计组、OS

操作系统是什么?

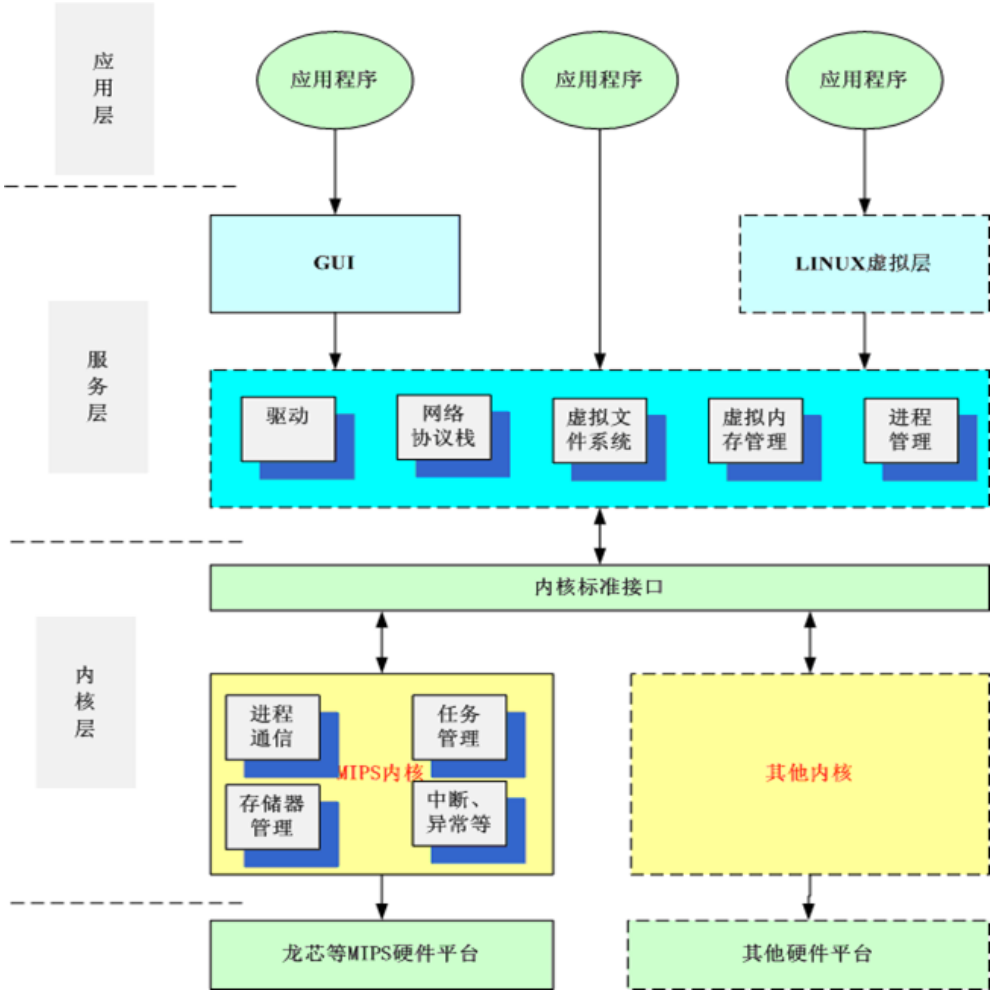
操作系统是最基本的系统软件，控制和管理整个计算机系统的软硬件资源，向上为进程提供抽象的接口，向下管理好硬件资源如CPU、内存、外存、IO设备。

操作系统的基本特征有：

- 并发：并发是指两个或多个事件在同一时间间隔内发生，例如引入进程的概念后，进程轮流使用CPU，实现了程序并发执行；并行是指两个或多个事件在同一时刻发生，如鸟同时扇动两个翅膀。
- 共享：系统中的资源可供内存中多个并发进程使用。互斥共享：临界资源也就是一段时间内只允许一个进程访问的资源如打印机需要互斥共享，同时访问：允许多个进程在一段时间内轮流访问如磁盘。
- 虚拟：将物理设备映射为多个虚拟设备：虚拟处理机（多道程序并发使用一台虚拟机）、虚拟内存、虚拟设备
- 异步：进程的执行是走走停停地，以不可预知的速度向前推进。操作系统需要保证在同样的环境下每次运行进程都有同样的结果。

内核

- 强内核：—基于传统的集中式操作系统的内核结构，系统调用式通过程序陷入内核实现，内核完成相应的服务后返回应用程序，同时返回结果给用户。
- 微内核：内核尽可能小，功能尽可能少（基本），把其他所有功能以服务器的形式放到核外的用户级来完成



多道程序设计

内存中同时存放几个作业，即允许多个进程交替使用CPU，提高了CPU的利用率
这些程序在宏观上并行（几道程序都处于运行过程中，从用户的角度来看，它们在同时推进），微观上串行，任意一个时刻，只能有一个程序占有处理机，从处理机的角度来看，多道程序轮流使用处理机，它们是交替推进的
硬件上需要中断系统；通道技术来支持

操作系统做了什么?

操作系统是计算机资源的管理者

- 处理机管理（进程管理）：进程控制，同步、通信，调度
- 存储器管理：内存分配、地址映射、内存保护
- 文件管理：存储空间管理、目录管理
- 设备管理：缓冲管理、虚拟设备
- 用户接口：—命令接口—程序接口—图形接口

系统调用

系统调用就是用户在程序中调用操作系统所提供的一些子功能，比如涉及到资源有关的操作，都必须通过系统调用方式向操作系统提出服务请求，由操作系统代为完成。这样可以保证系统的稳定性和安全性，防止用户随意更改系统的数据或命令。

进程、线程

- 进程是程序在某个数据集合上的一次执行过程，进程包括程序段、数据段、PCB。PCB是进程存在的唯一标志，包含进程id、进程状态、优先级、拥有的资源和处理机信息等。
- 线程是轻量级进程，是程序执行流的最小单元。线程自己不拥有系统资源，只拥有一点运行中必不可少的资源如线程ID、PC、寄存器集合和堆栈。同一进程中的多条线程将共享该进程中的全部系统资源
- 线程是独立调度的最小单位，进程是资源分配的最小单位。引入线程是为了增加程序的并发现性，提高资源利用率。同一个进程中的多个线程可以并发执行，线程切换只需很少开销。

用户级线程：有关线程管理的工作都由应用程序完成，内核意识不到线程的存在
内核级线程：线程管理的所有工作由内核完成

进程切换

保存处理机上下文 -> 更新PCB -> 把PCB移入相应队列(就绪、阻塞) -> 选择另一个进程并更新其PCB -> 更新内存管理的数据结构 -> 恢复处理机上下文

进程同步/互斥

在多道程序环境下，进程是并发执行的，同步是为了协调不同进程之间执行顺序而产生的相互制约关系。

互斥是防止两个进程同时访问临界资源。

一次仅允许一个进程使用的资源称为临界资源。
—临界区(critical section)：临界段，在每个程序中，访问临界资源的那段程序。
实现临界区互斥的方法有:

1. 软件：Peterson算法
2. 硬件：
   - 中断屏蔽方法：某进程在访问临界区时，关中断，访问完临界区再开中断
   - TestAndSet原子指令：读出标志位后把标志设置为真
3. 信号量（PV操作）：
   - 同步：设定信号量empty=max表示还能存放多少商品，full=0表示已经存放多少商品；消费者要消耗某资源，就p(empty)；生产者生产了某商品就v(full)
   - 互斥：设定信号量mutex=1，哪个进程要进入临界区就需要p(empty)，退出临界区就v(empty)

进程通信

1. 高级通信方式
   - 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，通过对这片共享空间进行读写操作实现进程间的信息交换。
   - 消息传递：进程通过OS提供的“发送消息”和“接收消息”原语进行数据交换，其中数据是格式化的消息。
   - 管道通信：管道是用于连接一个读进程和一个写进程以实现它们之间通信的一个共享文件。缓冲区一边写入，一边读出，是半双工通信。
2. 低级通信方式

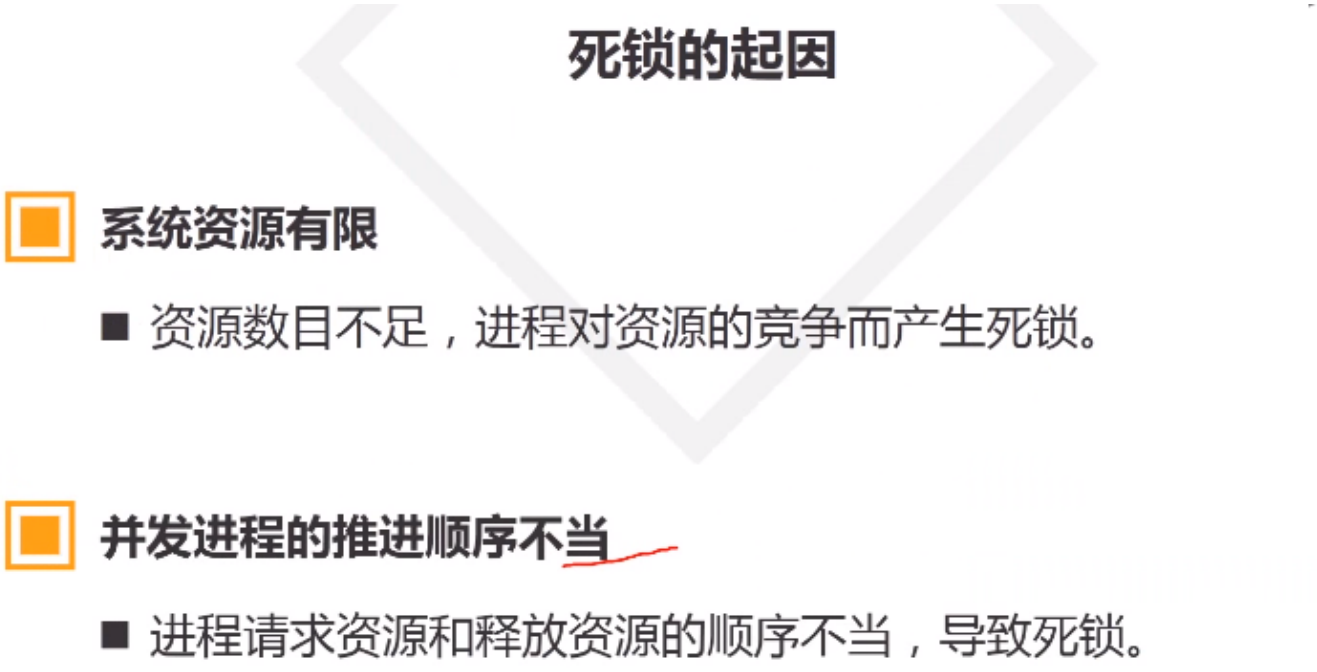
   PV操作（信号量机制）
   - – P：wait(S)原语，申请S资源
   - – V：signal(S)原语，释放S资源

管程

由一组数据及对这组数据操作的定义组成的模块。进程只有通过进入管程并使用管程内部的操作才能访问其中数据。管程是互斥使用的，进程释放管程后需唤醒申请管程资源的等待队列上的进程。

什么是死锁，如何预防/避免/解除?

