计算机科学与技术专业

《计算机操作系统》复习提纲

第一章 操作系统引论

*1 操作系统的定义、作用

定义：

操作系统是计算机系统中的一个系统软件，它是这样一些程序模块的集合：能有效地组织和管理计算机系统的硬件和软件资源，合理地组织计算机工作流程，控制程序的执行，并向用户提供各种服务功能，使得用户能够灵活、方便、有效地使用计算机，使整个计算机系统能高效地运行。

操作系统的作用：

(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（处理器、存储器、I/O设备、数据程序）

(3) OS实现了对计算机资源的抽象（在硬件上覆盖I/O设备、文件和窗口管理软件，即虚拟机）

2 操作系统的分类、特点、适用场合

1、批处理操作系统：

单道批处理系统：自动性。 顺序性。 单道性

多道批处理系统：多道性。 无序性。 调度性

优缺点：资源利用率高。 系统吞吐量大。 平均周转时间长。 无交互能力。

2、分时系统（典型系统： BSD, SRV4, OSF1, SCO UNIX, AIX, Solaris, Linux)：

多路性。 独立性。 及时性。 交互性。

用户的需求：(1) 人—机交互。

(2) 共享主机。

(3) 便于用户上机。

3、实时系统（典型系统：VxWorks, pSoS, Nucleus)：

多路性。 独立性。 及时性。 交互性。 可靠性。

应用需求：(1) 实时控制。

(2) 实时信息处理。(如订票系统，银行管理系统)

4、网络操作系统（典型系统：Novell Netware)：

它是基于计算机网络的,

是在各种计算机操作系统上,

按网络体系结构协议标准开发的软件,

包括网络管理,通信,安全,资源共享

和各种网络应用。

目标：是相互通信及资源共享

5、分布式操作系统：

由多个分散的处理单元经互连网络的连接而形成的，且可以实现分布处理的系统。其中，每个处理单元既具有高度的自治性，又相互协调，能在系统范围内实现资源管理、动态地分配任务，并能并行地运行分布式程序。

操作系统	分布性	并行性	透明性	共享性	健壮性
网络操作系统	分布处理，集中控制	任务在本地计算机上运行	操作透明，但需指明物理位置	一般只能共享服务器上的资源	控制集中于服务器，可靠性弱
分布式操作系统	分布处理，分布控制	多任务在多处理单元中并行执行	操作透明，而且物理位置透明	各站点资源可供全系统共享	容错能力强，可靠性高

3　操作系统的特征

操作系统的基本特征：

(1)并发性（两个或多个事件在同一时间间隔内发生；进入进程和线程），并行、并发？

(2)共享性（系统中资源可供内存中多个并发执行的进程（线程）共同使用，方式为互斥共享方式和同时访问方式）

(3)虚拟性（通过某种技术把一个物理实体变为若干个逻辑上的对应物。方式：时分复用技术和空分复用技术）

(4)异步性（进程以不可预知的速度向前推进，多道程序设计固有的特点）

***4　操作系统的功能**

OS的主要功能：

(1) 处理机管理（进程管理）功能；（主要包括创建和撤销进程、协调诸进程的运行、实现进程间信息交换、把处理机分配给进程。进程同步机制功能是协调多个进程的运行，分为竞争和协作两种方式，实现进程同步常用的及时是信号量机制。调度包括作业调度和进程调度两步。）

(2) 存储器管理功能；（内存分配、内存保护、地址映射和内存扩充等功能。内存分配有动态和静态两方式。内容扩充的功能是请求调入和置换）

(3) 设备管理功能（缓冲管理、设备分配、设备处理和虚拟设备。缓冲管理包括单、双、公用缓冲机制。设备处理程序（设备驱动程序）的基本任务是实现CPU和设备控制器之间的通信）

(4) 文件管理功能；（文件存储空间管理、目录管理、文件读写管理、共享保护功能）

(5) 操作系统与用户之间的接口；（用户接口和程序接口）

第二章 操作系统用户界面

1　作业的概念，作业调度算法

作业(Job)：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

1. 先来先服务和短作业(进程)优先调度算法：
   先来先服务(FCFS)调度算法
   短作业(进程)优先调度算法 SJ(P)F
2. 高优先权优先调度算法：非抢占式优先权算法 和 抢占式优先权调度算法
3. 基于时间片的轮转调度算法

