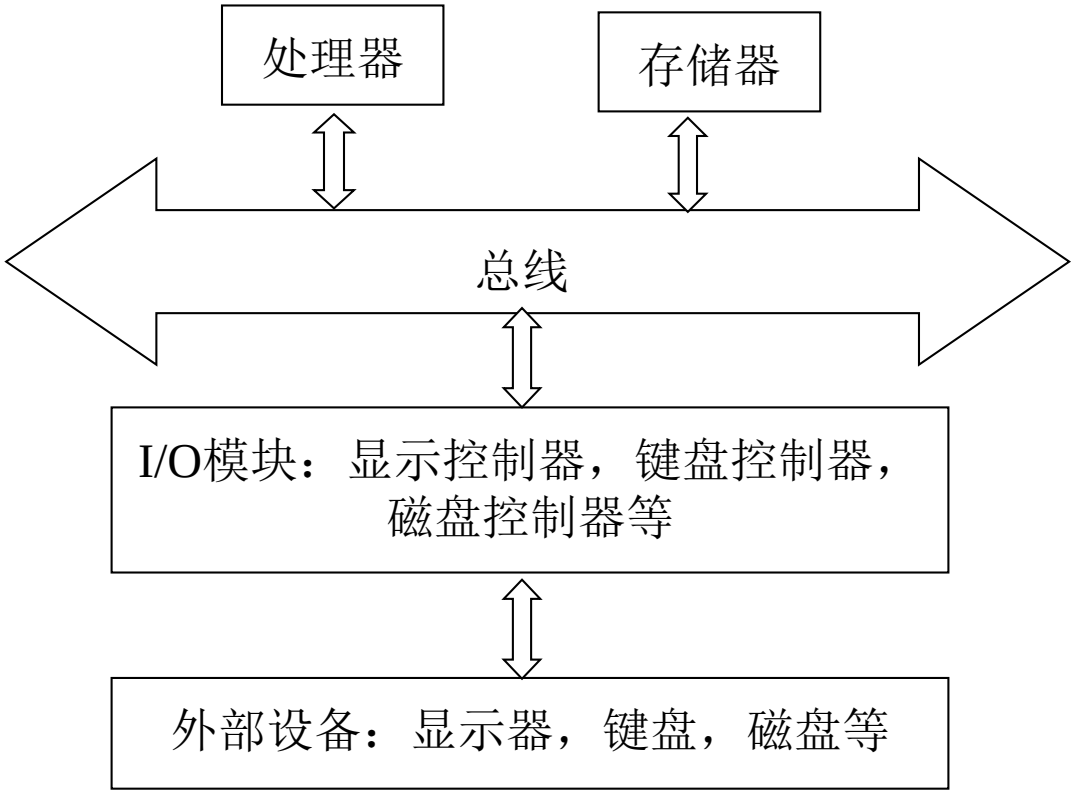
操作系统复习

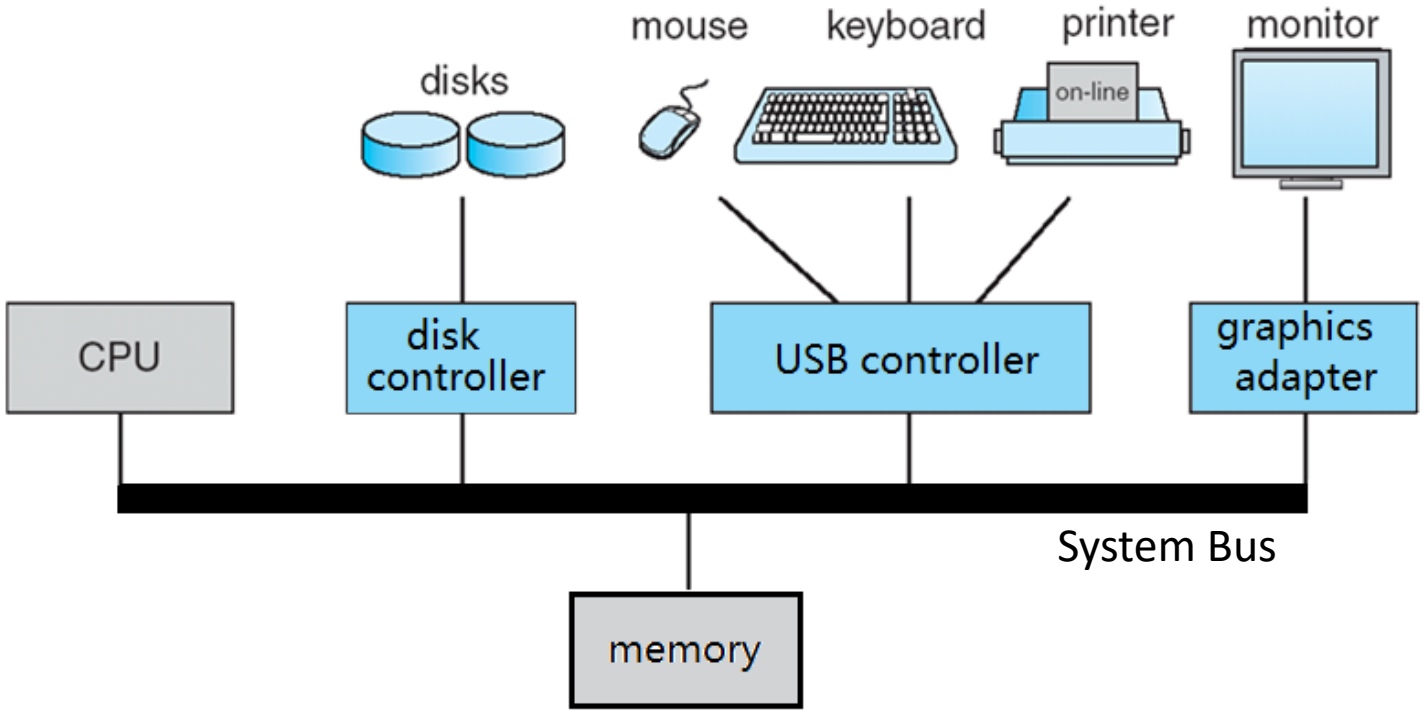
操作系统概述

- 计算机硬件结构



操作系统概述

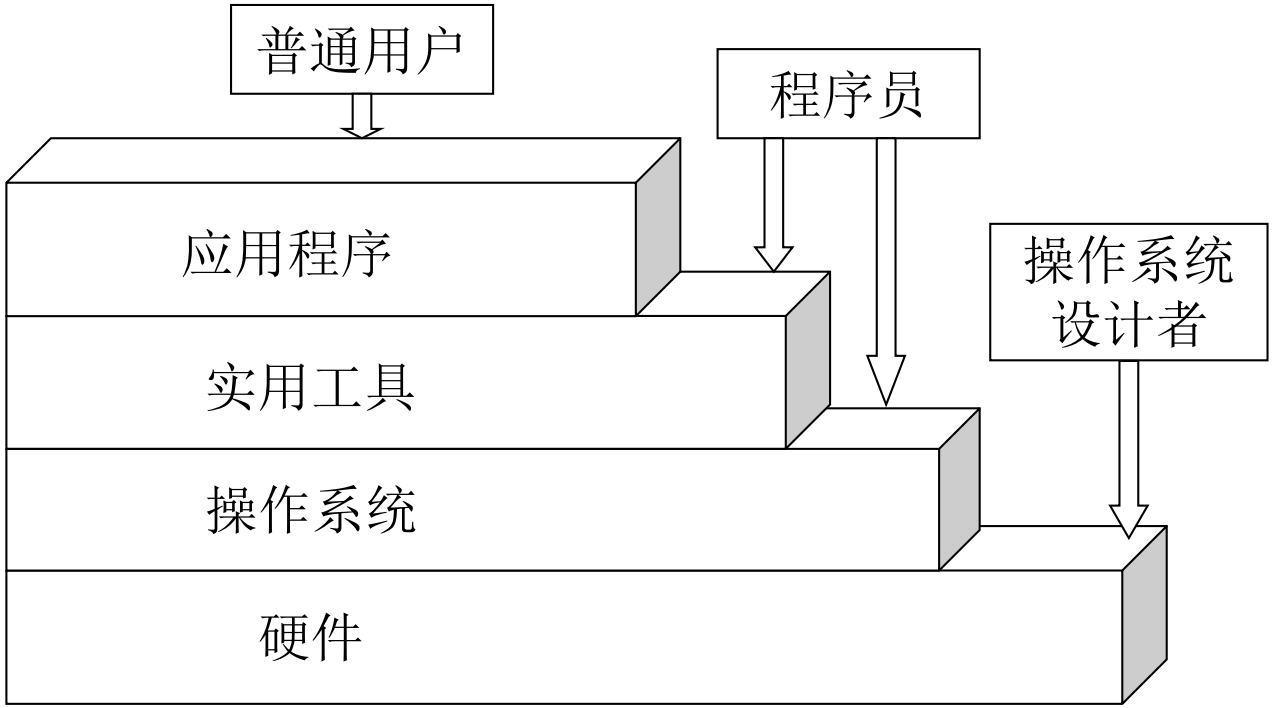
- I/O结构



- 处理器、主存储器、I/O模块之间通过总线交换信息
- 控制总线：负责传送对磁盘及其它外部设备的控制信号
- 数据总线：负责计算机系统各部件间的数据传送
- 地址总线：负责传送存储器单元的地址和I/O设备的地址

操作系统概述

- 操作系统的地位



- 操作系统是计算机系统中的基础系统软件
- 操作系统是计算机硬件上的第一层软件
- 其它软件通过操作系统使用计算机系统硬件

操作系统概述

- 理解操作系统
  - 高效地组织软件运行、高效地管理和使用计算机系统硬件
  - 并发性：“并发”是指同时存在多个平行的活动
  - 共享性：共享性是指操作系统程序与多个用户程序共用系统中的各种资源
  - 随机性：操作系统是在一个随机的环境中运行的，它必须对发生的不可预测事件进行响应

操作系统概述

- 操作系统的功能
  - 处理机管理：把CPU的时间合理地分配给各个执行的程序，使其得到充分有效的利用
