操作系统原理
第二十五讲

马佳曼

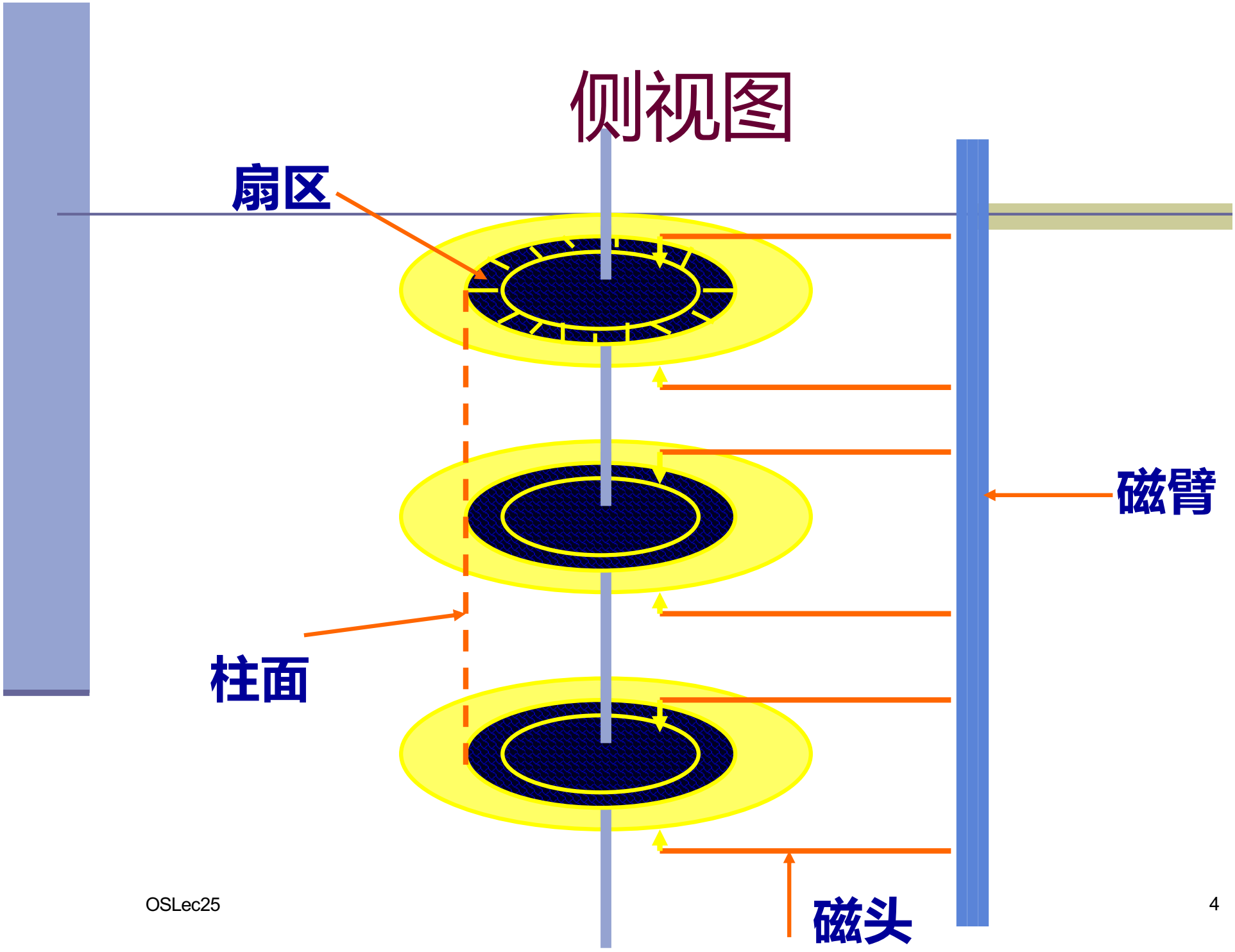
jiaman.ma@nwpu.edu.cn

6.5 磁盘的驱动调度

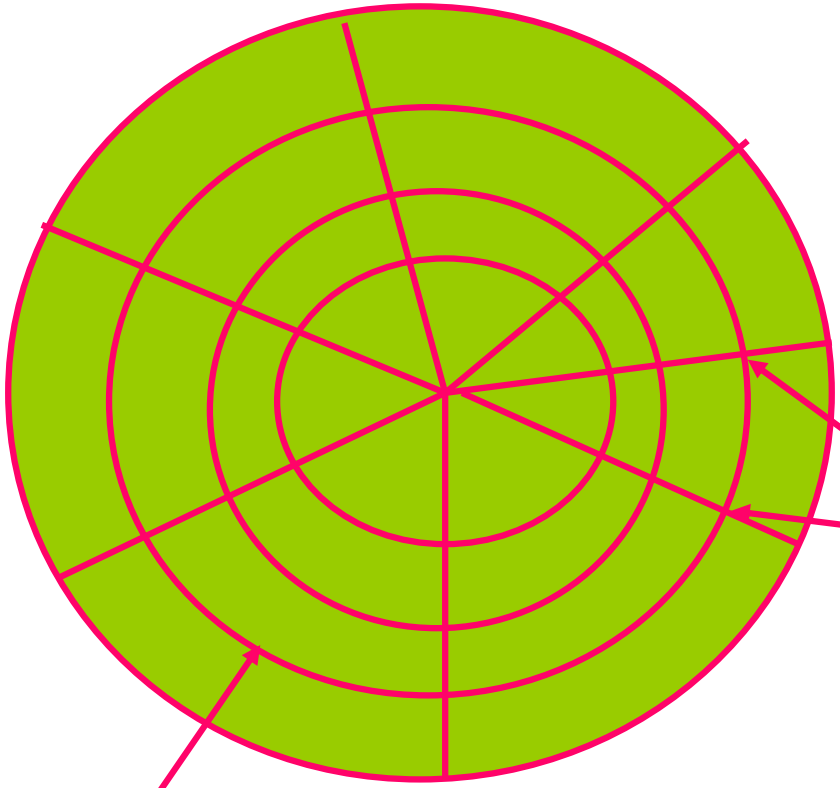
- 磁盘概述
- 磁盘调度算法
- 提高磁盘I/O速度的方法

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
扇区
磁臂
柱面
磁头

俯视图



扇区

磁道

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Ts：大约几ms到几十ms
  - 旋转延迟时间Tr：对于7200转/分，平均延迟时间为4.2ms
  - 数据传输时间Tt：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

