



第7章 设备管理

7.1 设备管理概述

7.2 I/O控制方式

7.3 I/O缓冲

7.4 设备分配与设备处理

7.5 I/O管理中的几个重要思想

7.6 磁盘I/O

7.7 小结



7.1 设备管理概述

一、I/O设备的分类

1. 按从属关系分类

(1) 系统设备

指在操作系统生成时已经登记在系统中的标准设备。

如键盘、显示器、打印机等。

(2) 用户设备

指操作系统生成时未登记入系统的非标准设备。如绘图仪、扫描仪等。



I/O 设备分类

2. 按传输速率分类

(1) 低速设备

指传输速率为每秒钟几个字符至数百个字节的设备

如键盘、鼠标、语音输入等。

(2) 中速设备

指传输速率为每秒钟数千个字节至数万个字节的设备

如针式打印机、激光打印机等。

(3) 高速设备

指传输速率为数兆字节的设备，如磁带机、磁盘机、光盘机等。



I/O 设备分类

3. 按使用特性分类

(1) 存储设备

是计算机用来保存各种数据的设备，如磁盘、磁带等。

(2) I/O设备

是向CPU传输数据或输出CPU加工处理数据的设备。 例如：键盘，CRT



I/O 设备分类

4. 按共享属性分类

(1) 独占设备

指在**一段时间内**只允许一个用户（进程）访问的设备。

也就是在某个用户（进程）对设备的一次使用过程（包含多次I/O操作）中，不允许其他用户（进程）使用该设备。一般是低速的I/O设备,如打印机等。

独占设备属于临界资源，多个并发进程必须互斥访问独占设备。



按设备共享属性分类

(2) 共享设备

指在一段时间内允许多个进程同时访问的设备，
多个进程以交叉的方式来使用设备，其资源利用率高，如硬盘。

(3) 虚拟设备

指通过虚拟技术将一台独占设备变换为若干台供多个用户（进程）共享的逻辑设备。一般可以利用假脱机（SPOOLing）技术实现虚拟设备。



I/O 设备分类

5. 按数据传送的基本单位分类

(1) 字符设备

以字节为单位传送数据，如键盘。
传输速度慢。

(2) 块设备

以块（如512B）为单位传送数据，如磁盘。
传输速度快。

是**可寻址**的和**可随机访问**的设备



7.1 设备管理概述

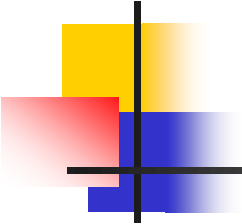
二、设备管理的2个重要目标：效率、通用性

(1) 效率

提高设备的利用率和I/O效率

充分利用各种技术（中断、DMA、通道、缓冲等）提高CPU与设备、设备与设备之间的并行工作能力，充分利用资源，提高资源利用率，提高I/O处理的效率

- ✓ 并行性
- ✓ 均衡性（使设备充分忙碌）



设备管理的2个重要目标

(2) 通用性

为用户提供方便、统一的接口，希望能**用统一的方式处理所有设备**
屏蔽硬件细节（设备的物理细节，错误处理，不同I/O的差异性）
使用户摆脱繁琐的程序设计负担

- ✓ 方便性
- ✓ 接口的友好性
- ✓ **透明性**



7.1 设备管理概述

三、设备管理的功能

(1) 设备分配

按照设备类型和相应的分配算法决定将I/O设备分配给哪一个要求使用该设备的进程。凡未分配到所需设备的进程被放入一个等待队列。

(2) 设备处理

设备处理程序实现CPU和设备控制器之间的通信。即当CPU向设备控制器发出I/O指令时，设备处理程序应启动设备进行I/O操作，并能对设备发来的中断请求作出及时的响应和处理。

(3) 实现其他功能

包括对缓冲区的管理功能以及实现设备独立性。



7.2 I/O 控制方式

I/O控制方式：主机和I/O设备之间的数据传送方式

4种I/O控制方式:

(1) 程序直接控制方式 (轮询)

(2) 中断方式

(3) DMA方式

(4) 通道方式

发展的思路：解放CPU



7.2 I/O 控制方式

1. 程序直接控制方式（轮询）

由用户进程控制，不断测试设备状态

缺点：

- ✓ 忙等待
- ✓ CPU与I/O设备只能串行工作



7.2 I/O 控制方式

2. 中断方式

为了减少设备驱动程序不断询问设备控制器中状态寄存器的开销

当I/O操作结束后，由设备控制器主动通知设备驱动程序

不足：

- ✓ 数据传送是在中断处理时由CPU控制完成
- ✓ 每次传输的数据量小，比如1个字节或1个字。尽管，可以通过在I/O控制器中设置字符缓冲区而增大每次的数据传输量
- ✓ 可能造成数据由于CPU来不及取而丢失（当外设速度快时）



7.2 I/O 控制方式

3. DMA方式

- ✓ 在内存与I/O设备之间直接传送数据块
- ✓ CPU在开始时向设备发“传送一块”命令，数据传送由DMA控制器控制完成，每次1个数据块
- ✓ 传送结束时，由DMA控制器给CPU发送一个中断信号
- ✓ DMA的功能可以以独立的DMA部件在系统I/O总线上完成，或者整合到I/O部件中完成



DMA 方式

DMA方式与中断的主要区别：

- ✓ 中断方式是在数据缓冲寄存器满后，发中断请求，CPU进行中断处理
DMA方式则是在所要求传送的数据块全部传送结束时要求CPU进行中断处理，大大减少了CPU进行中断处理的次数
- ✓ 中断方式的数据传送是由CPU控制完成的
DMA方式则是在DMA控制器的控制下不经过CPU控制完成的



