性，即程序执行的结果有时是不确定的，三是程序执行时间的不可预知性，即一个程序何时执行，执行顺序以及完成时间是不确定的。

15. 处理机管理有哪些主要功能？它们的主要任务是什么？

答：处理机管理的主要功能是：进程管理、进程同步、进程通信和处理机调度；  
进程管理：为作业创建进程，撤消已结束进程，控制进程在运行过程中的状态转换。

进程同步：为多个进程（含线程）的运行进行协调。

进程通信：用来实现在相互合作的进程之间的信息交换。

调度：（1）作业调度。从后备队里按照一定的算法，选出若干个作业，为他们分配运行所需的资源（首先是分配内存）。

（2）进程调度：从进程的就绪队列中，按照一定算法选出一个进程，把处理机分配给它，并设因运行现场，使进程投入执行。

16. 内存管理有哪些主要功能？他们的主要任务是什么？

答：内存管理的主要功能有：内存分配、内存保护、地址映射和内存扩充。

内存分配：为进程分配内存。

内存保护：确保每个用户程序都只在自己的内存空间运行，彼此互不干扰。

地址映射：将地址空间的逻辑地址转换为内存空间与对应的物理地址。

内存扩充：用于实现请求调用功能，狩猎功能等。

17. 设备管理有哪些主要功能？其主要任务是什么？

答：主要功能有：缓冲管理、设备分配和设备处理以及虚拟设备等。

主要任务：完成用户提出的I/O请求，为用户分配I/O设备；提高CPU和I/O设备的利用率；提高I/O速度；以及方便用户使用I/O设备。

18. 文件管理有哪些主要功能？其主要任务是什么？

答：文件管理主要功能：文件存储空间的管理、目录管理、文件的读/写管理和保护。

文件管理的主要任务：管理用户文件和系统文件，方便用户使用，保证文件安全性。

19. 试说明推动传统OS演变为现在OS的主要因素是什么？

（1）系统安全。（2）网络的功能和服务。（3）支持多媒体。

20. 试描述什么是微内核OS。

答：1）足够小的内核 2）基于客户/服务器模式 3）应用机制与策略分离原理 4）采用面向对象技术。

21. 微内核操作系统只有哪些优点？它为何能有这些优点？

答：（1）提高了系统的可扩展性；（2）增强了系统的可靠性；（3）可移植性；（4）提供了对分布式系统的支持；（5）引入了面向对象技术。

22. 在微内核OS中，为什么要采用客户/服务器模式？

答：C/S模式具有独特的优点：

（1）数据的分布处理和存储。（2）使开发中管理。（3）灵活性和可扩充性。（4）易于改编应用软件。

23. 现代操作系统较之传统操作系统又增加了哪些功能和特征？

（1）进程（线程）管理。（2）低级存储器管理。（3）中断和陷入处理。

24. 在基于微内核结构的OS中，应用了哪些新技术？

答：在基于微内核结构的OS中，采用面向对象的程序设计技术。

25. 何谓微内核技术？在微内核中通常提供了哪些功能？

答：把操作系统中更多的成分和功能放到更高的层次（即用户极式）中去运行，而留下一个尽量小的内核，用它来完成操作系统最基本的核心功能，称这种技术为微内核技术。在微内核中通常提供了进程（线程）管理、低级存储器管理、中断和陷入处理等功能。

## 第二章

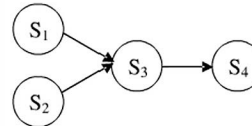
1. 什么是前趋图？为什么要引入前趋图？

答：前趋图(Precedence Graph)是一个有向无循环图，记为 DAG (Directed Acyclic Graph)，用于描述进程之间执行的前后关系。

2. 画出下面四条语句的前趋图：

$S1 = a := x + y$  ;  $S2 = b := z + 1$  ;  $S3 = c := a - b$  ;  $S4 = w := c + 1$  ;

答：其前趋图为：



3. 为什么程序并发执行会产生间断性特征？

答：程序在并发执行时，由于它们共享系统资源，为完成同一项任务需要相互合作，致使这些并发执行的进程之间，形成了相互制约关系，从而使得进程在执行期间出现间断性。

4. 程序并发执行时为什么会失去封闭性和可再现性？

答：程序并发执行时，多个程序共享系统中的各种资源，因而这些资源的状态由多个程序改变，致使程序运行失去了封闭性，也会导致其失去可再现性。

5. 在操作系统中为什么要引入进程概念？它会产生什么样的影响？

答：为了使程序在多道程序环境下能并发执行，并对并发执行的程序加以控制和描述，在操作系统中引入了进程概念。影响使程序的并发执行得以实行。

6. 试从动态性，并发性和独立性上比较进程和程序？

答：(1) 动态性是进程最基本的特性，表现为由创建而产生，由调度而执行，因得不到资源而暂停执行，由撤销而消亡。进程有一定的生命期，而程序只是一组有序的指令集合，是静态实体。

(2) 并发性是进程的显著特征，同时也是 OS 的主要特征。引入进程的目的在于正是为了使程序能和其它进程的并行执行，而程序是不能并发执行的。

(3) 独立性是指进程实体是一个能独立运行的基本单位，也是系统中独立获得资源和独立调度的基本单位。对于未建立任何进程的 program，不能作为独立单位参加运行。

7. 试说明 PCB 的作用，为什么说 PCB 是进程存在的唯一标志？

答：PCB 是进程实体的唯一部分，是操作系统中最主要的记录型数据结构。作用是使一个在多道程序环境下不能独立运行的程序，成为一个能独立运行的基本单位，成为能与其它进程并发执行的进程。OS 是根据 PCB 对并发执行的进程进行控制和管理的。

8. PCB 提供了进程管理和进程调度所需要的哪些信息？

进程管理：通用寄存器、指令计数器、程序状态字、用户栈指针

进程调度：进程状态、进程优先级、事件、其他信息

9. 进程控制块的组织方式有几种？

线性方式、链接方式、索引方式

10. 何谓操作系统内核？内核的主要功能是什么？

现在操作系统一般将 OS 划分为若干层次，再将 OS 的不同功能分别设置在不同的层次中。通常将一些与硬件紧密相关的模块（如中断处理程序等）、各种常用设备的驱动程序以及运行频率较密的模块（如时钟管理、进程调度和许多模块所公用的某些基本操作），都安排在紧靠硬件的软件层次中，将它们常驻内存，即通常被称为的 OS 内核。

支撑功能：中断处理、时钟管理、原语操作

资源管理功能：进程管理、存储器管理、设备管理

11. 试说明进程在三个基本状态之间转换的典型原因。

答：（1）就绪状态—执行状态：进程分配到 CPU 资源

（2）执行状态—就绪状态：时间片用完

（3）执行状态—阻塞状态：I/O 请求

（4）阻塞状态—就绪状态：I/O 完成

12. 为什么要引入挂起状态？该状态有哪些性质？

答：引入挂起状态处于五种不同的需要：终端用户需要，父进程需要，操作系统需要，对换需要和负荷调节需要。处于挂起状态的进程不能接收处理机调度。

13. 在进行进程切换时，所要保存的处理机状态信息有哪些？

答：进行进程切换时，所要保存的处理机状态信息有：

（1）进程当前暂存信息

（2）下一指令地址信息

（3）进程状态信息

（4）进程和系统调用参数及调用地址信息。

14. 试说明引起进程创建的主要事件。

答：引起进程创建的主要事件有：用户登录、作业调度、提供服务、应用请求。

15. 试说明引起进程被撤销的主要事件。

答：引起进程被撤销的主要事件有：正常结束、异常结束（越界错误、保护错、非法指令、特权指令错、运行超时、等待超时、算术运算错、I/O 故障）、外界干预（操作员或操作系统干预、父进程请求、父进程终止）。

16. 在创建一个进程时所完成的主要工作是什么？

答：（1）OS 发现请求创建新进程事件后，调用进程创建原语 `Creat()`：

（2）申请空白 PCB；

（3）为新进程分配资源；

（4）初始化进程控制块；

（5）将新进程插入就绪队列。

17. 在撤销一个进程时所完成的主要工作是什么？

答：（1）根据被终止进程标识符，从 PCB 集中检索出进程 PCB，读出该进程状态

（2）若被终止进程处于执行状态，立即终止该进程的执行，置调度标志真，指示该进程被终止后重新调度。

（3）若该进程还有子进程，应将所有子进程和终止，以防它们成为不可控进程。

（4）将被终止进程拥有的全部资源，归还给父进程，或归还给系统。

（5）将被终止进程的 PCB 从所在队列或列表中移出，等待其它程序搜集信息。

18. 试说明引起进程阻塞或被唤醒的主要事件是什么？

答：a. 请求系统服务；b. 启动某种操作；c. 新数据尚未到达；d. 无新工作可做。



19. 为什么要在OS中引入线程?

答: 在操作系统中引入线程, 则是为了减少程序在并发执行时所付出的时空开销, 使OS具有更好的并发性, 提高CPU的利用率。进程是分配资源的从本单位, 而线程则是系统调度的基本单位。

20. 试说明线程具有哪些属性?

答: (1) 轻型实体 (2) 独立调度和分派的基本单位 (3) 可并发执行 (4) 共享进程资源。

21. 试从调度性, 并发性, 拥有资源及系统开销方面对进程和线程进行比较。

答: (1) iJ/iJ/良性。线程在OS 中作为调度和分派的基本单位, 进程只作为资源拥有的基本单位。

(2) 并发性。进程可以并发执行, 一个进程的多个线程也可并发执行。

(3) 拥有资源。进程始终是拥有资源的基本单位, 线程只拥有运行时必不可少的资源, 本身基本不拥有系统资源, 但可以访问隶属进程的资源和资源。

(4) 系统开销。操作系统在创建、撤销和切换进程时付出的开销显著大于线程。

23. 何谓用户级线程和内核支持线程?

