1．操作系统概述

（1）操作系统的概念、特征、功能和提供的服务

操作系统的概念：控制和管理整个计算机系统的硬件和软件资源，合理地组织和调度计算机的资源分配，进而为用户和其他软件提供接口与环境的程序集合。

特征：基本特征包括并发、共享、虚拟和异步。并发和共享是操作系统两个最基本特征。

并发是指两个或多个事件在同一时间间隔内发生。引入进程的目的是使程序能并发执行。

资源共享即共享，是指系统中的资源可供并发执行的进程同时使用。

并发与共享的关系：1、资源共享以进程并发为条件。2、若系统不能对资源共享作出有效管理，则必将影响到程序的并发。

虚拟是指把一个物理上的实体变为若干逻辑上的对应物。利用多道程序设计把一个物理上的 CPU 虚拟为多个逻辑上的 CPU，称为虚拟处理器。还有虚拟存储器。

异步（不确定性）：多道程序环境允许多个进程并发执行，但由于资源有限，进程的执行并不是一贯到底，而是走走停停，它以不可预知的速度向前推进，这就是进程的异步性。

功能和目标：作为计算机系统资源管理者，操作系统有以下几个方面功能：处理机管理（可归结为进程的管理）、存储器管理（内存分配、地址映射、内存保护、内存扩充等）、文件管理、设备管理（缓冲管理、设备分配、设配处理、虚拟设备等）。

操作系统还可以作为用户与计算机硬件系统之间的接口。操作系统提供的接口有两类：命令接口、程序接口。

程序接口：由一组系统调用命令组成，用户通过在程序中使用这些系统调用命令来请求操作系统为其提供服务。

（2）操作系统的发展与分类

批处理阶段：

单道批处理系统：系统对作业的处理时成批进行的，但内存中始终保持一道作业。

特征：1、自动性，顺利情况下，磁带上的作业自动逐个运行，无需人工干预。

2、顺序性，各道作业顺序进入内存。

3、单道性，内存中仅有一道程序运行。

多道批处理系统：允许多个程序同时进入内存并允许他们在 CPU 中交替运行，这些程序共享系统中的各种软硬件资源。

特点：1、多道，内存中同时存放多道相互独立的程序。

2、宏观上并行，微观上串行。

优点：资源利用率高、各种资源得到充分利用；系统吞吐量大，CPU 保持忙碌。

缺点：用户响应时间较长；不提供人机交互能力，用户不了解程序的运行情况，又不能控制计算机。

分时操作系统：把处理器运行时间分成很短的时间片，按时间片轮流把处理器分配给各联机的作业使用。也是支持多道程序设计的系统，但分时系统是实现人机交互的系统。

特征：1、同时性，允许多个终端用户同时使用一台计算机。

2、交互性。

3、独立性，多个用户彼此独立进行操作。

4、及时性，用户请求能在极短的时间内获得响应。

实时操作系统：特点：及时性和可靠性

（3）操作系统的基本结构

2．进程管理

(1) 前驱图以及程序顺序执行和并发执行的特点

前驱图：一个有向无环图，用户描述及进程之间执行的前后关系。

程序的并发执行：特征：1、提高了系统吞吐量 2、间断性，程序并发执行时，共享系统资源，程序之间形成相互制约的关系。**3、失去封闭性**，并发执行时，多个程序共享资源，因而这些资源的状态将有多个程序改变，致使程序运行失去封闭性。**4、不可在现性，**由于失去封闭性，也将导致其失去再现性。

程序顺序执行：特征：1、顺序性，处理机的操作严格按照程序规定的顺序执行。**2、封闭**

性，资源的状态只有正在运行的程序能够改变，程序一旦开始运行，执行结果不受外界影响。

3、可再现性，只要程序执行时的环境和初始条件相同，当程序重复执行时，都将获得相同的结果。

(2) 进程的基本概念和思想

进程是进程实体运行的过程，是系统进行资源分配和调度的一个独立单位。

进程实体：程序段、相关数据段和 PCB 三部分构成进程实体。其中 PCB 是进程存在的唯一标志!

进程的基本特征：动态性，进程的实质是进程实体的一次执行过程，因此动态性是进程最基本的特征；**并发性**；**独立性**，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；**异步性**；**结构性**。

(3) 进程的状态与转换

进程通常有三种基本状态，运行态；就绪态；阻塞态。进程还有个**挂起状态**

就绪态获得处理机→运行态

运行态让出处理机→就绪态

进程请求某一资源的使用和分配或等待某一事件发生，运行态→阻塞态

进程等待的事件到来，阻塞态→就绪态

进程的切换是在内核的支持下实现的，是指处理机从一个进程的运行转到另一个进程上运行，**切换过程如下：**

1、保存处理机上下文，包括程序计数器和其他寄存器。

2、更新 PCB 信息。

3、把 PCB 移到相应队列，如就绪、阻塞等队列。

4、选择另一个进程执行并更新其 PCB。

5、更新数据管理的内存结构。

6、恢复处理机上下文

(4) 进程控制块及其作用

为了描述和控制进程的运行，系统为每个进程定义了一个数据结构，进程控制块，是进程实体的一部分，是一种重要的**记录型数据结构**。操作系统通过 PCB 来管理和控制进程。

