

## 文件系统总结

### 逻辑文件

- 顺序文件

- 直接文件

- 索引文件

- 逻辑文件的访问要求

### 目录文件

- FCB (file control block)

- 目录结构

- 文件路径

- 应用

### 物理文件

- 连续分配

- 链接分配

- 索引分配

- Unix inode

### 应用题

- 磁盘I/O类型

- 文件系统设计题

- 根据字节和不同分配方式来计算I/O次数

# 文件系统总结

逻辑文件面向用户，学习时应该掌握不同的逻辑文件之间的特点

目录文件是链接逻辑文件和物理文件的桥梁，学习时应该体会到目录文件时如何优化访问时间

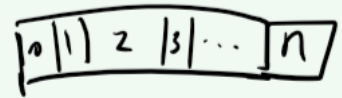
物理文件是指文件的数据如何存储在磁盘等存储设备上，主要内容就是分配方式，掌握不同的分配方式特点

## 逻辑文件

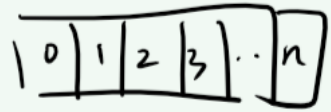
逻辑文件是对用户而言，包括顺序文件，直接（随机）文件，索引文件

## 逻辑文件:

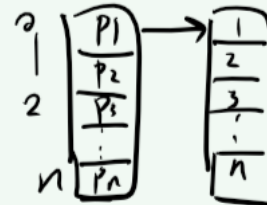
顺序: 不等长, 无法随机访问



直接: 等长, 支持随机访问



索引: 建立索引表:



## 顺序文件

顺序文件是最常用的文件

优缺点	记录	访问方式	占空间	读取所需数据	应用场景
优点存储空间小, 缺点访问效率差	每条不等长	顺序访问	最小	最大	大文件, 对记录批量存取

## 直接文件

直接文件记录定长, 支持随机访问和顺序访问, 访问时间短, 访问需要读入的数据最小, 但是占空间大

优缺点	记录	访问方式	占空间	读取所需数据	应用场景
优点访问效率高, 却缺点浪费存储空间	每条记录所占空间等长	直接访问	最大	最小	需要直接访问, 要求访问速度的

# 索引文件

索引文件是添加了**索引表**的顺序文件，用额外的空间来实现顺序文件随机访问

**索引表**是经常提到的名词，注意辨析到底是索引文件的索引表还是索引分配的索引表

优缺点	记录	访问方式	占空间	读取所需数据	应用场景
优点存储空间和访问效率适中，缺点是索引表可能占很大空间	每条不等长，但索引表记录等长	顺序访问，直接访问	适中	适中	存储空间不足又需要直接访问，追求访问速度

## 逻辑文件的访问要求

顺序访问：访问第*i*条记录的时候要读入*i-1*条记录（符合物理分配中的隐式链接分配）

直接（随机）访问：访问第*i*条记录的时候无需多读入其它记录

## 目录文件

目录文件是链接逻辑文件和物理文件的桥梁

目录文件的统一形式：

每个目录都有自己的目录文件

目录：

10111

1

2

⋮

n

⋮

→ 文件目录项  
(一般为FLB)

## FCB (file control block)

操作系统将文件的信息都存储在FCB中，FCB相当于一个文件的身份证，其中还有文件在磁盘中的存储地址。

文件目录的每一项都是文件目录项，在大多数系统中文件目录项就是FCB。由于FCB文件可能很大，所以操作系统把文件主要信息存放在索引结点中，索引结点在文件目录项中只保留指向索引结点的指针，这样做目录可以存放的项数变多

目录文件是指存储当前目录下的FCB的文件

一般查找当前目录下的某个文件，先查找文件目录，再根据里面的FCB得到文件在磁盘中的首地址，然后再访问磁盘。题目中都会提到FCB在内存/磁盘中，根据FCB不同有不同的解决方式

