

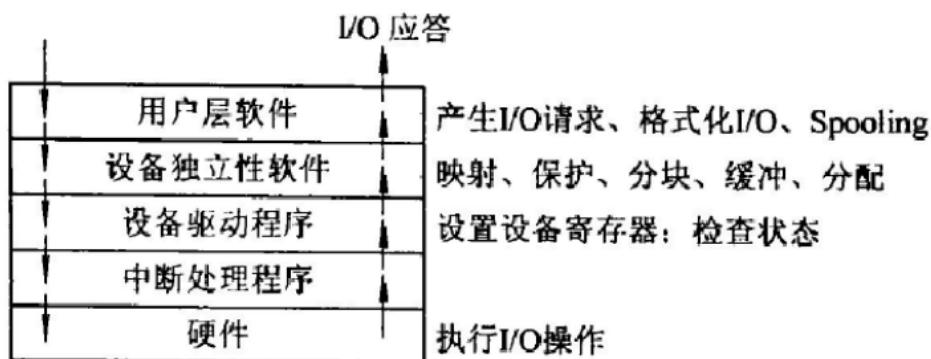
第五章 设备管理

by Jiacheng Xu

- 第五章 设备管理
 - 5.0 I/O系统的层次结构
 - 5.1 I/O系统
 - 5.1.1 I/O系统的基本结构
 - (1) 硬件基本结构
 - (2) I/O系统组成部分
 - 1. I/O设备
 - 2. 设备控制器
 - 3. I/O通道
 - (3) I/O系统结构
 - 1. 单通路I/O系统
 - 2. 多通路I/O系统
 - 5.1.2 I/O控制方式
 - (1) 通道控制方式
 - 1. 原理
 - 2. 通道指令
 - 3. 通道程序 (存放在内存中)
 - 5.2 缓冲管理
 - 5.2.1 缓冲的引入
 - (1) 引入缓冲的原因
 - 5.2.2 单缓冲
 - (1) 原理
 - (2) 处理时间
 - 5.2.3 双缓冲 (缓冲对换)
 - (1) 原理
 - (2) 处理时间
 - 5.2.4 循环缓冲
 - (1) 原理
 - (2) 实现
 - 5.2.5 缓冲池 (公用)
 - (1) 原理
 - (2) 两个基本操作

- (3) 工作方式
- 5.3 设备分配
 - 5.3.1 设备分配的基本概念
 - (1) 设备类型对设备分配的影响
 - (2) 设备分配算法和安全性
 - 5.3.2 设备独立性
 - (1) 概念
 - (2) 实现——逻辑设备与物理设备
 - (3) 优点
 - 5.3.3 SPOOLing技术
 - (1) 概述
 - (2) SPOOLing基本组成
 - (3) SPOOLing系统的典型应用——共享打印机
 - (4) SPOOLing技术实例——假脱机打印技术
 - (5) 特点
- 5.4 磁盘存储管理
 - 5.4.1 磁盘工作原理
 - (1) 磁盘的基本概念
 - (2) 磁盘工作相关数值量 (是否需要掌握计算?)
 - 5.4.2 磁盘调度算法
 - (0) 磁盘调度的概念
 - (1) 先来先服务调度算法 (FCFS)
 - (2) 最短寻道时间优先调度算法 (SSTF)
 - FCFS SSTF比较
 - (3) 扫描算法 (SCAN)
 - (4) 循环扫描算法 (CSCAN)
 - 5.4.3 磁盘高速缓存 (Disk Cache)
 - (1) 实现技术
 - 5.4.4 提高磁盘I/O速度的其他方法

