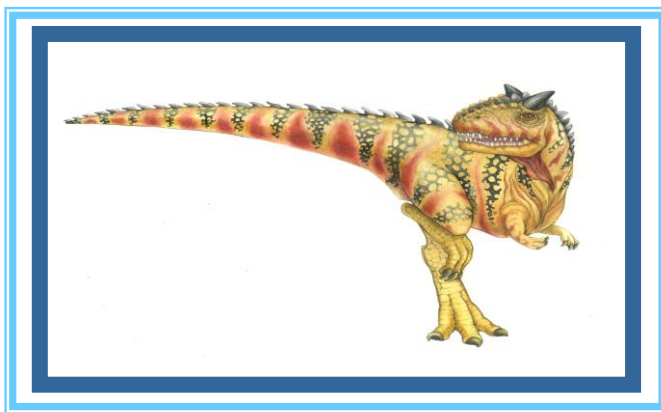


第6章 大容量存储器结构





主要内容

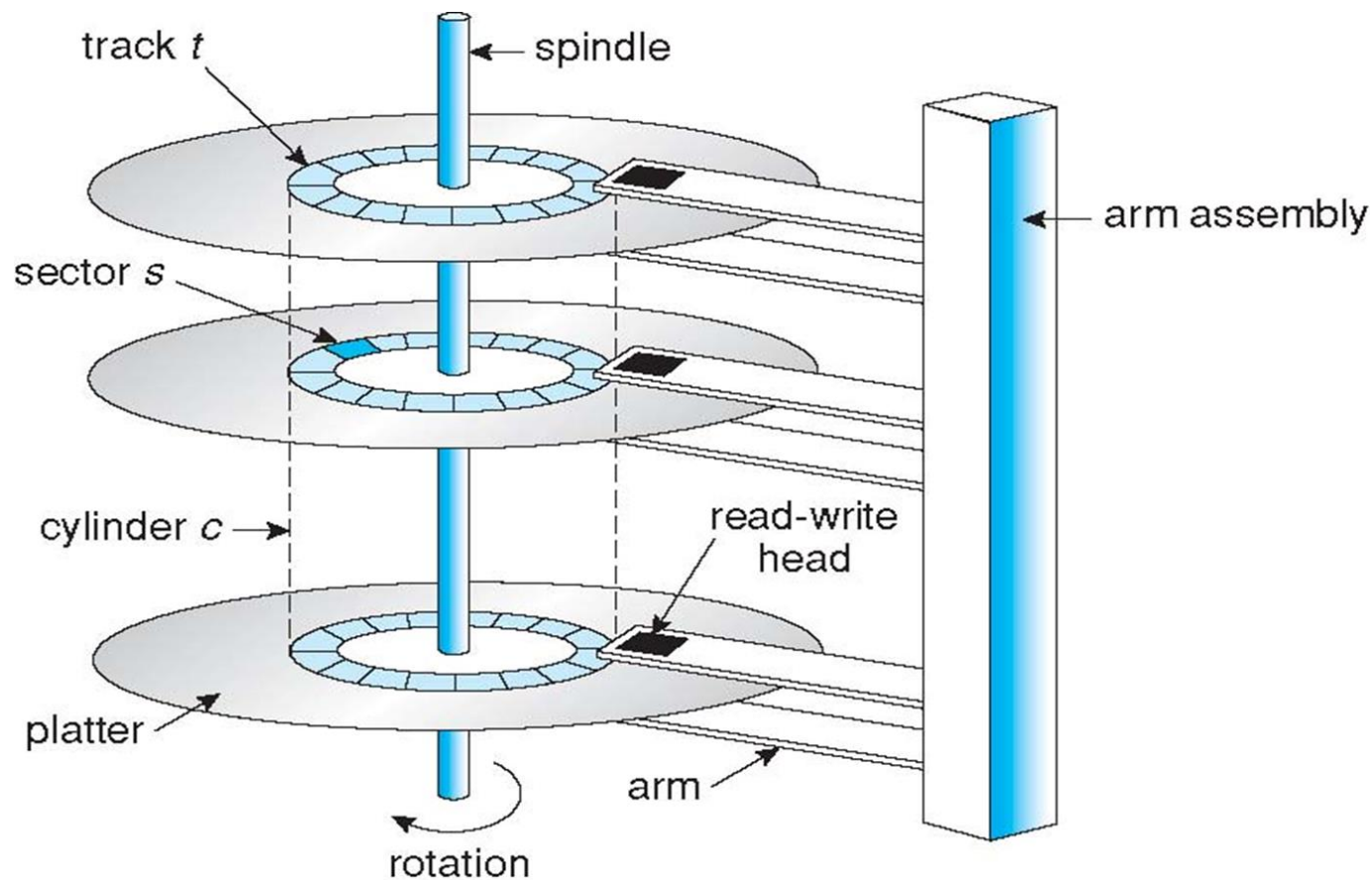
- 磁盘性能简述
- 磁盘调度算法





大容量存储器的结构

- 磁臂 (disk arm)
- 磁道 (track)
- 柱面 (cylinder)
- 扇区 (sector)





磁盘性能简述

■ 扇区

- 是磁盘空间分配和进行I/O传输的数据单位
- 确定一个扇区的三个参数：盘面号、柱面号、扇区号

■ 磁盘的编址方式

- 一般应用时，把磁盘上的所有扇区统一进行编号：0、1、2、.....称为扇区的“逻辑块号”
- 逻辑块号是一维的





磁盘性能简述

■ 扇区各部分的含义