PCB 中主要包括下述四个方面的信息：1、进程标识符，用于唯一的标识一个进程。2、处理机状态，保存 CPU 各寄存器信息。3、进程调度信息。4、进程控制信息。

(5) 进程组织

进程一般有三部分组成。**1 进程控制块 2 程序段 3 数据段**

程序段就是被进程调度程序调度到 CPU 的程序代码段。程序可被多个进程共享，即多个进程运行一个程序。

数据段，可以是进程对应的程序加工处理的原始数据，也可以是程序执行时产生的中间或最终结果。

进程控制块的组织方式：链接方式、索引方式

(6) 进程同步：进程同步的概念和同步原则，临界资源和临界区的概念，**信号量及其应用，经典进程同步问题**

概念：多道程序环境下，进程并发执行，不同进程之间存在不同的制约关系，为了协调相互制约的关系，引入进程同步的概念。

同步机制遵循以下原则：1、空闲让进，临界区空闲时，可以允许一个请求进入临界区的进程立即进入临界区。**2、忙则等待**，当已有进程进入临界区时，其他试图进入临界区的进程必须等待。**3、有限等待**，对请求访问的进程，应保证能在有限的时间内进入临界区。**4、让全等待**，当进程不能进入临界区时，应该释放处理器，防止进程忙等待。

临界资源：将一次仅允许一个进程使用的资源成为临界资源。

临界区：在每个进程中，访问临界资源的那段代码称为临界区。

(7) 进程通信的基本概念和方法

进程通信是指进程之间的信息交换。Pv 操作是低级通信方式，高级通信方式是指以较高效率传输大量数据的通信方式，**高级通信方式主要有三类**:

1）共享存储：在通信的进程之间存在一块可直接访问的共享空间，需要使用同步互斥工具（如 pv 操作）对共享空间进行读写控制，共享存储又分为两种：低级方式的共享基于数据结构的共享，高级方式的共享是是基于存储区的共享。

2）消息传递：进程之间的数据交换是以格式化的消息为单位。若通信的进程之间不存在可以直接访问的共享空间，则必须利用操作系统提供的消息传递方法进行通信（发消息和接消息）。分为直接通信方式和间接通信方式，**直接通信方式**：发送进程直接把消息发送给接受进程，并将其挂在接受进程的消息缓冲队列上，接受进程从消息缓冲队列获取信息。**间接通信方式：**发送进程把消息发给某个中间实体，接受进程从中间实体取得消息。这种中间实体一般称为**信箱**，这种通信方式又叫**信箱通信方式。**

3）管道通信：是消息传递的一种特殊方式，“管道”是指用于连接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，即 pipe 文件。管道机制必须提供一下三方面的协调能力：互斥、同步、和确定对方存在。

管道只能采用半双工同信，及某一时刻只能单向传输。要实现父子进程双方互动通信，需要定义两个管道。

(8) 线程的概念和多线程模型

线程的引入：引入进程的目的，是为了使多个程序并发执行，以提高资源利用率和系统吞吐量。引入线程的目的，是为了减少程序在并发执行时的时空开销，使 OS 具有更好的并发性。线程可看做轻量级的进程，是一个基本的 CPU 执行单元，也是程序执行流的最小单元。引入线程后，进程只作为除 CPU 外的系统资源分配单位，线程则作为处理机的分配单元。

线程与进程的比较：

1）调度：线程是独立调度的基本单位，进程是拥有资源的基本单位。同一进程中，线程切换不会引起进程切换，不同进程中，线程切换会引起进程切换。

2）拥有资源：进程是拥有资源的基本单位，线程不拥有有系统资源。但线程可以访问隶属进程的系统资源。

3）并发性：多个线程之间也可以并发执行。

4）系统开销：

5）地址空间和其他资源：进程的地址空间之间相互独立，，某进程内的线程对于其它进程不可见。

线程属性：

线程的实现有两类：用户型线程，内核级线程

多线程模型：实现用户级线程和内核级线程的链接

1）多对一模型：多个用户级线程映射到一个内核级线程。用户级线程对内核级线程不可见。

优点：线程管理在用户空间进行，效率较高

缺点：一个线程在内核服务时被阻塞，整个进程都将被阻塞；多个线程不能并行的运行在多个处理机上。

2）**一对一模型：**每个用户级线程映射到一个内核级线程

优点：一个进程阻塞后，允许另一个继续执行，并发能力较强。

缺点：创建线程开销较大，会影响到程序的性能。

3）**多对多模型：**将 n 个用户级线程映射到 m 个内核级线程上。m<=n

特点：克服了前两者的不足，并继承其优点

3．调度与死锁

（1）调度的概念

处理机调度：从就绪队列按照一定的算法，选择一个进程，并将处理机分配给他运行。

（2）调度队列模型

一个作业从提交到开始，要经历三级调度：

1）作业调度（高级调度），根据作业控制块中的信息，审查系统能否满足用户作业的资源需求，并按照一定的算法，将外存后备队中选取某些队列调入内存，并为他们创建进程，分配必要资源，属于内存和辅存之间的调度。频率较低。

2）低级调度（进程调度），调度对象是进程，有抢占和非抢占两种方式。

3）中级调度：引入中级调度的目的主要是提高内存利用率和系统吞吐量。是那些暂时不能运行的进程不占用宝贵的内存资源，把他们调至外存等待，此时进程状态为挂起状态。当这些进程重新具备运行条件时，再将其调入内存。