7.2 I/O 控制方式

4. 通道方式

通道：独立于CPU的专门负责数据输入/输出传输工作的处理机，对外部设备实现统一管理，代替CPU对输入/输出操作进行控制，从而使输入/输出可与CPU并行操作。

引入通道的目的：

为了使CPU从I/O操作中解脱出来，同时为了提高CPU与设备、设备与设备之间的并行工作能力

数据传送的方向、长度、内存地址等都由通道控制。



7.3 I/O 缓冲

一、什么是I/O缓冲？

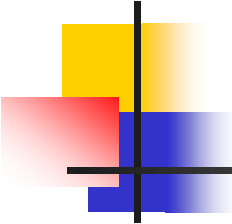
缓冲 (buffering) :

为了缓解通信双方速度不匹配而引入的一个中间环节。

I/O缓冲 :

在CPU和I/O设备之间设立缓冲区，用以暂存CPU与外设之间交换的数据，从而缓和CPU与外设速度不匹配所产生的矛盾。

其实，凡是数据到达和离去速度不匹配的地方均可采用缓冲技术。



7.3 I/O 缓冲

引入I/O缓冲的目的：

- (1) 改善CPU与I/O设备之间速度不匹配的矛盾
- (2) **减少对CPU的中断频率**，放宽对I/O中断响应时间的限制
- (3) **减少访问I/O设备（如磁盘）的次数**
- (4) 提高CPU与I/O设备之间的并行性



7.3 I/O 缓冲

I/O缓冲的实现方式：

(1) 硬件缓冲

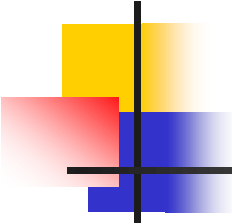
I/O设备或控制器内部设置的纯硬件缓冲区

(2) 软件缓冲

为I/O在内存开辟的缓冲区，由软件来管理

容量大，使用灵活

下面要讲的是OS采用的I/O缓冲技术。



7.3 I/O 缓冲

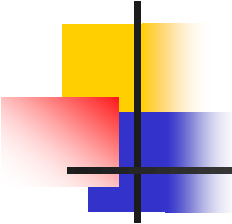
二、OS为什么要引入I/O缓冲？

【例】 应用程序要从磁盘中读一块数据（512B）到自己的地址空间1000-1511(虚拟地址)。在执行I/O命令后阻塞，等待数据变成可用的。

存在的问题：

该进程的虚拟地址单元1000-1511必须锁定在内存中，不能换出，干扰了OS的交换决策；

否则有可能造成单进程死锁。即进程发出I/O命令后阻塞，若开始I/O操作之前被换出，则I/O操作也会阻塞，以等待该进程被换入。



7.3 I/O 缓冲

三、I/O缓冲的种类

1. 单缓冲

当用户进程发出I/O请求时，操作系统在内存的系统空间为该操作分配一个缓冲区

输入：数据输入到系统缓冲区，传送完成时，用户进程将其复制到用户空间

输出：用户进程将数据复制到系统缓冲区，然后由OS输出。

如果连续输入/输出多个数据块呢？



I/O 缓冲的种类

2. 双缓冲

由OS指定2个缓冲区

当用户进程从一个缓冲区取数据（或向一个缓冲区写数据）时，
OS可以向另一个缓冲区输入数据（或输出另一个缓冲区的数据）。

3. 缓冲池

又称循环缓冲

多个缓冲区构成循环队列

类似于生产者/消费者问题



7.4 设备分配与设备处理

一、设备分配

当某进程向系统提出I/O请求时，设备分配程序按一定策略分配设备、控制器和通道，形成一条数据传输通路，以供主机和设备间信息交换。



设备分配

1. 与设备分配有关的数据结构

✓ 设备类表

系统中拥有1张设备类表

每类设备对应于表中一栏，包括内容有：

设备类、总台数、空闲台数、设备驱动程序入口和设备表起始地址等。

✓ 设备表

每一类设备都有各自的设备表，用来登记这类设备中每一台设备的状态，

包含的内容有：物理设备名、逻辑设备名、占有设备的进程号、已分配/未分配、好/坏等。



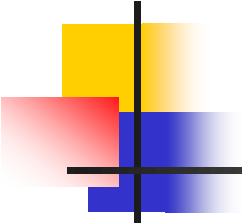
与设备分配有关的数据结构

采用通道结构的系统中设备分配采用的数据结构：

系统设备表、通道控制表、控制器控制表和设备控制表

- ✓ 系统建立1张系统设备表，记录配置在系统中的所有物理设备的情况。
- ✓ 每个通道、控制器、设备各设置一张表，记录各自的地址(标识符)、状态(忙/闲)、等待获得此部件的进程队列指针、及一次分配后相互链接的指针，以备分配和执行I/O时使用。

具体内容如下：



与设备分配有关的数据结构

- ✓ 设备控制块DCB (设备控制表DCT)

记录本设备的使用情况。主要内容：设备类型、设备标识符、设备状态、与此设备相连的COCT、重复执行的次数或时间、等待队列的队首和队尾指针

- ✓ 控制器控制块COCB (控制器控制表COCT)

- ✓ 通道控制块CHCB (通道控制表CHCT)

- ✓ 系统设备表SDT

整个系统一张表，记录系统中所有I/O设备的信息，表目包括：设备类型、设备标识符、设备驱动程序入口、DCT表指针等，是分配程序首先查找的数据结构。



设备分配

2. 设备分配策略

由于在多道程序系统中，进程数多于资源数，引起资源的竞争。
因此，要有一套合理的分配策略。

考虑的因素：

- ✓ I/O设备的固有属性
- ✓ I/O设备使用场合的目标需求：
 - 设备分配的安全性
 - 与设备的无关性（即设备独立性）
 - 设备利用率
 - 公平性
 - 请求设备的进程优先级