时间片轮转法（RR）
多级反馈队列调度算法

2　用户与操作系统的接口：联机命令接口与系统调用

联机命令接口：也称为联机用户接口，终端用户利用该接口可以调用操作系统的功能，取得操作系统的服务。用户可以使用联机控制命令来对自己的作业进行控制。联机用户接口可以实现用户与计算机间的交互。

系统调用提供了用户程序和操作系统之间的接口，应用程序通过系统调用实现其与 OS 的通信，并可取得它的服务。系统调用可供所有的应用程序使用，也可供 OS 自身的其它部分，尤其是命令处理程序使用。根据其功能分类：进程控制(类)、文件管理(类)、设备管理(类)及进程通信等类的系统调用。

第三章 进程的管理

***1　进程的定义、进程的特征、进程的基本状态及其转换过程和转换条件，进程的挂起状态**

进程定义：(1) 进程是程序的一次执行。
(2) 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。
(3) 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。

进程的特征：
1)结构特征；由程序段、相关的数据项和PCB三部分构成了进程实体。
2)动态性；指从创建、调度执行到撤销的过程是动态的。
3)并发性；
4)独立性；因为有PCB，可以独立运行、独立分配资源、独立接受调度等功能
5)异步性；各进程按各自独立、不可预知的速度向前推进。

进程的三种基本状态：1）就绪(Ready)状态　：处CPU外，已占有其他必要的资源的进程
2) 执行状态
3) 阻塞状态：因事故是正在执行的进程停止，并让出CPU。

进程转换过程和转换条件：

(1) 活动就绪→静止就绪。当进程处于未被挂起的就绪状态时，称为活动就绪状态，表示为 Readya。当用挂起原语 Suspend 将该进程挂起后，该进程便转变为静止就绪状态，表示为 Readys，处于 Readys 状态的进程不再被调度执行。

(2) 活动阻塞→静止阻塞。当进程处于未被挂起的阻塞状态时，称处于活动阻塞状态，表示为 Blockeda。当用 Suspend 原语将它挂起后，进程便转变为静止阻塞状态，表示为Blockeds。处于该状态的进程在其所期待的事件出现后，将从静止阻塞变为静止就绪。

(3) 静止就绪→活动就绪。处于 Readys 状态的进程，若用激活原语 Active 激活后，该进程将转变为 Readya 状态。

(4) 静止阻塞→活动阻塞。处于 Blockeds 状态的进程，若用激活原语 Active 激活后，该进程将转变为 Blockeda 状态。

进程的挂起状态（引入挂起状态的原因有:）:
(1) **终端用户的请求**。当终端用户在自己的程序运行期间发现有可疑问题时，希望暂时使自己的程序静

止下来。
(2) **父进程请求**。有时父进程希望挂起自己的某个子进程，以便考查和修改该子进程，或者协调各子进程间的活动。
(3) **负荷调节的需要**。当实时系统中的工作负荷较重，已可能影响到对实时任务的控制时，可由系统把一些不重要的进程挂起，以保证系统能正常运行。
(4) **操作系统的需要**。操作系统有时希望挂起某些进程，以便检查运行中的资源使用情况或进行记账

2 进程控制块PCB

进程控制块的作用是使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。**OS是根据PCB来对并发执行的进程进行控制和管理的**。

进程控制块中的信息：

1) 进程标识符 2) 处理机状态 3) 进程调度信息 4) 进程控制信息

3 进程控制：进程的创建、终止、阻塞与唤醒、挂起与激活

进程的创建事件：用户登录。 作业调度。 提供服务。 应用请求

流程：(1)申请空白PCB。
(2) 为新进程分配资源。
(3) 初始化进程控制块。
(4) 将新进程插入就绪队列，如果进程就绪队列能够接纳新进程， 便将新进程插入就绪队列。

进程的终止事件：1) 正常结束

2) 异常结束：① 越界错误。② 保护错。③ 非法指令。④ 特权指令错。⑤ 运行超时。⑥ 等待超时。⑦ 算术运算错。⑧ I/O故障。

3) 外界干预： ① 操作员或操作系统干预。 ② 父进程请求。 ③ 父进程终止。

流程：(1) 根据被终止进程的标识符，从PCB集合中检索出该进程的PCB，从中读出该进程的状态。
(2) 若被终止进程正处于执行状态，应立即终止该进程的执行，并置调度标志为真，用于指示该进程被终止后应重新进行调度。
(3) 若该进程还有子孙进程，还应将其所有子孙进程予以终止，以防他们成为不可控的进程。
(4) 将被终止进程所拥有的全部资源，或者归还给其父进程， 或者归还给系统。
(5) 将被终止进程(它的PCB)从所在队列(或链表)中移出， 等待其他程序来搜集信息。