答: (1) 用户级线程: 仅存在于用户空间中的线程, 无须内核支持。这种线程的创建、撤销、线程间的同步与通信等功能, 都无福利用系统调用实现。用户级线程的切换通常发生在一个应用进程的诸多线程之间, 同样无需内核支持。

(2) 内核支持线程: 在内核支持下运行的线程。无论是用户进程中的线程, 还是系统线程中的线程, 其创建、撤销和切换等都是依靠内核, 在内核空间中实现的。在内核空间中还为一个内核支持线程设置了线程控制块, 内核根据该控制块感知某线程的存在并实施控制。

24. 试说明用户级线程的实现方法。

答: 用户级线程是在用户空间中的实现的, 运行在“运行时系统”与“内核控制线程”的中间系统上。运行时系统用于管理和控制线程的函数的集合。内核控制线程或轻型进程和LWP可通过系统调用获得内核提供服务, 利用L\VP进程作为中间系统。

25. 试说明内核支持线程的实现方法。

答: 系统在创建新进程时, 分配一个任务数据区 PTDA, 其中包括若干个线程控制块TCB空间。创建一个线程分配一个TCB, 有关信息写入TCB, 为之分配必要的资源。当PTDA中的TCB用完, 而进程又有新线程时, 只要所创建的线程数目未超过系统允许值, 系统可在为之分配新的TCB; 在撤销一个线程时, 也应回收线程的所有资源和TCB。

26. 多线程模型有哪几种类型? 多对一模型有何优缺点?

多对一模型、一对一模型和多对多模型。

多对一模型的主要缺点在于, 如果一个线程在访问内核时发生阻塞, 则整个进程都会被阻塞; 此外, 在任一时刻, 只有一个线程能够访问内核, 多个线程不能同时在多个处理机上运行。

### 第三章

1 高级调度不低级调度的主要任务是什么? 为什么要引入中级调度?

答: 高级调度的主要任务是根据某种算法, 把外存上处于后备队列中的那些作业调入内存。低级调度是保存处理机的现场信息, 按某种算法先取进程, 再把处理器分配给进程。引入中级调度的主要目的是为了减少内存利用率和系统吞吐率。使那些暂时不能运行的进程不再占用内存资源, 将它们调至外存等待, 把进程状态改为就绪驻外存状态或挂起状态。

2 处理机调度算法的共同目标是什么? 批处理系统的调度目标又是什么?

共同目标: 资源利用率, 公平性, 平衡性, 严格强制执行。

批处理系统的调度目标: 平均周转时间短, 系统吞吐量高, 处理机利用率。

3. 何谓作业、作业步和作业流？

答：作业包含通常的程序和数据，还配有作业说明书。系统根据该说明书对程序的运行进行控制。批处理系统中是以作业为基本单位从外存调入内存。

作业步是指每个作业运行期间都必须经过若干个相对独立相互关联的顺序加工的步骤。

作业流是指若干个作业进入系统后依次存放在外存上形成的输入作业流；在操作系统的控制下，逐个作业进程处理，于是形成了处理作业流。

4. 在什么情况下需要使用作业控制块JCB？其中包含了哪些内容？

答：每当作业进入系统时，系统便为每个作业建立一个作业控制块JCB，根据作业类型将它插入到相应的后备队列中。

JCB 包含的内容通常有：1) 作业标识2) 用户名称3) 用户账户4) 作业类型(CPU繁忙型、I/O 繁忙型、批处理型、终端型)5) 作业状态6) 调度信息(优先级、作业已运行)7) 资源要求8) 进入系统时间9)、开始处理时间10) 作业完成时间11) 作业退出时间12) 资源使用情况等

5. 在作业调度中应如何确定接纳多少个作业和接纳哪些作业？

答：作业调度每次接纳进入内存的作业数，取决于多道程序度。应将哪些作业从外存调入内存，取决于采用的调度算法。最简单的是先来服务调度算法，较常用的是短作业优先调度算法和基于作业优先级的调度算法。

7. 试说明低级调度的主要功能。

答：(1) 保存处理机的现场信息(2) 按某种算法选取进程(3) 把处理机分配给进程。

8. 在抢占调度方式中，抢占的原则是什么？

答：抢占的原则有：时间片原则、优先权原则、短作业优先权原则等。

9. 在选择调度方式和调度算法时，应遵循的准则是什么？

答：(1) 面向用户的准则：周转时间短、响应时间快、截止时间的保证、优先权准则。

(2) 面向系统的准则：系统吞吐量的、处理机利用率好、各类资源的平衡利用。

10. 在批处理系统、分时系统和实时系统中，各采用哪几种进程(作业)调度算法？

答：批处理系统的调度算法：短作业优先、优先权、高响应比优先、多级反馈队列调度算法。

分时系统的调度算法：时间片轮转法。

实时系统的调度算法：最早截止时间优先即EDF、最低松弛度优先即LLF 算法。

11. 何谓静态和动态优先级？确定静态优先级的依据是什么？

答：静态优先级是指在创建进程时确定且在进程的整个运行期间保持不变的

优先级。

动态优先级是指在创建进程时赋予的优先权，可以随进程推进或随其等待时间增加而改变的优先级，可以获得更好的调度性能。

确定进程优先级的依据：进程类型、进程对资源的需求和用户要求。

12. 试比较FCFS和SPF两种进程调度算法。

答：相同点：两种调度算法都可以用于作业调度和进程调度。

不同点：FCFS调度算法每次都从后备队列中选择一个或多个最先进入该队列的作业，将它们调入内存、分配资源、创建进程、插入到就绪队列。该算法有利于长作业/进程，不利于短作业/进程。SPF算法每次调度都从后备队列中选择一个或若干个估计运行时间最短的作业，调入内存中运行。该算法有利于短作业/进程，不利于长作业/进程。

13. 在时间片轮转法中，应如何确定时间片的入小？

答：时间片应略大于一次典型的交互需要的的时间。一般应考虑三个因素：系统对相应时间的要求、就绪队列中进程的数目和系统的处理能力。

14. 通过一个例子来说明通伟的优先级调度算法不能适用于实时系统？

答：实时系统的调度算法很多，主要是基于任务的开始截、H时间和任务紧急/松弛程度的任务优先级调度算法，通伟的优先级调度算法不能满足实时系统的调度实时性要求而不适用。

15. 为什么说多级反馈队列调度算法能较好地满足各方面用户的需要？

终端型用户：由于终端型用户提交的作业多属于交互型作业，通伟较小，系统只要能使这些作业在第二队列规定的时间片内完成，便可使终端型用户感到满意。

短批处理作业用户：对于这类作业，如果可在第二队列中执行完成，使获得与终端型作业一样的响应时间。对于稍长的知作业，也只可在第二和第三队列各执行一个时间片完成，其周转时间仍然较短。

长批处理作业用户：对于长作业，它将依次在第一、2、...、n个队列中运行，然后再按轮转方式运行，用户不必担心其作业长期得不到处理。

16. 为什么说传统的几种调度算法都不能算是公平调度算法？

以上介绍的几种调度算法所保证的只是优先运行，如优先级算法是优先级最高的作业优先运行，但并不保证作业占用了多少处理机时间。另外也未考虑到调度的公平性。

17. 保证调度算法是如何做到调度的公平性的？

保证调度算法是另外一种类型的调度算法，它向用户所做出的保证并不是优先运行，而是明确的性能保证，该算法可以做到调度的公平性。一种比较容易实现的性能保证是处理机分配的公平性。如果在系统中有n个相同类型的进程同时运行，为公平起见，须保证每个进程都获得相同的处理机时间 $1/n$ 。

18. 公平分享调度算法又是如何做到调度的公平性的？

在公平分享调度算法中，调度的公平性主要是针对用户而言，使所有用户能获得相同的处理机时间，或所要求的时间比例。

19. 为什么在实时系统中，要求系统（尤其是CPU）有较强的处理能力？

答：在实时系统中，不但包括周期任务、偶发任务、非周期任务，还包括非实时任务。实时任务要求要满足时限，而非实时任务要求要使其响应时间尽可能的短。多种类型任务的混合，使系统的可调度性分析更加困难。实际上有些实时系统CPU处理能力并不强，比如一些嵌入式实时系统，这就要求系统尽量少做一些并发设计任务，留出足够冗余处理实时任务。

20. 按调度方式可将实时调度算法分为哪几种？

答：按调度方式不同，可分为非抢占调度算法和抢占调度算法两种。



21. 什么是最早截止时间优先调度算法，请举例说明之。

答：根据任务的开始截止时间确定的任务优先级调度算法。截止时间越早则优先级越高。该算法要求在系统中保持一个实时任务就绪队列，该队列按各任务截止时间的先后排序。

22. 什么是最低松弛度优先调度算法，请举例说明之。

答：该算法是根据任务的紧急（或松弛）程度，来确定任务的优先级。任务的紧急程度越高，为该任务所赋予的优先级就越高，以使之优先执行。

例如，一个任务在 200ms 时必须完成，而它本身所需的运行时间就有 100ms，因此，调度程序必须在 100ms 之前调度执行，该任务的紧急程度（松弛程度）为 100ms。又如，另一个任务在 400ms 时必须完成，它本身需要运行 150ms，则其松弛程度为 250ms。

最早截止时间优先调度算法：任务要求的截止时间越早，其优先级就越高。

最低松弛度优先调度算法：任务的紧急程度越高，其优先级就越高。

23. 何谓“优先级倒置”现象，可采取什么方法来解决？

当前OS广泛采用优先级调度算法和抢占方式，然而在系统中存在着影响进程运行的资源而可能产生“优先级倒置”的现象，即高优先级进程（或线程）被低优先级进程（或线程）延迟或阻塞。

24. 试分别说明可用资源和可消耗资源的性质。

可用资源：每一个可用资源中的单元只能分配给一个进程使用，不允许多个进程共享。进程在使用可用资源时，须按照这样的顺序：请求资源、使用资源、释放资源。系统中每一类可用资源中的单元数目是相对固定的，进程在运行期间既不能创建也不能删除它。

