

# 文件管理

文件是用户 I/O 的基本单位

## FCB 文件控制块

文件元数据 (属性)

- 名称
- 类型
- 创建者 ID
- 所有者 ID
- 位置
- 大小
- 保护

创建时间、最后一次修改时间、最后一次存取时间

FCB 实现了按名存取

|        |
|--------|
| 文件时间   |
| 文件权限   |
| 文件所有者组 |
| 文件大小   |
| 文件数据块  |

FCB

- 基本信息
  - 文件名
  - 物理位置
  - 逻辑结构
  - 物理结构
- 存取控制信息 { 不同用户存取权限
- 使用信息 { 建立时间  
上次修改时间

文件目录 (也是一个文件, 目录文件)

FCB 的有序集合

| 文件名 | 索引结点的编号 |
|-----|---------|
| ... | ...     |
| ... | ...     |

# 索引结点

以UNIX为例

将文件名和文件描述信息分开

文件描述信息 形成索引结点 (inode)

## 磁盘 inode

位于磁盘中的 inode, 每个文件唯一

内容 {

- 文件主 标识等
- 文件类型
- 文件存取权限
- 物理地址
- 文件长度
- 链接计数
- 存取时间

## 内存 inode

位于内存的 inode

当文件打开时, 将磁盘 inode 复制到内存 inode 中

新增内容 {

- inode 编号
- 状态
- 访问计数
- 逻辑设备号
- 链接指针

## 文件操作

OS提供系统调用

创建文件 { 分配必要的外存空间  
在目录中创建个目录项

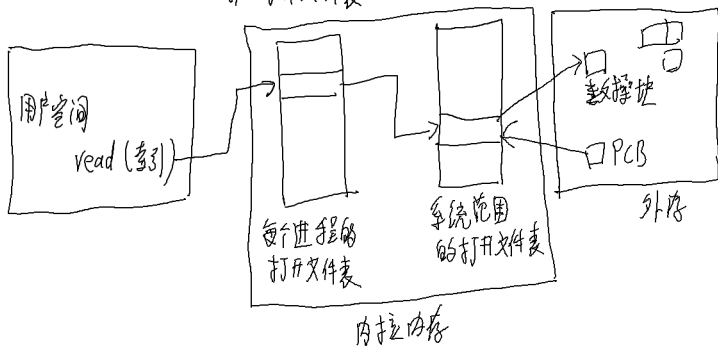
写、读文件 执行系统调用, 写、读指针 (使用同一指针)

重新定位文件 (文件定位) 不涉及写、读, 将当前文件位置指针重新定位

删除文件 { 检索目录项, 释放所占存储空间  
删除目录项

截断文件 文件属性不变, 删除内容

文件打开、关闭 系统调用 open、close  
OS维护 打开文件表



文件名不一定是打开文件表一部分

访问打开文件表的索引 { UNIX 文件描述符  
Windows 文件句柄

打开文件的关联信息 { 文件指针  
文件打开次数  
文件在磁盘位置  
访问权限

文件保护 { 访问控制  
口令  
加密 } > 防止文件被窃取

访问控制表 ACL

用户类型 { 所有者  
组  
其他

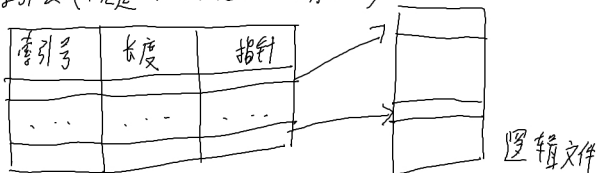
文件的逻辑结构

文件 { 有结构 (记录式文件)  
按逻辑结构 { 无结构 (流式文件) 以 Byte 为单位

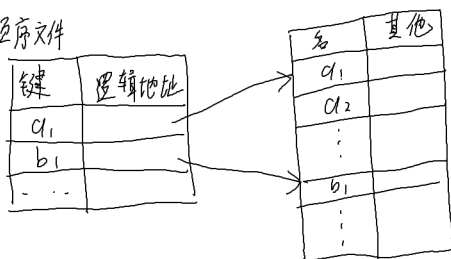
有结构文件 { 顺序文件 一次性读、写一大批记录时效率高  
索引文件  
索引顺序文件  
Hash 文件

## 索引文件

索引表 (本质是一个定长记录的顺序文件)



## 索引顺序文件



文件物理结构 (文件分配) { 连续  
链接  
索引

## 连续分配

每个文件占磁盘上一组连续的块

优点 { 简单  
存取快

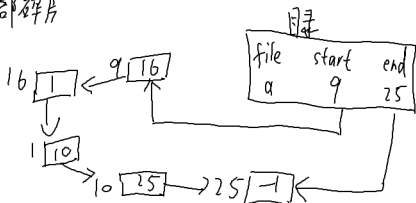
缺点 { 文件长度不易动态增加  
插入、删除时要移动邻块  
会产生外部碎片  
难以确定一个文件所需空间大小

