



张涛

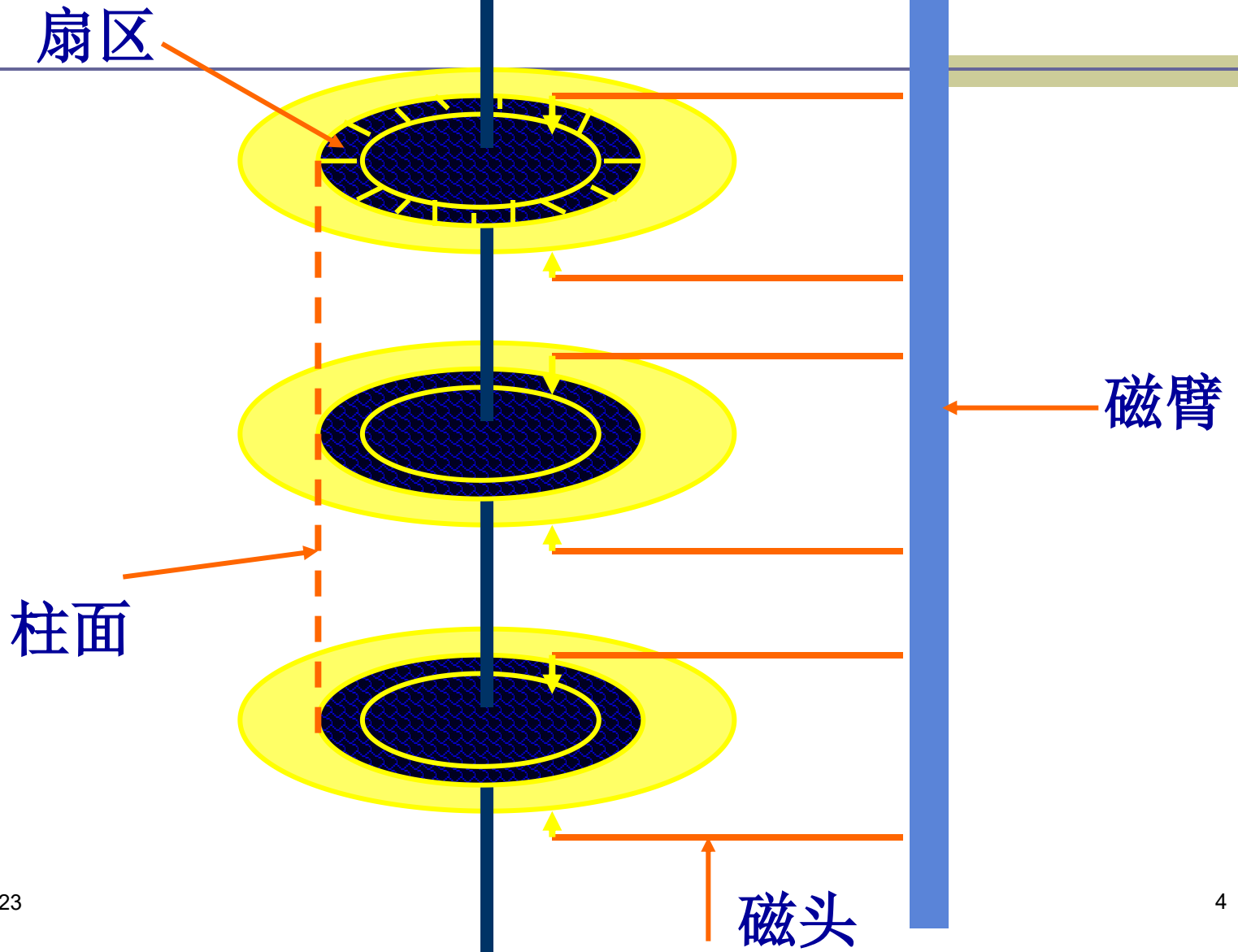
# 6.5 磁盘的驱动调度

- 磁盘概述
- 磁盘调度算法
- 提高磁盘I/O速度的方法

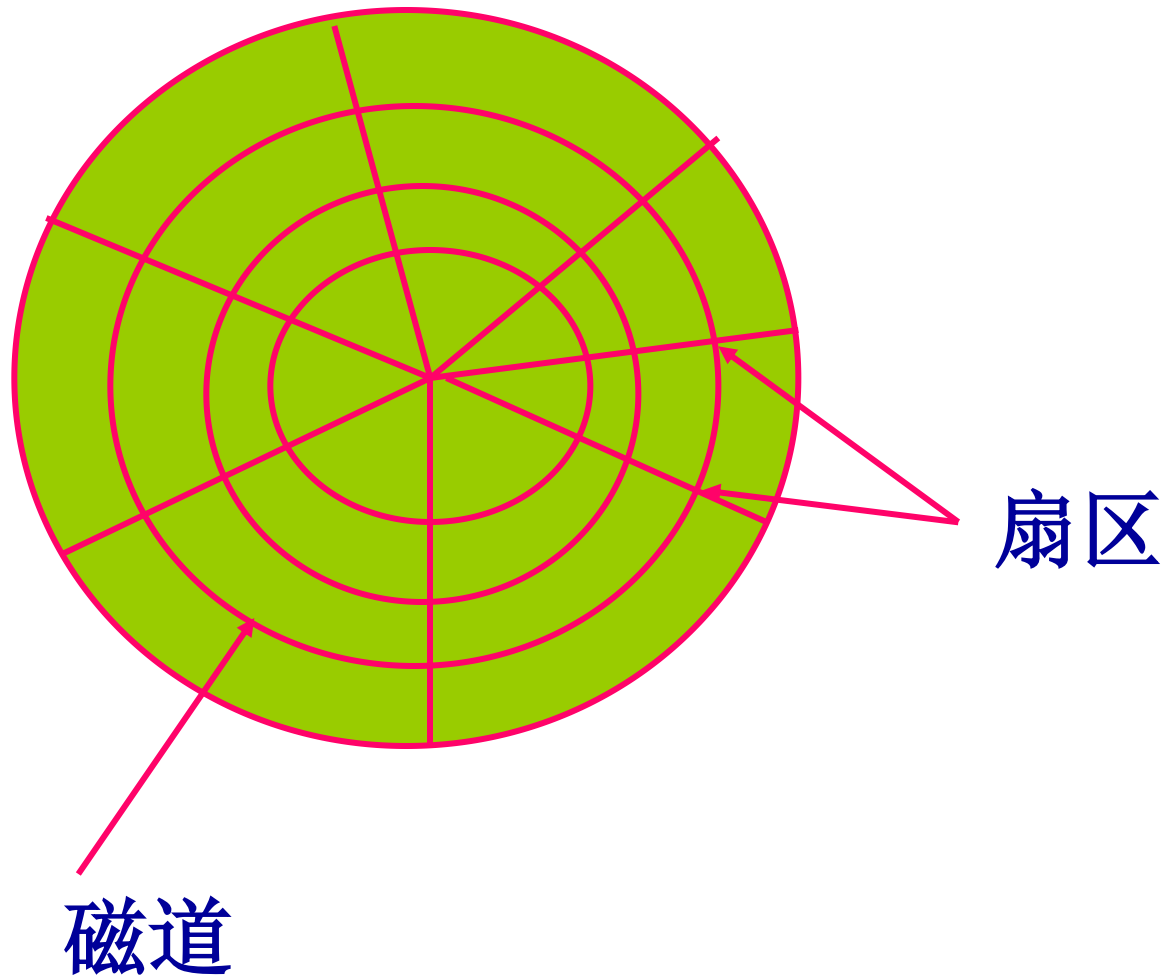
## 6.5.1 磁盘概述

- 目前，几乎所有随机存取的文件，都是存放在磁盘上，磁盘I/O速度的高低将直接影响文件系统的性能。
- 硬盘分为两种：
  - **固定头磁盘**：每个磁道设置一个磁头，变换磁道时不需要磁头的机械移动，速度快但成本高
  - **移动头磁盘**：一个盘面只有一个磁头，变换磁道时需要移动磁头，速度慢但成本低

# 侧视图



# 俯视图



# 柱面、磁头、扇区

- 信息记录在磁道上，多个盘片，正反两面都用来记录信息，每面一个磁头
- 所有盘面中处于同一磁道号上的所有磁道组成一个柱面
- 每个扇区大小为512字节
- 物理地址形式：
  - 柱面号
  - 磁头号
  - 扇区号

## 典型参数

20G:

39813 柱面

16 头

63 扇区

60G:

28733 柱面

16 头

255 扇区

# 磁盘的访问过程

## ■ 由三个动作组成：

- **寻道**：磁头移动定位到指定磁道
- **旋转延迟**：等待指定扇区从磁头下旋转经过
- **数据传输**：数据在磁盘与内存之间的实际传输

## ■ 磁盘的访问时间：

- **寻道时间 $T_s$** ：大约几ms到几十ms
- **旋转延迟时间 $T_r$** ：对于7200转/分，平均延迟时间为4.2ms
- **数据传输时间 $T_t$** ：目前磁盘的传输速度一般有几十M/s，传输一个扇区的时间小于0.05ms

# 思考

- 要提高磁盘的数据访问速度，主要应在哪方面下功夫？
- 应从以下两方面入手：
  - 数据的合理组织
  - 磁盘的调度算法



## 6.5.2 磁盘调度算法

- 当多个访盘请求在等待时，采用一定的策略，对这些请求的服务顺序调整安排，旨在降低平均磁盘服务时间，达到公平、高效
  - **公平**：一个I/O请求在有限时间内满足
  - **高效**：减少设备机械运动所带来的时间浪费
- 磁盘调度算法
  - 先来先服务
  - 最短寻道时间优先
  - 扫描算法
  - 单向扫描调度算法