  - 存储管理：管理计算机的内存，保证程序运行互不冲突，互不侵扰
  - 设备管理：对系统中所有I/O设备进行管理
  - 文件管理：有效地管理外存储器空间存储及文件
  - 进程调度：操作系统负责进程的调度管理
  - 用户接口：为用户使用计算机系统提供了良好的工作环境和用户界面

操作系统概述

- 操作系统结构
  - 整体式结构
  - 层次式结构
  - 微内核结构
  - 模块化结构
- 典型的操作系统

操作系统概述

- 用户接口
  - 命令接口
  - 系统调用
  - API

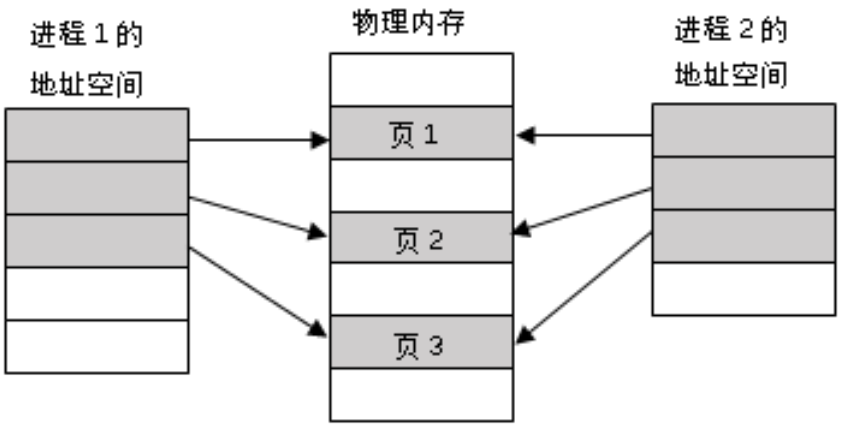
系统调用与API的区别与联系

处理机管理

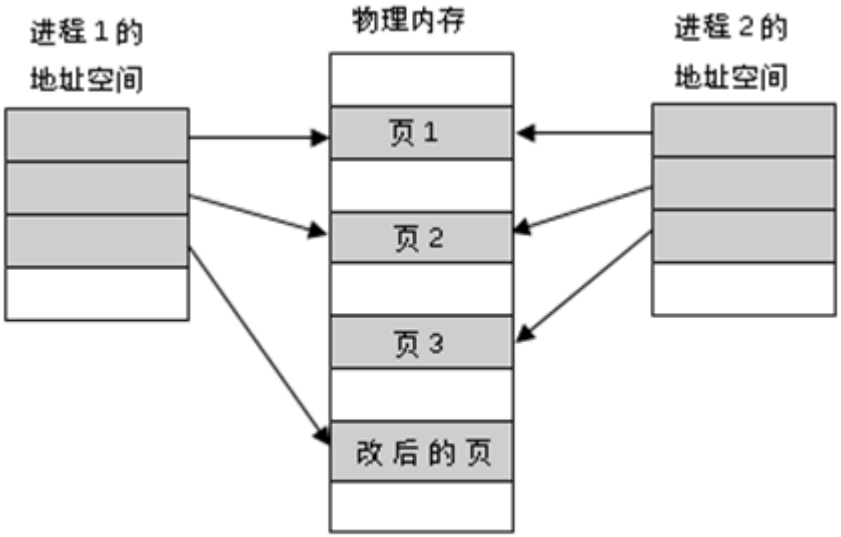
- 功能
  - 进程控制和管理：对系统中所有进程从创建、执行到撤销的全过程实行有效的管理和控制
  - 进程同步和互斥管理：提供软件或硬件的机制，来确保多个进程按照预想的步调顺利推进
  - 进程通信管理：协调不同的进程，使之能在一个操作系统里同时运行，并相互传递、交换信息
  - 处理器调度：以满足系统目标（如响应时间、吞吐量、效率）的方法，把进程分配到一个或多个处理器中执行
  - 线程控制和管理：对线程进行必要的控制与管理

