

# 第7章 文件管理 (3 学时)



主讲教师：张春元

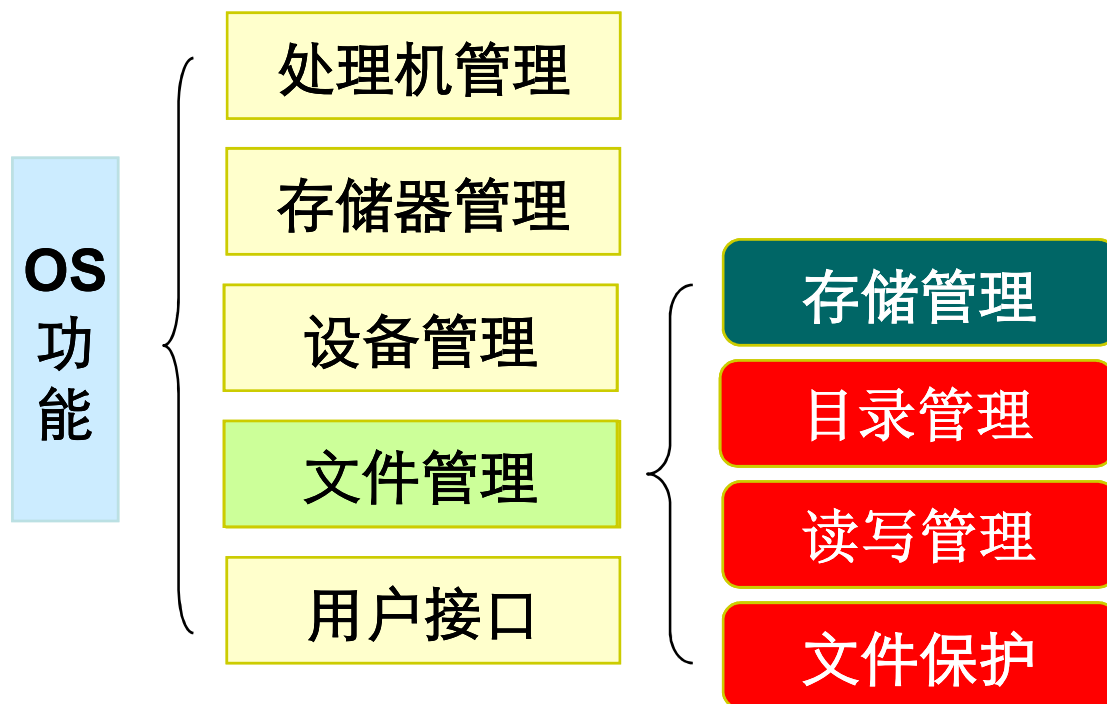
联系电话：13876004640

课程邮箱：[haidaos@126.com](mailto:haidaos@126.com)

邮箱密码：[zhangchunyuan](#)



# 本章内容所处位置





# 本章主要内容

- ❖ 7.1 文件和文件系统
- ❖ 7.2 文件的逻辑结构
- ❖ 7.3 目录管理
- ❖ 7.4 文件共享
- ❖ 7.5 文件保护

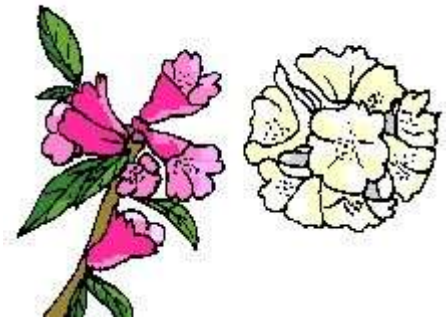




# 7.1 文件和文件系统

**文件**是具有文件名的若干相关元素的集合；OS通过**文件系统**把存储在计算机上的程序和数据组织成一系列文件来进行管理。

- ❖ 7.1.1 数据项、记录和文件
- ❖ 7.1.2 文件名和类型
- ❖ 7.1.3 文件系统的层次结构
- ❖ 7.1.4 文件操作





## 7.1.1 数据项、记录和文件

### ❖ 1、数据项

#### \* 1> 基本数据项

- 基本数据项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。
- 基本数据项除数据名外，还应有数据类型。
- 例如描述一个学生：学号、姓名、年龄、班级

#### \* 2> 组合数据项

- 组合数据项由若干基本数据项组成，简称组项。
- 例如工资包括基本工资、工龄工资、奖金等。



## 7.1.1 数据项、记录和文件

### ❖ 2、记录

- \* 记录是用于描述一个对象在某一方面的属性的一组相关数据项的集合。
- \* 在记录中，能唯一标识该记录的数据项称为**关键字(key)**。
- \* 一个记录应包含哪些数据项，取决于该对象的应用环境以及该对象需要描述的具体方面。
- \* 例如一个学生，当把他作为**班上**的一名学生时，对他的描述应使用**学号、姓名、年龄及所在系班**，也可能还包括他所学过的**课程的名称、成绩**等数据项。



## 7.1.1 数据项、记录和文件

### ❖ 3、文件

- \* **文件**是指具有文件名的一组相关元素的集合。
- \* 文件是文件系统中**最大的数据单位**。
- \* 文件可分为有结构文件和无结构文件：
  - 有结构文件是指由**若干个相关记录**组成的文件，如一个班的学生记录组成的文件。
  - 无结构文件是由一串**字符流**构成的文件，也称**流式文件**，如一段源程序构成的文件。
- \* 文件的属性：文件类型、文件长度、文件的物理位置、文件建立的时间等。



## 7.1.1 数据项、记录和文件

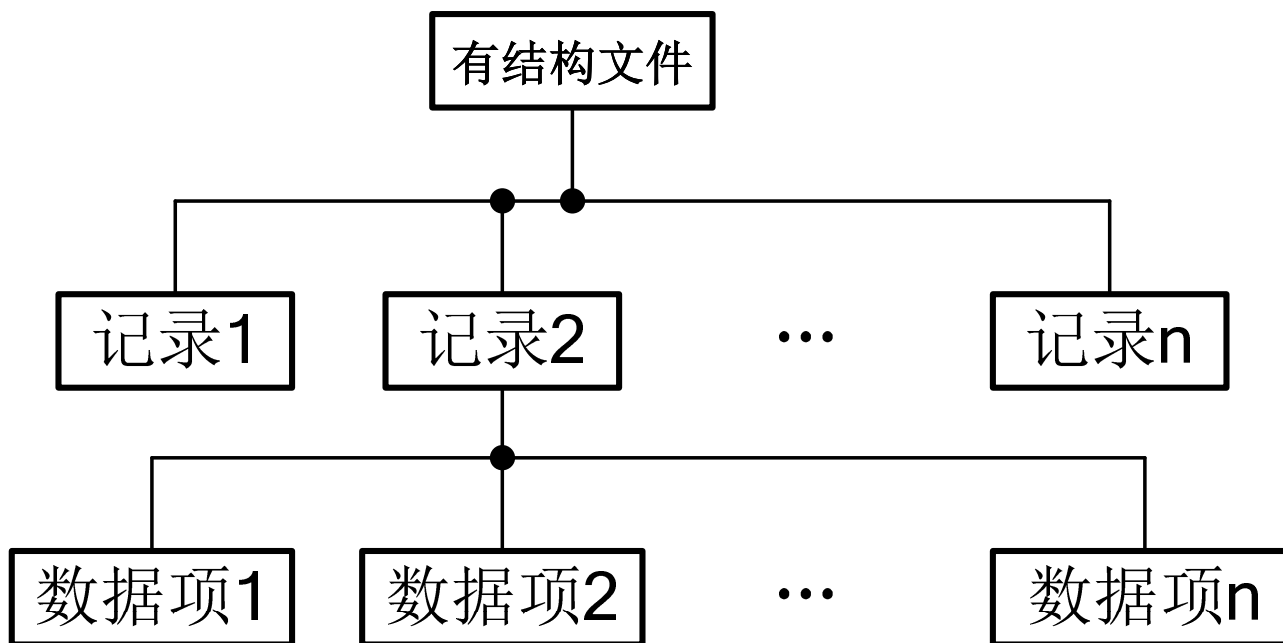


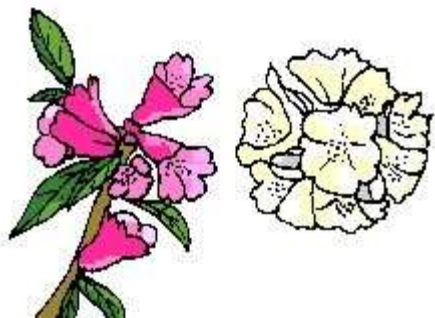
图7-1 有结构文件、记录和数据项之间的层次关系





# 7.1 文件和文件系统

- ❖ 7.1.1 数据项、记录和文件
- ❖ 7.1.2 文件名和类型
- ❖ 7.1.3 文件系统的层次结构
- ❖ 7.1.4 文件操作





## 7.1.2 文件名和类型

### ❖ 1、文件名和扩展名

#### \* 文件名

- 不同的操作系统对文件的规定是不同的
  - MS-DOS最多只允许文件名8个字符长
  - Windows不作区分文件名大小写
  - Unix和Linux区分文件名大小写

