

# Lecture 9: 知识 - 目录

## 考试大纲

- 1. 目录的基本概念
- 2. 树形目录
- 3. 目录的操作
- 4. 硬链接和软链接

## 目录的基本概念

### 概念：目录和文件控制块

- 引入：
  - 目录本身就是一种有结构文件，由一条条记录组成。每条记录对应一个在该放在该目录下的文件；
  - **FCB 的有序集合称为“文件目录”**；
  - 目录文件中的一条记录就是一个**文件控制块 (FCB)**；
- 组成：
  - 基本信息 (文件名、物理地址、逻辑结构、物理结构等)；
  - 存取控制信息 (是否可读/可写、禁止访问的用户名单等)；
  - 使用信息 (如文件的建立时间、修改时间等)；
- 文件目录的目的：
  - 1. 实现按名存取：因为 **FCB** 实现了文件名和文件之间的映射；
  - 2. 提高检索效率：目录的存取效率直接影响系统的效率；
  - 3. 多用户共享文件：目录要能够支持多个用户共享文件的使用，因此要提高控制文件访问的信息；
  - 4. 重命名：应允许用户进行对自己的文件的命名；

## 目录结构

### 单级目录结构

#### 概念：单级目录结构

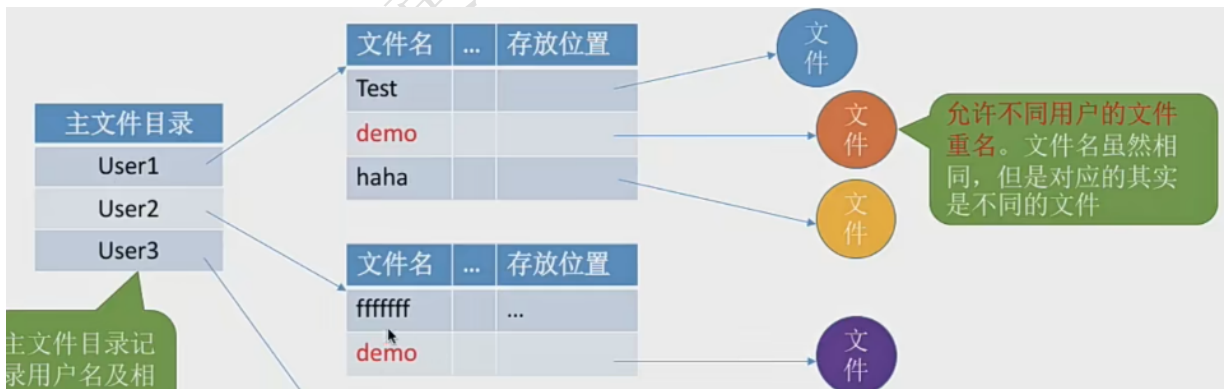
- 概念：
  - 早期操作系统并不支持多级目录，**整个系统中只建立一张目录表，每个文件占一个目录项**；
- 创建文件：
  - 在创建一个文件时，需要先检查目录表中有没有重名文件，确定不重名后才能允许建立文件，并将新文件对应的目录项插入目录表中；
- 查找文件：
  - 先按照文件名在目录中找到对应的 PCB，然后再经过合法性检查后，执行指定操作；

- 特点：
  - 1. 支持按名存取；
  - 2. 不支持文件重命名；
  - 3. 不适用于多用户系统；
  - 4. 查找速度慢；

## 两级目录结构

**概念：**单级目录结构

- 概念：
  - 早期的多用户操作系统，采用两级目录结构。
  - 分为主文件目录（MFD, Master File Directory）和用户文件目录（UFD, User File Directory）。
- 结构：
  - 主文件目录项：
    - 记录用户名以及相应用户文件目录所在的存储位置；
  - 用户文件目录项：
    - 记录此用户所有的 PCB；
- 特点：
  - 1. 两级目录结构 **允许不同用户的文件重名**；
  - 2. 可以在目录上实现访问限制（检查此时登录的用户名是否匹配）；
  - 3. 用户不可以把自己的文件进行分类；
- 图示：