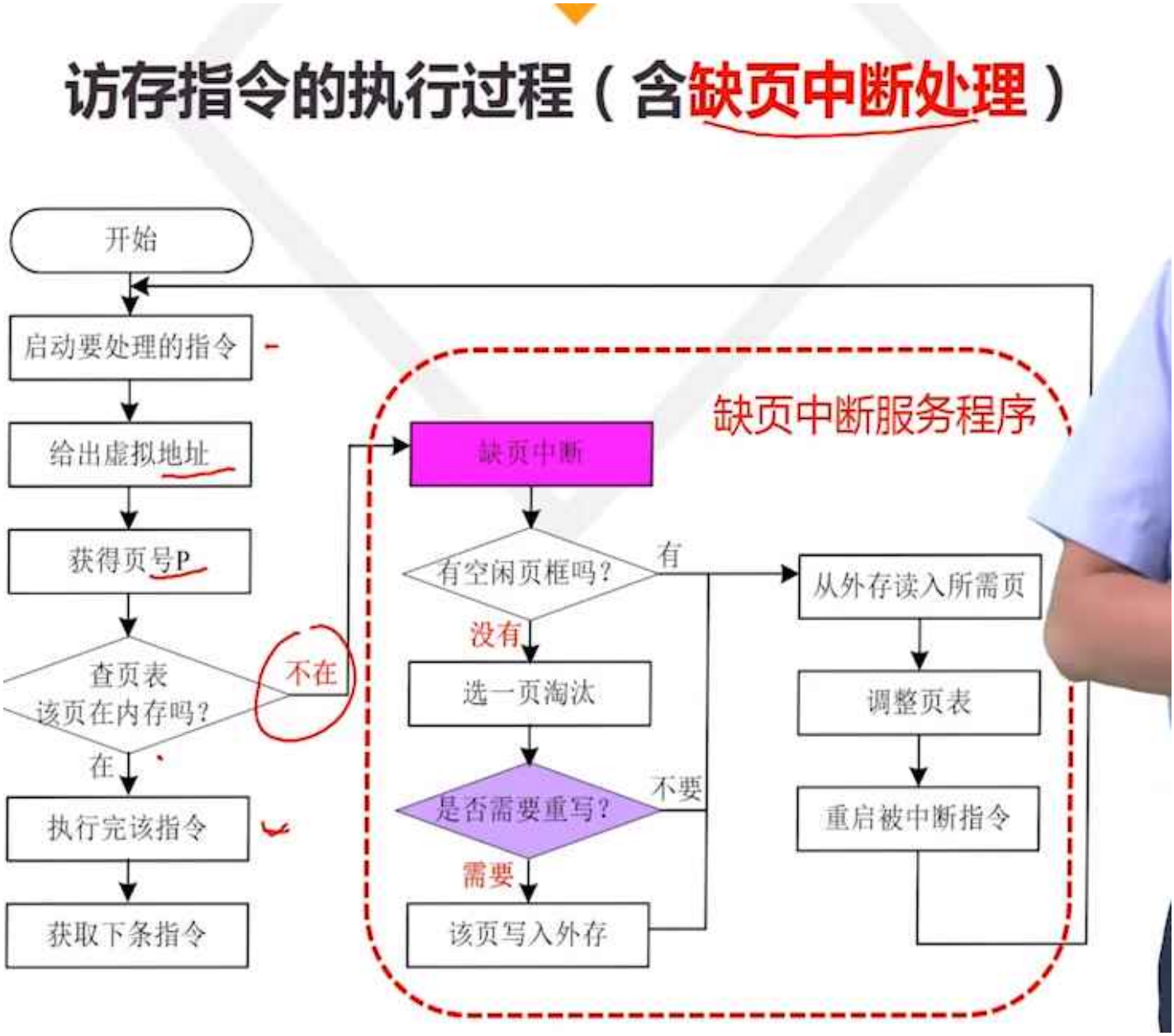
所谓死锁是指多个进程因竞争资源而造成的一种互相等待的僵局，若无外力作用，这些进程都将无法向前推进。
起因:



1. 预防：破死锁的四个必要条件之一：**互**（斥）、**不**（可剥夺）、**请**（求与保持）、**循**（环等待）。
2. 避免：防止系统进入不安全状态

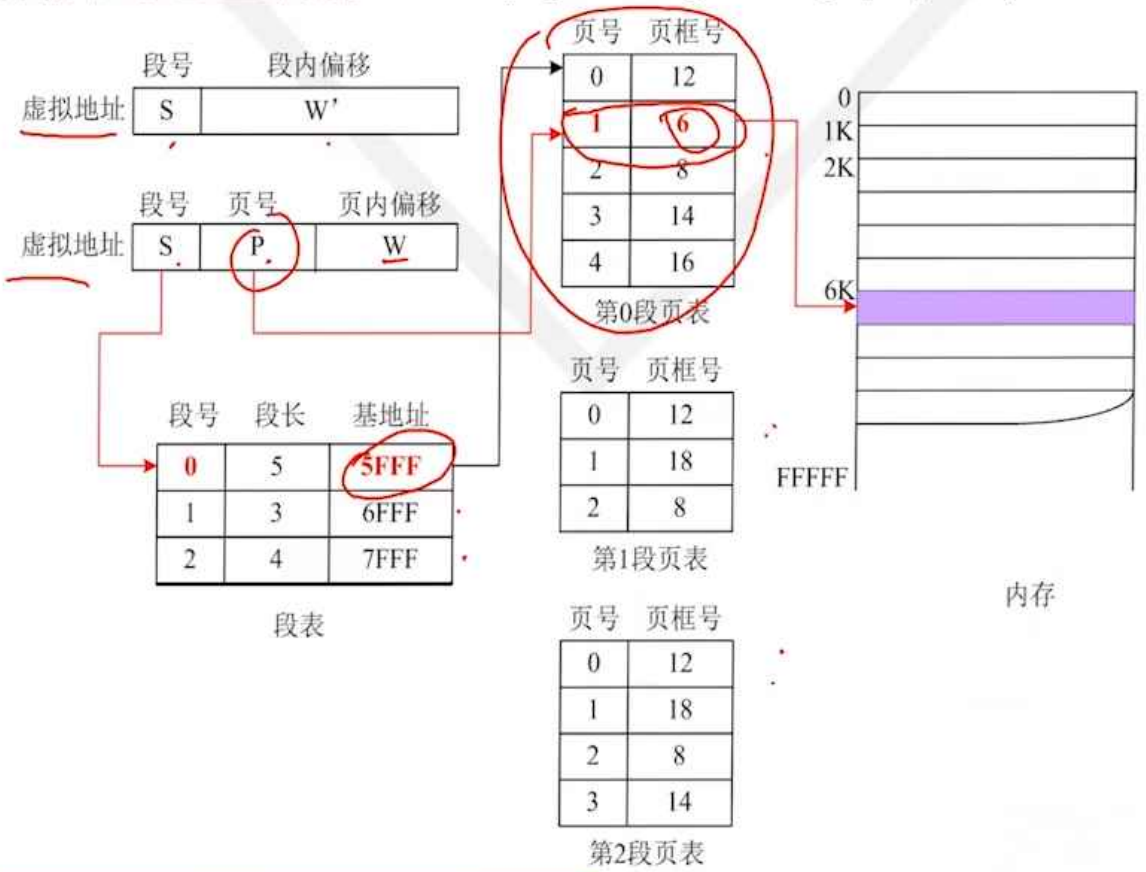
   （银行家算法：进程申请资源时，测试该进程已占用的资源数（Allocation）和本次申请的资源数（request向量）之和是否超过了该进程对资源的最大需求量（MAX矩阵），若未超过则再测试本次申请的资源数之和是否小于可用资源量（available），小于则分配资源）
3. 检测：死锁定理：当且仅当S状态的资源分配图是不可完全简化的时候，S状态为死锁
4. 解除：资源剥夺法（挂起死锁进程并抢夺其资源）、撤销进程法（撤销某些进程）、进程回退法（让进程回退到足以回避死锁的地步）

访存指令如何访存?