可消耗性资源：每一类可消耗性资源的单元数目在进程运行期间是可以不断变化的，有时它可以有许多，有时可能为 0。进程在运行过程中，可以不断创造可消耗性资源的单元，将它们放入该资源类的缓冲区中，以增加该资源类的单元数目。进程在运行过程中，可以请求若干个可消耗性资源单元，用于进程自己的消耗，不再将它们返回给该资源类中。

25. 试举例说明竞争不可抢占资源所引起的死锁。

例如，系统中有两个进程 P1 和 P2，它们都准备写两个文件 F1 和 F2，而这两者都属于可重用和不可抢占性资源。进程 P1 先打开 F1，然后再打开文件 F2；进程 P2 先打开文件 F2，后打开 F1，下面示出了这段代码。

```
P1
P2
...
...
Open (f1, w) ;           Open (f2, w) ;
Open (f2, w);           Open (f1, w) ;
```

两个进程 P1 和 P2 在并发执行时，如果 P1 先打开 F1 和 F2，然后 P2 才去打开 F1（或 F2），由于文件 F1（F2）已被 P1 打开，故 P2 会被阻塞。当 P1 写完文件 F1（或 F2）而关闭 F1（F2）时，P2 会由阻塞状态转为就绪状态，被调度执行后重新打开文件 F1（或 F2）。在这种情况下，P1 和 P2 都能正常运行下去。若 P2 先打开 F1 和 F2，然后 P1 才去打开 F1（或 F2），P1 和 P2 同样也可以正常运行下去。

但如果在 P1 打开 F1 的同时，P2 去打开 F2，每个进程都占有一个打开的文件，此时就可能出现问题。因为当 P1 试图去打开 F2，而 P2 试图去打开 F1 时，这两个进程都会因文件已被打开而失败，它们希望对方关闭自己所需要的文件，但谁也无法运行，因此这两个进程将会无限期地等待下去，而形成死锁。

26. 为了破坏“请求和保持”条件而提出了两种协议，试比较这两种协议。

第一种协议在所有进程开始运行之前，必须一次性地申请其在整个运行过程中所需的全部资源，并且在分配资源时，只要有某种资源不能满足进程的要求，即使其它所需的各种资源都空闲也不分配给该进程，而让该进程等待。因此有资源被严重浪费、进程经常会发生饥饿现象等缺点。

第二种协议是对第一种协议的改进，它允许一个进程只获得运行初期所需的资源后，便开始运行。进程运行过程中再逐步释放已分配给自己的，且已用毕的全部资源，然后和请求新的所需资源。如此便可提高设备的利用率，还可减少进程发生饥饿的概率。

27. 何谓死锁？产生死锁的原因和必要条件是什么？

答：(1) 死锁是指多个进程因竞争资源而造成的一种僵局，若无外力作用，这些进程都将永远不能向前推进；

(2) 产生死锁的原因有二：一是竞争资源，二是进程推进顺序非法；

(3) 必要条件是：互斥条件，请求和保持条件，不剥夺条件和环路等待条件。

28. 在解决死锁问题的几个方法中，哪种方法最易于实现？哪种方法是资源利用率最高？

答：解决/处理死锁的方法有预防死锁、避免死锁、检测和解除死锁。其中预防死锁方法最容易实现，但由于所施加的限制条件过于严格，会导致系统资源利用率和系统吞吐量降低；而检测和解除死锁方法可是系统获得较好的资源利用率和系统吞吐量。

加：请详细说明可通过哪些途径预防死锁？

答：(1) 摒弃“请求和保持”条件：系统规定所有进程开始运行之前，都必须一次性地申请其在整个运行过程中所需的全部资源，但在分配资源时，只要有某种资源不能满足某进程的要求，即使其它所需的各资源都空闲，也不分配给该进程，而让该进程等待；

(2) 摒弃“不剥夺”条件：系统规定，进程是逐个地提出对资源的要求的。当一个已经保持了某些资源的进程，再提出新的资源请求而不能立即得到满足时，必须释放它已经保持了的所有资源，待以后需要时再重新申请；

(3) 摒弃“环路等待”条件：系统将所有资源按类型进行线性排序，并赋予不同的序号，且所有进程对资源的请求必须严格按序号递增的次序提出。这样，在所形成的资源分配图中，不可能构成出现环路，因而摒弃了“环路等待”条件。

30. 在教材银行家算法的例子中，如果P0发出的请求向量由Request0 (0, 2, 0) 改为Request0 (0, 1, 0)，问系统可否将资源分配给它？

答：P0发出请求向量Request0 (0, 1, 0)，按银行家算法进行检查

CD Request0 (0, 1, 0) Need0 (4, 4, 3);

⑧ Re4ue斗0 (0, 1, 0) Av ail~ule (2, 3, 0);

⑩系统暂时先假定可为P0分配资源，修改Available, Allocation1和Need1向量在下面数据结构中的数值：

Available[j] :=Available[j] -Request i[j] ;

$Allocation[i, j] := Allocation[i, j] + Request[i, j];$   
 $Need[i, j] := Need[i, j] - Request[i, j];$

计算结果为:

$Available_0 = Available_0 - Request_0(0, 1, 0) = (2, 2, 0)$

$Allocation_0 = Allocation_0 + Request_0(0, 1, 0) = (0, 2, 0)$

$Need_0 = Need_0 - Request_0(0, 1, 0) = (7, 3, 3)$

进程	Work ABC	Allocation ABC	Need ABC	Work-Allocation ABC	Finish
P0	10 5 7	0 2 0	7 3 3	10 7 7	True
P1	5 2 2	3 0 2	0 2 0	8 2 4	True
P2	10 3 7	3 0 2	6 0 0	13 3 9	True
P3	7 3 3	2 1 1	0 1 1	9 4 4	True
P4	7 3 5	0 0 2	4 3 1	7 3 7	True

可以找到一个安全序列 {P1, P3, P4, P2, P0}，所以系统是安全的，系统可以立即将P1所申请的资源(0, 1, 0)分配给它。给P1分配资源之后，系统的资源数目Available= (2, 2, 0)

31. 在银行家算法中，若出现下述资源分配情况：

Process	Allocation	Need	Available
P <sub>0</sub>	0032	0012	1622

P <sub>1</sub>	1000	1750	
P <sub>2</sub>	1354	2356	
P <sub>3</sub>	0332	0652	
P <sub>4</sub>	0014	0656	

试问：

(1) 该状态是否安全？

(2) 若进程P<sub>1</sub>提出请求Request (1, 2, 2, 2)后，系统能否将资源分配给它？

答：

(1) 该状态是安全的，因为存在一个安全序列< P<sub>0</sub>, P<sub>1</sub>, P<sub>3</sub>, P<sub>4</sub> >。下表为该时刻的安全序列表。

资源	Work	Need	Allocation	Work+Allocation	Finish
P <sub>0</sub>	1 6 2 2	0 0 1 2	0 0 3 2	1 6 5 4	true
P <sub>3</sub>	1 6 5 4	0 6 5 2	0 3 3 3	1 9 8 7	true
P <sub>4</sub>	1 9 8 7	0 6 5 6	0 0 1 4	1 9 9 11	true
P <sub>1</sub>	1 9 9 11	1 7 5 0	1 0 0 0	2 9 9 11	true
P <sub>2</sub>	2 9 9 11	2 3 5 6	1 3 5 4	3 12 14 17	true

(2) 若进程P<sub>2</sub>提出请求Request (1, 2, 2, 2)后，系统不能将资源分配给

它，若分配给进程P2, 系统还剩的资源情况为(0, 4, 0, 0)，此时系统中的资源将无法满足任何一个进程的资源请求，从而导致系统进入不安全状态，容易引起死锁的发生。

#### 第四章

##### 1. 为什么要配多层次存储器？

这是因为：

- a. 设置多个存储器可以使存储器两端的硬件能并行工作。
- b. 采用多级存储系统，特别是 Cache 技术，这是一种减轻存储器带宽对系统性能影响的最佳结构方案。
- c. 在做处理机内部设四各种缓冲存储器，以减轻对存储器存取的几力。增加 CPU 中寄存器的数量，也可大大缓解对存储器的压力。

##### 2. 可采用哪几种方式将程序装入内存？它们分别适用于何种场合？

将程序装入内存可采用的方式有：绝对装入方式、重定位装入方式、动态运行时装入方式：绝对装入方式适用于单道程序环境中，重定位装入方式和动态运行时装入方式适用于多道程序环境中。

##### 3. 何为静态链接？静态链接时，需要解决两个什么问题？

答：静态链接是指在程序运行之前，先将各自目标模块及它们所需的库函数，链接成一个完整的装入模块，以后不须拆开的链接方式。

将几个目标链接装配成一个装入模块时，须解决以下两个问题：

将相对地址进行修改。即将除第一个模块外的相对地址修改成装入模块中的模块的相对地址。

变换外部调用符号。即将各个模块中所用的外部调用符号，都变换为相对地址。

##### 4. 何谓装入时动态链接？装入时动态链接方式有何优点？

答：装入时动态链接是指将用户源程序编译后所得到的目标模块，在装入内存时，采用边装入边链接的一种链接方式，即在装入一个目标模块时，若发生一个外部模块调用事件，将引起装入程序去找相应的外部目标模块，把它装入内存中，并修改目标模块中的相对地址。

装入时动态链接方式有以下优点：

- 1) 便于修改和更新
- 2) 便于实现目标模块的共享

##### 5. 何谓运行时动态链接？运行时动态链接方式有何优点？

答：运行时动态链接是将对某些模块的链接推迟到程序执行时才进行链接，也就是，在执行过程中，当发现一个被调用模块尚未装入内存时，立即由 OS 去找到该模块并将之装入内存，把它链接到调用者模块上。

优点：凡是在执行过程中未被用过的目标模块，都不会被调入内存和被链接到装入模块上，这样不仅能加快程序的装入过程，而且可节省大量的内存空间。

6. 在动态分块分配方式中，应如何将各空闲分区链接成空闲分区链？

应在每个分区的起始地址部分，设置一些用于控制分区分配的信息，以及用于链接各分区的前向指针；在分区尾部则设置后向指针，通过前、后向指针将所有的分区链接成一个双向链。P128 图 4-7。主要使用双向链表。

