

# 第七章 文件管理

# 文件管理

- 文件和文件系统
- 文件的逻辑结构
- 文件目录
- 文件共享
- 文件保护

## 7.1 文件和文件系统

# 7.1.1 文件、记录和数据项

- 1. 数据项
  - 最低级的数据组织形式
  - (1) 基本数据项
    - 这是用于描述一个对象的某种属性的字符集
    - 是数据组织中可以命名的最小逻辑数据单位，即原子数据，又称为数据元素或字段
  - (2) 组合数据项
    - 由若干个基本数据项组成的，简称组项。

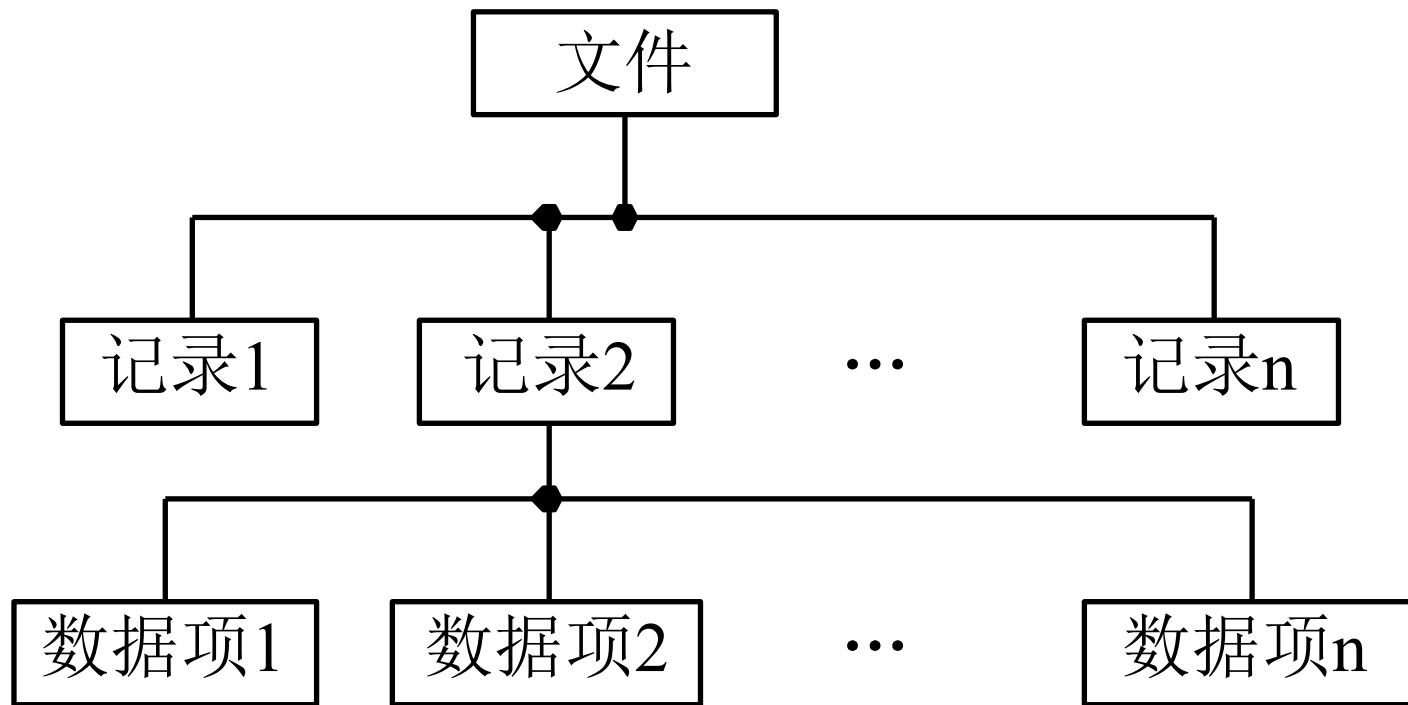
- 2. 记录

- 一组相关数据项的集合
- 用于描述一个对象在某方面的属性

- 3. 文件

- 文件是指由创建者所定义的、具有文件名的一组相关元素的集合
- 分为有结构文件和无结构文件两种。在有结构的文件中，文件由若干个相关记录组成；而无结构文件则被看成是一个字符流
- 文件在文件系统中是一个最大的数据单位，它描述了一个对象集。
- 属性：类型、长度、物理位置、建立时间等。