设备分配策略

(1) 独占设备的分配

所谓独占式共享使用设备是**以一次设备使用过程（包含多次I/O操作）为单位使用设备**

在申请设备时，如果设备空闲，就将其独占，不再允许其他进程申请使用，一直等到该设备被释放，才允许其他进程申请使用

考虑效率问题，并且避免由于不合理的分配策略造成死锁

静态分配：在进程运行前，完成设备分配；运行结束时，收回设备

缺点：设备利用率低

动态分配：在进程运行过程中，当用户提出设备请求时，进行分配，一旦停止使用立即收回

优点：设备利用率高；缺点：分配策略不好时，产生死锁



设备分配策略

(2) 分时式共享设备的分配

所谓分时式共享就是**以一次I/O操作为单位分时使用设备，不同进程的I/O操作请求以排队方式分时地占用设备进行I/O**

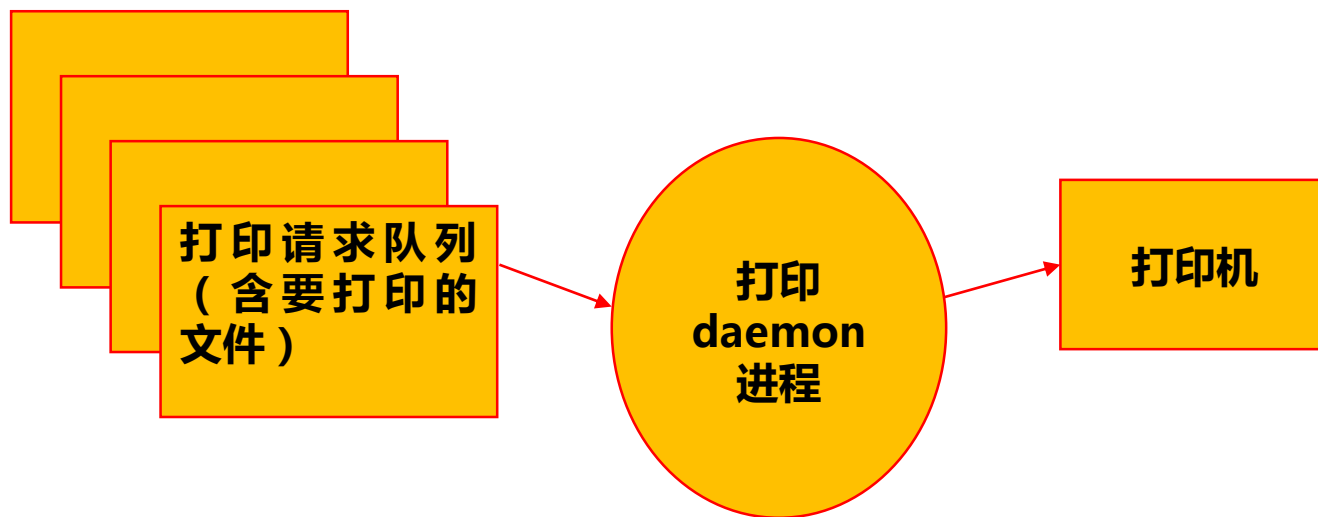
由于同时有多个进程同时访问，且访问频繁，因此要考虑多个访问请求到达时服务的顺序，使平均服务时间越短越好

设备分配策略

(3) 以SPOOLing方式使用外设

SPOOLing 技术是在批处理操作系统时代引入的，即假脱机输入/输出技术。

例如：所有输出数据已经写到文件中，并排到打印输出队列，打印进程申请占用打印机后，成批读出文件中的数据，并送打印机打印





设备分配策略

常用的I/O设备分配算法：

- ✓ 先来先服务
- ✓ 优先级高者优先

设备请求队列：

当多个进程对同一设备提出I/O请求时，系统响应后，为它们分别建立I/O请求包，按先来先服务或者优先级高者优先的原则组织成设备请求队列。

设备分配程序总是把设备首先分配给队首进程。具体分配是从设备类表或者系统设备表开始顺序查找相应的数据结构进行的。



7.4 设备分配与设备处理

二、设备处理

1. 设备驱动程序

每类设备对应1个设备驱动程序，以控制I/O传输

任务：

主要负责接收和分析从设备分配转来的信息，把用户I/O请求转换为具体要求后，发送给设备控制器，启动设备执行。



设备驱动程序

设备驱动程序的处理过程：

- (1) 将抽象I/O请求转换为具体的操作参数
- (2) 检查I/O请求的合法性
- (3) 读出和检查设备的状态
- (4) 传送必要的参数，预置设备的初始状态
- (5) 设置设备的工作方式（在有通道的系统中，构造通道程序）
- (6) 启动设备进行I/O操作
- (7) 响应来自设备的中断

2. I/O中断处理程序

处理来自设备或通道的中断

包括正常结束，或异常结束



7.5 I/O 管理中的几个重要思想

一、设备独立性 (Device Independence)

设备独立性是I/O软件设计中的一个重要目标。

设备独立性的含义：

用户在编写程序时，能独立于具体使用的物理设备，甚至不关心设备类型。

- ① 应用程序与给定设备类型的哪一台具体设备无关；
- ② 应用程序尽可能地与设备类型无关。



设备独立性

设备独立性的实现方法：

(1) 引入逻辑设备

由OS管理一个逻辑设备映射表，记录逻辑设备对应的物理设备。

用户程序对I/O设备的请求不指定特定的设备，而采用逻辑设备名，程序执行时由OS完成逻辑设备到物理设备的映射

例如：用户申请使用设备时，只需要指定设备类型，而无须指定具体物理设备，系统根据当前的请求及设备分配情况，在相同类别设备中，选择一个空闲设备，并将其分配给一个申请进程。

(2) 统一命名，统一接口

对不同的设备采取统一的操作方式

例如：把设备看作文件，所有设备和文件使用相同的方式（路径名定位）



7.5 I/O 管理中的几个重要思想

二、SPOOLing技术

SPOOLing:

Simultaneous Peripheral Operation On Line

即外围设备同时联机操作，又称假脱机操作

SPOOLing的概念最早出现在作业处理中，那时多道程序的概念还没有提出
这个概念至今还有很大的意义



SPooling 技术

1. 脱机输入/输出 (Offline I/O)

脱机I/O：I/O是脱离主机的。

目的：为了解决CPU和I/O设备的速度不匹配。

输入：由一台低档计算机（外围机）将作业输入到磁盘（磁带）；

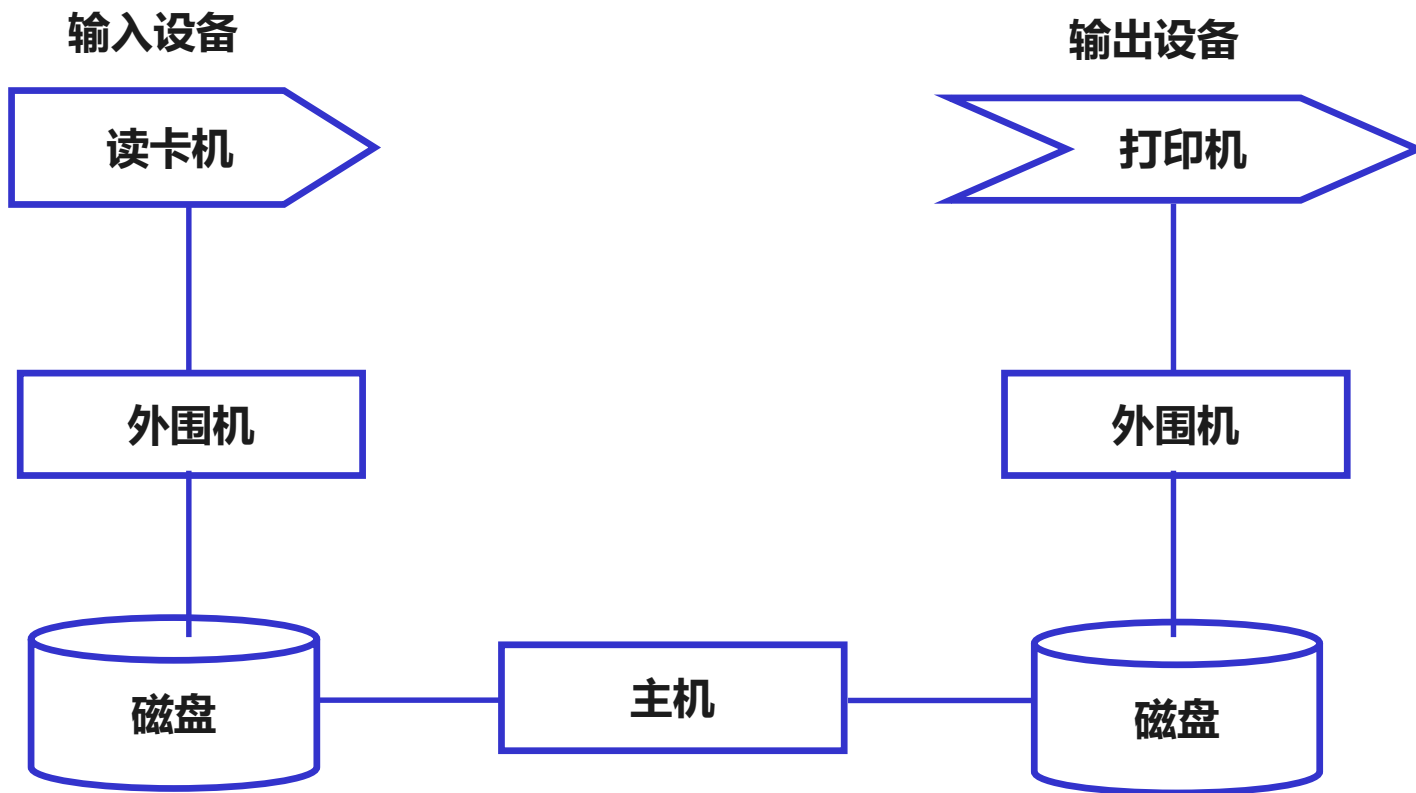
CPU需要时，从磁盘读入内存。

输出：CPU需要输出时，将数据从内存送到磁盘；

由一台外围机将磁盘中的数据输出。

SPooling 技术

脱机I/O示意图：





SPOOLing 技术

2. 什么是SPOOLing?