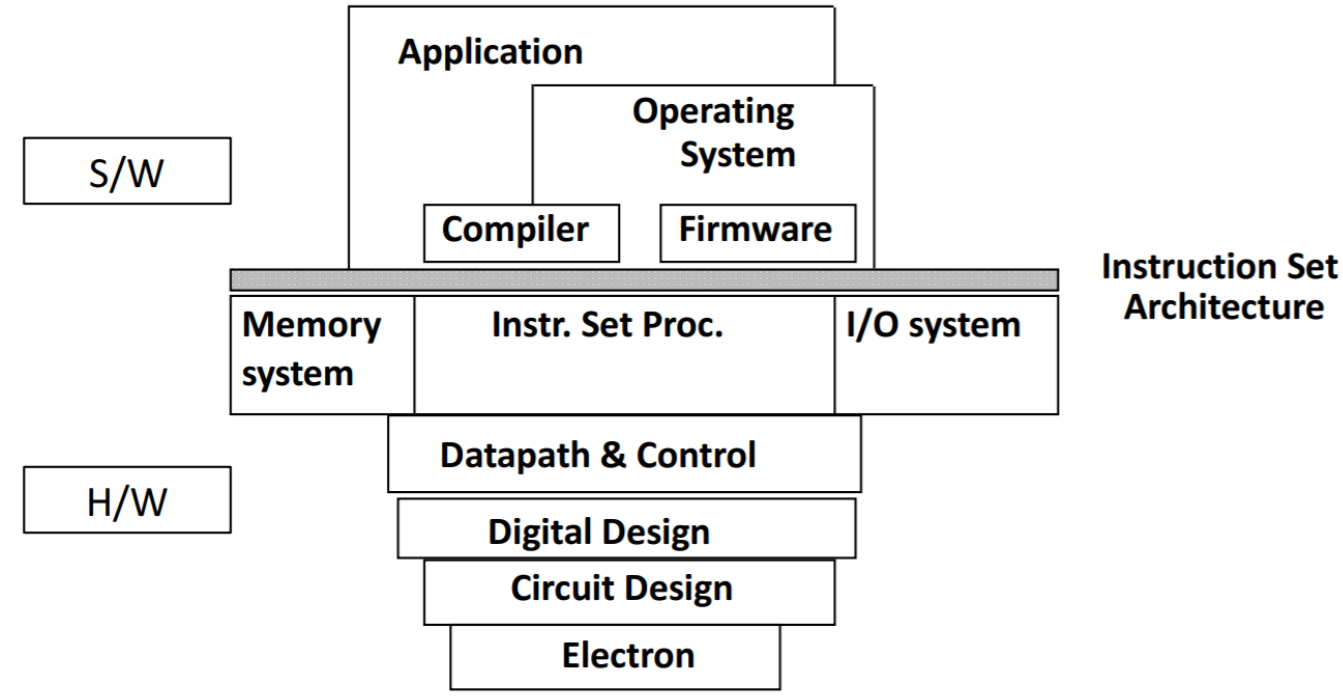

访存指令的执行过程（含缺页中断处理）
开始
启动要处理的指令
给出虚拟地址
获得页号P
查页表
该页在内存吗?
不在
在
执行完该指令
获取下条指令
缺页中断服务程序
缺页中断
有空闲页框吗?
有
没有
选一页淘汰
是否需要重写?
不要
需要
该页写入外存
从外存读入所需页
调整页表
重启被中断指令

段页式存储管理

段页式地址映射：VA(S, W′) → (S, P, W) → MA



整机概念



冯诺依曼机?

1. 计算机硬件系统由运算器、存储器、控制器、输入设备和输出设备5大部分组成，运算器是中心，控制器通过控制线与各个部件通信。
2. 指令和数据均用二进制代码表示，以同等地位存于存储器，按地址访问。
3. 采用“存储程序”的思想，把指令事先输入到存储器中，然后按顺序依次执行指令。

摩尔定律

微处理器的性能每隔18个月提高一倍，而价格下降一半

执行时间

平均CPI：

Op	Freq	CPI_i	Freq x CPI_i
ALU	50%	1	0.5
Load	20%	5	1.0
Store	10%	3	0.3
Branch	20%	2	0.4
			$\sum$ = 2.2

CPU time = Instruction_count x CPI x clock_cycle （指令数*CPI*时钟周期）

原码、反码的0不唯一，补码、移码的0唯一

IEEE 754浮点数

数符s+阶码E+尾数部分M
float：1+8+23=32---阶码偏置值127
Double: 1+11+52=64---阶码偏置值1023

SRAM和DRAM

SRAM，静态随机存储器，原理是双稳态触发器， 读写速度快，生产成本高，多用于容量较小的高速缓冲存储器，不需要刷新操作。
DRAM，动态随机存储器，原理是用电容存储电荷，因此需要定时对其进行刷新，否则2ms后，电容其中存储的电荷将逐渐消失。读写速度较慢，集成度高，生产成本低，多用于容量较大的主存储器。
DRAM的刷新有

1. 集中刷新（快到2ms的时候，停止一切对内存的读取操作，使用0.5μs×64对64行依次刷新，这段时间称为死时间）、
2. 分散刷新（在每个存取操作后绑定一个刷新操作，新的系统的存取周期内前半部分存取，后半部分刷新）、
3. 异步刷新（刷新周期除以行数，2ms ÷ 64 作为每次刷新的周期）三种方式。

主存容量的扩展

1. 位扩展：增加存储器中字的位数。各芯片连接到同一地址线当中，但是分别连在数据总线的一部分，共同组成很多位的数据总线
2. 字扩展：增加存储器中字的数量。各芯片的数据线、地址线都并联，由片选信号来确定哪个芯片有效
3. 字位同时扩展

Cache的工作原理

高速缓冲技术就是利用程序访问的局部性原理，把程序中正在使用的部分存放在一个高速的、容量较小的Cache中，使CPU的访存操作大多数情况下针对Cache进行，从而使程序的执行效率大大提高
P118图
首先根据指令中的访存地址得到主存块号和块内偏移，再访问 主存cache地址映射变换机构 看该内存块是否在cache中。
如果未命中，将该块从主存中装入cache，此时若cache已满还需要根据替换算法替换掉某些cache行
如果命中，就得到了该内存块在cache中对应的行号，计算出cache地址，去访问cache就得到数据了

Cache和主存的映射方式

1. 直接映射：主存块只能装入Cache唯一位置
2. 全相联映射：任何位置
3. 组相联映射：主存块通过直接映射，映射到不同的组中，在组内采用全相联映射。

快表TLB

放在cache里的页表叫快表，它是页表的chche（高速缓冲副本），使用相联存储器组成，提高查找速度。

总线仲裁 集中仲裁方式

1. 链式查询：总线允许信号从串行的从一个部件传到下一个部件
2. 计数器定时查询方式：
3. 独立请求方式：每个设备均有一对总线请求线和总线允许线

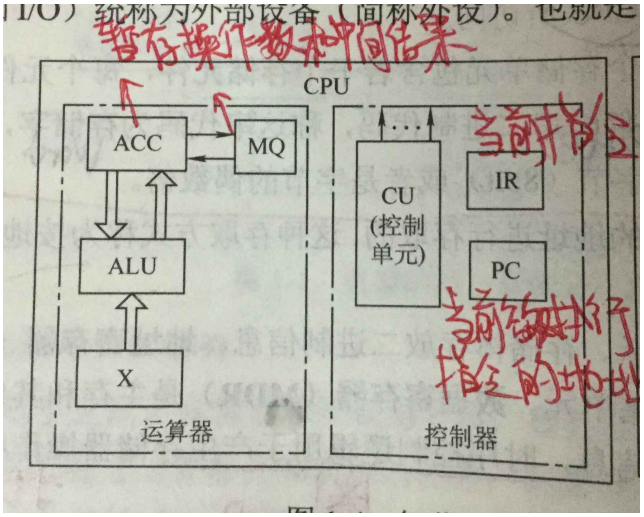
总线周期

申请分配（总线使用权）---》寻址--》传输--》让出（总线控制权）

CPU是什么?

CPU是中央处理器，由运算器和控制器组成。

- 控制器发出每条指令对应的操作序列，对计算机进行控制。控制器由程序计数器PC、指令寄存器IR、控制单元CU组成。
- 运算器接收从控制器送来的命令，完成算数运算或逻辑运算，对数据进行加工和处理。运算器的核心是算数逻辑单元ALU，此外还有累加器、通用寄存器组和程序状态寄存器PSW。



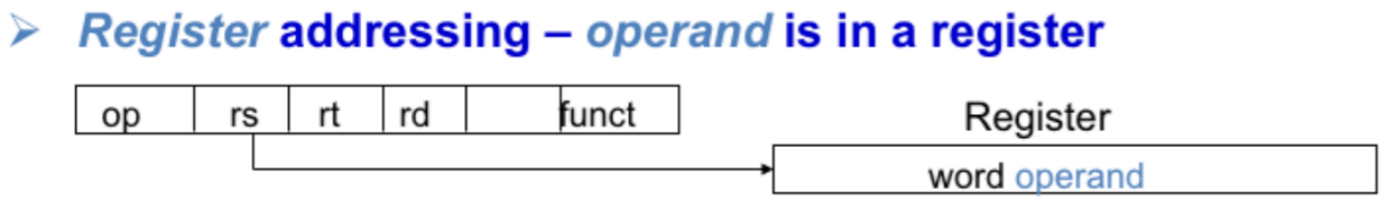
能不能设计CPU?

