操作系统结构分析
第5章设备管理

南京邮电大学
计算机学院 信息安全系
曹晓梅 陈丹伟

手机：189-0518-4599
QQ： 757375652
email: caoxm@njupt.edu.cn

引言

- 设备管理是指操作系统对计算机系统中除CPU和内存以外的设备的管理，它与其他功能联系密切，特别是文件系统。
- 设备不但种类繁多，而且它们的特性和操作方式相差很大，因此，设备管理是操作系统资源管理中最为复杂、最多样化，且与硬件密切相关的部分。
  - 为了提升系统性能，使用了I/O中断、通道、缓冲区管理、磁盘驱动调度等技术；效率
  - 为了屏蔽不同设备的物理细节和操作过程，为I/O子系统提供统一的设备访问接口，配置了设备驱动程序；
  - 为了统一设备和文件的处理，将设备文件和普通文件纳入同一保护机制，OS将所有设备抽象为文件。 通用性

目标

- I/O设备的基本要素
- 4种I/O控制方式(主机和设备的交互方式)
- 用户如何向设备发起操作？OS充当什么角色？
- 设备如何分配？
- 什么是假脱机技术？如何实现？
- 磁盘管理与调度

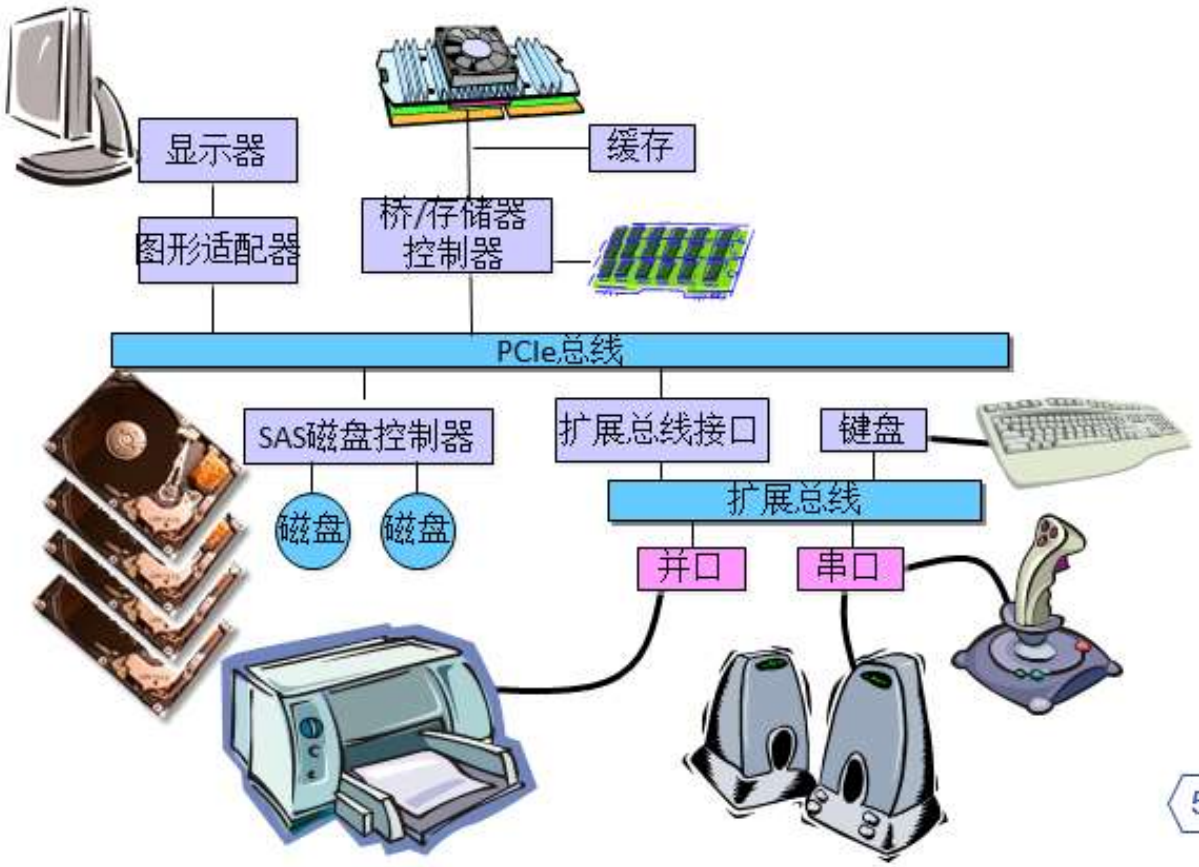
3

内容纲要

Contents Page

4

I/O硬件系统概览



纲要

1.1 I/O系统的硬件组成

I. 设备分类
II. 总线
III. 端口
V. 控制器

1.2 I/O控制方式

设备分类

- 存储型设备 磁盘
  - 如磁盘、磁带、DVD-ROM等；
  - 存储持久性信息，以大容量存储和快速检索为目标；
  - 相关调用有 open(), read(), write(), seek()等。
- I/O型设备 终端
  - 如显示器、打印机、键盘、鼠标、通信设备等；
  - 将外界信息输入计算机，把计算结果从计算机输出，完成人机交互或计算机间通信；
  - 相关调用有 get(), put()等。

设备连接计算机的基本要素：
- 总线(bus)
- 端口(port)
- 控制器(controller)

总线

- 一组线路和一组定义在线路上传输数据的协议
  - 并行总线(Parallel Bus)
  - 串行总线(Serial Bus)

端口

- 设备与计算机的连接点
  - 并行端口 (Parallel Port)
  - 串行端口(Serial Port)

PCIe、SAS、USB、VGA、HDMI、DVI、Thunder Blot等

设备控制器

- 定义　用于操作端口、总线或设备的一组电子器件。
  - 有些控制器较为简单，是计算机内的单个芯片（或芯片的一部分），可以集成在主板上，以控制串口线路的信号，如串口控制器。
  - 有些控制器实现功能复杂，包含处理器、微代码和一些专用内存，需要做成一块独立的集成电路板并与计算机相连，通常称为适配器，如图形适配器、SCSI总线适配器。
  - 有的设备有内置控制器，如SATA硬盘控制器板就在硬盘的一侧。

- 功能

接收和识别命令 、数据交换 、标识和报告设备状态、地址识别、数据缓冲、差错控制。

CPU

纲要

1.1 I/O系统的硬件组成

1.2 I/O控制方式

I. 轮询
II. 中断
III. DMA
IV. 通道

CPU对I/O操作的控制方式

总述

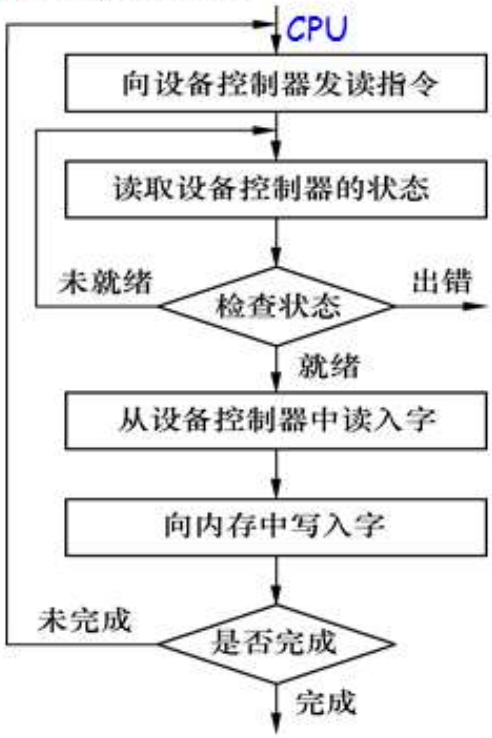
- **轮询** 又称：程序I/O控制方式
  - 简单的忙-等待方式
- **中断驱动I/O控制方式**
  - 中断机制的引入
- **直接存储器访问控制方式**
  - DMA控制器、数据传输单位扩大
- **I/O通道控制方式**
  - 通道、I/O操作和数据传送的独立

I/O控制方式发展过程中贯穿着这样的宗旨：尽量减少CPU对外设的干预，把CPU从繁杂的I/O控制中解脱出来，以便有更多的时间进行输出处理。

1.2 I/O控制方式 1. I/O硬件原理

轮询(Polling) 读操作为例

- 工作流程



- 问题
  - CPU和外设**串行工作**

效率低下

中断！

1.2 I/O控制方式

中断(Interrupt)

- 工作流程

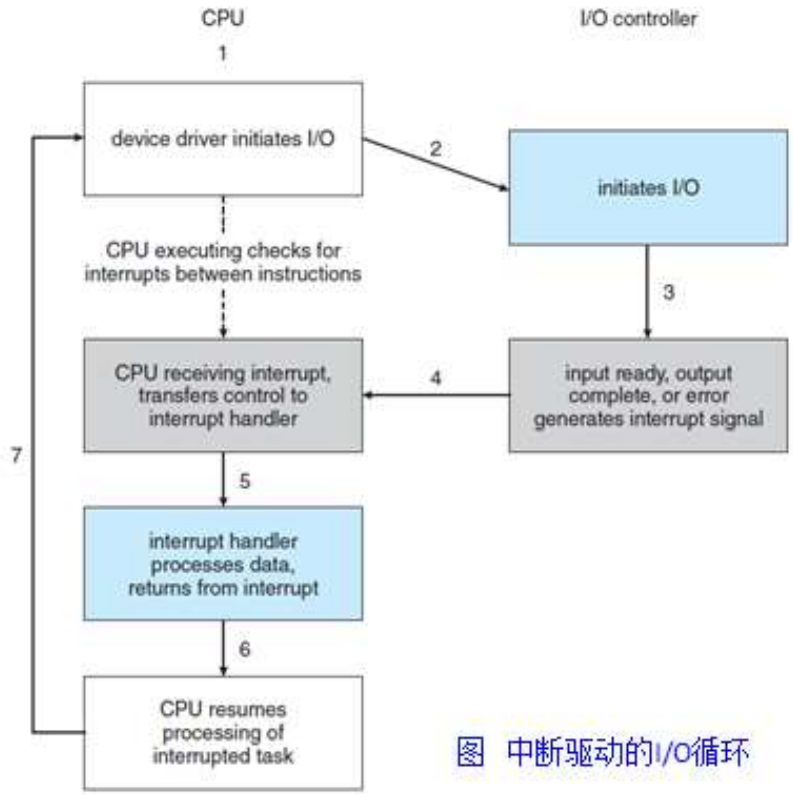


图 中断驱动的I/O循环

- 问题
  - 中断方式实现了一定程度上的并行操作，但CPU仍全程参与数据传输操作。

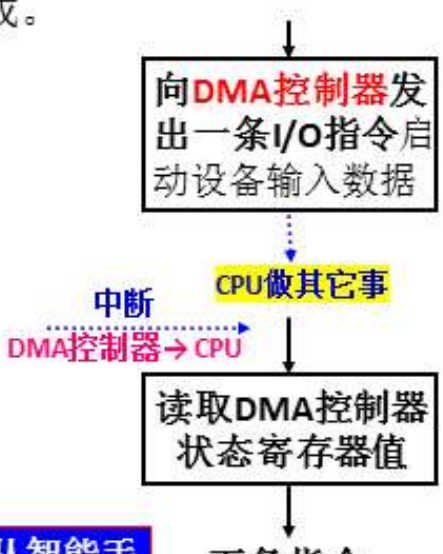
能否直接让内存与设备直接交换数据而不占用CPU呢？

DMA

1.2 I/O控制方式

DMA(Direct Memory Access)(1/3)