- 文件、记录和数据项之间的层次关系



## 7.1.2 文件名和类型

- 文件名 扩展名
- 文件类型
  - 按用途分类
    - 系统文件、用户文件、库文件。
  - 按文件中数据的形式分类
    - 源文件、目标文件、可执行文件
  - 按存取控制属性分类
    - 只执行文件、只读文件、读写文件
  - 按组织形式和处理方式分类
    - 普通文件、目录文件、特殊文件(I/O设备)

## 7.1.3 文件系统模型

### —三个层次

文件系统接口  
(程序, 命令)

操纵与管理软件  
(核心部分)

对象及其属性  
(文件, 目录, 磁盘)



## 7.1.4 文件操作

- 1 基本文件操作
  - 创建文件
    - 分配外存空间、在目录中建立目录项
  - 删除文件
    - 删除目录项，收回存储空间
  - 读文件、写文件
    - 查找目录项，获得外存位置
  - 设置文件的读/写位置

## 7.1.4文件操作

- 2. 文件的“打开”和“关闭”操作
  - 文件操作需检索文件目录以获得文件属性和外存位置，为避免多次重复检索目录，引入打开操作。
  - 所谓“打开”，是指系统将指名文件的属性(包括该文件在外存上的物理位置)从外存拷贝到内存打开文件表的一个表目中，并将该表目的编号(或称为索引)返回给用户。以后，可利用系统所返回的索引号向系统提出操作请求。系统直接利用该索引号到打开文件表中去查找。

# “关闭”操作

- 文件在使用后必须关闭，相应的系统调用：`close (fd)`
- 将进程打开文件表中的相应表项删除；
- 将系统打开文件表中相应表项的共享计数值减1，如果该值仍大于0，说明还有其它进程在使用该文件，此次操作结束，返回；否则转下步；
- 把更新后的文件信息（文件控制块）复制回外存中的目录结构，一些缓存中的内容必要时写入外存。然后把系统打开文件表中的相应表项删除。

- 3 其它文件操作
  - 对文件属性进行操作
  - 目录操作
  - .....

## 7.2 文件的逻辑结构

- 文件的逻辑结构(File Logical Structure)
  - 用户观点所看到的文件组织形式，与存储设备特性无关
- 文件的物理结构
  - 又称为文件的存储结构，是指文件在外存上的存储组织形式。
- 选择逻辑结构的原则
  - 检索速度快
  - 便于修改
  - 占用存储空间小，不要求占用大片的连续存储空间

## 7.2.1 文件逻辑结构的类型

- 1 有结构文件（记录式文件）
  - 定长记录、变长记录
  - 组织形式
    - 顺序文件、索引文件、索引顺序文件。
- 2. 无结构文件（流式文件）
  - 长度以字节为单位，采用读写指针来指出下一个要访问的字符