# 先来先服务FCFS

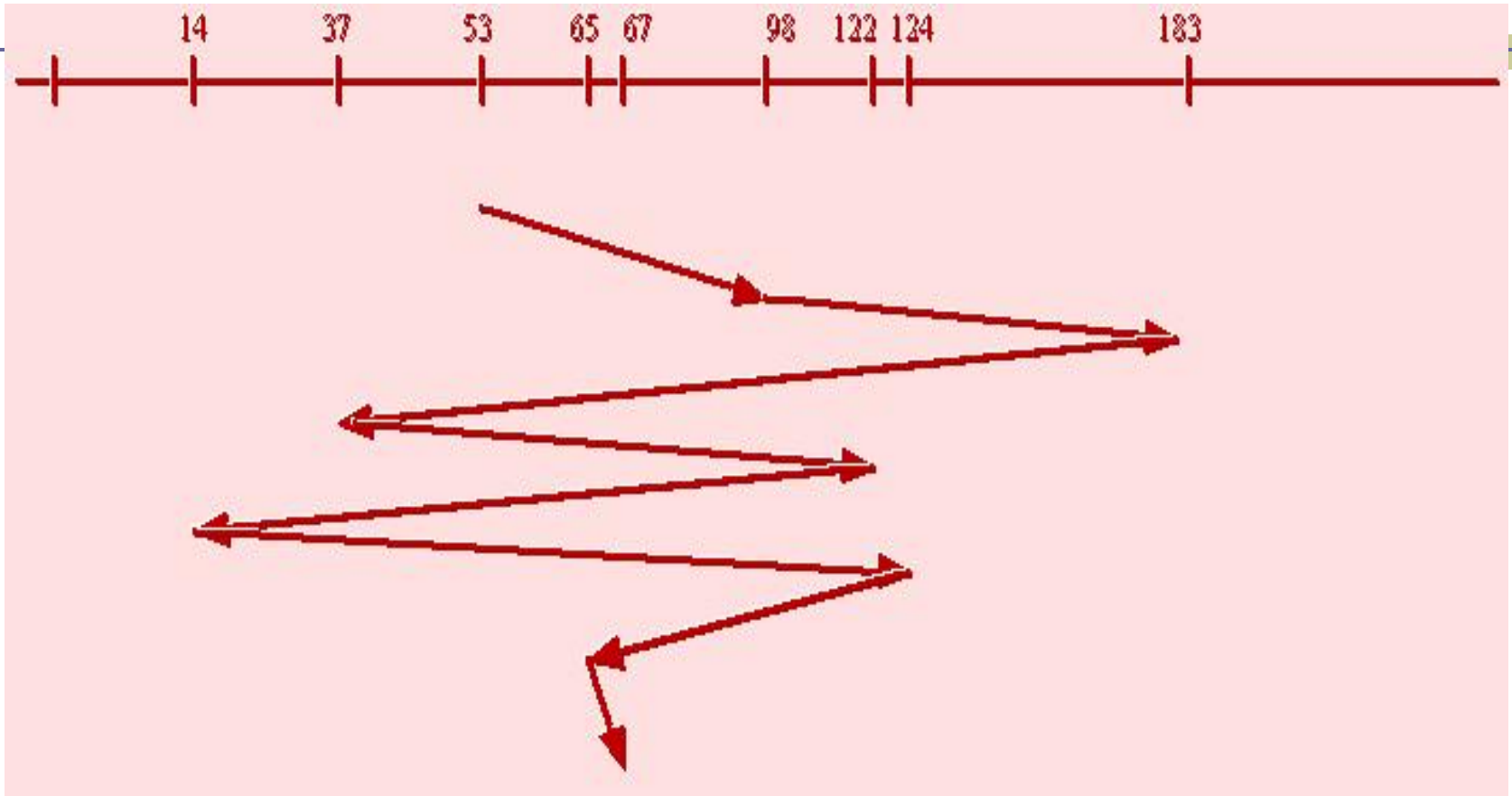
## First-Come, First Served

- 按访问请求到达的先后次序服务
- **优点：**简单，公平；
- **缺点：**效率不高，相邻两次请求可能会造成最内到最外的柱面寻道，使磁头反复移动，增加了服务时间，对机械也不利

# 例

- 假设磁盘访问序列：98, 183, 37, 122, 14, 124, 65, 67
- 读写头起始位置：53
- 安排磁头服务序列
- 计算磁头移动总距离（道数）

## 图解



**98, 183, 37, 122, 14, 124, 65, 67**

**磁头走过的总道数: 640**

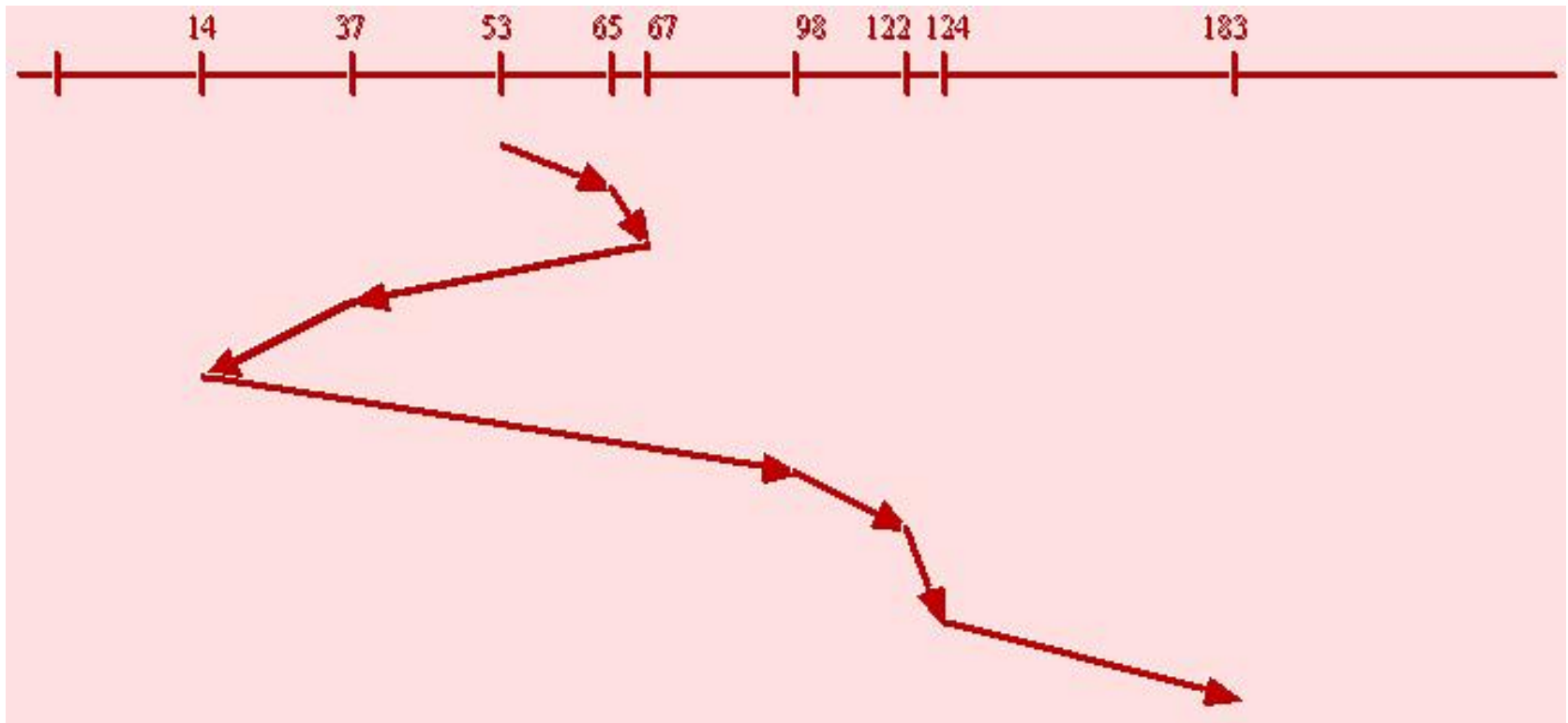
# 最短寻道时间优先SSTF

## Shortest Seek Time First

- 优先选择距当前磁头最近的访问请求进行服务，主要考虑寻道优先
- **优点：** 改善了磁盘平均服务时间；
- **缺点：** 造成某些访问请求长期等待得不到服务