MIPS指令集的设计过程：首先设计指令格式，分为R型I型J型指令，再根据每类指令设计它们的数据通路，选择流水线方案，每级流水线要加流水线寄存器，然后把所有指令的数据通路通过多选器合并在一起，通过硬布线的方式连接好各个部件，包括PC、指令存储器、寄存器堆、ALU、数据存储器等，最后设计控制信号，驱动cpu的运转。

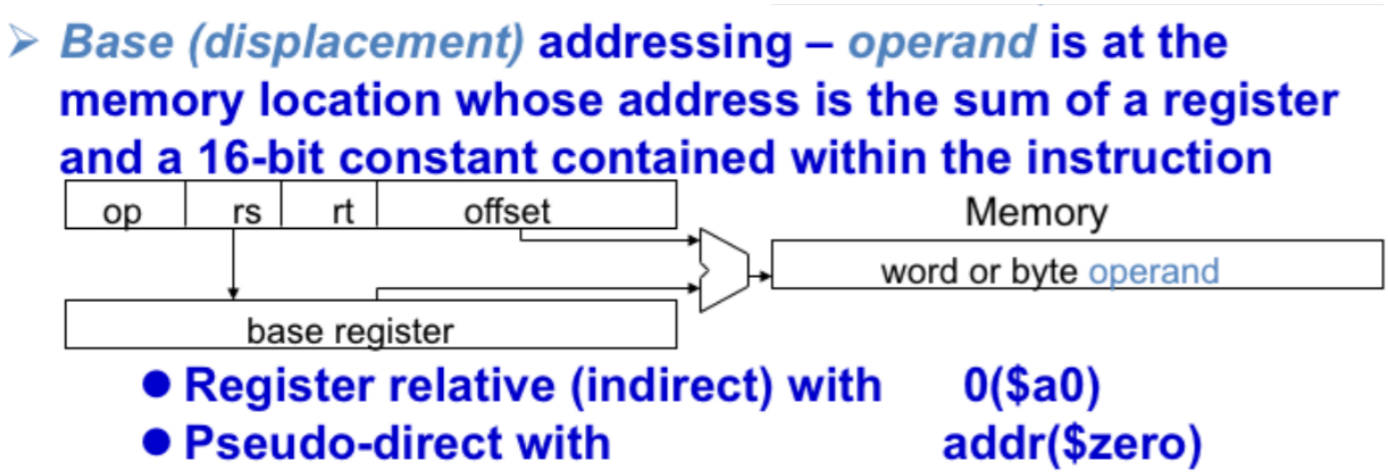
Name	Fields						Comments
Field size	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits	All MIPS instructions are 32 bits long
R-format	op	rs	rt	rd	shamt	funct	Arithmetic instruction format
I-format	op	rs	rt	address/immediate			Transfer, branch, imm. format
J-format	op	target address					Jump instruction format

MIPS操作数寻址方式

1. 寄存器直接寻址：常用于R型指令

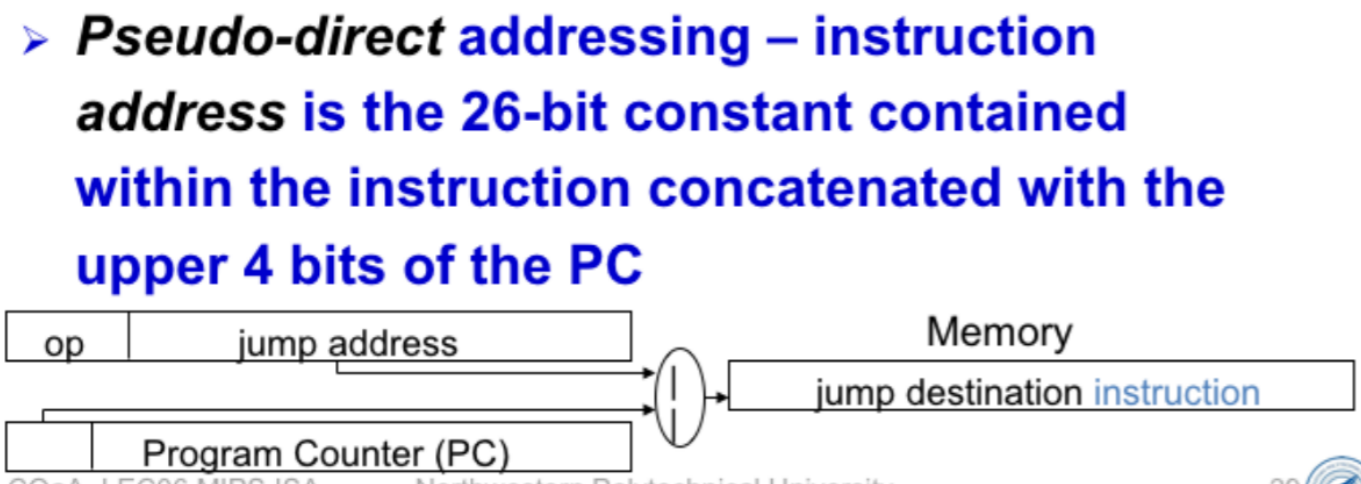


2. 基址寻址：rs寄存器中的地址加上偏移地址得到内存中的地址



3. 立即数寻址:操作数就是I型指令的立即数字段

4. PC相对寻址
5. 伪直接寻址:J型指令中的跳转地址经过符号扩展后，与PC的高4位组成的地址



MIPS数据通路

复制下黑书的图

CPU如何工作?

CPU在一个指令周期内取出并执行一条指令，不断重复。指令周期分为：

1. 取指周期：根据PC中的地址，从指令存储器中取出指令，存放在指令寄存器IR中
2. 间指周期：根据指令，从数据存储器中取出操作数的有效地址，存放在MDR中
3. 执行周期：根据操作码，将操作数送入算术逻辑单元ALU产生结果
4. 中断周期：处理中断请求

所有指令采用同样的指令周期称为单指令周期，采用不同的指令周期称为多周期指令，采用流水线方案并行执行称为流水线

流水线冒险/冲突/相关

类型	定义	解决方法
结构冒险	多条指令在同一时刻争用同一资源	1. 阻塞后续指令若干个时钟周期 2. 分离存储器为数据存储器和指令存储器
数据冒险	前一条指令必须执行完才能执行后一条指令，如“写后读”指令	1. 阻塞后续指令 2. 编译优化，调整指令的顺序 3. 旁路（转发）
控制冒险	遇到改变PC值的指令如转移指令	1. 进行分支预测，来提前生成转移目标地址。静态预测总是预测不跳转，动态预测根据历史情况预测。

流水线加速比

不用流水线花费的时间/用流水线花费的时间

中断过程

中断定义

- 指CPU对突发的外部事件的反应过程或机制。
- CPU收到外部信号（中断信号）后，停止当前工作，转去处理该外部事件，处理完毕后回到原来工作的中断处（断点）继续原来的工作。

CPU内部引起的中断称为内中断

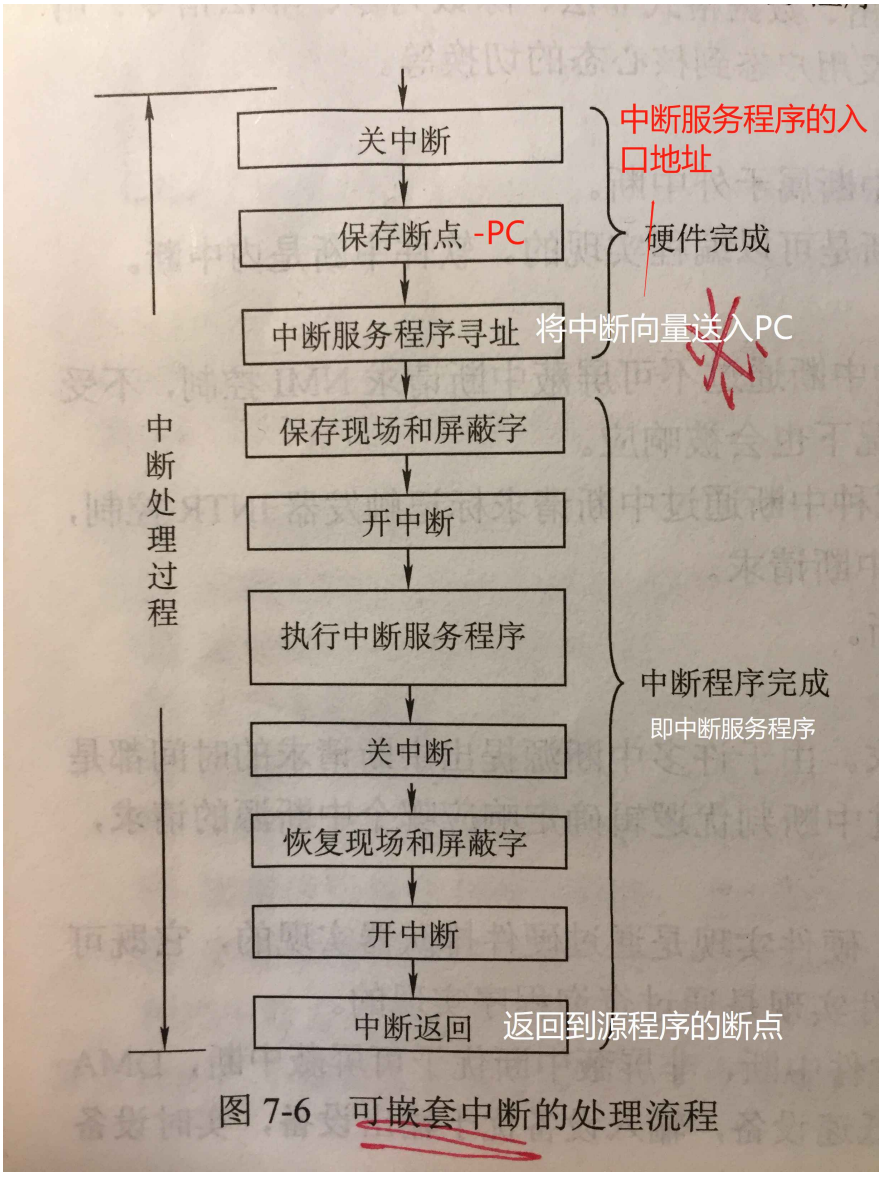
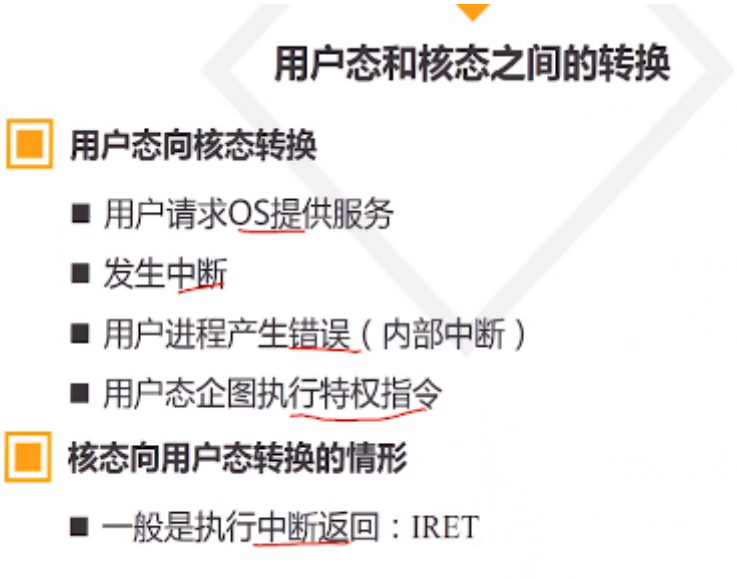


图 7-6　可嵌套中断的处理流程

用户态与核心态的转换



中断嵌套?