- 设计思想
  - 在DMA方式中，内存和外设之间有一条数据通路，在内存和外设之间成块传送数据过程中，由DMA控制器取代CPU来控制数据传输，直接执行到数据块传输完成。
  - 特点：
    - 数据传输单位
      数据块
    - 数据传输途径
      设备⇔内存
    - CPU干预
      限于数据块传送开始与结束



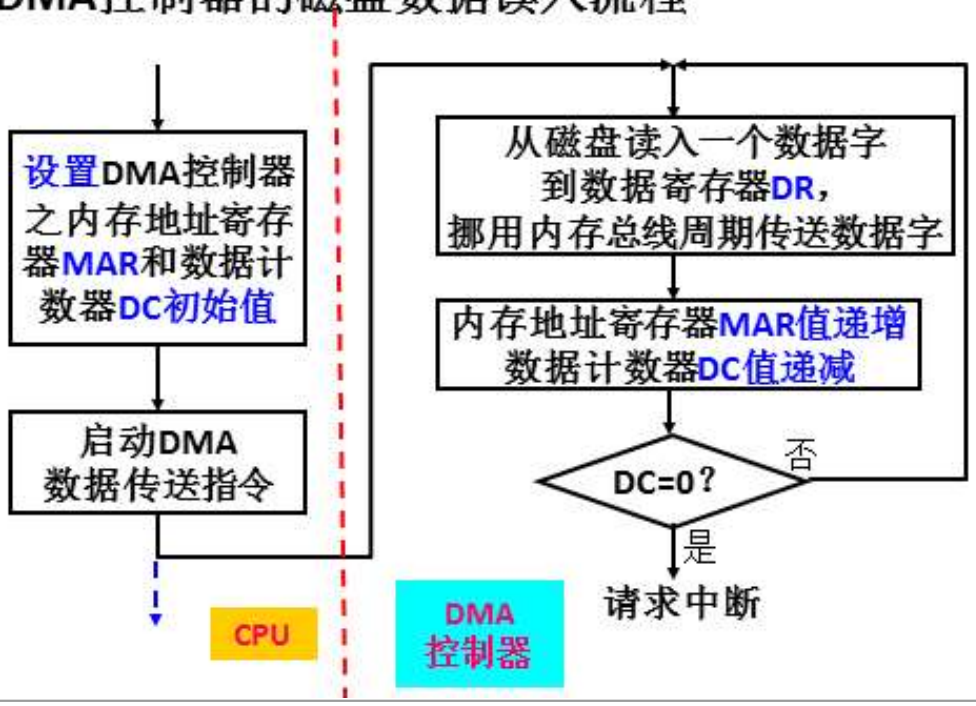
用于数据块传输的DMA控制器是涵盖从智能手机到大型机的所有现代计算机的标准组件。

1.2 I/O控制方式

① 命令寄存器(Command Register, CR)
② 内存地址寄存器(Memory Address Register, MAR)
③ 数据寄存器(Data Register, DR)
④ 数据计数器(Data Counter, DC)

DMA(2/3)

- DMA控制器中的寄存器
- 基于DMA控制器的磁盘数据读入流程



1.2 I/O控制方式

DMA(3/3)

- 讨论

• 周期窃取(cycle stealing)

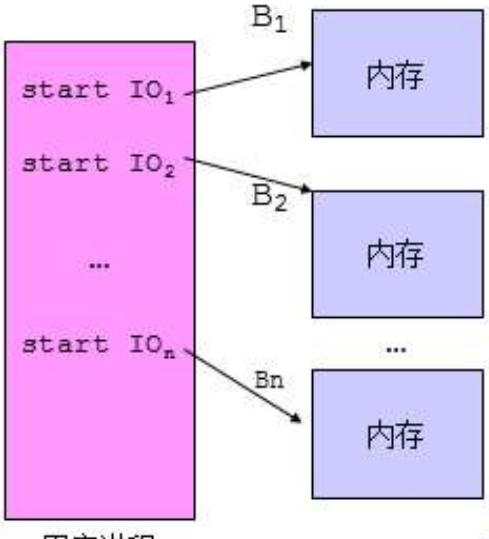
DMA在主存和设备之间交换数据时需要占用内存总线，如果此时CPU也要使用总线则总是将占有权让给DMA，这种现象被称为“周期窃取”。

• DMA方式的问题：

问题一：以数据块为周期访问设备，每一个数据块传输的开始和结束需要CPU干预

问题二：大型计算机系统中高速设备共享DMA接口的问题。

通道



通道(1/2)

定义

- 通道又称输入输出处理器，是独立于CPU的专门实现输入输出工作的处理器。
- 通道以内存为中心，控制设备和内存之间进行数据传送，把CPU从琐碎的输入输出操作中解放出来。
- I/O通道可以控制若干个控制器工作，控制器又可以连接若干台同类型的外部设备并控制这些I/O设备工作。
- 通道一般用于具有较多高速外设的大型计算机系统。

通道(2/2)

工作流程

CPU:

1. 生成通道程序并装入内存
2. 进程i 进入阻塞状态
3. 发送I/O请求至通道
4. 调度一个新进程进入CPU执行

通道

5. 从内存中取出通道程序
6. 执行通道命令，在内存和设备间传输数据
7. 执行结束后发出中断请求CPU干预
8. 进程i 进入就绪状态，等候进程调度

单选题 2.5分

CQ1.1 以下不使用中断机构的I/O控制方式是（ ）。

A 中断控制方式

B DMA控制方式

C 程序I/O方式

D 通道控制方式

单选题 2.5分

CQ1.2 DMA控制方式是在（ ）之间建立一条直接数据通路。

A I/O设备和内存

B 两个I/O设备

C I/O设备和CPU

D CPU和内存

单选题 2.5分

CQ1.3 在以下I/O控制方式中，需要CPU干预最少的是（ ）。

A 程序I/O方式

B 中断控制方式

C DMA控制方式

D 通道控制方式

单选题 2.5分

CQ1.4 以下叙述中正确的是（ ）。

I. 在DMA控制方式下，外部设备和CPU之间直接进行成批的数据交换。

II. 通道执行CPU指令构成的程序，与设备控制器一起共同实现I/O设备的控制。

III. I/O通道控制方式是一种以内存为中心，实现设备和内存直接交互数据的控制方式

A 仅I、III B 仅II、III C 仅III D 仅I、II

内容纲要

Contents Page

1. I/O硬件原理
2. I/O软件系统
3. 磁盘管理

纲要

2.1 概述

I. 功能
II. 设计目标
III. I/O软件系统层次

2.2 内核I/O子系统

2.3 I/O请求生命周期

功能

- 提供使用设备的用户接口
  - 命令接口和编程接口
- 设备分配和释放
  - 设备、设备控制器、通道等
- 设备的访问和控制
  - 并发访问及差错处理
- I/O缓冲和调度
  - 提高I/O访问效率，缓解CPU与外设速度上的矛盾

设计目标

I/O系统 = 硬件 + 软件

- 高效率(efficiency)

  I/O中断、DMA、通道、缓冲区管理、虚拟机、磁盘驱动调度.....

  核心效率问题：**磁盘I/O的效率**

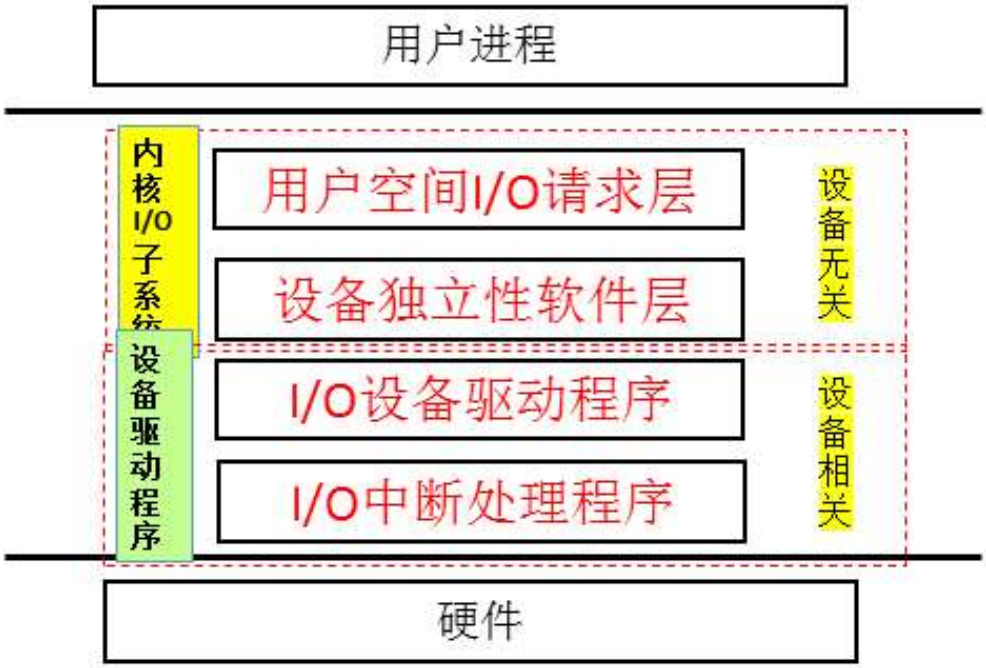
- 通用性 (generality)

