

4_文件管理

4.1_1 初识文件管理

- 文件的属性
 - 文件名，标识符，类型，位置，大小，创建时间，上次修改时间，文件所有者信息，保护信息
- 操作系统提供的功能
 - 创建文件功能（create系统调用）
 - 读文件功能（read系统调用）
 - 写文件功能（write系统调用）
 - 删除文件功能（delete系统调用）
 - 打开文件功能（open系统调用）
 - 关闭文件功能（close系统调用）

4.1_2 文件的逻辑结构

- 无结构文件
 - 文件由一系列二进制文件流组成
- 有结构文件（记录式文件）
 - 顺序文件
 - 物理上顺序存储的顺序文件
 - 文件中的记录一个接一个顺序排列，定长或变长，可以顺序存储或者链式存储
 - 按照是否与关键字顺序有关，可以分为串结构和顺序结构
 - 串结构：记录之间的顺序与关键字无关，通常按照记录存入的时间
 - 顺序结构：记录之间的顺序按关键字顺序排列
 - 链式存储：无法随机存取

- 顺序存储：
 - 可变长记录：无法随机存取
 - 定长记录：可以随机存取，采用串结构，无法快速找到关键字；采用顺序结构，可以快速查找关键字
- 索引文件
 - 索引表本身是定长的顺序文件
 - 主要用于对信息处理的及时性要求比较高的场合
- 索引顺序文件
 - 多级索引表嵌套查找

4.1_3 文件目录

- 文件控制块 (FCB)
 - FCB的有序集合称为“文件目录”，一个FCB就是一个文件目录项
 - FCB包含文件的基本信息，存取控制信息，使用信息
 - FCB实现了文件名和文件之间的映射，即按名存取
 - 操作
 - 搜索、创建文件、删除文件、显示目录、修改目录
- 目录结构
 - 单级目录结构
 - 实现按名存取，但是不允许文件重名
 - 两级目录结构
 - 主文件目录 (MFD) + 用户文件目录 (UFD)
 - 允许不同用户的文件重名
 - 但不能对文件进行分类
 - 多级目录结构 (树形目录结构)
 - 从根目录出发的路径称为绝对路径
 - 当代操作系统采用方法
 - 不便于文件共享
 - 无环图目录结构

- 整个目录成为有向无环图
- 可以用不同的文件名指同一个文件
- 设置共享计数器
- 可以共享
- 索引节点 (FCB改进)
 - 将除了文件名之外的文件描述信息都放到索引结点

4.1_4 文件的物理结构（文件分配方式）

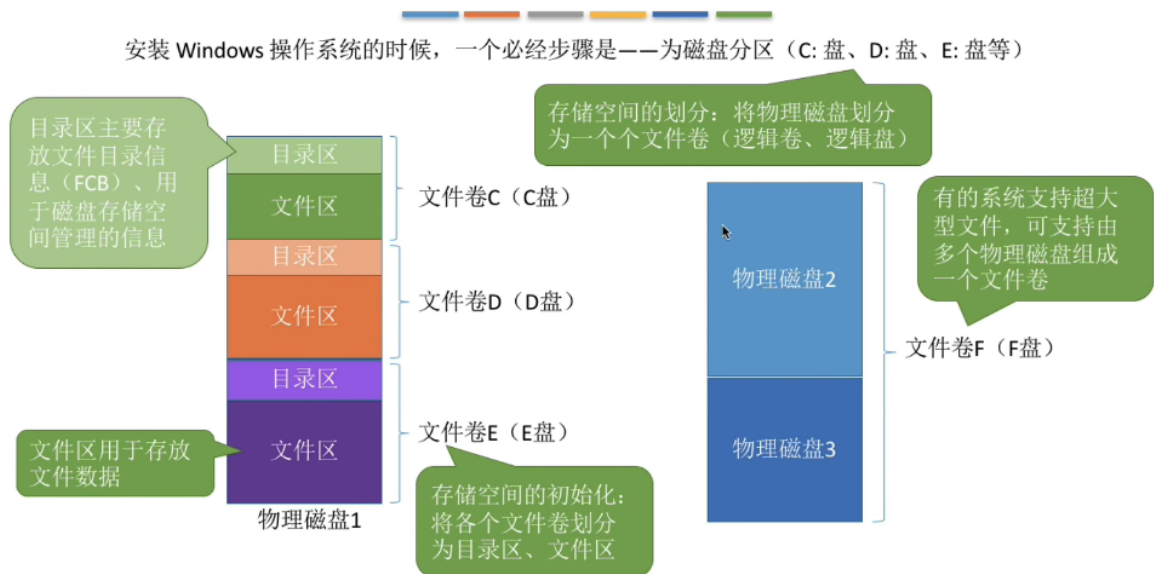
- 文件的分配方式
 - 连续分配
 - 物理块号=逻辑块号+起始块号（检查逻辑块号是否小于长度）
 - 连续分配方式支持顺序访问和直接访问（即随机访问）
 - 每个文件在磁盘上占用一组连续的块
 - 优点
 - 连续分配的文件在顺序读/写时速度最快
 - 缺点
 - 物理上采用连续分配的文件不方便扩展
 - 物理上采用连续分配，存储空间利用率低，会产生难以利用的磁盘碎片
 - 链接分配
 - 隐式链接
 - 只支持顺序访问，不支持随机访问
 - 方便文件的扩展，不会有碎片问题
 - 读入i号逻辑块，总共需要i+1次磁盘I/O
 - 显示链接
 - 存在文件分配表（FAT）
 - 逻辑块号转换成物理块号的过程不需要读磁盘操作
 - 支持顺序访问，也支持随机访问，并不需要依次访问之前的0-i-1号逻辑块