## 链接分配

### 隐式链接

按簇分配，一个簇为几个盘块

有内部碎片



### 显式链接

文件分配表 FAT

| 文件名 | 起始块号 |
|-----|------|
| a   | 2    |
| b   | 7    |

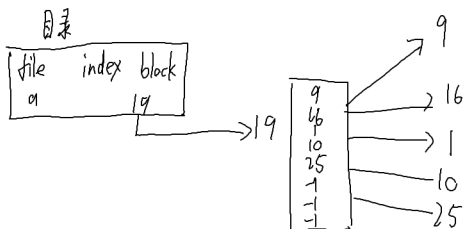
### FAT

| 盘块号 | 下一块 |
|-----|-----|
| 0   | -2  |
| 1   | -1  |
| 2   | 8   |
| 3   | -2  |
| 4   | -2  |
| 5   | -1  |
| 6   | -2  |
| 7   | 1   |
| 8   | 5   |
| ... | ... |

FAT需要占用较大的内存空间

# 索引分配

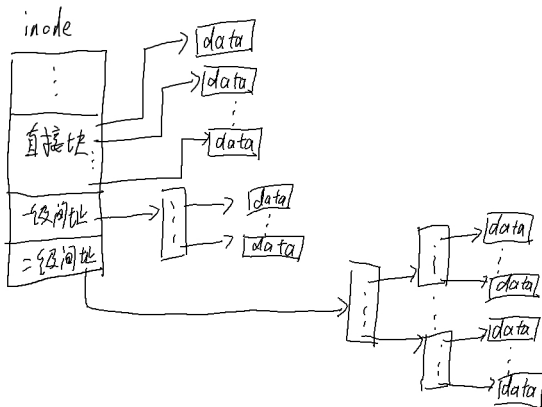
索引块(表)



索引块分配 { 链接索引  
多段索引  
混合索引

## 混合索引分配

UNIX为例

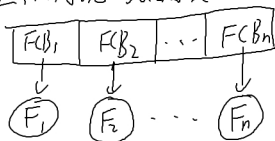


# 目录

## 目录结构

### 单级目录

整个文件系统一张目录表



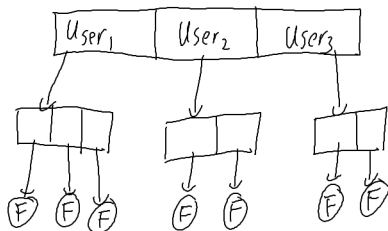
### 两级目录

主文件目录

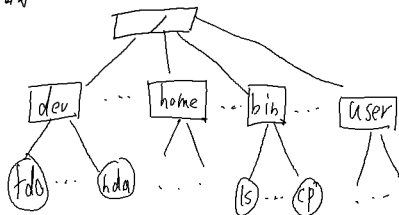
MFD

用户文件目录

UFD



### 树形目录



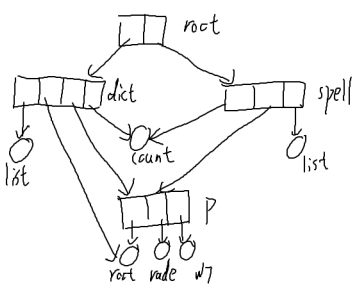
绝对路径

相对路径



# 无环图目录

有向无环图



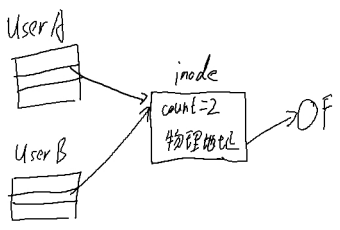
实现了文件的共享  
但使管理变得复杂

- 目录操作
- 搜索
  - 创建文件      增加目录项
  - 删除文件      删除目录项
  - 创建目录
  - 删除目录      { 不删除非空目录
  - 移动目录      { 删除非空目录

## 文件共享

硬链接 (基于 inode)

链接计数 count

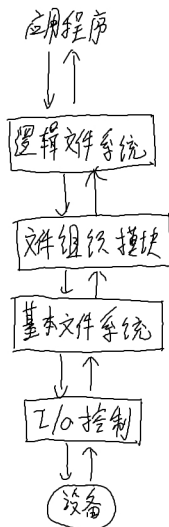


软链接 (符号链接)