前导码	数据区	ECC
-----	-----	-----

- 前导码

- ▶ 位于每个扇区头部，记录着该扇区的有关信息，比如扇区号

- ECC（纠错码）

- ▶ 检查该扇区的数据区是否已损坏

- 数据区

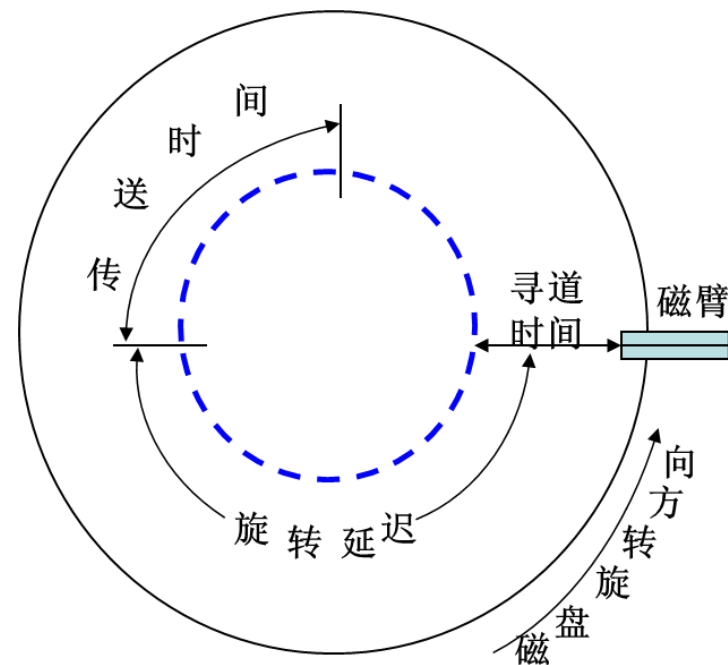
- ▶ 存放数据，其尺寸由进行低级格式化的程序决定，通常有：512B、1KB、4KB





磁盘访问时间

- 随机访问时间(random-access time)
 - 寻道时间(seek time) T_s
 - ▶ 把磁头移动到指定磁道上所经历的时间
 - 旋转延迟时间(rotational latency) T_r
 - ▶ 指定扇区到达磁头下方的平均时间
 - 传输时间 T_t
 - ▶ 磁头进行读/写数据的时间