- 不会产生外部碎片，方便扩展
- 分配表需要占用一定的存储空间
- 索引分配
 - 索引分配允许文件离散地分配在各个磁盘块中，系统会为每个文件建立一张索引表，索引表记录了文件的各个逻辑块对应的物理块
 - 支持随机访问
 - 文件扩展易实现
 - 索引表需要占用一定的空间
 - 索引表太大，放不下时：
 - 链接方案
 - 第一个索引块中存储指向下一个索引表的指针
 - 多层索引
 - 建立多层的索引表
 - 各层的索引表大小不能超过一个磁盘块
 - 采用K层索引结构，且顶级索引表为调入内存，则访问一个数据块需要K+1次读磁盘操作
 - 混合索引
 - 多种索引方式的结合，包含直接地址索引，又包含一级间接索引，二级间接索引
 - 直接地址索引----两次读磁盘操作
 - 一级间接索引----三次读磁盘操作
 - 二级间接索引----四次读磁盘操作

	How?	目录项内容	优点	缺点
顺序分配	为文件分配的必须是连续的磁盘块	起始块号、文件长度	顺序存取速度快，支持随机访问	会产生碎片，不利于文件拓展
链接分配	隐式链接 除文件的最后一个盘块之外，每个盘块中都存有指向下一个盘块的指针	起始块号、结束块号	可解决碎片问题，外存利用率高，文件拓展实现方便	只能顺序访问，不能随机访问。
	显式链接 建立一张文件分配表(FAT)，显式记录盘块的先后关系（开机后FAT常驻内存）	起始块号	除了拥有隐式链接的优点之外，还可通过查询内存中的FAT实现随机访问	FAT需要占用一定的存储空间
索引分配	为文件数据块建立索引表。若文件太大，可采用链接方案、多层索引、混合索引	链接方案记录的是第一个索引块的块号，多层/混合索引记录的是顶级索引块的块号	支持随机访问，易于实现文件的拓展	索引表需占用一定的存储空间。访问数据块前需要先读入索引块。若采用链接方案，查找索引块时可能需要很多次读磁盘操作。

4.1_5 文件存储空间管理

• 存储空间的划分与初始化



• 管理方法

- 空闲表法
 - 适用于“连续分配方式”
- 空闲链表法
 - 空闲盘块链
 - 以盘块为单位组成一条空闲链
 - 适用于离散分配的结构
 - 空闲盘区链

- 以盘区为单位组成一条空闲链
- 离散分配，连续分配都使用
- 位示图法
 - 每一个二进制位对应一个盘块，0代表空闲，1代表已分配
 - 如果盘块号，字号，位号从0开始
 - 盘块号 $b=ni+j$
 - b 号盘块号对应的字号 $i=b/n$ ，位号 $j=b\%n$
- 成组链接法
 - 文件卷的目录区中专门用一个磁盘块作为超级块，当系统启动时需要将超级内存块读入内存。并且保证内存与外存中的超级块数据一致。

