

Lecture 13: 知识 - 磁盘管理

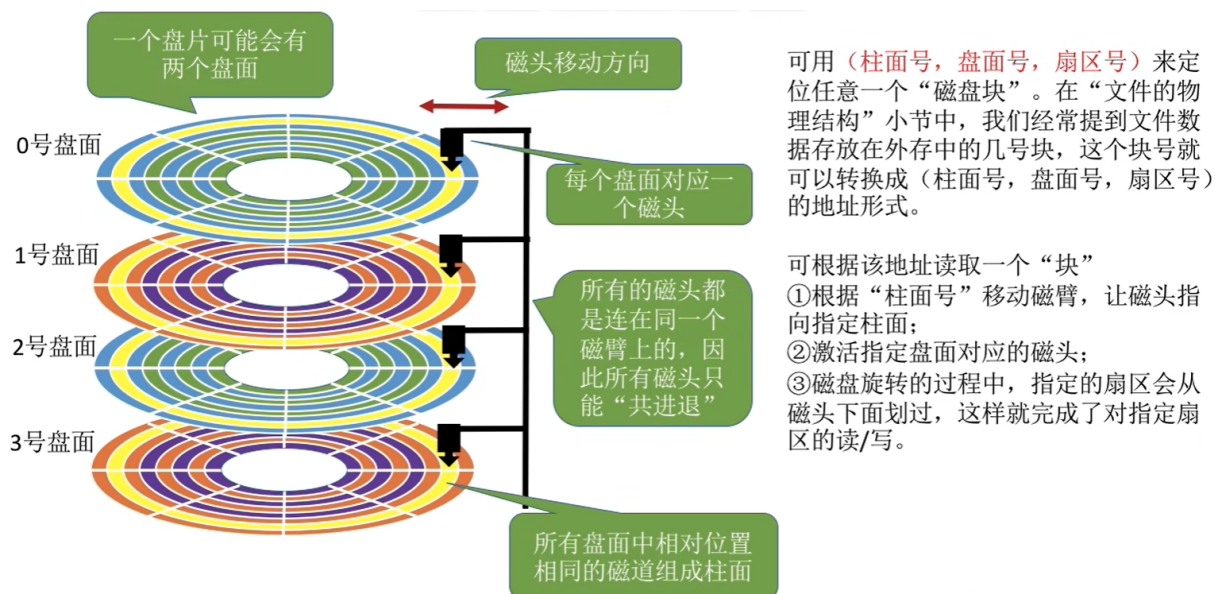
磁盘的结构

概念：磁盘

- 磁盘构成：
 - 磁盘：
 - 磁盘的表面由一些磁性物质组成，可以用这些磁性物质来记录二进制数据；
 - 磁道与扇区：
 - 磁盘的盘面被划分成一个个磁道这样的“圈”就是一个磁道，一个磁道又被划分成一个个扇区，每个扇区就是一个“磁盘块”，各个扇区存放其的数据量相同；
 - 真题：2019年
 - 磁盘存储器的**最小读写单元**为一个**扇区**；
 - 盘面：
 - 一个盘片可能会有两个盘面；
 - 磁头方向：
 - 每个盘面对应一个磁头，所有的磁头都是连在同一个磁臂上的，因此**所有磁头都只能是“共进退”**；
- 数据读取：
 - 概述：
 - 需要把“磁头”移动到想要读/写的扇区所在的磁道；
 - 磁盘会转起来，让目标扇区从磁头下面划过，才能完成对扇区的读/写操作；
 - 过程：
 - 1. 根据“柱面号”前后移动磁臂，让磁头指向指定柱面（因此柱面号也可以称之为磁道号）
 - 注意是前后移动，不是旋转移动。比如将磁头从最外侧磁道移动到最内侧磁道；
 - 2. 激活指定盘面对应的磁头；
 - 选择盘面；
 - 3. 磁盘旋转的过程中，指定的扇区会从磁头下面划过，这样就完成了对指定扇区的读/写。
- 磁盘地址：
 - 结构：
 - (柱面号，盘面号，扇区号)
 - 别称：
 - (磁道号，磁头号，扇区号)
 - 此地址用来定位任意一个“磁盘块”；