是指中断系统正在执行一个中断服务时，有另一个优先级更高的中断提出中断请求，这时会暂时终止当前正在执行的级别较低的中断源的服务程序，去处理级别更高的中断源，待处理完毕，再返回到被中断了的中断服务程序继续执行，这个过程就是中断嵌套。

调度?

分类：

- 高级调度：作业调度。决定哪些作业从后备状态进入运行状态，并给他们分配资源
- 中级调度：内存调度，通过挂起和解除挂起决定哪些进程参与CPU的竞争
- 低级调度：进程调度，按照某些原则将处理机分配给就绪队列中的某进程

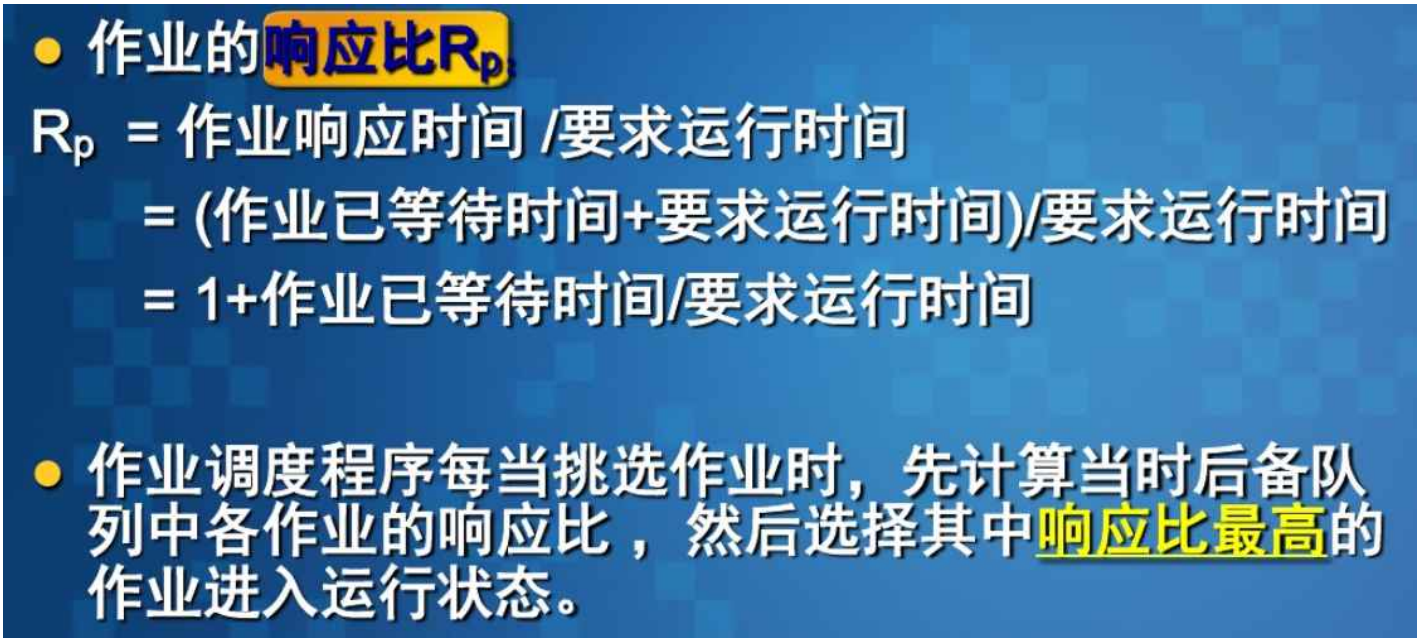
用到调度的地方有：作业调度、处理机调度、内存调度、磁盘调度等。

1. 进程的调度算法有：
    - 先来先服务FCFS
    - 时间片轮转法RR
    - 短进程优先
    - 优先级调度算法：为每个进程静态或动态地设定优先级
    - 多级反馈队列调度算法：多级反馈队列调度算法既能使高优先级的作业得到响应又能使短作业（进程）迅速完成。 队列遵循的是先来先服务算法，每一进程分配一定的时间片，若时间片运行完时进程未结束，则进入下一优先级队列的末尾

—多个就绪队列，每个队列优先级不同，优先级越低则时间片越长；
—新进程进入内存后，先投入高优先级队列末尾按RR算法调度；
—若一个时间片未能执行完，则降低优先级，如此下去，直到降低到最后的队列；
—若时间片未用完发生阻塞，返回至高一级/最高优先级就绪队列；

—仅当较高优先级的队列为空，才调度较低优先级的队列。

2. 作业的调度算法有：
   - FCFS
   - 短作业优先法
   - 高响应比优先法：响应比=响应时间/运行时间



   - 优先级调度算法
   - 时间片轮转法
3. 虚拟内存页面调度(替换)算法有：

解释：进程运行时，若其访问的页面不在内存而需将其调入，但内存已无空闲空间时，就需要从内存中调出一页程序或数据，送入磁盘的对换区。

- 最佳置换算法OPT：未来最长时间内不会被访问的
- 先进先出算法：FIFO
- LRU：最近最久未使用
- CLOCK=NRU最近未用：给每个页面设置一个使用位和修改位，替换页面时像时钟一样循环扫描，查找使用位为0的页面将其替换掉。遇到使用位为1的页，将其使用位变为0

4. 磁盘调度算法有：
   - FCFS
   - 最短寻道时间优先SSTF
   - 电梯算法（扫描算法）SCAN
   - **循环扫描调度算法CSCAN 总是从0号柱面开始向里扫描**

静态重定位/动态重定位?

源程序经过编译、链接后生成可执行文件（目标程序、模块）。程序执行时，由装入程序将模块装入到内存中。装入时，由于各程序其实地址都是以0开始，为了根据内存的情况装入到内存的适当位置，需要将模块中的指令和数据中的相对地址修改为绝对地址，称为重定位。

- 静态重定位：装入时即将相对地址转为绝对地址，这样程序装入后不能再移动
- 动态重定位：装入时不进行重定位，因为程序在执行前可能在内存中移动位置。等到程序真正要执行时再进行重定位。需要硬件的支持

覆盖与交换

—覆盖技术：一个作业的若干程序段，或几个作业的某些部分共享某一个存储空间。
—交换技术：系统将内存中某些进程暂时移到外存，把外存中某些进程换进内存，占据前者所占用的区域。

什么是内存连续分配?

为一个程序分配连续的物理内存空间，根据对物理内存的划分不同，分为以下几种：

- 单一连续分配：内存中仅有一道程序，放在用户区内存
- 固定分区分配
- 动态分区分配

什么是内存非连续分配?

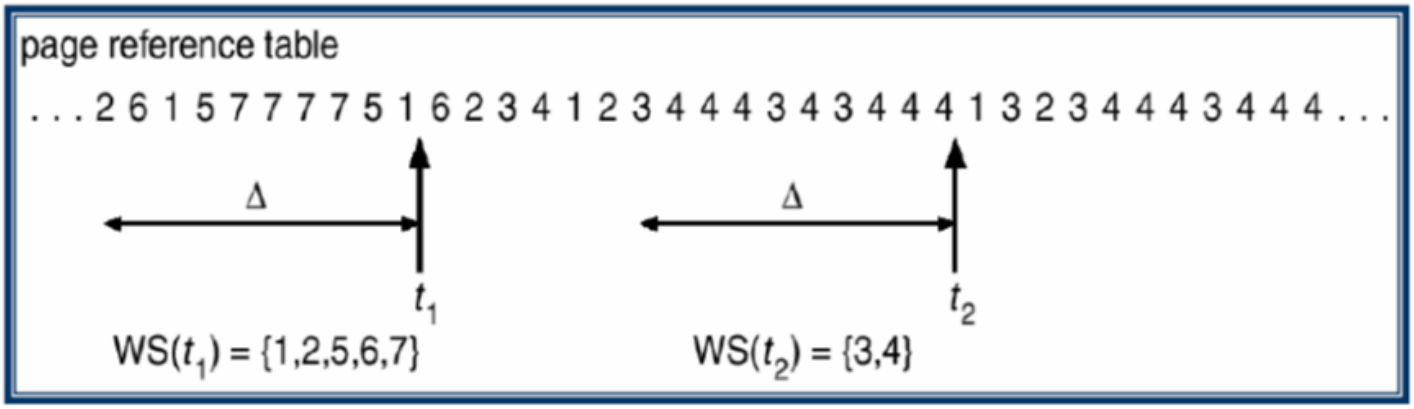
允许一个程序分散地装入到不相邻的内存分区中。有基本分页、基本分段、基本段页式存储管理

虚拟存储?