处理机管理

- 进程与线程
  - 程序
  - 进程
  - 线程
  - 进程与程序的区别与联系
  - 进程与线程的区别与联系：调度、并发性、系统开销、效率、拥有资源
  - 写时拷贝/写时复制



a) 进程 1 发生页面修改之前



(b) 进程1写数据之后

处理机管理

- 进程的执行与状态
  - 并发执行
    - 间断性
    - 失去封闭性
    - 不可再现性
  - 5个基本状态
- 互斥与同步
  - 进程间制约
  - 临界资源
  - 临界区
  - 实现
    - 硬件方法
    - 信号量方法
    - 管程方法

处理机管理

- 信号量方法
  - 结构（记录）型信号量

```
struct semaphore{
    int value;
    struct QUEUETYPE  *queue;
}
```

  - 信号量值
    - 当 x.value>0，表示当前有 x.value 个资源可以使用
    - 当 x.value=0，表示当前没有个资源可以使用
    - 当 x.value<0，表示当前有 |x.value| 个进程等待使用资源

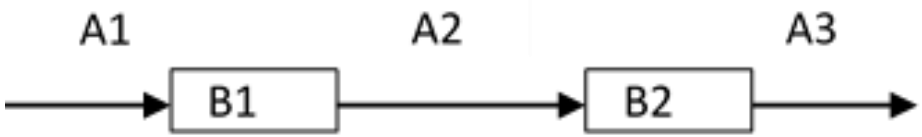
信号量值的估计：最大值，最小值

m个资源，n个进程，m < n：x.value∈[n-m, m]

处理机管理

- 信号量操作
  - Wait()，P操作
  - signal()，V操作
- 应用
  - 会使用信号量机制来解决给定问题
  - 生产者/消费者问题

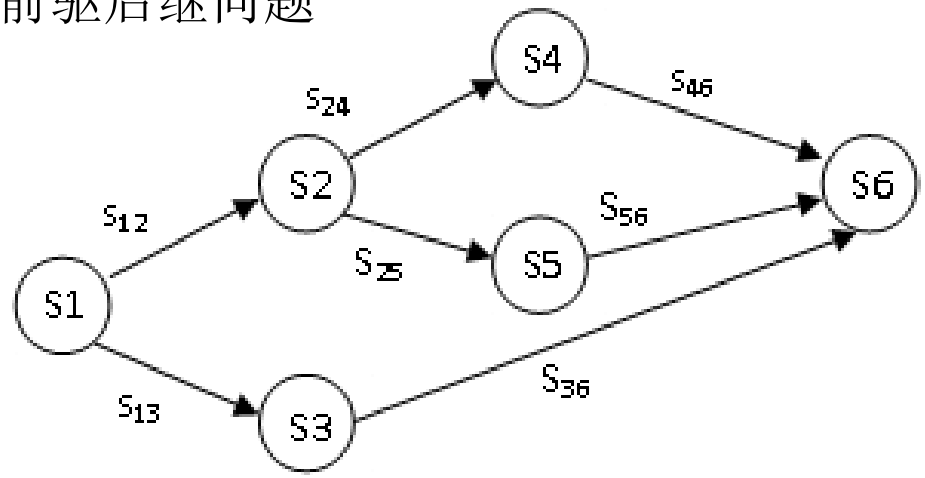
(1) 同步关系：顺序
(2) 互斥关系：mutex



读者/写者问题；哲学家问题

n m

- 进程的前驱后继问题



处理机管理

- 进程的前驱后继问题

  semaphore $s_{12},s_{13},s_{24},s_{25},s_{36},s_{46},s_{56}$;

  $s_{12}=s_{13}=s_{24}=s_{25}=s_{36}=s_{46}=s_{56}=0$;

  void S1() { S1 execute; signal(s_{12});signal(s_{13}) }

  void S2() { wait(s_{12});S2 execute;signal(s_{24});signal(s_{25}) }

  void S3() { wait(s_{13});S3 execute;signal(s_{36}) }

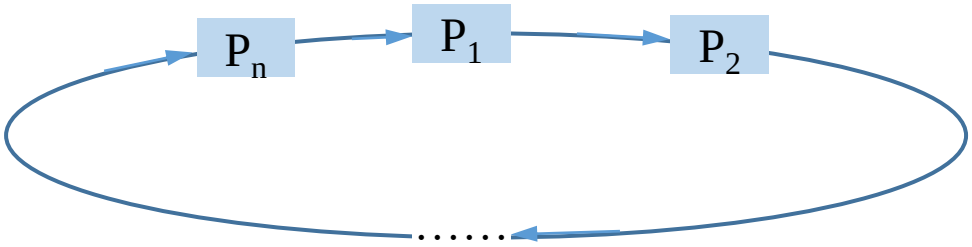
  void S4() {wait(s_{24});S4 execute;signal(s_{46}) }

  void S5() { wait(s_{25});S5 execute;signal(s_{56}) }

  void S6() {wait(s_{36});wait(s_{46});signal(s_{56});S6 execute }

处理机管理

- 死锁问题
  - 死锁与饥饿
  - 产生死锁的必要条件
    - 互斥条件：每个资源每次只能分配给一个进程使用
    - 部分分配（占有且等待）条件：进程得到一部分资源后，还允许继续申请新的资源
    - 不可抢占（非剥夺）条件：任一个进程不能从另一个进程那里抢占资源，只能由占用进程自己释放
    - 循环等待条件：存在一个循环的等待序列 P_1，P_2，P_3，…，P_n，从而形成一个进程循环等待环



处理机管理

- 解决死锁问题
  - 死锁预防
  - 死锁避免
  - 检测解除
- 银行家算法
  - 安全序列
  - 安全状态
  - 算法
    - 当前是否为安全状态？
    - 满足资源请求后是否为安全状态？
  - 应用
- 死锁解除
  - 原则：最小代价

处理机管理

– 例子：进程P3请求1个R2资源，即请求向量为（0,1,0）。是否进行资源分配以满足该请求？

	R1	R2	R3
P1	1	0	0
P2	6	1	2
P3	2	1	1
P4	0	0	2

Allocation 矩阵

	R1	R2	R3
P1	3	2	2
P2	0	0	1
P3	3	1	4
P4	4	2	2

Need 矩阵

R1	R2	R3
9	3	6

Resource 矩阵

R1	R2	R3
0	1	1

Available 矩阵

t1时刻状态（时刻t=t1）

思路

- t1时刻是否为安全状态？若不是，则拒绝分配
- 按需求将资源进行预分配，如果分配后系统仍是安全状态，则确认分配，否则拒绝分配

处理机管理

- 进程调度
  - 评价参数
    - 周转时间
    - 平均周转时间
    - 等待时间
    - 平均等待时间
    - ……
  - 算法
    - 先来先服务
    - 优先级
    - 短作业优先
    - 高响应比优先
    - 轮转
    - ……