## 目录结构

指的是操作系统的目录结构

常见的目录结构有单级目录，二级目录，树形目录，无环图目录

## 文件路径

分为绝对路径和相对路径。

绝对路径是指从根目录到当前文件的路径

相对路径是指当前打开目录到当前文件的路径

假设当前目录有 $n$ 个文件，一个文件目录项占一个物理块，则访问一个文件的平均读取物理块数是  $(n+1)/2$

## 应用

如果想要降低访问文件造成的读取物理块数，有以下：

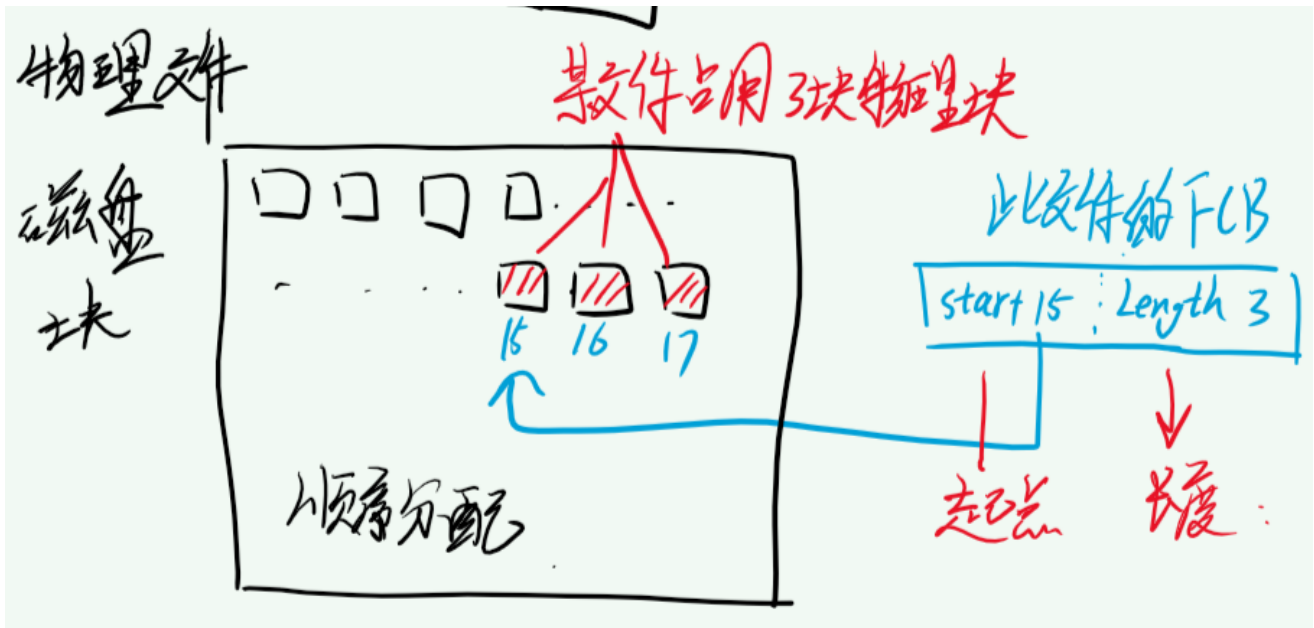
1. 设置当前目录
2. 使用索引结点
3. 在内存中设置打开文件表，里面直接存放着打开文件的文件指针
4. 把目录(FCB)常驻内存，如果某个目录常驻内存，指的是该目录下所有文件信息都常驻内存，例如A目录下有a,b,c,d，那么读入a~d都不用访问磁盘块！