## 7.2.2 顺序文件

- 1. 逻辑记录的排序

- 串结构

- 各记录之间的顺序与关键字无关。通常的办法是由时间来决定，即按存入时间的先后排列，最先存入的记录作为第一个记录，其次存入的为第二个记录，……依此类推。

- 顺序结构

- 指文件中的所有记录按关键字(词)排列。可以按关键词的长短从小到大排序，也可以从大到小排序；或按其英文字母顺序排序
    - 具有更高的查找效率



## 7.2.2 顺序文件

- 2. 顺序文件的优缺点
  - 批量存取时效率高
  - 用户(程序)要求查找或修改单个记录，系统便要去逐个地查找诸记录
  - 增加或删除一个记录 困难

## 7.2.3 记录寻址

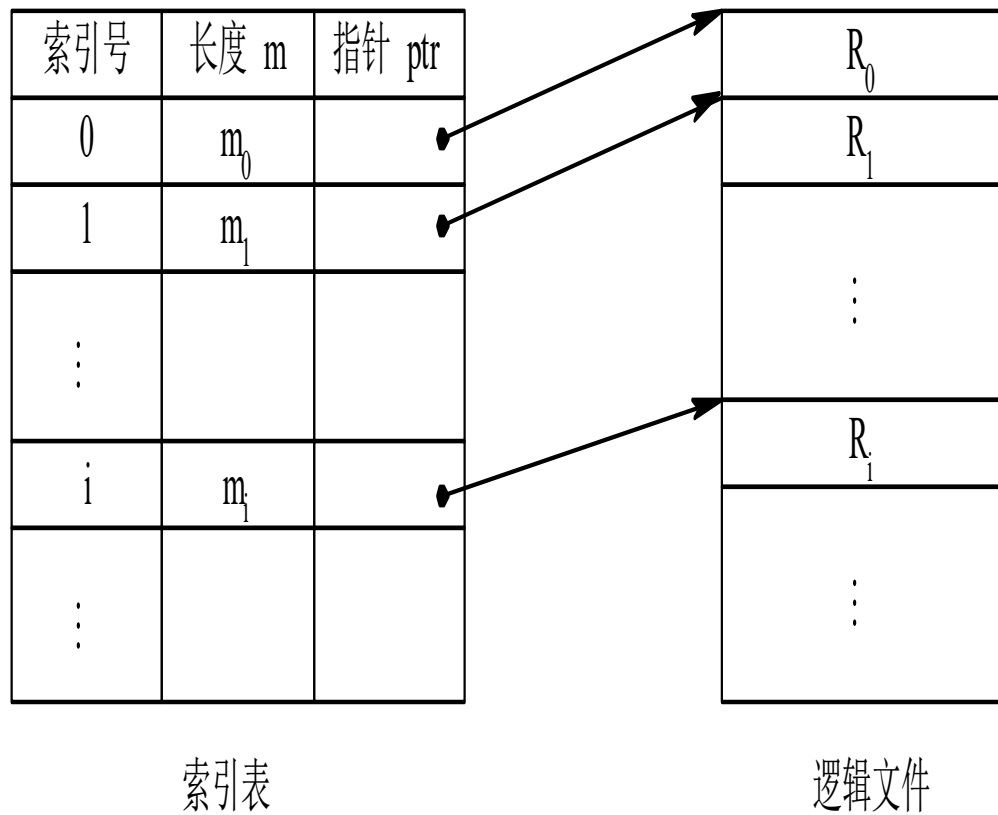
- 查找第*i*个记录的首址
  - 定长记录文件
    - $A_i = i \times L$
  - 可变长度记录的文件
    - 顺序地查找每个记录，从中获得相应记录的长度 $L_i$ ，然后才能按下式计算出第*i*个记录的首址

$$A_i = \sum_{j=0}^{i-1} L_j + i$$

## 7.2.4 索引文件

- 索引表

- 每个记录在表中有一表项，记录该记录的长度和指向该记录的指针（该记录在逻辑地址空间的首地址）
- 索引表按记录键排序，本身为定长记录顺序文件
- 检索索引文件时，根据用户提供的关键字查找索引表，根据表项给出指向记录的指针访问记录。



# 索引顺序文件

- 常见的逻辑文件形式；
- 结合使用顺序文件和索引文件；
- 将顺序文件的记录分组；
- 顺序文件建立索引表，指向每组的第一个记录

键	逻辑地址
An Qi	
Bao Rong	
Chen Lin	

姓 名	其它属性
An Qi	
An Kang	
Bao Rong	
⋮	

逻辑文件

## 7.2.4 索引顺序文件

- 查找过程
  - 根据用户提供的关键字检索索引表，得到该记录组第一个记录的位置
  - 顺序查找主文件
  - 顺序文件平均查找次数 $N/2$ ；索引顺序文件为 $\sqrt{N}$ ;
- 多级索引
  - 适用大文件