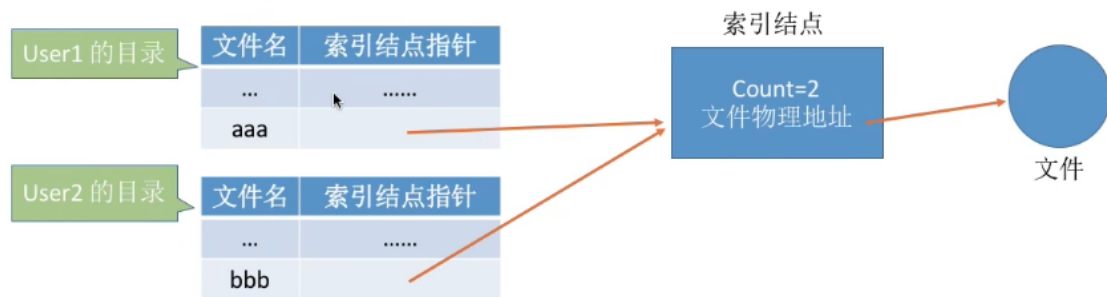
4.1_6 文件的基本操作

- 创建文件（create系统调用）
 - 在外存中找到文件所需的空間
 - 创建该文件对应的目录项
- 删除文件(delete系统调用)
 - 找到文件名对应的目录项
 - 回收文件占用的磁盘块
 - 删除文件对应的目录项
- 读文件(read系统调用)
- 写文件(write系统调用)
- 打开文件(open系统调用)
 - 找到文件名对应的目录项
 - 将目录项复制到内存中的“打开文件”中
- 关闭文件(close系统调用)
 - 将进程的打开文件表相应的表项删除
 - 回收分配给该文件的内存空间等资源

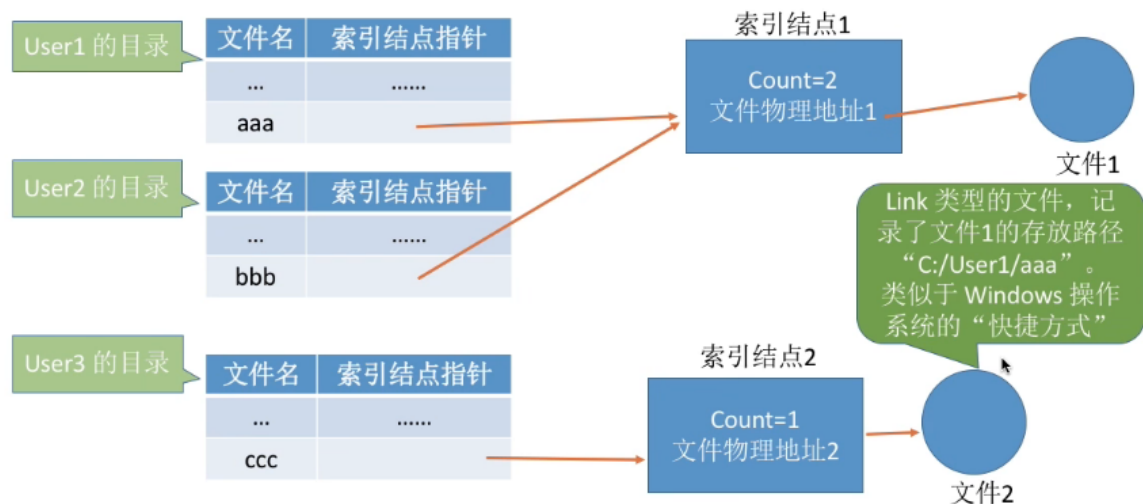
- 系统打开文件表的打开计数器count减一，若count=0，则删除对应表项

4.1_7 文件共享

- 基于索引结点的共享方式（硬链接）
 - 直接指向文件的索引节点



- 不同用户，文件的名字可以不同
- 只有count=0的时候才能真正删除文件数据和检索结点，否则会导致指针悬空
- 基于符号链的共享方式（软链接）
 - 相当于win的快捷方式