假脱机：

联机情况下，即在主机的控制下，模拟脱机I/O。

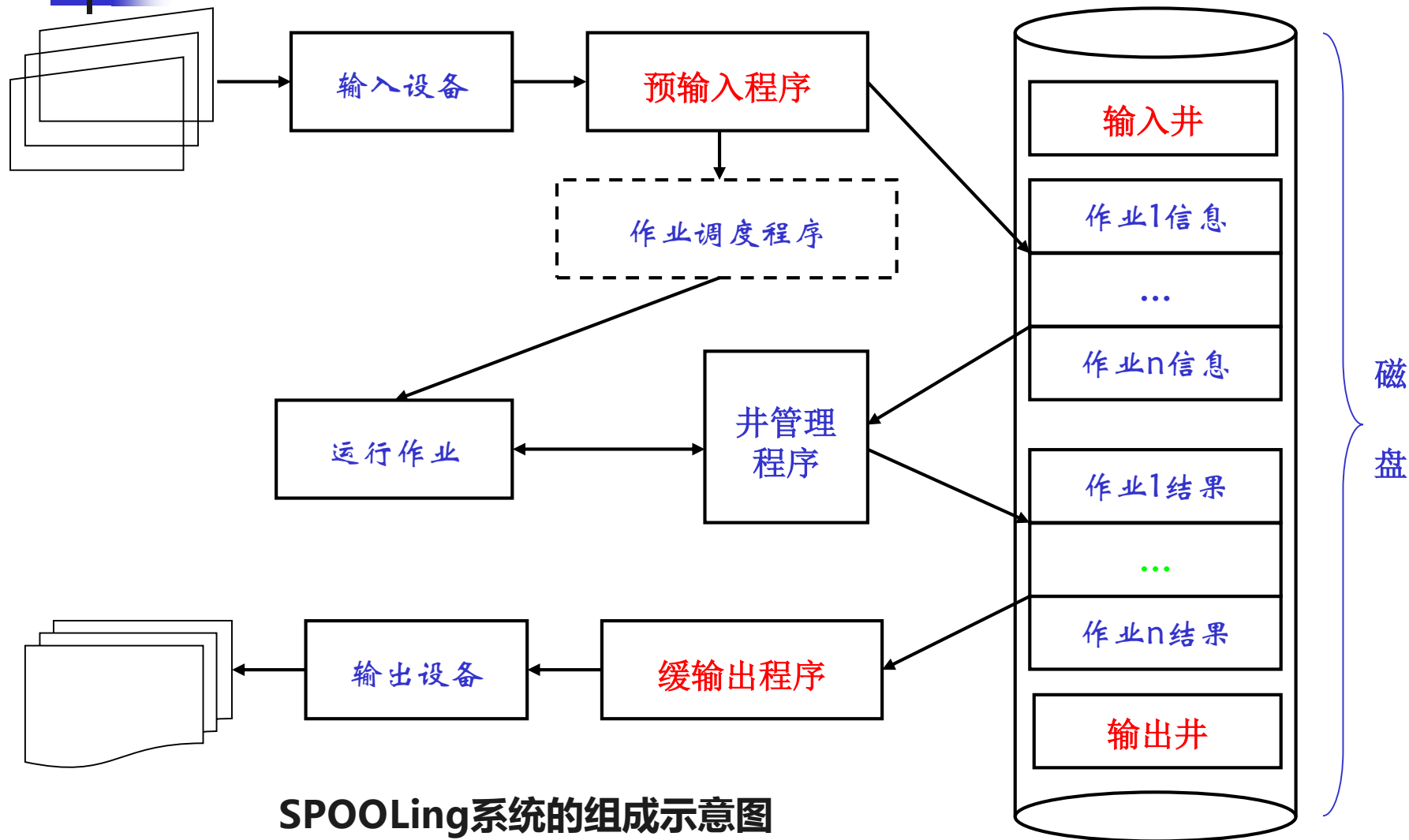
实现方法：

OS利用2个进程分别模拟脱机I/O时外围机的功能：

其中一个进程负责将输入设备的数据传送到磁盘；

另一个进程负责将数据从磁盘传送到输出设备。

SPOOLing 技术





SPOOLing 技术

SPOOLing系统的组成：

(1) 输入井和输出井

在磁盘上开辟出来的两个专用的存储区域。

“井”是用作缓冲的存储区域。输入井和输出井分别用于收容从输入设备输入的数据和用户程序的输出数据。

输入井和输出井可分别看作是对输入设备（如读卡机）和输出设备（如打印机）的虚拟或者模拟。



SPooling系统的组成

(2) 预输入进程和缓输出进程

预输入进程模拟脱机输入时的外围控制机，将用户要求输入的数据从输入设备通过输入缓冲区再送到输入井。当CPU需要输入数据时，直接从输入井读入内存。

缓输出进程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。



SPOOLing 技术

3. SPOOLing技术的基本思想

是一种虚拟技术

是OS协调并发I/O（主要用于输出）的一种技术

用来**把一台独占设备改造成为可共享的虚拟设备**，使得每个进程都以为是独占一台设备



SPOOLing 技术

SPOOLing技术的应用：

打印机的共享：

实际上系统并没有很多打印机，**只不过是磁盘的一个存储区**

实现方法：

- 1) 创建一个守护进程 (daemon)、一个打印目录spooling；
- 2) 某个进程要打印文件时，首先生成要打印的文件，将文件放入spooling目录下，形成一个请求打印队列；
- 3) 统一由daemon负责在打印机空闲时，按队列的先后次序打印spooling目录里的文件。