引起进程阻塞和唤醒的事件：请求系统服务 2) 启动某种操作 3) 新数据尚未到达 4) 无新工作可做

阻塞过程：1）找到将要被阻塞进程的标识号对应的PCB
2）若该进程为运行态，则保护现场，将其状态转为阻塞态，停止运行。
3）把该PCB插入相应事件的等待队列，将处理机的资源调度给其他就绪进程

唤醒过程：1）在该事件的等待队列中找到相应进程的PCB
2）将其从等待队列中移出，并置其状态为就绪态
3）把该PCB插入就绪队列，等待调度程序调度

挂起原语的执行过程是：首先检查被挂起进程的状态，若处于活动就绪状态，便将其改为静止就绪；对于活动阻塞状态的进程，则将之改为静止阻塞。为了方便用户或父进程考查该进程的运行情况而把该进程的 PCB 复制到某指定的内存区域。最后，若被挂起的进程正在执行，则转向调度程序重新调度。

激活原语先将进程从外存调入内存，检查该进程的现行状态，若是静止就绪，便将之改为活动就绪；若为静止阻塞，便将之改为活动阻塞。假如采用的是抢占调度策略，则每当有新进程进入就绪队列时，应检查是否要进行重新调度，即由调度程序将被激活进程与当前进程进行优先级的比较，如果被激活进程的优先级更低，就不必重新调度；否则，立即剥夺当前进程的运行，把处理机分配给刚被激活的进程。

***4　线程的基本概念、线程与进程的区别**

进程与线程的基本概念

1)进程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。

2)线程是为了减少程序在并发执行时所付出的空间开销，是OS具有更好的并发性。

进程和线程的区别

1)调度：线程作为调度和分派的基本单位；进程作为资源拥有的基本单位。

2)并发性：进程之间可以并发执行，进程中的诸线程之间也可并发执行。

3)拥有资源：进程拥有资源，线程无资源，但可以访问所属进程的资源

4)系统开销：创建可撤销进程的代价比创建和撤销线程的代价大的多。

5　进程同步的基本概念、临界区与临界资源、临界区的进入与退出、进程互斥

进程同步的基本概念：两种形式的制约关系：1）间接相互制约关系。（2）直接相互制约关系

临界区：把在每个进程中访问临界资源的那段代码称为临界区

临界资源：多道程序系统中存在许多进程，它们共享各种资源，然而有很多资源一次只能供一个进程使用。一次仅允许一个进程使用的资源称为临界资源。许多物理设备都属于临界资源，如输入机、打印机、磁带机等。

进入区：为了进入临界区使用临界资源，在进入区要检查可否进入临界区，若能进入临界区，则应设置正在访问临界区的标志，以阻止其他进程进入临界区

退出区：将正在访问临界区的标志清除

进程互斥：两个或两个以上的进程，不能同时进入关于同一组共享变量的临界区域，否则可能发生与时间有关的错误，这种现象被称作进程互斥• 也就是说，一个进程正在访问临界资源，另一个要访问该资源的进程必须等待。

***6　通信量机制实现互斥，整型信号量与记录型号量**

信号量机制是一种卓有成效的进程同步工具。包括整形信号量、记录型信号量、AND型信号量、信号量集。

整型信号量：由Dijkstra把整型信号量定义为一个整型量，除初始化外，仅能通过两个标准的原子操作(Atomic Operation) wait(S)和signal(S)来访问。