- 文件数据存放在外存中的几号块 -> 这个块号就可以转换成 (柱面号, 盘面号, 扇区号) 的地址形式;

图示:



真题: 2022年

题目:

【2022 统考真题】假设某磁盘驱动器中有 4 个双面盘片, 每个盘面有 20000 个磁道, 每个磁道有 500 个扇区, 每个扇区可记录 512 字节的数据, 盘片转速为 7200rpm (转/分), 平均寻道时间为 5ms。请回答下列问题。

1) 每个扇区包含数据及其地址信息, 地址信息分为 3 个字段。这 3 个字段的名称各是什么? 对于该磁盘, 各字段至少占多少位?

解析:

3 个字段的名称为:

- 1. 柱面号 (或磁道号);
- 2. 磁头号 (或盘面号);
- 3. 扇区号;

由于每个面有 20000 个磁道, 因此该磁盘共有 20000 个柱面, 柱面号字段至少占 15 位;

由于该磁盘共有 4 个盘片, 每个盘片有 2 个盘面, 因此一共 8 个盘面, 磁头号字段至少占 3 位;

由于每个磁道有 500 个扇区, 因此扇区号字段至少占 9 位;

概念: 磁盘的分类

磁头移动分类

活动头磁盘:

- 磁头可以移动的称为活动头磁盘;
- 磁臂可以来回伸缩来带动磁头定位磁道;

固定头磁盘:

- 磁头不可以移动的称为固定头磁盘;
- 这种磁盘中每个磁道有一个磁头;

- 盘片更换分类
 - 可换盘磁盘：
 - 盘片可以更换的称为可换盘磁盘；
 - 固定盘磁盘：
 - 盘片不可更换的称为固定盘磁盘；

磁盘性能优化

概念：减少磁盘延迟时间的方法

- 方法一：交替编号
 - 若采用交替编号的策略，即让逻辑上相邻的扇区在物理上有一定的间隔，可以使读取连续的逻辑扇区所需要的延迟时间更小；
- 方法二：优化磁盘地址结构的设计
 - 读取地址连续的磁盘块时，采用（柱面号，盘面号，扇区号）的地址结构可以减少磁头移动消耗的时间；

概念：改善磁盘设备 IO 性能的方法

- 真题：2012年
- (1) 重排 IO 请求次序；
- (2) 预读取以及滞后写；
- (3) 优化文件物理块分布；
- 注意：
 - 设置多个磁盘分区并不会改善磁盘 IO 的性能；

磁盘管理

初始化

概念：磁盘初始化

- 真题：2017年
- 第一步：物理格式化
 - 进行低级格式化(物理格式化)，将磁盘的各个磁道划分为扇区。
 - 一个扇区通常可分为 头，数据区域，尾 3 部分；
 - 包括：
 - 1. 确定扇区校验码所占位数；
 - 2. 划分扇区；
- 第二步：磁盘分区
 - 将磁盘分区，每个分区由若干柱面组成 (即分为我们熟悉的 C 盘，D 盘，E 盘)
- 第三步：逻辑格式化
 - 进行逻辑格式化，创建文件系统。
 - 包括：
 - 1. 创建文件系统的根目录；

- 2. 初始化存储空间管理所用的数据结构（比如位图）；

引导块

概念：初始化程序

- 概念：初始化程序
 - 计算机开机时需要进行一系列初始化的工作，这些初始化工作通过执行初始化程序(自举程序)完成的
 - 初始化程序可以放在 ROM (只读存储器)中。
 - ROM 中的数据在出厂时就写入了，并且以后不能再修改
 - 注意：ROM 一般出厂时就集成在主板上的
 - 在现代计算机 ROM 中只存放很小的自举装入程序
 - 开机时计算机先运行自举装入程序，通过执行程序就可找到引导块，并将完整的自举程序读入内存，完成初始化
- 概念：启动块
 - 完整的自举程序放在磁盘的启动块 (即引导/启动分区)上，启动位于磁盘的固定位置；
 - 拥有启动分区的磁盘称为启动磁盘或系统磁盘 (C: 盘)；