（3）调度的基本准则与方式

面向用户的准则：**1）周转时间短**，掌握**平均周转时间**，**带权周转时间**。

2）响应时间快 3)截止时间保证 4）优先权准则

面向系统的准则：1）系统吞吐率高 2）处理机利用率高 3）各类资源平衡利用

（4）各种调度算法及其评价

1）先来先服务（FCFS），有利于长作业（进程），不利于短作业（进程）

2）短作业优先（SP(J)F），缺点：对长作业不利，完全未考虑作业紧迫程度。

3）高优先权优先调度（FPF)：又可分为抢占式优先权和非抢占式优先权，优先权有静态和动态两种。

4）**高相应比优先调度算法，相应比=(等待时间+要求服务时间)/要求服务时间，**既考虑到短作业，又考虑了作业到达的先后次序。但是要进行相应比的计算，增加了开销。

5）**基于时间片转轮调度算法，多及反馈队列调度算法**。

（5）死锁问题及其处理方法，包括死锁的概念和原因，产生死锁的必要条件，死锁处理策略，死锁的预防、避免、以及检测和解除。

死锁：多个进程在运行过程中因争夺资源而造成的一种僵局。

产生死锁的原因：1、竞争资源，只有对不可剥夺资源的竞争才可能导致死锁，对可剥夺资源的竞争不会导致死锁 **2、进程间推进顺序非法**

产生死锁的必要条件：1）互斥条件，进程所分配的资源进行排他性使用，即在一段时间某资源只能由一个进程占用。**2）请求和保持条件**，进程已经保持了至少一个资源，又提出了新的资源请求，而该资源又被其他资源占有，此时请求进程阻塞，但又对自己获得的其他资源不放。**3）不剥夺条件**，进程已获得的资源，在未使用完之前不能被剥夺，只能在使用完时由自己释放。4）环路等待条件，发生死锁时，必然存在一个进程----资源的环形链。

死锁处理的基本方法：1）预防死锁 2）避免死锁 3）检测死锁 4）解除死锁

预防死锁：使四个必要条件的 2,3,4 之一不能成立，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：银行家算法，采用预分配策略检查分配完成时，系统是否属于安全状态，重点掌握。

安全状态：能找到一个资源分配的序列能让所有进程都顺利完成。

死锁的检测：注意**资源分配图**和**死锁定理**

死锁的解除：两种方法：**1）剥夺资源 2）撤销进程 3）进程回退**

4．内存管理

（1）内存管理的基本概念：链接与装入，逻辑地址与物理地址空间，对换与覆盖，重定位

源程序变为可执行程序的过程：**首先是编译**，由编译程序将用户源代码编译成若干目标模块；**其次是连接**，由链接程序将个目标模块，以及他们所要的库函数链接在一起，形成一个完整

的装入模块；**最后是装入**，由装入程序将装入到内存。

程序的链接：由链接程序将目标模块和所需库函数链接在一起，形成完整的装入模块。

链接有三种方式：

1）静态链接：在程序运行前，将目标模块和所需的库函数，连接成一个完整的装入模块，以后不再分开。

2）装入时动态链接：将得到的目标模块，再装入内存时，采用边装入边链接的链接方式。优点：1）**便于更新和修改**，各目标模块分开存放，所以要修改和更新目标模块非常容易。2）**便于实现对目标模块的共享。**

3）运行时动态链接：对某些模块的链接推迟到程序执行时才去链接。

程序的装入：由装入程序将装入模块装入到内存。**优点：**不仅加快了程序的装入过程，而且可以节省大量内存空间。

装入有三种方式：

1）绝对装入方式：

2）可重定位装入（静态重定位）：特点：一个作业装入内存时，必须给他分配要求的全部内存空间，若没有足够的内存则不能装入。此外，作业一旦装入内存，整个运行期间不能在内存中移动，也不能申请内存空间。

3）动态运行时装入（动态重定位）：特点：可将程序分配到不连续的存储区；在程序装入前，装入它的部分代码即可运行，运行期间，根据需要动态申请分配内存；便于程序段共享，可以向用户提供一个比存储空间大的多的地址空间。

逻辑地址与物理地址：

逻辑地址空间：编译后每个每个目标模块都从 0 号单元开始编址，这称为该目标模块的相对地址（逻辑地址）。当连接程序将各个目标模块链接一个完整的可执行目标程序时，链接程序顺序依次按各个模块的相对地址构成统一的从 0 号单元开始编址的**逻辑地址空间。**

物理地址空间：是指内存中物理单元的集合，是地址转化的最终地址。当装入程序将可执行代码装入内存时，必须通过地址转换将逻辑地址装换为物理地址，**这个过程称为地址重定位。**

对换与覆盖：

对换（交换）：把处于等待状态的程序从内存移到辅存，再把内存空间腾出来，这一过程称为**换出，**再把准备好竞争 CPU 运行的程序从辅存移到内存，这一过程称为**换入。中级调度采用的就是对换技术。**

覆盖：由于程序运行时，并非任何时候都要访问程序及数据的各个部分，因此可以把用户空间分为**一个固定区和若干个覆盖区，将经常活跃的部分放在固定区，其余部分按调用关系分段。首先将那些即将访问的段放入覆盖区，其他段放在外存中，在需要调用前，系统将其调入覆盖区，覆盖原有的段。**

（2）连续内存分配方法，离散内存分配方法（分页、分段、段页），

连续内存分配：是指为一个程序分配一个连续的内存空间。分配方式主要有三种:

1）单一连续分配：此方式下分为系统区和用户区，低地址的系统区仅供操作系统使用。内存中永远只有一道程序**。优点：**简单，无外部碎片，可采用覆盖技术，不要额外的技术支持。**缺点：**只能用于单任务，单用户的操作系统，有内部碎片，存储器利用率极低。

2）固定分区分配：最简单的一种可运行多道程序的存储管理方式。将内存用户空间划分为若干个固定大小的区域，在每个分区中只装入一道作业，允许几道作业并发运行。

划分分区的方法：1）分区大小相等 **2)**分区大小不等

固定分区分配通常会建立一张**分区使用表，**每个表项包括包括每个分区的**起始地址，大小，和状态（是否已分配），当某一用户程序要装入时，由分配程序检索该表，找到一个大小满足且未分配的分区，分配给该程序。并置状态为已分配。**

缺点：程序可能太大而放不进任何一个分区；主存利用率低，有内部碎片。

3）动态分区分配：不预先划分内存，而是在进程装入内存时，根据进程的大小动态的建立分区，并使分区大小刚好适合进程需要。在开始 i 分配时很好，但在之后会导致内存中出现许多小的内存块，随着时间推移，内存中会出现越来越多的碎片，内存利用率随之下降。这碎片称为外部碎片，克服外部碎片可以采用紧凑技术。

在进程装入或换入时，需要确定分配哪个内存块给进程，这就是动态分区的分配策略。**1）首次适应算法 2）最佳适应算法 3）最坏适应算法 4）临近适应算法（循环首次适应算法，分配内存时从上次查找结束的位置开始查找）**

4）可重定位分区分配（主要看书）

离散内存分配方法：

离散内存分配：允许一个程序分散的装入不相邻的内存分区。**根据分区大小是否固定**分为分页储存管理方式和分段储存管理方式。

基本分页存储管理：分页存储管理将一个进程的逻辑地址空间分成若干个大小相等的片，称为页。相应的，把内存空间分成与页面大小相等的若干个储存块，称为物理块或页框。

页表：实现页号到物理块号的地址映射。

基本分段存储管理：引入的目主要是为了满足程序员在编程和使用上的诸多要求。1）方便编程 2）信息共享 3）信息保护 4）动态增长 5）动态链接

分段和分页的主要区别：1）分页是信息的物理单位，段是信息的逻辑单位 2)页的大小固定且由系统固定，段的长度不固定，决定于用户编写程序的长度 3）分页的作业地址空间是一维的，程序员只用一个记忆符即可表示一个地址；分段的地址空间是二维的，程序员在表示一个地址时，既需要给出段名又需要给出段内地址。

段页式存储管理：分段与分页管理相结合，先将用程序分为若干段，再把每个段分为若干个页，地址结构由段号，段内页号，页内地址组成。**地址变换过程：**配置一个段表寄存器，其中存放段表始址和段表长。进行地址变换时，先将段号与段表长比较，若小于段表长，表示未越界，于是利用段表始址和段号求出该段所对应的段表项在段表中的位置，从中得到该段的页表始址，并利用段内页号来获得对应的页表项位置，从中得出该页所在的物理块号，在利用块号和页内地址构成物理地址。(三次访问内存!)

（3）虚拟内存分配方法（虚拟内存的概念，局部性原理，实现虚拟内存所需的硬件和软件支持，请求分页（段）管理，页面置换算法）

虚拟存储器：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。

局部性原理：时间局部性：典型原因：程序中存在大量的循环操作。

空间局部性：程序在一定时间内访问的地址，可能集中在一定范围内。

实现虚拟内存所需的硬件和软件支持：

请求分页管理：硬件支持 1）一定容量的内存和外存 2）页表机制 3）中断机构 4）地址变换机构　软件支持 1）实现请求调页的软件 2）实现页面置换的软件

请求分段管理：硬件支持 1）一定容量的内存和外存 2）段表机制 3）中断机构 4）地址变换机构　软件支持 1）1）实现请求调段的软件 2）实现段置换的软件

虚拟存储器的特征：1）多次性 2）对换性 3）虚拟性

请求分页（段）管理：

页表机制：页表除了含有**页号及其对对应的物理块号**以外，还有**状态位：**用于指示该页是否调入内存。**访问字段**：记录一段时间内被访问次数。**修改位**：表示该页在调入内存后是否被修改过。

缺页中断机构：缺页中断是一种特殊的中断，与一般中断相比，主要区别有以下两个方面：**1）指令执行期间产生和处理中断信号 2）一条指令执行期间，可能产生多次缺页中断。**

地址变换机构：P146，一定要理解。

物理块的分配策略：1）固定分配局部置换 2）可变分配全局置换 3）可变分配局部置换

调入页面的时机：1）预调页策略 2）请求调页策略（每次调入一页）

页面置换算法（重点掌握）：

1）最佳置换算法（opt）

2）先进先出页面置换算法（FIFO）

3）最近最久未使用置换算法（LRU）

4）时钟置换算法（CLOCK）

5）改进型 clock 置换算法

（4）内存保护与共享

内存保护：保护操作系统不受用户进程的影响，同时保护用户进程不受其他用户进程的影响。主要有两种方法：

1）在 CPU 设置一对上下限寄存器，存放用户作业在主存中的上限和下限地址，每当 CPU 访问一个地址时，分别和两个寄存器的值比较，判断有无越界。

2）采用重定位寄存器（基址寄存器）和界地址寄存器（限长寄存器）。重定位寄存器含有最小的物理地址值，界地址寄存器含逻辑地址最大值。每个逻辑地址必须小于界地址寄存器，内存管理机构动态的将逻辑地址与界地址寄存器进行比较，若未发生地址越界，则加上重定位地址寄存器的值形成物理地址。（重定位寄存器用来加，界地址寄存器用来比）

