

操作系统引论

1、多道程序设计：P7

答：进一步提高资源的利用率和系统吞吐量，在该系统中，用户所提交的作业都先存放在外存上并排成一个队列，称为“后备队列”；然后，由作业调度程序按一定的算法从后备队列中选择若干个作业调入内存，使它们共享 CPU 和系统中各种资源。在 OS 中引入多道程序设计技术可带来以下好处：提高 CPU 利用率、可提高内存和 I/O 设备利用率、增加系统吞吐量。

2、进程：P37-P38

答：程序的一次执行。进程是一个程序及其数据在处理机上顺序执行时所发生的活动。进程是程序在一个数据集合上运行的过程，它是系统进行资源分配和调度的一个独立单位。进程是进程实体的运行过程。是系统进行资源分配和调度的一个独立单位。

3、作业：P84

答：作业是一个比程序更为广泛的概念，它不仅包含了通常的程序和数据，而且还应配有一份作业说明书，系统根据该说明书来对程序的运行进行控制。在批处理系统中，是以作业为基本单位从外存调入内存的。

4、任务：

答：从系统资源分配的角度通过进程在系统中运行。

5、文件：P204

答：文件是指由创建者所定义的、具有文件名的一组相关元素的集合，可分为有结构文件和无结构文件两种。在有结构的文件中，文件由若干个相关记录组成；而无结构文件则被看成是一个字符流。文件在文件系统中是一个最大的数据单位，它描述了一个对象集。

6、虚拟存储：P142

答：是指具有请求调入功能和置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。其逻辑容量由内存容量和外存容量之和所决定，其运行速度接近于内存速度，而每位的成本却又接近于外存。可见，虚拟存储技术是一种性能非常优越的存储器管理技术，故被广泛地应用于大、中、小型机器和微型机中。

7、接口：P22—P23

答：用户接口：它是提供给用户使用的接口，用户可通过该接口取得操作系统的服务；程序接口：它是提供给程序员在编程时使用的接口，是用户程序取得操作系统服务的惟一途径。

8、进程：P73(同第 2 题)

答：

9、系统调用：P261

答：在操作系统中提供了系统调用，使应用程序可以通过系统调用的方法，间接调用操作系统的相关过程，取得相应的服务。系统调用在本质上是应用程序请求 OS 内核完成某功能时的一种过程调用，但它是一种特殊的过程调用。

10、死锁：P103

答：是指多个进程的运行过程中因争夺资源而造成的一种僵局，当进程处于这种僵持状态时，若无外力作用，它们都将无法再向前推进。

第一章

1、设计现代 OS 的主要目标是什么？ P1

答：其主要目标是有效性、方便性、可扩充性、开放性。

2、OS 的作用可表现在哪几个方面？ P2-P3

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 实现了计算机资源的抽象？ P4

答：完全无软件的计算机系统（即裸机），它向用户提供的是实际硬件接口（物理接口），用户必须对物理接口的实现细节有充分的了解，并利用机器指令进行编程，因此该物理机器必定是难以使用的。为了方便用户使用 I/O 设备，人们在裸机上覆盖上一层 I/O 设备管理软件。通常把覆盖了上述软件的机器称为扩充机器或虚拟机。它向用户（进程）提供了一个对硬件操作的抽象模型，用户更容易地使用计算机硬件资源。由该层软件实现了对计算机硬件操作的第一个层次的抽象。为了方便用户使用文件系统，人们又在第一层软件上再覆盖上一层用于文件的管理软件，同样由它来实现对文件操作的细节，并向上提供一组对文件进行存取操作的命令，用户可利用这组命令进行文件的存取。此时用户所看到的是一台功能更强、使用更方便的虚拟机。该层软件实现了对硬件资源操作的第二个层次的抽象。OS 是铺设在计算机硬件上的多层系统软件，它们不仅增强了系统的功能，而且还隐藏了对硬件操作的细节，由它们实现了对计算机硬件操作的多个层次的抽象。值得说明的，对一个硬件在底层进行抽象后，在高层还可再次对该资源进行抽象，成为更高层的抽象模型。随着抽象层次的提高，抽象接口所提供的功能就越来越强，用户使用起来也更加方便。

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？ P4-P5

答：为了进一步提高资源的利用率和系统吞吐量，在该系统中，用户所提交的作业都先存放在外存上并排成一个队列，称为“后备队列”；然后，由作业调度程序按一定的算法从后备队列中选择若干个作业调入内存，使它们共享 CPU 和系统中各种资源。在 OS 中引入多道程序设计技术可带来以下好处：提高 CPU 利用率、可提高内存和 I/O 设备利用率、增加系统吞吐量。主要动力：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？ P6

答：由于程序和数据的输入和输出都是在外围机的控制下完成的，或者说，它们是在脱离主机的情况下进行的，该技术是脱机输入/输出方式；反之，在主机的直接控制下进行输入/输出的方式称为联机输入/输出（ON-LINE I/O）方式。1、减少了 CPU 的空闲时间；2、提高了 I/O 速度。

6、试说明推动分时系统形成和发展的主要动力是什么？ P9

答：分时系统它能很好地将一台计算机提供给多个用户同时使用，提高计算机的利用率。1、人-机交互；2、共享主机；3、便于用户上机。

7、实现分时系统的关键问题是什么？应如何解决？ P10

答：其最关键的问题是如何使用户能与自己的作业进行交互，即当用户在自己的终端上键入命令时，系统应能及时接收并及时处理该命令，再将结果返回给用户。用户可继续键入下一条命令，此即人-机交互。应强调指出，即使有多个用户同时通过自己的键盘键入命令，系统也应能全部地及时接收并处理这些命令。1、及时接收；2、及时处理；

8、为什么要引入实时 OS？ P11

答：实时系统是指系统能及时（或即时）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。1、应用需求；2、实时任务；