坏块的管理

概念：坏块

- 概念
 - 坏了、无法正常使用的扇区就是坏块；
 - 这属于硬件故障，操作系统是无法修复的；
 - 应该将坏块标记出来，以免错误地使用到它
- 解决
 - 对于简单的磁盘
 - 可以逻辑格式化时 (建立文件系统时)对整个磁盘进行坏块检查，表明哪些扇区是坏扇区
 - 比如：在 FAT 表上标明 (在这种方式中，坏块对操作系统不透明)
 - 对于复杂分磁盘
 - 磁盘控制器 (磁盘设备内部的一个硬件部件) 会维护一个坏块链表。
 - 在磁盘出厂前进行低级格式化 (物理格式化)时就将坏块链进行初始化。
 - 会保留一些备用扇区，用于替换坏块。
 - 这种方案称为扇区备用；
 - 且这种处理方式中，坏块对操作系统透明；

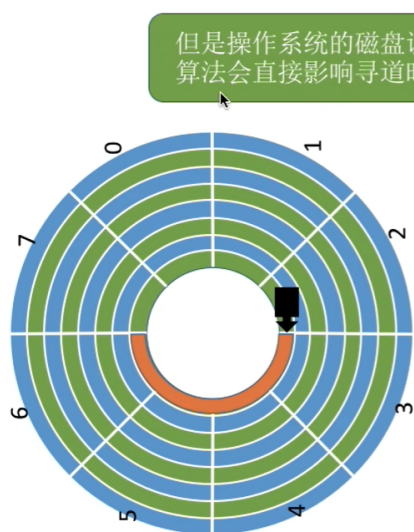
磁盘调度算法

基本概念

概念：磁臂粘着

- **真题：** 2018年
- 概念：
 - 系统总是访问磁盘某个磁道而不响应其他磁道的访问请求，称之为磁臂粘着；
- 不会导致磁臂粘着：
 - **FCFS** 先来先服务；
- 会导致磁臂粘着：
 - **SSTF** 最短寻道时间优先；
 - 假设有大量靠近当前磁道的磁盘访问请求，磁臂就会停留该磁道附近；
 - **SCAN** 扫描算法；
 - **SCAN** 扫描算法和 **CSCAN** 循环扫描算法中，如果持续出现与当前磁头移动方向相同且距离磁头最近的访存请求，有可能出现磁管停留在某处不动的情况；
 - **CSCAN** 循环扫描算法；

概念： 磁盘调度的性能指标



寻找时间（寻道时间） T_s ： 在读/写数据前，将磁头移动到指定磁道所花的时间。

① **启动磁头臂**是需要时间的。假设耗时为 s ；

② **移动磁头**也是需要时间的。假设磁头匀速移动，每跨越一个磁道耗时为 m ，总共需要跨越 n 条磁道。则：

寻道时间 $T_s = s + m * n$

延迟时间 T_R ： 通过旋转磁盘，使磁头定位到目标扇区所需要的时间。设磁盘转速为 r （单位：转/秒，或转/分），则

平均所需的延迟时间 $T_R = (1/2) * (1/r) = 1/(2r)$

传输时间 T_t ： 从磁盘读出或向磁盘写入数据所经历的时间，假设磁盘转速为 r ，此次读/写的字节数为 b ，每个磁道上的字节数为 N 。则：

传输时间 $T_t = (1/r) * (b/N) = b/(rN)$

总的平均存取时间 $T_a = T_s + 1/2r + b/(rN)$

延迟时间和传输时间都与磁盘转速相关，且为线性相关。而转速是硬件的固有属性，因此操作系统也无法优化延迟时间和传输时间

FCFS

概念： 先来先服务算法 (FCFS)