7. 为什么要引入动态重定位？如何实现？

- 程序在运行过程中经常要在内存中移动位置，为了保证这些被移动了的程序还能正常执行，必须对程序和数据地址加以修改，即重定位。引入重定位的目的就是为了满足程序的这种需要。
- 要在不影响指令执行速度的同时实现地址变换，必须有硬件地址变换机构的支持，即须在系统中增设一个重定位寄存器，用它来存放程序在内存中的起始地址。程序在执行时，真正访问的内存地址是相对地址与重定位寄存器中的地址相加而形成的。

8. 什么是基于顺序搜索的动态分区分配算法？它可分为哪几种？

答：为了实现动态分区式分配，将系统中的空闲分区组织成空闲分区表或空闲分区链。所谓顺序搜索，是指按表或链的组织顺序，检索表或链上记录的空闲分区，去寻找一个最符合算法的、大小能满足要求的分区。分区存储管理中常采用的分配策略有：首次适应算法、循环首次适应算法、最佳适应算法、最坏适应算法。

9. 在采用首次适应算法回收内存时，可能出现哪几种情况？应怎样处理这些情况？

- 回收区与插入点的前一个分区相邻接，此时可将回收区与插入点的前一个分区合并，不再为回收分区分配新表项，而只修改前邻接分区的大小；
- 回收分区与插入点的后一个分区相邻接，此时合并两区，然后用回收区的首址作为新空闲区的首址，大小为两区之和；
- 回收区同时与插入点的前后两个分区邻接，此时将三个分区合并，使用前邻接分区的首址，大小为三区之和，取后邻接分区的表项；
- 回收区没有邻接空闲分区，则应为回收区单独建立一个新表项，填写回收区的首址和大小，并根据其首址，插入到空闲链中的适当位置。

10. 什么是基于索引搜索的动态分区分配算法？它可分为哪几种？

答：P131 快速适应算法、伙伴系统、哈希算法

11. 令  $buddyK(x)$  表示大小为  $2^k$  次方、地址为  $x$  的块的伙伴系统地址，试写出  $buddyK(x)$  通用表达式？

答：P126。考研的同学可以关注一下。Buddy System 是一种经典的内存管理算法，在 Unix 和 Linux 操作系统中都有用到，其作用是减少存储空间中的空洞，减少碎片，提高利用率。在有的“数据结构”课本中有约法介绍。

这题我看不懂其题意，大概是分配内存时  $x$  要找的空闲块是：

$$pow(2, k-1) \leq buddyK(x) \leq pow(2, k) \quad // \text{ pow 是乘方函数}$$

Buddy System 是一种连续的内存管理方法，可以结合离散的分页分配方法使用，有利于为进程分配连续的物理块，以提高分页法的效率和程序的局部性。

12. 分区存储管理中常采用哪些分配策略？比较它们的优缺点。

分区存储管理中常采用的分配策略有：首次适应算法、循环首次适应算法、最佳适应算法、最坏适应算法。

- 首次适应算法的优缺点：保留了地址部分的大空闲区，有利于后到来的大型作业的分配；低址部分不断被划分，留下许多难以利用的、小的空闲区，且每次分区分配查找时都是从低址部分开始，会增加查找时的系统开销。
- 循环首次适应算法的优缺点：使内存中的空闲分区分布得更均匀，减少了查找时的系统开销；缺乏大的空闲分区，从而导致不能装入大型作业。
- 最佳适应算法的优缺点：每次分配给文件的都是最适合该文件大小的分区；内存中留下许多难以利用的小的空闲区。
- 最坏适应算法的优缺点：给文件分分区后剩下的空闲区不至于太小，产生碎片的几率最小，对中小型文件分配分区操作有利；使存储器中缺乏大的空闲区，对大型文件的分区分配不利。



13.为什么要引入对换?对换可分为哪几种类型?

答:在多道环境下,一方面,在内存中的某些进程由于某事件尚未发生而被阻塞,但它却占用了宝贵的内存空间,甚至有时可能出现在内存中所有进程都被阻塞而迫使CPU停;另一方面,却又有许多作业在外存上等待,因无内存而不能进入内存运行的情况。显然这对系统资源是一种严重的浪费,且使系统吞吐率下降。为了解决这一问题,在操作系统中引入了对换(也称交换)技术。可以将整个进程换入、换出,也可以将进程的一部分(页、段)换入、换出。前者主要用于缓解目前系统中内存的不足,后者主要用于实现虚拟存储。

14.对文件区管理的目标和对换空间管理的目标有何不同?

答:对文件区管理的主要目标是提高文件存储空间的利用率,然后才提高对文件的访问速度,因此,对文件区空间的管理采取离散分配方式。

对换空间管理的主要目标是提高进程换入和换出的速度,然后才是提高文件存储空间的利用率,因此,对换区空间的管理采取连续分配方式,较少的考虑外存中的碎片问题。

15.为实现对换,系统应具各哪几方面功能?

答:系统应具备三方面功能:对换空间管理,进程换出,进程换入。

16.在以进程为单位进行对换时,每次是否将整个进程换出?为什么?

答:在以进程为单位进行对换时,并非每次将整个进程换出。这是因为:

- a. 从结构上讲,进程是由程序段、数据段和进程控制块组成的,其中进程控制块总有部分或全部常驻内存,不被换出。
- b. 程序段和数据段可能被若干进程共享,此时它们也不能被换出。

17.某离散分配时所用的非本单位不同,可将离散分配分为哪几种?

答:分页存储管理方式、分段存储管理方式和段页式存储管理方式。

18.什么是页面?什么是物理块?页面的大小应如何确定?

答:页面:分页存储管理将进程的逻辑地址空间分成若干个页,并为各页加以编号。物理块:把内存的物理地址空间分成若干个块,并为各块加以编号。页面大小应选择适中,且页面大小应该是2的幂,通常为1KB~8KB。

19.什么是页表?页表的作用是什么?

答:页表是分页式存储管理使用的数据结构。一个进程分为多少页,它的页表就有多少行。每一行记录进程的页和它存放的物理块的页号、块号对应关系。页表用于进行地址变换。

20.为实现分页存储管理,需要哪些硬件支持?

答:需要有页表机制、地址变换机构的硬件支持。

21.在分页系统中是如何实现地址变换的?

答:利用地址变换机构实现从逻辑地址到物理地址的转变,通过页表来实现从页号到物理块号的变换,将逻辑地址中的页号转换为内存中的物理块号。

22.具有快表时是如何实现地址变换的?

23.较详细的说明引入分段存储管理方式是为了满足用户哪几个方面的需求。

答:1)方便编程。用户通常把自己的作业按照逻辑关系划分为若干段,每段都从0编址,并有自己名字和长度。因此,希望要访问的逻辑地址是由段名和段内偏移量决定。

2)信息共享。在实现对程序和数据的共享时,是以信息逻辑单位为基础。分页系统中的页是存放信息的物理单位,无完整意义,不便共享;段是信息的逻辑单位。为了实现段的共享,希望存储管理能与用户程序分段的组织方式相适应。

3) 信息保护。对信息的逻辑单位进行保护, 分段能更有效方便地实现信息保护功能。

4) 动态增长。在实际应用中, 有些段特别是数据段, 在使用过程中会不断增长, 事先又无法确切知道增长多少, 分段存储管理方式能较好解决这个问题。

5) 动态链接。运行时先将主程序对应的目标程序装入内存并启动运行, 运行过程中又需要调用某段时, 才将该段调入内存链接。所以动态链接也要求以段作为管理单位。

24. 在具有快表的段页式存储管理方式中, 如何实现地址变换?

答: 在CPU给出有效地址后, 由地址变换机构自动将页号P送入高速缓冲寄存器, 并将此页号与高速缓存中的所有页号比较, 若找到匹配页号, 表示要访问的页表项在快表中。可直接从快表读出该页对应物理块号, 送到物理地址寄存器中。如快表中没有对应页表项, 则再访问内存页表, 找到后, 把从页表项中读出物理块号送地址寄存器; 同时修改快表, 将此页表项存入快表。但若寄存器已满, 则OS必须找到合适的页表项换出。

25. 为什么说分段系统较之分页系统更易于实现信息共享和保护?

答: a. 对分页系统, 每个页面是分散存储的, 为了实现信息共享和保护, 则页面之间需要一一对应起来, 为此需要建立大量的页表项;

b. 而对分段系统, 每个段都从0开始编址, 并采用一段连续的地址空间, 这样在实现共享和保护时, 只需为所要共享和保护的程序设置一个段表项, 将其中的地址与内存地址一一对应起来即可。

26. 分页和分段有何区别?

答: a. 分页和分段都采用离散分配的方式, 且都要通过地址映射机构来实现地址变换, 这是它们的共同点;

b. 对于它们的不同点有三, 第一, 从功能上看, 页是信息的物理单位, 分页是为实现离散分配方式, 以消减内存的外零头, 提高内存的利用率, 即满足系统管理的需要, 而不是用户的需要; 而段是信息的逻辑单位, 它含有一组其意义相对完整的信息, 目的是为了能更好地满足用户的要求; 第二页的大小固定且由系统确定, 而段的长度却不固定, 决定于用户所编写的程序; 第三分页的作业地址空间是一维的, 而分段的作业地址空间是二维的。

27. 试全面比较连续分配和离散分配方式。

答: a. 连续分配是指为一个用户程序分配一个连续的地址空间, 包括单一连续分配方式和分区式分配方式, 前者将内存分为系统区和用户区, 系统区供操作系统使用, 用户区供用户使用, 是最简单的一种存储方式, 但只能用于单用户单任务的操作系统中; 分区式分配方式分为固定分区和动态分区, 固定分区是最简单的多道程序的存储管理方式, 由于每个分区的大小固定, 必然会造成存储空间的浪费; 动态分区是根据进程的实际需要, 动态地为它分配连续的内存空间, 常用三种分配算法: 首次适应算法, 该法容易留下许多难以利用的小空闲分区, 加入查找开环循环首次适应算法, 该法能使内存中的空闲分区分布均匀, 但会致使缺少大的空闲分区; 最佳适应算法, 该算法也易留下许多难以利用的小空闲区;

b. 离散分配方式是将一个进程且分散地分配到许多不相邻的分区中的思想,

分为分页式存储管理、分段存储管理和段页式存储管理。分页式存储管理旨在提高内存利用率, 满足系统管理的需要, 分段式存储管理则旨在满足用户(程序员)的需要, 在实现共享和保护方面优于分页式存储管理, 而段页式存储管理则是将两者结合起来, 取长补短, 即具有分段系统便于实现, 可共享, 易于保护, 可动态链接等优点, 又能像分页系统那样很好的解决外部碎片的问题, 以及为各个分段可离散分配内存等问题, 显然是一种比较有效的存储管理方式。

c. 综上所述, 连续分配方式和离散分配方式各有各的特点, 应根据实际情况加以改进和利用。

## 第4章

1. 常规存储管理方式具有哪两特征? 它对系统性能有何影响?