5.0 I/O系统的层次结构



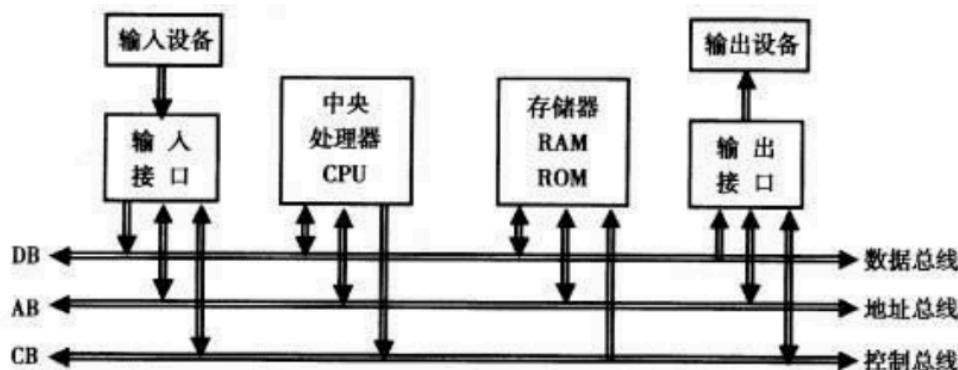
I/O执行的过程：

1. 硬件执行I/O操作
2. 发出中断，进行中断处理
3. 预备设备驱动
4. 保证设备独立
5. 到达用户层软件，实现I/O应答

5.1 I/O系统

5.1.1 I/O系统的基本结构

(1) 硬件基本结构



(2) I/O系统组成部分

1. I/O设备

- **定义**：具体完成数据I/O的设备
- **分类**：
 - 按速率分类：
 - **低速设备** (键盘)
 - **中速设备** (打印机)

- **高速**设备 (磁盘)
- 按信息交换单位分类:
 - **字符**设备 (键盘)
 - **块**设备 (磁盘)

2. 设备控制器

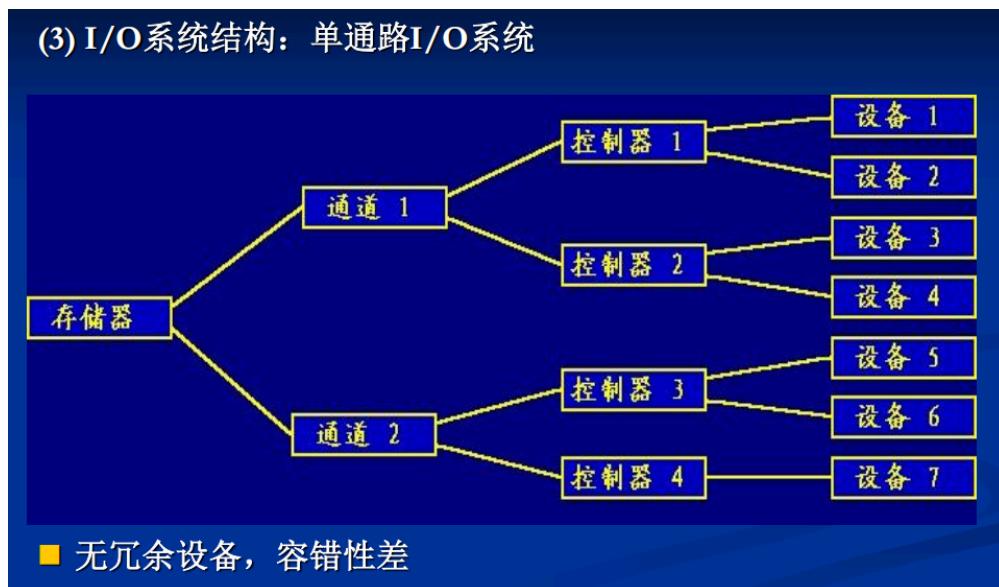
- **定义**: 负责连接I/O设备和数据总线，完成**设备控制和数据格式转换**
- **功能**: 接收和识别命令、数据交换、标志和报告设备状态、地址识别、数据缓冲、差错控制
- **编址**: 端口 (I/O端口)

3. I/O通道

- **定义**: 目的是使原来由CPU处理的I/O任务转由通道来承担，从而把CPU从繁杂的I/O任务中解脱出来
- **功能**:
 1. I/O通道是一种**特殊的处理器**，专门负责输入/输出。它具有自己的**指令系统**，但该指令系统比较简单
 2. 一般只有数据传送指令、设备控制指令等；
 3. 通道没有自己的内存，通道所执行的程序(即通道程序)是存放在**主机内存**中的，它与CPU共享内存。
- **编址**: 端口 (I/O端口)