内存共享：

（5）抖动的概念和处理方法

抖动：刚刚换出的页面马上就要换入主存，刚刚换入的页面马上就要换出主存，这种频繁的页面调度现象称为**抖动或颠簸。**

主要原因是：某个进程频繁访问的页面数目高于可用的物理页帧数目。

处理方法:

5．设备管理

（1）I/O 体系结构

整个 IO 系统可视为具有四个层次的系统结构：用户层 IO、设备独立性软件、设备驱动程序、中断处理程序

用户层 IO：实现与用户交互的接口。

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、设备保护及设备分配与释放等等，同时为设备管理与数据传送提供必要的存储空间。

设备驱动程序：与硬件直接相关，负责具体实现系统对设备发出的操作指令，驱动 IO 设备工作的驱动程序。

中断处理程序：用户保存被中断进程的 CPU 环境，处理完并回复被中断程序的现场后，返回到被中断进程。

设备控制器的基本功能：1）接受和识别命令 2）数据交换 3）标识和报告设备状态 4）地址识别 5）数据缓冲 6）差错控制

设备控制器的组成：1）设备控制器与处理机接口 2）设备控制器与设备的接口 3）IO 逻辑

（2）I/O 控制方法

外围设备与内存的输入输出控制有四种

1）程序直接控制

优点：简单易于实现 **缺点：**CPU 和 IO 设备串行工作，导致 CPU 利用率相当低。

2）中断驱动方式：

中断驱动方式的思想是：允许 IO 设备主动打断 CPU 运行并请求服务，从而“解放”CPU，使得其向 IO 控制器发出读命令后，可以继续做其他有用的工作。

优点：大大提高了 CPU 利用率 **缺点：**数据中每个字在存储器与 IO 设备控制器之间的传输必须经过 CPU，导致了终端驱动仍然会消耗较多的 CPU 时间。

3）DMA 方式（直接存储器存取）：

基本思想：在 IO 设备和内存之间开辟直接的数据交换通路，彻底解放 CPU。

特点：1）基本单位数据块 **2）**所传送的数据，是从设备直接送入内存的，或者相反 **3）**仅在传送一个或多个数据块开始或结束时，才需要 CPU 的干预，整块数据的传送是在 DMA 控制器的控制下完成的。

DMA 控制器的组成（机组要求 P168）：

DMA 的工作方式：CPU 接到 DMA 的工作请求时，给 IO 控制器发一条命令，启动 DMA 控制器，然后继续其他工作。之后 CPU 就把控制操作委托给 DMA 控制器，由该控制器负责处理。DMA 控制器直接与存储器交互，传送整个数据块，每次传送一个字，这个过程不需要 CPU 的参与。传送完成后，DMA 控制器发送一个中断信号给 CPU。

DMA 与中断驱动方式的区别是：1）中断驱动方式在每个数据传送需要中断 CPU，而 DMA 控制方式则是在所要求传送的一批数据全部传送结束时才中断 CPU **2）**中断驱动方式数据传送是在中断处理时由 CPU 控制完成，而 DMA 控制方式则是在 DMA 控制器的控制下完成的。

4）通道控制方式：

IO 通道：是指专门负责输入输出的处理机，是 DMA 方式的发展，他可以进一步减少 CPU 的干预，即把对一个控制块的读写为单位的干预，减少为对一组数据块的读写的及有关控制和管理为单位的干预。同时，又可以实现 CPU、通道和 IO 设备三者的并行操作，从而更有效的提高整个系统的资源利用率。

IO 通道与一般处理机的区别：通道指令的类型单一，没有自己的内存，通道执行的通道程序是放在主机的内存中的，也就是说通道与 CPU 共享内存。

IO 通道与 DMA 方式的差别：DMA 方式需要 CPU 来控制和传输数据块的大小以及传输的内存位置，而通道方式中的这些信息是由通道控制的；另外，每个 DMA 控制器对应一台设备与内存传递数据，而一个通道可以控制多台设备与内存的数据交换。

（3）I/O 分配中的数据结构和分配方法

在设备分配时，通常借助一些表格：设备控制表、控制器控制表、通道控制表、系统设备表。

设备控制表（DCT）：每个设备配备一张设备控制表，用于记录本设备的情况。

控制器控制表（COCT）：系统为每个控制器设置了一张用于记录本控制情况的控制器控制表。

通道控制表（CHCT）：每个通道都配有一张通道控制表。

系统设备表（SDT）：这是系统范围的数据结构，记录了系统中全部设备的情况。每个设备占有一个表目，其中包括设备类型，设备标识符，设备控制表以及设备驱动程序的入口等。

设备分配时应考虑到的因素：1）设备固有属性 2）设备分配算法 3）设备分配时的安全性 4）设备独立性

设备的固有属性有三种 1）独占性设备 2）共享性设备 3）可虚拟设备

设备分配算法：1）先来先服务 2）优先级高者有优先。

设备分配中的安全性：1）安全分配方式：放弃了“请求和保持”条件，CPU 与 IO 串行工作 2）不安全分配方式：可能造成死锁。

（4）通道和通道程序

通道：一种特殊的处理机，具有执行 IO 指令的能力，并通过执行通道程序来控制 IO 操作。类型：1）字节多路通道 2）数组选择通道 3）数组多路通道

字节多路通道：按字节交叉方式工作。含有许多非分配型子通道，这些子通道按时间片转轮方式共享主通道。只要扫描各子通道速度足够快，而连接到子通道设备速率不是太高时，不会丢失信息。（不适用于连接高速设备）