答: 一次性: 进程必须全部装入内存, 对空间浪费非常大;

驻留性: 在程序运行过程中, 进程全部驻留在内存, 暂时不用的数据无法释放。

2. 什么是程序运行时的时间局限性和空间局限性？

答：(1)时间局限性：如果程序中的某条指令一旦执行，则不久的将来该指令可能再次被执行；如果某个存储单元被访问，则不久的将来该存储单元可能再次被访问。产生时间局限性的典型原因是在程序中存在大贵的循环操作。

(2)空间局限性：一旦程序访问了某个存储单元，则在不久的将来，其附近的存储单元也很有可能被访问，即程序在一段时间内所访问的地址，可能集中在一定的范围内。产生空间局限性的典型原因是程序是顺序执行的。

3. 虚拟存储器有哪些特征？其中最本质的特征是什么？

答：虚拟存储器有多次性、对换性、虚拟性三大特征。最本质的特征是虚拟性。

4. 实现虚拟存储器需要哪些硬件支持？

a. 诸水分页（段）的页（段）表机制 b. 缺页（段）中断机构 c. 地址变换机构

5. 实现虚拟存储器需要哪几个关键技术？

答：(1)在分页诸求系统中是在分页的基础上，增加了诸求调页功能和页而页换功能所形成的页式虚拟存储系统。允许只装入少数页面的程序（及数据），便启动运行。(2)在诸求分段系统中是在分段系统的基础上，增加了诸求调段及分段页换功能后形成的段式虚拟存储系统。允许只装入少数段（而非所有段）的用户程序和数据，即可启动运行。

6. 在诸求分页系统中，页表应包括哪些数据项？知项的作用是什么？

答：页表应包括：页号、物理块号、状态位P、访问字段A、修改位M和外存地址。其中状态位P指示该页是否调入内存，供程序访问时参考；访问字段A用于记录本页在一段时间内被访问的次数，或最近已有多长时间未被访问，提供给页换算法选择换出页面时参考；修改位M表示该页在调入内存后是否被修改过；外存地址用于指出该页在外存上的地址，通常是物理块号，供调入该页时使用。

7. 试比较缺页中断机构与一般的中断，它们之间有何明显的区别？

答：a. 一般中断只需要保护现场然后就直接跳到需及时处理的地方。

b. 缺页中断除了保护现场之外，还要判断内存中是否有足够的空间存储所需的页或段，然后再把所需页调进来和使用。

8. 试说明诸求分页系统中的地址变换过程。

- 1) 取逻辑地址分解为页号P和页内偏移w；
- 2) 根据页号查找页表，获得该页的描述信息；
- 3) 若该页中断位为1，产生缺页中断；
- 4) 更新该页的描述信息；
- 5) 根据页块号和页内偏移W，计算物理地址。

9. 何谓固定分配局部置换和可变分配和全局页换的内存分配策略？

(1) 固定分配局部置换

固定分配是指，为每个进程分配一组固定数目的物理块，在进程运行期间不再改变。

局部置换是指，如果进程在运行中发现缺页，则只能从分配给该进程的n个页面中，选出一页换出，然后再调入一页。

(2) 可变分配全局置换

可变分配是指，先为每个进程分配一定数目的物理块，在进程运行期间，可根据情况做适当地改变。

全局置换是指，如果进程在运行中发现缺页，则将OS所保留的空闲物理块或者以所有进程的全部物理块为标的，选择一块换出，然后将所缺之页调入。

10. 在请求分页系统中，应从何处将所需页面调入内存？

答：请求分页系统中的缺页从何处调入内存分三种情况：

(1) 系统拥有足够对换区空间时，可以全部从对换区调入所需页面，提高调页速度。在进程运行前将与该进程有关的文件从文件区拷贝到对换区。

(2) 系统缺少足够对换区空间时，不被修改的文件直接从文件区调入；当换出这些页面时，未被修改的不必换出，再调入时，仍从文件区直接调入。对于可能修改的，在换出时调入对换区，以后需要时再从对换区调入。

(3) UNIX方式。未运行页面从文件区调入。曾经运行过但被换出页面，下次从对换区调入。UNIX系统允许页面共享，某进程请求的页面有可能已调入内存，直接使用不再调入。

11. 试说明在请求分页系统中页面的调入过程。

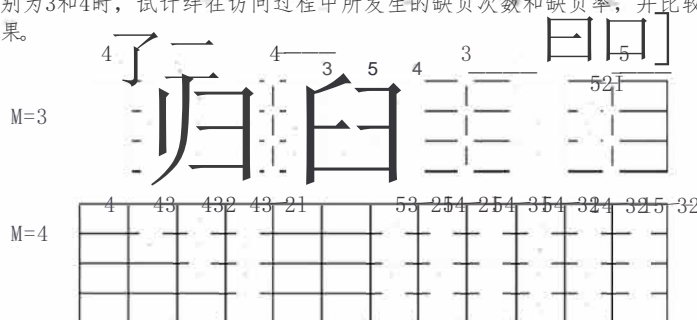
每当程序所要访问的页面未在内存时（存在位为“0”），便向CPU发出一缺页中断，中断处理程序首先保留CPU环境，分析中断原因后，转入缺页中断处理程序。该程序通过查找页表，得到该页在外存的物理块后，如果此时内存能容纳新页，则启动磁盘I/O，将所缺之页调入内存，然后修改页表。如果内存已满，则须先按照某种置换算法，从内存中选出一页准备换出；如果该页未被修改过（修改位为“0”），可不必将该页写回磁盘；但如果此页已被修改（修改位为“1”），则必须将它写回磁盘，然后再把所缺的页调入内存，并修改页表中的相应表项，置其存在位为“1”，并将此页表项写入快表中。在缺页调入内存后，利用修改后的页表，去形成所要

访问数据的物理地址，再去访问内存数据。整个页面的调入过程对用户是透明的。

12. 在请求分页系统中，应采用哪种页面置换算法？

答：常用的页面置换算法有：最佳置换法和先进先出置换算法，最近最久未使用（LRU），置换算法，Clock 置换算法，最少使用置换算法，页面缓冲约法等。

13. 在一个请求分页系统中，采用FIFO页面置换算法时，假如一个作业的页面走向为4、3、2、1、4、3、5、4、3、2、1、5，当分配给该作业的物理块数M分别为3和4时，试统计在访问过程中所发生的缺页次数和缺页率，并比较所得结果。



M=3时，采用FIFO页面置换算法的缺页次数为9次，缺页率为75%。

M=4时，采用FIFO页面置换算法的缺页次数为10次，缺页率为83%。

由此可见，附加分配给作业的内存块数，反而增加了缺页次数，提高了缺页率，这种现象被称为 Belady 现象。

14. 实现LRU算法所需的硬件支持是什么？

答：需要寄存器和栈等硬件支持。寄存器用于记录某进程在内存中各页的使用情况，栈用于保存当前使用的各个页面的页面号。

15. 试说明改进型Clock置换算法的基本原理。

答：因为修改过的页面在换出时付出的开销比未被修改过的页面大，在改进型Clock算法中，既考虑页面的使用情况，还要增加置换代价的因子：在选择页面作为淘汰页面时，把同时满足未使用过和未被修改作为首选淘汰页面。



16影响页面换进换出效率的若干因素是什么？

(1) 页面型换算法：影响页面换进换出效率最重要的因素，直接影响进程在运行过程中的缺页率，影响页面换进换出的开销。

(2) 写回磁盘的频率：如果是采取每个页面换出时，就将它写回磁盘的策略，这意味着每换出一个页面，便需要启动一次磁盘。但如果在系统中建立了一个已修改换出页面链表，对每一个要被换出的页面（已修改），系统可暂不把它们写回磁盘，而是将它们挂在已修改换出页面链表上，仅当被换出页面数目达到一定值时，再将它们一起写回到磁盘上，这样就显著地减少了磁盘I/O的操作次数。

或者说，减少已修改页面换出的开销。

(3) 读入内存的频率：在设置了已修改换出页面链表后，在该链表上就暂时有一批装有数据的页面，如果需要再次访问这些页面时，就不需从外存上调入，而直接从已修改换出页面链表中获取，这样也可以减少将页面从磁盘读入内存的频率，减少页面换进的开销。或者说，只需花费很小的开销，便可使这些页面，又回到该进程的驻留集中。

17页面缓冲算法的主要特点是什么？它是如何降低页面换进换出的频率的？

②显著地降低了页面换进、换出的频率，使磁盘I/O的操作次数大为减少，因而减少了页面换进、换出的开销；

③由于换入换出的开销大幅度减小，才能使其采用一种较简单的置换策略，如先进先出(FIFO)算法，它不需要特殊硬件的支持，实现起来非常简单。

在该系统中，内存分配策略上采用了可变分配和局部置换方式，为了能显著地降低了页面换进、换出的频率，在内存中设置了如下两个链表：

(1) 空闲页面链表：是一个空闲物理块链表，用于分配给频繁发生缺页的进程，以降低该进程的缺页率。当有一个未被修改的页要换出时，实际上并不将它换出到外存，而是把它们所在的物理块，挂在空闲链表的末尾。