9、什么是硬实时任务和软实时任务？试举例说明。 P12

答：硬实时任务是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。

软实时任务是它也联系着一个截止时间，但并不严格，若偶尔错过了任务的截止时间，对系统产生的影响也不会太大。

举例说明：硬实时任务为订车票、工业；软实时任务为网页更新；

10、在 8 位微机和 16 位微机中，占据了统治地位的是什么 操作系统？ P13

答：在 8 位微机和 16 位微机中，最有代表性的单用户单任务微机操作系统是 CP/M 和 MS-MOS。

11、试列出 Windows OS 中五个主要版本，并说明它们分别较之前一个版本有何改进。 P13

答：1、WIN1.0 和 WIN2.0；2、WIN3.1 版本，针对 386 和 486 等 32 位微机开发的，较之前的操作系统有着重大的改进，引入了友善的图形用户界面，支持多任务和扩展内存的功能，使计算机更好使用，从而成为 386 和 486 等微机的主流操作系统；3、1 WIN95、WIN3.1 有许多重大改进，彩了全 32 位处理技术，并兼容了以前的 16 位应用程序，在该系统中还集成了支持 INTERENET 的网络功能。4、WIN98 把微软公司自己开发的 INTERNET 浏览器整合到系统中，大大方便了用户上网浏览，另一个特点是增加了对多媒体的支持。5、32 位版本的 WIN XP。

12、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。 P12

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确定的；而实时控制系统的及时性，则是以控制对象所要求的开始截止时间或完成截时间确定的，一般为秒级到毫秒级，甚至有的要低于 100 微秒。

2、交互性，实时信息处理系统虽然也具有交互性，但这里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。

3、可靠性，分时系统虽然也要求系统可靠，但相比之下，实时系统则要求系统具有高度的可靠性。

13、OS 有哪几个特征？其最基本的特征是什么？ P14

答：OS 有并发、共享、虚拟和异步这四个基本特征。并发特征是操作系统最重要的特征；

14、处理机管理有哪些主要功能？它们的主要任务是什么？ P18

答：主要功能：创建和撤消进程（线程），对诸进程（线程）的运行进行协调，实现进程（线

程)之间的信息交换,以及按照一定的算法把处理机分配给进程(线程)。

1、进程控制:进程控制的主要功能是为作业创建进程,撤消已结束的进程,以及控制进程在运行过程中的状态转换。2、进程同步:进程同步的主要任务是为多个进程(含线程)的运行进行协调。3、进程通信:在多道程序环境下,为了加速应用程序的运行,应系统中建立多个进程,并且再为一个进程建立若干个线程,由这些进程(线程)相互合作去完成一个共同的任务。而在这些进程(线程)之间,又往往需要交换信息。当相互合作的进程(线程)处于同一计算机系统时,通常在它们之间是采用直接通信方式,即由源进程利用发送命令直接将消息(Message)挂到目标进程的消息队列上,以后由目标进程利用接收命令从其消息队列中取出消息。4、调度:在后备队列上等待的每个作业都需经过调度才能执行(1)作业调度:作业调度的基本任务是从后备队列中按照一定的算法,选择出若干个作业,为它们分配运行所需的资源(首行是分配内存)。(2)进程调度:进程调度的任务是从进程的就绪队列中,按照一定的算法选出一个进程,把处理机分配给它,并为它设置运行现场,使进程投入执行。

15、内存管理有哪些主要功能?它们的主要任务是什么? P20

答:主要任务是为多道程序的运行提供良好的环境,方便用户使用存储器,提高存储器的利用率以及能从逻辑上扩充内存。有内存分配、内存保护、地址映射和内存扩充等功能。

1、内存分配:内存分配的主要任务是为每道程序分配内存空间,使它们“各得其所”;提高存储器的利用率,以减少不可用的内存空间;允许正在运行的程序申请附加的内存空间,以适应程序和数据动态增长的需要。2、内存保护:内存保护的主要任务是确保每道用户程序都只在自己的内存空间内运行彼此互不干扰;绝不允许用户程序访问操作系统的程序和数据;也不允许用户程序转移到非共享的其它用户程序中去执行。3、地址映射:存储器管理必须提供地址映射功能,以将地址空间中的逻辑地址转换为内存空间中与之对应的物理地址。该功能应在硬件的支持下完成。4、内存扩充:存储器管理中的内存扩充任务并非是去扩大物理内存的容量,而是借助于虚拟存储技术,从逻辑上去扩充内存容量,使用户所感觉到的内存容量比实际内存容量大得多,以便让更多的用户程序并发运行。

16、设备管理有哪些主要功能?其主要任务是什么? P21

答:主要任务是:完成用户进程提出的I/O请求;为用户进程分配其所需的I/O设备;提高CPU和I/O设备的利用率;提高I/O速度;方便用户使用I/O设备。有缓冲管理、设备分配和设备处理以及虚拟设备等功能。

1、缓冲管理:在I/O设备和CPU之间引入缓冲,提高CPU的利用率,进而提高系统吞吐量。在现代计算机系统中,都无一例外地在内存中设置了缓冲区,而且还可通过增加缓冲区容量的方法来改善系统的性能。对于不同的系统,可以采用不同的缓冲区机制。2、设备分配:设备分配的基本任务是根据用户进程的I/O请求、系统的现有资源情况以及按照某种设备的分配策略,为之分配其所需的设备。如果在I/O设备和CPU之间还存在着设备控制器和I/O通道时,还须为分配出去的设备分配相应的控制器和通道。3、设备处理:设备处理程序又称为设备驱动程序。其基本任务是用于实现CPU和设备控制器之间的通信,即由CPU向设备控制器发出I/O命令,要求它完成指定的I/O操作;反之,由CPU接收从控制器发来的中断请求,并给予迅速的响应和相应的处理。

17、文件管理有哪些主要功能?其主要任务是什么? P21-P22

答:文件管理的主要任务是对用户文件和系统文件进行管理,以方便用户使用,并保证文件的安全性。为此,文件存储空间的管理、目录管理、文件的读/写管理,以及文件的共享与

保护等功能。

1、文件存储空间的管理：其主要任务是为每个文件分配必要的外存空间，提高外存的利用率，并能有助于提高文件系统的存、取速度。2、目录管理：目录管理的主要任务是为每个文件建立其目录项，并对众多的目录项加以有效的组织，以实现方便的按名存取，即用户只须提供文件名便可对该文件进行存取。3、文件的读/写管理和保护：文件的读/写管理其该功能是根据用户的请求，从外存中读取数据，或将数据写入外存。文件保护其为了防止系统中的文件被非法窃取和破坏，在文件系统中必须提供有效的存取控制功能。

18、是什么原因使操作系统具有异步性特征？

答：进程是以人们不可预知的速度向前推进，此即进程的异步性。

第二章

1、什么是前趋图？为什么要引入前趋图？ P35

答：前趋图是一个有向无循环图，记为 DAG，用于描述进程之间招待的前后关系。可用来简单讲述进程之间的关系。

2、试画出下面四语句的前趋图： P36

S1:a:=x+y;

S2:b:=z+1;

S3:c:=a-b;

S4:w:=c+1;

3、为什么程序并发执行会产生间断性特征？ P36-P37

答：因程序在并发执行时，由于它们共享系统资源，以及为完成同一项任务而相互合作，致使在这些并发执行的程序之间，形成了相互制约的关系。

4、程序并发执行时为什么会失去封闭性和可再现性？ P37

答：程序在并发执行时，是多个程序共享系统中的各种资源，因而这些资源的状态将由多个程序来改变，致使程序的运行换去了封闭性，这样，某程序在执行时，必然会受到其它程序的影响。程序在并发执行时，由于失去了封闭性，也将导致其再失去可再现性。

5、在操作系统中为什么要引入进程的概念？它会产生什么样的影响？ P37 ? ? ? ? ?