(3) I/O系统结构

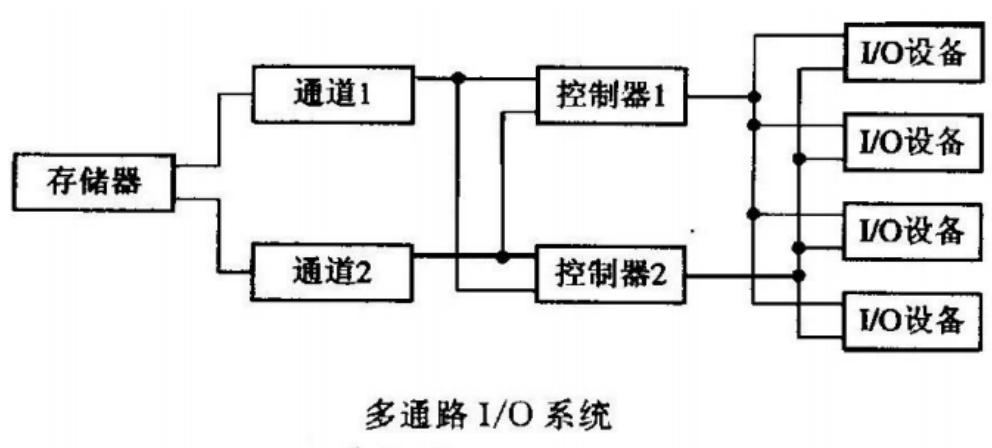
1. 单通路I/O系统



- 只有一种通路连接各个I/O设备

- 无冗余设备，容错性差

2. 多通路I/O系统



- 有冗余设备，容错性佳

5.1.2 I/O控制方式

(1) 通道控制方式

1. 原理

- 类似DMA方式，内存与设备直接进行数据交换
- 与DMA方式不同在于
 - 数据传送方向存放数据的内存始址及传送的数据块长度均由一个专门负责输入/输出的硬件——通道来控制。
 - DMA方式要求每台设备至少一个DMA控制器；通道控制方式下，一个通道可控制多台设备

2. 通道指令

- 操作码**：它规定指令所执行的操作，如读、写等。
- 内存地址**：标明数据传送时内存的首址。
- 计数**：表示传送数据的字节数。
- 通道程序结束位R0**：表示通道程序是否结束。
- 记录结束标志R1**：表示所处理的记录是否结束

3. 通道程序 (存放在内存中)

操作	P	R	计数	内存地址
WRITE	0	0	80	813
WRITE	0	0	140	1034
WRITE	0	1	60	5830
WRITE	0	1	300	2000
WRITE	0	0	250	1850
WRITE	1	1	250	720

对上图的解释 (了解) :

- 一个由六条通道指令所构成的简单的通道程序。该程序的功能是将内存中不同地址的数据，写成多个记录。
- 其中，前三条指令是分别将813-892单元中的80个字符和1034 -1173单元中的140个字符及5830 - 5889单元中的60个字符写成一个记录；
- 第4条指令是单独写一个具有300个字符的记录；
- 第5、6条指令共写含500个字符的记录

5.2 缓冲管理

5.2.1 缓冲的引入

(1) 引入缓冲的原因

- 缓和CPU和I/O设备的矛盾；
- 减少CPU中断的频率；
- 提高CPU和I/O设备的并行性

5.2.2 单缓冲

(1) 原理



(2) 处理时间

- $M+T > C$;

- $M+C < T$;

5.2.3 双缓冲 (缓冲对换)