(2) 修改页面链表：由已修改的页面所形成的链表。设置该链表的目的，是为了减少已修改页面换出的次数，降低将已修改页面写回磁盘的频率，以及降低将磁盘内容读入内存的频率。

18. 什么是抖动？产生抖动的原因是什么？

a. 抖动(Thrashing)就是指当内存中已无空闲空间而又发生缺页中断时，须要从内存中调出一页程序或数据送磁盘的对换区中，如果办法不当，刚被换出的页很快被访问，于是新调入，马上此页再选一页调出，而此时被换出的页很快又要被访问，因而又需将它调入，如此频繁更换页面，使得系统把大部分时间用在了页面的换进换出上，而几乎不能完成任何有效的工作，我们称这种现象为“抖动”。  
b. 产生抖动的原由是由于CPU的利用率和多道程序的对立统一矛盾关系引起的，为了提高CPU利用率，提高多道程序度，但单纯提高多道程序度又会造成缺页率的急剧上升，导致CPU的利用率下降，而系统的调度和序又会对提高CPU利用率而继续提高多道程序度，形成恶性循环，我们称这时的进程是处于抖动状态。

19何谓工作集？它是基于什么原理确定的？

答：工作集(或驻留集)是指在某段时间间隔内，进程要访问的页面集合。经常，新被使用的页面需要在工作集中，而长期不被使用的页面要从工作集中被丢弃。为了防止系统出现抖动现象，需要选择合适的工作集大小。工作集模型的原理是：让操作系统跟踪每个进程的工作集，并为进程分配大于其工作集的物理块。如果还有空闲物理块，则可以再调一个进程到内存以增加多道程序度。如果所有工作集之和增加以至于超过了可用物理块的总数，那么操作系统会暂停一个进程，并将其页面调出并且将其物理块分配给其他进程，防止出现抖动现象。正确选择了工作集的大小，对存储器的利用率和系统吞吐量的提高，都将产生重要影响。

20. 当前可以利用哪几种方法来防止“抖动”？

(1) 采取局部置换策略 (2) 把工作集算法融入到处理机调度中

(3) 利用“L-S”准则调节缺页率 (4) 选择暂停的进程



21 试说明如何利用 “ $L=S$ ” 准则来调节缺页率，以避免“抖动”的发生？P 173

于1980年Denning提出了 “ $L=S$ ” 的准则，来调节多道程序度，其中 $L$ 是缺页之间的平均时间， $S$ 是平均缺页服务时间，即用于置换一个页面所需的时间。利用 “ $L=S$ ” 准则，对于调节缺页率是十分有效的。

22 为了实现请求分段式存储管理，应在系统中增加配哪些硬件结构？

答：请求段表机制、缺段中断机制和地址变换机构。

23 在请求段表机制中，应设置哪些段表项？

段名	1段长	：	1：1：\$f	补	外存始址
----	-----	---	---------	---	------

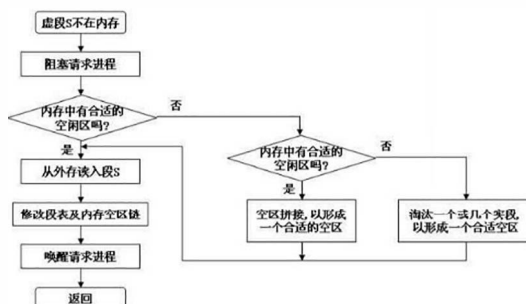
存取方式，访问字段A，修改1!fM，存在位P，填补位，外存始址。

24 说明请求分段系统中的缺页中断处理过程。

答：请求分段系统中的缺页中断处理过程描述如下：

- (1) 根据当前执行指令中的逻辑地址查页表，判断该页是否在主存储器中
- (2) 该页标志为“0”形成缺页中断，中断装置通过交换PSW让操作系统的中断处理程序占用处理器。
- (3) 操作系统处理缺页中断处理的办法是查主存分配表找一个空闲的主存块，查页表找出该页在磁盘上的位置，启动磁盘读出该页信息。
- (4) 把从磁盘上读出的信息装入找到的主存块中。
- (5) 当页面住处被装入主存后，应修改页表中对应的表目，项上该页所占用的主存块把标志置为“1”，表示该页已在主存储器中
- (6) 由于产生缺页中断时的那条指令并没执行完，所以在把页面装入之后应重新执行被中断指令。

请求分段系统中的缺页中断处理过程如下图所示：



25 请对共享段表中的各项作简要说明。

在系统中配置一张共享段表，所有各共享段都在共享段表中占有一表项。在表项的上面记录了共享段的段号、段长、内存始址、状态（存在）位、外存始址以及共享计数等信息。接下去就是记录了共享此分段的每个进程的情况。

段长	内存始址	状态	外存始址
共享进程记录count			

六手我可

0 共享进程计数count: 记录有多少进程正在共享该分段。

@存取控制字段：对于一个共享段，应为不同的进程赋予不同的存取权限。

@段号：每个进程可用自己进程的段号，去访问该共享段。

26如何实现共享分段的分配和回收？

◎共享段的分配 在为共享段分配内存时，对第一个请求使用该共享段的进程，由系统为该共享段分配一物理区，当又有其它进程需要调用该共享段时，无须再为该段分配内存。

◎共享段的回收 当共享此段的某进程不再需要该段时，若无其他进程使用该段，则由系统回收该共享段的物理内存，否则只是取消调用者进程在共享段表中的有关记录。

## 第六章

1. 试说明I/O系统的基本功能。

答：a. 隐藏物理设备的细节 b. 与设备的无关性 c. 提高处理机和I/O设备的利用率 d. 对I/O设备进行控制 e. 确保对设备的正确共享 f. 错误处理

2. 简要说明I/O软件的4个层次的基本功能。

答：中断处理程序：用于保存被中断进程的CPU环境，转入相应的中断处理程序进行处理，处理完后恢复现场，并返回到被中断的进程

设备驱动程序：与硬件直接有关，用来具体实现系统对设备发出的操作指令，驱动I/O设备工作

设备独立性软件：用于实现用户程序与设备驱动器的统一接口、设备命令、设备保护，以及设备分在与释放等。

用户层I/O软件：用于实现用户与I/O设备交互

3. I/O系统接口与软件/硬件(RW/HW)接口分别是什么接口？

答：I/O系统接口是I/O系统与上层系统之间的接口，向上层提供对设备进行操作的抽象I/O命令，以方便高层对设备的使用；软件/硬件(RW/HW)接口的上面

是中断处理程序何用于不同设备的设备驱动程序，它的下面是各种设备的控制器。

4. 与设备无关性的基本含义是什么？为什么要设置该层？

答：为了提高OS的可适应性和可扩展性，在现代OS中都毫无例外地实现了设备独立性，也称设备无关性。基本含义：应用程序独立于具体使用的物理设备。为了实现设备独立性而引入了逻辑设备和物理设备两概念。在应用程序中，使用逻辑设备名称来请求使用某类设备；而系统在实际执行时，还必须使用物理设备名称。优点1. 设备分配时的灵活性2易于实现I/O重定向（用于I/O操作的设备可以更换（即重定向），而不必改变应用程序）。

5. 试说明设备控制器的组成。

答：设备控制器与处理机的接口；设备控制器与设备的接口；I/O逻辑。

6. 为了实现CPU与设备控制器之间的通信，设备控制器应该具备哪些功能？

答：基本功能：接收和识别命令；数据交换；标识和报告设备的状态；地址识别；数据缓冲；差值控制。

7. 什么是内存映像I/O？它是如何实现的？ P186

8为什么说中断是OS赖以生存的基础？

答：中断在操作系统中有特殊重要的地位，它是多道程序得以实现的基础，没有中断，就不可能实现多道程序，因为进程之间的切换是通过中断来完成的。另一方面，中断也是设备管理的基础，为了提高处理机的利用率和实现CPU和I/O设备并行执行，也必需有中断的支持。中断处理程序是I/O系统中最低的一层，它是整个I/O系统中最低的一层。

9对中断源的两种处理方式分别用于那种场合？

答：1)屏蔽（禁止）中断：当处理机正在处理一个中断时，将屏蔽掉所有的中断，且到处理机已处理完本次中断，再去检查是否有中断产生。所有中断按顺序处理，优点是简单，但不能用于实时性要求较高的中断请求。

2)嵌套中断：在设置了中断优先级的系统中，当同时有多个不同优先级的中断请求，CPU优先响应优先级最高的中断请求，高优先级的中断请求可以抢占正在运行的低优先级中断的处理机。

10设各中断处理程序通常要完成哪些工作？

答：1、唤醒被阻塞的驱动进程。2、保护被中断进程的CPU环境。3、转入相应的设备处理程序。4、中断处理。5、恢复被中断进程的现场。

11 简要说明中断处理程序对中断进行处理的几个步骤。

答：1、测定是否有未响应的中断信号 2、保护被中断进程的CPU环境  
3、转入相应的设备处理程序 4、中断处理 5、恢复CPU的现场并退出中断

12试说明设备驱动程序具有哪些特点

- (1)将接收到的抽象要求转为具体要求；
- (2)检查用户I/O请求合法性，了解I/O设备状态，传递有关参数，设置设备工作方式；
- (3)发出I/O命令，启动分配到的I/O设备，完成指定I/O操作；
- (4)及时响应由控制器或通道发来的中断请求，根据中断类型调用相应中断处理程序处理；
- (5)对于有通道的计算机，驱动程序还应该根据用户I/O请求自动构成通道程序。

13设各驱动程序通常要完成哪些工作？

答：(1)将接收到的抽象要求转为具体要求； (2)检查用户I/O请求合法性，了解I/O设备状态，传递有关参数，设置设备工作方式； (3)发出I/O命令，启动分配到的I/O设备，完成指定I/O操作； (4)及时响应由控制器或通道发来的中断请求，根据中断类型调用相应中断处理程序处理； (5)对于有通道的计算机，驱动程序还应该根据用户I/O请求自动构成通道程序。