答：因为使程序能并发执行，且为了对并发执行的程序加以描述和控制，人们引入了“进程”的概念。

6、试从动态性、并发性和独立性上比较进程和程序。 P37-P38

答：动态性进程的实质是进程实体的一次执行过程，因此，动态性是进程的最基本的特征。并发性这是指多个进程实体同存于内存中，且能在一段时间内同时运行。独立性是指进程实体是一个能独立运行、独立分配资源和独立接受调度的基本单位。凡未建立 PCB 的程序都不能作为一个独立的单位参与运行。

7、试说明 PCB 的作用，为什么说 PCB 是进程存在的唯一标志？ P41

答：进程控制块的作用是一个在多道程序环境下不能独立运行的程序（含数据），成为一个

能独立运行的基本单位，一个能与其它进程并发执行的进程。-----在进程的整个生命期中，系统总是通过 PCB 对进程进行控制的，系统是根据进程的 PCB 而不是任何别的什么而感知到该进程的存在的，所以说，PCB 是进程存在的惟一标志。

8、试说明进程在三个基本状态之间转换的典型原因。 P38

答：处于就绪状态的进程，在调度程序为之分配了处理机之后，该进程便可执行，相应地，它就由就绪状态转变为执行状态。正在执行的进程也称为当前进程，如果因分配给它的时间片已完而被暂停执行时，该进程便由执行状态又回复到就绪状态；如果因发生某事件而使进程的执行受阻，使之无法继续执行，该进程将由执行状态转变为阻塞状态。

9、为什么要引入挂起状态？该状态有哪些性质？

- a. 引入挂起状态主要是出于 4 种需要（即引起挂起的原因）：终端用户的请求，父进程请求，负荷调节的需要，操作系统的需要。
- b. 被挂起的进程是处于静止状态，并且不能直接被处理机调度。

17、为什么进程在进入临界区之前应先执行“进入区”代码？而在退出前又要执行“退出区”代码？

为了实现多个进程对临界资源的互斥访问，必须在临界区之前加一段用于检查临界资源是否正在被访问的代码，如未被访问，该进程可进入临界区对此临界资源进行访问；如正被访问，则该进程不能进入临界区访问临界资源。

在退出临界区后，执行恢复访问标志的代码为“退出区”，而在退出前执行“退出区”代码主要是为了使其它进程能再访问此临界资源。

18、同步机构应遵循哪些基本准则？为什么？

- a. 空闲让进、忙则等待、有限等待、让权等待四条准则
- b. 为实现进程能互斥地进入到自己的临界区

19、试从物理概念上说明记录型信号量 wait 和 signal。

Wait(S): 当 $S.value > 0$ 时，表示目前系统中这类资源还有可用的，执行一次 wait 操作，意味着进程请求一个单位的该类资源，是系统中可供分配的该类资源减少一个，因此描述为 $S.value := S.value - 1$ ；当 $S.value < 0$ 时，表示该类资源已分配完毕，因此进程应调用 block 原语，进行自我阻塞，放弃处理机，并插入到信号量链表 S.L 中。

Signal(S): 执行一次 signal 操作，意味着释放一个单位的可用资源，使系统中可供分配的该类资源数增加一个，故执行 $S.value := S.value + 1$ 操作。若加 1 后 $S.value \leq 0$ ，则表示在该信号量链表中，仍有等待该资源的进程被阻塞，因此应调用 wakeup 原语，将 S.L 链表中的第一个等待进程唤醒。

22、试写出相应的程序来描述图 2-17 所示的前趋图。 P54

```
a. Var a, b, c, d, e, f, g, h; semaphore:= 0, 0, 0, 1, 0, 0, 0, 0;
begin
  parbegin
    begin S1; signal(a); signal(b); end;
    begin wait(a); S2; signal(c); signal(d); end;
    begin wait(b); S3; signal(e); end;
    begin wait(c); S4; signal(f); end;
    begin wait(d); S5; signal(g); end;
    begin wait(e); S6; signal(h); end;
```

```

begin wait(f); wait(g); wait(h); S7; end;
parend
end

```

23. 在生产者—消费者问题中，如果缺少了 `signal(full)` 或 `signal(empty)`，对执行结果将会有何影响？

如果缺少了 `signal(full)`，那么表明从第一个生产者进程开始就没有对信号量 `full` 值改变，即使缓冲池存放的产品已满了，但 `full` 的值还是 0，这样消费者进程在执行 `wait(full)` 时会认为缓冲池是空的而取不到产品，那么消费者进程则会一直处于等待状态。

如果缺少了 `signal(empty)`，例如在生产者进程向 `n` 个缓冲区投满产品后消费者进程才开始从中取产品，这时 `empty=0`，`full=n`，那么每当消费者进程取走一个产品时 `empty` 并没有被改变，直到缓冲池中的产品都取走了，`empty` 的值也一直是 0，即使目前缓冲池有 `n` 个空缓冲区，生产者进程要想再往缓冲池中投放产品会因申请不到空缓冲区而被阻塞。

24. 试修改下面生产者—消费问题中，如果将两个 `wait` 操作即 `wait(full)` 和 `wait(mutex)` 互换位置，或者将 `signal(mutex)` 与 `signal(full)` 互换位置，结果会如何？

答：a. `wait(full)` 和 `wait(mutex)` 互换位置后，因为 `mutex` 在这儿是全局变量，执行完 `wait(mutex)`，则 `mutex`