  屏蔽硬件细节，对各种硬件提供一个统一使用接口。

  采用层次化的方法设计I/O软件系统功能，屏蔽低层硬件细节。

I/O软件系统层次(1/4)

参见课本P260



I/O软件系统层次(2/4)

用户空间I/O请求层

- 实现与用户交互的接口，用户可以直接调用在用户层提供的、与I/O操作有关的库函数，如I/O系统调用、I/O格式化、SPOOLing等。2.2.3

设备独立性软件层

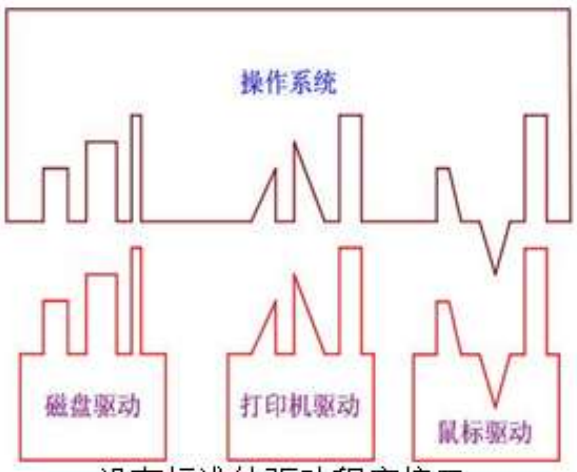
- 执行适用于所有设备的常用I/O功能，并向用户层软件提供一致性接口。

主要功能：
- 设备命名和设备保护
- 提供独立于设备的块大小
- 缓冲区管理 2.2.1
- 设备分配和: 2.2.2 ⋮
- 错误报告

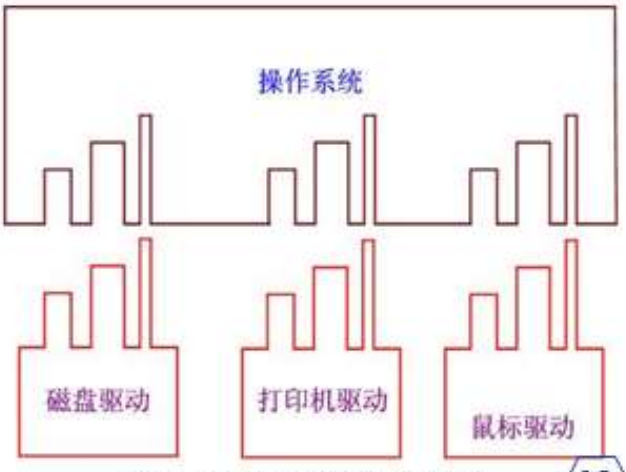
I/O软件系统层次(3/4)

设备驱动程序

- 该层包含了与设备密切相关的代码，每种设备类型都应该有对应的设备驱动程序。
- 设备驱动负责I/O设备的初始化，执行I/O设备的驱动例程，执行中断处理例程。



没有标准的驱动程序接口

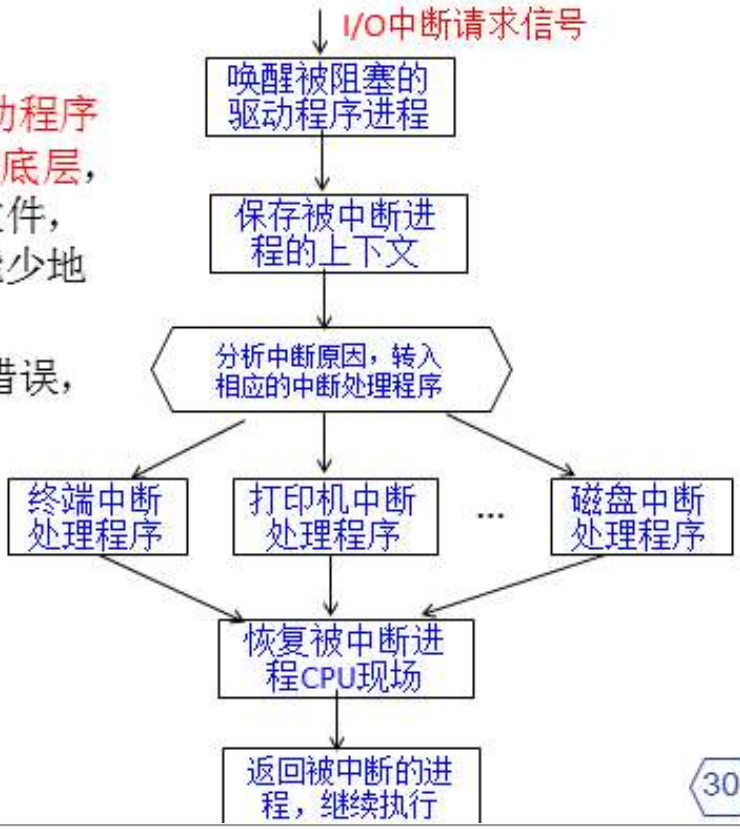


具有标准的驱动程序接口

I/O软件系统层次(4/4)

中断处理程序

- I/O中断处理程序是设备驱动程序的组成部分，位于操作系统底层，是与硬件设备密切相关的软件，它与系统的其余部分尽可能少地发生联系。
- 功能：处理I/O中断，报告错误，唤醒驱动程序等



纲要

2.1 概述

2.2 内核I/O子系统

I. 缓冲技术

II. 设备分配及调度

III. SPOOLing技术

2.3 I/O请求生命周期

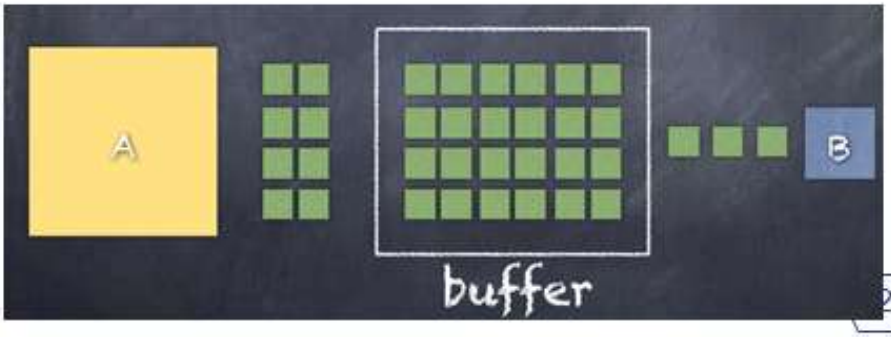
缓冲区的定义和用途

buffer vs cache

- **缓冲区(buffer)**
  - 系统在内存中开辟的一个专门的区域，用于临时存放I/O操作的数据。
- **缓冲区的主要用途**
  - 处理生产者设备和消费者设备之间数据流速度不匹配的问题。

设备速度：

$S_A >> S_B$



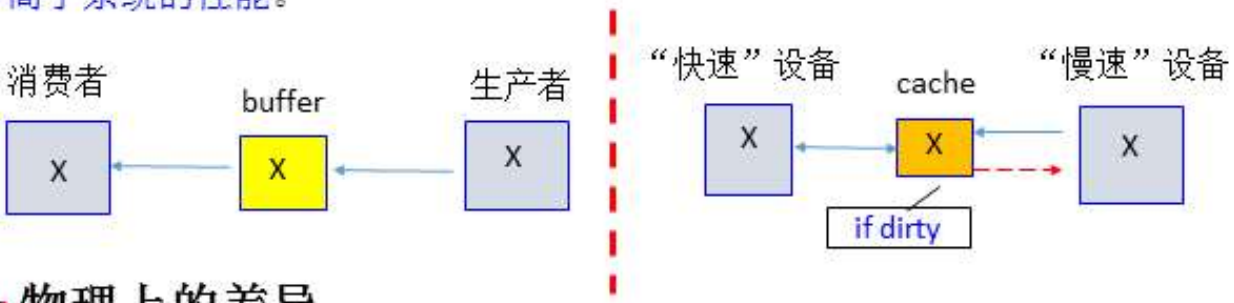
Buffer is a segment of memory in which items are placed while waiting to be transferred from an input device or to an output device.
Cache speeds up processing time because it stores frequ-ently used instructions and data.

Buffer VS Cache

用途上的区别

Buffer引入的主要目的是在输入、输出过程中进行流量整形，把突发的大数量较小规模的I/O整理成平稳的小数量较大规模的I/O，以减少响应次数。

基于程序局部性原理，Cache保存着CPU刚用过的数据或循环使用的部分数据，在命中时从Cache中读取数据会更快，减少了CPU等待的时间，提高了系统的性能。



物理上的差异

Buffer一般位于传统的DRAM中；
Cache的位置不绝对，一般位于专门的SRAM（也可能在DRAM，或者硬盘板载），其速度相对更快。

设备分配及调度(1/2)

设备分配方式 依赖于设备的物理特性

- 独占分配方式 大多数低速设备都适合采用这种分配方式

  将一个设备分配给某个进程后便一直由该进程独占，直至该进程完成或释放该设备，系统才能将该设备分配给其他进程使用。

- 共享分配方式

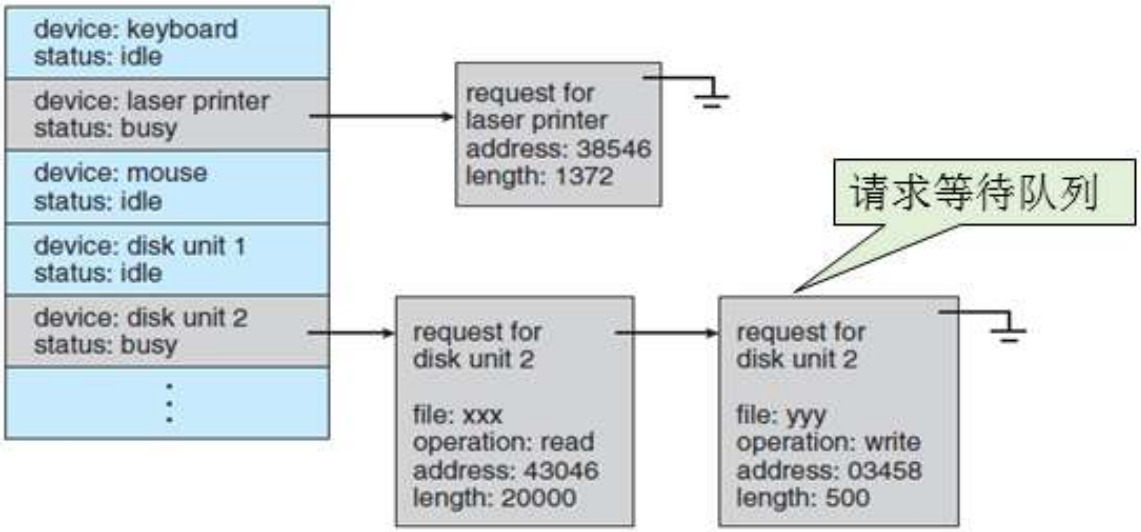
  对于高速共享设备(如磁盘)一般不进行分配，主要工作是驱动调度。

- 虚拟分配方式 如 SPOOLing技术

  在分配时并不是真实的物理设备，而只是在高速磁盘中为进程分配的存储区，实现对慢速独占设备的模拟分配和共享。

设备分配及调度(2/2)