数组选择通道：可以连接多台高速设备，但只有一个分配型子通道，在一段时间内只能执行一道通道程序。通道利用率极低。

数组多路通道：前两者优点结合。既具有很高的数据传送速率，又有令人满意的 CPU 利用率。

瓶颈问题：通道价格昂贵，因此较少，通道不足会造成瓶颈问题。，解决方法：增加设备到主机间的通路，而不增加通道。为了防止出现瓶颈。通常采用多通路的 IO 系统结构。

通道程序：通道是通过执行通道程序，并与设备控制器共同实现对 IO 设备的控制的。由一系列通道指令（命令）构成。与一般机器指令不同，在他的每条指令中，都包含如下信息:1)操作码 2）内存地址 3）技计数 4）通道程序结束位 5）记录结束标志。

（5）设备独立性及其实现方法

为提高 OS 的可适应性和可扩展性，必须实现**设备独立性（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。**

好处：1）设备分配时的灵活性 2）易于实现 IO 重定向。

实现方法：在应用程序中使用逻辑设备名来请求使用某类设备，在系统中设置一张逻辑设备表（LUT），用于将逻辑设备名映射成物理设备名。表的表目中包含三项：逻辑设备名，物理设备名和设备驱动程序的入口地址。

建立逻辑设备表的两种方式：**1）**在整个系统中只设置一张 LUT，适用于单用户系统 **2）**为每个用户设置一张 LUT。

（6）虚拟设备和 SPOOLing 技术

SPOOLing（假脱机）技术可以将一台物理 IO 设备虚拟为多台逻辑 IO 设备，允许多个用户共享一台物理 IO 设备。实质上是一种**以空间换时间**的技术，而请求分页的页面调度算法时典型的**以时间换空间**。

SPOOLing 是对脱机输入、输出系统的模拟。必须建立在多道程序功能的基础上，而且还要

有高速随机外存的支持，通常采用磁盘存储技术。

SPOOLing 主要有三个部分：

1）输入井和输出井，在磁盘上开辟的两个大存储空间，输入井是模拟脱机输入的磁盘设备，用于暂存 IO 设备输出得数据，输出井是模拟脱机输出的设备，用于暂存用户程序的输出数据。

2）输入缓冲区和输出缓冲区，为缓和 CPU 和 IO 设备速度不匹配的矛盾，在内存中开辟两个缓冲区：输入和输出缓冲区，输入缓冲区用来暂存由输入设备送来的数据，以后在传送到输入井；输出缓冲区用于暂存从输出井送来的数据，以后在传送给输出设备。

3）输入进程和输出进程：利用这两个进程模拟脱机 IO 时的外围控制机。**其中输入进程模拟脱机输入时的外围控制机，将用户要求的数据从输入机通过输入缓冲区在送到输入井，当 CPU 需要输入数据时，直接从输入井读到内存**；**输出进程模拟输出时的外围控制机，把用户要求输出的数据先从内存送到输出井，待输出设备空闲时，再将输出井中的数据经过输出缓冲区送到输出设备上。**

打印机是经常要用到的**独占输出设备**。利用 SPOOLing 可以改造成共多个用户共享的设备，从而提高设备利用率。

（7）缓冲管理

引入缓冲区的主要原因：1）缓和 CPU 和 IO 设备间速度的不匹配的矛盾 2）减少对 CPU 的中断频率，放宽 CPU 中断响应时间的限制 3）提高 CPU 和 IO 设备之间的并行性。缓冲区数据非空时，不能往缓冲区冲入数据，只能把数据传出；当缓冲区为空时，可以冲入数据，但必须把缓冲区充满后，才能从缓冲区把数据传出。

根据系统设置缓冲器的个数，缓冲技术可分为四种：

1）单缓冲：从磁盘把一块数据输入到缓冲区的时间为 **T**，操作系统将该缓冲区的数据送到用户区的时间为 **M**，CPU 对这块数据的处理时间为 **C，T 和 C 是可以并行的**，因此可把系统对每一块数据的处理时间表示为 **max（T,C）+M**。

2）双缓冲：为了加快输入输出速度，提高设备利用率，引入双缓冲也成缓冲对换。处理一块数据所用时间为 **max（C+M,T），单缓冲机制不允许双方同时向对方发送数据，而双缓冲可以。**

3）循环缓冲：包含**多个大小相等的缓冲区**，每个缓冲区中由一个**链接指针**指向下一个缓冲区，构成一个环形。

4）缓冲池：在池中设置多个可供进程共享的缓冲区。

组成：**1）空闲缓冲区（空缓冲队列） 2）装满输入数据的缓冲区（输入缓冲队列） 3）装满输出数据的缓冲区（输出缓冲队列）。**

（8）设备处理与 I/O 软件

IO 软件总体设计目标是高效率和通用性。

IO 软件应实现的目标：1）与具体设备无关 2）统一命名 3）对错误的处理 4）缓冲技术 5）设备分配和释放 6）IO 控制方式

四个层次：用户层软件、设备独立性软件、设备驱动程序、中断处理程序。

（9）设备分配

6．磁盘与文件系统

（1）磁盘的结构和基本概念

磁盘设备包括多或一个个物理盘片，每个磁盘片分为一个或两个储存面。每个磁盘被组织成若干个同心环，称为磁道。每个磁道从逻辑上划分为多个扇区（通常为 512B），一个扇区称为一个盘块（数据块），各个扇区之间保留一定的间隙。