赋值为 0，倘若 `full` 也为 0，则该生产者进程就会转入进程链表进行等待，而生产者进程会因全局变量

`mutex` 为 0 而进行等待，使 `full` 始终为 0，这样就形成了死锁。

b. 而 `signal(mutex)` 与 `signal(full)` 互换位置后，从逻辑上来说应该是一样的。

25. 我们为某临界资源设置一把锁 `W`，当 `W=1` 时表示关锁；当 `W=0` 时表示锁已打开，试写出开锁和关锁原语，并利用它们去实现互斥。

整型信号量：lock(W): while W=1 do no-op

`W:=1;`

`unlock(W): W:=0;`

记录型信号量：lock(W): `W:=W+1;`

`if(W>1) then block(W.L)`

`unlock(W): W:=W-1;`

`if(W>0) then wakeup(W.L)`

例子：

`Var W:semaphore:=0;`

`begin`

`repeat`

`lock(W);`

`critical section`

`unlock(W);`

`remainder section`

`until false;`

`end`

26. 试修改下面生产者——消费者问题解法中的错误：

```

producer:
  begin
    repeat
      .....
      produce an item in nextp;
      wait(mutex);
      wait(full);
      buffer(in):=nextp;.....
      signal(mutex);
    until false;
  end

```

```

consumer:
  begin
    repeat
      .....
      wait(mutex);
      wait(empty);
      nextc:=buffer(out);
      out:=out+1;
      signal(mutex);
      consume item in nextc;
    until false;
  end

```

27、试利用记录型信号量写出一个不会出现死锁的哲学家进餐问题的算法。 P61

答: Var chopstick:array[0,...,4] of semaphore;

所有信号量均被初始化为 1, 第 i 位哲学家的活动可描述为:

```

Repeat
  Wait(chopstick[i]);
  Wait(chopstick[(i+1) mod 5]);
  :
  Eat;
  :
  Signal(chopstick[i]);
  Signal(chopstick[(i+1) mod 5])
  Eat;
  :
  Think;
Until false;

```

28. 在测量控制系统中的数据采集任务时,把所采集的数据送往一单缓冲区;计算任务从该单缓冲区中取出数据进行计算。试写出利用信号量机制实现两任务共享单缓冲区的同步算法。

a. Var mutex, empty, full: semaphore:=1, 1, 0;

gather:	
begin	gather data in nextp;
repeat	wait(empty);

```

        wait(mutex);
        buffer:=nextp;
        signal(mutex);
        signal(full);
    until false;
end
compute:
begin
    repeat
        b. Var empty, full: semaphore:=1, 0;
        gather:
        begin
            repeat
                .....
                gather data in nextp;
                wait(empty);
                buffer:=nextp;
                signal(full);
            until false;
        end
    repeat
        .....
        wait(full);
        nextc:=buffer;
        signal(empty);
        compute data in nextc;
    until false;
end
end

```

33. 试比较进程间的低级通信工具与高级通信工具。

用户用低级通信工具实现进程通信很不方便，因为其效率低，通信对用户不透明，所有的操作都必须由程序员来实现，而高级通信工具则可弥补这些缺陷，用户可直接利用操作系统所提供的一组通信命令，高效地传送大量的数据。

36. 为什么要在 OS 中引入线程？

在 OS 中引入进程的目的，是为了使多个程序能并发执行，以提高资源利用率和系统吞吐量。在 OS 中再引入线程，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。

38. 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。

- 调度性。在传统的操作系统中，拥有资源的基本单位和独立调度、分派的基本单位都是进程，在引入线程的 OS 中，则把线程作为调度和分派的基本单位，而把进程作为资源拥有的基本单位；
- 并发性。在引入线程的 OS 中，不仅进程之间可以并发执行，而且在一个进程中的多个线程之间，亦可并发执行，因而使 OS 具有更好的并发性；
- 拥有资源。无论是传统的操作系统，还是引入了线程的操作系统，进程始终是拥有资源的一个基本单位，而线程除了拥有一点在运行时必不可少的资源外，本身基本不拥有系统资源，但它可以访问其隶属进程的资源；
- 开销。由于创建或撤销进程时，系统都要为之分配和回收资源，如内存空间等，进程切换时所要保存和设置的现场信息也要明显地多于线程，因此，操作系统在创建、撤消和切换进程时所付出的开销将显著地大于线程。

第三章

1、高级调度与低级调度的主要任务是什么？为什么要引入中级调度？ P84 P86 P87

答：高级调度其主要功能是根据某种算法，把外存上处于后备队列中的那些作业调入内存，也就是说，它的调度对象是作业。低级调度其主要功能是保存处理机的现场信息；按某种算法先取进程；把处理器分配给进程。中级调度：引入中级调度的主要目的为了提高内存利用率和系统吞吐量。为此，应使那些暂时不能运行的进程不再占用宝贵的内存资源，而将它们调至外存上去等待，把此时的进程状态称为就绪驻外存状态或挂起状态。

2、何谓作业、作业步和作业流？ P84

答：作业：作业是一个比程序更为广泛的概念，它不仅包含了通常的程序和数据，而且还应配有一份作业说明书，系统根据该说明书来对程序的运行进行控制。在批处理系统中，是以作业为基本单位从外存调入内存的。作业步：通常，在作业运行期间，每个作业都必须经过若干个相对独立，又相互关联的顺序加工步骤才能得到结果。作业流：若干个作业进入系统后，被依次存放在外存上，这便形成了输入的作业流；在操作系统的控制下，逐个作业进程处理，于是便形成了处理作业流。

3、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？ P85

答：为了管理和调度作业，在多道批处理系统中为每个作业设置了一个作业控制块。在 JCB 中所包含的内容因系统而异，通常应包含的内容有：作业标识、用户名称、用户帐户、作业类型（CPU 繁忙型、I/O 繁忙型、批量型、终端型）、作业状态、调度信息（优先级、作业已运行时间）、资源需求（预计运行时间、要求内存大小、要求 I/O 设备的类型和数量等）、进入系统时间、开始处理时间、作业完成时间、作业退出时间、资源使用情况等。

4、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？ P85

答：作业调度每次要接纳多少个作业进入内存，取决于多道程序度，即允许多少个作业同时在内存中运行。应将哪些作业从外存调入内存，这将取决于所采用的调度算法。最简单的是先来先服务调度算法，这是把将最早进入外存的作业最先调入内存；较常用的一种算法是短作业优先调度算法，是将外存上最短的作业最先调入内存；另一个较常的是基于作业优先级的调度算法，该算法是将外存上优先级最高的作业优先调入内存；