- 设备状态表(device-status table)



- I/O调度算法
  - 先来先服务
  - 优先级高者优先

SPOOLing技术(1/3)

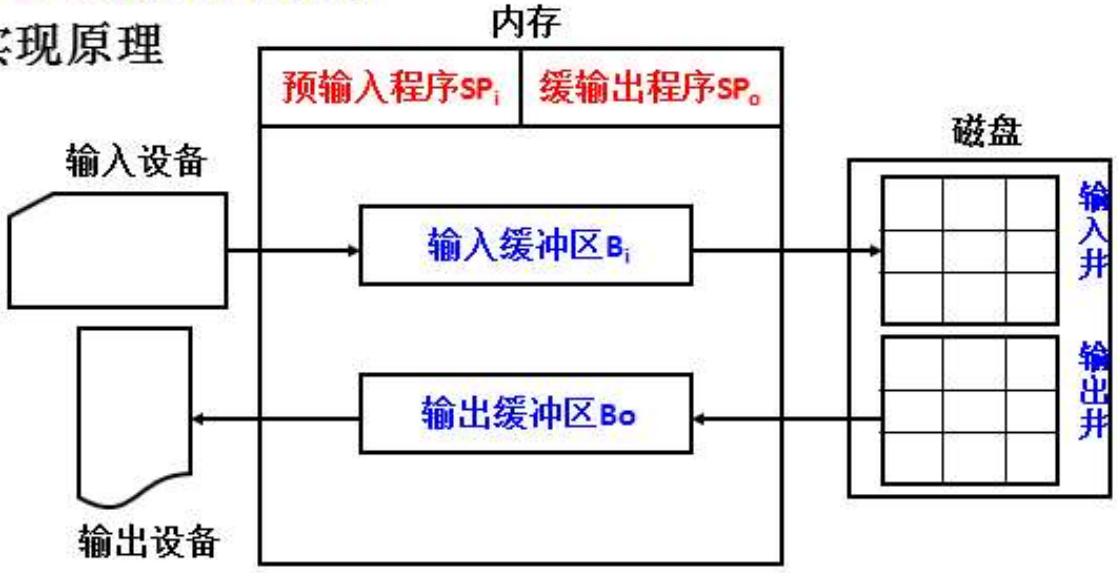
Simultaneous Peripheral Operations On-Line, SPOOLing

外围设备同时联机操作

- 概念
  - 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。
  - 借助SPLOOLing同时处理多个进程对独占设备的请求，让每个进程都感觉自己拥有设备，我们称这种设备为“虚拟设备”
  - SPOOLing技术借助于高速随机外存（磁盘）的支持。

SPOOLing技术 (2/3)

- 实现原理



磁盘
- 输入井：模拟脱机输入时的磁盘缓冲区，用于收容I/O设备输入的数据
- 输出井：模拟脱机输出时的磁盘缓冲区，用于收容用户进程的输出数据

内存
- 输入缓冲区：用于暂存由输入设备送来的数据，以后再传送到输入井
- 输出缓冲区：用于暂存从输出井送来的数据，以后再传送给输出设备

SPOOLing技术的实现(3/3)

- 预输入程序　模拟脱机输入时的外围控制机，将用户要求的数据从输入设备通过输入缓冲区再送到输入井。
- 缓输出程序　模拟脱机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。

- 优点
  - 从对低速I/O设备进行I/O操作变为对对输入井/输出井的操作，提升了I/O速度。
  - CPU数据处理与设备I/O操作的并行化，提升了系统资源的使用效率。

纲要

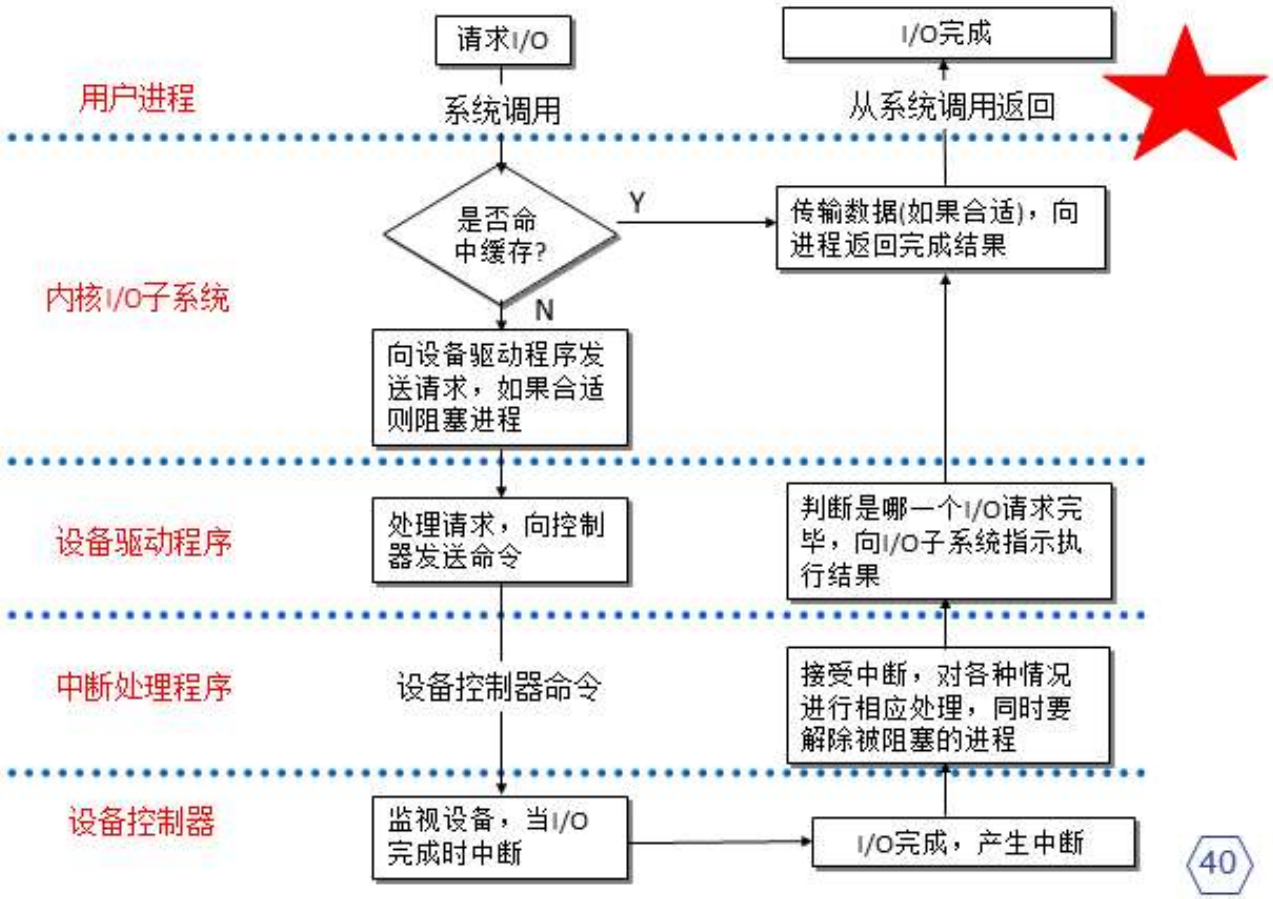
2.1 概述

2.2 内核I/O子系统

2.3 I/O请求生命周期

2.3 I/O请求生命周期

课本P264 图5-1



I/O操作过程举例

count=read(fd, buffer, n);

1. 执行用户层软件的用户进程，对已打开文件的文件描述符fd执行read函数；
2. 设备独立性软件检查参数是否正确，若正确，检查cache中有无要读的信息块；若有，就将该数据返回给用户进程并完成I/O请求；
3. 若数据不在cache，就需要执行物理I/O操作，设备独立性软件将设备的逻辑名转换成物理名，检查对设备操作的许可权，该用户进程会从运行队列移到设备的等待队列，并调度I/O请求；
4. 内核启动设备驱动程序，分配存放读出块的缓冲区，准备接收数据，并向设备控制状态寄存器发送启动命令，或建立DMA传输，启动I/O；
5. 设备控制器控制设备硬件以执行数据传输；
6. 设备驱动程序可以轮询检测状态和数据，或通过DMA控制器控制数据传输，一旦传输完成后产生I/O结束中断；
7. CPU响应中断，转向磁盘中断处理程序，保存必要的数据，并向设备驱动程序发送信号，然后从中断返回；
8. 设备驱动程序接收到信号，确定I/O请求是否完成，并向设备独立性软件发送信号，通知请求已完成；
9. 内核将数据或返回代码传递给用户进程的地址空间，将该用户进程从等待队列移动到就绪队列；
10. 将该用户进程移到就绪队列会使其不再阻塞，当CPU调度该用户进程时，该进程继续执行read系统调用之后的语句。