$T_\tau = \frac{1}{2r}$

- 寻道时间T_s：启动磁头臂 s，跨越一个磁道耗时 m，总共跨越 n 条磁道

$$T_S = m \times n + s,$$

- 平均旋转延迟时间T_τ：r为磁盘每秒的转数 （1/r 转一圈花费的时间，找到扇区平均需要转半圈））

$$T_\tau = \frac{1}{2r}$$

- 传输时间T_t：r为磁盘每秒的转数，b 为读/写的字节数，每个磁道上字节数为 N

$$T_t = \frac{b}{rN},$$

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}。$$

思考

- 要提高磁盘的性能，主要应在哪方面下功夫?
- 应从以下三方面入手:
- 选择好的磁盘调度算法，以减少磁盘的寻道时间;
- 提高磁盘I/O速度，以提高对文件的访问速度;
- 采取冗余技术，提高磁盘系统的可靠性，建立高度可靠的文件系统 。

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

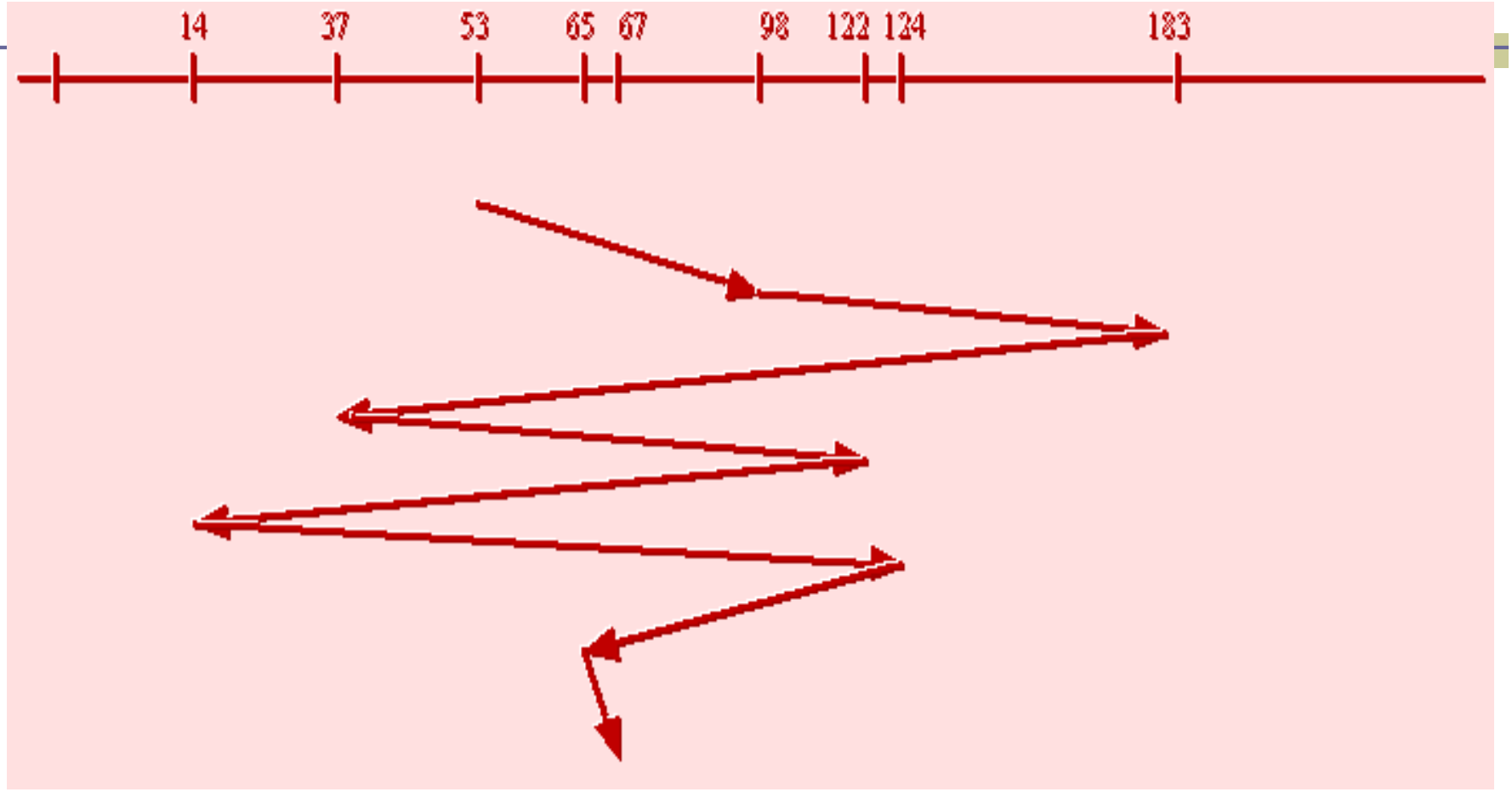
- 按访问请求到达的先后次序服务
- 优点：简单，公平；
- 缺点：效率不高，相邻两次请求可能会造成最内到最外的柱面寻道，使磁头反复移动，增加了服务时间，对机械也不利