5、试说明低级调度的主要功能。 P86

答：1、保存处理机的现场信息：在进程调度进行调度时，首先需要保存当前进程的处理机的现场信息，将它们送入该进程的进程控制块（PCB）中的相应单元。2、按某种算法选取进程：低级调度程序按某种算法如优先数

算法、轮转法等，从就绪队列中选取一个进程，把它的状态改为运行状态，并准备把处理机分配给它。3、把处理器分配给进程：由分派程序把处理器分配给进程。此时需为选中的进程恢复处理机现场，即把选中进程的进程控制块内有关处理机现场的信息装入处理器相应的各个寄存器中，把处理器的控制权交给该进程，让它从取出的断点处开始继续运行。

6、在抢占调度方式中，抢占的原则是什么？ P87

答：抢占调度方式是 1、优先权原则：通常是对一些重要的和紧急的作业赋予较高的优先权。2、短作业（进程）优先原则：当新到达的作业（进程）比正在执行的作业（进程）明显的短时，将轻暂停当前长作业（进程）的执行，将处理机分配给新到的短作业（进程），使之优先执行；或者说，短作业（进程）或以抢占当前较长作业（进程）的处理机。时间片原则：各进程按时间片轮流运行，当一个时间片用完后，便停止该进程的执行而重新进行调度。

10. 试比较 FCFS 和 SPF 两种进程调度算法

相同点：两种调度算法都是既可用于作业调度，也可用于进程调度；

不同点：FCFS 调度算法每次调度都是从后备队列中选择一个或是多个最先进入该队列的作业，将它们调入内存，为它们分配资源，创建进程，然后插入到就绪队列中。该算法有利于长作业/进程，不利于短作业/进程。

SPF 调度算法每次调度都是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，将它们调入内存中运行。该算法有利于短作业/进程，不利于长作业/进程。

15. 按调度方式可将实时调度算法分为哪几种？

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。

18、何谓死锁？产生死锁的原因和必要条件是什么？ P103 P103 P105

答：所谓死锁是指多个进程在运行过程中因争夺资源而造成的一种僵局，当进程处于这种僵持状态时，若无外力作用，它们都将无法再向前推进。产生死锁的原因为竞争资源和进程间推进顺序非法。其必要条件为互斥条件，请求和保持条件，不剥夺条件，环路等待条件。

19、解决死锁问题的几个方法中，哪种方法最易于实现？哪种方法使资源利用率高？ P105 P106

答：预防死锁；解除死锁；

20、请详细说明可通过哪些途径预防死锁。 P106

答：1、摒弃“请求和保持”条件：在采用这种方法时，系统规定所有进程在开始运行之前，都必须一次性地申请其在整个运行过程所需的全部资源。此时，若系统有足够的资源分配给某进程，便可把其需要的所有资源分配给该进程，这样，该进程在整个运行期间便不会再提出资源要求，从而摒弃了请求条件。2、摒弃“不剥夺”条件：在采用这种方法时系统规定，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，等以后需要时再重新申请。这意味着某一进程已经占有的资源，在运行进程中会被暂时地释放掉，也可认为是被剥夺了，从而摒弃了“不剥夺”条件。3、摒弃“环路等待”条件：在这种方法中规定，系统将所有资源按类型进行线性排队，并赋予不同的序号。所有进程对资源的请求必须严格按照资源序号递增的次序提出，这样，在所形成的资源分配图中，不可能再出现环路，因而摒弃了“环路等待”条件。

21、在银行家算法的例子中，如果 P0 发出的请求向量由 Request(0,2,0)改为 Request(0,1,0)，问系统可否将资源分配给它？

22. 在银行家算法中，若出现下述资源分配情：

Process	Allocation	Need	Available
P ₀	0032	0012	1622
P ₁	1000	1750	
P ₂	1354	2356	
P ₃	0332	0652	
P ₄	0014	0656	

试问：

(1) 该状态是否安全？

(2) 若进程 P_2 提出请求 $\text{Request}(1, 2, 2, 2)$ 后，系统能否将资源分配给它？

(1) 该状态是安全的，因为存在一个安全序列 $\langle P_0P_3P_4P_1P_2 \rangle$ 。下表为该时刻的安全序列表。

资源情况 进程	Work	Need	Allocation	Work+Allocation	Finish
P_0	1 6 2 2	0 0 1 2	0 0 3 2	1 6 5 4	true
P_3	1 6 5 4	0 6 5 2	0 3 3 3	1 9 8 7	true
P_4	1 9 8 7	0 6 5 6	0 0 1 4	1 9 9 11	true
P_1	1 9 9	1 7 5 0	1 0 0 0	2 9 9 11	true
P_2	11	2 3 5 6	1 3 5 4	3 12 14 17	true
	2 9 9				
	11				

(2) 若进程 P_2 提出请求 $\text{Request}(1, 2, 2, 2)$ 后，系统不能将资源分配给它，若分配给进程 P_2 ，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. 为什么要配置层次式存储器？

这是因为：

- 设置多个存储器可以使存储器两端的硬件能并行工作。
- 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。
- 在微处理机内部设置各种缓冲存储器，以减轻对存储器存取的压力。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？ P119

答：1、绝对装入方式：绝对装入方式只能将目标模块装入到内存中事先指定的位置。在多道程序环境下，编译程序不可能预知所编译的目标模块应放在内存的何处，因此，绝对装入方式只适用于单道程序环境。

2、可重定位装入方式：在多道程序环境下，所得到的目标模块的起始地址通常是从 0 开始的，程序中的其它地址也都是相对于起始地址计算的。此时应采用可重定位装入方式，根据内存的当前情况，将装入模块装入到内存的适当位置。

3、动态运行时装入方式：可重定位装入方式可将装入模块装入到内存中任何允许的位置，故可用于多道程序环境；但这种方式并不允许程序运行时在内存中移动位置。

3. 何谓静态链接？何谓装入时动态链接和运行时的动态链接？ P120

答：1、静态链接：在程序运行之前，先将各目标模块及它们所需的库函数，链接成一个完整的装配模块，以后不再拆开，我们把这种事先进行链接的方式称为静态链接方式。2、装入时动态链接：这是指将用户源程序编译后所得到的一组目标模块，在装入内存时，采用边装入边链接的链接方式。3、运行时动态链接：这是指对某些目标模块的链接，是在程序执行中需要该（目标）模块时，才对它进行的链接。