## 7.2.5 直接文件和哈希文件

- 1. 直接文件
  - 根据给定的记录键值，直接获得指定记录的物理地址。换言之，记录键值本身就决定了记录的物理地址。
  - 这种由记录键值到记录物理地址的转换被称为键值转换(Key to address transformation)。
  - 组织直接文件的关键，在于用什么方法进行从记录值到物理地址的转换。
- 2. 哈希(Hash)文件
  - 键值 ——— 散列函数 ——— 地址

## 7.3 文件目录

# 文件目录

- 实现文件管理的途径
  - (1) 实现“按名存取”；
  - (2) 提高对目录的检索速度；
  - (3) 文件共享；
  - (4) 允许文件重名。



# 7.3.1 文件控制块和索引结点

- 1 文件控制块
  - 描述和控制文件的数据结构
  - 内容
    - 基本信息：文件名、文件物理位置、文件逻辑结构；
    - 存取控制信息：不同用户的存取权限；
    - 使用信息类：建立时间、当前的使用信息；
- 文件目录
  - 文件控制块的有序集合
  - 每个文件控制块称为目录项
  - 存放在外存

- 2 索引结点

- 查找目录时，需要多次启动外存
- 因查找时只用到文件名信息，只有找到了匹配的目录项，才会需要物理地址等其它信息
- 为减少从硬盘读文件目录的开销，将文件名与文件描述信息分开
- 文件描述信息单独形成一个数据结构，称为索引结点
- 文件目录项仅由文件名和指向索引结点的指针构成