14. 简要说明设备驱动程序的处理过程可分为哪几步。

答：1)将抽象要求转换为具体要求；2)对服务请求进行校验；3)检查设备的状态；4)传送必要的参数。

15 试说明I/O控制发展的主要推动因素是什么？

促使I/O控制不断发展的几个主要因素如下：

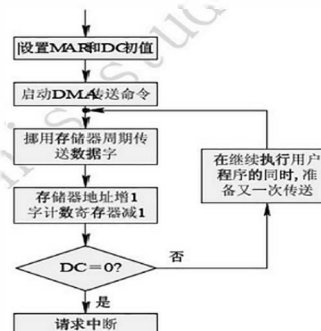
- a. 尽量减少CPU对I/O控制的干预，把CPU从繁杂的I/O控制中解脱出来，以便更多地去完成数据处理任务。
- b. 缓和CPU的高速性和设备的低速性之间速度不匹配的矛盾，以提高CPU的利用率和系统的吞吐率。
- c. 提高CPU和I/O设备操作的并行程度，使CPU和I/O设备都处于忙碌状态，从而提高整个系统的资源利用率和系统吞吐量。

16. 有哪几种I/O控制方式？各适用于何种场合？

I/O控制方式：程序I/O方式、中断驱动I/O控制方式、DMA I/O控制方式、I/O通道控制方式。程序I/O方式适用于早期的计算机系统中，并且是无中断的计算机系统；中断驱动I/O控制方式是普遍用于现代的计算机系统中；DMA I/O控制方式适用于I/O设备为块设备时在和主机进行数据交换的一种I/O控制方式；当I/O设备和主机进行数据交换是一组数据块时通常采用I/O通道控制方式，但此时要求系统必须配置相应的通道及通道控制器。

17. 试说明DMA的工作流程。P197

答：以从磁盘读入数据为例，说明DMA的工作流程。当CPU要从磁盘读入数据块时，先向磁盘控制器发送一条读命令。该命令被送到命令寄存器CR中。同时还发送本次要读入数据的内存起始目标地址，送入伪地址寄存器MAR；本次要读数据的字节数送入数据计数器DC，将磁盘中的源地址直接送到DMA控制器的I/O控制逻辑上。然后启动DMA控制器传送数据，以后CPU便处理其它任务。整个数据传送过程由DMA控制器控制。下图为DMA方式的工作流程图。



18. 为什么要引入与设备的无关性？如何实现设备的独立性？

答：引入设备独立性，可使应用程序独立于具体的物理设备，是设备分配具有灵活性。另外容易实现I/O重定向。为了实现设备独立性，必须在设备驱动程序之上设置一层设备独立性软件，用来执行所有I/O设备的公用操作，并向用户层软件提供统一接口。关键是系统中必须设置一张逻辑设备表LUT用来进行逻辑设备到物理设备的映射，其中每个表目中包含了逻辑设备名、物理设备名和设备驱动程序入口地址三项；当应用程序用逻辑设备名请求分配I/O设备时，系统必须为它分配相应的物理设备，并在LUT中建立一个表目，以后进程利用该逻辑设备名请求I/O操作时，便可从LUT中得到物理设备名和驱动程序入口地址。

19. 与设备的无关的软件中，包括了哪些公有操作的软件？

答：1、设备驱动程序的统一接口2、缓冲管理3、差错控制4、对独立设备的分配与回收5、独立于设备的逻辑数据块

20. 在考虑到设备的独立性时，应如何分配独占设备？

- (1) 进程以逻辑设备名提出I/O请求。
- (2) 根据逻辑设备表相应表项获得I/O请求的逻辑设备对应类型的物理设备在系统设备表中的指针。
- (3) 从指针所指位置起顺序检索系统设备表，直到找到一个属于对应I/O请求所用类型、空闲可用且满足设备分配安全性算法验证为安全分配的设备的设备控制表，将对应设备分配给请求进程；如果未找到安全可用的空闲设备，则把请求进程的进程控制块挂到相应类型设备的等待队列上等待唤醒和分配。
- (4) 系统把设备分配给I/O请求进程后，再到该设备的设备控制表中找出与其相连接的控制器控制器控制表，根据其状态字段判断该控制器是否忙碌，若忙则把请求进程的进程控制块挂到该控制器的等待队列上；否则将该控制器分配给进程。
- (5) 系统把控制器分配给I/O请求进程后，再到该控制器的控制器控制表中找出与其相连接的通道的通道控制表，根据其状态字段判断该通道是否忙碌，若忙则把请求进程的进程控制块挂到该通道的等待队列上；否则将该通道分配给进程。
- (6) 只有在设备、控制器和通道三者都分配成功时，这次的设备分配才算成功，然后便可启动设备进行数据传送。

21. 何谓设备虚拟？实现设备虚拟式所依赖的关键技术是什么？

答：通过虚拟技术可将一台独占设备变换成若干台逻辑设备，供若干个用户（进程）同时使用，通常把这种经过虚拟技术处理后的设备称为虚拟设备。其实现所依赖的关键技术是SPOOLING技术。

22. 在实现后台打印时，SPOOLing系统应为请求I/O的进程提供哪些服务？

1、由输出进程在输出井中为之申请一空闲盘块区，并将要打印的数据送入其中；  
2、输出进程和为用户进程申请一张空白的用户打印表，并将用户的打印要求填入其中，并将该表挂到请求打印队列上。  
3、一旦打印机空闲，输出进程便从请求打印队列的队首取出一张请求打印表，根据表中的要求将要打印的数据从输出井传送到内存缓冲区，再由打印机进行打印。

23. 假脱机系统向用户提供共享打印机的根本思想是什么？

答：对每个用户而言，系统并非即时执行其程序输出数据的真实打印操作，而只是即时将数据输出到缓冲区，这时的数据并未真正被打印，只是让用户感觉系统已为他打印；真正的打印操作，是在打印机空闲且该打印任务在等待队列中已排到队首时进行的；以上过程是对用户隐蔽的，用户是不可见的。

24. 引入缓冲的主要原因是什么？

答：缓和CPU与I/O设备之间速度不匹配的矛盾；减少对CPU依中断频率；放宽对中断响应时间的限制；解决数据力度不匹配的问题；提高CPU和I/O设备之间的并行性。

25. 在单缓冲情况下，为什么系统对一块数据的处理时间为  $\max(C, T) + M$ ？

在块设备输入时，假定从磁盘把一块数据输入到缓冲区的时间为T，操作系统将缓冲区数据传送给用户区的时间为M；而CPU对这一块数据进行计算得时间为C。在单缓冲情况下，由于设备的输入操作和CPU的处理操作可以并行，所以系统对每一整块数据的处理时间为  $\max(C, T) + M$ 。

26. 为什么在双缓冲情况下，系统对一块数据的处理时间为  $\max(C, T)$ ？

该方式又称缓冲对换方式，在设备输入时，先将数据送入第一缓冲区，装满后便转向第二缓冲区。此时操作系统可以从第二缓冲区移出数据，升送入用户进程。接由CPU对数据进行计算。在双缓冲区中，不仅设备的输入操作和CPU的处理操作可以并行，设备的输入操作和数据的传送操作也可以并行，因此耗时大约为  $\max(C+T, T)$ 。考虑到M是内存中数据块的“搬家”耗时，非常短暂可以省略，因此近似地认为是： $\max(C, T)$ 。

27. 试绘图说明把多缓冲用于输出时的情况。



28. 试说明收容输入工作缓冲区和提取输出工作缓冲区的工作情况。

答：@收容输入工作缓冲区的工作情况为：在输入进程需要输入数据时，调用GetBuf (EmptyQueue)过程，从EmptyQueue队列的队首摘下一个空缓冲区，作为收容输入工作缓冲区扣入，然后把数据输入其中，装满后再调用PutBuf (InputQueue, Hin)过程，将该缓冲区挂在输入队列InputQueue的队尾。  
@提取输出工作缓冲区的工作情况为：当要输出数据时，调用GetBuf (OutputQueue)过程，从输出队列的队首取得一装满输出数据的缓冲区作为提取输出工作缓冲区Sout。在数据提取完后，再调用PutBuf (EmptyQueue, Sout)过程，将该缓冲区挂到空缓冲队列EmptyQueue的队尾。



29. 何谓安全分配方式和不安全分配方式？

答：CD安全分配方式是指当进程发出I/O请求后，便进入阻塞状态，直到其I/O操作完成时才被唤醒。在采用这种分配策略时，一旦进程已获得某种设备资源后便阻塞，使它不可能再请求任何资源，而在它运行时又不保持任何资源。这种分配方式已经摒弃了造成死锁的“请求和保持”条件，分配是安全的。缺点是进程进展缓慢，CPU与I/O设备出行工作。

②不安全分配方式是指进程发出I/O请求后仍继续执行，需要时又可发出第二个I/O请求、第三个I/O请求。仅当进程请求的设备已被另一个进程占有时，进程才进入阻塞状态。优点是几个进程可同时操作多个设备，进程推进迅速。缺点是分配不安全，可能具有“请求和保持”条件，可能造成死锁。因此，在设备分配程序中需增加一个功能，用于对本次的设备分配是否会发生死锁进行安全性计算，仅当计算结果表明分配安全的情况下才进行分配。

30. 磁盘访问时间由哪几部分组成？各部分时间应如何计算？

答：磁盘访问时间由寻道时间 $T_s$ 、旋转延迟时间 $T_r$ 、传输时间 $T_t$ 三部分组成。

(1)  $T_s$ 是启动磁臂时间 $s$ 与磁头移动 $n$ 条磁道的时间和，即 $T_s = m \times n + s$ 。  
(2)  $T_r$ 是指定扇区移动到磁头下面所经历的时间。硬盘15000r/min时 $T_r$ 为2ms；软盘300或600r/min时 $T_r$ 为50~100ms。  
(3)  $T_t$ 是指数据从磁盘读出或向磁盘写入经历的时间。 $T_t$ 的大小与每次读/写的字节数 $b$ 和旋转速度有关： $T_t = b/tN$ 。