41

单选题 2分

CQ2.1 与设备相关的中断处理过程是由（　　）完成的。

A 用户层I/O软件

B 设备无关的操作系统软件

C 硬件

D 设备驱动程序

单选题 2分

CQ2.2 调制解调器正在接受一个文件，并且保存在硬盘。调制解调器大约比硬盘慢1000倍。为解决这一问题，可采用的技术是（ ）。

A 并行技术

B 通道技术

C 缓冲技术

D 虚存技术

单选题 2分

CQ2.3 缓冲技术中的缓冲池在（ ）中。

A 内存

B 外存

C ROM

D cache

单选题 2分

CQ2.4 通过硬件和软件的功能扩充，把原来独占的设备改造成若干用户共享的设备，这种设备称为（ ）。

A 虚拟设备

B 存储设备

C 系统设备

D 用户设备

单选题 2分

CQ2.5 假脱机输入输出利用（ ）作为缓冲区来实现虚拟设备，以提升独占设备的利用率。

A 打印机

B 内存

C 磁盘

D 磁带

内容纲要

Contents Page

1. I/O硬件原理

2. I/O软件系统

3. 磁盘管理 secondary storage
   - Hard Disk Drives
   - Solid State Disk

纲要

3.1 基础知识

I. 磁盘构造
II. 磁盘参数
III. 磁盘格式化
IV. 磁盘性能指标

HDD

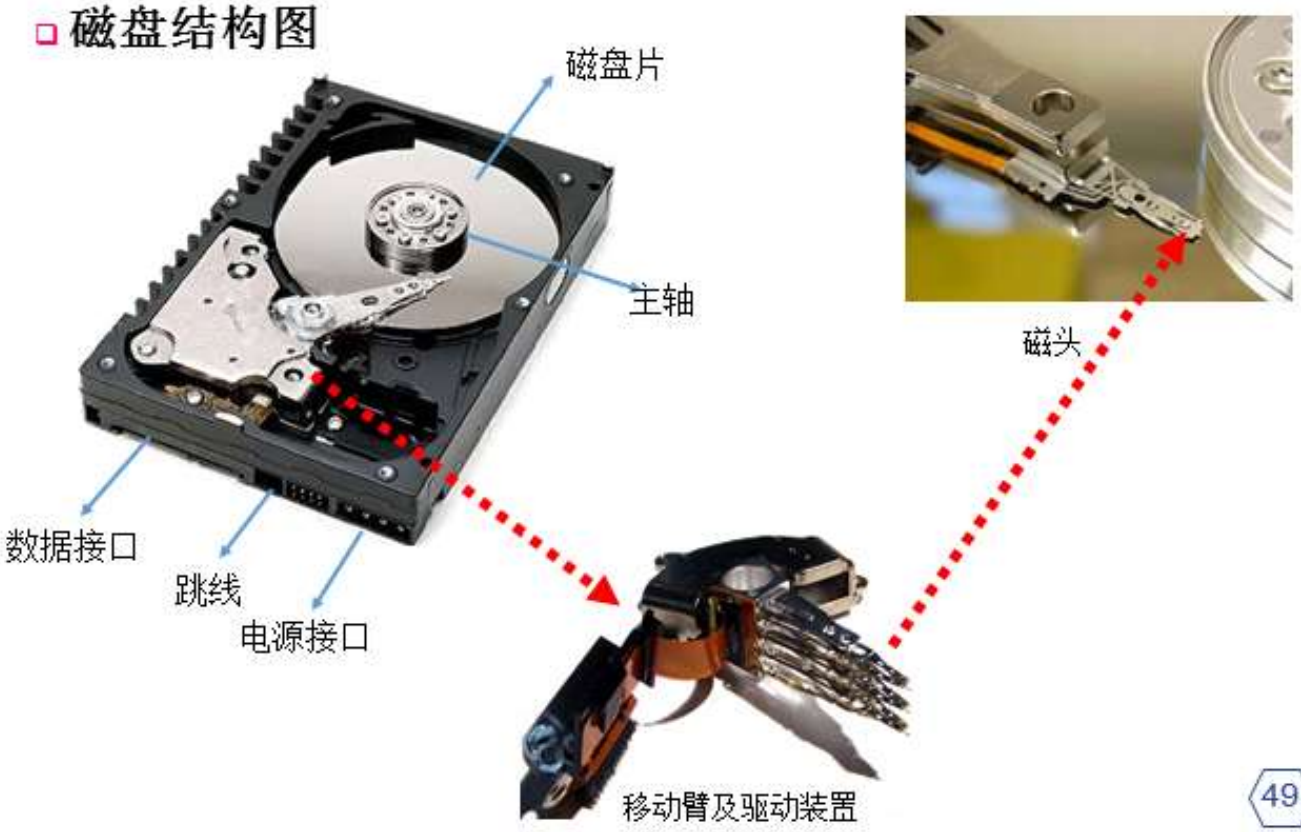
V. 固态硬盘 SSD

3.2 移臂调度策略

3.3 独立磁盘冗余阵列

磁盘构造(1/3)

- 磁盘结构图

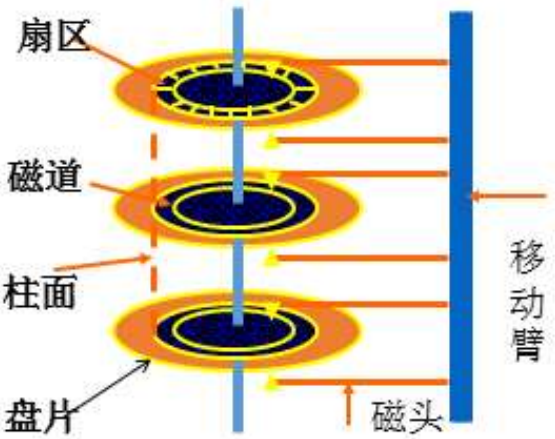


磁盘构造(2/3)

- 磁盘结构图(续)

磁盘构造(3/3)

- **磁道(track)**
  能被磁头访问的一组同心圆
- **柱面(cylinder)**
  磁头位置下所有磁道组成的柱体
- **扇区(sector)**
  数据存放的基本单位，通常为512B



磁盘逻辑块 Logic Blocks

- 信息读写、传输的最小单位，一般为512B，每个逻辑块映射到一个物理扇区。

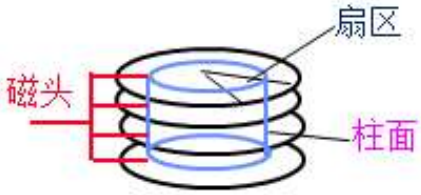
磁盘容量 = 磁头数×磁道数×每道扇区数×每扇区字节数

磁盘参数(1/3)

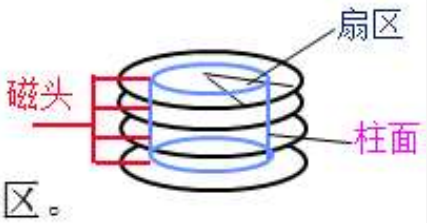
磁盘块定位三要素

柱面号、磁头号、扇区号

- **柱面号(C)**
  指盘面上的磁道号，磁道号从0开始由外向里编号，不同盘面上具有相同编号的磁道属于同一柱面；
- **磁头号(H)**
  指出读写磁头所在的盘面，磁头号按从上到下的盘面次序从0开始编号。
- **扇区号(R)**
  按磁盘旋转方向从0开始给各扇区编的号。



磁盘参数(2/3)



- **磁盘块号** Logic Block Address(LBA)
  - 磁盘块0是第0号柱面、第0号磁头的第0个扇区。
  - **先按磁道内扇区顺序，再按柱面内磁头顺序，再按从外到内的柱面顺序**进行编号。
- **参数间的转换** **磁盘块块号 ⟺ (柱面号，磁头号，扇区号)**

例6-1 设t为每个柱面的磁头数，s为每个磁道的扇区数，某个块号的定位参数为(i, j ,k)，其中i, j, k分别表示该块的柱面号、磁头号和扇区号，则对应的块号x为：

$$x = i*t*s + j*s + k$$

（t*s：每个柱面的扇区数）

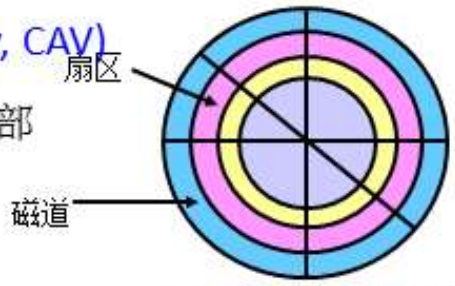
由块号求它在磁盘上的位置，则：

```
i=  x DIV (t * s)
j= (x MOD (t*s)) DIV s
k= (x MOD (t*s)) MOD s
```

前提： 柱面号、磁头号、扇区号、物理块号均从0开始编号！

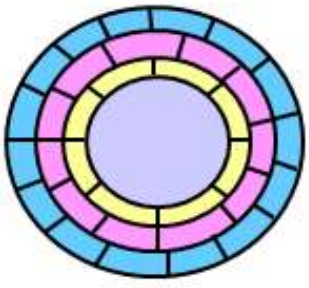
磁盘参数(3/3)

- **恒定角速度**(Constant Angular Velocity, CAV)
  - 所有磁道的扇区数相等，内部磁道到外部磁道的比特密度不断降低
  - 磁盘转速恒定
  - 用于硬盘



每条磁道上的扇区数相等

- **恒定线速度** (Constant Linear Velocity, CLV)
  - 每个磁道的比特密度均匀
  - 外磁道扇区比内磁道的多，可存放更多数据
  - 为保持传输数据率的恒定，转速要不断变化
  - 用于CD-ROM和DVD-ROM驱动器


每条磁道上的数据密度相等

磁盘格式化

- 低级格式化(low-level formatting)
  - 又称物理格式化(physical formatting)
  - 将盘片划分出磁道、扇区
  - 为每个扇区使用特殊的数据结构进行填充，包括一个头部、数据区域和一个尾部
- 高级格式化(high-level formatting)
  - 又称逻辑格式化(logical formatting)
  - 划分由相邻柱面组成的多个分区(partition)
  - 为分区构建文件系统，在磁盘上初始化文件系统数据结构，如空闲、已分配分区表、一个空目录等。

磁盘的性能指标(1/2)

柱面号、磁头号、扇区号 确定磁道

- 寻道时间(seek time) T_s
  - 将磁头定位到正确磁道(柱面)上所花的时间，与盘片直径和传动臂速度相关；平均20ms
- 旋转延迟(rotational latency) T_r
  - 指定扇区旋转到磁头下面所经历的平均延迟时间Tr= 1/(2*r)，r为磁盘的旋转速度。
  - 典型的旋转速度大多为5400r/min，每转需时11.1 ms，平均旋转延迟时间Tτ为5.55ms