## 7.3.2 目录结构

目录结构的组织，关系到文件系统的存取速度，共享性和安全性

- 单级目录
- 两级目录
- 多级目录

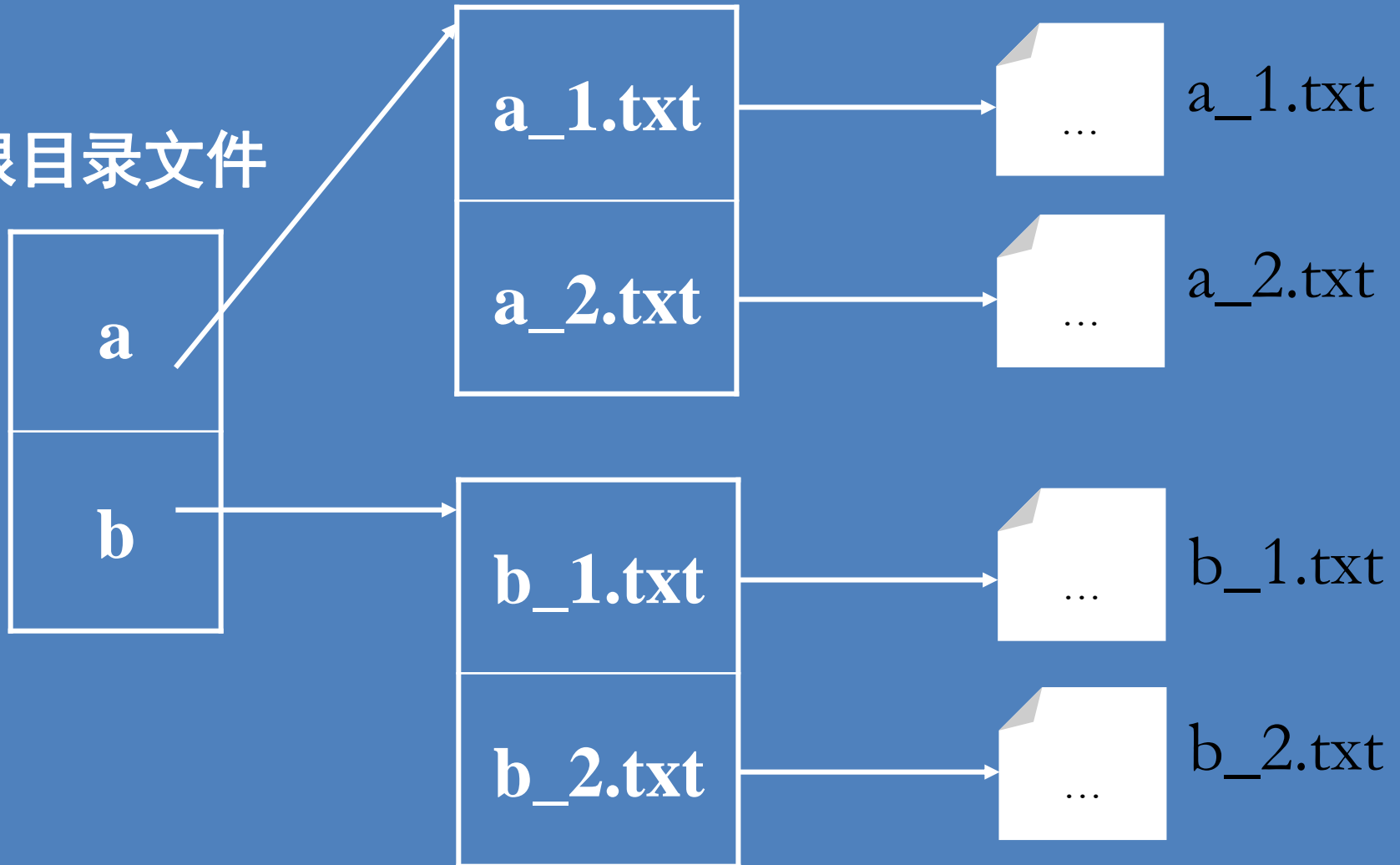
绝对路径,相对路径与当前目录(工作目录)