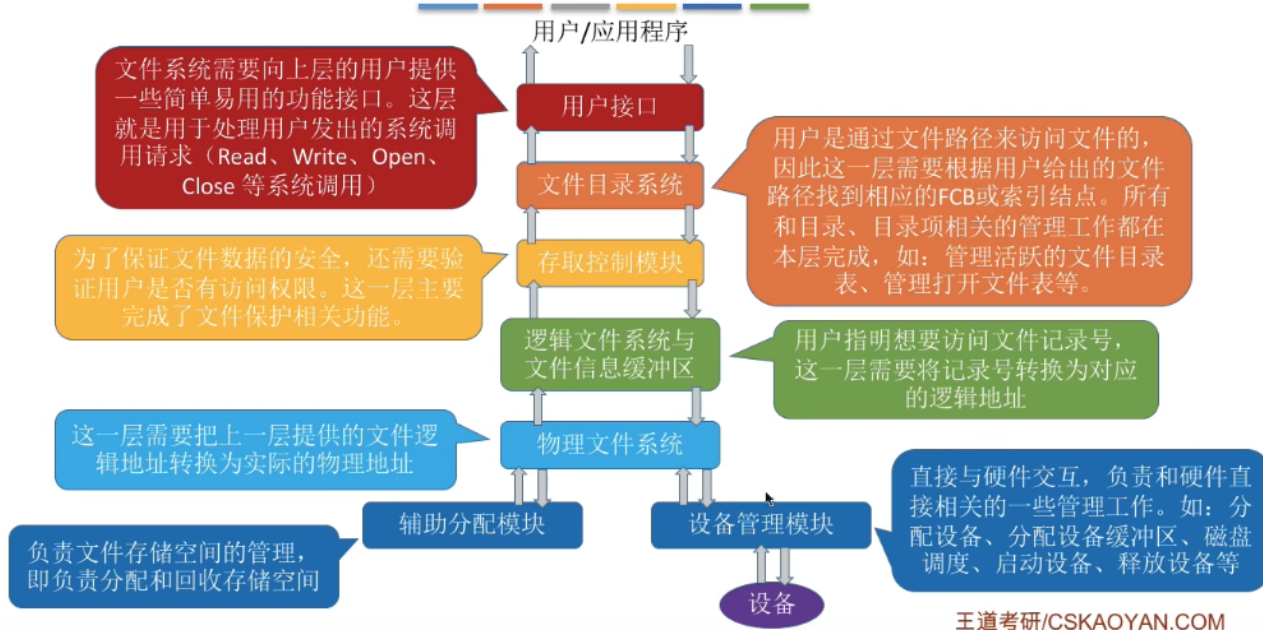
- 创建一个link型的文件，间接查找
- link已经被删除时，Link型文件仍然存在，只是在再次点击的时候会出现查找失败

4.1_8 文件保护

- 口令保护
 - 为文件设置一个口令，口令存在于FCB中或索引节点中，用户请求访问时必须提供相应的口令
 - 优点
 - 保存口令的空间开销不多，验证口令的时间开销也很小
 - 缺点
 - 正确的口令存放在系统内部，不够安全
- 加密保护
 - 使用某个密码对文件进行加密
 - 异或加密
 - 优点
 - 保密性强，不需要在系统中存储“密码”
 - 缺点
 - 编码/译码，需要花费一定时间
- 访问控制
 - 在每个文件的FCB中增加一个访问控制表（ACL），该表记录了各个用户可以对文件执行哪些操作
 - 优点
 - 实现灵活，可以实现复杂的文件保护功能