处理机管理

– 例子

进程	到达时间	执行时间
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

进程ID	到达时间	运行时间	开始时间	结束时间	周转时间	等待时间
P_1	0	5				
P_2	0	4				
P_3	0	9				
P_4	0	3				
平均周转时间=		平均等待时间=		系统吞吐量=		CPU利用率=

存储管理/内存管理

- 存储管理目的
  - 充分利用内存，为多道程序执行提供存储基础
  - 尽可能方便用户使用
  - 系统能够解决程序空间比实际内存空间大的问题
  - 内存存取速度快
  - 存储保护与安全
  - 共享与通信
  - ……
- 存储管理的任务
  - 内存空间的分配与回收
  - 地址变换/地址重定位
  - 内存共享
  - 存储保护与安全
  - 内存“扩充”

存储管理/内存管理

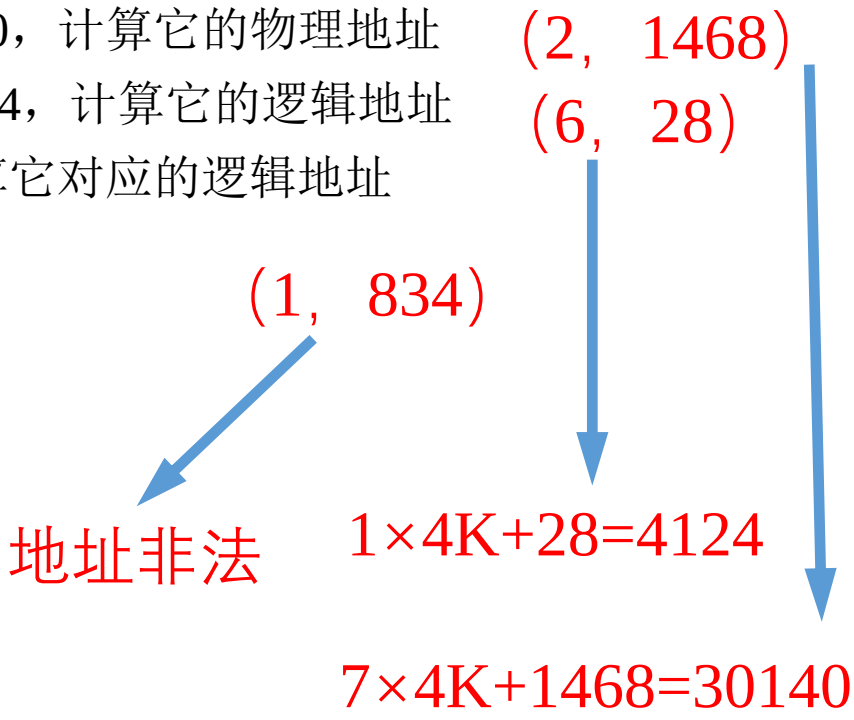
- 内存管理方法
  - 分区管理
  - 分页管理
  - 分段管理
  - 段页式管理
- 分页管理
  - 分页方法
    - 页帧（页框）
    - 页（页面）
  - 数据结构
    - 页表
    - 空闲页表

存储管理/内存管理

– 地址变换

- 例子：在一个分页系统中，页的尺寸为4Kbyte，假设进程x的页表如下。请回答：
  - ✓ 一条指令的逻辑地址为 9660，计算它的物理地址
  - ✓ 一条指令的物理地址为24604，计算它的逻辑地址
  - ✓ 如果物理地址为 4930，计算它对应的逻辑地址