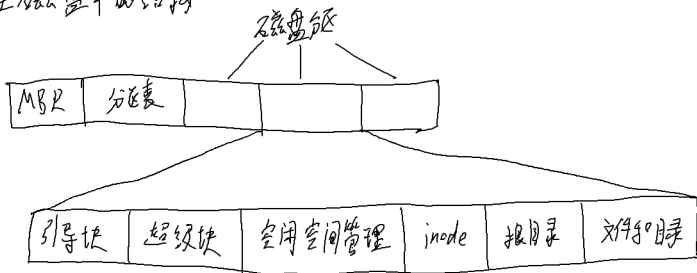
快捷方式

文件系统

层次结构



文件系统在磁盘中的结构



## MBR 主引导记录

位于磁盘 0 扇区

## 分区表

给出每个分区的起始、终止地址

## 引导块

引导块中的程序负责启动该分区中的 OS

每个分区都从引导块开始

## 超级块

包含文件系统的元键信息，如

{ 分区的块的数量  
块大小  
空闲块的数量和指针  
空闲 FCB 数量  
FCB 指针

## 文件系统在内存中的结构

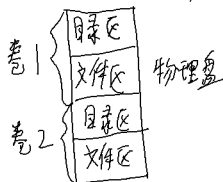
内存中的信息用于管理文件系统 并通过缓存提高性能

信息内容包括

{ 目录表  
目录结点的缓存  
整个系统的打开文件表  
每个进程的打开文件表

## 外存空间管理 (空闲块的组织和管理)

包含文件系统的分区为卷



# 空闲块管理、分配

## 空闲表法

连续分配式，与内存动态分配相似

分配方法 { FF  
BF

## 空闲表

| 序号 | 第一个空闲盘块号 | 空闲盘块数 |
|----|----------|-------|
|----|----------|-------|

## 空闲链表法

{ 空闲盘块链

空闲盘区链

一个盘区含多个盘块

## 位图法

|   | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 |
| 2 | 1 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 | 1 |

0 表示空闲

1 表示已分配

$m \times n$  个盘块

$map[m, n]$

盘块号  $b$  行号  $i$  列号  $j$

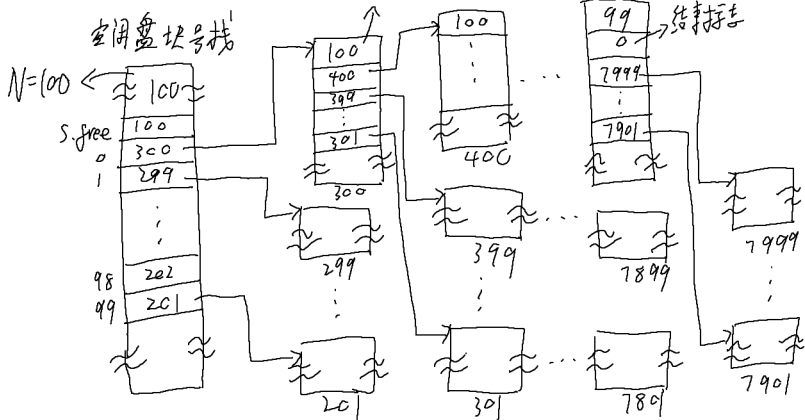
行、列从 1 开始

$$b = n(i-1) + j$$

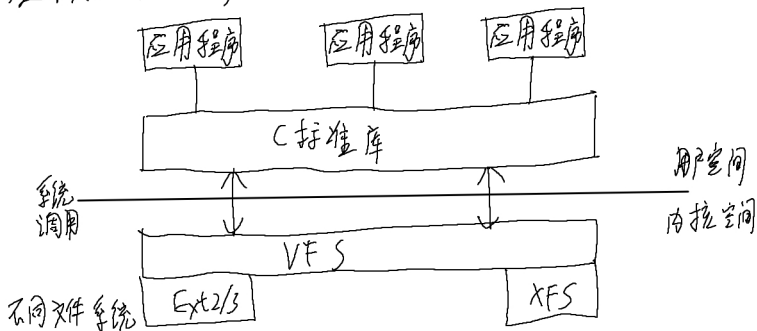
$$i = (b-1)/n + 1$$

$$j = (b-1) \% n + 1$$

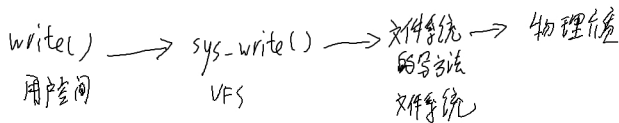
## 成组链接法



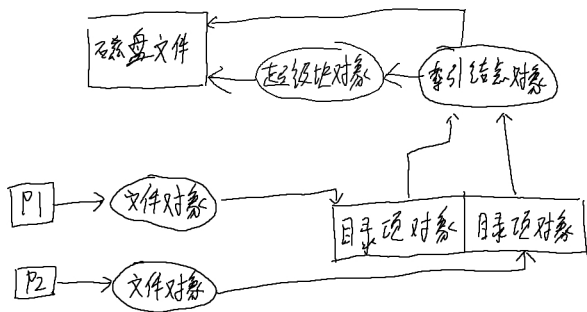
## 虚拟文件系统 (VFS)



例 Linux 调用 write()



## 进程与 VFS 对象交互



文件系统在使用前必须先挂载 mount