磁盘访问时间

■ 寻道时间 T_s

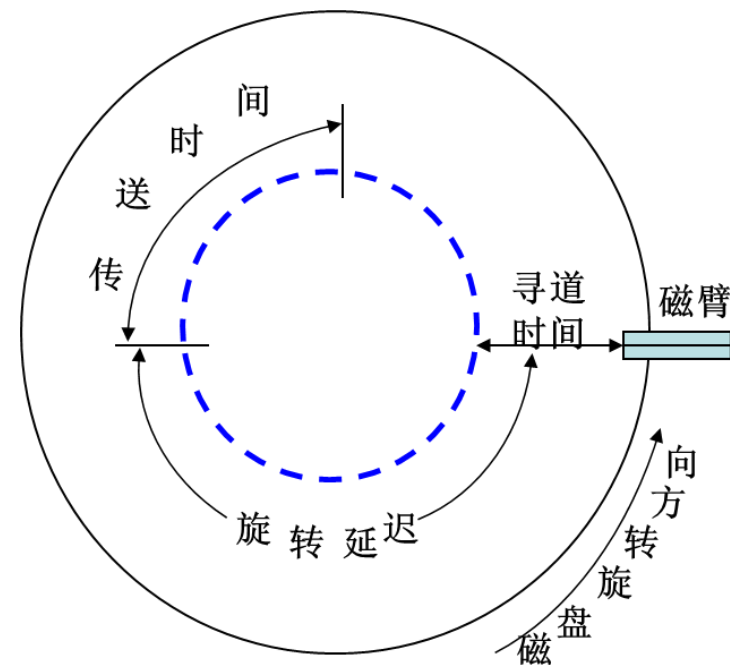
- 是启动磁臂的时间 s 与磁头移动 n 条磁道所花费的时间之和，即

$$T_s = s + m * n$$

- ▶ s : 磁臂启动时间
- ▶ m : 常数，与磁盘驱动器的速度有关
- ▶ n : 磁道数

■ 说明

- 寻道时间将随寻道距离的增加而增大





磁盘访问时间

■ 旋转延迟时间 T_r

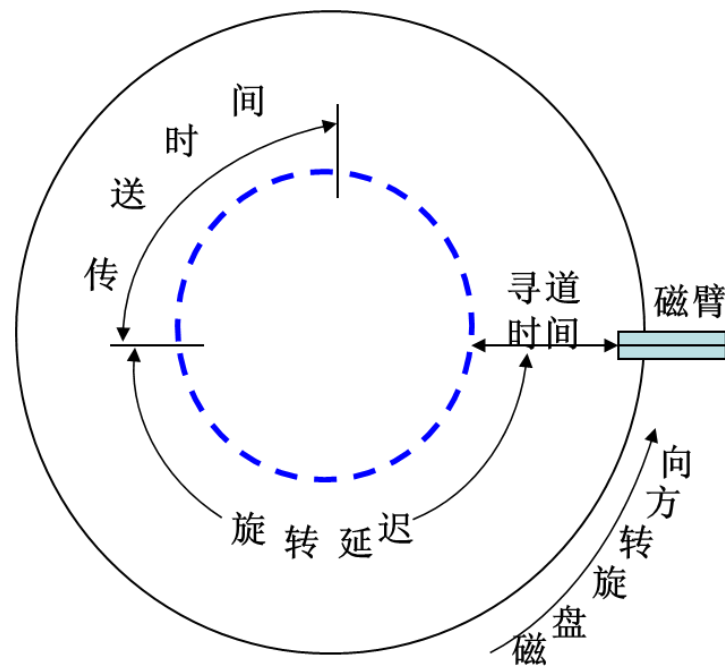
● 平均旋转延迟时间

$$T_r = 1/2r$$

▶ r : 旋转速度

■ 说明

● 平均旋转延迟时间与磁盘的旋转速度有关





磁盘访问时间

■ 数据传输时间 T_t

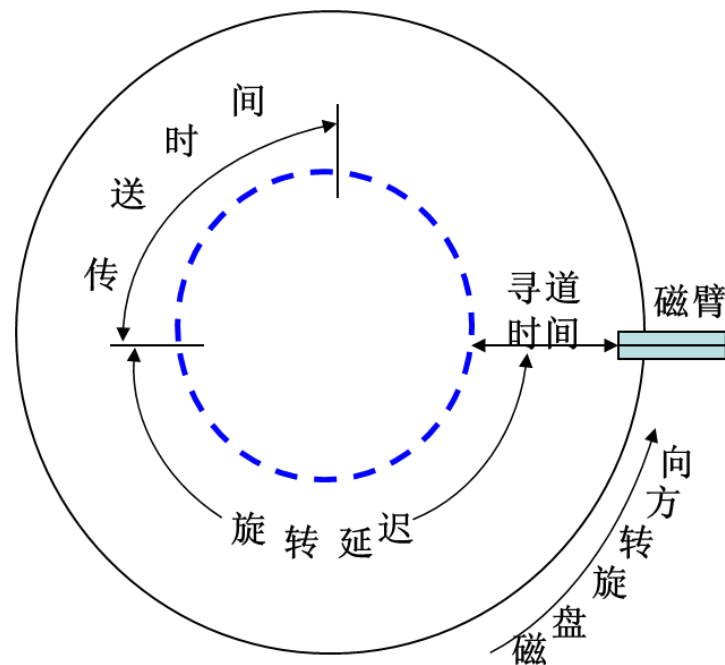
- 与旋转速度和读写的数据量有关

$$T_t = b/rN$$

- ▶ b : 读写字节数
- ▶ N : 一条磁道上的字节数
- ▶ r : 磁盘转速

■ 说明

- 由于磁盘的数据传输率的不断提高，数据传输时间 T_t 通常在磁盘访问时间中占的比例很小





磁盘访问时间

■ 磁盘访问时间 T_a

$$\begin{aligned} T_a &= T_s + T_r + T_t \\ &= m * n + S + 1/2r + b/rN \end{aligned}$$