一个10,000 r/min的磁盘平均旋转延迟为？ 3ms

- 传输时间(transfer time) T_t
  - 传输扇区内的数据的时间，同样取决于磁盘的旋转速度。

Tt = b/(r*N) ，b为要传送的字节数，N为一个磁道中的字节数，r为转速

磁盘完成一次I/O的平均访问时间： $T_a = T_s + T_r + T_t$

磁盘的性能指标(2/2)

常见题型

例6-2 某软盘有40个磁道，磁头从一个磁道移到另一个磁道需要6ms。文件在磁盘上非连续存放，逻辑上相邻的数据块的平均距离为13磁道，每块的旋转延迟时间及传输时间分别为100ms、25ms，问读取一个100块的文件需要多长时间？

如果系统对磁盘进行整理，使逻辑上相邻的数据块的平均距离降为2磁道，这时读取一个100块的文件需要多长时间？

磁盘访问时间=寻道时间+旋转延迟时间+传输时间

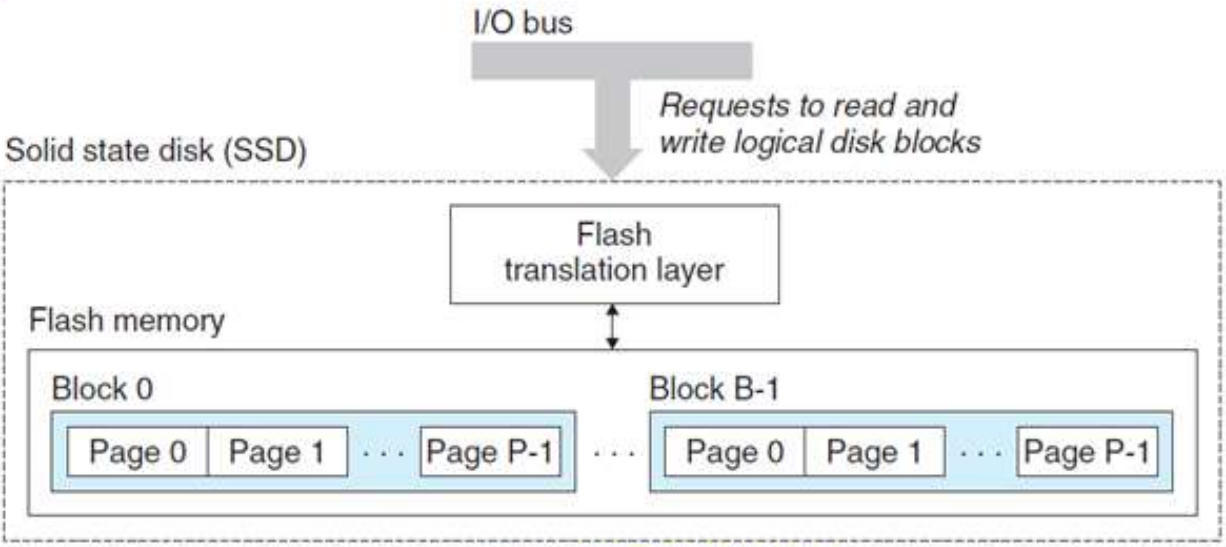
整理前： 寻道时间Ts=13*6=78ms，则Ta=78+100+25=203ms，所以读取100个块的文件需要时间为20300ms

整理后： 寻道时间Ts=2*6=12ms，则Ta=12+100+25=137ms，所以读取100个块的文件需要时间为13700ms

寻道时间和旋转延迟时间与信息在磁盘上的位置有关，可以通过相关调度来减少访问时间，而传输时间是硬件设计时就固定的。

固态硬盘(1/2)

- 结构



页的大小：512B~4KB ➡ 读写基本单位

块包含 32~128个页

块的大小：16KB~512KB

固态硬盘(2/2)

- 性能特性

表 Intel SSD 730性能特性

先擦除再写

读		写	
顺序读吞吐量	550MB/s	顺序写吞吐量	470MB/s
随机读吞吐量	365MB/s	随机写吞吐量	303MB/s
平均顺序读访问时间	50μs	平均顺序写访问时间	60μs

- SSD vs HDD
  - 优点 更可靠、更快、更节能
  - 缺点 易损耗、更昂贵

纲要

概述(1/2)

引入

- 具有多个进程的系统，磁盘队列可能有多个待处理的请求。当一个请求完成时，这时OS可以决定谁下一个访问磁盘，这种调度技术被称为磁盘调度技术，或驱动调度技术。

目标

- 通过管理磁盘I/O请求的处理次序，缩短I/O请求的平均访问时间。

移臂调度：根据访问者指定的柱面位置来决定执行次序的调度，力求减少寻道时间。

旋转调度：根据旋转延迟时间来决定执行次序的调度，力求减少旋转延迟时间。

难以实现

61

概述(2/2)

经典移臂调度算法

- 先来先服务(First-Come First Served, FCFS)调度
- 最短寻道时间优先(Shortest-Seek-Time-First , SSTF)调度
- 扫描(SCAN)调度
- 电梯调度(LOOK)
- 循环扫描(Circular SCAN, C-SCAN)调度

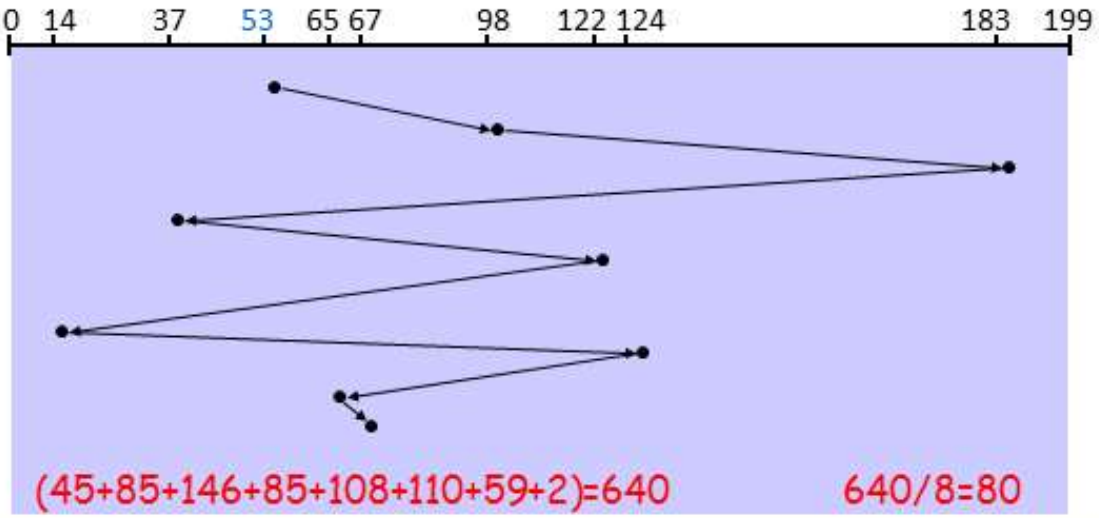
衡量标准：（平均）寻道长度

62

3.2 移臂调度策略

FCFS

- 工作原理　按请求访问者的先后次序启动磁盘驱动器，而不考虑它们要访问的物理位置。
- 举例　200个柱面，编号0~199
  IO请求队列=98、183、37、122、14、124、65、67 磁头开始于53

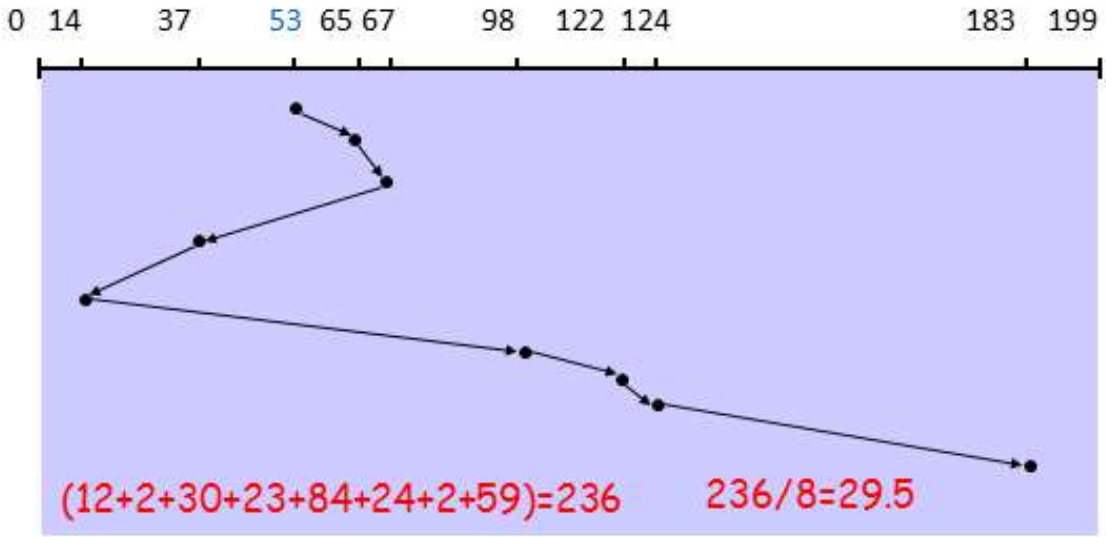


优点： 公平简单
缺点： 未对寻道进行优化，平均寻道时间较长，仅适合磁盘请求较少的场合。

3.2 移臂调度策略

SSTF

- 工作原理　选择所访问磁道与磁头当前所在磁道距离最近的I/O请求优先调度。
- 举例　IO请求队列=98、183、37、122、14、124、65、67
  磁头开始于53号柱面