## 图解

98, 183, 37, 122, 14, 124, 65, 67



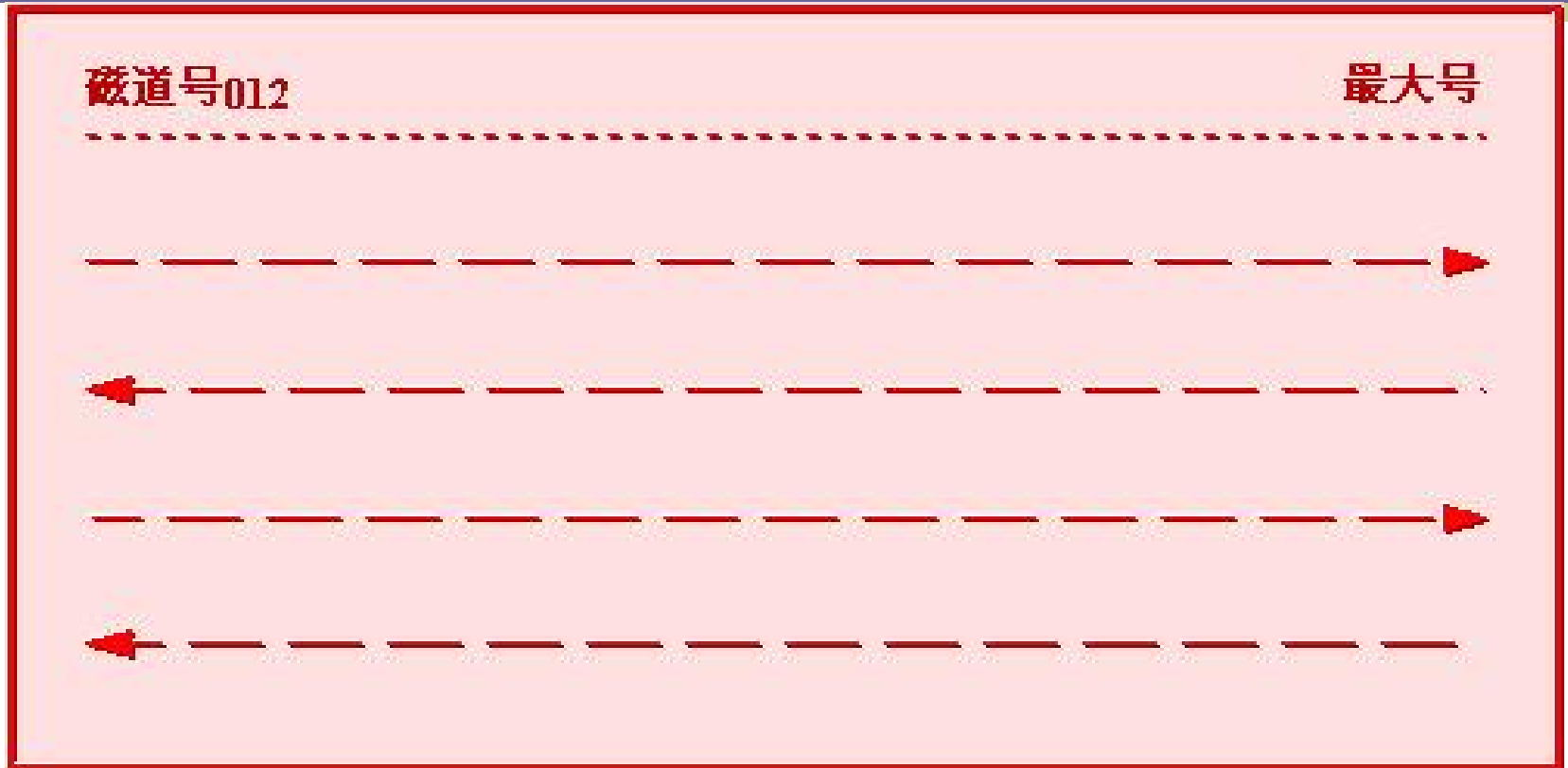
65, 67, 37, 14, 98, 122, 124, 183

磁头走过的总道数: 236

# 扫描算法（电梯算法）

- 克服了最短寻道优先的缺点，既考虑了距离，同时又考虑了方向
- 具体做法：当设备无访问请求时，磁头不动；当有访问请求时，磁头按一个方向移动，在移动过程中对遇到的访问请求进行服务，然后判断该方向上是否还有访问请求，如果有则继续扫描；否则改变移动方向，并为经过的访问请求服务，如此反复

