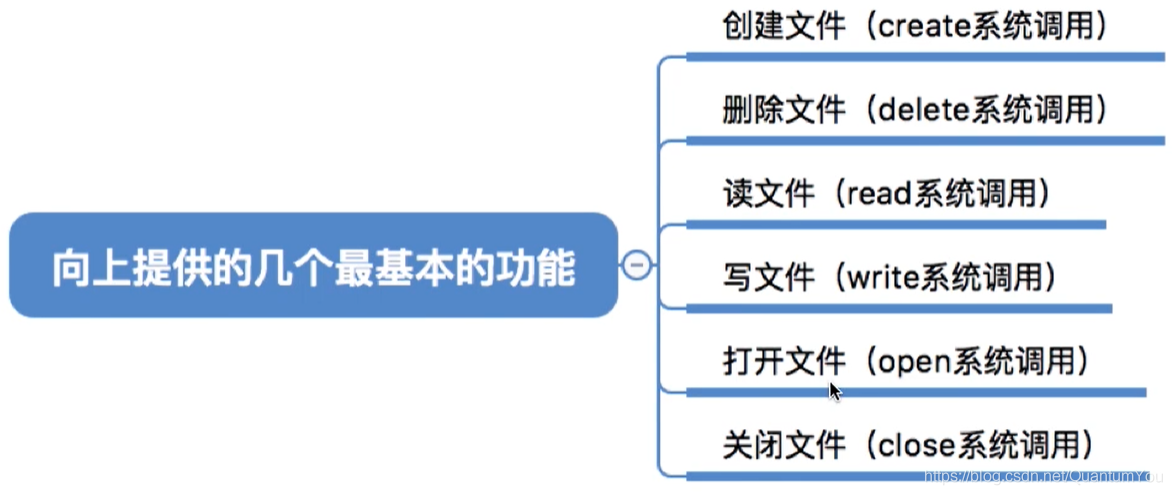
@[toc]

# 文件基本操作

## 知识总览思维导图



## 创建文件

* 进行 Create系统调用时，需要提供的几个主要参数：
* 1.所需的外存空间大小（如：一个盘块，即1KB）
* 2.文件存放路径（“D：/Demo”）
* 3.文件名（这个地方默认为“新建文本文档txt”）
* 操作系统在处理Create系统调用时，主要做了两件事
* 1、在外存中找到文件所需的空间（结合上小节学习的空闲链表法、位示图、成组链接法等管理策略，找到空闲空间）
* 2、根据文件存放路径的信息找到该目录对应的目录文件（此处就是D：/Demo目录），在目录中创建该文件对应的目录项。目录项中包含了文件名、文件在外存中的存放位置等信息。

## 删除文件

* 进行 Delete系统调用时，需要提供的几个主要参数：

1、文件存放路径（“D：/Demo”）  
2、文件名（“ test. txt”）

操作系统在处理 Delete系统调用时，主要做了几件事

* 1.根据文件存放路径找到相应的目录文件，从目录中找到文件名对应的目录项。
* 2.根据该目录项记录的文件在外存的存放位置、文件大小等信息，回收文件占用的磁盘块。（回收磁盘块时，根据空闲表法、空闲链表法、位图法等管理策略的不同，需要做不同的处理）
* 3、从目录表中删除文件对应的目录项。