记录型号量：记录型信号量机制，则是一种不存在“忙等”现象的进程同步机制。

***7　经典进程同步问题：生产者-消费者问题**

利用记录型信号量解决生产者—消费者问题：

假定在生产者和消费者之间的公用缓冲池中，具有n个缓冲区，这时可利用互斥信号量mutex实现诸进程对缓冲池的互斥使用；利用信号量empty和full分别表示缓冲池中空缓冲区和满缓冲区的数量。又假定这些生产者和消费者相互等效，只要缓冲池未满，生产者便可将消息送入缓冲池；只要缓冲池未空，消费者便可从缓冲池中取走一个消息。对生产者—消费者问题可描述如下：

```
Var mutex, empty, full:semaphore  :=  1,n,0;
    buffer:array [0, …, n-1]  of item;
    in, out: integer  :=  0, 0;
    begin
     parbegin
      proceducer:begin
            repeat
               …
```

```
            producer an item nextp;
               …
            wait(empty);
            wait(mutex);
            buffer(in)  :=  nextp;
            in  :=  (in+1) mod n;
            signal(mutex);
            signal(full);
            until false;
          end
consumer:begin
           repeat
            wait(full);
            wait(mutex);
            nextc  :   =  buffer(out);
            out  :   =  (out+1) mod n;
            signal(mutex);
            signal(empty);
            consumer the item in nextc;
           until false;
          end
    parend
  end
```

***8 进程通信的类型，管道、共享存储区、消息通信原理。**

进程通信的类型：共享存储器系统(Shared-Memory System)：（1）基于共享数据结构的通信方式。(2) 基于共享存储区的通信方式。

消息传递系统(Message passing system)

管道(Pipe)通信

管道：是指用于连接一个读进程和一个写进程以实现他们之间通信的一个共享文件，又名pipe文件。管道机制提供以下三方面的协调能力：(1) 互斥(2) 同步(3) 确定对方是否存在，只有确定了对方已存在时，才能进行通信

共享存储区：在通信的进程之间存在一块可直接访问的共享空间，通过对这片共享空间进行写读操作实现进程间的信息交换

在进程之间通信时，源进程可以直接或间接地将消息传送给目标进程，由此可将进程通信分为直接通信和间接通信两种通信方式。

直接通信方式是指发送进程利用 OS 所提供的发送命令，直接把消息发送给目标进程。

间接通信方式指进程之间的通信需要通过作为共享数据结构的实体。

9 进程调度类型（与作业调度的区别），进程调度算法：先来先服务算法FCFS、短作业优先算法SJ（B）F、时间片轮转算法、优先权调度算法、多级反馈队列

区别：作业就是从外存放到内存的一个过程，它可以包含一个或多进程，作业的调度属于高级调度，进程的调度属于低级调度，所以，进程是一个系统中最基本的也是必需要求的调度，而作业调度是为是了提高系统性能而调度的高级调度 。一个作业可以分为很多进程，进程只是作业中的一个元素