优缺点： 具有较好的寻道性能，但可能导致进程饥饿现象。

64

3.2 移臂调度策略

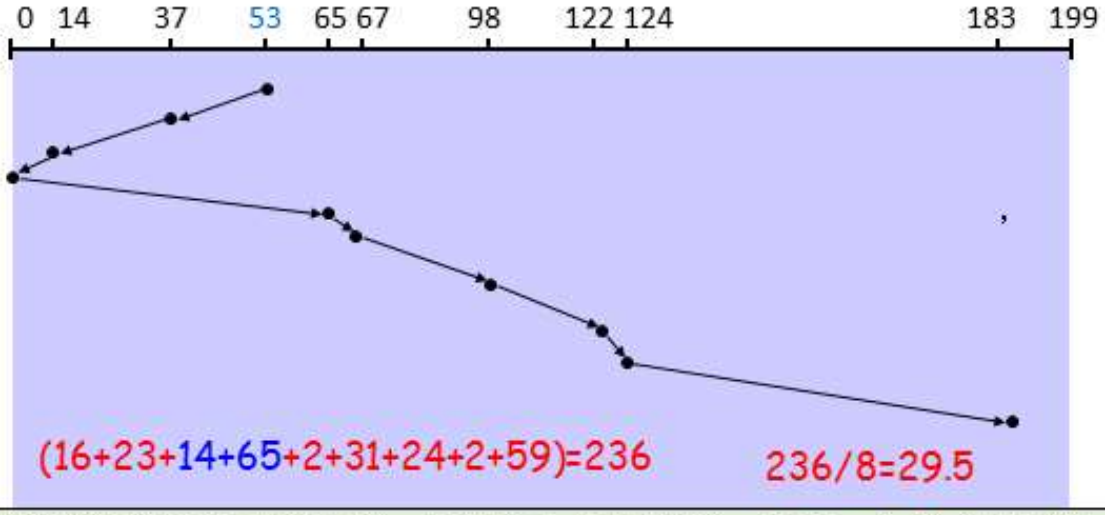
SCAN

- 工作原理

移动臂每次沿一个方向移动，扫过所有柱面，遇到最近的I/O请求便进行处理，直到磁盘的最后一个柱面后，再向相反方向移动回来。

- 举例

IO请求队列=98、183、37、122、14、124、65、67
磁头开始于53号柱面，方向是向外。



优缺点：既能获得较好的寻道性能，又能防止进程饥饿；存在一个请求刚好被错过而需要等待很长时间的问题。

3.2 移臂调度策略

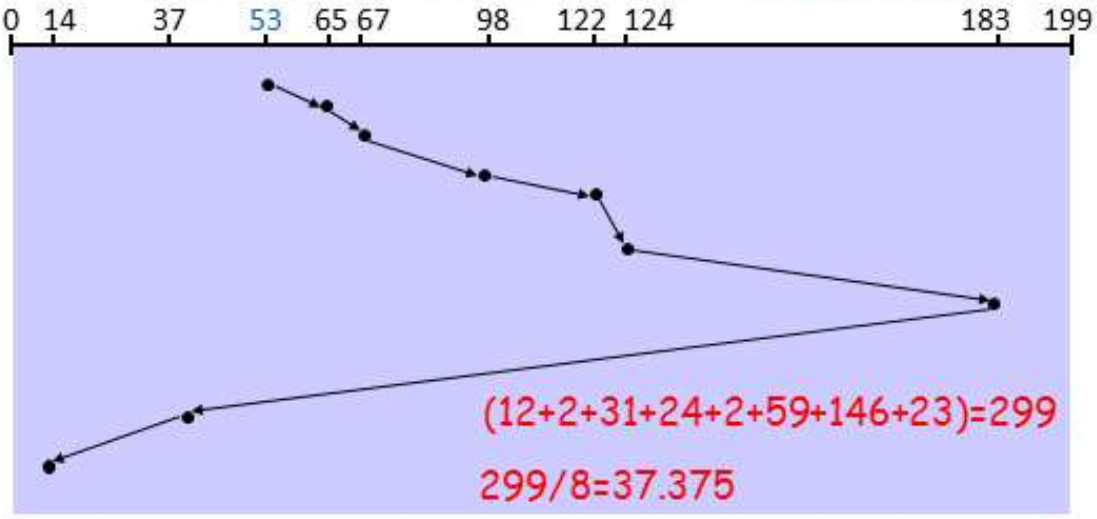
LOOK(电梯调度)

- 工作原理

电梯调度不像扫描算法那样一直扫到底，如果沿着目前的方向无请求的话则立即折返。

- 举例

IO请求队列=98、183、37、122、14、124、65、67
磁头开始于53号柱面，方向是向内. 如果方向向外呢？



3.2 移臂调度策略

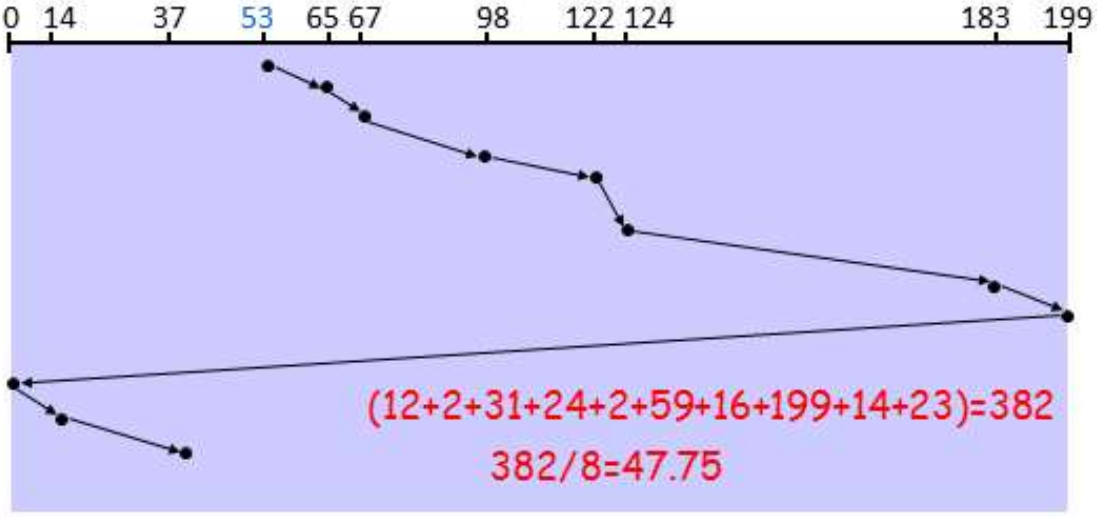
C-SCAN

- 工作原理

与SCAN相似，只是C-SCAN在扫描到最大柱面后直接返回0号柱面重复进行，归途中不再服务，构成了一个循环。

- 举例

IO请求队列 = 98、183、37、122、14、124、65、67
磁头开始于53号柱面，方向总是向内。



67

3.2 移臂调度策略

- 常见题型

例6-3 假定一磁盘有200个柱面，编号为0~199，当前存取臂的位置在143号柱面上，并刚刚完成125号柱面的服务请求，如果请求队列的先后顺序为86、147、91、177、94、150、102、175、130。则为完成上述请求：

FCFS算法移动臂移动的总量是（1），其移动臂移动的顺序是（2）

SSTF算法移动臂移动的总量是（3），其移动臂移动的顺序是（4）

SCAN算法移动臂移动的总量是（5），其移动臂移动的顺序是（6）

LOOK算法移动臂移动的总量是（7），其移动臂移动的顺序是（8）

68

常见题型

例6-3 假定一磁盘有200个柱面，编号为0~199，当前存取臂的位置在143号柱面上，并刚刚完成125号柱面的服务请求，如果请求队列的先后顺序为86、147、91、177、94、150、102、175、130。则为完成上述请求：

FCFS算法移动臂移动的总量是（1），其移动臂移动的顺序是（2）
SSTF算法移动臂移动的总量是（3），其移动臂移动的顺序是（4）
SCAN算法移动臂移动的总量是（5），其移动臂移动的顺序是（6）
LOOK算法移动臂移动的总量是（7），其移动臂移动的顺序是（8）

FCFS: 143->86->147->91->177->94->150->102->175->130 565
SSTF: 143->147->150->130->102->94->91->86->175->177 162
SCAN: 143->147->150->175->177->199->130->102->94->91->86 169
LOOK: 143->147->150->175->177->130->102->94->91->86 125

移臂调度算法的选择

- SSTF算法性能较好，但可能发生饥饿现象。
- SCAN和C-SCAN不会发生饥饿，很适合磁盘负荷较大的系统，因为此种系统各柱面的请求比较平均。
- LOOK调度是扫描算法的另一种实现。
- 通常**SSTF**和**LOOK**是比较合理的缺省算法。

因为固态硬盘没有移动部件，所以算法的性能差异很小并且经常使用简单的FCFS策略。

磁盘的I/O请求效率很大程度受文件分配方法的影响，如果程序所读的文件是连续分配的，则会产生在磁盘上相近区域的请求，从而减少磁头移动次数。

单选题 1分

CQ3.1 磁盘是可共享设备，每一时刻（　）进程启动它。

A 可以由任意多个

B 能限定多个

C 至少能由一个

D 至多能由一个

单选题 1分

CQ3.2 磁盘上的文件以（　）单位读写。

A 盘块

B 记录

C 柱面

D 磁道

单选题 1分

CQ3.3 在磁盘输入输出的操作中，需要做的工作可以不包括（　　）。

A 确定磁盘的存储容量

B 移动磁臂使磁头移到指定的柱面上

C 旋转磁盘使指定的扇区处于磁头位置下

D 让指定的磁头读写信息，完成信息的传送操作

单选题 1分

CQ3.4 启动磁盘执行一次输入输出操作时，（　）是硬件设计时就固定的。

A 寻道时间

B 延迟时间

C 传输时间

D 一次I/O操作的总时间

单选题 1分

CQ3.5 一个磁盘的转速为7200转/分，每个磁道有160个扇区，每扇区有512字节，那么理想情况下，其数据传输率为( ).