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

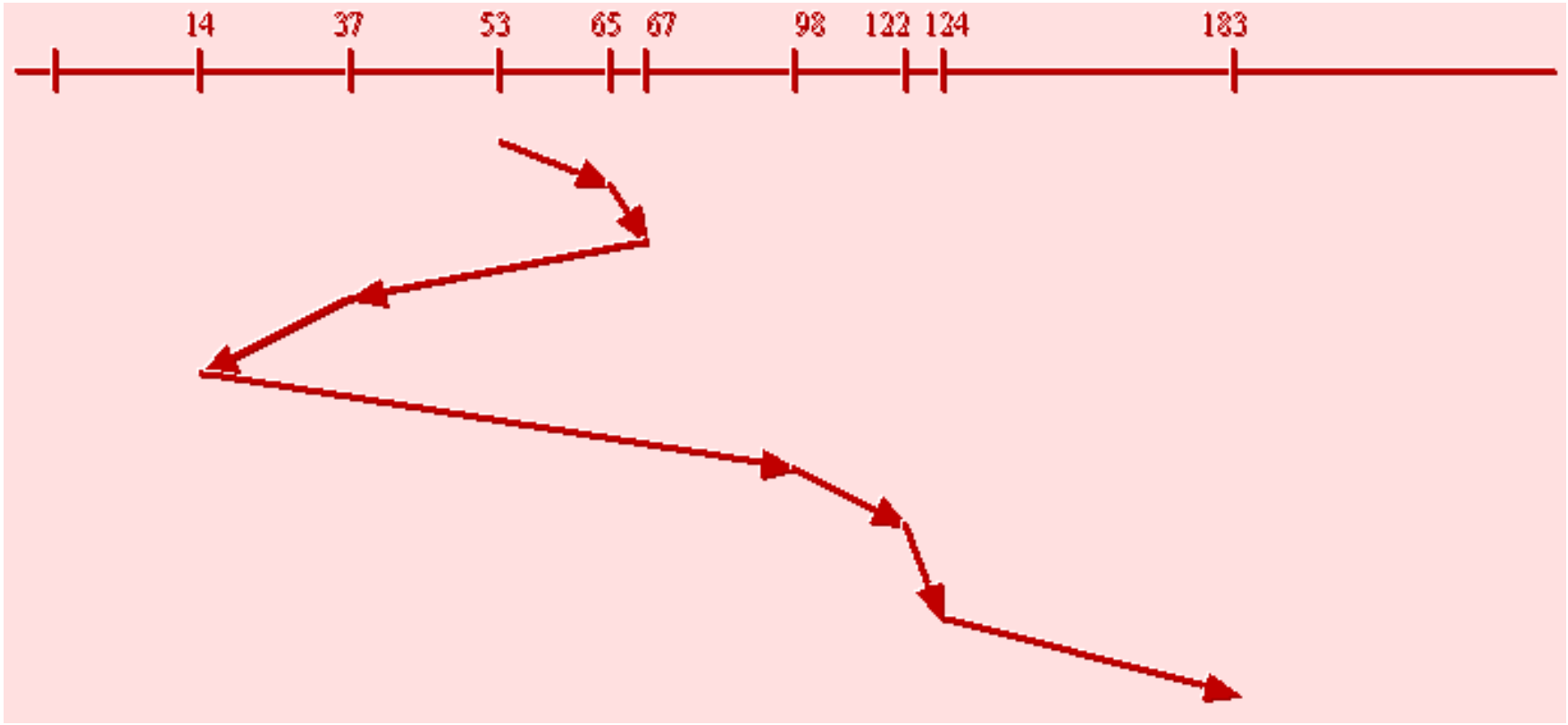
最短寻道时间优先SSTF
Shortest Seek Time First

- 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



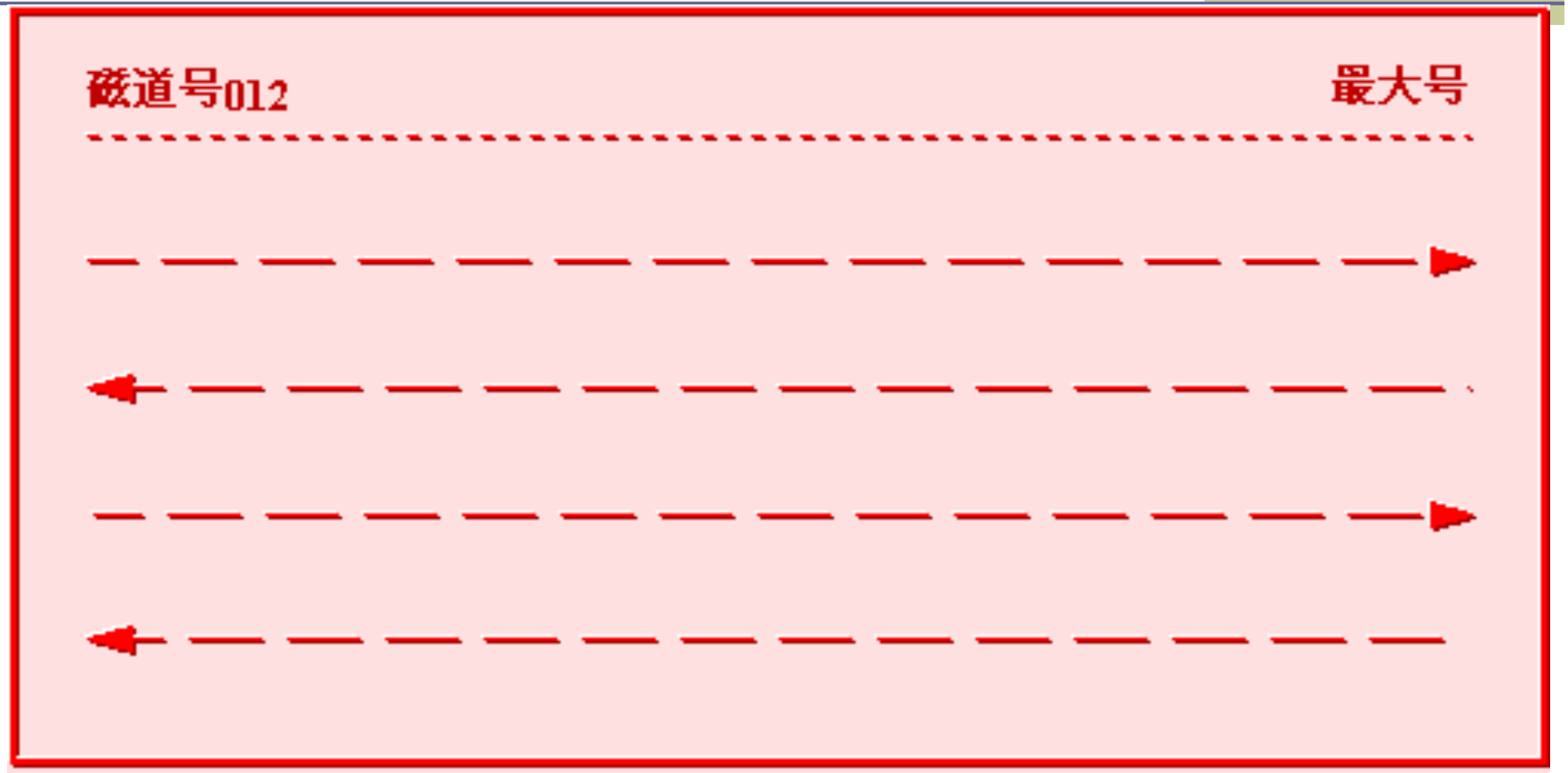
65，67，37，14，98，122，124，183

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

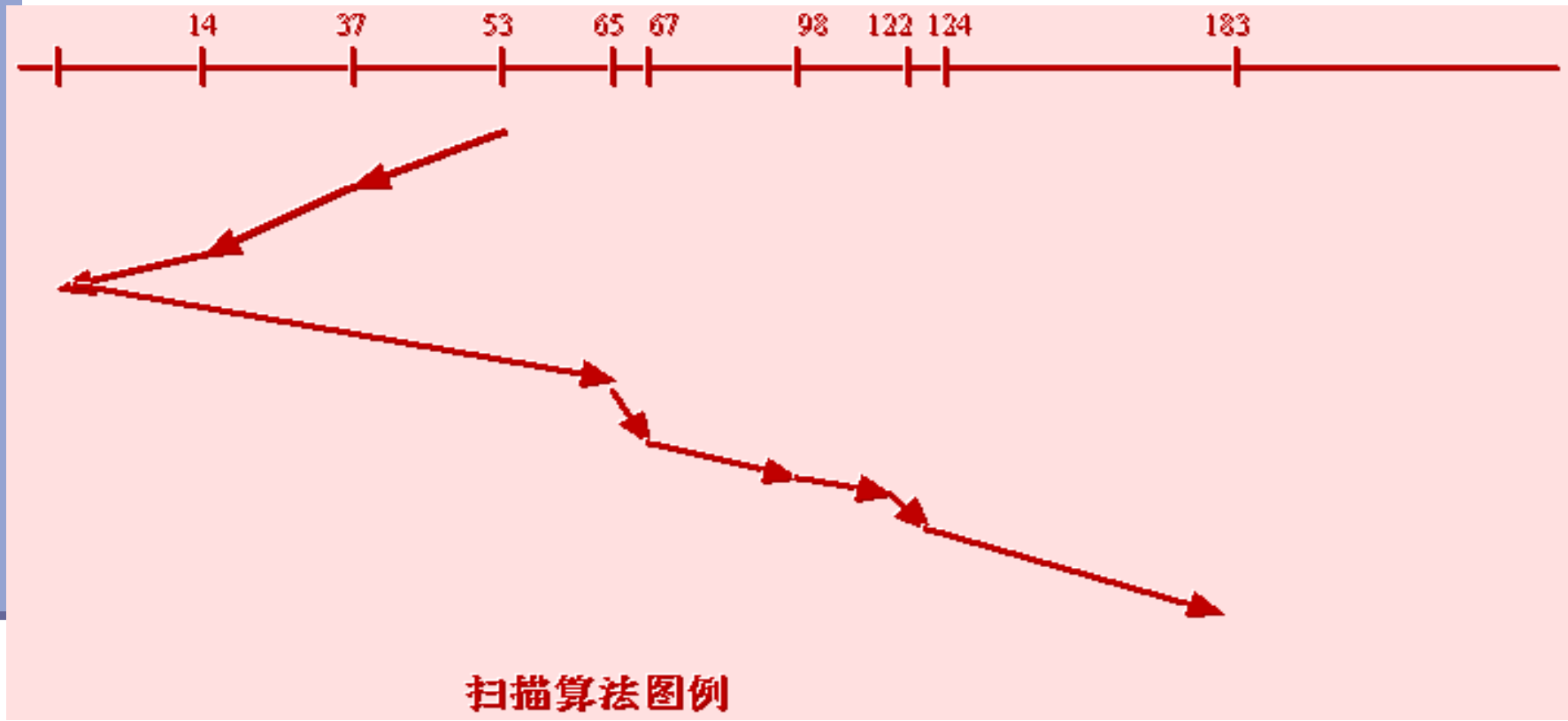