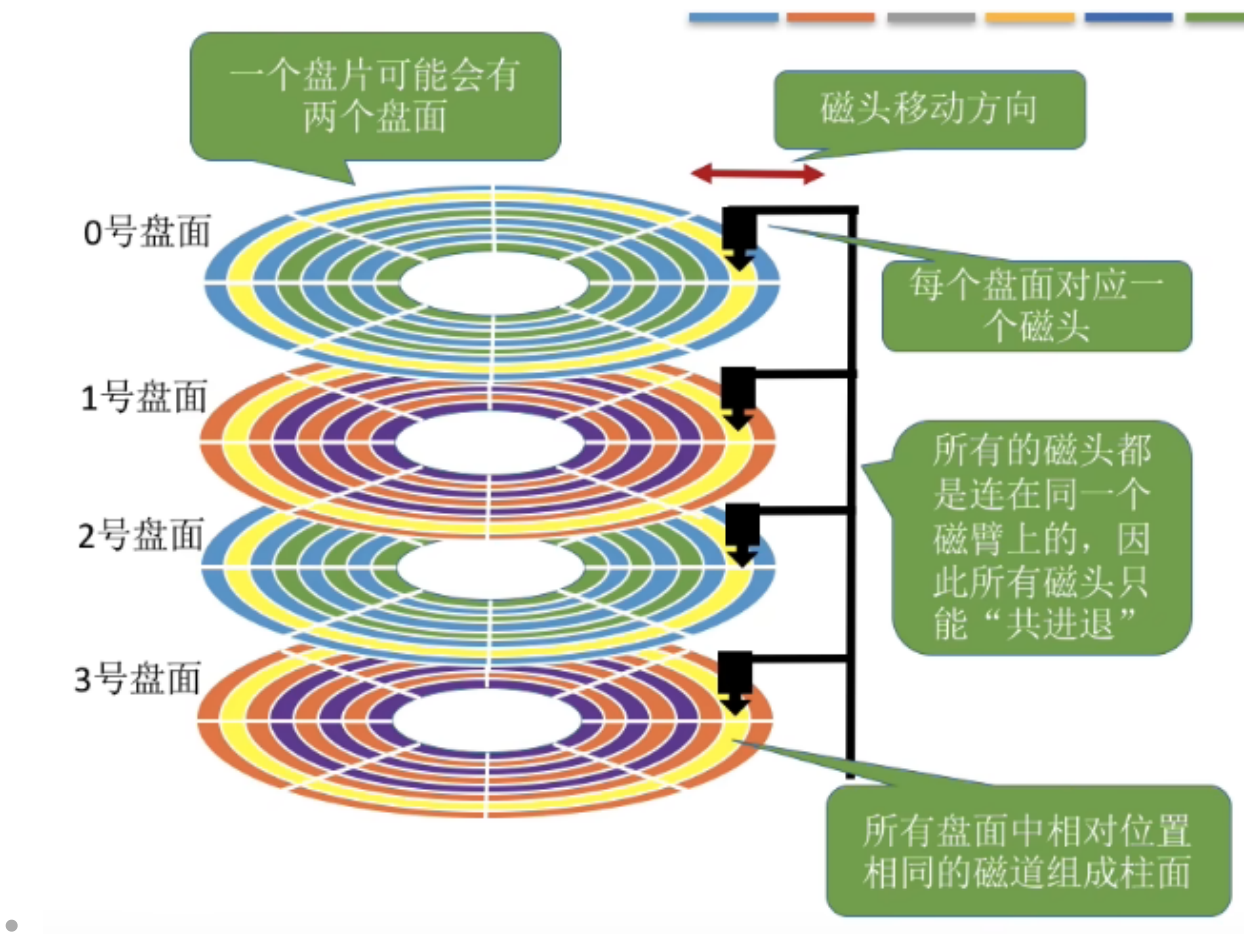
4.1_9 文件的层次结构

文件系统的层次结构



4.2_1 磁盘的结构

- 磁盘、磁道、扇区的概念
 - 磁盘：用磁性物质来记录二进制数据
 - 磁道：磁盘的盘面会被划分成一个一个的磁道（圈）
 - 扇区：磁道又被划一个一个扇区，各扇区存放的数据量相同
- 在磁盘中读写数据
 - 将磁头移动到对应区域，磁盘转动，划过即可读取
- 盘面，柱面的概念



- 磁盘的物理地址
 - 可用柱面号，盘面号，扇区号来定位一个磁盘块
- 磁盘的分类
 - 磁头是否可以移动
 - 活动头磁盘
 - 固定头磁盘
 - 盘面是否可以更换
 - 可换盘磁盘
 - 固定盘磁盘