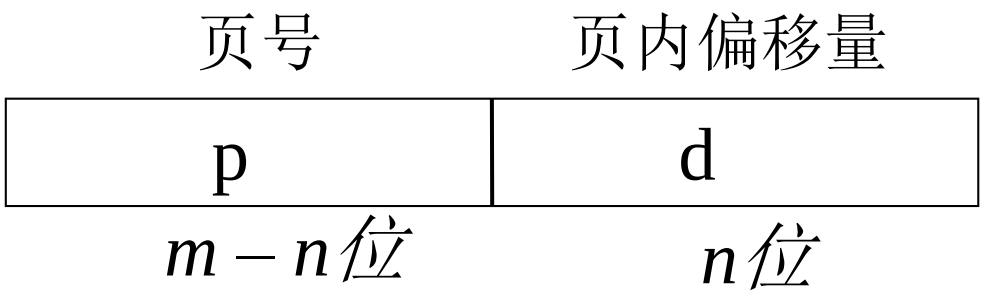
页号	页框号
0	4
1	6
2	7
3	9



存储管理/内存管理

- 硬件分页
  - ✓ 假设 $L = 2^n$

$$\begin{cases} p = \text{A的高(m - n)位} \\ d = A\text{的低端 n位} \end{cases}$$

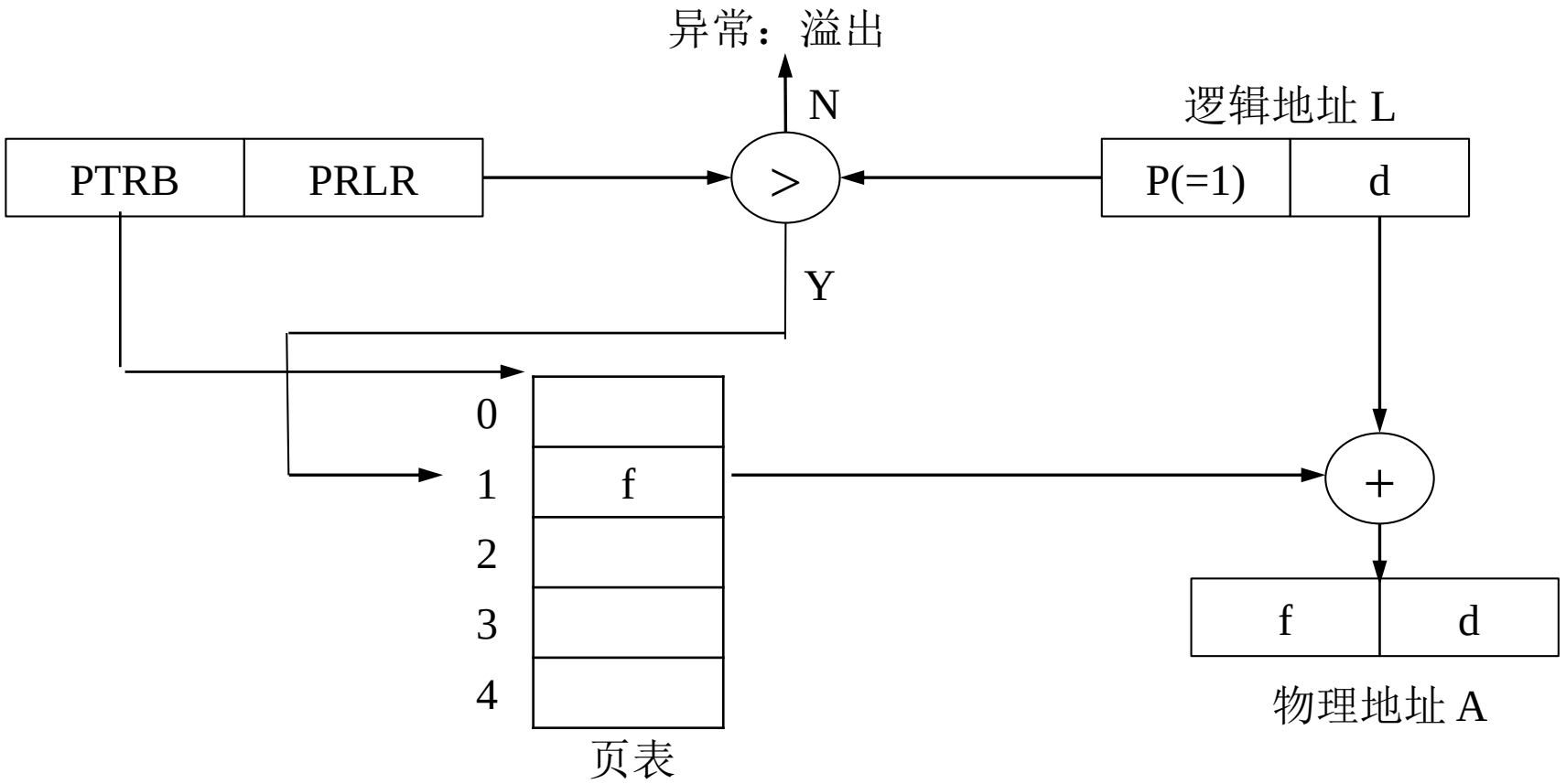


  - ✓ 例如，对32为系统，4KB页大小。有：m=32，n =12，m-n=20
  - ✓ 由逻辑页号p查页表得页框号f，地址变换为：

$$p+d \rightarrow f+d$$

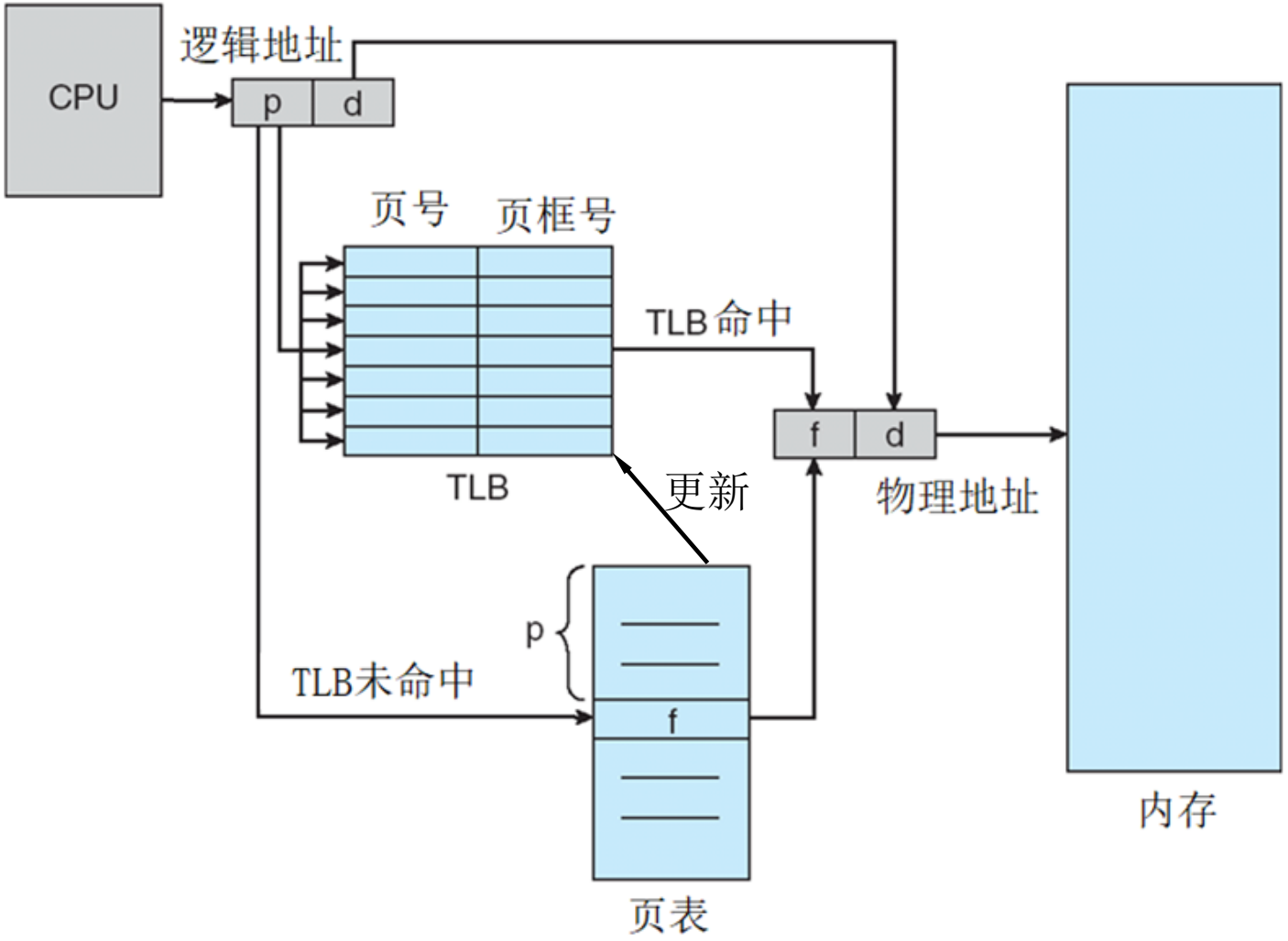
存储管理/内存管理

- 实现
  - ✓ 页表放在内存中
  - ✓ 页表基址寄存器 (PTBR) 指向页表（页表起始地址）
  - ✓ 页表长度寄存器 (PRLR) 记录页表长度（进程的分页数）