图



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



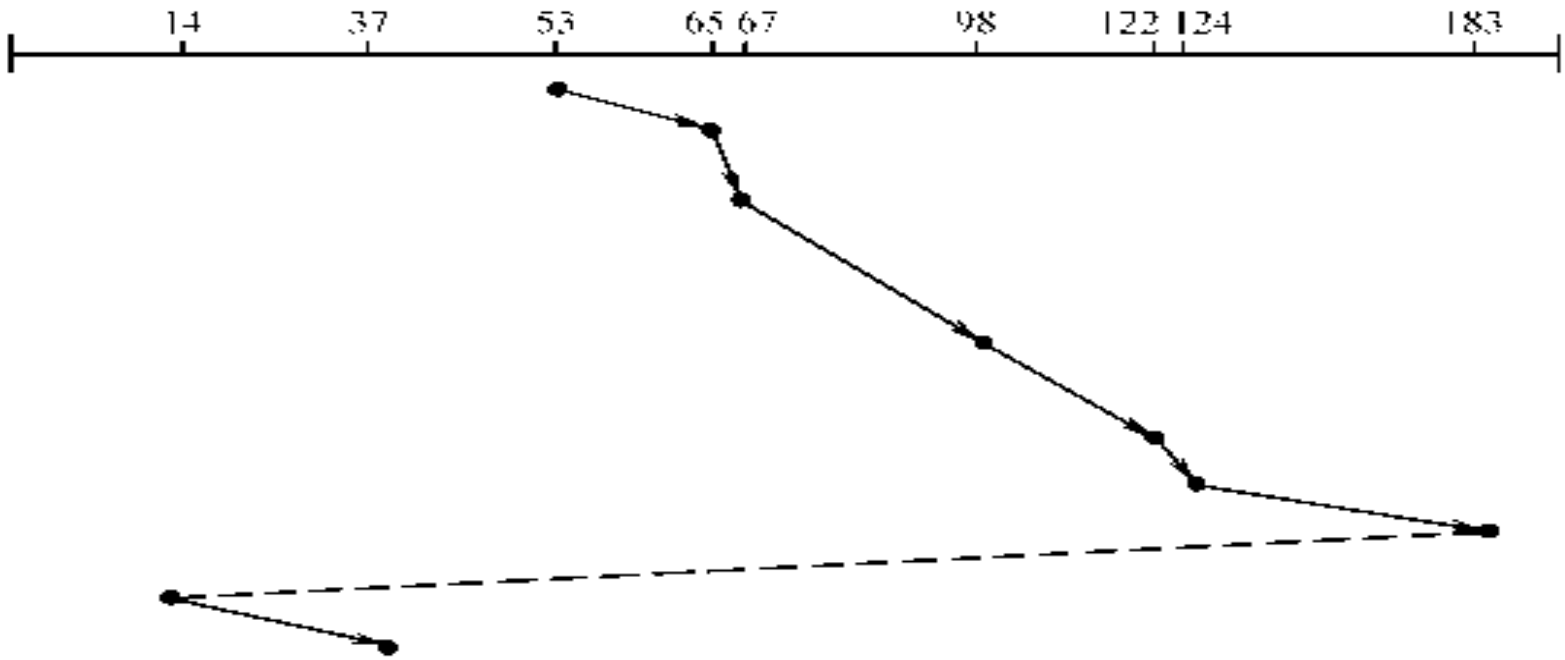
37，14，65，67，98，122，124，183

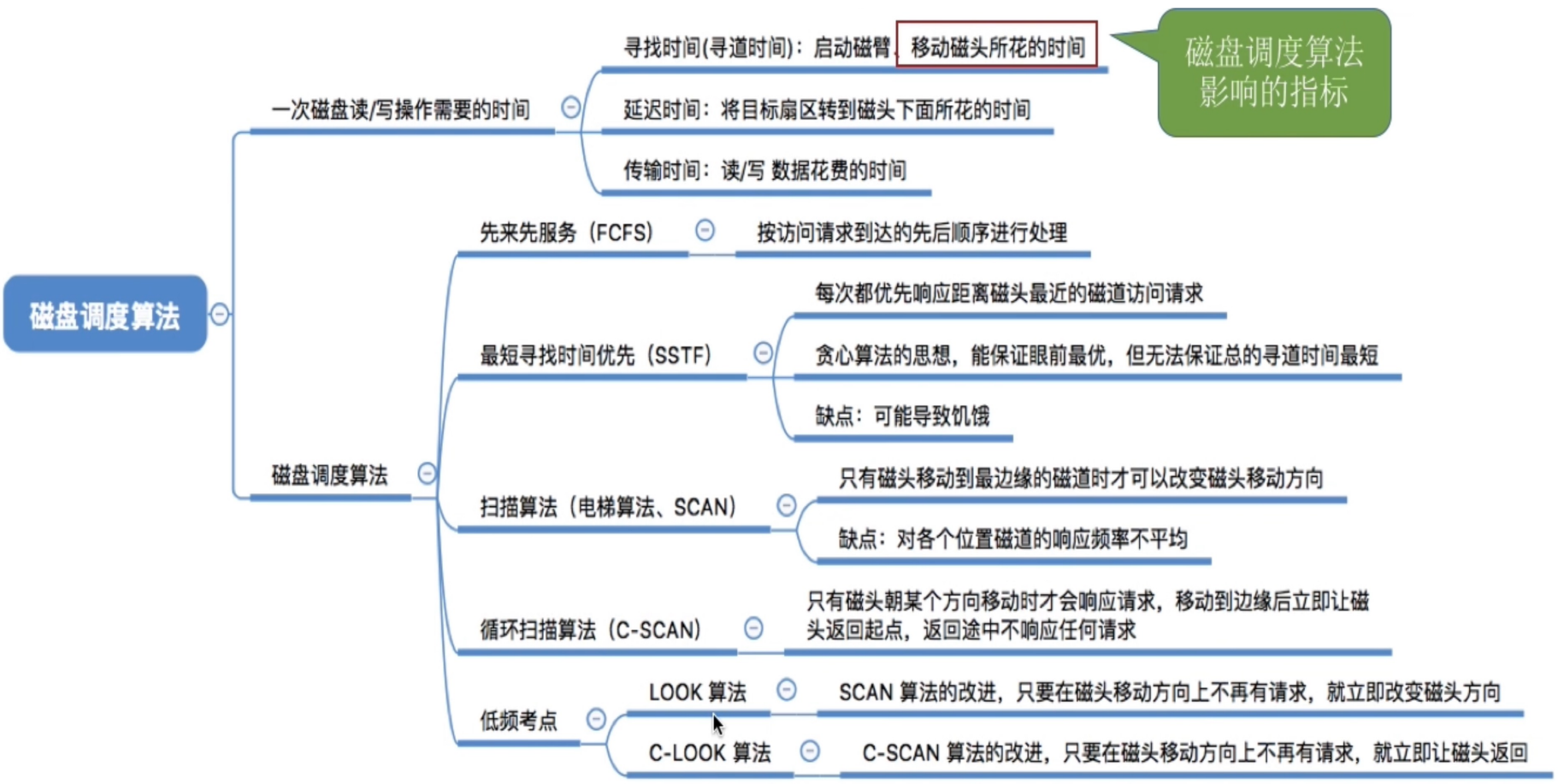
磁头走过的总道数：208

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




磁盘调度算法
一次磁盘读/写操作需要的时间
寻找时间(寻道时间)：启动磁臂，移动磁头所花的时间
延迟时间：将目标扇区转到磁头下面所花的时间
传输时间：读/写 数据花费的时间
磁盘调度算法影响的指标
磁盘调度算法
先来先服务（FCFS）
按访问请求到达的先后顺序进行处理
最短寻找时间优先（SSTF）
每次都优先响应距离磁头最近的磁道访问请求
贪心算法的思想，能保证眼前最优，但无法保证总的寻道时间最短
缺点：可能导致饥饿
扫描算法（电梯算法、SCAN）
只有磁头移动到最边缘的磁道时才可以改变磁头移动方向
缺点：对各个位置磁道的响应频率不平均
循环扫描算法（C-SCAN）
只有磁头朝某个方向移动时才会响应请求，移动到边缘后立即让磁头返回起点，返回途中不响应任何请求
低频考点
LOOK 算法
SCAN 算法的改进，只要在磁头移动方向上不再有请求，就立即改变磁头方向
C-LOOK 算法
C-SCAN 算法的改进，只要在磁头移动方向上不再有请求，就立即让磁头返回