***10　死锁的基本概念、死锁产生的原因、产生死锁的必要条件；死锁的预防和避免方法**

死锁的概念：多个进程在运行过程中因争夺资源而造成的一种僵局。

死锁产生的原因：（1）竞争资源。 (2) 进程间推进顺序非法。

产生死锁的必要条件：

1)互斥条件：临界资源的互斥访问

2)请求和保持条件：占着自己的资源不放，又去请求别人的

3)不剥夺条件：进程没有完成则不是放占有的资源

4)环路等待条件：发生死锁指必然存在一个资源环形链。

预防死锁：（1）摒弃“请求和保持”条件 2. 摒弃“不剥夺”条件 3. 摒弃“环路等待”条件

避免死锁：是在资源的动态分配过程中，用某种方法去防止系统进入不安全状态，从而避免发生死锁。

***11　Unix/Linux中的进程管理：进程创建与终止、管道通信、共享存储器、消息。(见实验3)**

***12　WIN32中的多线程与线程间的通信：事件、信号量、内存文映射、消息及其程序设计。（见课程笔记、课外习题、课外实验）**

第四章　存储器管理

***1　存储管理的功能**

操作系统的存储管理，负责对可执行存储器的分配、回收以及提供在存储层次间数据移动的管理机制

***2　连续分配存储管理：单一连续分配、固定分区分配、动态分区分配、动态重定位分区分配。动态分区分配算法、分配与回收过程。**

连续分配方式：一个用户程序分配一个连续的内存空间

1)　单一连续分配：一个程序装入其他程序就不允许被装入。只是用于单用户单任务的OS中。

2)　固定分区分配：把内存分为若干个固定大小的区域，每个分区装入一个作业，允许并发执行。

3)　动态分区分配：根据实际需要，动态地为之分配内存空间。

4)　动态重定位分区分配：通过重定位寄存器把相对地址转化成物理地址，此转化过程是在程序执行期间，随着每条指令或数据的访问自动进行的，故称为动态重定位。

分区分配算法

1)　首次适应算法（以地址递增次序访问）

2)　循环首次适应算法（从上一次分配处开始查找）

3)　最佳适应算法（小内存到大内存依次查找）

4)　最坏适应算法（每次分配从大内存开始割让）

5)　快速适应算法（对空闲分区进行分类，并建立索引表，选最适合的控件分配给请求的进程）

在动态分区存储管理方式中，主要的操作是分配内存和回收内存。

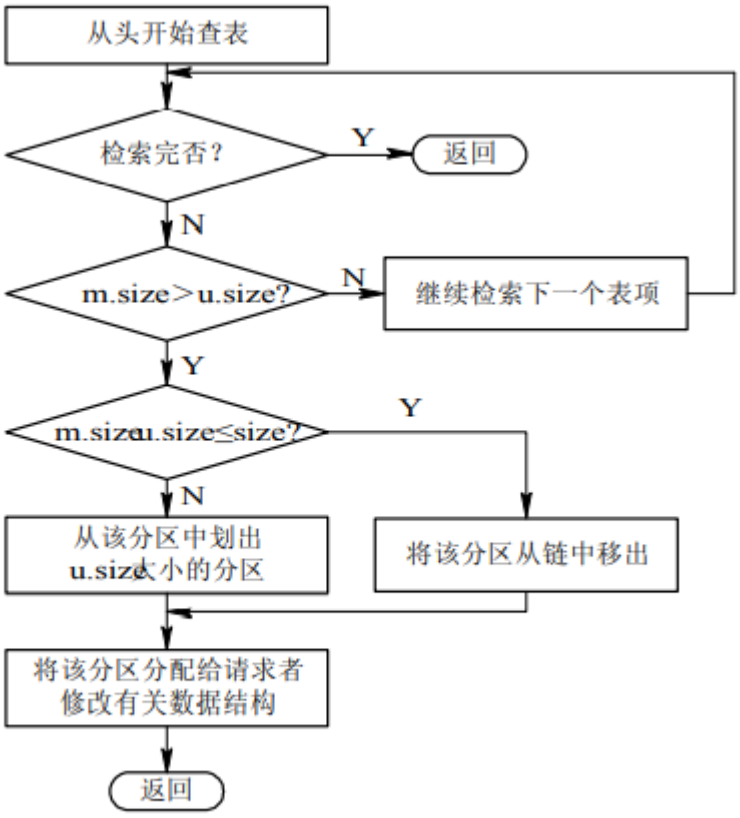

图 4-7　内存分配流程

可能出现以下四种情况之一：

(1) 回收区与插入点的前一个空闲分区 F1相邻接，见图 4-8(a)。此时应将回收区与插入点的前一分区合并，不必为回收分区分配新表项，而只需修改其前一分区 F1的大小。

(2) 回收分区与插入点的后一空闲分区 F2相邻接，见图 4-8(b)。此时也可将两分区合并，形成新的空闲分区，但用回收区的首址作为新空闲区的首址，大小为两者之和。

(3) 回收区同时与插入点的前、后两个分区邻接，见图 4-8(c)。此时将三个分区合并，使用 F1的表项和 F1的首址，取消 F2的表项，大小为三者之和。

(4) 回收区既不与 F1邻接，又不与 F2邻接。这时应为回收区单独建立一新表项，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。