磁盘地址：柱面号-盘面号-扇区号

磁盘访问时间包括三部分：寻道时间，旋转延迟时间，传输时间。

1）寻道时间：把磁头移动到指定磁道上的时间。是移动磁臂时间 s 与磁头移动 n 条磁道所花费时间之和。T=m*n+s

2）旋转延迟时间：指扇区移动到磁头下面所经历的时间。T=1/2r,r 为磁盘旋转速度

3）传输时间：把数据从磁盘读入或写入磁盘所经历的时间，T=b/rN，b 为每次所读的字节数，N 为一个磁道上的字节数。

（2）磁盘的调度

调度算法直接决定寻找时间，从而决定总的存取时间。

常用的磁盘调度算法：

1）先来先服务（FCFS）

2）最短寻找时间优先（SSTF）

3）扫描算法（SCAN）

4）循环扫描算法（C-SCAN）

（3）磁盘的性能改善和容错

提高磁盘 IO 速度的主要技术是采用**磁盘高速缓存，利用内存中的存储空间来暂存从磁盘读出一系列盘块中的信息**。

数据交付方式有数据交付和指针交付。

置换算法：。。。。不少系统在设计其高速缓存的置换算法时，除了考虑最近最久未使用，还考虑以下几点：1）访问频率 2）可预见性 3）数据一致性

提高 IO 速度的其他方法：1）提前读 2）延迟写 3）优化物理块分布 4）虚拟盘

容错技术：廉价磁盘冗余阵列 RAID（除了 0 级以外），RAID 的优点： 1）可靠性高 2）磁盘 IO 速度高 3）性价比高

（4）外存分配方法与物理文件组织

文件的物理结构又称文件的存储结构，是指文件在外存上的存储组织形式，不仅与存储介质的存储性能有关，而且与采用的外存分配方式有关。

外存分配方法：连续分配、连接分配、索引分配

1）连续分配：要求为每个文件分配一组相邻的盘块。一组盘块的地址定义了定义了磁盘上一段线性地址。这样形成的文件结构是**顺序文件结构**，此时的物理文件称为**顺序文件**。

优点：顺序访问容易、顺序访问速度快。缺点：要求有连续的存储空间、必须事先知道文件长度。

2）链接分配：通过每个盘快上的链接指针，将同属于一个文件的离散盘块链接成一个链表，

这样形成的物理文件称为链接文件。

优点：消除了外部碎片，显著提高了外存利用率；当文件动态增长时，可动态的在为他分配盘块，故无需实现知道文件大小；此外，对文件的增删改也很方便。

有两种方式：1）显式链接 2）隐式链接

1）隐式链接：在文件目录的每个目录项中，都需要含有指向链接文件的第一个盘块和最后一个盘块，每个盘块中含有指向下一个盘块的指针。

主要问题：只适合于顺序访问，对于随机访问及其低效，其次只通过一大批指针把离散的盘块链接起来，可靠性差，只要任何一个盘块的指针出现问题，都将会导致整条链断裂。

为了提高检索速度和减少指针所占用的存储空间，可以将几个盘块组成一个**簇**，在盘块分配时，以簇为单位。这样会成倍的减少超找指定块的时间。

2）显式链接：把用于链接文件各个物理块的指针，显式的存放在内存的一张链接表中。该表在整个磁盘仅设置一张。**由于查找记录是在内存中进行的，不仅显著的提高了检索的速度，而且大大减少了访问磁盘的次数，**该表也成为**文件分配表（FAT）**。

3）索引分配：链接分配的缺点：1）不支持高效的直接存取 2）FAT 占用较大的内存空间。

索引分配（文件的物理组织结构是　索引式文件机构）为每个文件分配一个索引块（表），再把分配给该文件的所有盘块号都记录在该索引块中，因而该索引块就是一个含有多个盘块号的数组。在建立一个文件时，只需在为其建立的目录项中填上指向该索引块的指针。

索引分配优点：支持直接访问，当要读第 i 个盘块时，可以直接从索引块中找到第 i 个盘块的盘块号。此外，索引分配不会产生外部碎片。文件较大时，索引优于链接。

索引分配的主要问题：可能会花费较多的外存空间。

多级索引

混合索引。

（5）文件存储空间的管理

储存空间的基本分配单位是磁盘块而非字节。管理包括对空闲块的组织，分配和回收。

1）空闲表法：属于连续分配方式，系统为外存上所有的空闲区建立一张空闲表。　存储空间的分配：同样采取首次适应算法，循环首次适应算法等，回收时，也类似于内存回收的方法，即要考虑回收区是否与空闲表中插入点的前区和后区相临接，对相临接者予以合并。**文件较小时，采取空闲表法。**

2）空闲链表法：将所有空闲盘区拉成一条空闲链，根据构成链所用的基本元素不同，可把

链表分为两种形式**：空闲盘块链，**以盘块为单位拉成一条链**，优缺点：**用于分配和回收一个盘块的过程非常简单，但在为一个文件分配盘块时，可能要重复操作多次**；空闲盘区链，**将磁盘上所有的空闲盘区拉成一条链。分配盘区的算法通常采用首次适应算法，回收时也要考虑合并，**为提高对空闲盘区的检索速度，可以采用显示连接法，为内存中的空闲盘区建立一张链表。**