4. 在进行程序链接时，应完成哪些工作？

答：表态链接、装入时动态链接、运行时动态链接；

5. 在动态分区分配方式中，应如何将各空闲分区链接成空闲分区链？ P123 要画图

答：为了实现对空闲分区的分配和链接，在每个分区的起始部分，设置一些用于控制分区分配的信息，以及用于链接各分区所用的前向指针；在分区尾部则设置一后向指针，通过前、后向链接指针，可将所有的空闲分区链接成一个双向链，为了检索方便，在分区尾部重复设置状态位和分区大小表目。当分区被分配出去以后，把状态位由“0”改为“1”，此时，前、后向指针已无意义。

6、为什么要引入动态重定位？如何实现？ P127 P128

答：a. 为了在程序执行过程中，每当访问指令或数据时，将要访问的程序或数据的逻辑地址转换成物理地址，引入了动态重定位。

b. 可在系统中增加一个重定位寄存器，用它来装入(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,从而实现动态重定位。

9. 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。

a. 首次适应算法的优缺点：保留了高址部分的大空闲区，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。

b. 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为均匀，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。

c. 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。

d. 最坏适应算法的优缺点：给文件分配分区后剩下的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。

10. 在系统中引入对换后可带来哪些好处？

能将内存中暂时不运行的进程或暂时不用的程序和数据，换到外存上，以腾出足够的内存空间，把已具备运行条件的进程或进程所需的程序和数据换入内存，从而大大地提高了内存的利用率。

12、在以进程为单位进行对换时，每次是否都将整个进程换出？为什么？ P129

答：a. 以进程为单位进行对换时，每次都应将整个进程换出；

b. 目的为了解决内存紧张的问题，提高内存的利用率。

13、为实现分页存储管理，需要哪些硬件支持？ P144

答：需要一台具有一定容量的内存及外存的计算机系统外，页表机制、缺页中断机构以及地址变换机构。

14、较详细地说明引入分段存储管理是为了满足用户哪几方面的需要。 P135

答：方便编程、信息共享、信息保护、动态增长、动态链接。

16. 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？

a. 对于分页系统，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. 而对于分段系统，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，将其中的基址与内存地址一一对应起来即可。

17、分页和分段存储管理有何区别？ P130 P135

答：主要表现在（1）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。或者说，分页仅仅是由于系统管理的需要而不是用户的需要。段则是信息的逻辑单位，它含有一组其意义相对完整的信息。分段的目的是为了能更好地满足用户的需要。（2）页的大小固定且由系统决定，由系统把逻辑地址划分为页号和页内地址两部分，是由机器硬件实现的，因而在系统中只能有一种大小的页面；根据信息的性质来划分。（3）分页的作业地址空间是一维的，即单一的线性地址空间，程序员只需利用一个记忆符，即可表示一个地址；而分段的作业地址空间则是二维的，程序员在标识一个地址时，即需给出段名，又需给出段内地址。

18. 试全面比较连续分配和离散分配方式。

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，动态地为之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，加大查找开销；循环首次适应算法，该算法能使内存中的空闲分区分布均匀，但会导致缺少大的空闲分区；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. 离散分配方式基于将一个进程直接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。

19、虚拟存储器有哪些特征？其中最本质的特征是什么？ P144

答：多次性、对换性、虚拟性；值得说明的是，虚拟性是以多次性和对换性为基础的，或者说，仅当系统允许将作业分多次调入内存，并能将内存中暂时不运行的程序和数据换至盘上时，才有可能实现虚拟存储器；而多次性和对换性又必须建立在离散分配的基础上。

20. 实现虚拟存储器需要哪些硬件支持？

- a. 对于为实现请求分页存储管理方式的系统，除了需要一台具有一定容量的内存及外存的计算机外，还需要有页表机制，缺页中断机构以及地址变换机构；
- b. 对于为实现请求分段存储管理方式的系统，除了需要一台具有一定容量的内存及外存的计算机外，还需要有段表机制，缺段中断机构以及地址变换机构；

21、实现虚拟存储器需要几个关键技术？ P143

答：1、分页请求系统：这是在分页的基础上，增加了请求调页功能和页面置换功能所形成的页式虚拟存储系统。它允许只装入少数页面的程序（及数据），便启动运行。2、请求分段系统：这是在分段系统的基础上，增加了请求调段及分段置换功能后所形成的段式虚拟存储系统。它允许只装入少数段（而非所有的段）的用户程序和数据，即可启动运行。

25. 在请求分页系统中，通常采用哪种页面分配方式——物理块分配策略？

三种分配方式：固定分配局部置换、可变分配全局置换、可变分配局部置换。

26. 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，试计算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。

	4	3	2	1	4	3	5	4	3	2	1	5
M=3	4	4	4	1	1	1	5			5	5	
		3	3	3	4	4	4			2	2	
			2	2	2	3	3			3	1	
M=4	4	4	4	4			5	5	5	5	1	1
		3	3	3			3	4	4	4	4	5
			2	2			2	2	3	3	3	3
				1			1	1	1	2	2	2

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，增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，这种现象被称为是 Belady 现象。

28. 试说明改进型 Clock 置换算法的基本原理。

基本原理：

在将一个页面换出时，如果该页已被修改过，便须将该页重新写回到磁盘上；但如果该页未被修改过，则不必将它写回磁盘上。在改进型算法中，除需考虑页面的使用情况外，还须再增加一个因素，即置换代价，这样，选择页面换出时，既要是未使用过的页面，又要是未被修改过的页面。

15 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，需要从内存中调出一页程序或数据送磁盘的对换区中，如果算法不适当，刚被换出的页很快被访问，需重新调入，因此需再选一页调出，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调进换出上，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. 产生抖动的原因是由于 CPU 的利用率和多道程序度的对立统一矛盾关系引起的，为了提高 CPU 利用率，可提高多道程序度，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，而系统的调度程序又会为了提高 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。