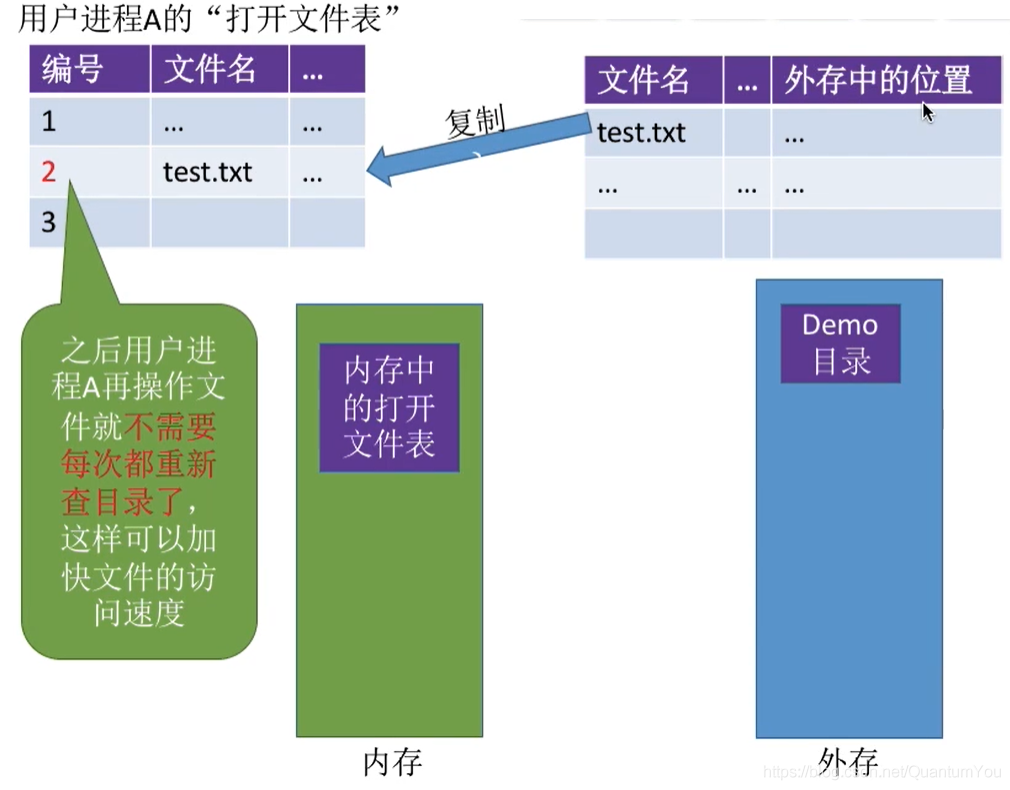
## 打开文件

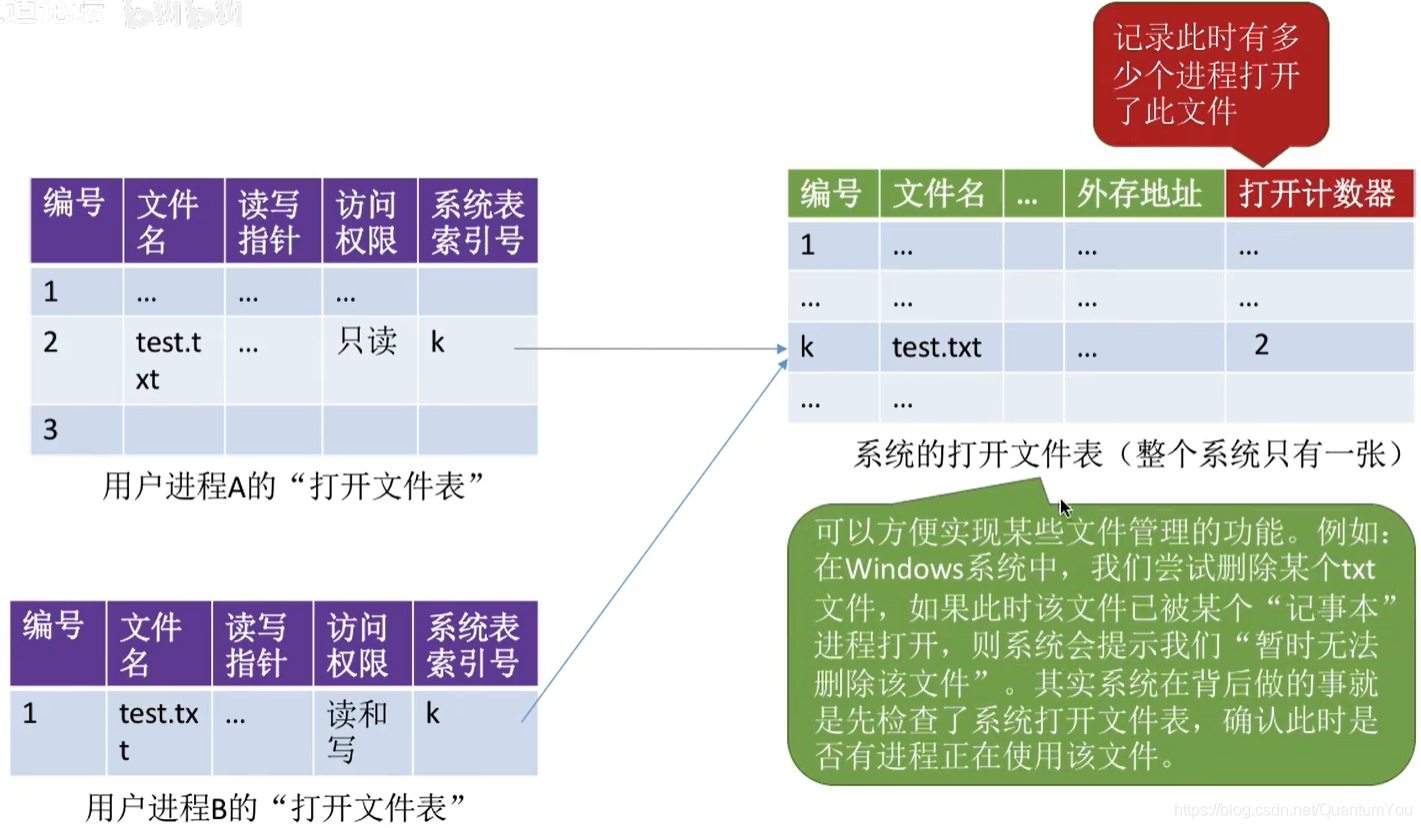
* 在很多操作系统中，在对文件进行操作之前，要求用户先使用open系统调用“打开文件”，需要提供的几个主要参数