# 图

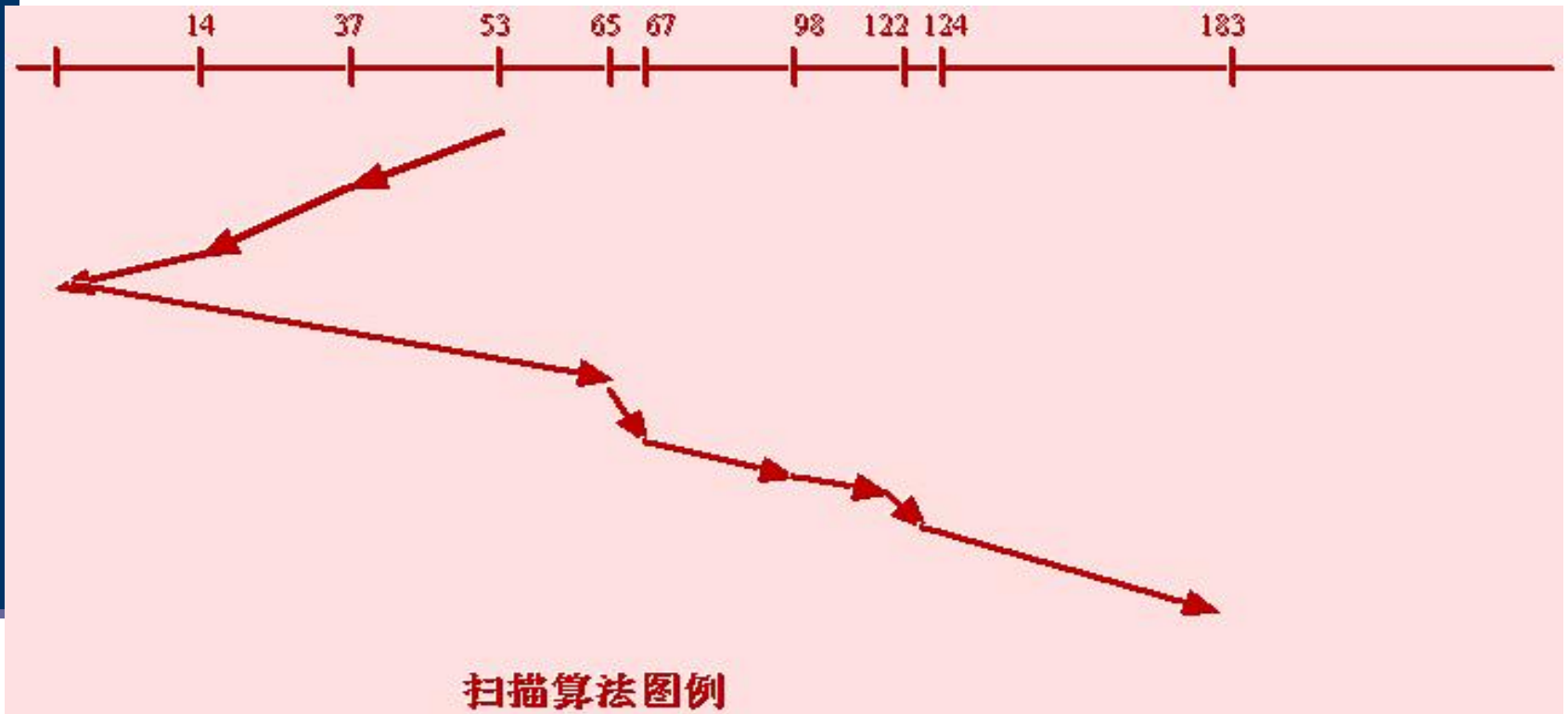


扫描算法（电梯算法）的磁头移动轨迹



## 图解

98, 183, 37, 122, 14, 124, 65, 67



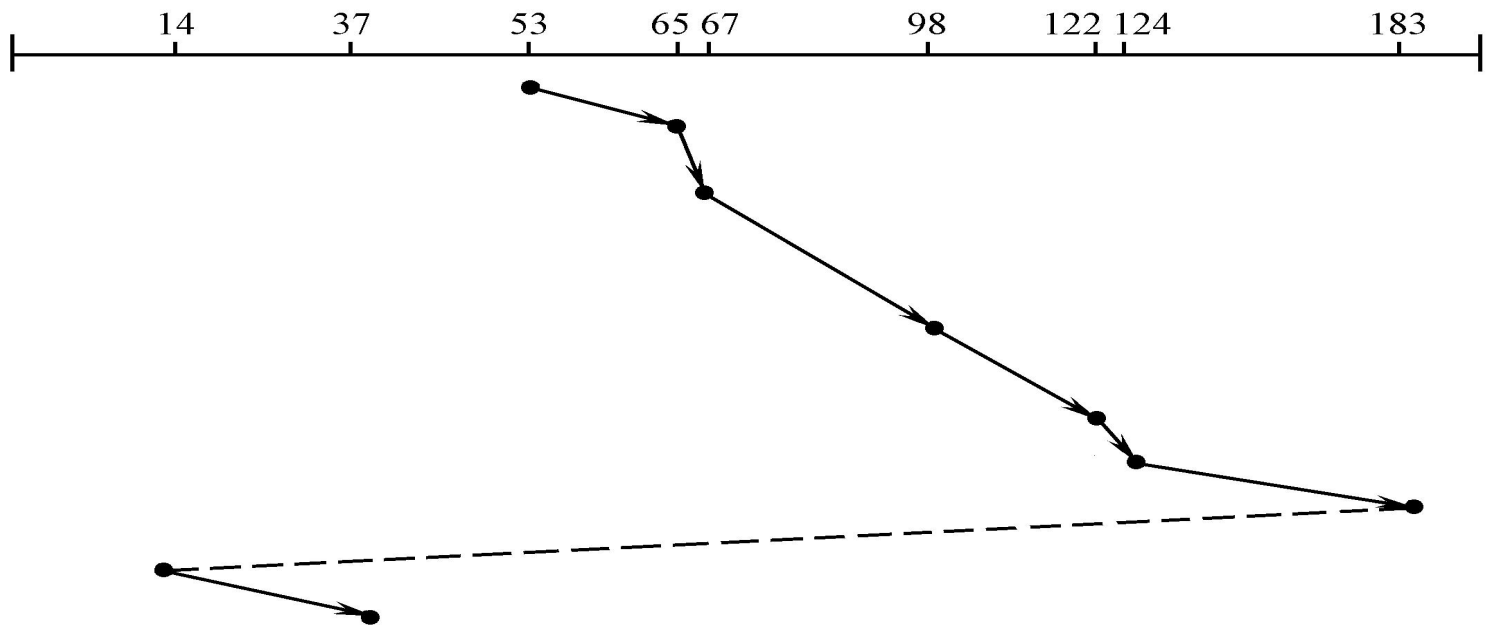
37, 14, 65, 67, 98, 122, 124, 183

磁头走过的总道数: 208

# 循环扫描调度算法CSCAN

- 电梯算法杜绝了饥饿，但当请求对磁道的分布是均匀时，磁头回头，近磁头端的请求很少（因为磁头刚经过），而远端请求较多，这些请求等待时间要长一些。
- 总是从0号柱面开始向里扫描。移动臂到达最后一个一个柱面后，立即带动读写磁头快速返回到0号柱面。返回时不为任何的等待访问者服务。返回后可再次进行扫描