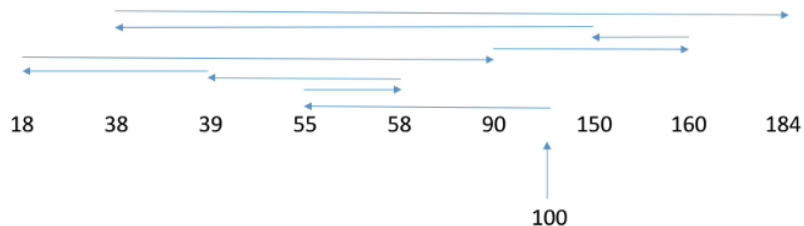
4.2_2 磁盘调度算法

- 一次磁盘读/写操作需要的时间
 - 寻找时间 $T_s = s + m * n$
 - s 为启动磁头臂的时间
 - 每跨越一个磁盘耗时为 m

- 总共需要跨越n条磁道
- 延迟时间 $T_r = 1/(2r)$
 - 磁盘的转速为r
- 传输时间 $T_t = b/(rN)$
 - 此次读/写字节数为b
 - 每个磁道上的字节数为N
- 磁盘调度算法
 - 先来先服务 (FCFS)

- 按照请求访问磁盘的先后顺序进行调度

假设磁头的初始位置是100号磁道，有多个进程先后陆续地请求访问 55、58、39、18、90、160、150、38、184 号磁道
按照 FCFS 的规则，按照请求到达的顺序，磁头需要依次移动到 55、58、39、18、90、160、150、38、184 号磁道

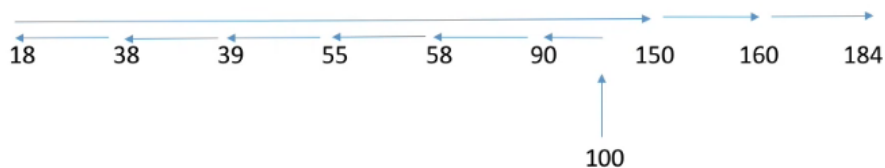


磁头总共移动了 $45+3+19+21+72+70+10+112+146 = 498$ 个磁道

- 响应一个请求平均需要移动 $498/9 = 55.3$ 个磁道 (平均寻找长度)

- 优点
 - 公平
- 缺点
 - 性能不好
- 最短寻找时间优先 (SSTF)
 - 优先处理离当前磁头最近的磁道

假设磁头的初始位置是100号磁道，有多个进程先后陆续地请求访问 55、58、39、18、90、160、150、38、184 号磁道



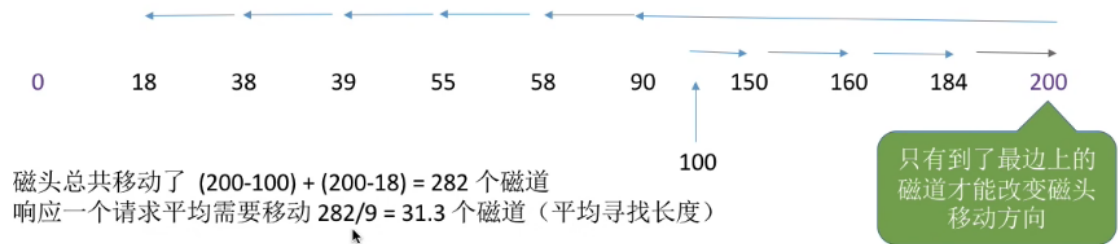
磁头总共移动了 $(100-18) + (184-18) = 248$ 个磁道

响应一个请求平均需要移动 $248/9 = 27.5$ 个磁道 (平均寻找长度)

-
- 优点

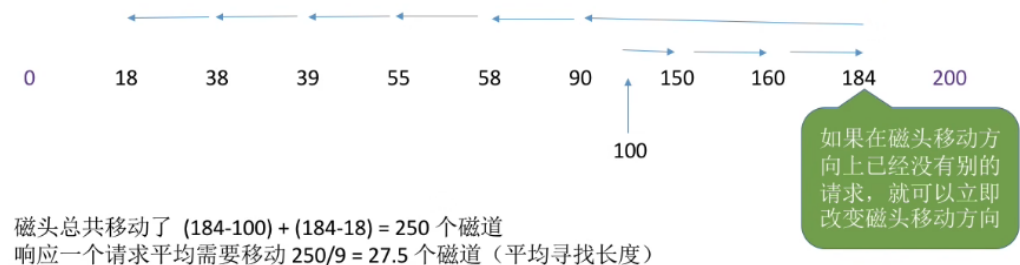
- 性能较好
- 缺点
 - 可能产生饥饿现象
- 扫描算法 (SCAN)
 - 只有磁头移动到最外侧磁道的时候才能往内移动，移动到最内侧磁道的时候才能往外移动