1、文件存放路径（“D；/Demo”）  
2、文件名（“ test. txt”）  
3、要对文件的操作类型（如：r只读，rw读写等）

操作系统在处理open系统调用时，主要做了几件事

* 1、根据文件存放路径找到相应的目录文件，从目录中找到文件名对应的目录项，并检查该用户是否有指定的操作权限。
* 2、将目录项复制到内存中的“打开文件表”中。并将对应表目的编号返回给用户。之后用户使用打开文件表的编号来指明要操作的文件。

  
注意：在系统中打开文件表的类型



* 读写指针： 记录该进程对文件的读 / 写操作进行的位置
* 如果打开文件时声明的是“只读”，则该进程不能对文件进行写操作

## 关闭文件

* 进程使用完文件后，要“关闭文件”  
  操作系统在处理 Close系统调用时，主要做了几件事：
* 1 、将进程的打开文件表相应表项删除
* 2、回收分配给该文件的内存空间等资源
* 3、系统打开文件表的打开计数器 count减1，若 count = 0，则删除对应表项。

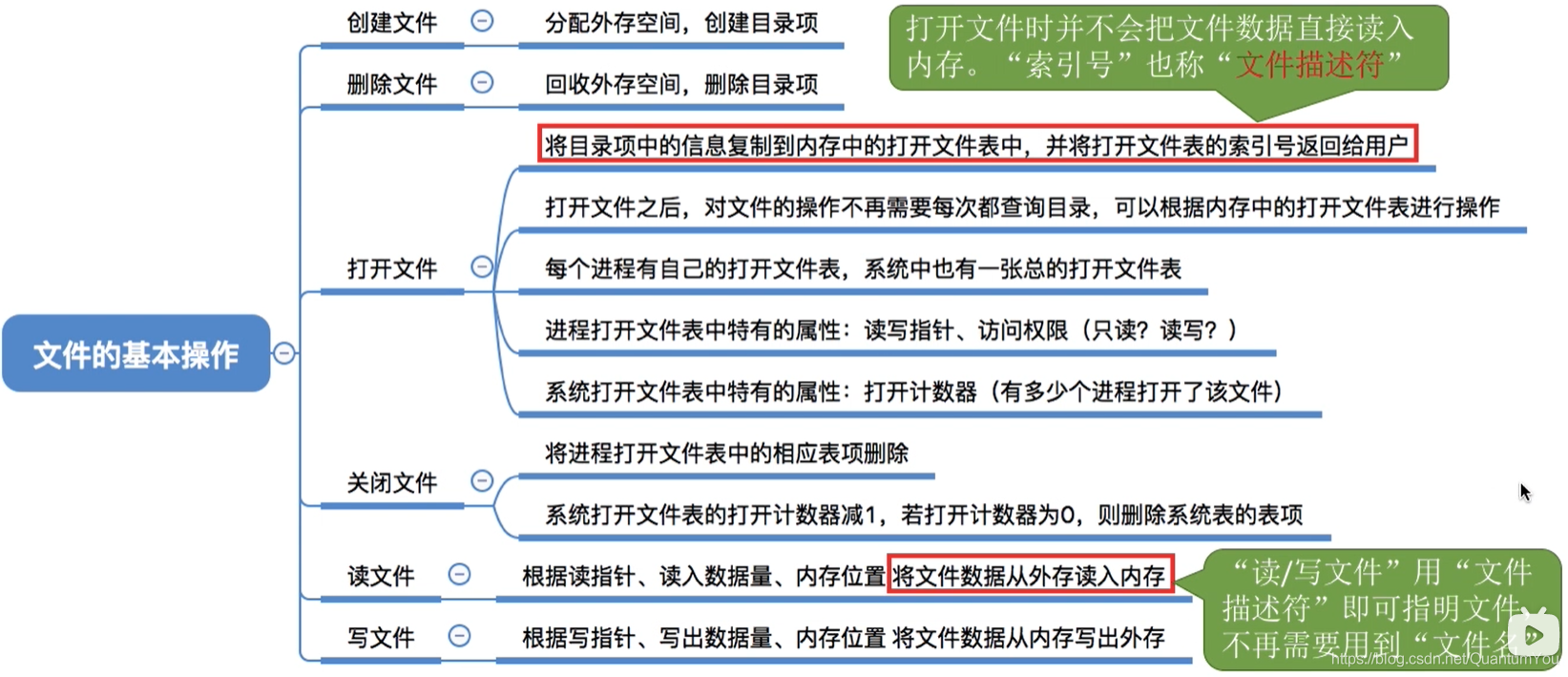
## 读文件

* 进程使用read系统调用完成写操作。需要指明是哪个文件（在支持“打开文件”操作的系统中，只需要提供文件在打开文件表中的索引号即可），还需要指明要读入多少数据（如：读入1KB）、指明读入的数据要放在内存中的什么位置。
* 操作系统在处理read系统调用时，会从读指针指向的外存中，将用户指定大小的数据读入用户指定的内存区域中。

## 写文件

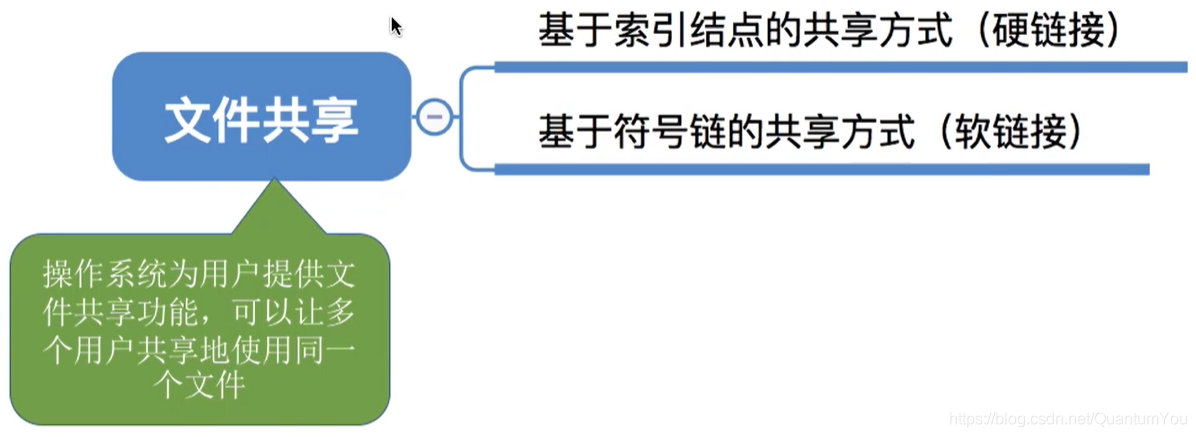
* 进程使用 write系统调用完成写操作，需要指明是哪个文件（在支持“打开文件”操作的系统中，只需要提供文件在打开文件表中的索引号即可），还需要指明要写出多少数据（如：写出1KB）、写回外存的数据放在内存中的什么位置
* 操作系统在处理 write系统调用时，会从用户指定的内存区域中，将指定大小的数据写回写指针指向的外存。