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

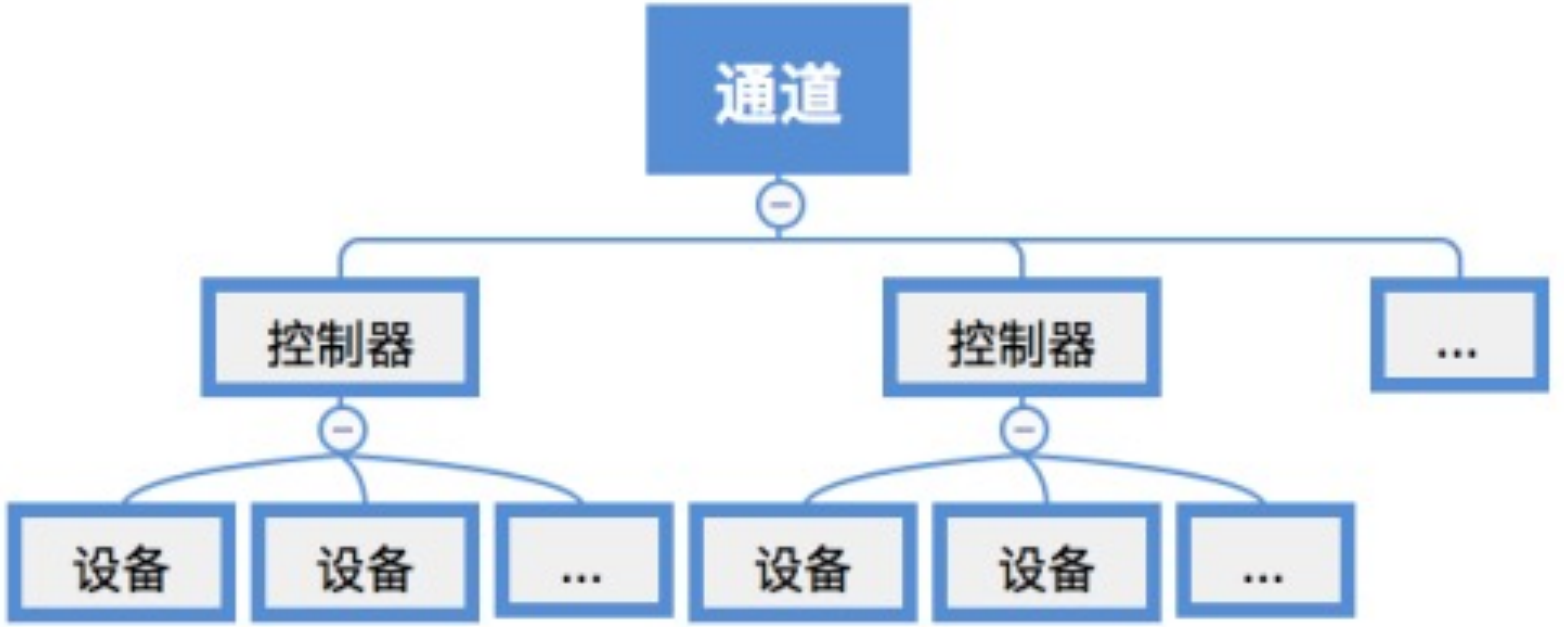
提高磁盘I/O速度的其它方法

- 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

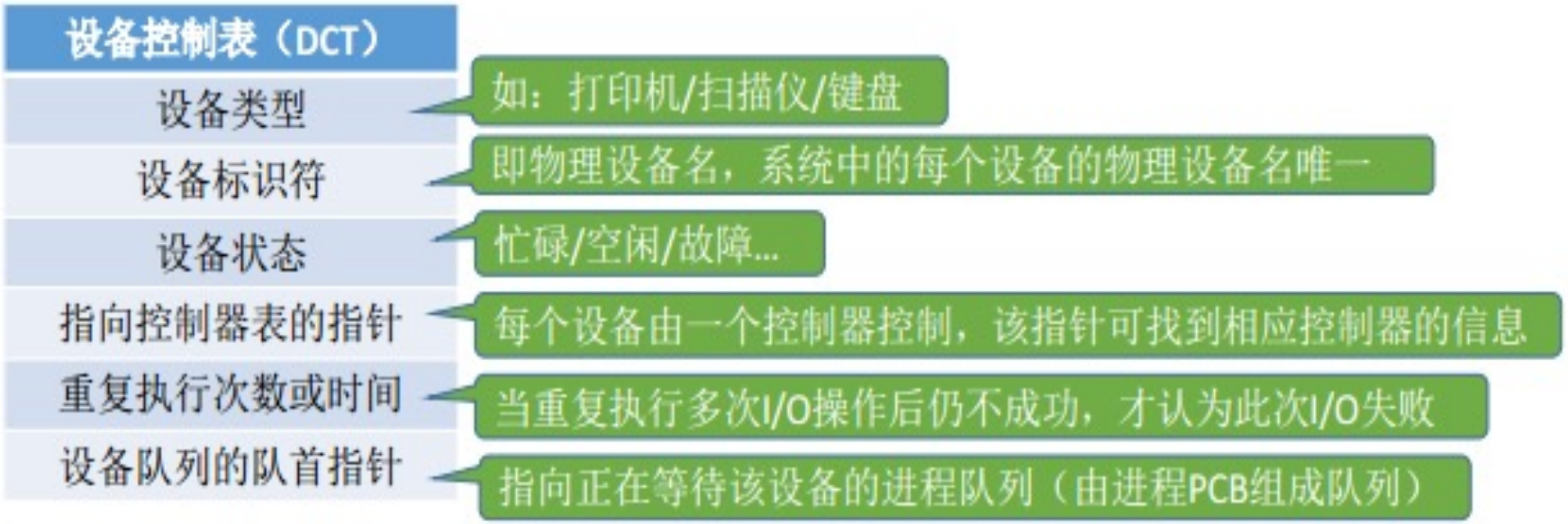
- 设备分配方式
- 设备分配算法
- 设备分配技术

“设备、控制器、通道”之间的关系：



一个通道可控制多个设备控制器，每个设备控制器可控制多个设备。

设备控制表（DCT）：系统为每个设备配置一张DCT，用于记录设备情况



注：“进程管理”章节中曾经提到过“系统会根据阻塞原因不同，将进程PCB挂到不同的阻塞队列中”