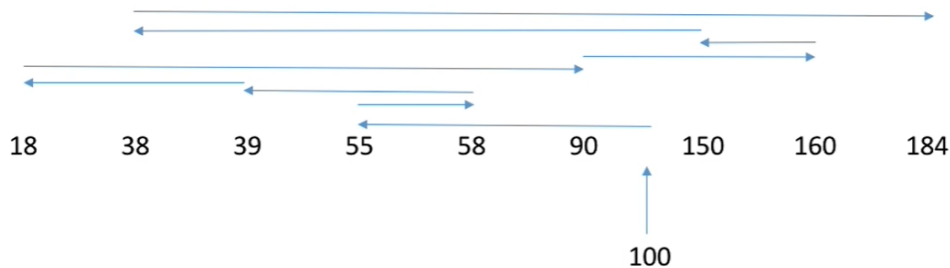
- 思想
 - 根据进程请求访问磁盘的先后顺序进行调度
- 特点
 - 优点
 - **公平**
 - 如果请求访问的磁道比较集中的话，算法性能还算过的去
 - **真题：** 2018年
 - **SSTF、SCAN、C-SCAN** 都可能会产生**磁臂黏着**的现象，比如 **SCAN** 中、如果持续出现和当前方向一致的磁盘请求，则会导致在反方向的请求一直无法获得访问，而在与当前方向一致的方向上持续读取
 - 而 **FCFS** 因为是按照先后顺序进行调度，因此不会产生**磁臂黏着**；

- 缺点
 - 如果有大量进程竞争使用磁盘，请求访问的磁道很分散，则 FCFS 在性能上很差，寻道时间长；

- 图示：

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）

SSTF

概念：最短寻找时间优先（SSTF）

- 思想
 - SSTF 算法会优先处理的磁道是与当前磁头最近的磁道。
 - 可以保证每次的寻道时间最短，但是并不能保证总的寻道时间最短。（其实就是贪心算法的思想，只是选择眼前最优，但是总体未必最优）
- 特点
 - 优点
 - 性能较好，平均寻道时间短
 - 缺点
 - 可能产生饥饿现象
 - 产生饥饿的原因在于：磁头在一个小区域内来回去地移动
- 分析：磁头移动磁道数
 - 题目：
 - 真题：2021年
 - 某系统中磁盘的磁道数为 200（0~199），磁头当前在 184 号磁道上。用户进程提出的磁盘访问请求对应的磁道号依次为 184, 187, 176, 182, 199；
 - 若采用最短寻道时间优先调度算法（SSTF）完成磁盘访问，则磁头移动的距离（磁道数）是多少？
 - 分析：
 - 184 磁道时，最近的磁道号为 182，距离为 2；
 - 182 磁道时，最近的磁道号为 187，距离为 5；
 - 187 磁道时，最近的磁道号为 176，距离为 11；
 - 176 磁道时，最近的磁道号为 199，距离为 23；
 - 199 磁道为最后一个访问序列。因此总距离为 $2+5+11+23=41$

- 答案:
 - 41

SCAN

概念：扫描算法 (SCAN)

- 思想
 - SSTF 算法会产生饥饿，为了防止这个问题可以规定，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。
 - 由于磁头移动的方式很像电梯，因此也叫电梯算法。
- 特点
 - 优点
 - 性能较好，平均寻道时间较短，不会产生饥饿现象
 - 缺点
 - 1. 只有到达最边上的磁道时才能改变磁头移动方向
 - 2. SCAN 算法对于各个位置的磁道的响应频率不平均
- 分析：磁盘访问序列
 - 题目：
 - 真题：2009年
 - 假设磁头当前位于第 105 道，正在向磁道序号增加的方向移动。现有一个磁道访问请求序列为 35, 45, 12, 68, 110, 180, 170, 195，采用 SCAN 调度 (电梯调度) 算法得到的磁道访问序列是什么？
 - 解析：
 - 磁头需要单方向移动到底后，才会调转方向。因此当前按照序号增加的顺序，执行的序列为：
 - 110、170、180、195
 - 执行到 195 后，磁体还需要继续沿着序号增加的方向移动。但是因为这里不知道最大序号是多少，因此不用管这个。执行到 195 之后开始掉头，执行的序列为：
 - 68、45、35、12
 - 答案：
 - 110、170、180、195、68、45、35、12
- 分析：磁头移动磁道数
 - 题目：
 - 真题：2015年
 - 某硬盘有 200 个磁道 (最外侧磁道号为 0)，磁道访问请求序列为 130, 42, 180, 15, 199，当前磁头位于第 58 号磁道并从外侧向内侧移动；
 - 按照 SCAN 调度方法处理完上述请求后，磁头移过的磁道数是多少？
 - 解析：
 - 已知最外侧为 0，所以从外向内移动 = 向磁盘序号增加方向移动；
 - 因此先按照向磁盘序号增加方向移动，得到以下序列：
 - 58 -72-> 130 -50-> 180 -19-> 199