## 物理文件

物理文件指的就是文件在磁盘等存储设备上以什么样的方式进行存储。

常见的分配方式有连续分配，链接分配（隐式，显示），索引分配，Unix inode分配

## 连续分配



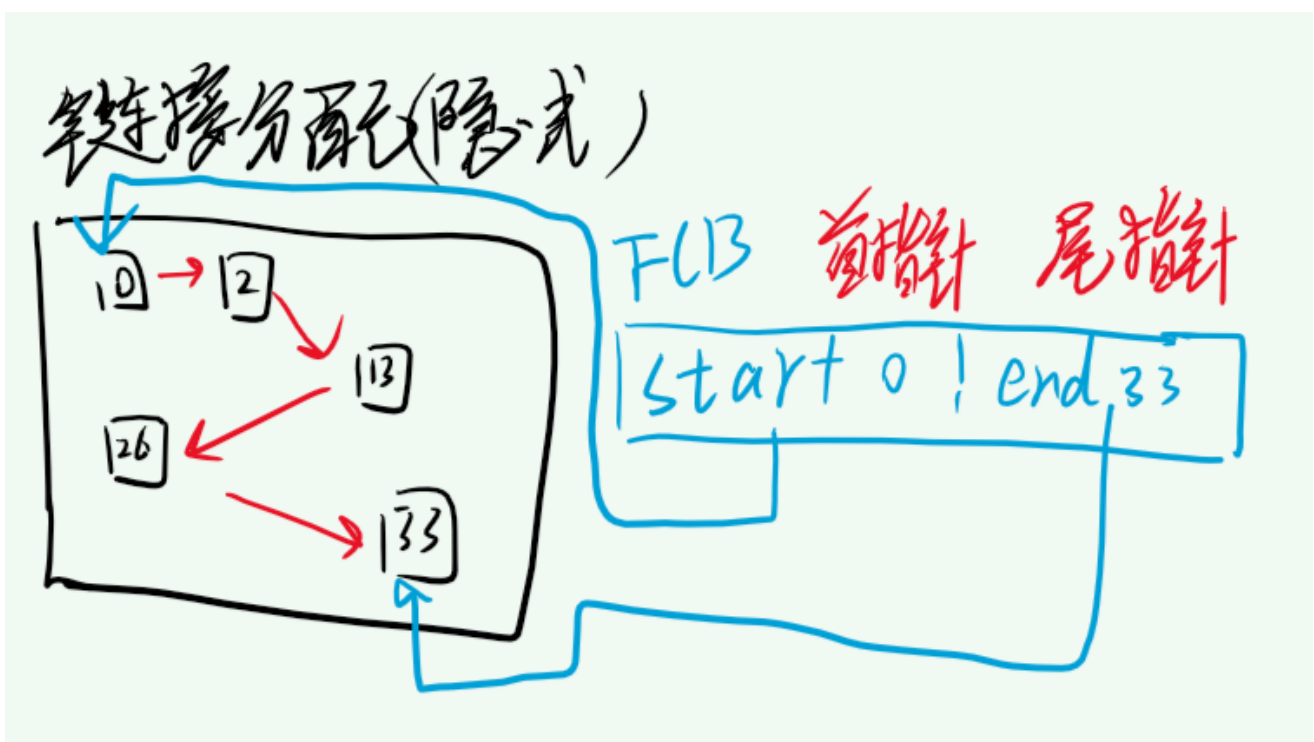
把磁盘块连续的存放在磁盘中，此时的FCB结构是（文件名，首块地址，长度）。

这是分配方式中唯一要求连续的存储空间

- 优点：支持随机访问，访问速度快，
- 缺点：浪费空间，文件不能动态增长，不利于文件的插入和删除
- 应用：一次性写入的文件，要求随机访问的文件

## 链接分配

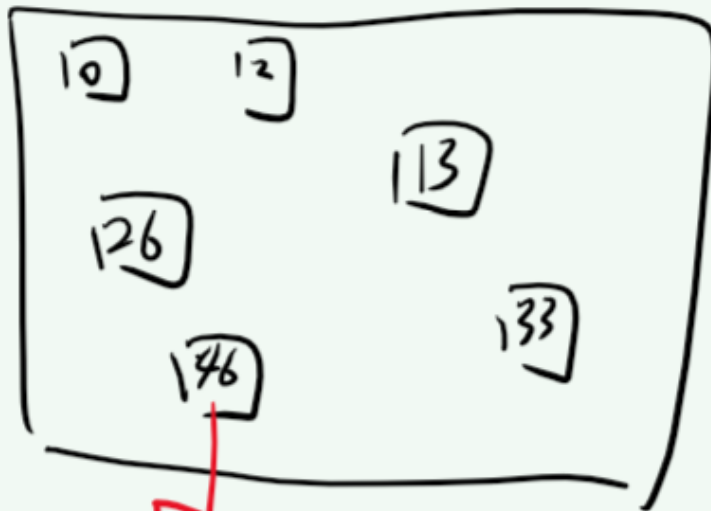
隐式：将存放文件的物理块离散存储，每个物理块有指针指向它下一个物理块。此时FCB结构是（文件名，首块地址指针，尾块地址指针）



- 优点：提高磁盘利用率，可以动态扩充文件大小，便于文件的插入和删除
- 缺点：无法实现**随机访问**，访问文件速度慢，可靠性差

显式：建立一张FAT表，此时FCB格式是（文件名，FAT中的起始块号）

## 链接分配显式



存级FAT(文件分配表)，一个目录一张FAT.