控制器控制表、 通道控制表和系统设备表：

控制器控制表（COCT）：每个设备控制器都会对应一张COCT。操作系统根据COCT的信息对控制器进行操作和管理。

通道控制表（CHCT）：每个通道都会对应一张CHCT。操作系统根据CHCT的信息对通道进行操作和管理。

系统设备表（SDT）：记录了系统中全部设备的情况，每个设备对应一个表目。

设备分配的步骤

①根据进程请求的物理设备名查找SDT（注：物理设备名是进程请求分配设备时提供的参数）
②根据SDT找到DCT，若设备忙碌则将进程PCB挂到设备等待队列中，不忙碌则将设备分配给进程。
③根据DCT找到COCT，若控制器忙碌则将进程PCB挂到控制器等待队列中，不忙碌则将控制器分配给进程。
④根据COCT找到CHCT，若通道忙碌则将进程PCB挂到通道等待队列中，不忙碌则将通道分配给进程。

缺点：
①用户编程时必须使用“物理设备名”，底层细节对用户不透明，不方便编程
②若换了一个物理设备，则程序无法运行
③若进程请求的物理设备正在忙碌，则即使系统中还有同类型的设备，进程也必须阻塞等待

改进方法：建立逻辑设备名与物理设备名的映射机制，用户编程时只需提供逻辑设备名

逻辑设备表（LUT）

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

逻辑设备表（LUT）建立了逻辑设备名与物理设备名之间的映射关系。

某用户进程第一次使用设备时使用逻辑设备名向操作系统发出请求，操作系统根据用户进程指定的设备类型（逻辑设备名）查找系统设备表，找到一个空闲设备分配给进程，并在LUT中增加相应表项。

如果之后用户进程再次通过相同的逻辑设备名请求使用设备，则操作系统通过LUT表即可知道用户进程实际要使用的是哪个物理设备了，并且也能知道该设备的驱动程序入口地址。

逻辑设备表的设置问题：

整个系统只有一张LUT：各用户所用的逻辑设备名不允许重复，适用于单用户操作系统

每个用户一张LUT：不同用户的逻辑设备名可重复，适用于多用户操作系统

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

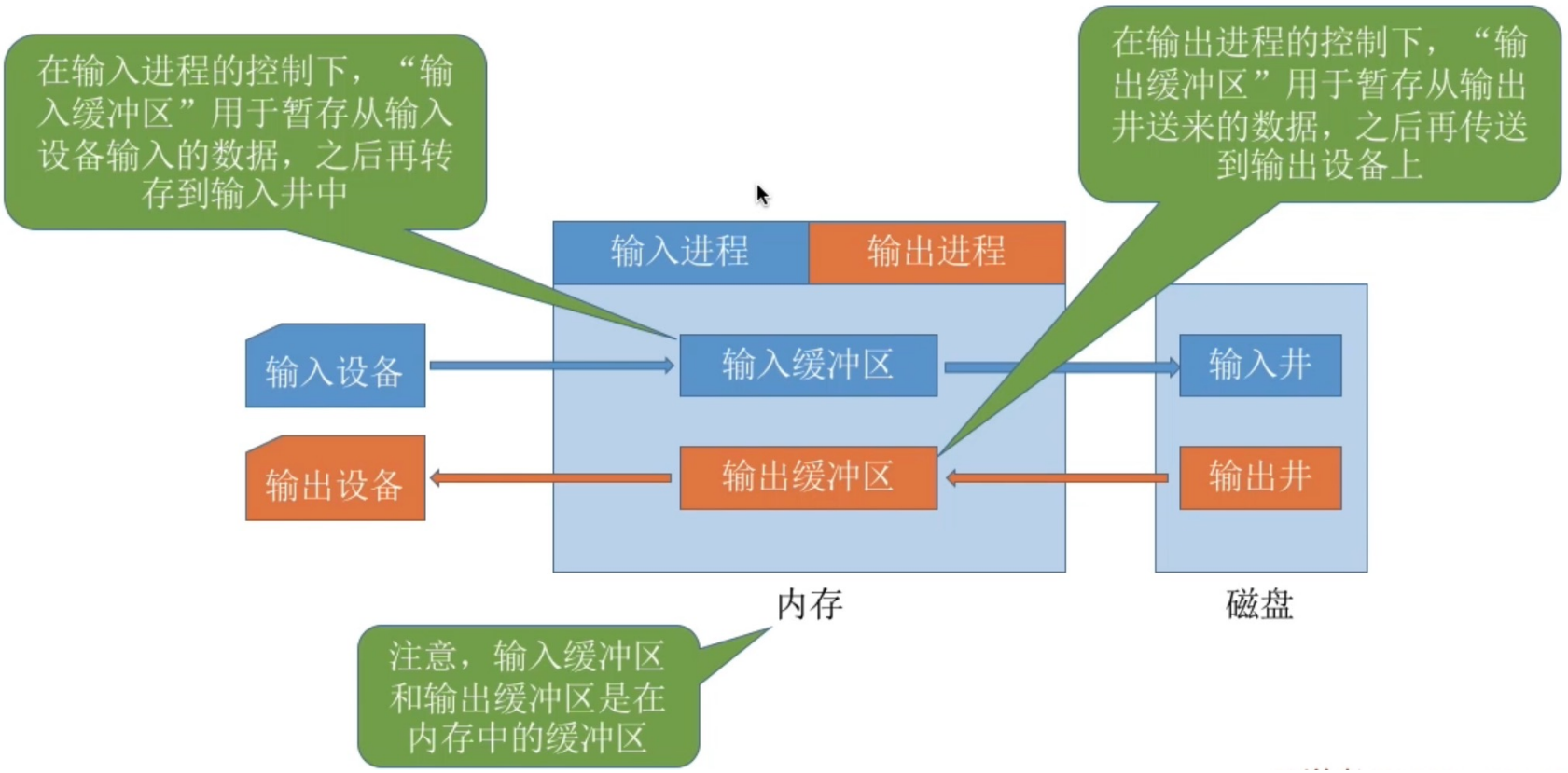
- 共享设备包括磁盘，磁带和磁鼓。
- 对这类设备的分配是采用动态分配的方式进行的，当一个进程要请求某个设备时，系统按照某种算法立即分配相应的设备给请求者，请求者使用完后立即释放。