#### \* 扩展名（后缀名）

- 表示文件类型，长度一般是1-4个字符



## 7.1.2 文件名和类型

### ❖ 2、文件类型

#### \* 1> 按用途分类

##### ■ 系统文件

- 由操作系统软件构成的文件。这类文件对用户一般不直接开放，不允许用户读、改，只能通过系统调用为用户服务。

##### ■ 用户文件

- 如源程序文件、目标程序文件、可执行文件以及由原始数据、计算结果等组成的文件。

##### ■ 库文件

- 由标准子程序及常用的应用程序组成的文件。这类文件允许用户调用，但不允许用户修改。



## 7.1.2 文件名和类型

### \* 2> 按文件中数据的形式分类

#### ■ 源文件

— 由源程序和数据构成的文件。

#### ■ 目标文件

— 源程序经过相应的编译程序编译生成但尚未经过链接程序链接的目标代码构成文件，目标文件属于二进制文件。

#### ■ 可执行文件

— 目标代码经过链接程序链接后形成的文件。



## 7.1.2 文件名和类型

### \* 3> 按存取控制属性分类

#### ■ 只执行文件

- 只允许被核准的用户调用执行，但不允许读、更不允许写的文件。

#### ■ 只读文件

- 允许文件主及核准的用户读，但不允许写的文件。

#### ■ 读写文件

- 允许文件主及核准的用户读、写，但禁止未核准的用户读、写的文件。



## 7.1.2 文件名和类型

### \* 4> 按组织和处理方式分类

#### ■ 普通文件

- 由ASCII码或二进制码组成的字符文件。用户文件、系统文件、库文件大多属于普通文件。

#### ■ 目录文件

- 由文件目录组成，通过目录文件可以对其下属文件的信息进行检索、修改，可像普通文件一样对目录文件进行操作。

#### ■ 特殊文件

- 特指系统中的各类I/O设备。在OS中，将I/O设备视作文件进行管理。
- 根据设备数据交换单位的不同，特殊文件分为块设备文件和字符设备文件。



## 7.1.2 文件名和类型

### \* 5> 按逻辑结构分类

- 有结构文件
  - 记录式文件
- 无结构文件
  - 流式文件

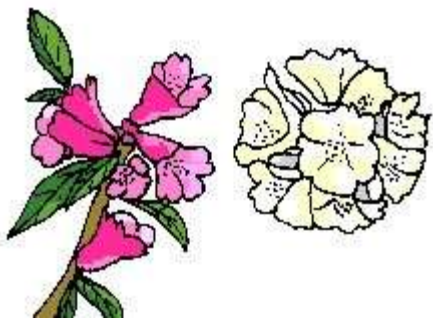
### \* 6> 按物理安排分类

- 顺序文件
- 链接文件
- 索引文件



# 7.1 文件和文件系统

- ❖ 7.1.1 数据项、记录和文件
- ❖ 7.1.2 文件名和类型
- ❖ 7.1.3 文件系统的层次结构
- ❖ 7.1.4 文件操作







## 7.1.3 文件系统的层次结构

### ❖ 1、文件系统模型

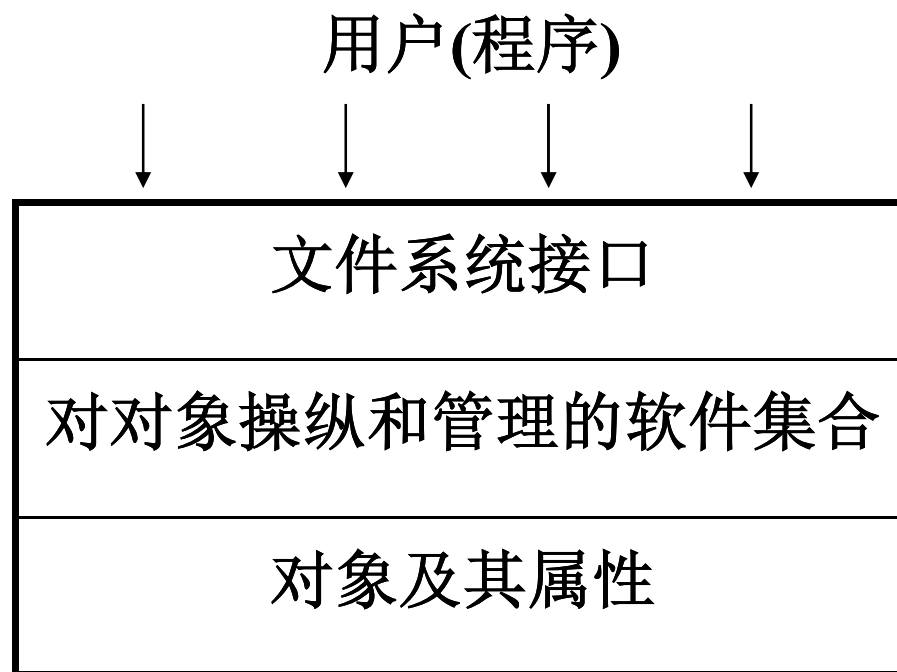


图7-2 文件系统模型



## 7.1.3 文件系统的层次结构

### \* 1>文件系统管理的对象及其属性

- ① 文件

- 作为文件管理的直接对象

- ② 目录

- 目录项：文件名 $\longleftrightarrow$ 文件的物理地址（或指针）

- 在文件系统中配置目录便于用户使用（提供文件逻辑名来访问文件）和提高文件存取速度

- ③ 磁盘(磁带)存储空间

- 提高外存的利用率和文件的存取速度



## 7.1.3 文件系统的层次结构

### \* 2> 对对象操纵和管理的软件集合

- 是整个文件管理系统的核心部分
- 本层软件具体功能：
  - 对文件存储空间进行管理
  - 对文件目录进行管理
  - 将文件的逻辑地址转换为物理地址
  - 对文件读和写进行管理
  - 提供文件共享与保护功能



## 7.1.3 文件系统的层次结构

- 本层软件通常按功能组织成四个层次
  - I/O控制层（设备驱动程序层）
    - » 主要由磁盘驱动程序等组成
  - 基本文件系统层
    - » 主要用于处理内存与磁盘间数据块的交换
  - 基本I/O管理程序
    - » 完成与磁盘I/O有关的事务
  - 逻辑文件系统
    - » 用于处理与记录和文件相关的操作