存储管理/内存管理

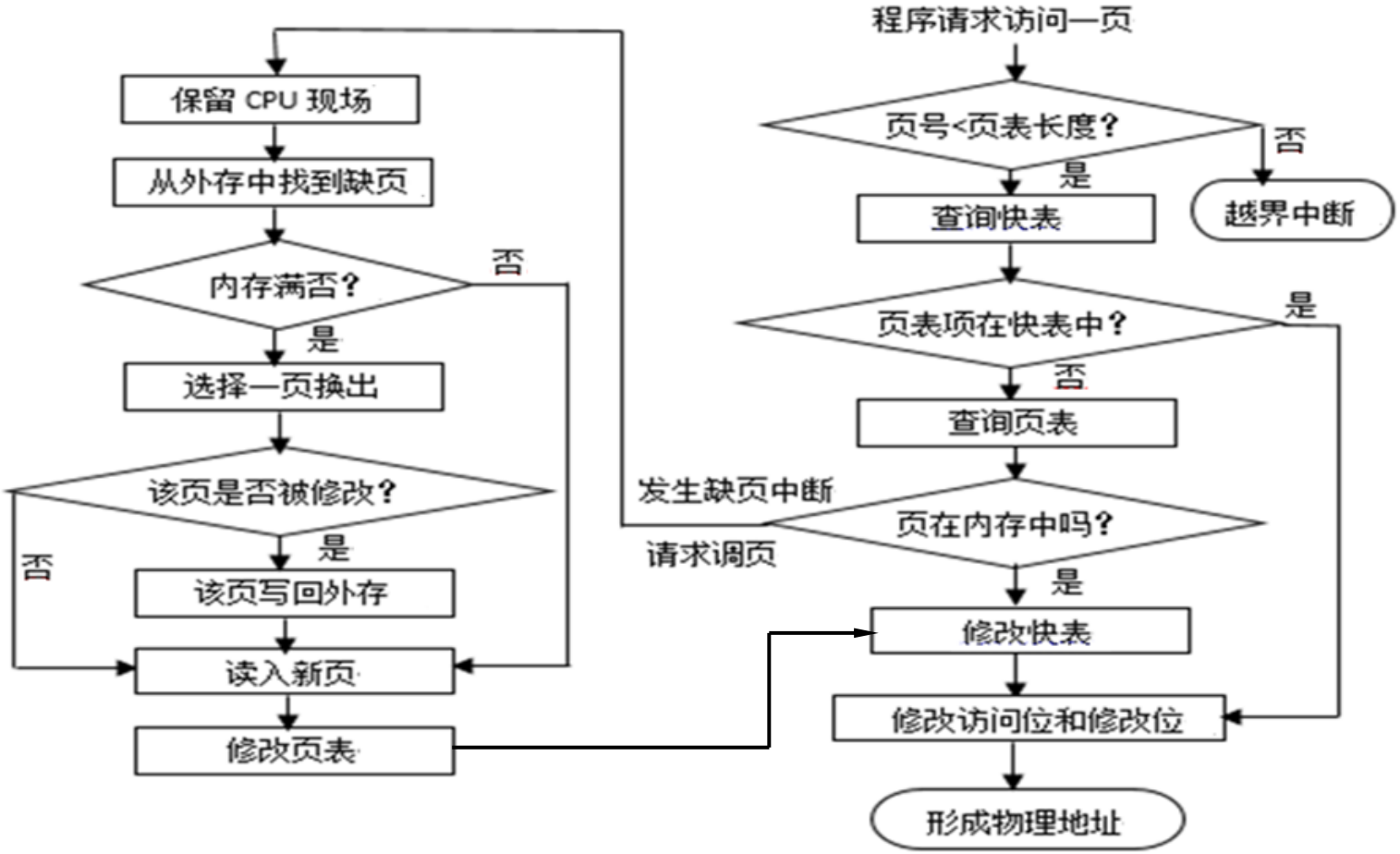
- 内存快表



存储管理/内存管理

- 虚拟存储技术
  - 局部性原理
    - 程序在执行过程中的一个较短时间内，所执行的指令地址或操作数地址分别局限于一定的内存区域中
    - 只将部分指令（代码）和数据装入内存并允许它执行
  - 虚拟存储器的特征
    - 多次性：一个进程被分成多次装入内存运行
    - 对换性：允许在进程运行的过程中，（部分）换入换出
    - 虚拟性：逻辑上的扩充
  - 装入时机
    - 预先装入：使用前装入
    - 请求装入：使用时装入

存储管理/内存管理



访问时间计算

- TLB命中：1次TLB+1次内存（读数据/指令）
- TLB未命中&页在内存：1次TLB+2次内存（1次页表+1次读数据/指令）
- TLB未命中&页不在内存：1次TLB+1次缺页处理+2次内存（1次页表+1次读数据/指令）

存储管理/内存管理

- 页面置换算法
  - FIFO、LRU、OPT、CLOCK
- 置换算法比较
  - FIFO：简单，易于实现，常用
  - OPT：理想的算法，不能实现，可以作为算法比较的标杆
  - LRU：最常用的算法，若干变种，需硬件支持
  - CLOCK：效率较高，易于实现

1	2	3	4	1	2	5	1	2	3	4	5
1	1	1	1								
	2	2	2								
		3	3								
			4								
F	F	F	F								

缺页数=？　缺页率=？

文件管理

- 概念
  - 文件
  - 文件系统
- 文件系统的功能
  - 统一管理文件的存储空间，实施存储空间的分配与回收
  - 实现文件的按名存取 目录管理
  - 名字空间到存储空间的映射
  - 实现文件信息的共享，并提供文件的保护和保密措施
  - 向用户提供一个方便使用的接口（提供对文件系统操作命令，以及提供对文件的操作命令：信息存取、加工等）
  - 系统维护及向用户提供有关信息
  - 文件系统的执行效率：文件系统在操作系统接口中占的比例最大,用户使用操作系统的感觉在很大程度上取决于对文件系统的使用效果
  - 提供与I/O的统一接口

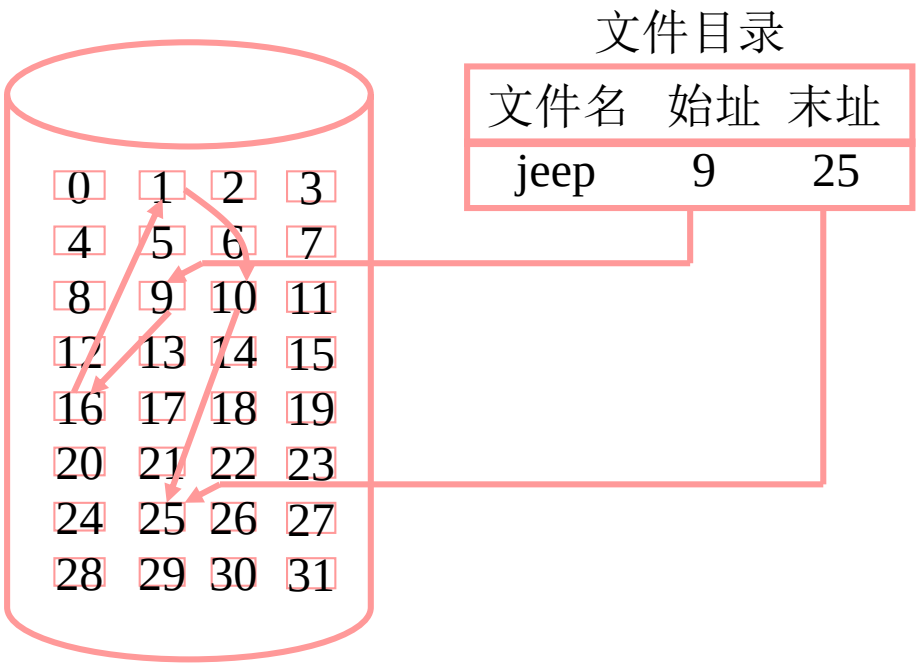
文件管理

- 文件逻辑结构
  - 堆结构
  - 顺序结构
  - 散列结构
- 文件读写方式
  - 顺序读写
  - 直接读写（随机读写）
  - 索引读写
- 文件物理结构
  - 连续文件
  - 链接文件
  - 索引文件