# 图解



# 调度算法的选择

- 实际系统相当普遍采用最短寻道时间优先算法，因为它简单有效，性价比好。
- 扫描算法更适于磁盘负担重的系统。
- 磁盘负担很轻的系统也可以采用先来先服务算法
- 一般要将磁盘调度算法作为操作系统的单独模块编写，利于修改和更换。

## 6.5.3 提高磁盘I/O速度的方法

### ■ 磁盘高速缓存

- 磁盘的I/O速度要比内存低4-6个数量级
- 分配一些内存作为磁盘高速缓存可以极大地提高磁盘I/O速度。

### ■ 优化数据分布

### ■ 其它方法

# 方法一：磁盘高速缓存

## ■ 两种方式：

- 在内存中开辟一个单独的存储空间作为磁盘高速缓存。
- 把所有未利用的内存空间变为一个缓冲池，供分页系统和磁盘I/O共享。

■ 数据交付：将磁盘高速缓存中的数据传送给请求者进程。数据交付有两种方式：

- **数据交付**：将数据从缓存传到进程空间
- **指针交付**：将指向缓存中数据的指针传给进程

# 置换算法

- 如果高速缓存已满，则需要进行淘汰。
- 常用置换算法：最近最久未使用LRU、最少使用LFU等。
- 周期性写回：
  - 磁盘LRU算法中，那些经常被访问的盘块可能会一直保留在高速缓存中，而长期不被写回磁盘中。留下了安全隐患。
  - 解决之道：周期性写回。周期性地强行将已修改盘块写回磁盘。周期一般为几十秒。

# 方法二：优化数据的分布

## ■ 优化物理块的分布

- 物理块连续分配可以减少磁头的移动。
- 增加物理块的大小也可减少磁头的移动。

## ■ 优化索引结点的分布

- 可将索引结点放在中间位置。
- 进一步可将磁道分组，每组都有索引结点和文件数据



# 提高磁盘I/O速度的其它方法

## ■ 提前读

- 在访问文件时经常是顺序访问，因此在读当前块时可以提前读出下一块。
- 提前读已经被广泛应用：UNIX、OS/2、Netware等。

## ■ 延迟写

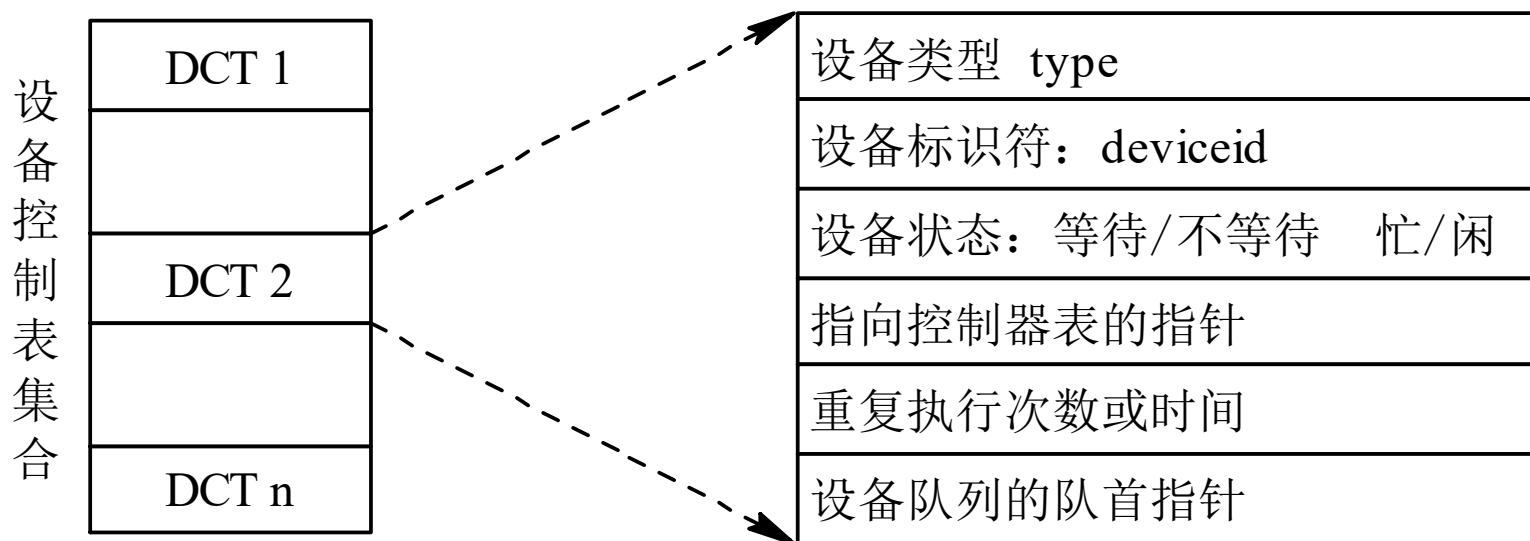
- 修改缓存中的数据后一般应立即写回磁盘，但该盘块可能还会被修改，立即写回会带来很大的开销。
- 置上延迟写标志。直到该盘块淘汰时或周期性写回时。
- 延迟写也被广泛应用：UNIX、OS/2等。