用户作业必须一次性装入内存、装入后就算进程阻塞也全部驻留在内存中，极大地浪费了资源。根据局部性原理，先将程序的一部分装入内存，执行程序时再由os将其他所需部分调入内存。在逻辑上扩充了内存的容量。有请求分页、请求分段、请求段页式存储管理
虚拟存储器将主存或辅存的地址空间统一编址，形成一个庞大的地址空间，在这个空间中用户可以自由编程，而不必在乎实际的主存容量和程序在主存中实际的存放位置

驻留集

v**驻留集，虚拟页式管理中OS给进程分配的物理页面数目**
§每个进程的**驻留集越小**，则同时驻留内存的进程就越多，可以**提高并行度**和处理器利用率；另一方面，进程的**缺页率上升**，使调页的开销增大。
v**工作集是一个进程在虚拟时间t时在过去的D个虚拟时间单位中被访问到的页的集合，可用一个二元函数W(t, D)表示。**



存储体系

为了使得计算机在存储器的容量，速度，价格之间有一个折中。比如CPU-cache-主存体系，是为了解决主存与CPU速度不匹配。而主存-页表-辅存体系是为了解决主存容量问题。

总线周期

申请分配阶段-寻址-传输-结束（让出总线控制权）

IO方式

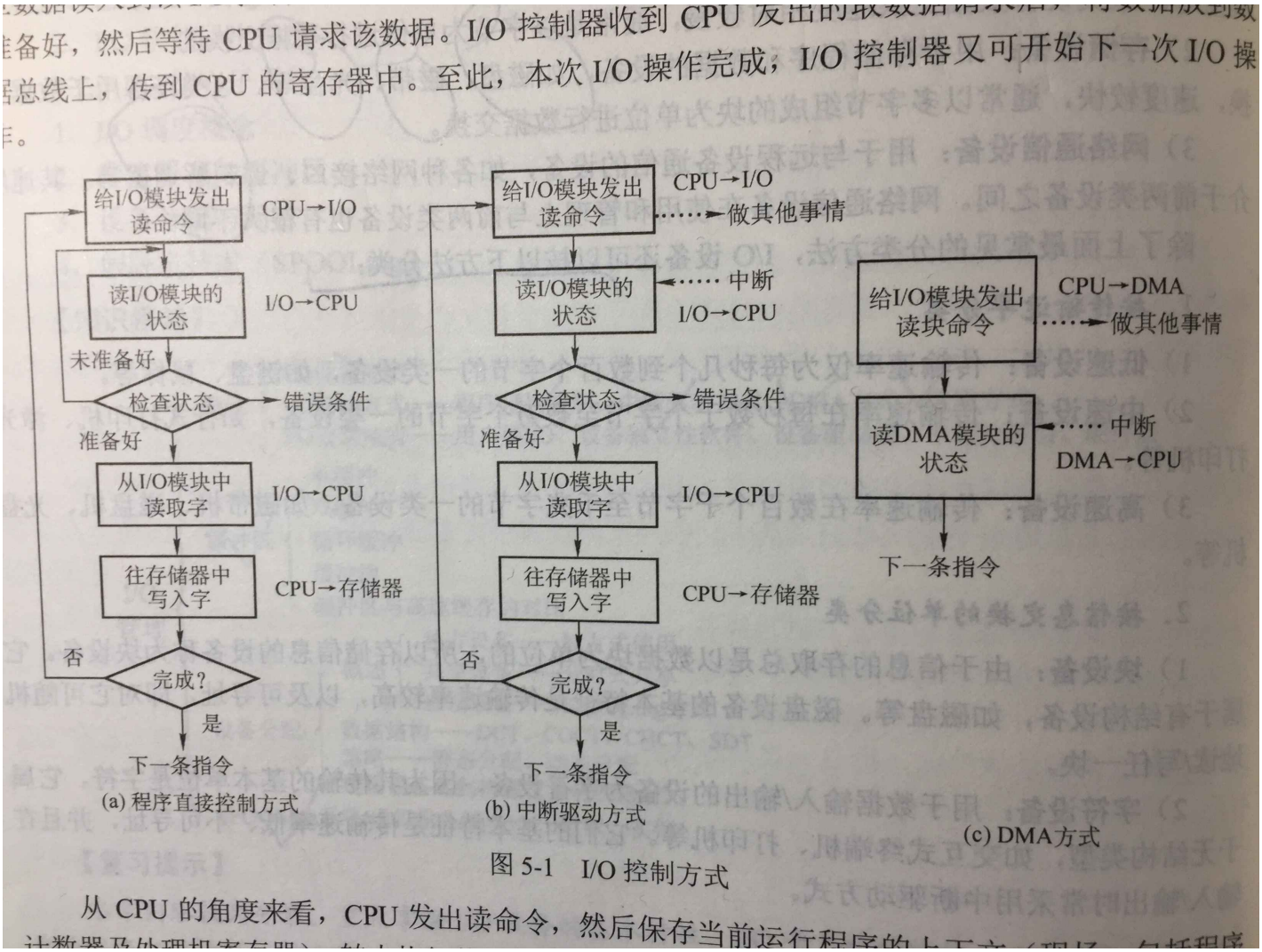


图 5-1 I/O 控制方式

从 CPU 的角度来看，CPU 发出读命令，然后保存当前运行程序的上下文（现场，包括程序

什么是DMA?

非DMA的数据传送方式是外设将数据通过中断的方式传送给cpu，cpu再传送给内存，cpu与外设并行工作，每次数据传输都要保存、恢复cpu现场，开销大。而DMA开辟了一条外设和内存的直接数据通路，传送消息不经过cpu，使得cpu与外设并行工作，开销小。适用于磁盘这种设备的高速大批量数据传送。

DMA的传送方式有：（CPU此时也想访存咋办）

1. 停止CPU访存
2. DMA和CPU交替访存：将一个CPU周期分为专供DMA访存的周期和专供CPU访存的周期
3. 周期窃取：CPU暂时放弃总线占有权，由IO设备挪用一个或几个存取周期

DMA和中断的区别

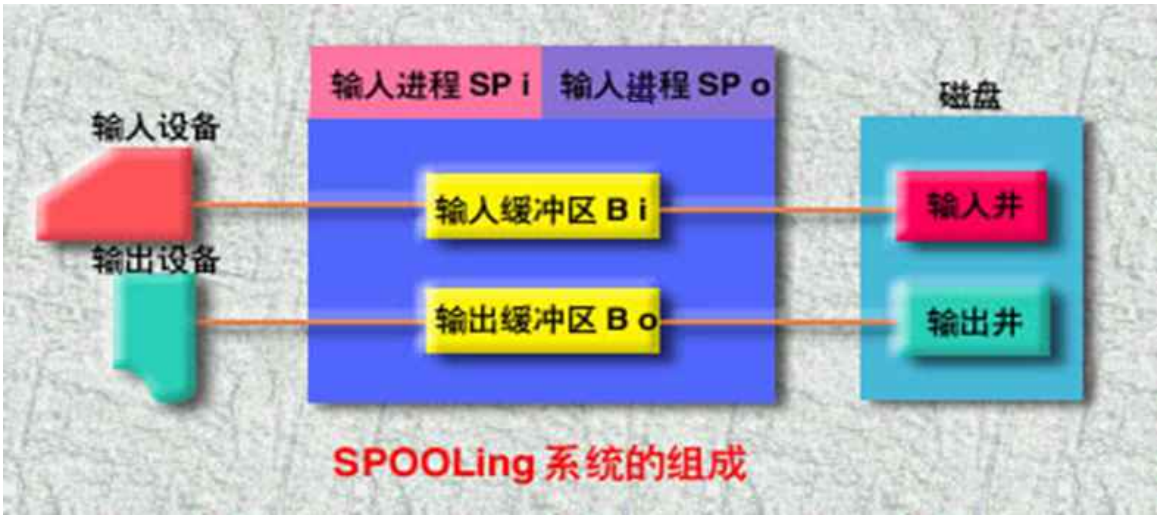
1. 中断出现了程序切换，且需要CPU传送数据，速度慢；DMA除了预处理和后处理，不需要CPU的干预，非常适合传送大批量高速外设数据
2. 对中断请求的响应只能发生在指令的中断周期；而对DMA的响应可以发生在每个机器周期结束时（周期窃取）

通道

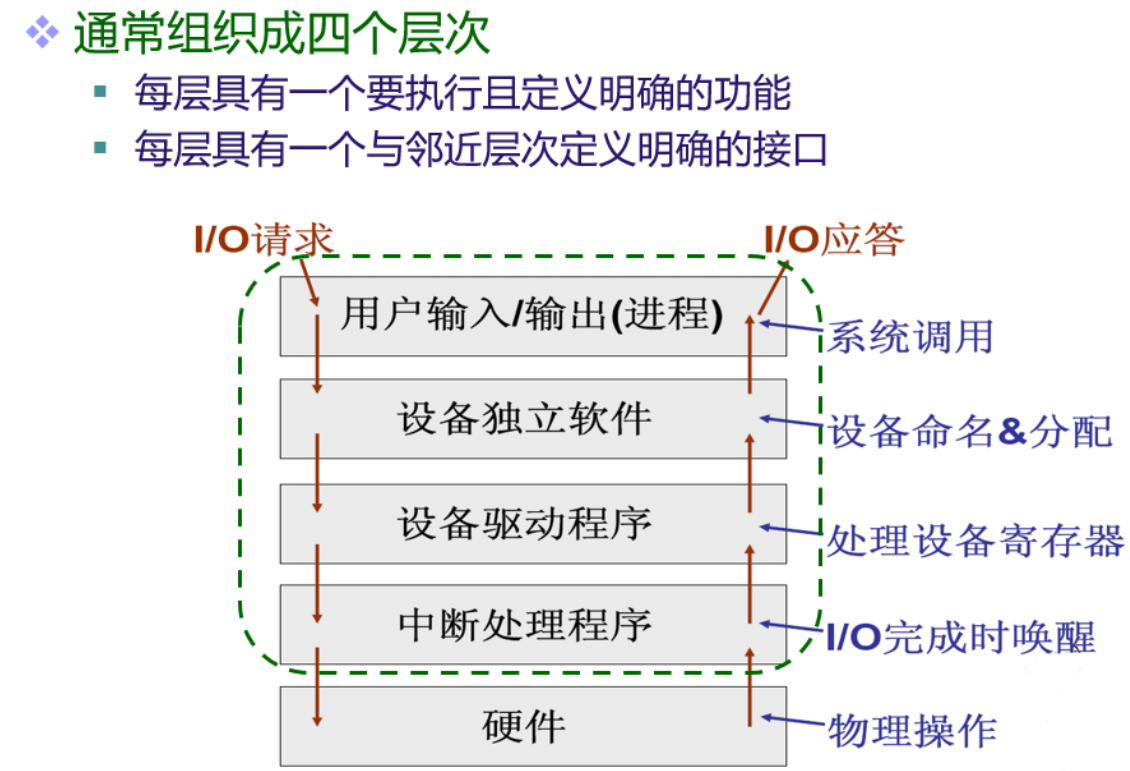
本质是一个外围处理机，有自己的存储器。用于控制I/O设备与内存间的数据传输。启动后可独立于CPU运行，实现CPU与I/O的并行