## 总结思维导图



# 文件共享

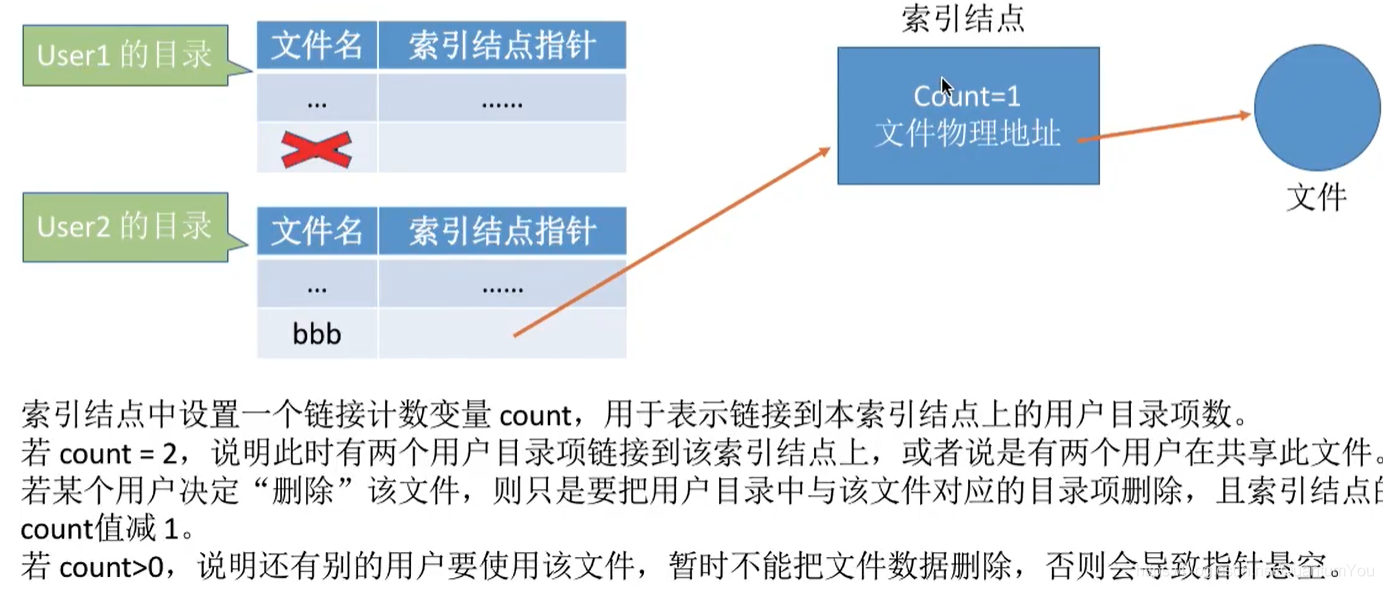
## 知识总览思维导图

  
注意： 复制文件与共享文件的区别

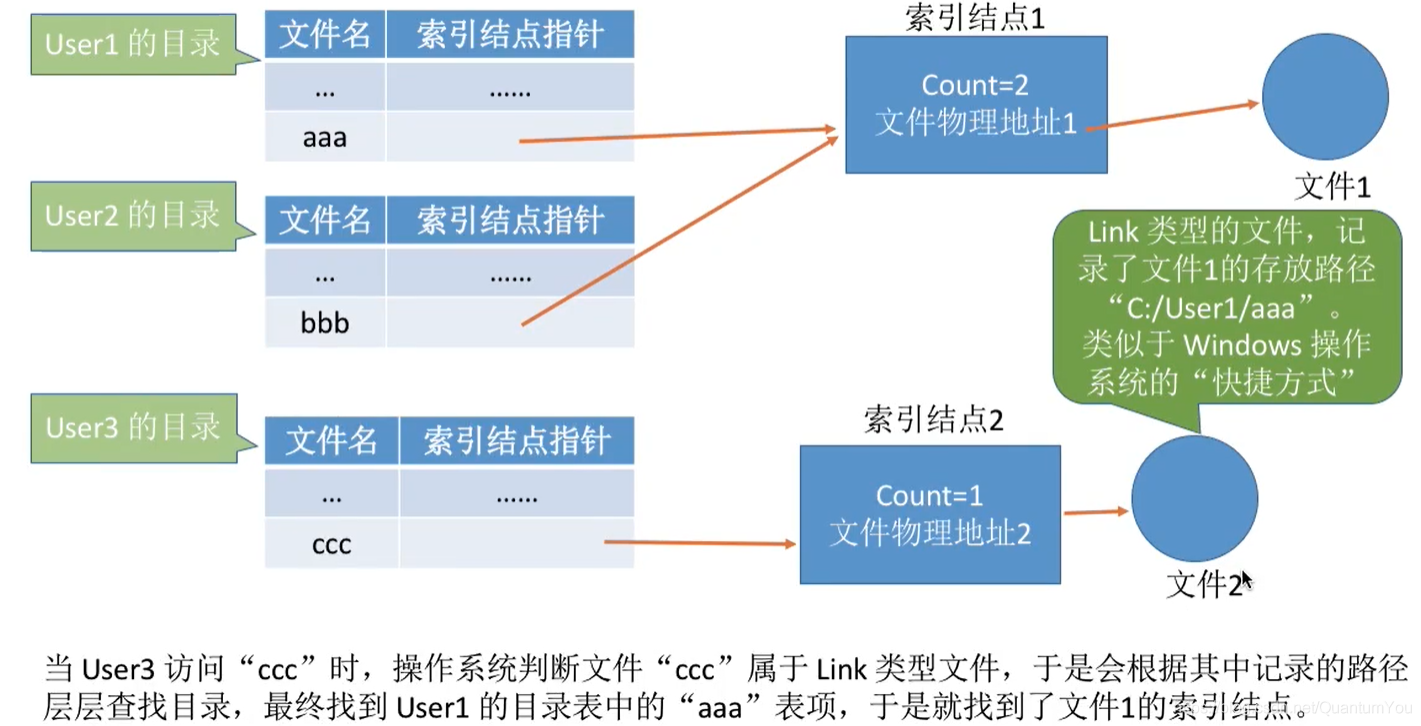
* 多个用户共享同一个文件，意咪着系统中只有“一份”文件数据。并且只要某个用户修改了该文件的数据，其他用户也可以看到文件数据的变化。