FAT: 物理块号 下一块.

该文件的FCB

文件名	起始块号
aac	0



- 优点：优点同上，还支持了**随机访问**（这里的随机访问是指先查一次FAT，命中了可以直接访问物理块）
- 缺点：不适合大容量磁盘，FAT表为了表示所有地址，可能占很大的空间

## 索引/分配

类似于内存管理中的分页和分段管理，用索引表来管理指针

索引分配为每个文件分配一个单独的索引表，此时FAT格式是（文件名，索引表的地址）

访问时先查FCB再找索引表，根据题目描述有不同的解题方法

假设一级索引表的指针是4B（32位，能表示0到  $2^{32} - 1$  大小的地址），假设一个物理块大小是4KB，则能存储  $2^{10}$  个指针，即指向  $2^{10}$  个物理块，那么一级索引表最大能表示的文件大小是  $4 * 2^{10} \text{KB} = 4\text{MB}$

为了突破这一限制，可以为一个文件引入多张索引表，多张索引表可以用首尾相连的指针链接起来（不常用），也可以用顶级索引表管理所有索引表

- 优点：数据可以离散存放，更高的空间利用率
- 缺点：索引指针占空间，无论如何都要分配存放索引表的索引块

这里经常考的是索引分配下的最多/最少I/O次数

一般来说最少的情况永远在i级索引表里面读入第1块物理块，然后根据地址读入即可总共i+1次读入

最多的情况是i级索引表里面最后一个物理块，然后再读入文件。

## Unix inode

混合索引，弄明白链接分配就能看懂这个

## 应用题

### 磁盘I/O类型

这里一般考的是对文件的某个位置进行插入，删除，增加要进行多少次IO

解题流程

查找该文件的FCB

FCB在内存中?

是

否

从磁盘中读入  
FCB

访问内存中的FCB  
得到首址

显式链接

↓且FAT不在内存中

读入FAT

顺序分配  
隐式分配

索引分配且索引表在

↓内存中

读入索引表

根据首址把物理块读入  
并按需操作

↓  
操作结束后,将修改过的  
物理块(包括FAT, FCB, 索引表)  
进行写回



## 文件系统设计题

给你具有一些特点的文件，要求你设计一个较好的文件系统，是难度较高的简答题，题目可以要求从文件目录，逻辑文件类型，文件分配方式等角度设计。在设计时应该从文件的**大小**，文件的**访问要求**，**访问速度的要求**，**I/O磁盘块次数**等方面入手

从文件组织方式入手：

流式/结构

从逻辑文件入手

文件记录是否等长？是的话优先考虑直接文件（记录已经等长了，你用索引文件反而会增加文件所占空间），不是的话再往下看

题目是否对空间有要求？没有要求则考虑顺序文件（占空间大，但访问效率最高），有的话考虑索引文件（折中的解决方案，没有其它两种逻辑文件那么极端）

从文件访问方式入手：

如图所示：

文件要求顺序访问：所有物理分配都支持  
文件要求直接访问：除了顺序分配，隐式链接分配都可以

有存在索引表的分配方式就能进行直接访问（读取索引表即可，然后直接访问目标物理块）

辨析出文件分配后，接着就从里面选择一个最好的分配方式，比如大中小文件搭配上不同的访问方式，有不同的答案。（个人觉得，小文件可以用连续分配，大中文件可以用索引/链接分配，或者inode分配）

再从文件的目录入手：

如何减少文件目录的访问时间，前面已经提到过了

## 根据字节和不同分配方式来计算I/O次数

如果文件给了字节大小，把物理地址转化成逻辑地址和偏移量进行访存，如果没给具体数字，那就分析极端情况