SPOOLing 技术

SPOOLing技术现在仍被广泛使用。

- ✓ 网络文件传送

先把文件送到网络spooling目录，由网络守护进程把它取出并传递到目标地址

- ✓ Internet电子邮件系统

在Internet上发Email时，电子邮件发送程序send先将待发信件存入spooling
电子邮件目录下，供以后传输。



7.5 I/O 管理中的几个重要思想

三、I/O软件的多层模型

I/O软件按分层的思想构成

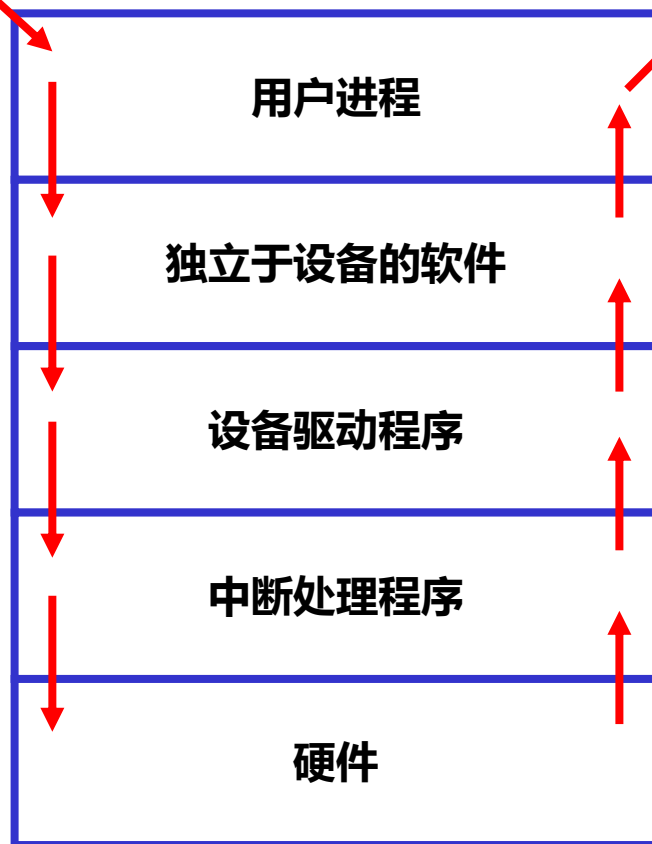
较低层软件要使较高层软件独立于硬件的特性

较高层软件则要向用户提供一个友好、清晰、简单、功能更强的接口

I/O软件的多层模型

I/O请求

I/O应答



用户进程

进行I/O调用

独立于设备的软件

提供与下层的统一接口：命名、
分配、缓冲、保护

设备驱动程序

与设备有关的代码：向设备
控制器发命令

中断处理程序

硬件



I/O 软件的多层模型

1. I/O软件各层的功能

(1) 用户进程层

执行I/O系统调用，对I/O数据进行格式化

(2) 独立于设备的软件

实现设备的命名、设备的分配、数据的缓冲、设备的保护，提供与下层的统一接口

(3) 设备驱动程序

与设备有关的代码。向设备控制器发命令，检查设备的执行状态

(4) 中断处理程序

负责I/O完成时，唤醒设备驱动程序进程，进行中断处理

(5) 硬件层（设备控制器，设备）

实现物理I/O的操作



I/O 软件的多层模型

2. 中断处理程序

- ✓ 每个进程在启动一个I/O操作后阻塞
- ✓ 直到I/O操作完成并产生一个中断
- ✓ 由操作系统接管CPU后唤醒该进程为止



I/O 软件的多层模型

3. 设备驱动程序

- ✓ 与设备密切相关的代码放在设备驱动程序中，每个设备驱动程序处理一种设备类型
- ✓ 每个设备控制器都设有一个或多个设备寄存器，用来存放向设备发送的命令和参数。设备驱动程序负责发出这些命令，并监督它们正确执行
- ✓ 一般地，设备驱动程序的任务是接收来自与设备无关的上层软件的抽象请求，并执行这个请求
- ✓ 在设备驱动程序的进程发出一条或多条命令后，系统有两种处理方式，多数情况下，执行设备驱动程序的进程必须等待命令完成，这样，在命令开始执行后，它阻塞自己，直到中断处理时将它解除阻塞为止。而在其它情况下，命令执行不必延迟就很快完成



I/O 软件的多层模型

4. 独立于设备的软件

- (1) 设备驱动程序与独立于设备的软件之间的确切界限是依赖于具体系统的
- (2) 独立于设备的软件的基本任务是实现所有设备都需要的功能，并且向用户级软件提供一个统一的接口
- (3) 如何给文件和设备这样的对象命名是操作系统要考虑的一个重要问题。
独立于设备的软件负责把设备的符号名映射到正确的设备驱动程序上
- (4) 设备保护。防止无权存取设备的用户存取设备



设备独立软件

- (5) 不同的磁盘可以采用不同的扇区尺寸。向较高层软件隐藏这一事实并提供大小统一的块尺寸，这正是独立于设备的软件的一个任务。它可将若干扇区合成一个逻辑块。这样，较高层的软件只与抽象设备打交道，独立于物理扇区的尺寸而使用等长的逻辑块
- (6) 缓冲
- (7) 设备分配
- (8) 出错处理



I/O 软件的多层模型

5. 用户空间的I/O软件

尽管大部分I / O软件都包含在操作系统中，但仍有一小部分是由与用户程序连接在一起的库过程、甚至完全由运行于核心外的程序构成。系统调用（包括I/O系统调用）通常由库过程进入

这些过程所做的工作只是将系统调用时所用的参数放在合适的位置，由其它的I/O过程实际实现真正的操作



I/O 软件的多层模型

说明：

各层之间的接口并不是死的，分层并不一定是严格的

上层中的某些功能可能放在下层中完成，如中断时的驱动以及驱动层中的某些与设备无关的处理等。



一个典型的读I/O设备的过程

①发出系统调用Read

用户进程

⑩当该用户进程被调度运行时，完成系统调用，继续执行

②核心的系统调用代码检查参数的正确性。若数据在缓冲区可得到，则转⑨

③向相应的设备驱动程序发送请求

独立于设备的软件

⑨将数据传送到请求进程的地址空间，并将该进程从阻塞队列移入就绪队列

④设备驱动程序分配一个内核缓冲区，向设备控制器发送命令，阻塞

设备驱动程序

⑧设备驱动程序确定是哪个I/O完成，确定请求的状态，通知独立于设备的软件“I/O请求已完成”