* 如果是多个用户都“复制”了同一个文件，那么系统中会有“好几份”文件数据。其中一个用户修改了自己的那份文件数据，对其他用户的文件数据并没有影响。

## 基于索引结点的共享方式（硬链接）

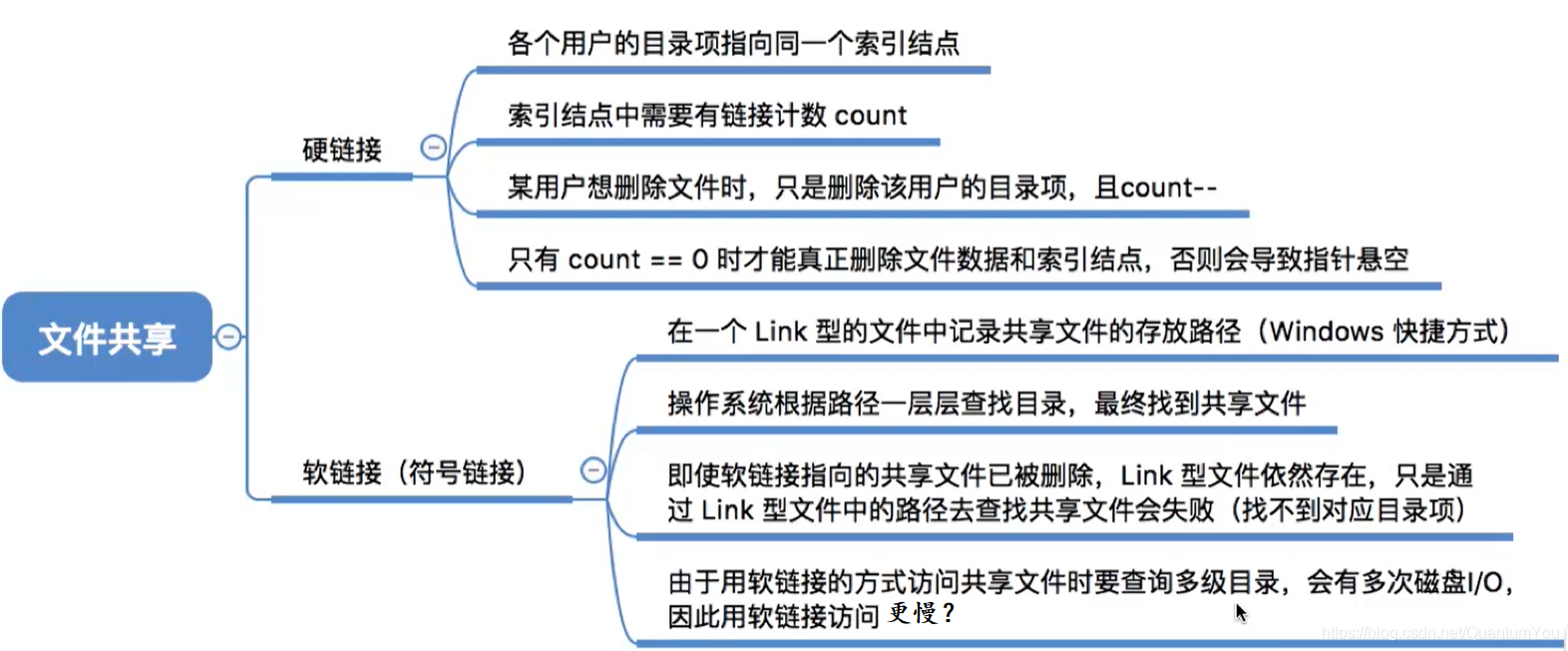
* **索引结点**，是一种文件目录瘦身策略。由于检索文件时只需用到文件名，因此可以将除了文件名之外的其他信息放到索引结点中。这样目录项就只需要包含文件名、索引结点指针。