3）位示图法：位示图是利用二进制中的一位来表示磁盘中一个盘块的使用情况，0 时空闲，1 时分配。

根据位示图**进行盘块的分配时，**分为三步：**1）顺序扫描位示图，找到 0。2）将找到的一个或一组二进制位转化成与之相对应的盘块号。3）修改位示图，使相应位置 1。**

回收分为两步：1）将回收的盘块号转化为位示图中的行号和列号。注意转化公式!!!! 2）修改位示图，使相应位置 0。

4）成组链接法：空闲表法和空闲链表法不适合大型文件，会使链表和空闲表太长。将上述两种方法结合在一起，兼备了优点，而克服了表太长的缺点。

大致思想：把顺序的 n 个空闲扇区的地址保存在第一个空闲扇区内，其后一个空闲扇区则保存另一个顺序空闲扇区的地址，在直至所有空闲扇区均已连接。系统只需要保存一个指向第一个空闲扇区的指针。

（6）逻辑文件组织

文件的逻辑结构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，独立于文件的物理特性，又称为文件组织。

文件逻辑结构的类型：有结构文件；无结构文件；

有结构文件三种：

1）顺序文件：由一系列记录按某种顺序排列所形成的文件，记录通常是记录文件。又分为串结构和顺序结构。串结构，记录之间的顺序与关键字无关。顺序结构，所有记录按关键字顺序排列。

优点：只有顺序文件能储存在磁带上，当每次要读或写一大批文件时，顺序文件的存储效率最高。缺：顺序文件对于查找、修改、增加、删除等操作比较困难。

2）索引文件：对于定长记录可以方便的实现顺序存取，但对于边长记录难以实现直接存取。可以为变长文件记录建立一张索引表（本身是一个定长记录的顺序文件）。

3）顺序索引文件：将顺序文件中的所有记录分为若干组，为顺序文件建立一张索引表，在索引表中为魅族第一条记录建立一个索引项，，其中含有该记录的关键字值，和指向该记录的指针。

4）直接文件：记录值本身就决定了自己的物理地址，键值转换 **哈希文件：**利用散列函数，将键值转化为相应的记录的地址，可能会发生冲突。

（7）文件的基本操作

创建文件、删除文件、读文件、写文件、截断文件、设置文件读写位置。

文件的打开和关闭。P207 熟读

（8）文件目录及其管理

对目录管理的要求如下：1）实现“按名存取”2）提高对目录的检索速度 3）文件共享 4）允许文件重名。

文件控制块（FCB）：是用来存放控制文件所需要的各种控制信息的数据结构。文件控制块的有序集合称为文件目录，即一个文件控制块就是一个文件目录项。一个文件目录也可以看做一个文件，称为目录文件。

文件控制块通常包含三类信息：基本信息：文件名，文件物理位置，文件逻辑结构，文件物理结构；**存取控制信息：**文件存取权限；**使用信息：**文件建立时间，修改时间。

索引结点：在检索目录文件的过程中，只用到了文件名，其他一些对该文件的描述信息，在检索目录时，一概不用，所以没必要调入内存。便采用了把文件名与文件描述信息分开的办法。文件描述信息单独形成一个称谓**索引结点**的数据结构。

在文件目录中，每个目录项仅由文件名和指向该文件所对应的 i 结点的指针构成。

目录结构：目录结构关系到文件系统的存取速度，也关系到文件的共享性和安全性**。通常有单击目录，两级目录和多级目录。**

单级目录：只建立一张目录表，每个文件占有一个目录项。访问时，先按文件名在该目录中找到对应的 FCB**。单级目录实现了“按名存取”但是，查找速度慢，不允许重名，不便于实现文件共享。**

两级目录：优点：提高了检索目录的速度；在不同用户名的目录中，可以使用相同文件名；不同用户可以使用不同的文件名来访问系统中的同一个共享文件。

多级目录文件（树形目录文件）：优点：可以很方便的对文件进行分类，层次结构清晰，能有效的进行文件的管理和保护。但是在树形文件中，查找一个文件时，需要按路径名逐级访问，增加了磁盘访问次数，影响查询速度。

（9）文件共享和保护

文件共享是多个用户共享同一文件，系统只要保留该文件的一个副本。

文件共享方法：1）基于索引结点的共享方式（硬链接） 2）利用符号链实现文件共享（软连接）

文件保护：文件保护通过口令保护，加密保护，访问控制等方式实现。口令保护和加密保护是为了防止用户文件被他人存取或窃取，访问控制是控制用户对文件的访问方式。

检查点：主要目的是对事务记录表中的事务记录清理工作经常化。

7．操作系统接口

（1）操作系统接口类型

1）用户接口 2）程序接口，是程序能取得操作系统服务的唯一途径，大多数程序接口由一组系统调用组成。

（2）系统调用概念

一方面由于系统提供保护机制，防止程序至直接调用操作系统得过程，从而避免不安全性，另一方面，应用程序必须取得操作系统提供的服务。为此，操作系统提供系统调用，使应用程序通过提供系统调用的方法，间接调用操作系统的相关过程。

（3）系统调用的类型和实现方法

类型：进程控制类系统调用、文件操纵类系统调用、进程通信类系统调用。

实现：P266 熟读。