SPooling 同时外围设备联机操作--假脱机技术

§把独享设备转变成具有共享特征的虚拟设备



IO层次



总线

总线是一组能为多个部件分时共享的公共信息传送线路，在某个时刻只允许有一个部件向总线发送信息，但多个部件可以同时从总线上接收相同的信息。
按功能划分为：片内总线、系统总线（数据总线、地址总线、控制总线）、通信总线
总线带宽是指单位时间内总线上可传输数据的位数，=总线频率*总线宽度。

什么是主设备

获得总线控制权的设备即为主设备，被主设备寻址的设备称为从设备，外设要往内存输入数据时是主设备，内存要往外设输出数据是从设备。
DMA控制器向CPU申请总线控制权后，等待CPU将总线交给他时，是从设备。而当他获得总线控制权后，就是主设备了。

设备驱动程序

- **什么是设备驱动程序？设备驱动的主要功能是什么？**
  - 设备驱动程序是I/O进程与设备控制器之间的通信程序。
  - 功能
    - 将抽象要求转换成具体要求
    - 检查用户I/O合法性
    - 了解I/O设备状态，传递参数，设置工作方式
    - 及时响应中断，并调用程序进行处理
    - 发出I/O，启动设备，完成I/O

CISC和RISC的区别

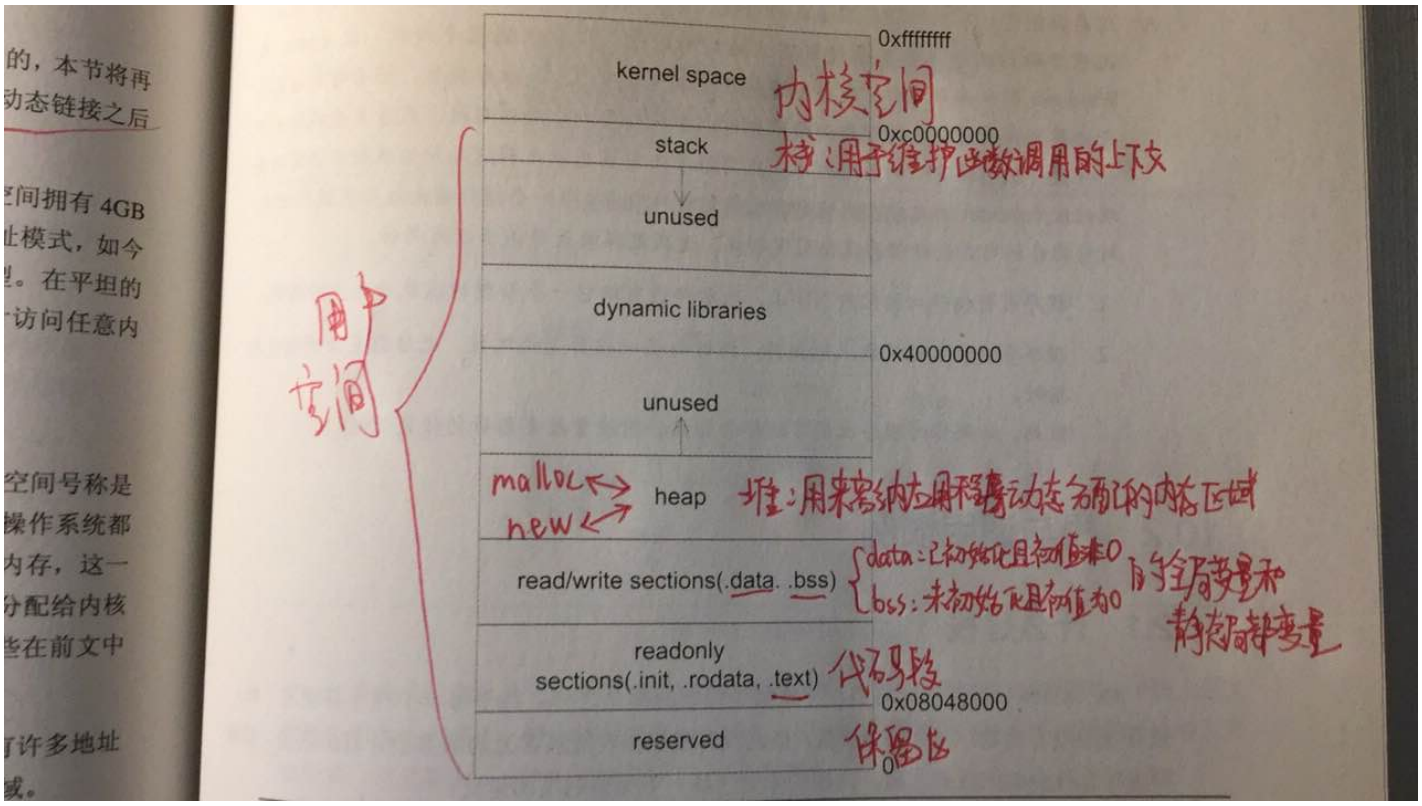
CISC的英文全称为“Complex Instruction Set Computer”，即“复杂指令系统计算机”

指令系统类型	指令	寻址方式	实现方式	其它
CISC（复杂）	数量多，使用频率差别大，可变长格式	支持多种	微程序控制技术（微码）	研制周期长
RISC（精简）	数量少，使用频率接近，定长格式，大部分为单周期指令，操作寄存器，只有Load/Store操作内存	支持方式少	增加了通用寄存器；硬布线逻辑控制为主；适合采用流水线	优化编译，有效支持高级语言

微程序控制器使用了存储程序的思想，每条机器指令对应一个小程序，灵活性好，由于每条指令执行都需要访问控制存储器，所以速度较慢

硬布线控制器使用专门的逻辑电路实现，所以难修改扩展，灵活性差，速度快

内存映像？动态存储和静态存储?堆栈?



- 动态存储变量是在程序执行过程中，使用它时才分配存储单元， 使用完毕立即释放。 典型的例子是**函数的形式参数**，在函数定义时并不给形参分配存储单元，**只是在函数被调用时，才予以分配**，调用函数完毕立即释放。动态存储区包含堆栈。
- 静态存储方式是指在程序编译期间分配固定的存储空间的方式。该存储方式通常是在变量定义时就分定存储单元并一直保持不变， 直至整个程序结束。const常量，**全局变量，static静态变量**等就属于此类存储方式。静态存储区有data段和bss段

Stack的空间由操作系统自动分配/释放，Heap上的空间手动分配/释放。Stack空间有限，Heap是很大的自由存储区。C中的malloc函数分配的内存空间即在堆上

一个文件在磁盘上，如何访问到?

结合文件的物理结构

在用户界面上双击文件时，首先资源管理器程序会给OS发送一个read系统调用，操作系统开始驱动硬盘：

1. 根据指令找到文件的位置（盘块、磁道、扇区）。
2. 寻道算法将磁头移动到指定位置，
3. 等磁头定位到该磁道的扇区，
4. 进行数据传输。

   后三部分时间称为寻道、延迟、传输时间。

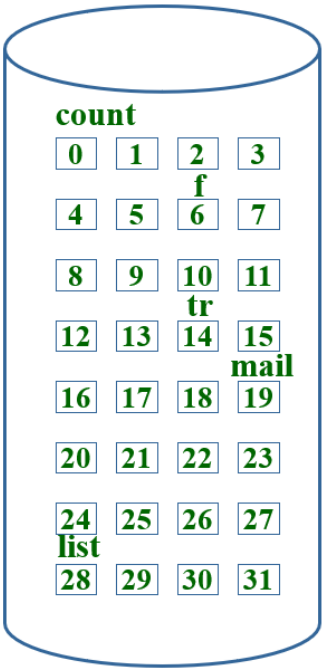
文件

1. 文件的逻辑结构：指的是用户的视角下，文件内部的数据是怎么组织起来的。对于有结构文件来说，内部的记录的组织形式有：（1）顺序文件，可以再细分为顺序存储和链式存储。（2）索引文件。（3）索引顺序文件。（4）散列文件。

2. 文件的目录结构：指的是用户视角下，文件与文件之间的逻辑结构。一个目录项包括文件名和索引节点编号。由文件名找到索引节点，也就是FCB，里面存储了包括文件物理地址等各种信息，可以由索引节点找到文件物理地址。

3. 文件共享的两种方式：
（1）基于索引节点的共享/硬链接：原本文件的物理地址及属性信息，不再是放在目录项中，而是放在索引结点中。所以文件目录每项中只剩文件名和指向索引结点的指针，多个共享用户的目录都记录有该文件的目录项。王道P213
（2）符号链形式/软链接：文件主之外的人想要访问文件，会先访问到一个LINK文件，里面是被共享文件的路径。