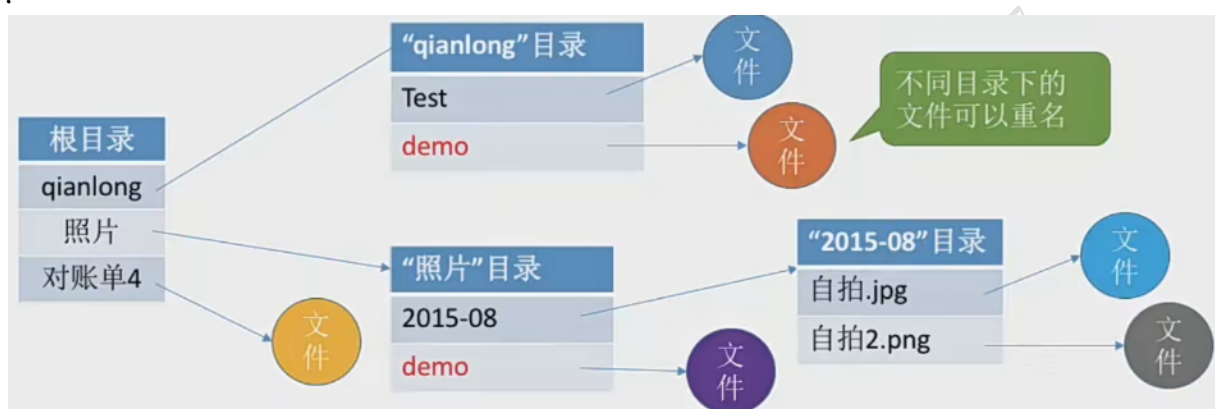


## 树形目录结构

**概念：**多级目录结构

- 概念：
  - 又称树形目录结构；
  - 当用户要访问某个文件时，用文件的路径名标识文件，**文件路径名**是个字符串，由从根目录到所找文件通路上所有目录名和数据名称用分隔符“/”构成；
- 特点
  - 优点：

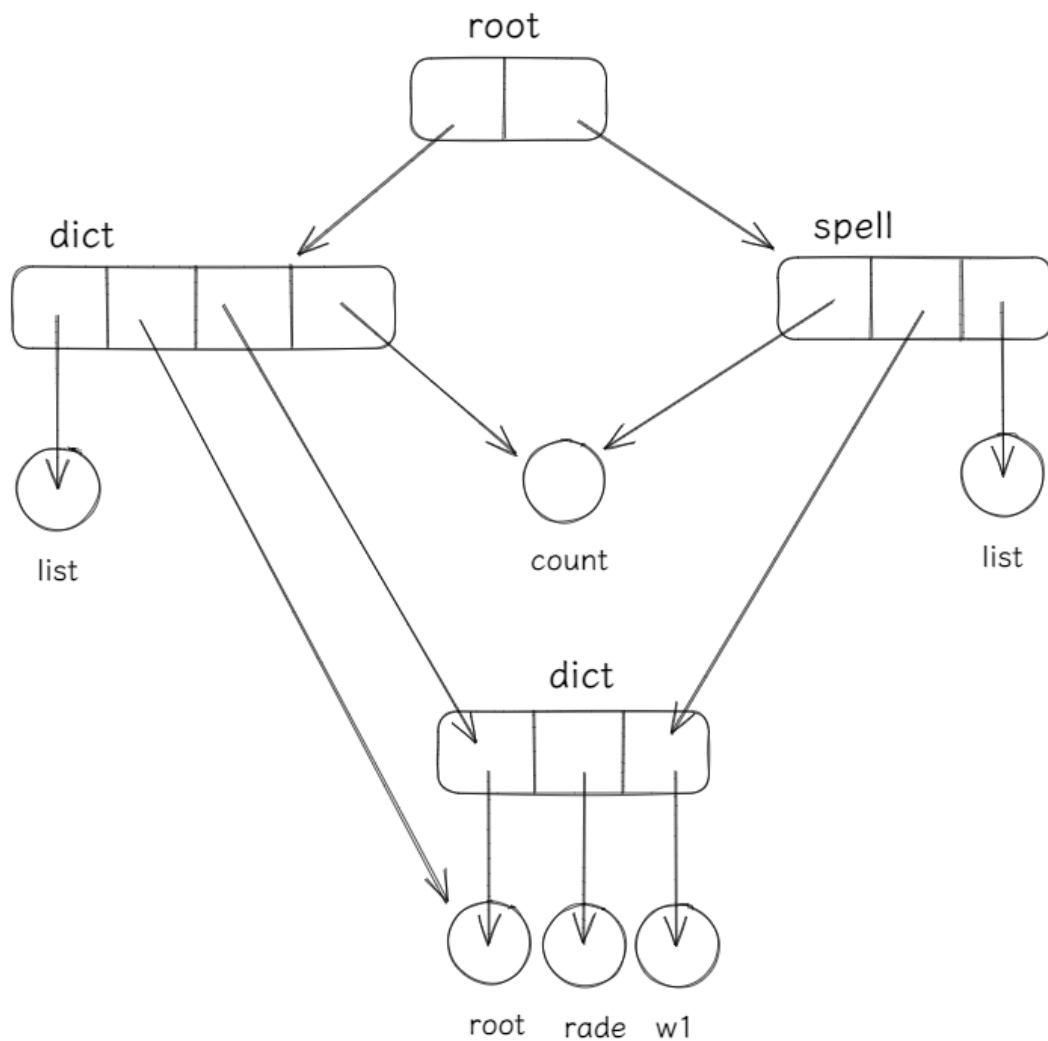
- 可以很方便地对文件进行分类；
- 层次结构清晰；
- 能更有效地进行文件的管理和保护；
- 缺点：
  - 不便于实现文件的共享；
  - 在树形目录中查找一个文件，需要按路径名逐级访问中间节点，增加了磁盘访问次数，会影响查询速度；
- 路径：
  - 绝对路径：从根目录开始；
  - 相对路径：从当前目录出发的相对路径；
- 图示：