## 7.1.3 文件系统的层次结构

### \* 3> 文件系统的接口

#### ■ ① 命令接口

– 是用户与文件系统交互的接口。用户可通过键盘终端键入命令取得文件系统的服务。

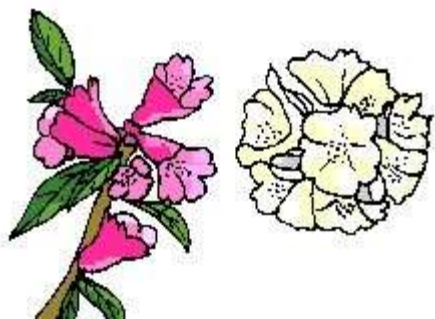
#### ■ ② 程序接口

– 是用户程序与文件系统的接口。用户程序可通过系统调用来取得文件系统的服务。



# 7.1 文件和文件系统

- ❖ 7.1.1 数据项、记录 and 文件
- ❖ 7.1.2 文件名和类型
- ❖ 7.1.3 文件系统的层次结构
- ❖ 7.1.4 文件操作





## 7.1.4 文件操作

### ❖ 1、最基本的文件操作

#### \* 1> 创建文件

- 分配必要的外存空间，在文件系统的目录中建立一个目录项。

#### \* 2> 删除文件

- 先从目录中删除文件的目录项，然后回收其存储空间。

#### \* 3> 读文件

- 查找到指定的目录项，从外存读文件到内存。

#### \* 4> 写文件

- 查找到指定的目录项，进行写操作。



## 7.1.4 文件操作

### \* 5> 截断文件（清空文件）

- 将文件长度设成0，放弃原有文件内容。

### \* 6> 设置文件的读/写位置

- 设置文件读/写指针的位置，以便能将文件的顺序存取方式改为随机存取方式。





## 7.1.4 文件操作

### ❖ 2、文件的“打开”和“关闭”操作

- \* OS中引入“打开”、“关闭”系统调用，避免在对同一文件实施多次操作时重复地到外存上检索目录。

- \* 1> “打开”操作

- 用户对文件第一次操作时，将文件的属性从外存拷贝到内存中的打开文件表的一个表目中，并将该表目的编号（索引号）返回给用户；当用户再次对该文件进行操作时，便可直接利用该索引号到打开文件表中去查找，从而避免了到目录中再次检索该文件。

- \* 2> “关闭”操作

- 用户不再需要对文件进行操作时，OS将会把该文件在打开文件表中的表目删除掉。



## 7.1.4 文件操作

### ❖ 3、其它文件操作

#### \* 1> 文件属性操作

- 改变文件名
- 改变拥有者
- 修改权限
- 查询状态

#### \* 2> 目录操作

- 创建目录
- 删除目录
- 改变当前目录

#### \* 3> 实现文件共享的系统调用

#### \* 4> 用于对文件系统进行操作的系统调用



# 本章主要内容

- ❖ 7.1 文件和文件系统
- ❖ 7.2 文件的逻辑结构
- ❖ 7.3 目录管理
- ❖ 7.4 文件共享
- ❖ 7.5 文件保护





## 7.2 文件的逻辑结构

### ❖ 文件内容的组织及存储结构

#### \* 1> 文件的逻辑结构（文件内容的组织结构）

- 是进行文件系统**高层设计**所涉及的主要问题
- 文件的逻辑结构是指用户所能观察和访问到的文件内容组织形式，独立于物理特性，应使文件内容**易检索、易修改、易存储**。

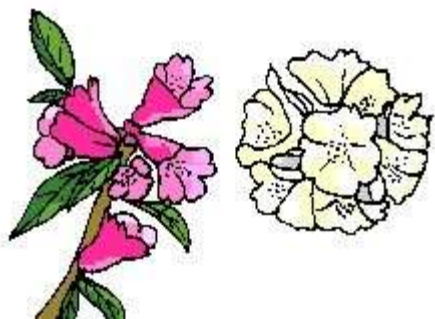
#### \* 2> 文件的物理结构（文件内容的存储结构）

- 是进行文件系统**低层设计**所涉及的主要问题
- 文件的物理结构是指文件内容在外存上的存储组织形式，与存储介质的存储性能、所采用的外存分配方式有关。



## 7.2 文件的逻辑结构

- ❖ 7.2.1 文件逻辑结构的类型
- ❖ 7.2.2 顺序文件
- ❖ 7.2.3 索引文件
- ❖ 7.2.4 索引顺序文件
- ❖ 7.2.5 直接文件和哈希文件





## 7.2.1 文件逻辑结构的类型

### ❖ 1、有结构文件（记录式文件）

#### \* 1> 按记录长度分类

- 定长记录

- 各记录长度相同，文件长度可用记录数目表示

- 变长记录

- 各记录长度不尽相同

#### \* 2> 按组织方式分类

- 顺序文件

- 记录（多指定长记录）按某种顺序组织成的文件

- 索引文件

- 按索引方式组织成的文件，多用于变长记录，每条记录建立一个索引项

- 索引顺序文件

- 将记录按某种顺序分成多个组，对每组记录中的第一个记录设置一索引项



## 7.2.1 文件逻辑结构的类型

### ❖ 2、无结构文件（流式文件）

- \* 由字符流构成的文件，其长度以字节为单位
- \* 采用读/写指针来指出下一个要访问的字符
- \* 如源程序、可执行文件、库函数等



## 7.2.1 文件逻辑结构的类型

### ❖ 3、两种文件的比较

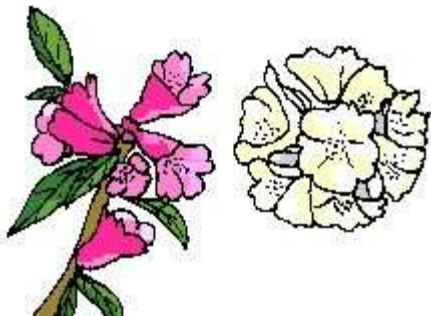
- \* **流式文件**就像给一张**白纸**给用户，用户可将他的信息任意地写到纸上，没有任何格式上的限制。
- \* **记录式文件**就像给一张**表格**给用户，用户要按表规定的格式填信息。
- \* 显然，有结构式文件对用户的限制很大，使用起来就不方便。在UNIX系统中，所有文件都被看作是流式文件，即使是有结构文件，也被视为流式文件。





## 7.2 文件的逻辑结构

- ❖ 7.2.1 文件逻辑结构的类型
- ❖ 7.2.2 顺序文件
- ❖ 7.2.3 索引文件
- ❖ 7.2.4 索引顺序文件
- ❖ 7.2.5 直接文件和哈希文件