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

- Simultaneaus Periphernal Operations On-Line(外部设备同时联机操作）。
- 在联机情况下实现的同时外围操作称为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


在输入进程的控制下，“输入缓冲区”用于暂存从输入设备输入的数据，之后再转存到输入井中
在输出进程的控制下，“输出缓冲区”用于暂存从输出井送来的数据，之后再传送到输出设备上
输入进程
输出进程
输入设备
输入缓冲区
输入井
输出设备
输出缓冲区
输出井
内存
磁盘
注意，输入缓冲区和输出缓冲区是在内存中的缓冲区

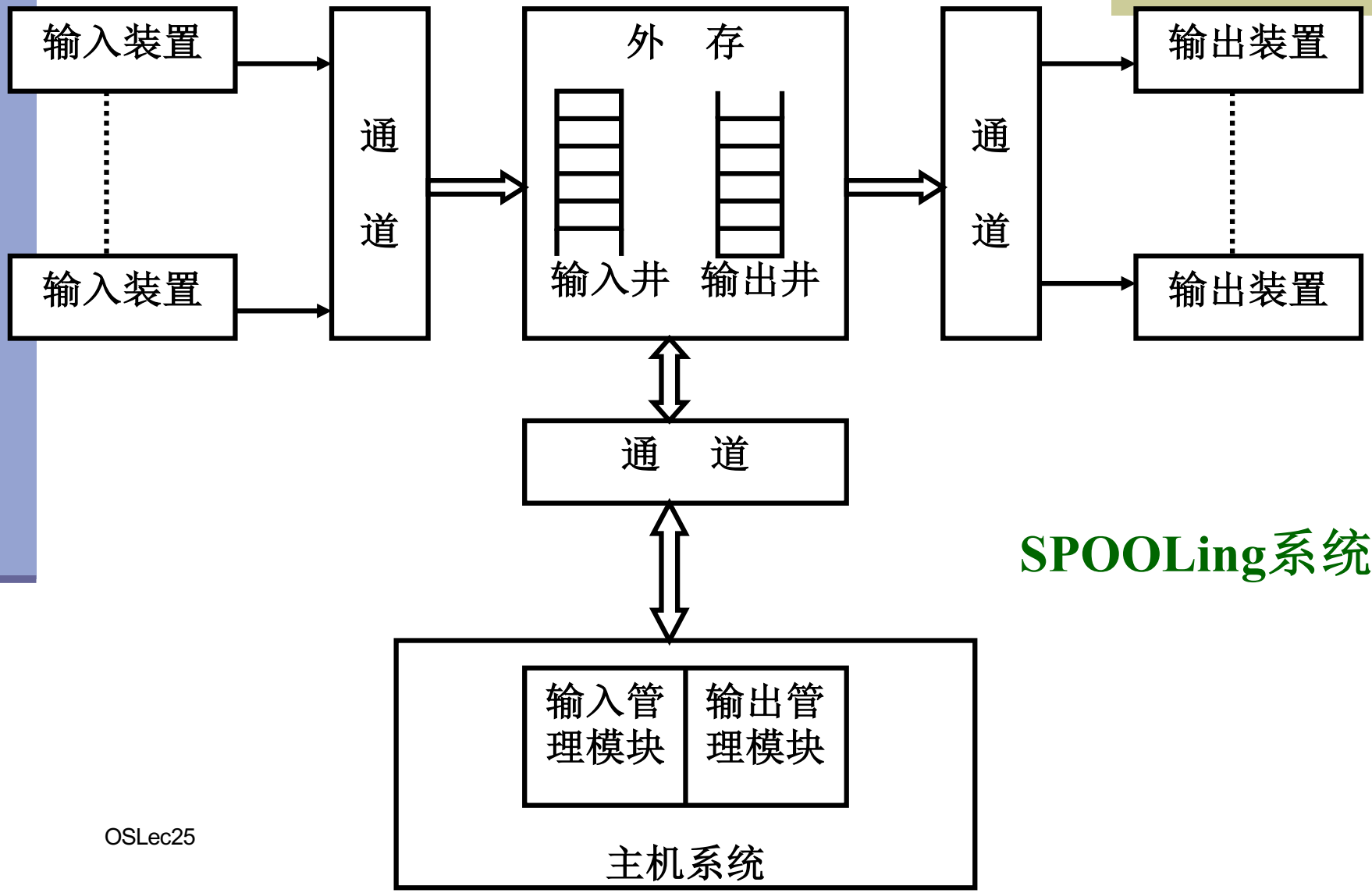
SPOOLing系统工作原理

- 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
- 使外设在CPU直接控制下，与CPU并行工作（假脱机）

图示



SPOOLing系统

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

- 某磁盘的转速为 10 000r/min，平均寻道时间为 6ms，磁盘传输速率为 20MB/s，磁盘控制器时延为 0.2ms，读取一个 4KB 的扇区所需的平均时间约为多少?

- 时延为0.2ms，平均寻道时间为6ms
- 磁盘转速是10 000r/min，平均转1转的时间是6ms，因此查询扇区的平均时间是(1/2)×6ms=3ms。
- 磁盘传输速率是20MB/s，所以读取4KB扇区信息 的时间为4KB/(20MB/s)=0.2ms。
- 因此总时间为 3ms+6ms+0.2ms+0.2ms=9.4ms。

- 假设有11个进程先后提出磁盘I/O请求，当前磁头正在110号磁道处， 并预向磁道序号增加的方向移动。请求队列的顺序为30、145、120、78、82、140、20、42、 165、65，分别用FCFS调度算法和SCAN调度算法完成上述请求，写出磁道访问顺序和每次磁头 移动的距离，并计算平均移动磁道数。

- (1)FCFS调度算法:访问顺序为30、145、120、78、82、140、20、42、165、55、65; 移动距离为80、115、25、42、4、58、120、22、123、110、10;平均移动磁道数为
(80+115+25+42+4+58+120+22+123+110+10)/11=64.45

- (2)SCAN调度算法:访问顺序为120、140、145、165、82、78、65、55、42、30、 20;移动距离为10、20、5、20、83、4、13、10、13、12、10;平均移动磁道数为 (10+20+5+20+83+4+13+10+13+10+10)/11=18.18。

That's all.

Thank you very much!