- 移动到当前的最大序列后，需要检查当前是否到了最内侧磁道。因为当前有 200 个磁道，因此磁道范围为 0~199，所以 199 为最内侧磁道。接下来调转磁体，向磁盘序号减少方向移动：
- 199 -157-> 42 -27-> 15
- 所以当前总移动数为 325；
- 答案：
 - 325

C-SCAN

概念：循环扫描算法（C-SCAN）

- 思想
 - 问题：
 - SCAN 算法对于各个位置磁道的响应频率不平均
 - 解决：
 - 规定只有磁头朝某个特定方向移动时才处理磁道访问请求，而返回时直接快速移动至起始端而不处理任何请求；
 - 理解：用扫把扫地。扫的时候处理请求，把扫把恢复原姿势时直接拿起扫把；
 - 注意：
 - 计算磁头移动距离时，直接快速移动至起始端这一过程的距离也需要纳入计算；
- 特点
 - 优点
 - 比起 SCAN 来，对于各个位置磁道的响应频率很平均；
 - 缺点
 - 只有到达最边上的磁道时才能改变磁头移动方向，另外，比起 SCAN 算法来，平均寻道时间更长；
- 分析：真题：2024年
 - 题目：

某磁盘的磁道数为 400（磁道号为 0 ~ 399），采用循环扫描算法（CSCAN）进行磁盘调度，完成对 200 号磁道的请求后，磁头向磁道号减小的方向移动。若还有 7 个磁盘请求，对应的磁道号分别为：300, 120, 110, 0, 160, 210, 399，则完成上述磁盘访问请求后磁头移动的距离是（ ）。

 - A. 599 B. 619 C. 788 D. 799
 - 解析：
 - CSCAN 算法只在一个移动方向上响应磁盘访问请求；
 - 由题可知，由于仅在磁头向磁道号减少的方向访问磁盘，因此磁头访问序列为：
 - 1. 200、160、120、110、0，一共移动了 200；
 - 2. 从 0 移动到 399，移动了 399；
 - 399、300、210，移动了 189
 - 磁头移动距离 = 200+399+189 = 788

LOOK

概念： LOOK 调度算法

- 思想：
 - 问题
 - 扫描算法 SCAN 中，只有到最边上的磁道时才能改变磁头移动方向
 - 解决：
 - 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方向。边移动边观察，因此叫 LOOK；
- 特点
 - 比起 SCAN 算法来，不需要每次都移动到最外侧或最内侧才改变磁头方向，使寻道时间进一步缩短；

C-LOOK

概念： C-LOOK 调度算法

- 思想：
 - 问题：
 - C-SCAN 算法的主要缺点时只有到达最边上的磁道时才能改变磁头移动方向，并且磁头返回时不一定需要返回到最边缘的磁道上。
 - 解决：
 - 规定如果磁头移动的方向上已经没有磁道访问请求了，就可以立即让磁头返回，并且磁头只需要返回到有磁道访问请求的位置即可。
- 特点
 - 优点
 - 比起 C-SCAN 算法来，不需要每次都移动到最外侧或最内侧才改变磁头方向，使寻道时间进一步缩短。

总结

- 注意：若无特别说明，也可默认 SCAN 算法和 C-SCAN 算法为 LOOK 调度和 C-LOOK 调度

	优 点	缺 点
FCFS 算法	公平、简单	平均寻道距离大，仅应用在磁盘 I/O 较少的场合
SSTF 算法	性能比“先来先服务”好	不能保证平均寻道时间最短，可能出现“饥饿”现象
SCAN 算法	寻道性能较好，可避免“饥饿”现象	不利于远离磁头一端的访问请求
C-SCAN 算法	消除了对两端磁道请求的不公平	—