- 寻道距离是影响磁盘访问时间的的主要因素





小结

- 磁盘结构
- 磁盘性能
 - 磁盘访问时间
 - ▶ 寻道时间
 - ▶ 旋转延迟时间
 - ▶ 传输时间
 - 寻道距离是影响磁盘性能的主要因素





磁盘调度

■ 主要目标

- 通过适当的访问顺序来调度磁盘I/O请求，提高访问速度

■ 磁盘调度算法

- 先来先服务FCFS
- 最短寻道时间优先SSTF
- 扫描算法SCAN
- 循环扫描算法CSCAN
- N-step-SCAN算法
- FSCAN算法





1、先来先服务算法FCFS

■ 算法思想

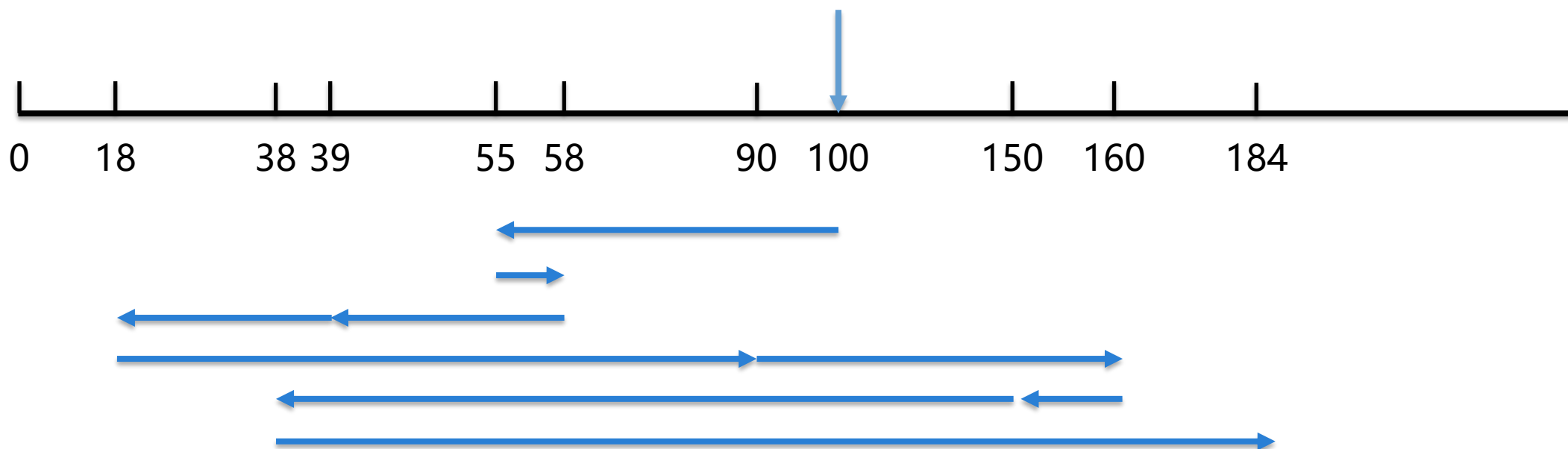
- 以进程对磁盘I/O请求的时间顺序进行磁盘调度





1、先来先服务算法FCFS

- 例：有9个进程提出磁盘I/O请求，请求依次访问磁道：55、58、39、18、90、160、150、38、184，磁头当前的位置是100号磁道。利用FCFS算法算出平均寻道距离



- 平均寻道距离
 - $498/9=55.33$





1、先来先服务算法FCFS

■ 特点

- 简单、公平
- 寻道时间长





2、最短寻道优先算法SSTF

■ 算法思想

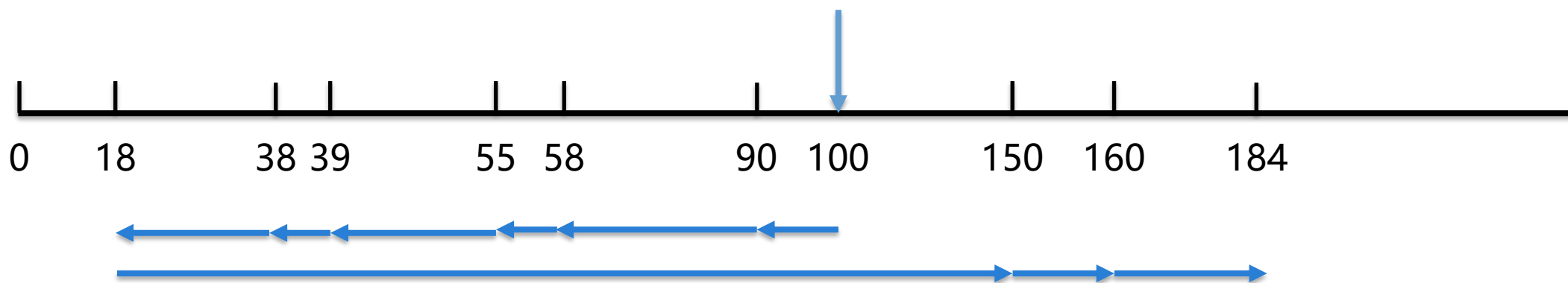
- 把距离磁头当前位置最近的那个I/O请求作为下一次调度的对象





2、最短寻道优先算法SSTF

- 例：有9个进程提出磁盘I/O请求，请求依次访问磁道：55、58、39、18、90、160、150、38、184，磁头当前的位置是100号磁道。利用FCFS算法算出平均寻道距离



- 平均寻道距离
 - $248/9=27.5$





2、最短寻道优先算法SSTF

■ 优点

- 每次寻道时间最短，缩短磁盘访问时间

■ 缺点

- 导致某些进程 “饥饿”