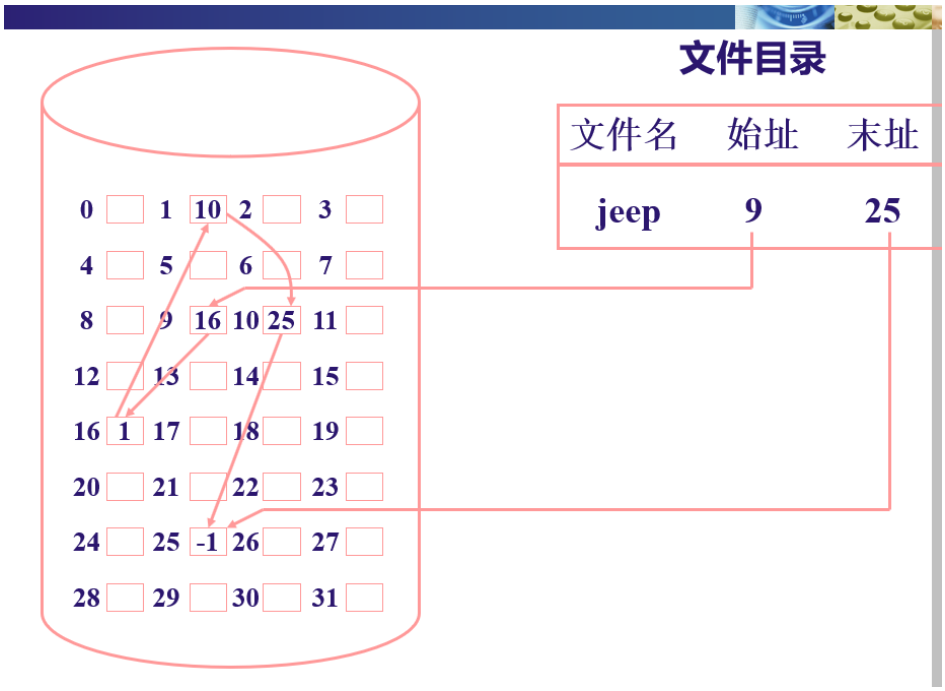
4. 文件的物理结构：指的是文件存放在外存中时，外存给文件分配空间的方式。
（1）连续分配方式：逻辑上连续的块在物理上也连续，每个文件在磁盘上占用一列连续的块。表里面有文件的起始物理块号和长度，用起始块号+逻辑块号（就是文件内部块的编号）=要访问的块的物理块号。连续分配支随机（直接）访问和顺序访问。



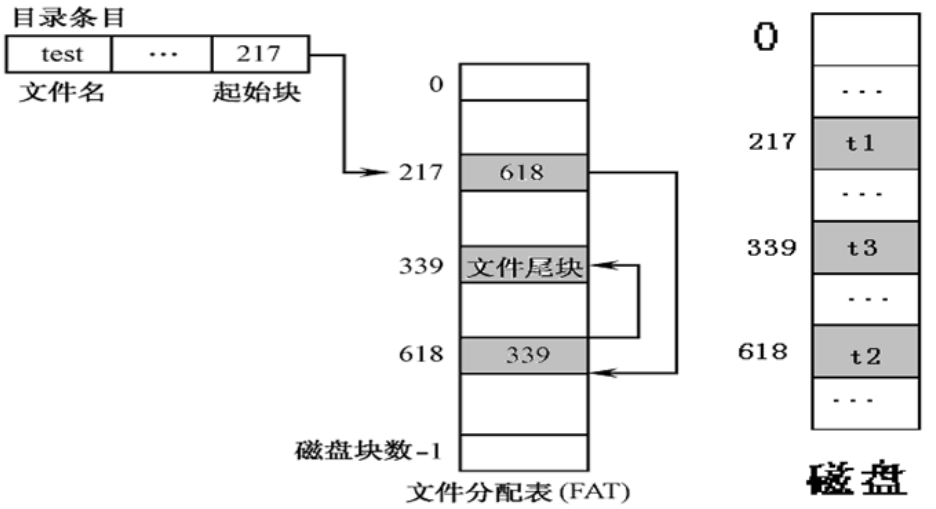
文件目录

文件名	始址	块数
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

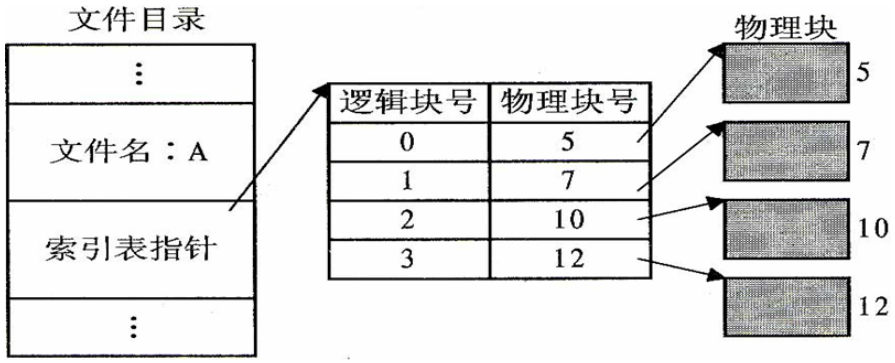
（2）隐式链接分配方式：类似于链表。每个物理块中都有一个指向下一块的指针。只适合顺序存取



（3）显式链接分配方式：把指针显式地存放在一张表中，叫做FAT，每个磁盘会有一张FAT，记录了这个磁盘上面的各个块之间的连接顺序。



（4）索引分配方式：为每个文件建立索引表，类似于页表，记录了逻辑块到物理块的映射。

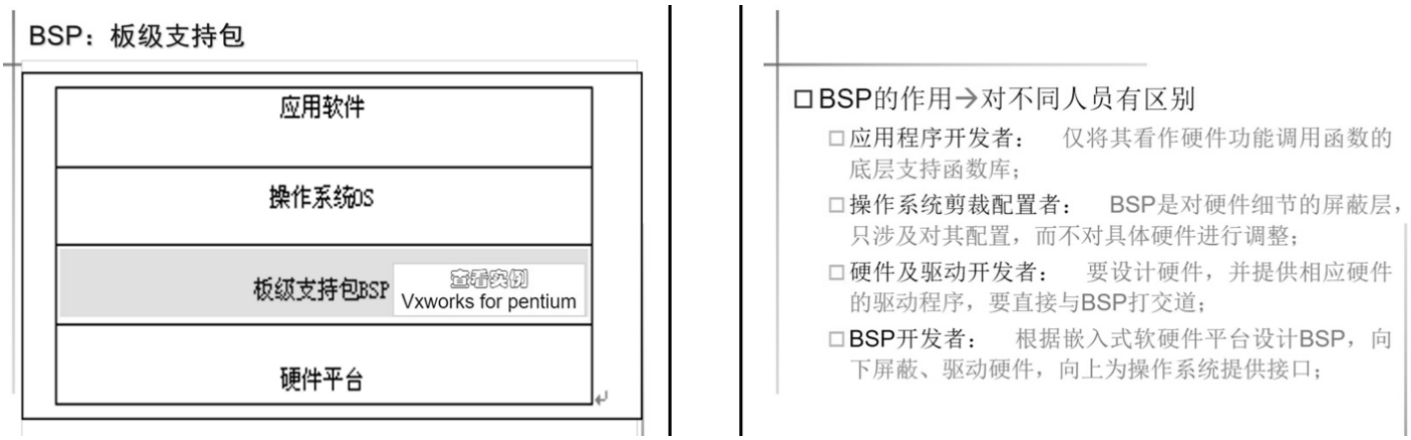


5.　文件存储空间管理：指的是对磁盘的空闲块的管理。（1）空闲表法：每个表项记录的是空闲区的序号，起始空闲块号和空闲块数。（2）空闲链表法：将所有空闲的盘块或盘区拉成一条链。（3）位示图法：每个盘块由一位二进制位01表示空闲与否。（4）成组链接法：空闲表和空闲链表的结合。

数字电路根据逻辑功能的不同特点，可以分成两大类，一类叫组合逻辑电路（简称组合电路），另一类叫做时序逻辑电路（简称时序电路）。组合逻辑电路在逻辑功能上的特点是任意时刻的输出仅仅取决于该时刻的输入，与电路原来的状态无关。而时序逻辑电路在逻辑功能上的特点是任意时刻的输出不仅取决于当时的输入信号，而且还取决于电路原来的状态，或者说，还与以前的输入有关。

嵌入式系统学了什么?

- 嵌入式硬件:包括嵌入式处理器、存储介质、电路、总线控制、IO接口等
- 嵌入式软件:
  - ROM monitor监控程序，具有管理系统资源以及与用户进行交互的功能
  - Boot loader（是在操作系统内核运行之前运行的一段小程序。通过这段小程序，我们可以初始化硬件设备、建立内存空间的映射图，从而将系统的软硬件环境带到一个合适的状态，以便为最终调用操作系统内核准备好正确的环境。）
  - BSP版级支持包



- 嵌入操作系统（约等于实时操作系统，以应用为中心）：内核、任务调度、实时性

—实时系统：能够在指定或者确定的时间内完成系统功能和对外部或内部、同步或异步时间做出响应的系统。

—在实时计算中，系统的正确性不仅仅依赖于计算的逻辑结果，而且依赖于结果产生的时间。

—硬实时系统：必须在规定的时刻或时间范围完成任务。

—软实时系统：接受偶尔违反最终时限的情况。

- 嵌入式软件开发

FPGA（Field Programmable Gate Array）

现场可编程逻辑门阵列

复习打卡

- [x] ~~我复习并且记忆了一遍~~
- [x] ~~我复习并且记忆了一遍~~
- [x] ~~我复习并且记忆了一遍~~
- [x] ~~我复习并且记忆了一遍~~
- [x] ~~我复习并且记忆了一遍~~
- [x] ~~我复习并且记忆了一遍~~
- [x] ~~我复习并且记忆了一遍~~
- [x] ~~我复习并且记忆了一遍~~
- [x] ~~我复习并且记忆了一遍~~
- [ ] 我复习并且记忆了一遍
- [ ] 我复习并且记忆了一遍