(1) 原理

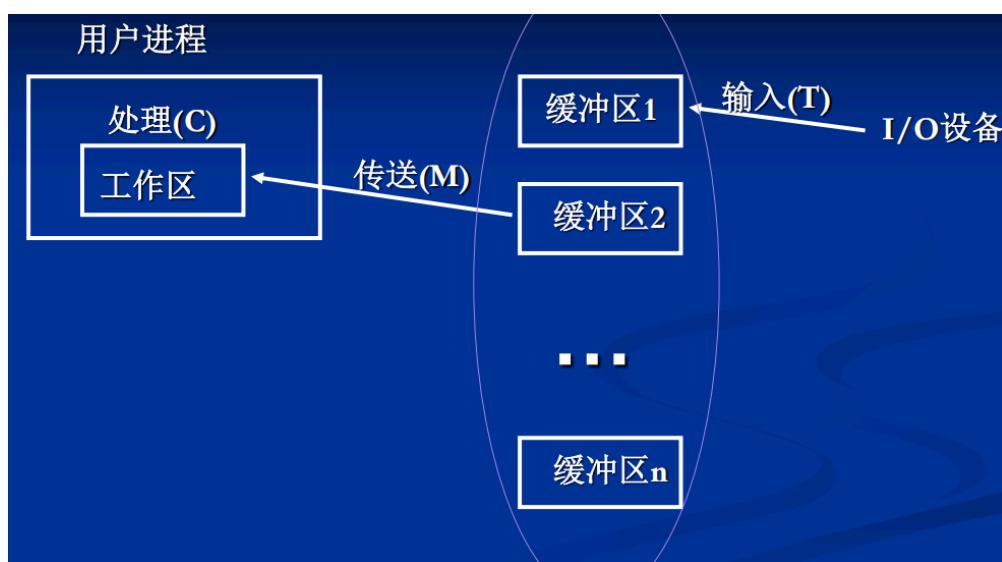


(2) 处理时间

- $\text{MAX}(C, T)$

5.2.4 循环缓冲

(1) 原理



(2) 实现

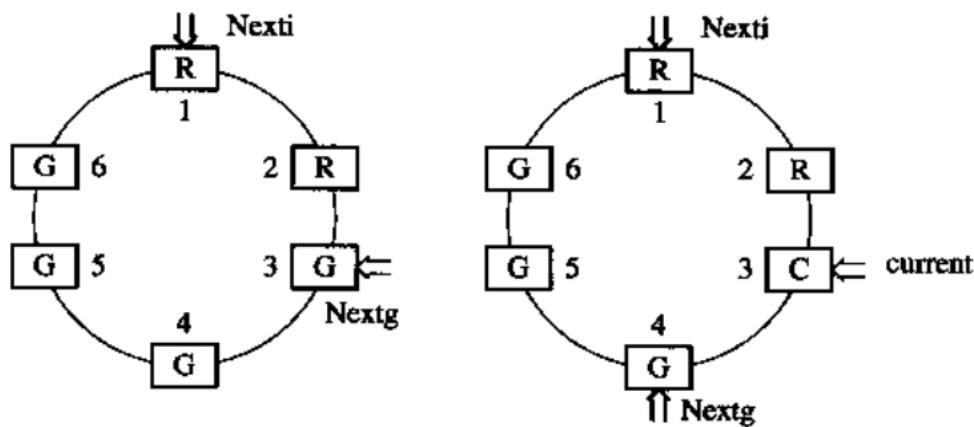
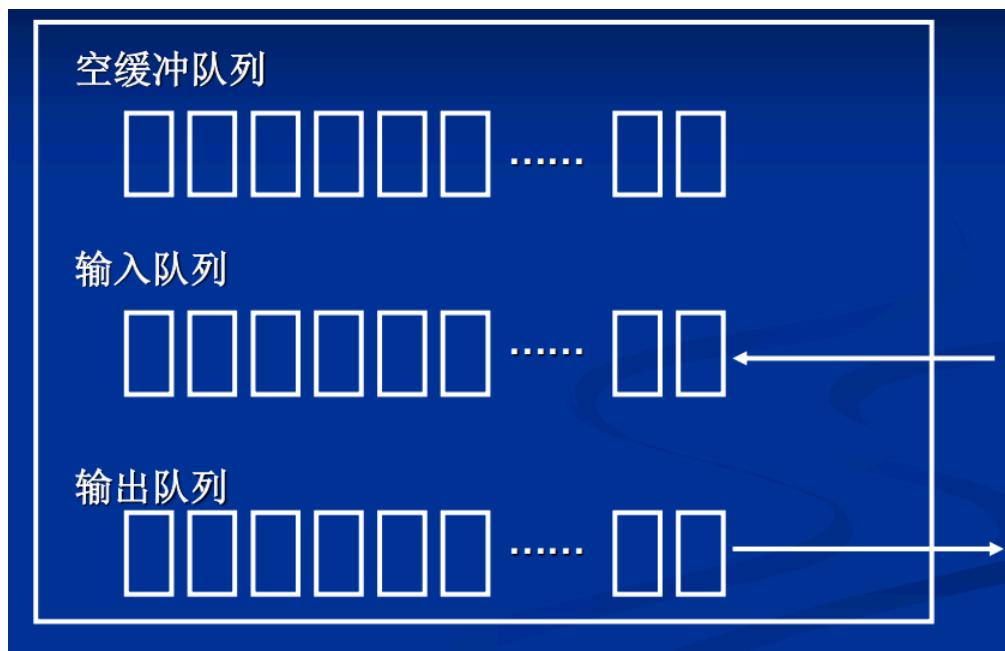


图 5-14 循环缓冲

5.2.5 缓冲池 (公用)

(1) 原理



(2) 两个基本操作

- GetBuf()
- PutBuf()

(3) 工作方式

- 收容输入
- 提取输入
- 收容输出

- 提取输出

5.3 设备分配

5.3.1 设备分配的基本概念

(1) 设备类型对设备分配的影响

1. 独占设备—独占分配

- **定义**：一段时间内只允许一个用户(进程)使用的设备
- **分配特点**：把该设备分配给某进程后，便让它独占直至释放
- **死锁**：该分配方式可能会引起进程死锁