中断处理程序

⑦中断服务程序将数据存入设备驱动程序的缓冲区，唤醒阻塞的设备驱动程序,中断返回

⑤控制器操作硬件设备执行数据传输

设备控制器

⑥当I/O完成时，产生中断。假定传输由DMA控制器管理。当然，也许是由设备驱动程序轮询



7.6 磁 盘 I/O

一、影响磁盘I/O操作性能的几个因素

1. 磁盘的结构

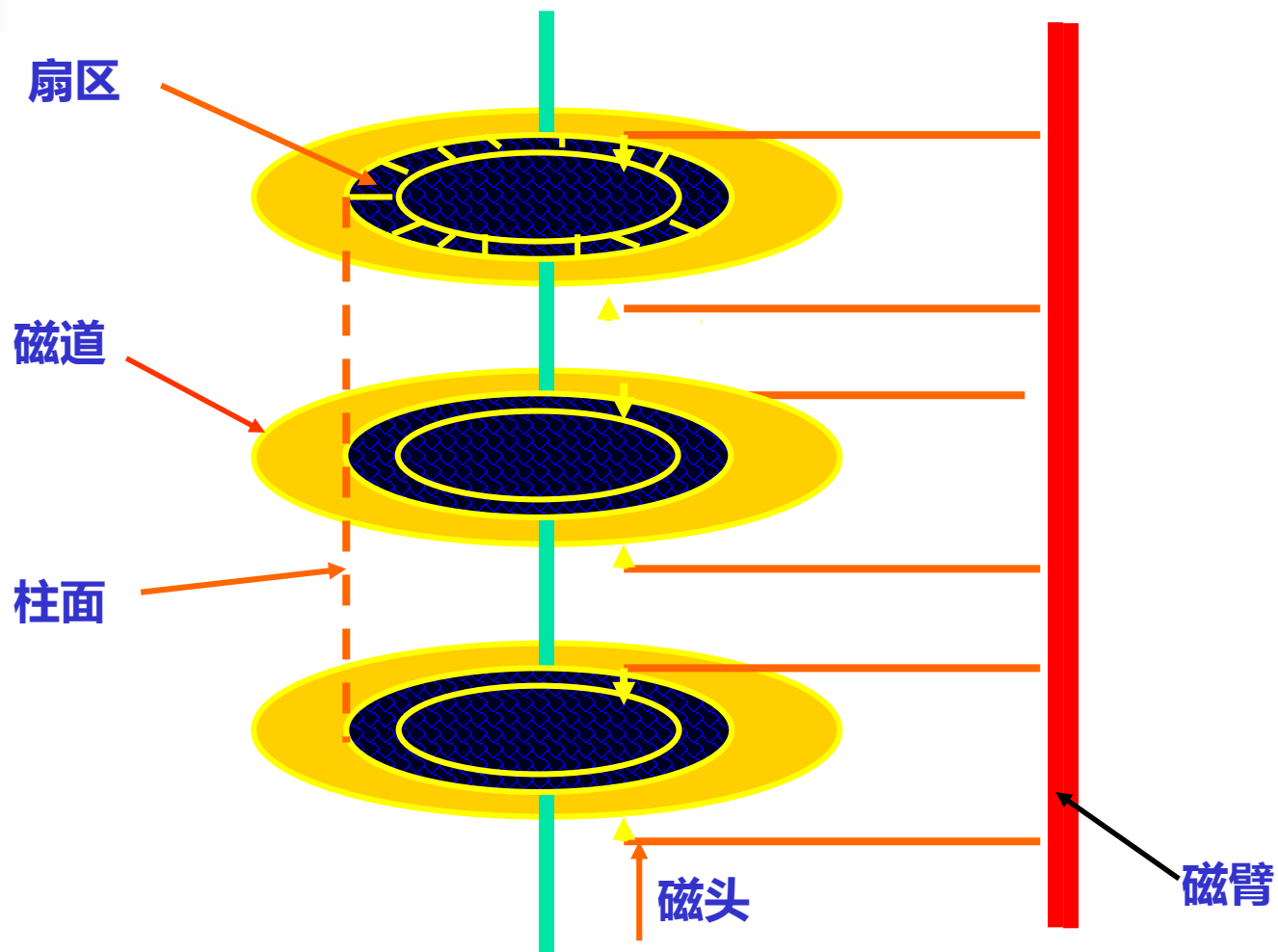
磁盘一般分为固定头磁盘和移动头磁盘两大类：

固定头磁盘：每个盘面的每条磁道都有一个读/写磁头。固定头磁盘各磁头可并行读写，但成本较高，主要用在大型机中。

移动头磁盘：每个盘面只有一个读/写磁头。每次读写须先移动磁头到目标磁道上，这称为**寻道 (seek) 操作**。磁头装在同一个架子（磁臂）上，在磁盘一致的方向上径向移动，不能单独移动。

个人计算机中的硬盘（Winchester盘）和软盘一般都是移动头磁盘。软盘由单盘片组成，硬盘则是个盘片组（因为单一盘片的容量越来越大，故硬盘所含的盘片数有逐渐减少的趋势）。盘片安装在一个高速旋转的枢轴上。读写头安装在移动臂上，移动臂可沿磁盘半径方向移动。

磁盘的结构



移动头硬盘驱动器结构示意图



磁盘的结构

磁盘的结构：

包含1个或多个盘片

每片分2面

每面分为若干磁道（同心环）

每个磁道分为若干扇区（Sector）

数据存储在磁道上

所有盘片的每面上相同位置的磁道称为一个柱面(cylinder)

每个盘块的地址由柱面号、磁头号 and 扇区号表示



7.6 磁 盘 I/O

2. 磁盘访问时间

(1) 寻道时间 (Seek Time) T_s

把磁臂 (磁头) 径向移动到指定磁道或柱面上所经历的时间，包含启动磁臂和磁头移动n条磁道或柱面所花费的时间。

(2) 旋转延迟 (Rotational Delay) 时间 T_r

指定扇区旋转到达磁头下面所经历的时间。与盘面的旋转速度有关。

一旦选择好磁道，磁盘控制器开始等待，直到相应扇区旋转到磁头处。
然后，开始执行读/写操作。

(3) 传输 (Transfer) 时间 T_t

把数据从磁盘读出或向磁盘写入数据所经历的时间。

与旋转速度和一次读写的数据量有关。



磁盘访问时间

磁盘访问时间 T_a 可表示为

$$T_a = T_s + T_r + T_t$$

$$\text{平均值} = T_s + 1/(2r) + b/(rN)$$

其中， r ：磁盘的旋转速度（转/秒）

N ：一个磁道中的 字节数

b ：要读/写的字节数



7.6 磁 盘 I/O

二、磁盘调度算法

目标：使磁盘的平均寻道时间最短。

当磁臂为一个请求寻道时，其他进程会产生其他磁盘请求

磁盘驱动程序维护一个请求队列，按柱面号索引，每个柱面的待处理请求组成一个链表。

1. 先来先服务 (First Come First Served , FCFS)