## ■ 虚拟盘

- 利用内存仿真磁盘，又称RAM盘。
- 虚拟盘同磁盘高速缓存的区别：虚拟盘的内容完全由用户控制，用户可见。缓存的内容完全由系统控制，用户不可见。

## 6.6 设备分配

- 设备分配方式
- 设备分配算法
- 设备分配技术



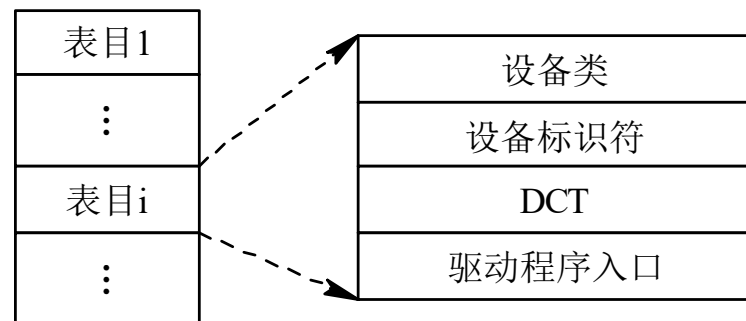
## 控制器控制表、通道控制表和系统设备表：

控制器标识符：controllerid
控制器状态：忙/闲
与控制器连接的通道表指针
控制器队列的队首指针
控制器队列的队尾指针

(a) 控制器表COCT

通道标识符：channelid
通道状态：忙/闲
与通道连接的控制器表首址
通道队列的队首指针
通道队列的队尾指针

(b) 通道表CHCT



(c) 系统设备表SDT

COCT、CHCT和SDT表

## 逻辑设备名到物理设备名映射的实现

逻辑设备名	物理设备名	驱动程序入口地址
/dev/tty	3	1024
/dev/printer	5	2046
⋮	⋮	⋮

(a)

逻辑设备名	系统设备表指针
/dev/tty	3
/dev/printer	5
⋮	

(b)

## 逻辑设备表

## 6.6.1 设备分配方式

### 静态分配：

在作业级进行的，当一个作业运行之前由系统一次分配满足需要的全部设备，这些设备一直为该作业占用，直到作业撤消。这种分配不会出现死锁，但设备的利用效率较低。

### 动态分配

在进程运行的过程中进行的，当进程需要使用设备时，通过系统调用命令向系统提出设备请求，系统按一定的分配策略给进程分配所需设备，一旦使用完毕立即释放。显然这种分配方式有利于提高设备的使用效率，但会出现死锁，这是应力求避免的。

## 6.6.2 设备分配算法

- 1、先请求先服务
- 2、优先级高的优先服务

## 6.6.3 设备分配技术

- 根据设备的特性把设备分成独占设备、共享设备和虚拟设备三种。
- 针对这三种设备采用三种分配技术：
  - 独享分配
  - 共享分配
  - 虚拟分配

# 独享分配

- 独占型设备有行打印机，键盘，显示器。磁带机可作为独占设备，也可作为共享设备。
- 若对这些设备不采用独享分配就会造成混乱。因此对独占设备一般采用独享分配，即当进程申请独占设备时，系统把设备分配给这个进程，直到进程释放设备。



# 共享分配

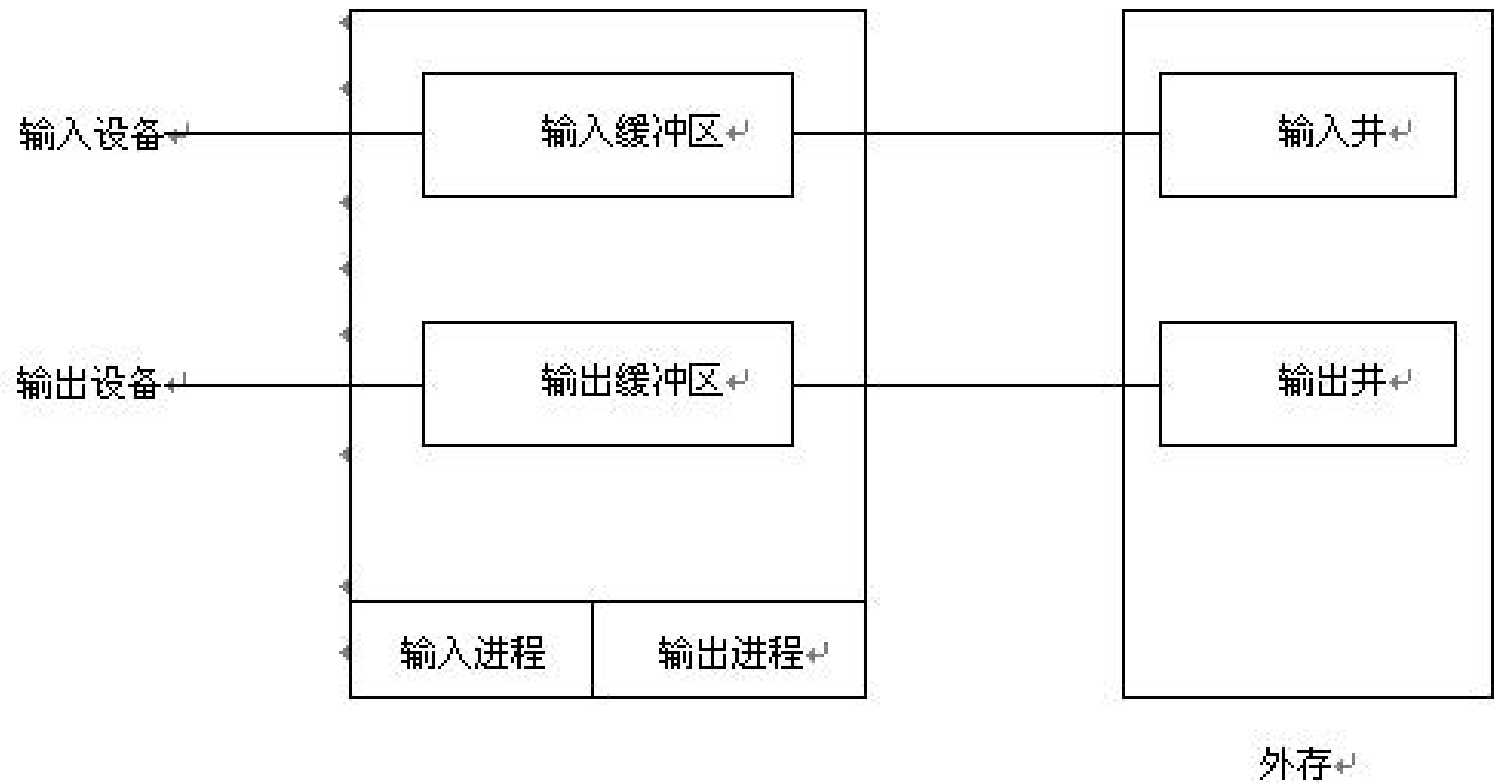
- 共享设备包括磁盘，磁带和磁鼓。
- 对这类设备的分配是采用动态分配的方式进行的，当一个进程要请求某个设备时，系统按照某种算法立即分配相应的设备给请求者，请求者使用完后立即释放。