2. 共享设备—共享分配

- **定义**：在一段时间内允许多个进程同时访问的设备
- **设备要求**：必须是可寻址的和可随机访问的设备。典型的共享设备是磁盘。
- **作用**：
 - 能获得良好的设备利用率
 - 是实现文件和数据共享的物质基础。

3. 虚拟设备—共享分配

- **定义**：通过某种技术将一台独占设备变换为能供若干个用户共享的设备。
- **作用**：可同时分配给多个用户，从而提高设备的利用率
- **SPOOLing技术**：典型的虚拟设备技术

(2) 设备分配算法和安全性

- **设备分配算法**：先来先服务、优先级高者优先...
- **设备分配的安全性**：
 - **安全方式**：串行分配，一个一个分配
 - **不安全方式**：并行分配，可同时分配多个设备资源

5.3.2 设备独立性

(1) 概念

定义：**用户程序**独立于具体使用的**物理设备**（含义是用户程序编写不需要考虑物理设备，只需要直到逻辑设备即可？）

(2) 实现——逻辑设备与物理设备

1. **核心思想**：采用**虚拟技术**，编程应用程序时不考虑实际物理设备
2. **引入逻辑设备和物理设备**：
 - **逻辑设备**：对应相同功能的物理设备的抽象

- **物理设备**: 应用程序执行时, 使用的**实际设备**

3. 基本方法:

- 编写程序时: 使用**逻辑设备名称**访问设备
- 程序执行时: 使用**具体物理设备**完成数据操作

4. 逻辑设备到物理设备的映射:

- 程序执行时, 使用**逻辑设备表 (LUT)** 完成逻辑设备到物理设备的映射

■ 逻辑设备表(LUT)

逻辑设备名	物理设备名	驱动程序入口
/dev/tty	3	0x00001024
/dev/printer	5	0x00002034
...

(3) 优点

1. 设备分配时的灵活性

- 物理设备请求, 只会给特定的某一台设备, 否则进程阻塞
- 而逻辑设备请求, 可以将同一类设备的任一台分配给进程

2. 易于实现I/O重定向

- I/O操作的设备可以更换(重定向), 而不需要改变应用程序
- 例如: 调试应用程序时, 可将程序的所有输出送往屏幕显示; 而在程序调试完后, 如需正式将程序的运行结果打印出来, 此时将I/O重定向的数据结构(逻辑设备表)中的显示终端改为打印机, 而不必修改应用程序

5.3.3 SPOOLing技术

(1) 概述

- **目的**: 缓和CPU的高速性与I/O设备低速性之间的矛盾

• 实现方法

- 早期: 引入脱机输入、脱机输出技术: 利用专门的外围控制机, 将低速I/O设备上的数据传送到高速磁盘上; 或者相反。
- 现在: 在引入多道程序技术后, 可以利用系统中的进程, 来模拟外围控制机功能, 从而:
 - 把低速I/O设备上的数据传送到高速磁盘上;
 - 把数据从磁盘传送到低速输出设备上。
 - 这样, 便可在主机的直接控制下, 实现脱机输入、输出功能。