# 二级目录结构

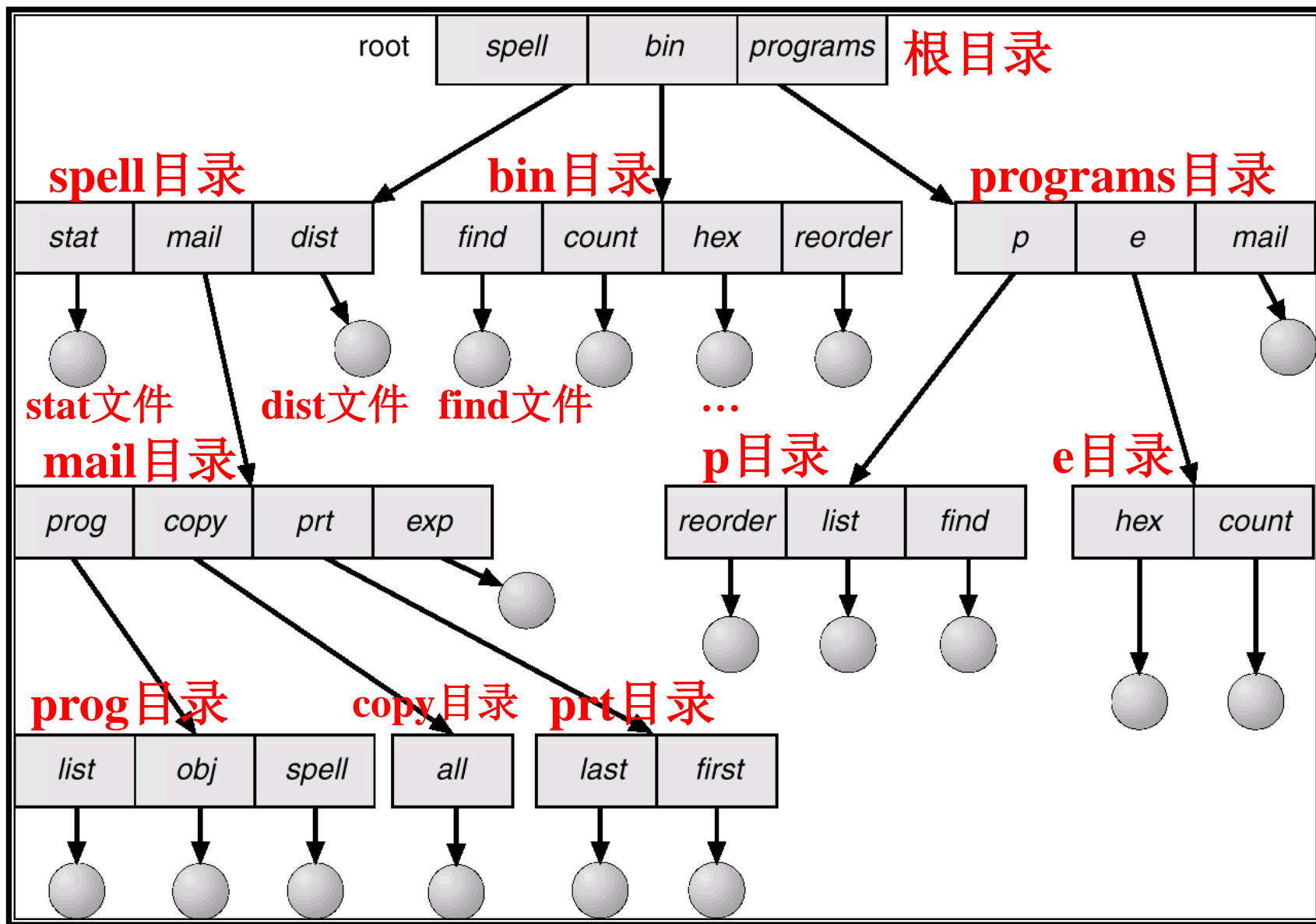
用户目录文件

普通文件

根目录文件



# 多级目录结构



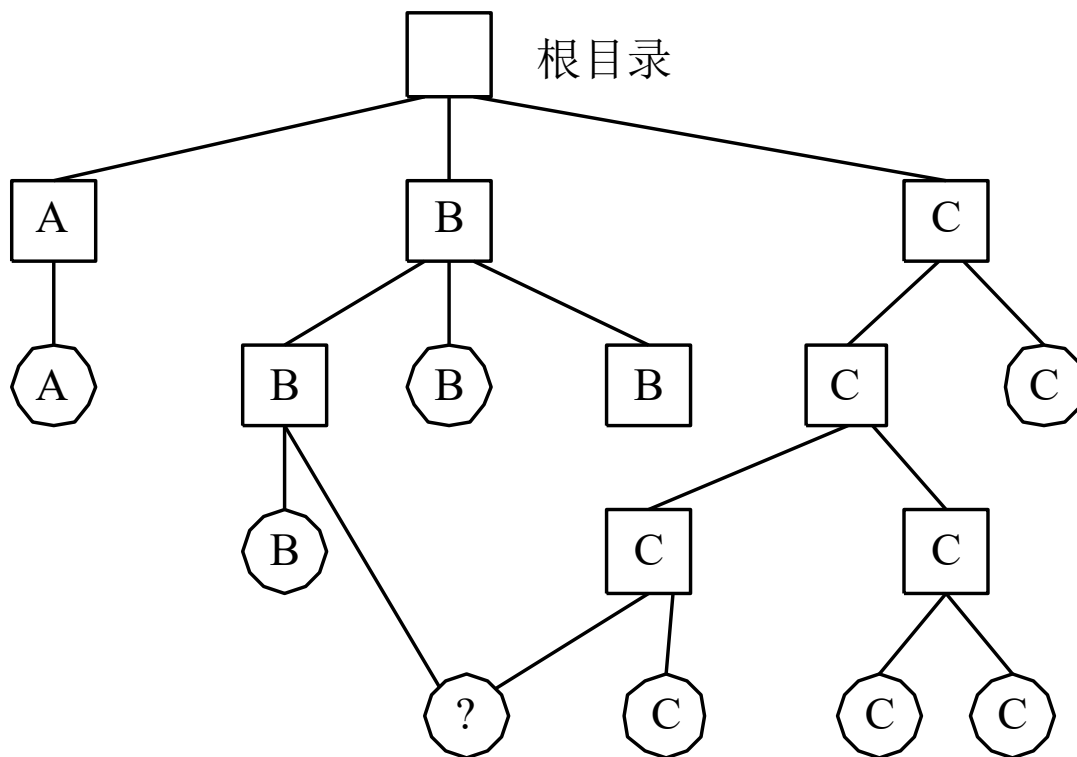
## 7.3.3 目录查询技术

- 用户要访问文件，首先通过文件名查询目录，找出文件控制块或对应索引结点；
- 然后根据文件物理地址（盘块号）算出在磁盘的物理位置；
- 最后通过磁盘驱动程序，将文件读入内存；
- 文件名检索：
  - 线性检索法
  - Hash方法