31. 目前常用的磁盘调度算法有哪几种？各种算法优先考虑的问题是什么？

答：目前常用的磁盘调度算法有先来先服务、最短寻道时间优先及扫描等算法。

(1) 先来先服务算法优先考虑进程请求访问磁盘的先后次序；

(2) 最短寻道时间优先算法优先考虑要求访问的磁道与当前磁头所在磁道距离是否最近；

(3) 扫描算法考虑欲访问的磁道与当前磁道间的距离，更优先考虑磁头当前的移动方向。

## 第七章

1. 何谓数据项、记录和文件？

a. 数据项是最低级的数据组织形式，可分为基本数据项和组合数据项。基本数据项是用于描述一个对象某种属性的字符集，是数据组织中可以命名的最小逻辑数据单位，即原子数据，又称为数据元素或字段。组合数据项则由若干个基本数据项构成。

b. 记录是一组相关数据项的集合，用于描述一个对象某方面的属性。

c. 文件是指有创建者所定义的、具有文件名的、一组相关信息的集合。

2. 文件系统的模型可分为三层，试说明其每一层所包含的基本内容。

答：【P224图7-2】

(1) 最低层为对象及其属性说明，主要包括物理文件相关功能，包括文件和目录、磁盘存储空间等对象。

(2) 中间层是对对象进行操纵和管理的软件集合，是文件系统的核心部分，主要是逻辑文件相关功能。包括文件存储空间管理、文件目录管理、逻辑文件到物理文件的映射、文件读写管理及文件共享与保护等多种功能。

(3) 最高层是文件系统提供给用户的接口，分为命令接口、图形化用户接口、程序接口(C语言函数形式)和等三种类型。

3. 试说明用户可以对文件施加的主要操作有哪些？

答：【P225 7.1.4】要特别注意文件的打开和关闭操作，知道为什么要这样做。

6. 何谓文件逻辑结构？何谓文件的物理结构？

文件的逻辑结构是指从用户的观点出发所观察到的文件组织形式，也就是用户可以直接处理的数据及其结构，它独立于物理特性；而文件的物理结构则是指文件在外存上的存储组织形式，与存储介质的存储性能有关。

8. 如何提高对变长记录顺序文件的检索速度?

为了提商对变长记录顺序文件的检索速度, 可为其建立一张索引表, 以主文件中知条记录的长度及指向对应记录的指针 (即该记录在逻辑地址空间的首址) 作为相应每个表项的内容。由于索引表本身是一个定长记录的顺序文件, 若将其按记录键排序, 则可以实现对主文件的方便快速的直接存取。需要指出的是, 如果文件较大, 应通过建立分组多级索引以进一步提高检索效率。

12. 试说明关于索引文件和索引顺序文件的检索方法。

答: P230

@对索引文件进行检索时, 首先根据用户 (程序) 提供的关键字, 并利用某种 (折半查找) 算法检索索引表, 从中找到相应的表项; 再利用该表项中给出的指向记录的指针值, 去访问对应的记录。

@对索引顺序文件结合了索引和顺序查找, 适合于巨量数据的查找, 它将数据分组建索引 (以减少索引表的长度), 首先利用用户 (程序) 提供的关键字以及某种查找方法, 去检索索引表, 找到该记录所在记录组中的第一条记录的表项, 然后在组内进行顺序查找, 由于组内的数据量不多, 所以组内顺序查找开销很小。有关效率的分析见P212, 这种分组索引的方式, 对检索巨量数据是很有效的。

13. 试从检索速度和存储费用两方面对索引文件和索引顺序文件进行比较。

答: P231

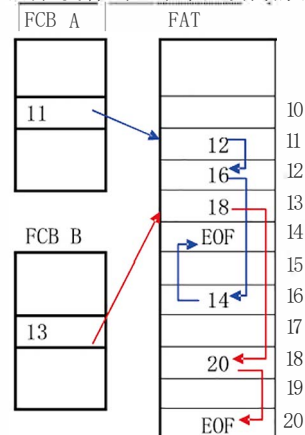
假设主文件拥有N条记录。

对索引文件, 主文件的每条记录均需配置一个索引项, 故索引存储开销为N,

而为检索到具有指定关键字的记录, 对分查找平均需要查找约  $\log_2(n)$  条记录。

对索引顺序文件, 假如为每100个记录分组配置一个索引项, 故索引存储开销为  $\log_2(n/100) + 100/2$  条记录; 而为检索到具有指定关键字的记录, 平均需要查找  $N/200 + 50$  条记录。对于更大的数据量, 还可以两级索引顺序文件。

// 在MS-DOS中有两个文件A和B, A占用11, 12, 16和14四个盘块; B占用13, 18和20三个盘块。试画出在文件A和B中个盘块间的链接情况及FAT的情况。



II// 假定一个文件系统的组织方式与MS-DOS相似, 在FAT中可有64K个指针, 磁盘的盘块大小为512B, 试问该文件系统能否索引一个512MB的磁盘?

解:  $512MB/512B=1M$ 个盘块, 而每个盘块都应有个指针来指示, 所以应该有1M个指针, 因此若有64K个指针则不能索引一个512MB的磁盘。

II// 为了快速访问, 又易于更新, 当数据为以下形式时, 应选何种文件组织方式。

(I) 不经常更新, 经常随机访问; 顺序结构

- (2) 经常更新, 经常按一定顺序访问: ——索引顺序结构  
 (3) 经常更新, 经常随机访问: ——索引结构

II / 在UNIX中, 如果一个盘块的大小为 1KB, 每个盘块号占 4 个字节, 即每个块可放 256 个地址。请转换下列文件的字节偏移量为物理地址。

① 9999:      ② 18000:      ③ 420000

盘块大小为 1KB, 盘块号占 4B, 即每个盘块最多可存放 256 个盘块号。又根据 UNIX 系统中采用的混合索引分配方式可知:

$999911024 = 9 \text{ 余 } 783$

$1800011024 = 17 \text{ 余 } 592$

$42000011024 = 410 \text{ 余 } 160$

14. 对目录管理的主要要求是什么?

答: 【P232】

- a) 实现“按名存取”
- b) 提高对目录的检索速度
- c) 文件共享
- d) 允许文件重名

15. 采用单级目录能否满足对目录管理的主要要求? 为什么?

采用单级目录不能完全满足对目录管理的主要要求, 只能实现目录性最基本的功能即按名存取。由于单级目录结构采用的是在系统只配一张目录表用来记录系统中所有文件的相关信息, 因此此目录文件可能会非常大, 在查找时速度慢, 另外不允许用户文件有重名的现象, 再者由于单级目录中要求所有用户须使用相同的名字来共享同一个文件, 这样又会产生重名问题, 因此不便于实现文件共享。

16. 目前广泛采用的目录结构是哪种? 它有什么优点?

目前广泛采用的目录结构是树型目录结构。它具有以下优点:

- a) 能有效提高对目录的检索速度: 假定文件系统中有  $N$  个文件, 在单级目录中, 最多要检索  $N$  个目录项, 但对于有 1 级的树型目录, 在目录中每个目录项指定一个文件, 最多可能要检索  $\sqrt{N}$  个目录项。
- b) 允许文件重名: 由于在树型结构的文件系统中, 是利用文件路径名来检索文件的, 故允许每个用户在自己的分目录中使开与其他用户文件相同的名字。
- c) 便于实现文件共享: 在树型目录中, 用户可通过路径名来共享其他用户的文件, 也可将一个共享文件链接到自己的目录下, 从而使文件的共享变得更为方便, 其实现方式也非常简单, 系统只需在用户的目录文件中增设一个目录项, 项上用户赋予该共享文件的新文件名, 以及该共享文件的唯一标识符即可。

18. Hash 检索法有何优点? 又有何局限性?

在 Hash 检索法中, 系统利用用户提供的文件名并将它变换为文件目录的索引值, 构利用该索引值到目录中去查找, 这样能有效地提高目录的检索速度, 但 Hash 检索法也有局限性即对于使用了通配符的文件名, 系统是无法使用 Hash 检索法检索目录的。

19. 在 HASH 检索法中, 如何解决“冲突”?

答: 【P240】

20. 试说明在树形目录结构中线性检索法的检索过程, 并给出相应的流程图。

答: 【P241】可以把它当成“数据结构”的一道应用题思考。

///. 有一计算机系统利用图 6-33 所示的位示图来管理空闲盘块。盘块的大小为 1KB, 现要为某文件分配 10 个盘块, 试说明盘块的具体分配过程。

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

图 6-33 某计算机系统的位示图

分配量个盘块的过程如下：

(1) 顺序扫描位示图，从中找到第一个值为 0 的二进制位，得到其行号  $i=3$ ，列号  $j=3$ 。

(2) 将所找到的二进制位转换成与之对应的盘块号。盘块号计算公式为：  
 $b = (3-1) * 16 + 3 = 35$ ;

(3) 修改位示图，令  $\text{map}[3, 3]=1$ ，并将该盘块分配出去。

类似地，可使用相同的方法找到第二个值为 0 的二进制位，得到行号  $i=4$ ，列号  $j=7$ ，其对应的盘块号为 55，令  $\text{map}[i, j]=1$ ，并将该盘块分配出去。

///. 某操作系统的磁盘文件空间共有 500 块，若用字长为 32 位的位示图管理磁盘空间，试问：

(1) 位示图需要多少字？

(2) 第  $i$  字第  $j$  位对应的块号是多少？

(3) 给出申请/归还一块的工作流程。

$[500/32]_{\text{z}}=16$  个字

$b = (i-1) * 32 + j = 32(i-1) + j$  (b 从 1 开始计数， $i, j$  也从 1 开始计数)

根据盘块号  $b$  求出：

$i = (b-1)/32 + 1$ ;  $j = (b-1)\%32 + 1$ ;

将第  $i$  字第  $j$  位置 0

26. 基于索引结点的文件共享方式有何优点？

27. 基于符号链的文件共享方式有何优点？ P243