假设某磁盘的磁道为 0~200 号，磁头的初始位置是 100 号磁道，且此时磁头正在往磁道号增大的方向移动，有多个进程先后陆续地请求访问 55、58、39、18、90、160、150、38、184 号磁道



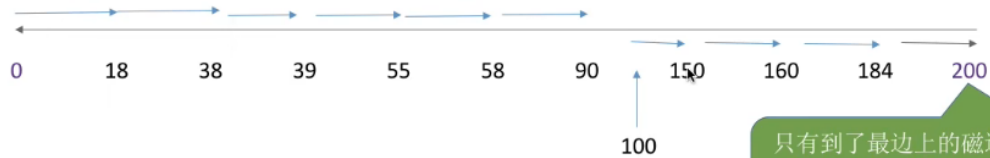
- 优点
 - 性能较好，不会产生饥饿
- 缺点
 - 只有到达最边上才能回头
 - 对各个地方的访问频率不同
- LOOK调度算法
 - 如果在磁头移动方向上已经没有别的请求，就可以立即改变磁头移动方向

假设某磁盘的磁道为 0~200 号，磁头的初始位置是 100 号磁道，且此时磁头正在往磁道号增大的方向移动，有多个进程先后陆续地请求访问 55、58、39、18、90、160、150、38、184 号磁道



- 循环扫描算法 (C-SCAN)
 - 规定只有磁头朝某个特定方向移动才能处理磁道访问请求，返回时直接快速移动到起始端而不做任何处理

假设某磁盘的磁道为 0~200 号，磁头的初始位置是 100 号磁道，且此时磁头正在往磁道号增大的方向移动，有多个进程先后陆续地请求访问 55、58、39、18、90、160、150、38、184 号磁道



磁头总共移动了 $(200-100) + (200-0) + (90-0) = 390$ 个磁道

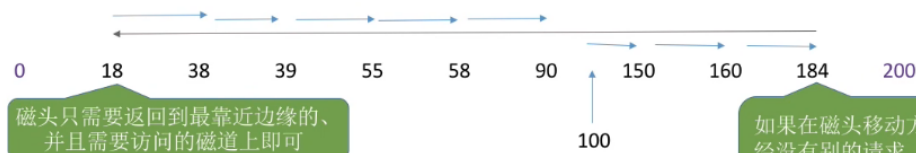
- 响应一个请求平均需要移动 $390/9 = 43.3$ 个磁道（平均寻找长度）

只有到了最边上的磁道才能改变磁头移动方向。磁头返回途中不处理任何请求

• C-LOOK调度算法

- 如果在磁头移动方向上已经没有别的请求，就可以立即改变磁头移动方向

假设某磁盘的磁道为 0~200 号，磁头的初始位置是 100 号磁道，且此时磁头正在往磁道号增大的方向移动，有多个进程先后陆续地请求访问 55、58、39、18、90、160、150、38、184 号磁道



磁头只需要返回到最靠近边缘的、并且需要访问的磁道上即可

磁头总共移动了 $(184-100) + (184-18) + (90-18) = 322$ 个磁道

- 响应一个请求平均需要移动 $322/9 = 35.8$ 个磁道（平均寻找长度）

如果在磁头移动方向上已经没有别的请求，就可以立即让磁头返回

4.2_3 减小磁盘延迟时间的方法

- 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化的时间
 - 磁头读取一块内容后，需要一小段的时间处理
 - 交替编号
 - 错位命名
- 传输时间
 - 读/写 数据花费的时间
- 为什么用（柱面号，盘面号，扇区号）而不是（盘面号，柱面号，扇区号）
 - 在读取地址连续的磁盘时，前者不需要移动磁头

4.2_4 减小磁盘延迟时间的方法

- 磁盘初始化
 - 进行低级格式化（物理格式化），将磁盘各个隧道划分为扇区
 - 将磁盘分区
 - 进行逻辑格式化
- 引导块
 - ROM可以存放初始化程序，ROM中只存放很小的“自举装入程序”
- 坏块的管理
 - 在FAT表上标明（坏块对操作系统不透明）

4.2_5 固态硬盘SSD