## 无环图目录结构

**概念：**无环图目录结构

- 概念：
  - 在多级目录结构基础上，增加一些指向同一节点的有向边，使整个目录成为一个有向无环图；
  - 可以更方便地实现多个用户间的**文件共享**；
  - 可以用不同的文件名指向同一个文件，甚至可以指向同一个目录（共享同一目录下的所有内容）
- 共享计数器：
  - 可以用不同的文件名指向同一个文件，甚至可以指向同一个目录（共享同一目录下的所有内容）；
  - 需要为每个共享结点设置一个共享计数器，用于记录**此时有多少个地方在共享该结点**；
  - 用户提出删除结点的请求时，只是删除该用户的 FCB、并使共享计数器减 1，并不会直接除共享结点；
  - 只有共享计数器减为 0 时，才删除结点；
- 注意：
  - 共享文件不同于复制文件；
  - 在共享文件中，由于各用户指向的是同一个文件，因此只要其中一个用户修改了文件数据，那么所有用户都可以看到文件数据的变化；
- 图示：



## 目录操作

**概念：**目录操作

- 搜索
  - 当用户要使用一个文件时，系统要根据文件名搜索目录，找到该文件对应的目录项
- 创建文件
  - 创建一个新文件时，需要在所属的目录增加一个目录项
- 删除文件
  - 当删除一个文件时，需要在目录中删除相应的目录项
- 显示目录
  - 用户可以请求显示目录的内容，如显示该目录中的所有文件及相应属性
- 修改目录
  - 某些文件属性保存在目录中，因此这些属性变化时需要修改相应的目录项 (如：文件重命名)

## 文件共享

### 硬链接

## 概念：索引结点 - 对 FCB 的改进

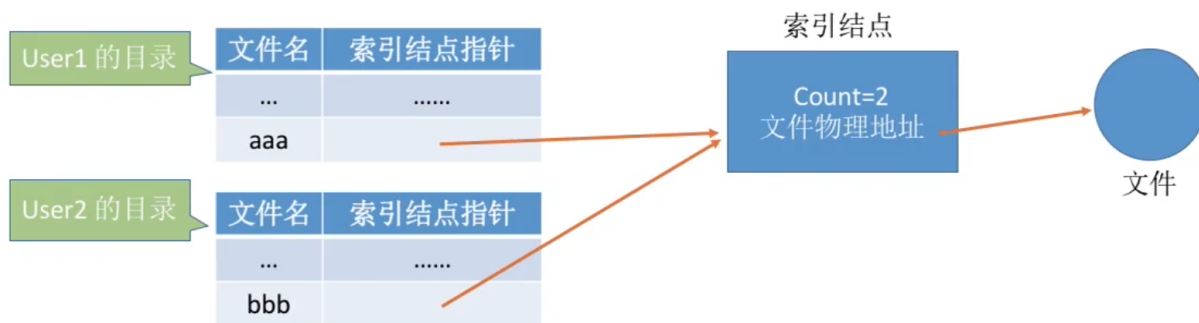
- 引入
  - 只有**文件名**匹配时，才需读出文件的其他信息；
  - 因此可以考虑让目录表**瘦身**来提高效率；
- 概念：
  - 除了**文件名之外的文件描述信息**都放到索引节点当中，每一个文件都有一个唯一的索引结点；
  - 即在文件目录中，只保存两个东西：
    - 1. 文件名；
    - 2. 指向相应索引节点的指针；
- 查找：
  - 当找到文件名对应的目录项时，才需要将索引结点调入内存，索引结点中记录了文件的各种信息，包括文件在外存中的存放位置，根据“存放位置”即可找到文件；
- 功能
  - 存放在外存中的索引结点称为磁盘索引结点，当索引结点放入内存后称为内存索引结点。
  - 相比之下内存索引结点中需要增加一些信息
    - 比如：文件是否被修改，此时有几个进程正在访问该文件等