# 虚拟分配

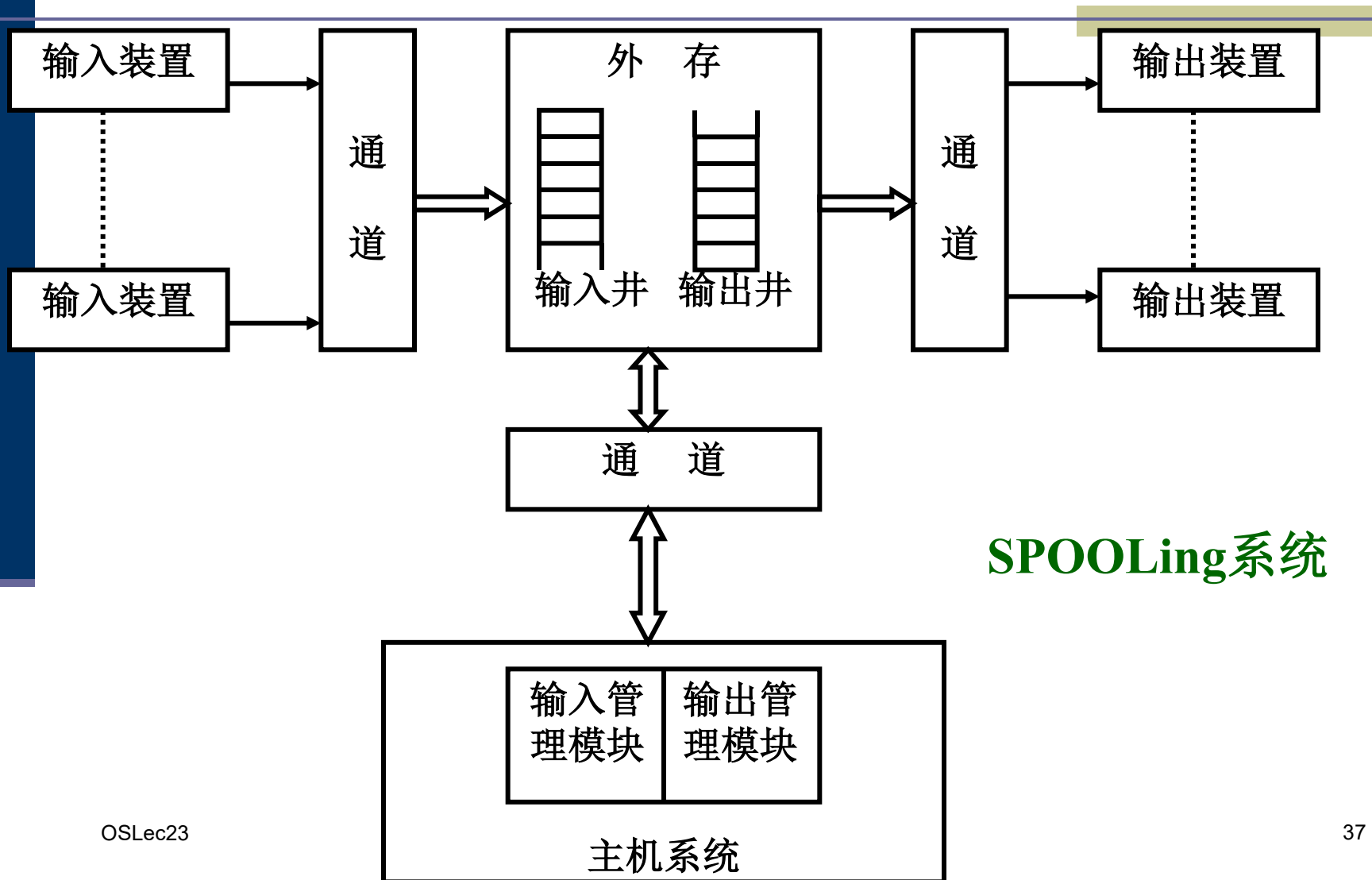
- 为提高计算机系统的效率，提出了在高速共享设备上模拟低速设备功能的技术，称为**虚拟设备技术**。
- 虚拟分配是针对虚拟设备而言的。实现过程是：
  - 当用户（或进程）申请独占设备时。系统给它分配共享设备的一部分存储空间。当程序要与设备交换信息时，系统就把要交换的信息存放在这部分存储空间。在适当的时候再将存储空间的信息传输到相应的设备上去处理。
- 共享设备中代替独占设备的那部分存储空间和相应的控制结构称为**虚拟设备**，并把对这类设备的分配称作虚拟分配。

# SPOOLing系统

- Simultaneous Peripheral Operations On-Line(外部设备同时联机操作)。
- 在联机情况下实现的同時外围操作称为SPOOLing, 也称为假脱机操作。
- SPOOLing系统的组成
  - 1、输入井和输出井
  - 2、输入缓冲区和输出缓冲区
  - 3、输入进程和输出进程



# 图示



# SPOOLing系统工作原理

- 作业执行前预先将程序和数据输入到输入井中
- 作业运行后，使用数据时，从输入井中取出
- 作业执行不必直接启动外设输出数据，只需将这些数据写入输出井中
- 作业全部运行完毕，再由外设输出全部数据和信息

好处：

- 实现了对作业输入、组织调度和输出的统一管理
- 使外设 CPU 直接控制下，与 CPU 并行工作（假脱机）

# SPOOLing系统的特点

---

- 1、提高了I/O速度
- 2、将独占设备改造为共享设备
- 3、实现了虚拟设备功能



That's all.



Thank you very much!