A 7200*160KB/s

B 7200KB/s

C 9600KB/s

D 19200KB/s

单选题 1分

CQ3.6 磁盘旋转调度的目的是为了缩短（ ）时间。

A 寻道

B 延迟

C 传输

D 启动

单选题 1分

CQ3.7 下列算法中，用于磁盘调度的是（ ）。

A 时间片轮转法

B LRU算法

C 最短寻道时间优先算法

D 优先级高者优先算法

单选题 1分

CQ3.8 对磁盘进行移臂调度时，既考虑了减少寻找时间，又不频繁变动臂的移动方向的调度算法是（ ）。

A 先来先服务

B 最短寻道时间优先

C 电梯调度

D 优先级高者优先

单选题 1分

CQ3.9 在以下调度中，（　　）算法可能出现饥饿现象。

A 电梯调度

B 最短寻道时间优先

C 循环扫描算法

D 先来先服务

单选题 1分

CQ3.10 如果当前磁头正在53号柱面上执行输入输出操作，依次有4个等待者分别要访问的柱面为98、37、124、65，当采用（　　）调度算法时，下一次读写磁头才可能达到37号柱面。

A 先来先服务

B 最短寻道时间优先

C 电梯调度（磁头移动方向向着小磁道方向）

D 循环扫描算法（磁头移动方向向着大磁道方向）

纲要

3.1 基础知识

3.2 移臂调度策略

3.3 独立磁盘冗余阵列

Redundant Arrays of Independent Disk, RAID

I. 概述

II. RAID常用级别

概述(1/2)

	register	cache	memory	SSD	HDD
Access time (ns)	0.25-0.5	0.5-25	80-250	25,000-50,000	5,000,000

引入

- 磁盘存储系统是提高整个计算机系统性能的关键。
- 如果使用一个组件对性能的影响有限，可以并行使用多个组件获得额外的性能提高。

实现思想

- RAID方案用多个小容量磁盘驱动器代替大容量磁盘驱动器，并以能同时从多个驱动器访问数据的方式来分布数据，因此提高了I/O性能，并能更容易地增加容量。

3.3 独立磁盘冗余阵列

概述(2/2)

- 目标及实现技术
  - **通过并行处理提高性能** 高数据传输率

    将数据分条后分散的存储在多个磁盘上，以提高存储系统的吞吐量。

    位级分条(bit-level striping): 每个字节的位i存在磁盘i上

    块级分条(block-level striping): 每个文件的块i存在磁盘i上
  - **通过冗余提高可靠性** 高可靠性

    加入冗余数据以提供较为完备的相互校验/恢复的措施。

    镜像(mirroring): 重复每个磁盘

    奇偶校验(parity): 加入奇偶校验信息，保证在一个磁盘失效时，数据具有可恢复性

XOR校验原理（位级分条为例）：对于存储在多个磁盘上的某字节，计算其纠错位P并存放在其他磁盘上。如果一个磁盘故障，则可从其他磁盘中读取字节的其余位和相关的奇偶校验位，以重构损坏的数据。

A值	B值	XOR结果
0	0	0
1	0	1
0	1	1
1	1	0

3.3 独立磁盘冗余阵列

RAID常用级别(1/5)

C：数据的备份

P：纠错位

RAID0: 非冗余条带

(a) RAID 0: non-redundant striping.

RAID1: 镜像磁盘

(b) RAID 1: mirrored disks.

RAID2: 内存式纠错码

(c) RAID 2: memory-style error-correcting codes.

RAID3:位交错奇偶校验

(d) RAID 3: bit-interleaved parity.

RAID4: 块交错奇偶校验

(e) RAID 4: block-interleaved parity.

RAID5: 块交错分布式奇偶校验

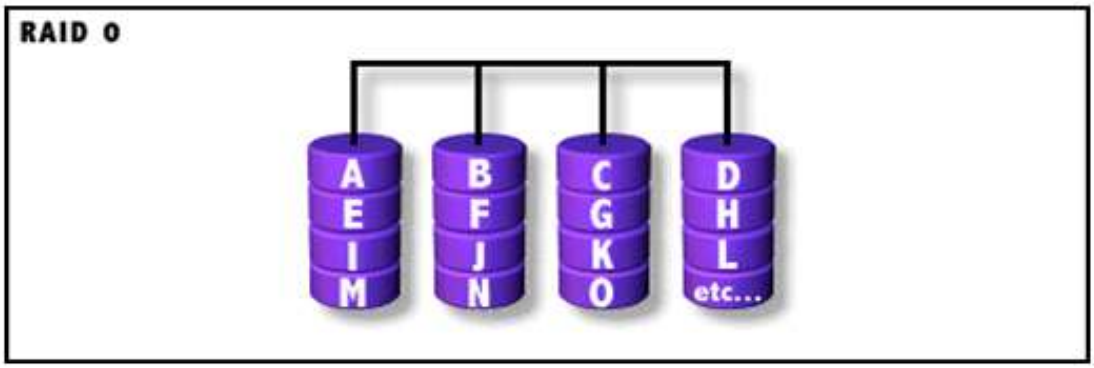
(f) RAID 5: block-interleaved distributed parity.

RAID6: P+Q 双重冗余

(g) RAID 6: P + Q redundancy.

RAID常用级别(2/5)

- RAID 0 无冗余的磁盘阵列

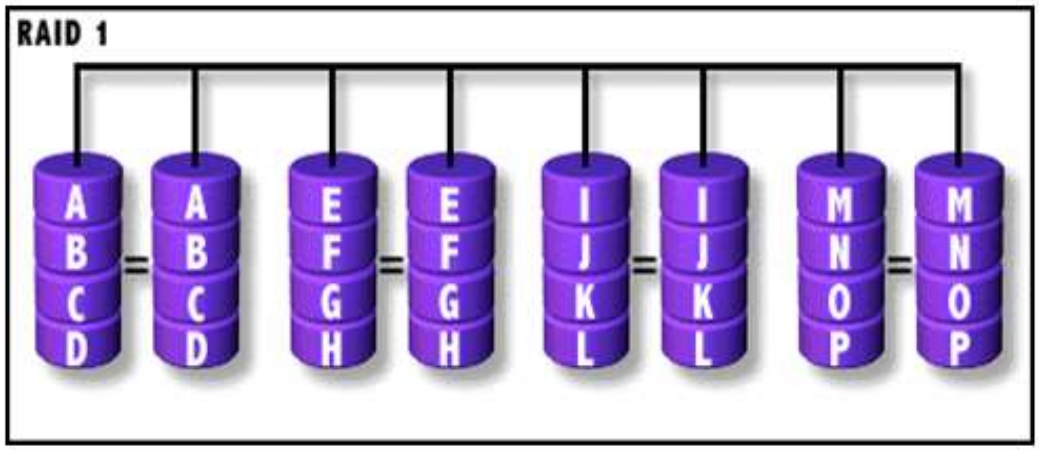


- 在存储数据时由RAID控制器将数据分割成大小相同的数据条，同时写入阵列中的磁盘。
- 数据并行的写入和读取，提升了存储系统的性能。
- 无容错能力，系统可靠性较差，适用于数据损失并不重要的高性能应用程序。

85

RAID常用级别(3/5)

- RAID 1 磁盘镜像

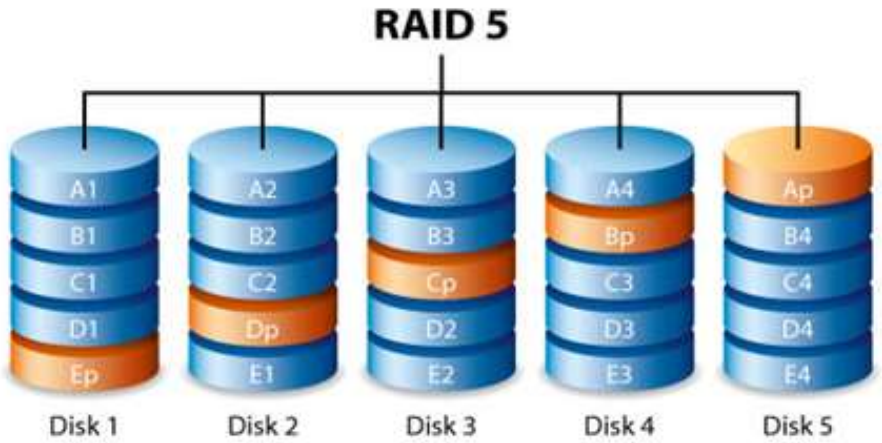


- 两个硬盘的内容完全一样，内容彼此备份。
- 系统可靠性好，重建简单，但硬件开销大，效率低。
- 适用于需要高可靠性和快速恢复的应用程序。

86

RAID常用级别(4/5)

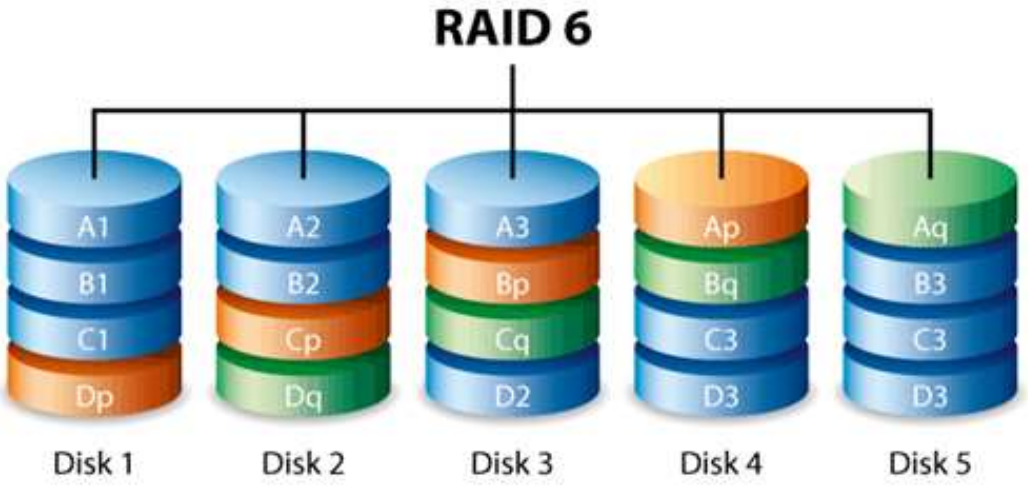
- RAID 5 块交错分布式奇偶结构 最常见的奇偶校验RAID系统



- 对于每个数据磁盘中相应的条带逐位计算一个奇偶检验(parity code)，奇偶校验条带分布在所有驱动器上。Ap=A1⊕A2 ⊕A3 ⊕A4
- 数据以块为单位和奇偶校验码分散存储在N+1个硬盘上，每当写操作时，阵列管理软件不仅须更新用户数据，而且须更新相应的奇偶校验位。

RAID常用级别(5/5)

- RAID 6 P+Q双重冗余



- 进一步提升容错性，同时使用奇偶校验码和差错纠正码(Reed-Solomon code)，并保存在不同磁盘的不同块中，防范多个磁盘故障。
- 每次写操作都会影响两个校验块，导致严重的写性能损失。

单选题 1分

CQ3.11 在磁盘冗余阵列中，以下不具有容错技术的是（ ）。

A RAID0

B RAID1

C RAID3

D RAID5

Have A Nice Day

南京邮电大学计算机学院、
软件学院、网络空间安全学院