第五章

1、试说明设备控制器的组成。P163

答：设备控制器的组成由设置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；

2、为了实现 CPU 与设备控制器间的通信，设备控制器应具备哪些功能？P162-P163

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；差错控制；

3、什么是字节多路通道？什么是数组选择通道和数组多路通道？P164-P165

答：1、字节多路通道：这是一种按字节交叉方式工作的通道。它通常都含有许多非分配型子通道，其数量可从几十到数百个，每个子通道连接一台 I/O 设备，并控制该设备的 I/O 操作。这些子通道按时间片轮转方式共享主通道。只要字节多路通道扫描每个子通道的速率足够快，而连接到子通道上的设备的速率不是太高时，便不致丢失信息。2、数组选择通道：字节多路通道不适于连接高速设备，这推动了按数组方式进行数据传送的数组选择通道的形成。3、数组多路通道：数组选择通道虽有很高的传输速率，但它却每次只允许一个设备数据。数组多路通道是将数组选择通道传输速率高和字节多路通道能使各子通道（设备）分时并行操作的优点相结合而形成的一种新通道。它含有多个非分配型子通道，因而这种通道既具有很多高的数据传输速率，又能获得令人满意的通道利用率。

4、如何解决因通道不足而产生的瓶颈问题？P166

答：解决“瓶颈”问题的最有效的方法，便是增加设备到主机间的通路而不增加通道，就是把一个设备连接到多个控制器上，而一个控制器又连接到多个通道上。多通路方式不仅解决了“瓶颈”问题。而且提高了系统的可靠性，因为个别通道或控制器的故障不会使设备和存储器之间没有通路。

5、试对 VESA 及 PCI 两种总线进行比较。P167

答：1、VESA 该总线的设计思想是以低价位迅速占领市场。VESA 总线的带宽为 32 位，最高传输速率为 132Mb/s。VESA 总线仍存在较严重的缺点，它所能连接的设备数仅为 2—4 台，在控制器中无缓冲，故难于适应处理器速度的不断提高，也不能支持后来出现的 Pentium 微机。2、PC 随着 Pentium 系列芯片的推出，PCI 在 CPU 和外设间插入一复杂的管理层，用于协调数据传输和提供一致的接口。在管理层中配有数据缓冲，通过该缓冲可将线路的驱动能力放大，使 PCI 最多能支持 10 种外设，并使高时钟频率的 CPU 能很好地运行，最大传输速率可达 132Mb/s。PCI 即可连接 ISA、EISA 等传统型总线，又可支持 Pentium 的 64 位系统，是基于奔腾等新一代微处理器而发展的总线。

6、试说明推动 I/O 控制发展的主要因素是什么？P167

答：在 I/O 控制方式的整个发展过程中，始终贯穿着这样一条宗旨，即尽量减少主机对 I/O 控制的干预，把主机从繁杂的 I/O 控制事务中解脱出来，以便更多地去完成数据处理任务。

7、有哪几种 I/O 控制方式？各适用于何种场合？P167-P170

答：1、程序 I/O 方式；2、中断驱动 I/O 控制方式；3、直接存储器访问（DMA）I/O 控制方式；I/O 通道控制方式；????????

8、试说明 DMA 的工程流程。P170 图要画

答：当 CPU 要从磁盘读入一数据块时，便向磁盘控制器发送一条读命令。该命令被送到其中的命令寄存器（CR）中。同时，还须发送本次要读数据读入的内存起始目标地址，该地址被送入内存地址寄存器（MAR）中；本次要读数据的字（节）数则送入数据计数器（DC）中，还须将磁盘中的源地址直接送至 DMA 控制器的 I/O 控制逻辑上，然后启动 DMA 控制器进行数据传送，以后 CPU 便可去处理其它任务。此后，整个数据传送过程便由 DMA 控制器进行控制。当 DMA 控制器已从磁盘中读入一个字（节）的数据送入数据寄存器（DR）后，再挪用下一个存储器周期，将该字（节）传送到 MAR 所指示的内存单元中。接着便对 MAR 内容加 1，将 DC 内容减 1。若减 1 后 DC 内容不为 0，表示传送未完，便继续传送下一个字（节）；否则，由 DMA 控制器发出中断请求。

10. 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 $\max(C, T) + M$ 。

11. 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一缓冲区移出数据，并送入用户进程。接着由 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 $\max(C+M, T)$ 。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： $\max(C, T)$

15、为何要引入设备独立性？如何实现设备的独立性？P184-P185

答：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逻辑设备和物理设备这两个概念。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。因此，系统须具有将逻辑设备名称转换为某物理设备名称的功能，这非常类似于存储器管理中所介绍的逻辑地址和物理地址的概念。在应用程序中的使用的是逻辑地址，而系统在分配和使用内存时，必须使用物理地址。1、对独立设备的分配与回收；2、将逻辑设备名映射为物理设备名，进一步可以找到相应物理设备的驱动程序；3、对设备进行保护，禁止用户直接访问设备；4、缓冲管理，即对字符设备和块设备的缓冲区进行有效的管理，以提高 I/O 的效率；5、差错控制，由于在 I/O 操作中的绝大多数错误都与设备无关，故主要由设备驱动程序处理，而设备独立性软件只处理那些设备驱动程序无法处理的错误；6、提供独立于设备的逻辑块，不同类型的设备信息交换单位是不同的，读取和传输速率也各不相同，即使同一类型的设备，其信息交换单位大小也是有差异的，如不同磁盘由于扇区大小的不同，可能造成数据块大小的不一致，因此设备独立性软件应负责隐藏这些差异，对逻辑设备使用并向高层软件提供大小统一的逻辑数据块。

16、在考虑到设备的独立性时，应如何分配独享设备？P189

答：1、分配设备；分配控制器；分配通道；2、增加设备的独立性；考虑多通路情况；

17、何谓设备虚拟？实现设备虚拟时所依赖的关键技术是什么？P187—P188

答：设备虚拟是指设备本身虽是独占设备，但经过某种技术处理，可以把它改造成虚拟设备。实现设备虚拟：由于可虚拟设备是指一台物理设备在采用虚拟技术后，可变成多台逻辑上的所谓虚拟设备，因而说，一台可虚拟设备是可共享的设备，可以将它同时分配给多个进程使用，并对这些访问该（物理）设备的先后次序进行控制。

18、试说明 SPOOLing 系统的组成。P190

答：1、输入井和输出井：这是在磁盘上开辟的两个大存储空间。2、输入缓冲区和输出缓冲区：为了缓和 CPU 和磁盘之间速度不匹配的矛盾，在内存中要开辟两个缓冲区，输入缓冲区和输出缓冲区。3、输入进程 SPi 和输出进程 SP0：这里利用两个进程来模拟脱机 I/O 时的外围控制机。

19. 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？

在实现后台打印时，SPOOLing 系统应为请求 I/O 的进程提供以下服务：（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）输出进程再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，再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