3、扫描算法SCAN

■ 算法思想

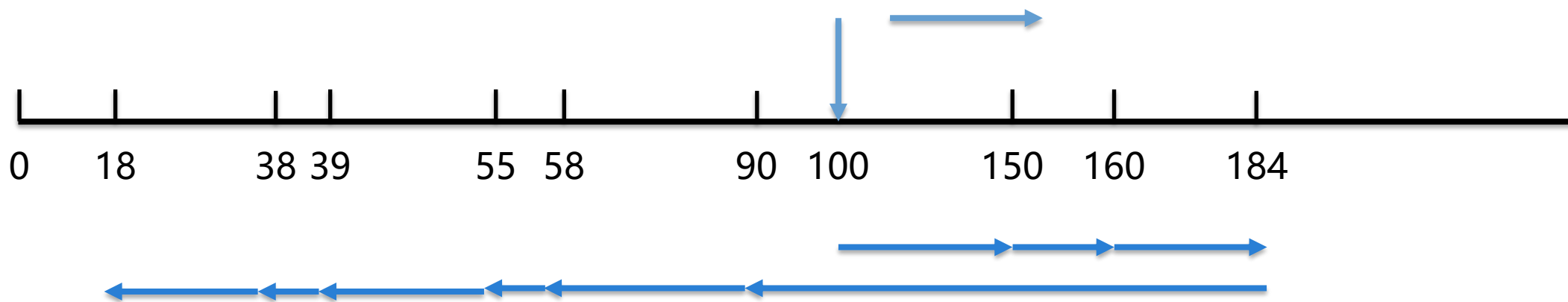
- 也称为“电梯调度算法”
- 磁臂从磁盘的一端向另一端移动，遇到有I/O请求的磁道时就进行处理，直到在这个方向上已没有其他I/O请求
- 随后，移动磁臂反方向移动，同样按顺序扫描完成所有I/O请求





3、扫描算法SCAN

- 例：有9个进程提出磁盘I/O请求，请求依次访问磁道：55、58、39、18、90、160、150、38、184，磁头当前的位置是100号磁道。利用FCFS算法算出平均寻道距离