隐式链接文件
显式链接文件（FAT维护物理块号）

文件管理

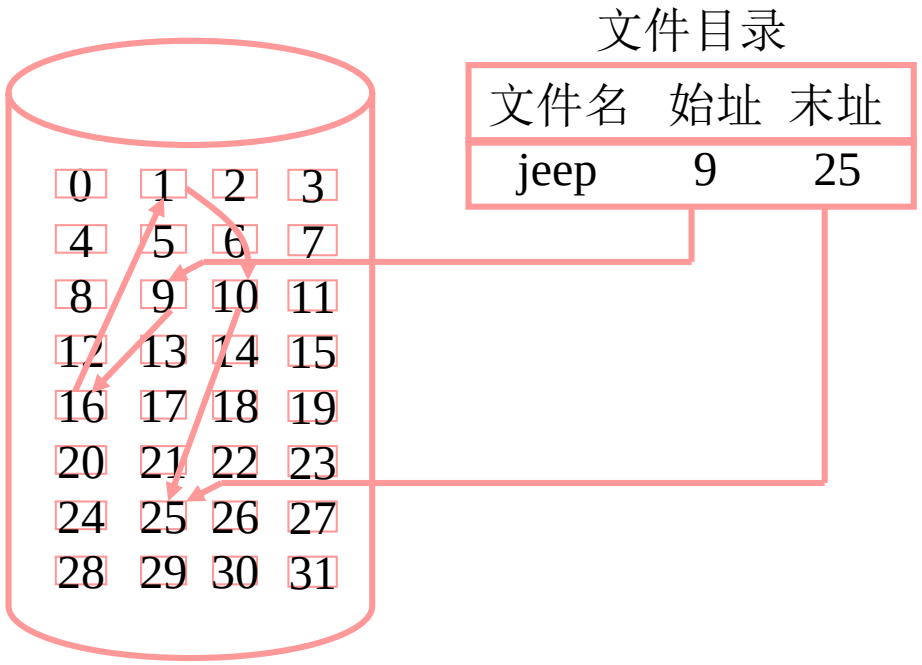
– 例子：链接文件（隐式链接文件）



- 优点
  - 提高了磁盘空间利用率,不存在外部碎片问题
  - 有利于文件插入和删除
  - 有利于文件动态扩充
  - 适合于顺序读写
- 缺点
  - 存取速度慢，不支持随机访问
  - 指针会占用额外的存储空间
  - 可靠性较低，如指针出错

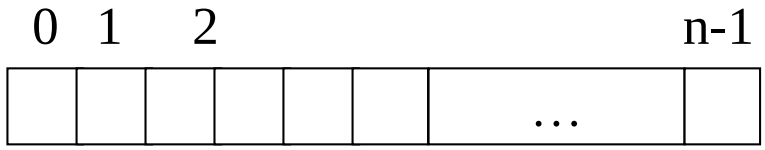
文件管理

– 例子：链接文件---FAT（显示链接文件）
  - 画出FAT?
  - FAT有多少项？每项多少位？
  - FAT有多大?