## 7.2.2 顺序文件

### ❖ 1、逻辑记录的排序

#### \* 1> 串结构

- 按记录录入的时间排序，与关键字无关

#### \* 2> 顺序结构

- 按关键字排序
- 顺序结构更有利于提高查询速度，如可用折半查找法等。

### ❖ 2、对顺序文件的读/写操作（图7-3）

#### \* 1> 定长记录顺序文件

- 易于迅速定位，甚至可随机读取（直接读取）

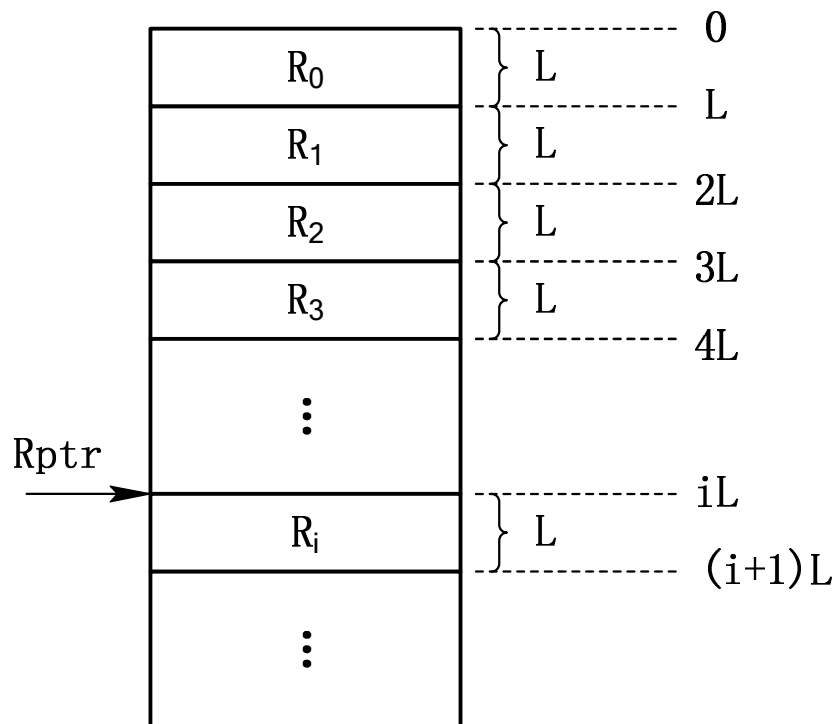
#### \* 2> 变长记录顺序文件

- 不易迅速定位，通常只能顺序存取



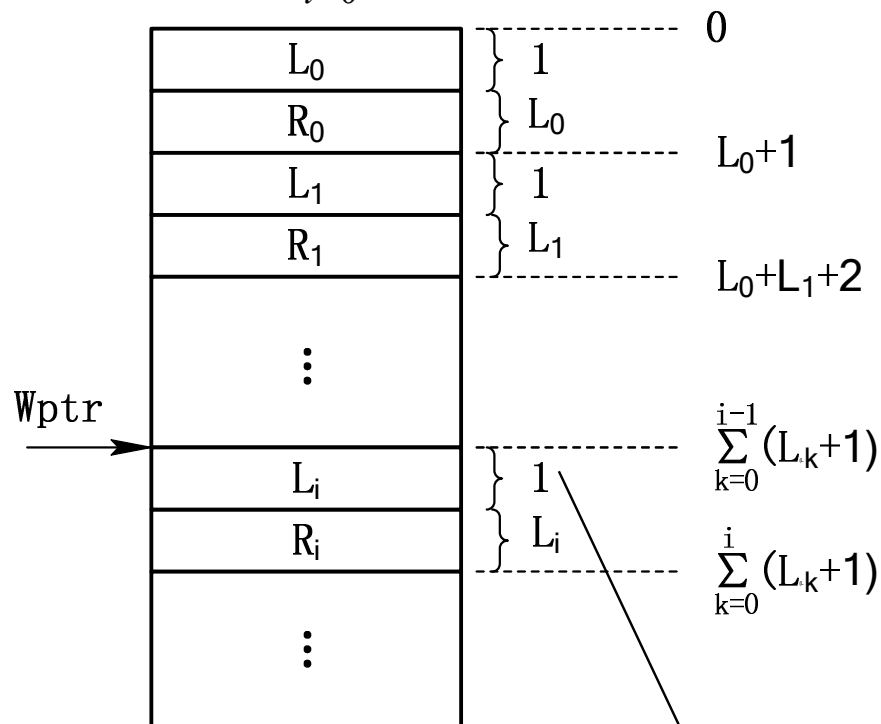
## 7.2.2 顺序文件

$$A_i = i \times L$$



(a) 定长记录文件

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



(b) 变长记录文件

图7-3 定长和变长记录文件

用一个字节  
指明该记录  
的长度



## 7.2.2 顺序文件

### ❖ 3、顺序文件的优缺点

#### \* 1> 优点

- 顺序文件的最佳应用场合，是在对诸记录进行批量存取时，即每次要读或写一大批记录
- 只有顺序文件才能存储在磁带上，并能有效地工作

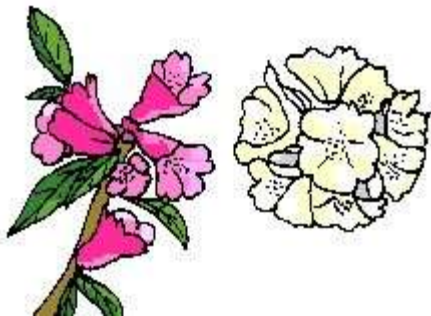
#### \* 2> 缺点

- 对串结构的顺序文件：如果用户(程序)要求查找或修改单个记录，通常只能顺序查找，其平均查找长度很大，顺序文件所表现出来的性能就可能很差。
- 对顺序结构的顺序文件：如果想增加或删除一个记录，需移动大量数据。



## 7.2 文件的逻辑结构

- ❖ 7.2.1 文件逻辑结构的类型
- ❖ 7.2.2 顺序文件
- ❖ 7.2.3 索引文件
- ❖ 7.2.4 索引顺序文件
- ❖ 7.2.5 直接文件和哈希文件





## 7.2.3 索引文件

### ❖ 概念

- \* 由于变长记录组成的顺序文件不容易直接存取，因此，为其建立一张**有序索引表**（按各记录的键值排序建立索引，索引表本身是一个定长记录顺序文件），用来记录各条记录的长度及指向各记录的指针，检索记录时先对索引表**折半查找**。

### ❖ 索引文件文件

- \* 单索引表索引文件、多索引表索引文件

### ❖ 优缺点

- \* 提高了检索速度
- \* **增加了存储开销（存放索引文件）**
- \* 增、删记录时，需对索引表作相应的修改



## 7.2.3 索引文件

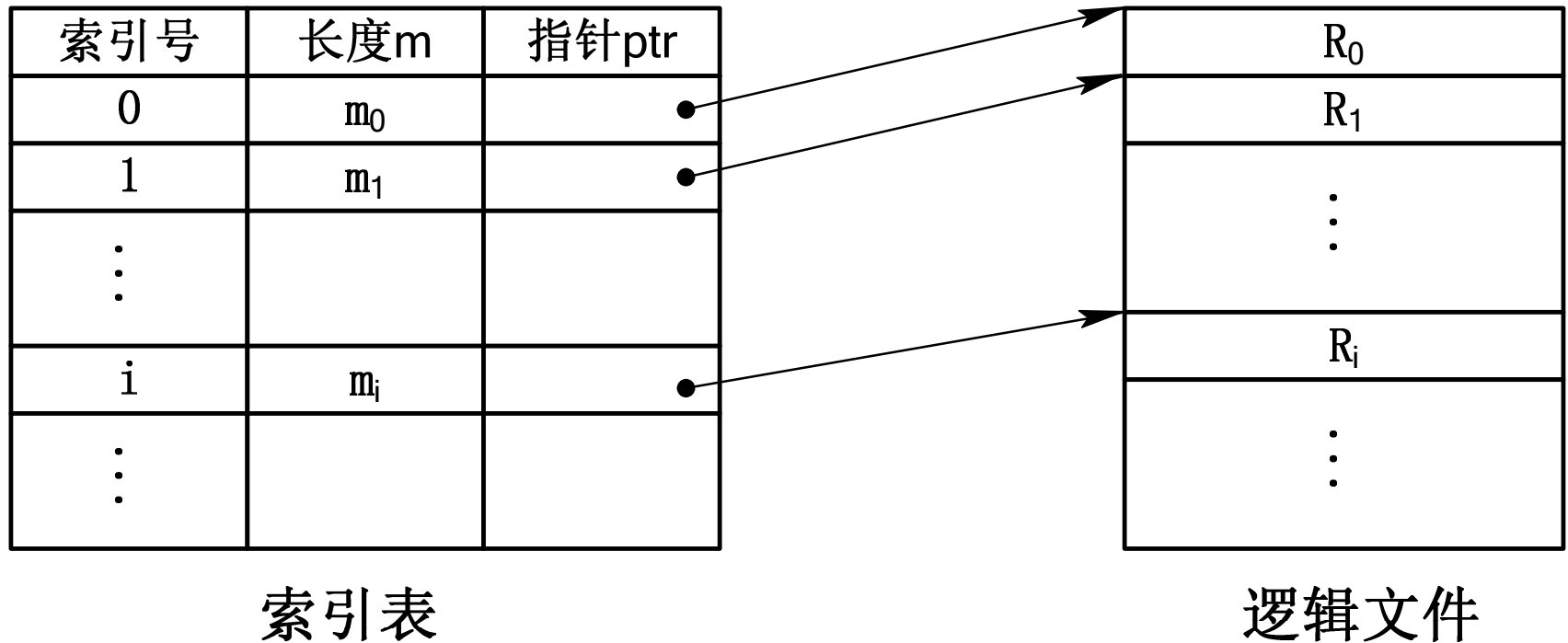
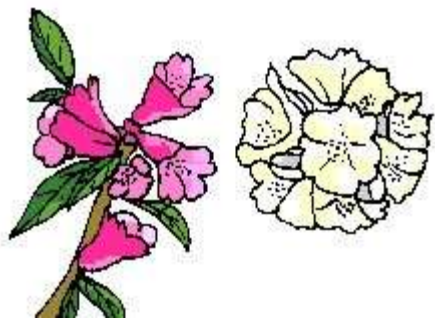