据从输出并传送到内存缓冲区，再由打印机进行打印。

第六章

1、何谓数据项、记录 and 文件？ P203 P204

答：数据项：数据项是最低级的数据组织形式，是数据组中可以命名的最小逻辑数据单位，若干个基本数据项组成的。记录：记录是一组相关数据项的集合，用于描述一个对象在某方面的属性。文件：文件是指由创建者所定义的、具有文件名的一组相关元素的集合，可分为有结构文件和无结构文件两种。在有结构的文件中，文件由若干个相关记录组成；而无结构文件则被看成是一个字符流。文件在文件系统中是一个最大的数据单位，它描述了一个对象集。

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。 P206 图要画

答：1、对象及其属性：文件、目录、硬盘（磁带）存储空间；2、对对象操纵和管理的软件集合：文件管理系统的核心部分；3、文件系统的接口：命令接口、程序接口；

3、试说明用户可以对文件施加的主要操作有哪些。 P207

答：1、最基本的文件操作：创建文件、删除文件、读文件、写文件、截断文件、设置文件的读/写位置；2、文件的“打开”和“关闭”操作；3、其它文件操作；

4、何谓逻辑文件？何谓物理文件？ P208

答：逻辑文件：这是从用户观点出发所观察到的文件组织形式，是用户可以直接处理的数据及其结构，它独立于文件的物理特性，又称为文件组织。物理结构：又称为文件的存储结构，是指文件在外存上的存储组织形式。这不仅与存储介质的存储性能有关，而且与所采用的外存分配方式有关。

5、如何提高对变长记录顺序文件的检索速度？ P210

答：对于变长记录的顺序文件，在顺序读或写时的情况相似，但应分别为它们设置读或写指针，在每次读或写完一个记录后，须将读或写指针加上 L_i 。 L_i 是刚读或刚写完的记录的长度。

6、试说明对索引文件和索引顺序文件的检索方法。 P211 P212

答：在对索引文件进行检索时，首先是根据用户（程序）提供的关键字，并利用折半查找法去检索索引表，从中找到相应的事项；再利用该表项中给出的指向记录的指针值，去访问所需的记录。在对索引顺序文件进行检索时，首先也是利用用户（程序）所提供的关键字以及某种查找算法去检索索引表，找到该记录所在记录组中第一个记录的表项，从中得到该记录组第一个记录在主文件中的位置；然后，再利用顺序查找法去查找主文件，从中找到所要求的记录。

7、试从检索速度和存储费用两方面来比较两级索引文件和索引顺序文件。 P212

答：

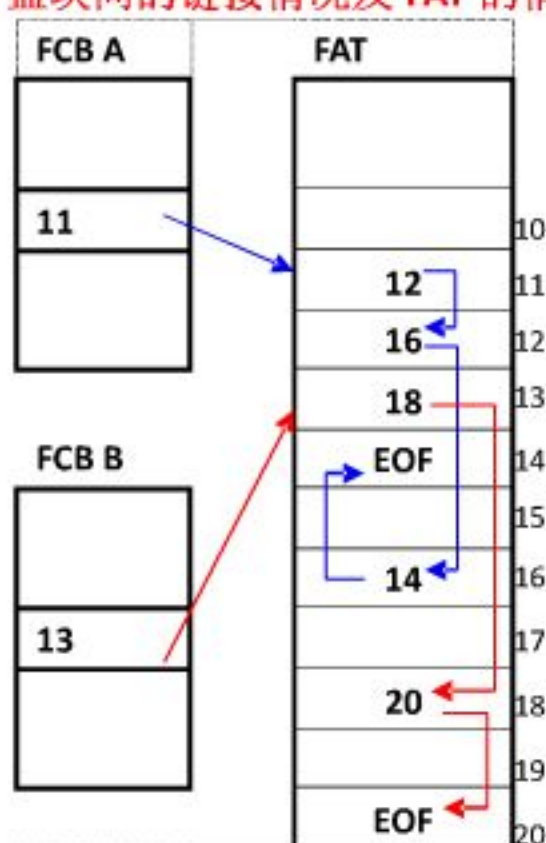
8、试说明顺序文件的结构及其优点。 P209 P210

答：第一种是结构：各记录之间的顺序与关键字无关。第二种情况是顺序结构：指文件中的所有记录按关键字（词）排列。可以按关键词的长短从小到大排序，也可以从大到小排序；或按其英文字母顺序排序。顺序文件的最佳应用场合是对诸记录进行指存取时，即每次要读或写一大批记录时。此时，对顺序文件的存取效率是所有逻辑文件中最高的；此外，也只有顺序文件才能存储在磁带上，并能有效地工作。

9、在链接式文件中常用哪种链接方式？为什么？ p215

答：采取离散分配方式：链接方式又可分为隐式链接和显式链接两种形式。在隐式链接分配方式时，在文件目录的每个目录项中，都须含有指向链接文件第一个盘块和最后一个盘块的指针。显式链接这是指把用于链接文件各物理块的指针，显式地存放在内存的一张链接表中。

10. 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。



12. 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指引一个 512MB 的磁盘？

解：512MB/512B=1M 个盘块，而每个盘块都应有一个指针来指示，所以应该有 1M 个指针，因此若有 64K 个指针则不能指引一个 512MB 的磁盘。

13. 为了快速访问，又易于更新，当数据为以下形式时，应选用何种文件组织方式。

- (1) 不经常更新，经常随机访问；
- (2) 经常更新，经常按一定顺序访问；
- (3) 经常更新，经常随机访问；
- (1) 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

14. 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：

$$9999/1024=9 \text{ 余 } 783$$

$$18000/1024=17 \text{ 余 } 592$$

$$420000/1024=410 \text{ 余 } 160$$

15. 什么是索引文件？为什么要引入多级索引？

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：(1)能有效提高对目录的检索速度；(2)允许文件重名；(3)便于实现文件共享。

18. 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，只能实现目录管理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置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。

19. 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；假定文件系统中存在 N 个文件，在单级目录中，最多要检索 N 个目录项，但对于有 i 级的树型目录，在目录中每检索一个指定文件，最多可能要检索 $i \cdot \sqrt{N}$ 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用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，填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。

20. 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，再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。

23. 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 5 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 5 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： $b=(i-1) \times 16 + j = 35$ ；
- (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。

24. 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。

$[500/32]z=16$ 个字

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