## 7.4 文件的共享

## 7.4.1 基于索引结点的共享方式

- 将共享的文件或目录链接到多个用户目录中



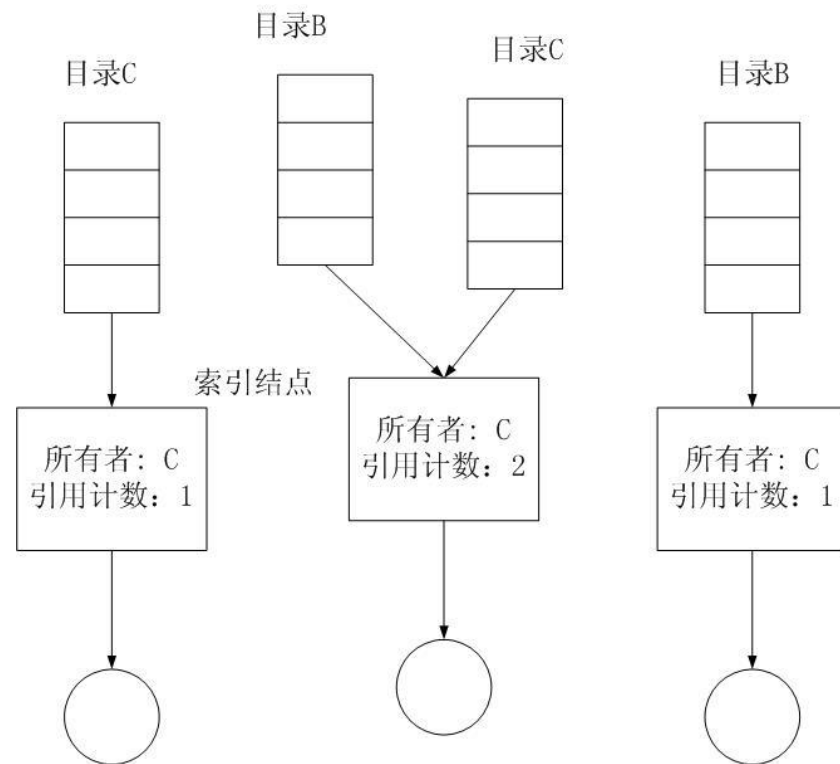


## 7.4.1 基于索引结点的共享方式

- 问题
  - 如果文件目录中包含共享文件的物理地址，则某用户对文件的增删操作对其它用户不可见，达不到共享的目的
- 解决方案
  - 文件的物理地址等其它属性信息放在单独的索引结点里；
  - 文件目录设置文件名和指向相应索引结点的指针；
  - 任何用户对文件的修改都会引起结点内容的改变，并对其它用户可见