图7-4 索引文件的组织



## 7.2 文件的逻辑结构

- ❖ 7.2.1 文件逻辑结构的类型
- ❖ 7.2.2 顺序文件
- ❖ 7.2.3 索引文件
- ❖ 7.2.4 索引顺序文件
- ❖ 7.2.5 直接文件和哈希文件







## 7.2.4 索引顺序文件

### ❖ 1、索引顺序文件

- \* **索引顺序文件(Index Sequential File)** 是顺序文件与索引文件的结合，将顺序文件中的所有记录分为若干个组，为每组中的第一个记录建立索引项，其中含有该记录的键值及指向该记录的指针。
- \* 索引顺序文件**克服了变长记录顺序文件不便直接存取**的缺点，**代价也不太大**。



## 7.2.4 索引顺序文件

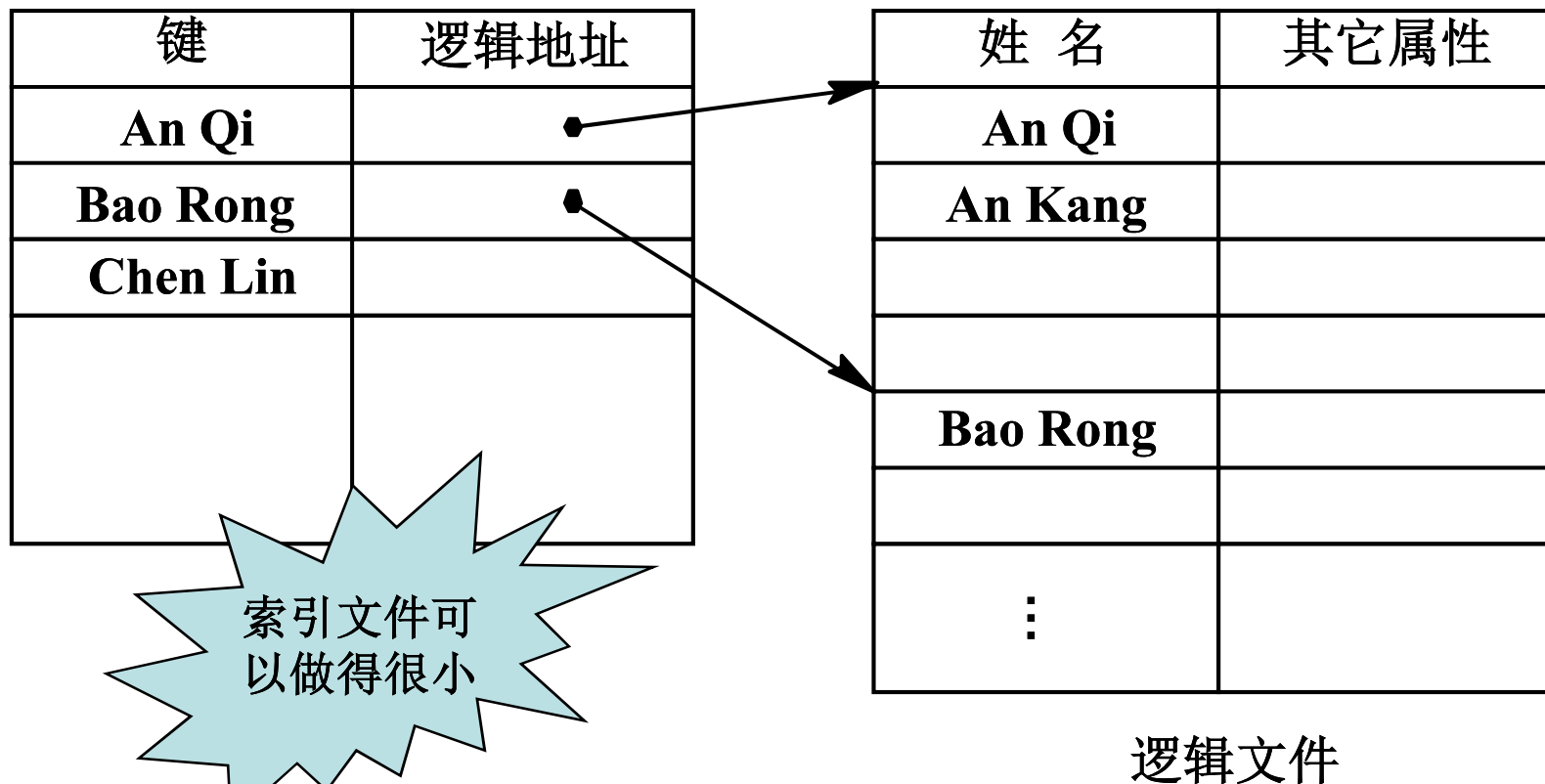


图7-5 索引顺序文件



## 7.2.4 索引顺序文件

### ❖ 2、索引顺序文件顺序查找速度分析

#### \* 【例1】10000条记录顺序查找

- **顺序文件**：每一条记录平均须查找5000次。
- **索引顺序文件**：设100条记录分成一组，则索引表表项数为100，索引表、各组记录均采用顺序查找，则每一条记录平均查找次数为 $50+50=100$ 次。

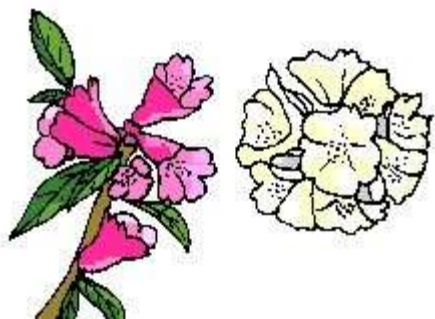
#### \* 【例2】1000000条记录顺序查找

- **一级索引**：若100条记录分成一组，则每一条记录平均查找5050次。
- **二级索引**：若100条记录分成一组，建两级索引表，则每一条记录平均查找 $50+50+50=150$ 次。



## 7.2 文件的逻辑结构

- ❖ 7.2.1 文件逻辑结构的类型
- ❖ 7.2.2 顺序文件
- ❖ 7.2.3 索引文件
- ❖ 7.2.4 索引顺序文件
- ❖ 7.2.5 直接文件和哈希文件





## 7.2.5 直接文件和哈希文件

### ❖ 1、键值转换(Key to address transformation)

- \* 指将记录的键值（关键字）转换成记录的物理地址。

### ❖ 2、直接文件

- \* 根据给定的记录键值，**通过键值转换直接获得指定记录的物理地址**；而不用像前面需要通过查找索引表来获得指定记录的物理地址。

#### \* 哈希(Hash)文件

- 利用哈希函数将记录键值转换为相应记录的地址
- 为了能实现文件存储空间的动态分配，通常由哈希函数求得的不是相应记录的物理地址，而是指向一目录表相应表目的指针，由该表目的内容指向相应记录所在的物理块。