(2) SPOOLing基本组成

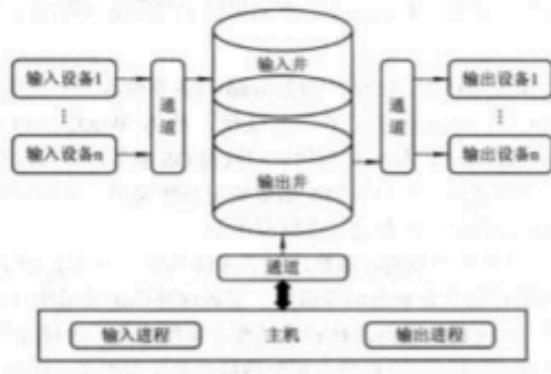
- 输入井和输出井
- 输入缓冲区和输出缓冲区

- 输入进程和输出进程

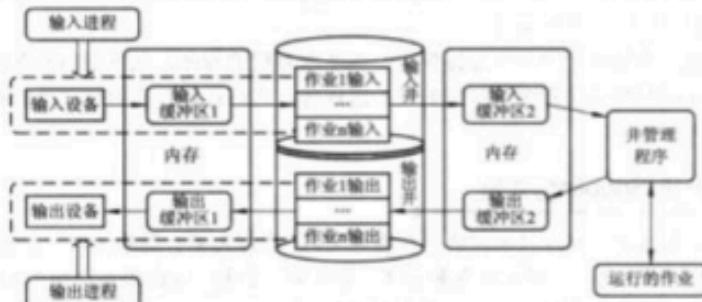
(3) SPOOLing系统的典型应用——共享打印机

2. SPOOLing 的组成

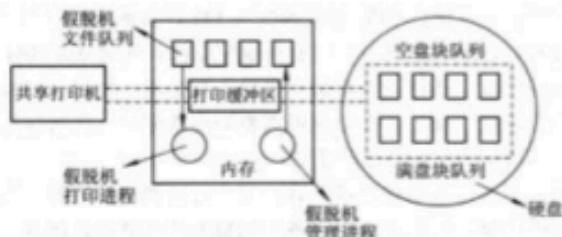
如前所述, SPOOLing 技术是对脱机输入/输出系统的模拟, 相应地, 如图 6-21(a)所示, SPOOLing 系统建立在通道技术和多道程序技术的基础上, 以高速随机外存(通常为磁盘)为后援存储器。SPOOLing 的工作原理如图 6-21(b)所示。



(a) SPOOLing系统的组成



(b) SPOOLing的工作原理



(c) 假脱机打印机系统的组成

图 6-21 SPOOLing 系统组成及工作原理

SPOOLing 系统主要由以下四部分构成:

(4) SPOOLing技术实例——假脱机打印技术



(5) 特点

- 提高I/O速度
- 改造独占设备为共享设备
- 实现虚拟设备

5.4 磁盘存储管理

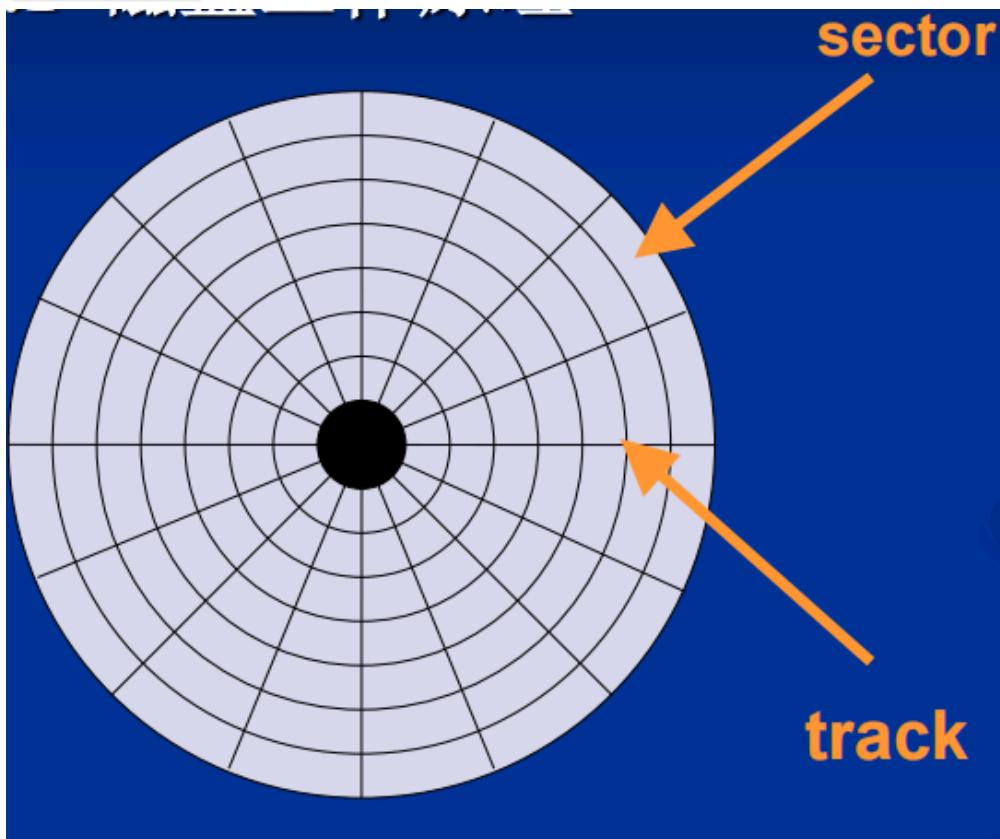
5.4.1 磁盘工作原理





(1) 磁盘的基本概念

- **组成**：硬盘由多个**盘片**组成
- **盘片的转速**：
 - 5400RPM
 - 7200RPM
 - 10000RPM
- **磁盘数据定位**：磁头号→磁道号→扇区号