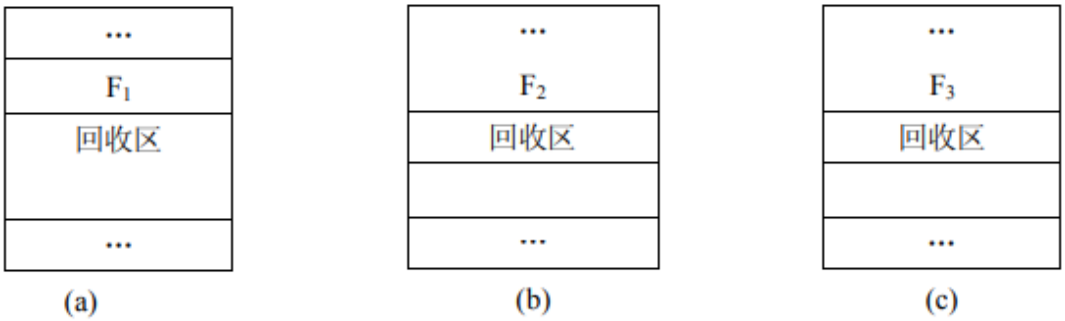

图 4-8　内存回收时的情况

3　了解对换作用与工作过程

作用：对换是提高内存利用率的有效措施。

对换过程：把暂时不运行的进程或程序和数据调到外存，再把已具备运行条件的进程或进程所需要的程

序和数据调到内存。

***4　分页存储管理方式基本**工作原理、页表及地址变换机构，**缺页中断与一般中断的区别。**

地址变换机制：将用户地址空间中的逻辑地址变换为内存空间中的物理地址
区别主要表现在下面两个方面：(1) 在指令执行期间产生和处理中断信号。(2) 一条指令在执行期间，可能产生多次缺页中断。

5　分段存储管理方式基本工作原理、段表及地址变换机构

***6　虚拟存储器的基本概念、局部性原理**
虚拟存储技术是一种从逻辑上扩充内存容量的方法，

***7　请求分页存储管理方式基本工作原理、调页策略**

***8　页面置换算法（重点OPT、FIFO、LRU、LFU），产生抖动的原因。**

第五章　设备管理

***1　设备管理的功能**
。设备管
主要功能有: 缓冲区管理、设备分配、设备处理、虚拟设备及实现设备独立性等

2　I/O控制方式：程序I/O、中断驱动I/O、DMA、I/O通道，什么是通道
I/O控制方式
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
4)　　I/O通道控制方式
I/O通道是一种特殊的处理机，它具有执行I/O指令的能力，可以控制I/O操作。类型分为：字节多路通道、数组选择通道、数组多路通道。

通道是用于控制外围设备(包括字符设备和块设备)的，根据信息交换方式的不同，可把通道分成以下三种类型: 1) 字节多路通道2) 数组选择通道3) 数组多路通道

***4　缓冲管理：缓冲区的概念、单缓冲、双缓冲、循环缓冲、缓冲池结构和工作原理**

5　设备分配：数据结构DCT、COCT、CHCT、SDT；独占设备、共享设备、虚拟设备、逻辑设备、设备的独立性概念，设备分配算法，独占设备的分配程序

***6　什么是SPOOLing，SPOOLing系统组成、基本工作原理和特点，共享打印机实现原理**

SPOOLing技术：通过SPOOLing技术便可将一台物理I/O设备虚拟为多台逻辑I/O设备，同样允许多个用户共享一台物理I/O设备。

Spooling系统的组成：输入井和输入井；输入缓冲区和输出缓冲区；输入进程和输出进程。

SPOOLing系统的特点

1)　提高了I/O的速度
2)　将独占设备改造为共享设备
3)　实现了虚拟设备功能

***7　什么是设备处理程序（设备驱动程序）、功能、特点（见实验）**

8　Unix/Linux中的设备及设备文件、逻辑设备（见实验）

第六章　文件管理

***1　文件管理的功能**

文件管理功能，即构成一个文件系统，负责管理在外存上的文件，并把对文件的存取、共享和保护等手段提供给用户。

2　文件类型，文件逻辑结构：顺序文件、索引文件、索引顺序文件的结构和特点（优缺点）

文件逻辑结构的类型

1)　有结构文件（由一个以上的记录构成的文件，又称记录式文件）
2)　无结构文件（由字符流构成的文件，又称流式文件）

顺序文件的优缺点

1)　适合进行批量存取
2)　存取效率是所有逻辑文件中最高的
3)　也只有顺序文件才能存储在磁带上，并能有效的工作
4)　不适合查找或修改单个记录
5)　增加或删除一个记录时比较困难

索引文件的缺点：除了有主文件外，还须配置一张索引表，而且每个记录都要有一个索引表，因此提高了存储费用。

3　目录管理：FCB和索引结点、目录查询技术
*4　文件共享方法，基于索引结点和符号链的文件共享
*5　文件保护方法，分级安全管理：系统级、用户级、目录级、文件级

第七章 磁盘存储管理

1　了解磁盘I/O：数据组织、磁盘性能指标、磁盘调度算法。掌握磁盘性能计算方法。
*2　外存分配方法：连续分配、链接分配、索引分配
*3　空闲存储空间的管理：空闲表法、空闲链表法、位示图法、成组链表法，分配与回收。
4　了解磁盘容错技术