## 7.2.5 直接文件和哈希文件

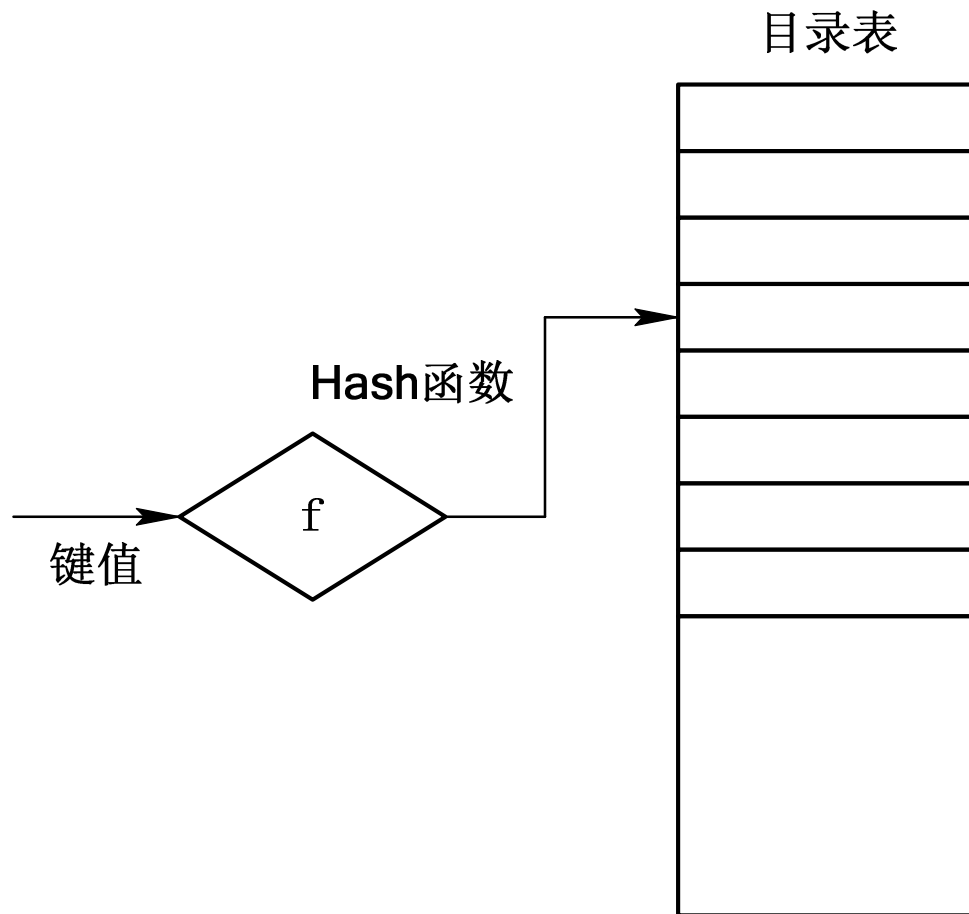


图7-6 Hash文件的逻辑结构



# 本章主要内容

- ❖ 7.1 文件和文件系统
- ❖ 7.2 文件的逻辑结构
- ❖ 7.3 目录管理
- ❖ 7.4 文件共享
- ❖ 7.5 文件保护





## 7.3 目录管理

### ❖ 文件目录

#### \* 概念

- 文件目录是一种数据结构，用于标识系统中文件的名称、属性、物理地址等信息，供检索时使用。

#### \* 目录管理所追求的四大目标（目录管理的功能）

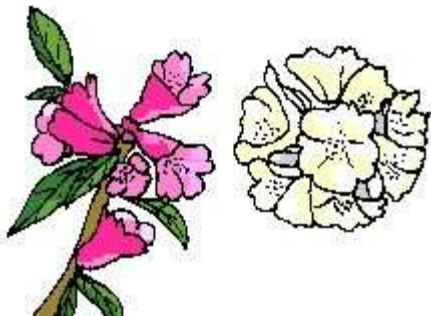
- 实现“按名存取”。（目录管理最基本的功能）
- 提高对目录的检索速度。
- 文件共享。
- 允许文件重名。





## 7.3 目录管理

- ❖ 7.3.1 文件控制块和索引结点
- ❖ 7.3.2 目录结构
- ❖ 7.3.3 目录查询技术





## 7.3.1 文件控制块和索引结点

### ❖ 1、文件控制块FCB（目录项）

#### \* 概念

- FCB是描述文件基本信息、存取控制信息和使用信息的数据结构。
- 文件控制块的有序集合称为文件目录，即一个文件控制块就是一个文件目录项。
- 一个文件目录本身也被看作是一个文件，称为目录文件。

文件名	扩展名	属性	备用	时间	日期	第一块号	盘块数
-----	-----	----	----	----	----	------	-----

图7-7 MS-DOS的文件控制块



## 7.3.1 文件控制块和索引结点

### \* FCB数据结构组成

#### ■ 1> 基本信息类

- 文件名、文件物理位置（设备名、起始盘块号、文件大小）、文件逻辑结构（有结构文件or无结构文件、定长记录or变长记录）、文件的物理结构（顺序文件or链接文件or索引文件）

#### ■ 2> 存取控制信息类

- 文件主/核准用户/一般用户存取权限

#### ■ 3> 使用信息类

- 文件的建立时间、文件上一次修改时间、当前使用信息（使用进程数、在内存中是否已被修改等）



## 7.3.1 文件控制块和索引结点

### ❖ 2、索引结点 (i结点)

#### \* 1> 索引结点的引入

- **问题：**如果把文件的描述信息全部放在FCB中，当文件很多时，目录文件将占用大量磁盘空间，检索时将频繁启动磁盘；在目录检索过程中，实际上仅用到了文件名，仅当找到目录项后才读出相关信息。
- **解决办法：**将FCB中的信息分离，将文件描述信息单独形成一个数据结构即索引结点，目录项中仅设置文件名和索引结点指针。
- **优点：**引入索引结点后，可使文件的目录项更小，从而减少磁盘启动次数，加快检索速度。



## 7.3.1 文件控制块和索引结点

### \* 2> 磁盘索引结点

- ①文件主标识符
  - 拥有该文件的个人或小组的标识符
- ②文件类型
  - 普通文件、目录文件或特别文件
- ③文件存取权限
  - 各类用户对该文件的存取权限
- ④文件物理地址
  - 13个地址项（UNIX系统），给出文件所在盘块编号
- ⑤文件长度
  - 以字节为单位的文件长度



## 7.3.1 文件控制块和索引结点

- ⑥文件连接计数
  - 指向该文件的指针的个数
- ⑦文件存取时间
  - 指出最近被进程存取的时间、最近被修改的时间及索引结点最近被修改的时间



## 7.3.1 文件控制块和索引结点

### \* 3> 内存索引结点

文件打开时，磁盘索引结点将被复制到内存，另行增加如下内容形成内存索引结点：

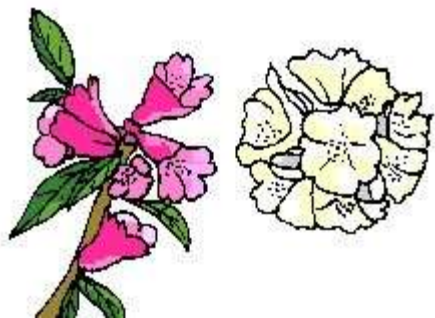
- ①索引结点编号
  - 用于标识内存索引结点
- ②状态
  - 指示本结点是否上锁或被修改。
- ③访问计数
  - 每当有一进程要访问本结点时，将该访问计数加1，访问完再减1
- ④文件所属文件系统的逻辑设备号
- ⑤链接指针
  - 设置有分别指向空闲链表和散列队列的指针



## 7.3 目录管理

- ❖ 7.3.1 文件控制块和索引结点
- ❖ 7.3.2 目录结构
- ❖ 7.3.3 目录查询技术

目录结构的组织，关系到文件的存取速度、安全性和共享性







## 7.3.2 目录结构

### ❖ 1、单级目录结构（最简单的目录结构）

- \* 整个系统只建立一张目录表，每个文件占一个目录项，目录项中含文件名、文件扩展名、文件长度、文件类型、文件物理地址和其它文件属性。

文件名	物理地址	文件说明	状态位
文件名1			
文件名2			
...			