  
当 count = 0 时，系统负责删除文件

## 基于符号链的共享方式（软链接）

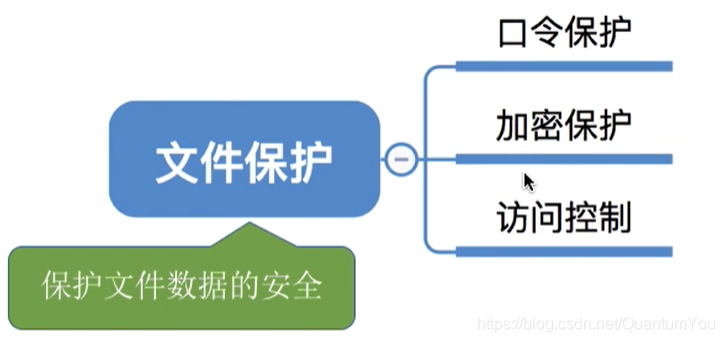


## 总结思维导图



# 文件保护

## 知识总览思维导图



## 口令保护

* 为文件设置一个“口令”（如：abc112233），用户请求访问该文件时必须提供“口令”
* 口令一般存放在文件对应的FCB或索引结点中。用户访问文件前需要先输入“口令”，操作系统会将用户提供的口令与FCB中存储的口令进行对比，如果正确，则允许该用户访问文件
* **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。