按请求访问磁盘的先后次序进行调度



磁盘调度算法

2. 最短寻道时间优先 (Shortest Seek Time First , SSTF)

选择处理与当前磁头距离最近的磁道请求，以减少寻道时间

但不能保证平均寻道时间最短

有可能出现 “饿死” (Starvation)



磁盘调度算法

3. SCAN（扫描）算法

要求磁头臂仅沿一个方向（假设向磁道号增加的方向）移动，并在途中满足所有未完成的请求，直到到达该方向的最后一个磁道或该方向上没有别的请求为止。

然后转向，沿相反方向扫描，同样按顺序完成所有请求。

如此不断反复。

又称**电梯调度算法**，其移动规律类似于电梯的运行。

克服了最短寻道时间优先的缺点，既考虑了距离，同时又考虑了方向



磁 盘 调 度 算 法

4. C-SCAN (Circular SCAN , 循环扫描)

将扫描限定在一个方向

当沿某个方向访问到最后一个磁道时，磁头臂返回到磁盘的另一端，再次开始扫描。



磁盘调度算法

【例】 设请求访问的磁道顺序为

55, 58, 39, 18, 90, 160, 150, 38, 184

假定从磁道100处开始。

(1) FCFS

**下一个访问
的磁道**

55 58 39 18 90 160 150 38 184

**移动距离
(磁道数)**

45 3 19 21 72 70 10 112 146

**平均寻道
长度**

55.3



磁盘调度算法

【例】 设请求访问的磁道顺序为

55, 58, 39, 18, 90, 160, 150, 38, 184

假定从磁道100处开始。

(2) SSTF

**下一个访问
的磁道**

90 58 55 30 38 18 150 160 184

**移动距离
(磁道数)**

10 32 3 16 1 20 132 10 24

**平均寻道
长度**

27.5



磁盘调度算法

【例】 设请求访问的磁道顺序为

55, 58, 39, 18, 90, 160, 150, 38, 184

假定从磁道100处开始。

(3) SCAN (假定开始时向磁道号增加的方向)

**下一个访问
的磁道**

150 160 184 90 58 55 39 38 18

**移动距离
(磁道数)**

50 10 24 94 32 3 16 1 20

**平均寻道
长度**

27.8



磁盘调度算法

【例】 设请求访问的磁道顺序为

55, 58, 39, 18, 90, 160, 150, 38, 184

假定从磁道100处开始。

(4) C-SCAN(向磁道号增加的方向)

**下一个访问
的磁道**

150 160 184 18 38 39 55 58 90

**移动距离
(磁道数)**

50 10 24 166 20 1 16 3 32

**平均寻道
长度**

27.5



磁盘调度算法

SSTF、SCAN、C-SCAN存在的问题：

磁臂粘着 (Arm Stickness) 现象：一个或多个进程反复请求某个磁道I/O，从而垄断了整个磁盘，导致“饥饿”。

5. N-Step-SCAN

将磁盘请求队列分成若干长度为N的子队列

每一次用SCAN处理一个子队列

在处理某个子队列时，新请求必须加到其他队列中

队列间用FCFS

若N较大，接近于SCAN；

当 $N = 1$ ，就是FCFS



磁盘调度算法

6. FSCAN

使用2个子队列

当扫描开始时，所有请求都在一个子队列中，另一个子队列为空

在扫描过程中，所有新到的请求加入另一个队列中

使得新请求的服务延迟到老请求处理完之后。



7.6 磁 盘 I/O

三、磁盘I/O性能的改善（提高磁盘I/O性能的方法）

1. 磁盘高速缓存（Disk Cache）

逻辑上属于磁盘，物理上驻留内存

在内存中为磁盘设置一个缓冲区，包含磁盘块的副本

当请求磁盘读时，先查看所需的块是否在高速缓存中。如果在，则可直接进行读操作。否则，首先要将块读到高速缓存，再拷贝到所需的地方

如果高速缓存已满，则需要进行淘汰

置换算法：常用LRU、NRU等



磁盘I/O性能的改善

2. 合理分配磁盘空间

分配盘块时，把有可能顺序访问的块放在一起，最好在上一柱面上，从而减少磁臂的移动次数

3. 提前读

读当前块时，提前将下一盘块读入高速缓存

4. 延迟写

数据不立即写回磁盘

周期性地成簇写回



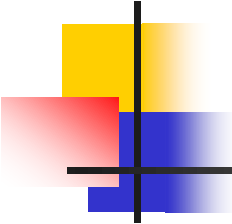
7.6 磁盘 I/O

四、RAID

RAID :

Redundant Array of Independent Disks, 独立磁盘冗余阵列

- ✓ RAID是一组物理磁盘驱动器，OS将其看作单个逻辑驱动器
- ✓ 数据分布在物理驱动器阵列中
- ✓ 一种容错技术，通过增加冗余来提高可靠性
- ✓ 若访问的数据块分布在多个磁盘上，I/O请求可以并行



7.7 小结

一、设备分类

- ✓ 字符设备与块设备
- ✓ 独占设备与共享设备

二、什么是设备独立性？为什么引入？如何实现？

三、设备驱动程序

四、I/O软件的分层结构以及每层的功能

五、SPOOLing技术的思想、应用

六、磁盘I/O

磁盘调度算法；提高磁盘I/O性能的几种方法