# 硬链接

- Unix采用本方法。用户可使用link/unlink命令建立/去除连接。
- 当目录B欲共用目录C下的文件/子目录时，
  - B的目录文件内包含指向索引结点的记录项；
  - 索引结点内的引用计数增1。
- 删除文件时，引用计数减1。减至0才删除文件拷贝。



## 7.4.2 利用符号链实现文件共享

- 如 **B** 想共享文件**C**，为共享的文件建立一个**LINK**类型的新文件，该文件记录了被链接文件的路径名，将其放在 **B** 的目录下；
- 当 **B** 读 **LINK** 类型文件时，被**OS**截获，根据路径名读该文件
- 文件主删除文件后，其它用户再访问该文件失败后，再删除**LINK**文件

## 7.5 文件保护

- 文件应该在以下两个方面得到保护：
  - 防止软/硬件系统发生故障而导致文件不可用。这可以通过文件备份得到保护。
  - 防止系统的用户或外来者窃取、破坏文件或进行未授权的访问。这可以通过访问控制得到保护。

# 文件备份

- 文件备份即为文件复制多个副本存放于不同场所。分批量备份和同步备份两种。
  - **批量备份**是周期性地定期进行文件备份复制。分成全量转储和增量转储两种。
    - **全量转储**是把全部文件定期复制到备份用的存储设备上。
    - **增量转储**是每隔一段时间把上次备份以来改变了的或新增的文件复制到备份用的存储设备上。
  - **同步备份**是在生成新文件或写文件的同时备份文件。分镜像备份和双机动态文件备份两种。
    - **镜像备份**是对磁盘进行写操作同时对镜像磁盘实施同样的操作。
    - **双机动态文件备份**是把写操作同时派发给两台机器，两者完全对称地工作。

# 文件访问保护

- 口令保护
  - 用户使用文件时，要求其提供口令。
- 加密保护
  - 文件被加密。用户使用文件前要先解密，此时需提供密钥。
- 访问控制
  - 用户使用文件时，检查其操作类型与文件的权限设置是否相符。
  - 实现方法：
    - 访问控制矩阵：为每个文件建立一个访问控制矩阵，一一指出各用户或域对文件的访问权限。
    - 简化的访问表：为每个文件分别指出所有者，用户组和其他用户或域对文件的访问权限。

# 保护域

- 进程对一组对象访问权的集合
- 进程与域的关联---静态，动态

# 访问矩阵

对象 域	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>4</sub>	F <sub>5</sub>	F <sub>6</sub>	Printer 1	Plotter 2	域 D <sub>1</sub>	域 D <sub>2</sub>	域 D <sub>3</sub>
域D <sub>1</sub>	R	R, W								S	C
域D <sub>2</sub>			R	R, W, E	R, W		W				S
域D <sub>3</sub>						R, W, E	W	W			

图 7-19 具有切换权的访问控制矩阵



# 矩阵修改

拷贝， 所有， 控制

域 \ 对象	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>
D <sub>1</sub>	O, E		W
D <sub>2</sub>		R*, O	R*, O, W
D <sub>3</sub>	E		

域 \ 对象	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>
D <sub>1</sub>	O, E		
D <sub>2</sub>		O, R*, W*	R*, O, W
D <sub>3</sub>		W	W

域 \ 对象	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>4</sub>	F <sub>5</sub>	F <sub>6</sub>	Printer 1	Plotter 2	域 D <sub>1</sub>	域 D <sub>2</sub>	域 D <sub>3</sub>
域 D <sub>1</sub>	R	R, W								S	
域 D <sub>2</sub>			R	R, W, E	R, W		W				S
域 D <sub>3</sub>						R, W, E	W	W			

# 访问矩阵的实现

- 访问控制表
  - 为每个对象创建一个表
  - ACL（域，权集）
- 访问权限表
  - 每域建立一张表
  - （对象，权集）