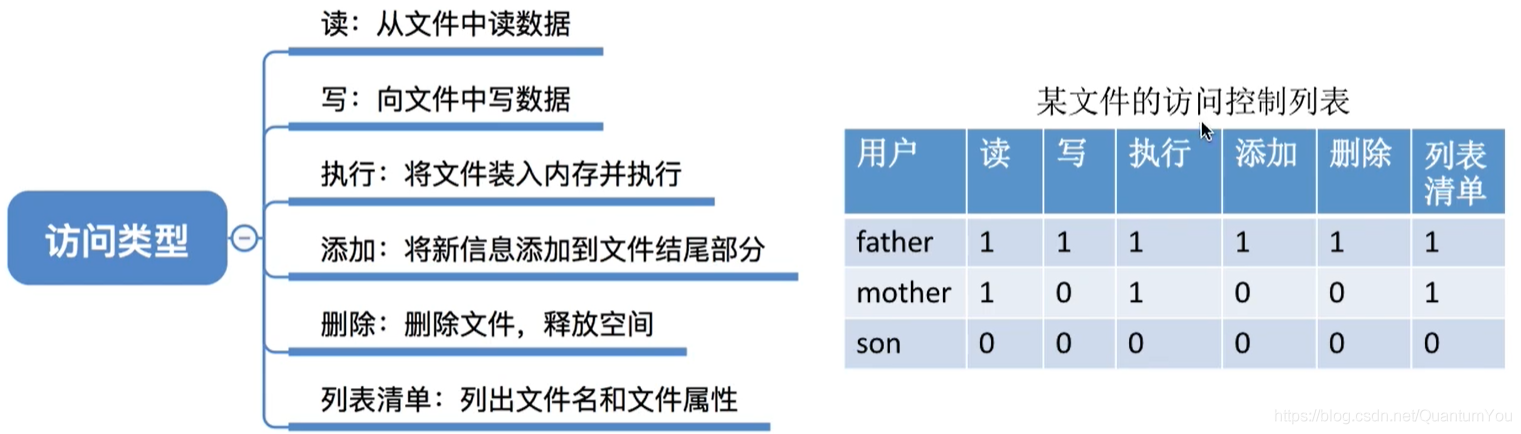
## 加密保护

* 使用某个“密码”对文件进行加密，在访问文件时需要提供正确的“密码”才能对文件进行正确的解密。
* **优点**：保密性强，不需要在系统中存储“密码”
* **缺点**：编码/译码，或者说加密/解密要花费一定时间。

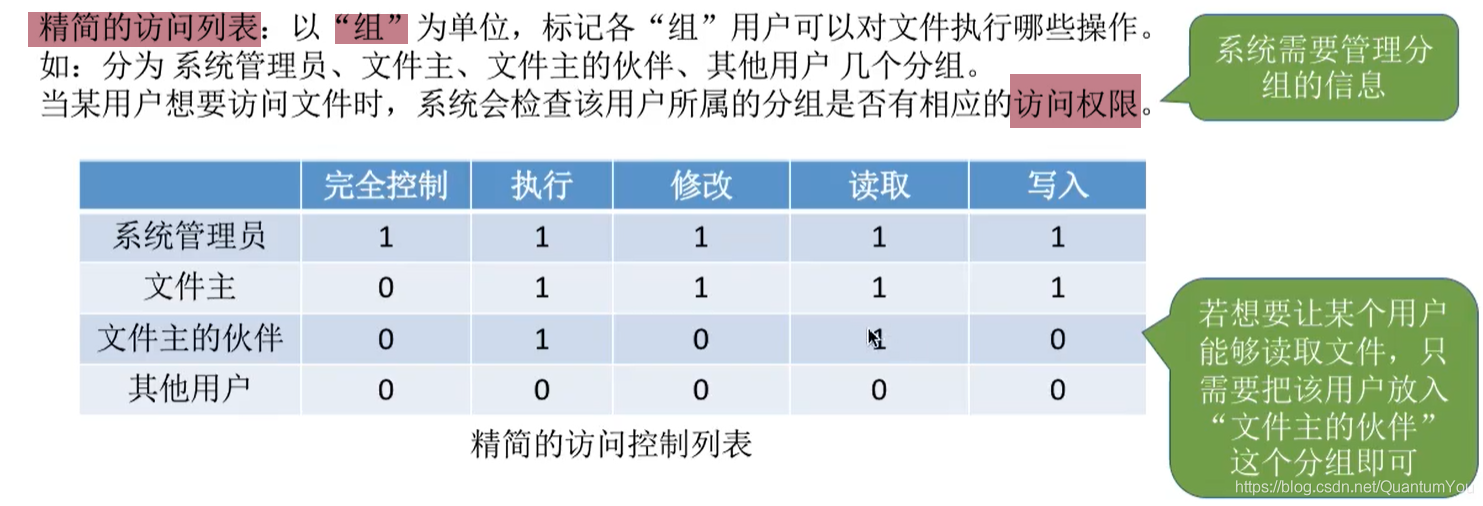


## 访问控制

* 在每个文件的FCB（或索引结点）中增加一个访问控制列表（ Access-Control list，ACL），该表中记录了各个用户可以对该文件执行哪些操作。



* **问题引入**：有的计算机可能会有很多个用户因此访问控制列表可能会很大，可以用精简的访问列表解决这个问题

解法 ：  


## 总结思维导图