(2) 磁盘工作相关数值量（是否需要掌握计算？）

磁盘访问时间： $T = T_s + T_r + T_t$

- 寻道时间 T_s : $T_s = 8 \sim 10 \text{ms}$
- 旋转延迟时间 T_r : $T_r = \frac{1}{2r}$, r 为转速
- 传输时间 T_t : $T_t = \frac{b}{rN}$, N 为每磁道字节数, b 为传输字节数

转速(RPM)	$T_s(\text{ms})$	$Tr(\text{ms})$	$Tt(\text{ms})$
5400	8-10	5.5	0.1
7200	8-10	4.3	0.07
10000	8-10	3	0.05

注: $b/N = 1/128$

结论: $T_s > Tr >> Tt$

5.4.2 磁盘调度算法

(0) 磁盘调度的概念

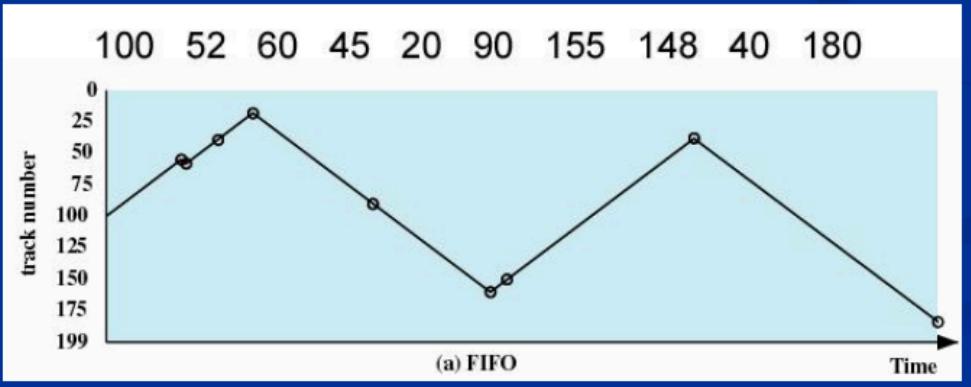
磁盘调度的原因:

- 磁盘可供多进程共享
- 需要一种最佳调度算法, 实现各进程对磁盘平均访问时间最小
- 磁盘访问时间主要是寻道时间, 故目标是使寻道时间最少

(1) 先来先服务调度算法 (FCFS)

- 方法: 根据进程请求磁盘的先后次序进行调度。
- 特点: 公平、简单, 每个进程的请求都能依次地得到处理, 不会出现某一进程的请求长期得不到满足的情况。

■ 磁道请求序列: 55,58,39,18,90,160,150,38,184

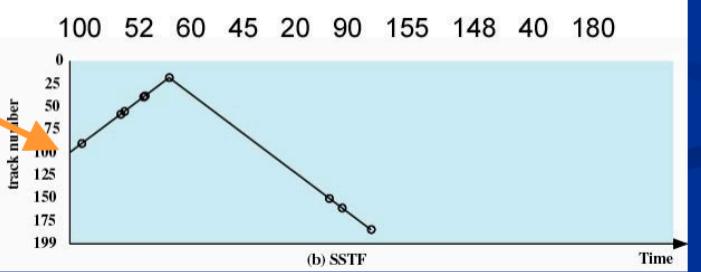


(2) 最短寻道时间优先调度算法 (SSTF)

- 方法: 要求访问的磁道, 与当前磁头所在的磁道距离最近, 以使每次的寻道时间最短
- 特点:
 - SSTF平均每次磁头移动距离<<FCFS
 - 故具有更好的寻道性能
 - 存在“磁臂黏着”现象

■ 磁道请求序列: 55,58,39,18,90,160,150,38,184

当前磁头所在的磁道



FCFS SSTF比较

(从 100 号磁道开始)	
被访问的下一个磁道号	移动距离 (磁道数)
55	45
58	3
39	19
18	21
90	72
160	70
150	10
38	112
184	146
平均寻道长度: 55.3	

图 5 - 23 FCFS 调度算法

(从 100 号磁道开始)	
被访问的下一个磁道号	移动距离 (磁道数)
90	10
58	32
55	3
39	16
38	1
18	20
150	132
160	10
184	24
平均寻道长度: 27.5	