- 平均寻道距离
 - $250/9=27.77$

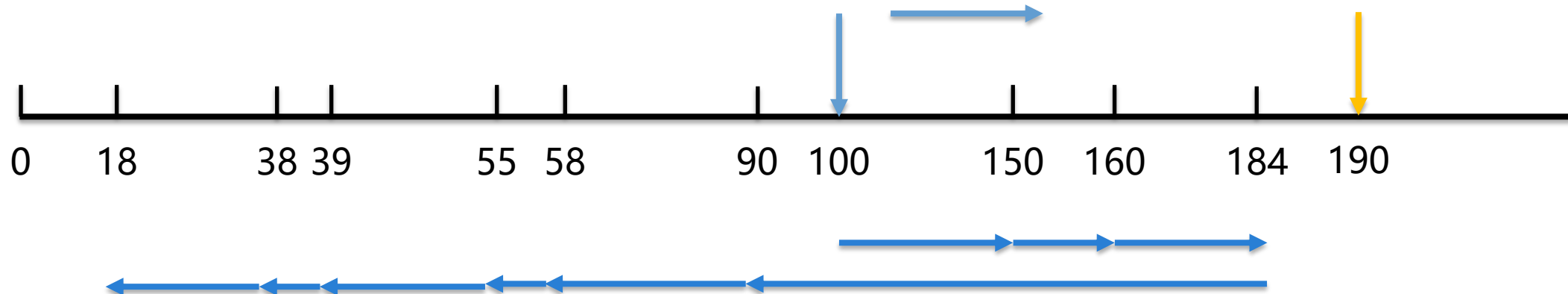




3、扫描算法SCAN

■ 特点

- 避免了“饥饿”现象的发生
- 对于某些特殊情况，寻道时间很长





4、循环扫描算法CSCAN

■ 算法思想

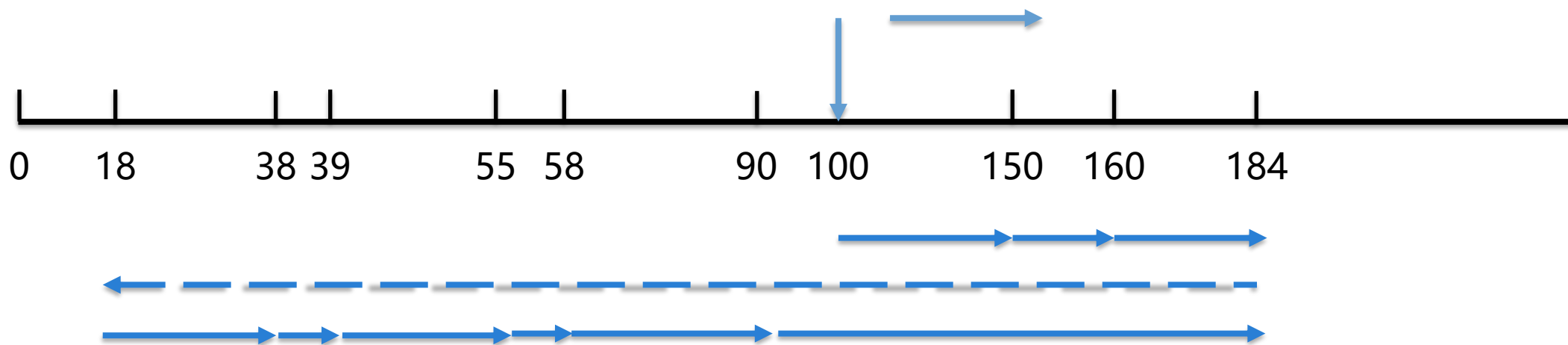
- 把扫描限定在一个方向上，总是从0号磁道开始向内移动磁臂，遇到有I/O请求就进行处理，直到到达最后一个请求磁道
- 然后，磁头直接返回到0号磁道，开始下一次扫描





3、扫描算法SCAN

- 例：有9个进程提出磁盘I/O请求，请求依次访问磁道：55、58、39、18、90、160、150、38、184，磁头当前的位置是100号磁道。利用FCFS算法算出平均寻道距离



- 平均寻道距离
 - $322/9=35.7$





4、循环扫描算法CSCAN

■ 特点

- 避免了SCAN算法带来的某些情况下寻道时间过长的问题





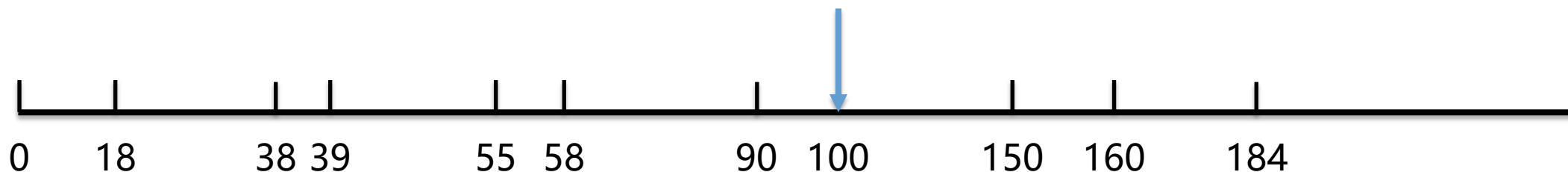
5、N-step-SCAN算法

■ 引入