文件管理

- 目录与目录管理
- 空闲空间管理
  - 空闲块列表
  - 位示图

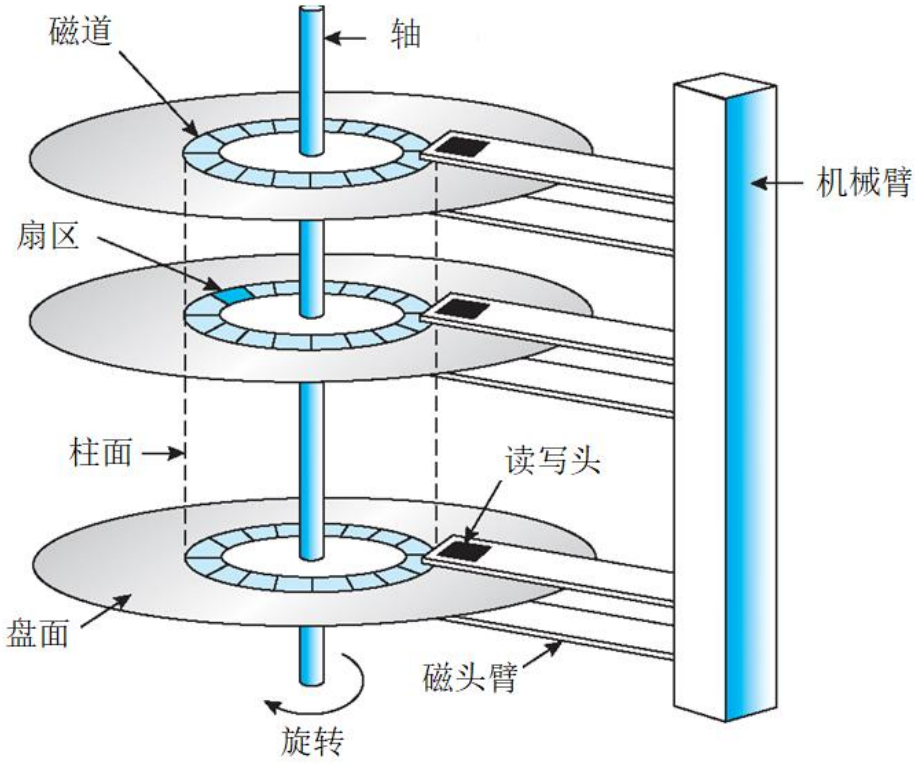


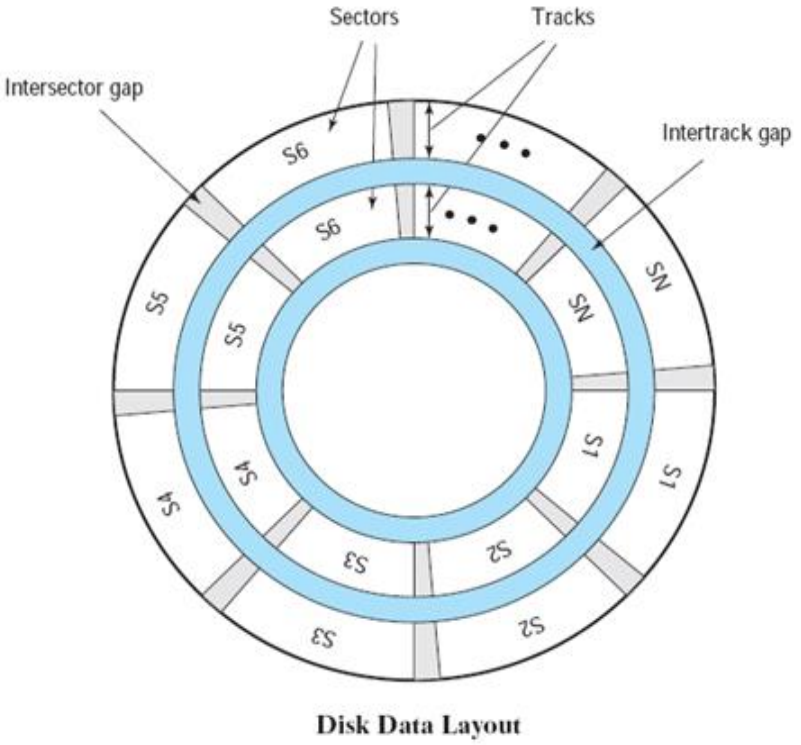
$$\text{bit}[i] = \begin{cases} 0 \Rightarrow \text{物理块 } i \text{ 空闲} \\ 1 \Rightarrow \text{物理块 } i \text{ 已使用} \end{cases}$$

  - 空闲块链
  - 成组链接

设备管理

- 磁盘结构


磁道
轴
机械臂
扇区
柱面
读写头
盘面
磁头臂
旋转


Sectors
Tracks
Intersector gap
Intertrack gap
Disk Data Layout

设备管理

- 磁盘读写指令的执行
  - 等待设备空闲（可用）
  - 等待数据通道可用
  - 寻道——定位磁道
  - 旋转——定位扇区
  - 数据传输
- 读写参数
  - 寻道时间：T_s
  - 旋转速度：r转/秒
  - 每条磁道N字节

T = 1/r
T旋转 = 1/(2T)
T传输 = b/(NT)

- 访问b字节的平均时间：T_a

$$T_a=T_s+1/(2\times r)+b/(r\times N)$$

设备管理

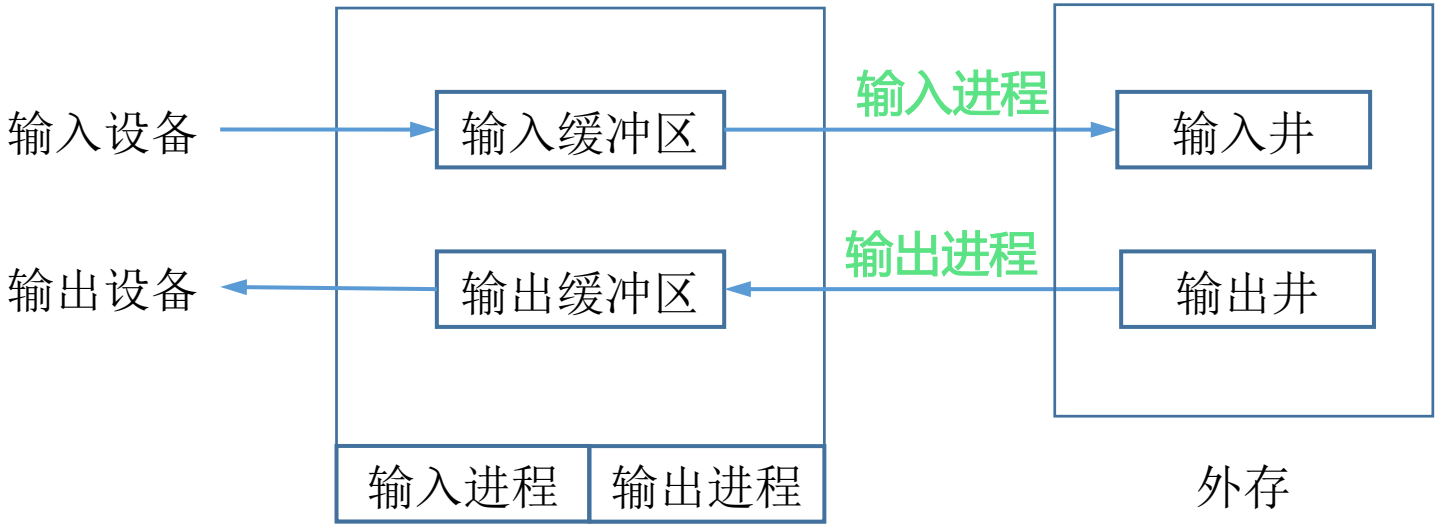
- 磁盘调度
  - 评价指标：磁头移动的磁道总数
  - 调度算法
    - 先进先出（FIFO）
    - 最短查找时间优先 （SSTF）
    - 扫描算法（SCAN）
    - 循环扫描（C-SCAN）
    - LOOK & C-LOOK算法
    - 例子：请求队列：98, 183, 37, 122, 14, 124, 65, 67；当前磁头位置：53；当前向磁道号0的方向移动。求：磁头移动的总磁道数
  - 磁盘调度算法的选择
    - SSTF 是常见的，并且具有自然的吸引力，因为它比 FCFS 具有更好的性能。
    - 对于磁盘负荷较大的系统，SCAN 和 C-SCAN 表现更好
    - SSTF 或 LOOK 是默认算法的合理选择

设备管理

- SPOOLING系统
  - Spooling是外部设备同时联机操作，又称为假脱机输入/输出操作，是操作系统中采用的一项将独占设备改造成共享设备的技术
  - Spooling系统是对脱机输入/输出工作的模拟，它必须有高速大容量且可随机存取的外存(如磁盘，磁鼓等)支持
  - 假脱机系统的组成
    - 输入井和输出井
    - 输入缓冲区和输出缓冲区
    - 输入进程和输出进程

设备管理

- Spooling系统的例子



设备管理

- I/O数据传输控制：原理&方法&特点
  - 轮询控制方式
    - 轮询方式也称为程序直接控制方式或“忙—等”方式
  - 中断控制方式
    - 现代计算机系统中广泛采用中断控制方式对I/O设备进行控制
  - 直接内存存取控制方式
    - DMA（Direct Memory Access，直接内存存取）方式用于高速外部设备与内存之间直接批量数据的传输
  - 通道控制方式
    - 通道本质上是一个简单的处理器，专门负责输入、输出控制，具有执行I/O指令的能力，并通过执行通道I/O程序来控制I/O操作

END