图 5 - 24 SSTF 调度算法

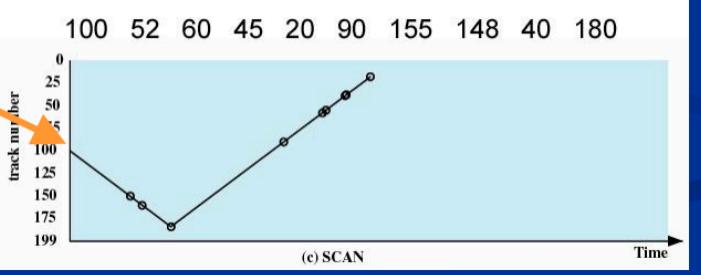
(3) 扫描算法 (SCAN)

- **方法**: 在磁头移动方向上, 且要求访问的磁道, 与当前磁头所在的磁道距离最近, 以使每次的寻道时间最短。
- **特点**: 有效解决磁臂粘着现象

■ 磁道请求序列: 55,58,39,18,90,160,150,38,184

当前磁头所在的磁道

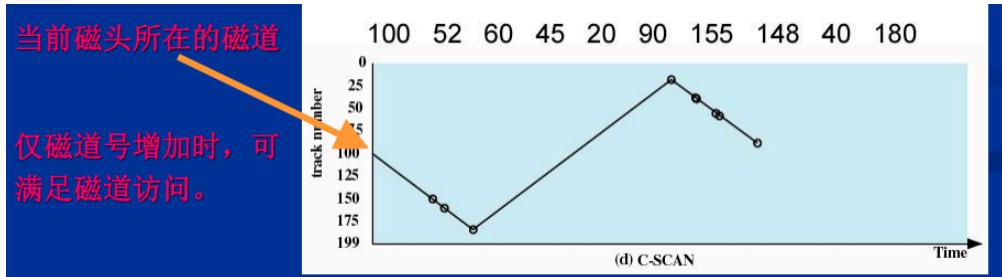
当前磁头移动方向:
磁道号增加。



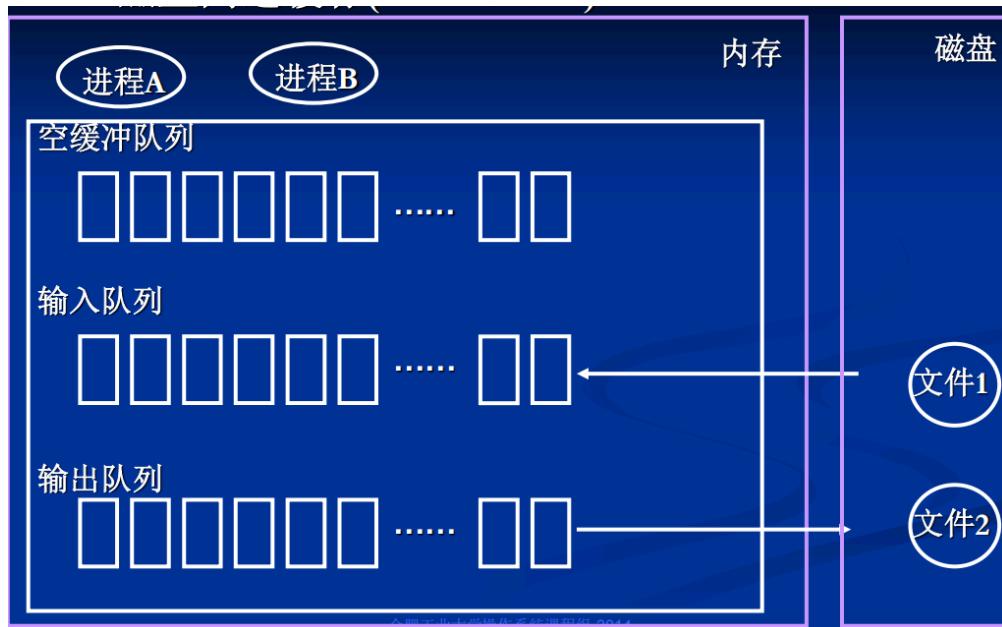
(4) 循环扫描算法 (CSCAN)

- **方法**:
 - 即为单向SCAN。只在一个磁头移动方向上满足磁道访问请求。
 - 要求访问的磁道, 与当前磁头所在的磁道距离最近, 以使每次的寻道时间最短

- **特点**: 有效降低磁道请求最大延迟



5.4.3 磁盘高速缓存 (Disk Cache)



(1) 实现技术

- 数据交付
- 置换算法: LRU
- 周期性写回

5.4.4 提高磁盘I/O速度的其他方法

- 提前读
- 延迟写
- 优化物理块分布