- 避免“磁臂粘着”现象

■ 磁臂粘着

- 有一个或几个进程对某一磁道反复访问，从而垄断了整个磁盘的现象





5、N-step-SCAN算法

■ 算法思想

- 把磁盘请求队列分成长度为N的子队列
- 每一个子队列采用SCAN算法
- 新到达的磁盘访问请求被添加到新队列中





6、FSCAN算法

■ 算法思想

- N-step-SCAN 算法的简化
- 是只使用两个队列的调度策略



- 开始扫描时，只对已到达的所有请求进行处理
- 新到达的请求排在另一个队列
- 只有等到老请求全部处理完后，才去处理新请求的I/O





习题

- 有一磁盘有100个柱面0~99，当前读写头的位置在48号柱面，并刚刚完成了46号柱面的请求，有一磁盘请求序列：25，90，30，86，43，72，50，45，81。分别利用FCFS算法、最短寻道优先算法和电梯调度算法写出它们的相应调度顺序和读写头移动距离。





小结

- 磁盘调度的目标
- 磁盘调度算法
 - 先来先服务FCFS
 - 最短寻道时间优先SSTF
 - 扫描算法SCAN
 - 循环扫描算法CSCAN
 - N-step-SCAN算法
 - FSCAN算法



End of Chapter 6