## 概念：基于索引结点的共享方式 - 硬链接

- 索引节点
  - 索引节点指针中**存储指向索引节点的指针**，索引节点中存储文件的物理地址；



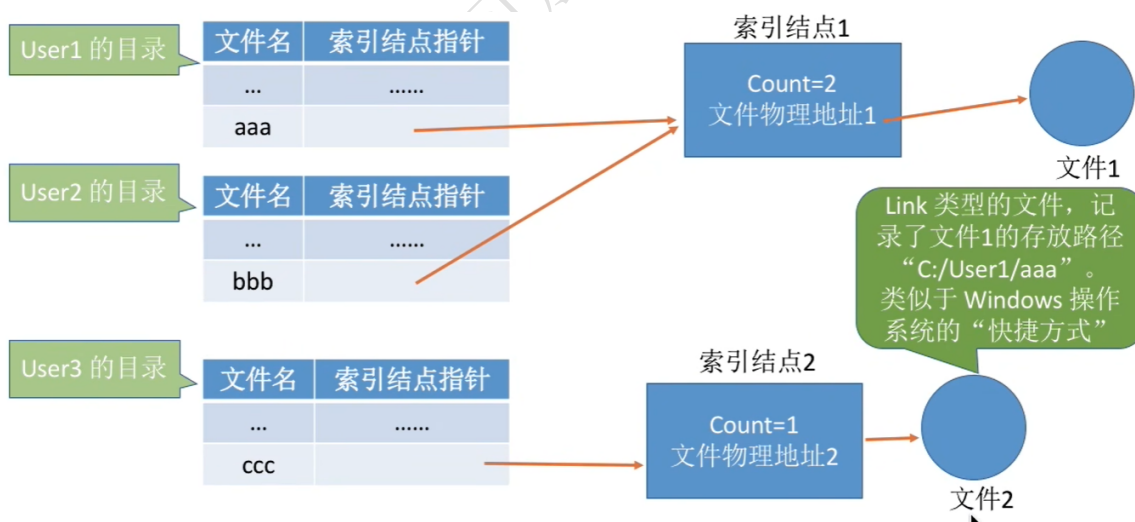
- 链接计数器：
  - 链接计数变量 `count`；
- 删除：
  - 若某个用户决定“删除”该文件，则只是要把用户目录中与该文件对应的目录项删除，且索引结点的 `count` 值减 1；
- 实现：
  - 用户目录项指向同一个索引结点
  - 索引结点中需要有链接计数 `count`
  - 某用户想删除文件时，只是删除该用户的目录项，且 `count--`
  - 只有 `count == 0` 时才能真正删除文件数据和索引结点，否则**会导致指针悬空**；
- 图示：



## 软链接

**概念:** 基于符号链接的共享方式 - 软链接

- 实现:
  - 在一个 **Link** 型的文件中记录被链接文件的存放路径 (Windows 快捷方式)
  - 操作系统根据路径一层层查找目录, 最终找到共享文件;
  - 只有文件主才用于此文件, 其他 **Link** 连接到此文件的用户不可以删除它;
- 特征:
  - 即使软链接指向的共享文件已经被删除, **Link** 型文件依然存在, 只是通过 **Link** 型文件中的路径去查找共享文件会失败 (找不到对应目录项);
- 注意:
  - 符号链接文件的引用计数值不受被链接文件的影响, 始终为 1;
- 图示:



当 User3 访问 "ccc" 时, 操作系统判断文件 "ccc" 属于 Link 类型文件, 于是会根据其中记录的路径层层查找目录, 最终找到 User1 的目录表中的 "aaa" 表项, 于是就找到了文件1的索引结点。