图7-9 单级目录



## 7.3.2 目录结构

### \* 单级目录结构优点

- (1) 易于实现，管理简单
- (2) 能实现按名存取

### \* 单级目录结构缺点

- (1) 查找速度慢(顺序查找，平均查找 $N/2$ 次)
- (2) 不允许重名(在多道程序设计下，很难保证)
- (3) 不便于实现文件共享(所有用户必须用同一个名字共享一个文件)
- 单级目录只实现了目录管理所要求的四项功能中的第一项功能，即“按名存取”，只适用于单用户环境。



## 7.3.2 目录结构

### ❖ 2、两级目录结构 (MFD+UFDs)

- \* **UFD**: 为每个用户建立一个单独的**用户文件目录UFD** (User File Directory), 由用户所有文件的FCB组成。
- \* **MFD**: 在系统中再建立**主文件目录MFD** (Master File Directory), 每个用户目录文件在主文件目录中占一个目录项。
- \* **优缺点**
  - 1> 提高了检索速度
  - 2> 在不同用户目录中, 可重名
  - 3> 可共享, 但不方便



## 7.3.2 目录结构

### 主文件目录MFD

用户名	指向子目录指针
Wang	
Zhang	
Gao	

### 用户文件目录UFD

#### Wang用户目录

Alpha	●
Test	●
D	●

Alpha

Test

#### Zhang用户目录

Report	●
Test	●

Report

Test

#### Gao用户目录

Beta	●
Device	●
Misx	●

Beta

Device

Misx

图7-10 两级目录结构



## 7.3.2 目录结构

### ❖ 3、多级目录结构 (树型目录结构, 大多数操作系统采用)

#### \* 1> 目录结构

- 主目录称为**根目录**, 数据文件称为**树叶**, 其他目录作为树的**结点**。
- 目录中的目录项可以是**目录文件的FCB**, 也可以是**数据文件的FCB**。

#### \* 2> 路径名

- 在树形目录结构中, 从树的根(即主目录)开始, 把**全部目录文件名与数据文件名**, 依次地用“/”连接起来, 即构成该数据文件的**路径名**。
- 系统中的每一个文件都有**唯一**的路径名。

## 7.3.2 目录结构

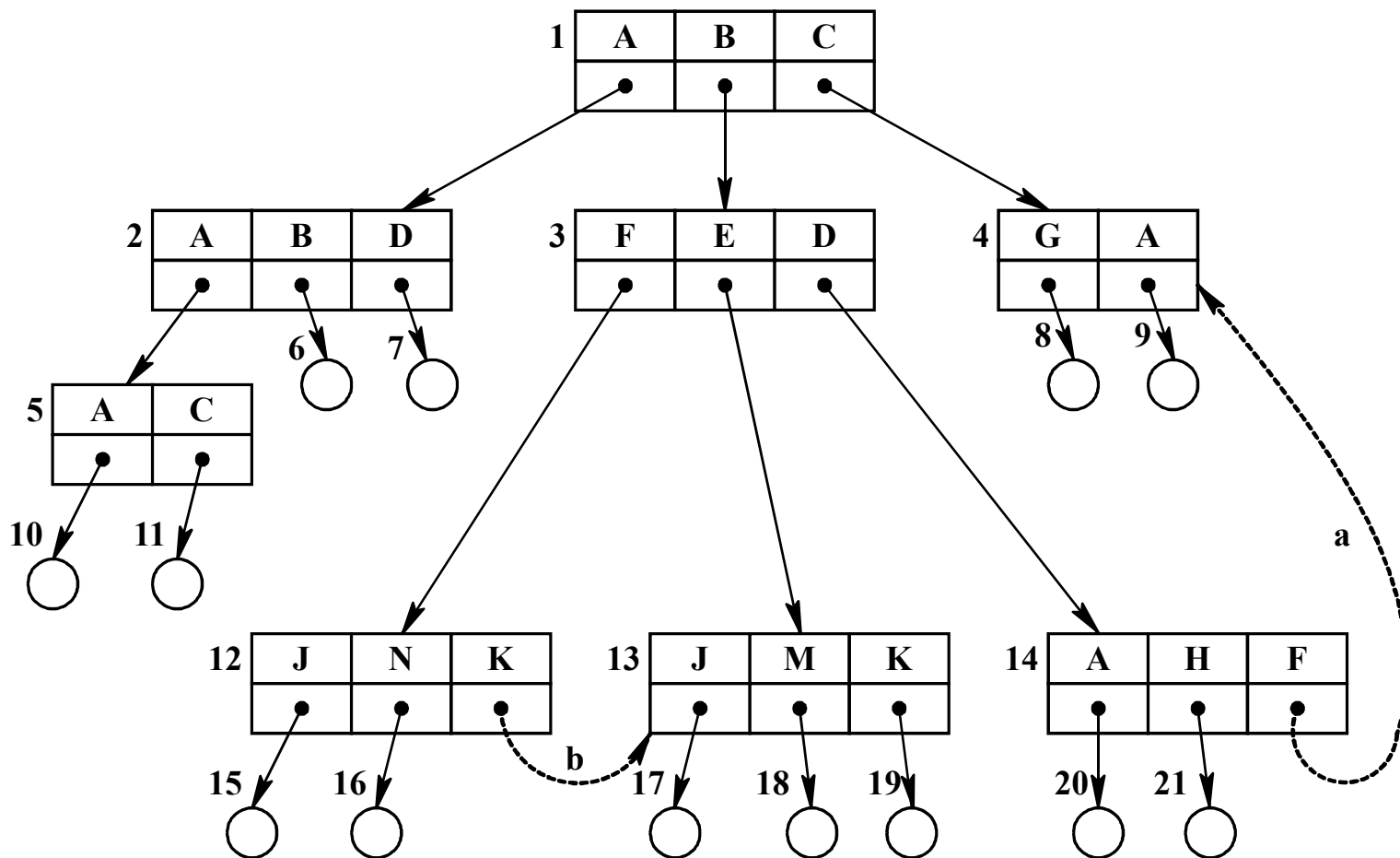


图7-11 多级目录结构



## 7.3.2 目录结构

### \* 3> 当前目录（工作目录）

- 由于一个进程运行时所访问的文件大多仅局限于某一范围，为了加快检索速度，可为各进程设置一个“**当前目录**”，进程对各文件的访问相对于其当前目录进行。
- 从当前目录开始直到数据文件为止所构成的路径名，称为**相对路径名**(relative path name)
- 从树根开始的路径名称为**绝对路径名**(absolute path name)。



## 7.3.2 目录结构

### \* 4> 多级目录结构优缺点

#### ■ 优点

- 检索速度较快
- 层次结构清晰
- 可对文件进行有效保护（容易对不同层次、不同子树中的文件设置不同存取权限）
- 解决了用户文件重名问题

#### ■ 缺点

- 查找文件时，需按路径名逐级访问中间结点，**增加了磁盘访问次数**，在一定程度影响了检索速度。





## 7.3.2 目录结构

### ❖ 4、目录操作

#### \* 1> 创建目录

- 添加子目录或文件时，只要不与其所加入的目录文件的现有目录名或文件名同名即可。

#### \* 2> 删除目录

##### ▪ 不删除非空目录

- 当目录不空时，不能将其直接删除；如要删除，必须先删除目录中的所有文件，后删除该目录。
- MS-DOS采用这种删除方式。

##### ▪ 可删除非空目录

- 当要删除一目录时，如果在该目录中还包含有文件，则目录中的所有文件和子目录也同时被删除。
- Windows采用这种删除方式。



## 7.3.2 目录结构

### \* 3> 改变目录

- 用户可通过指定路径名来改变当前目录

### \* 4> 移动目录

### \* 5> 链接操作

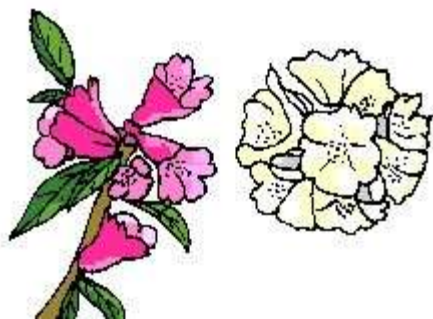
- 可通过链接操作实现共享，让一个文件具有多个父目录

### \* 6> 查找



## 7.3 目录管理

- ❖ 7.3.1 文件控制块和索引结点
- ❖ 7.3.2 目录结构
- ❖ 7.3.3 目录查询技术





## 7.3.3 目录查询技术

### ❖ 1、线性检索法（顺序检索法）

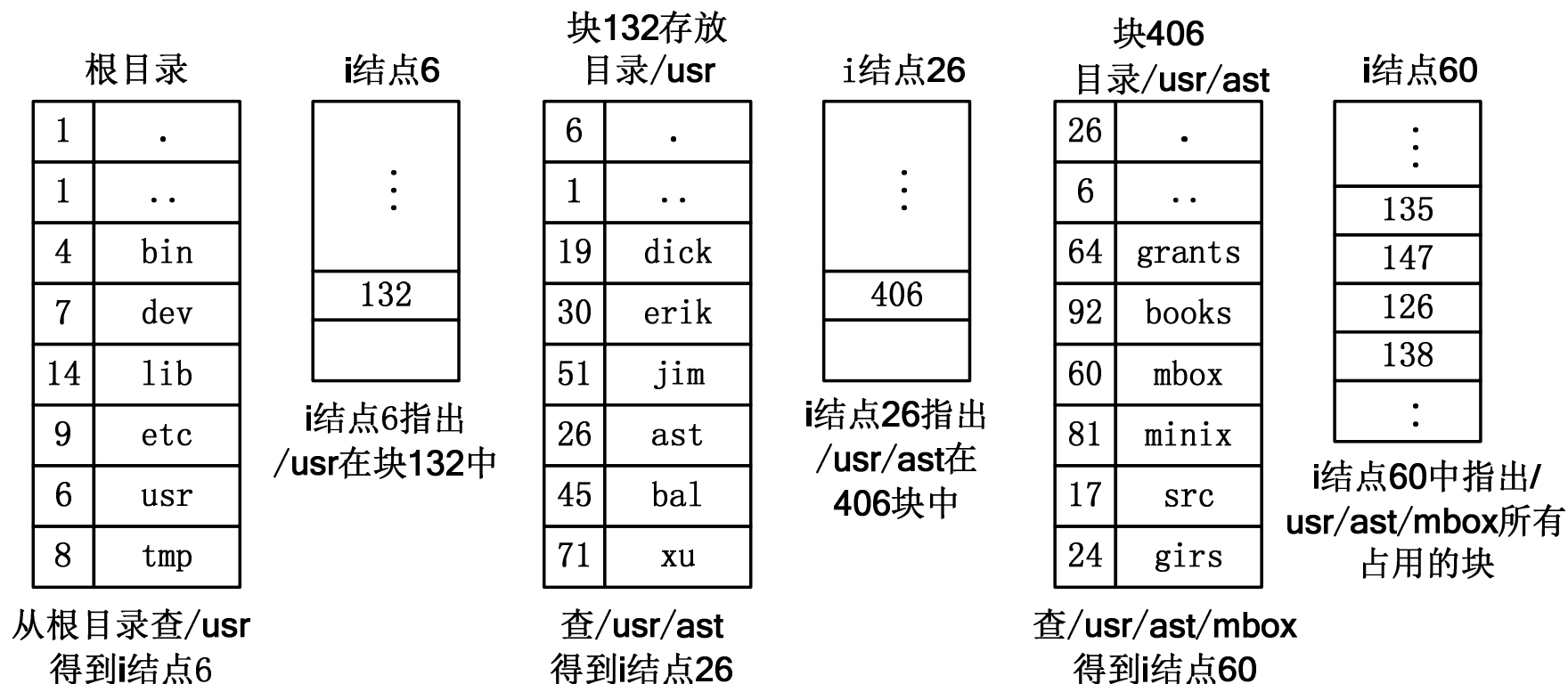


图7-12 查找/`usr/ast/mbox`的步骤



## 7.3.3 目录查询技术

### ❖ 2、Hash检索法

- \* 系统利用用户提供的**文件名**并将它变换为文件目录的**索引值**，再利用该索引值到目录中去查找，提高检索速度。
- \* Hash检索法**不支持模糊匹配**检索。
- \* Hash查找冲突处理规则
  - 1> 如果查找到的目录项为空，表示系统中并无指定文件。
  - 2> 如果找到的目录项中的文件名**与指定文件名相匹配**，**则查找成功**，从该目录项中可找到该文件所在物理地址。
  - 3> 如果找到的目录项中的文件名**与指定文件名不匹配**，**则表示发生了“冲突”**，按一定规则对当前Hash值进行转换生成新的索引值，再返回到第一步重新开始查找。



# 本章主要内容

- ❖ 7.1 文件和文件系统
- ❖ 7.2 文件的逻辑结构
- ❖ 7.3 目录管理
- ❖ 7.4 文件共享
- ❖ 7.5 文件保护





## 7.4 文件共享

### ❖ 1、基本概念

- \* 文件共享是指一个文件可以被多个授权的用户共同使用。
- \* 文件共享需要解决两个问题：一是如何实现共享；二是对各类共享文件的用户进行存取控制（详参7.5节）。

### ❖ 2、实现文件共享的方法

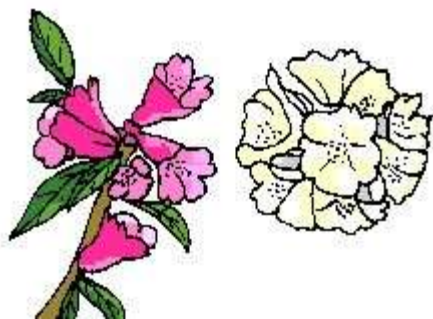
- \* (1) 绕弯路法
- \* (2) 连访法
- \* (3) 利用基本文件目录实现文件共享
- \* (4) 基于索引结点的共享方式（硬链接）
- \* (5) 利用符号链实现文件共享（软链接）

早期实现文件共享的方法



## 7.4 文件共享

- ❖ 7.4.1 基于索引结点的共享方式（硬链接）
- ❖ 7.4.2 利用符号链实现文件共享（软链接）







## 7.4.1 基于索引结点的共享方式

### ❖ 1、基于非索引结点的文件共享

- \* **基本思想**：建立链接时，**将共享文件的物理地址拷贝**。
- \* **缺点**：文件增、改时，其它用户不知，造成新增内容不能共享。

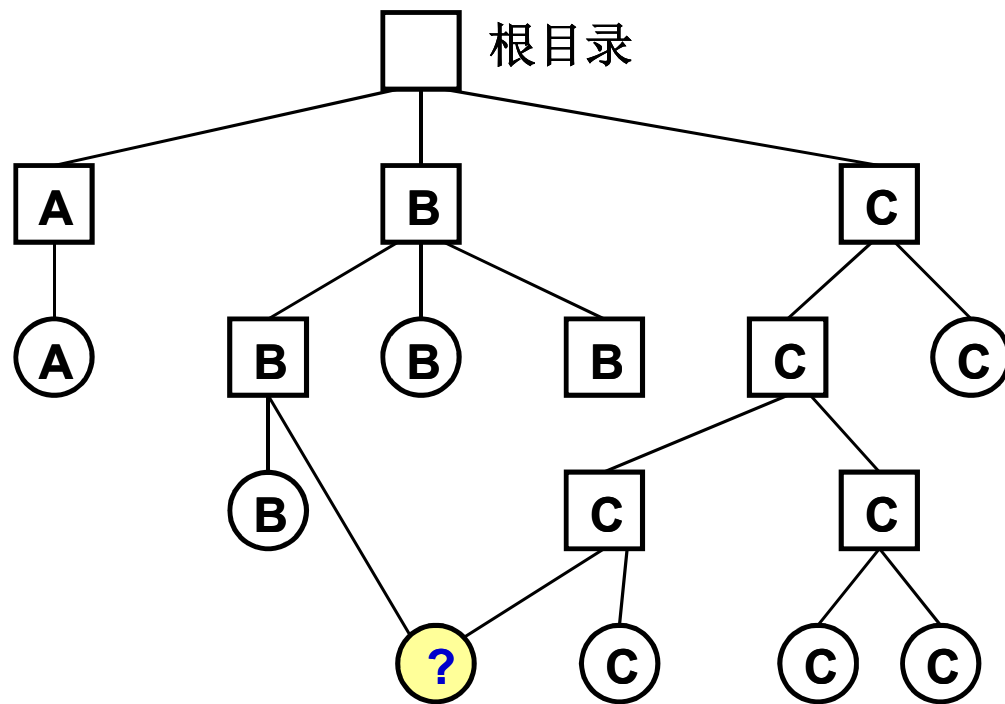


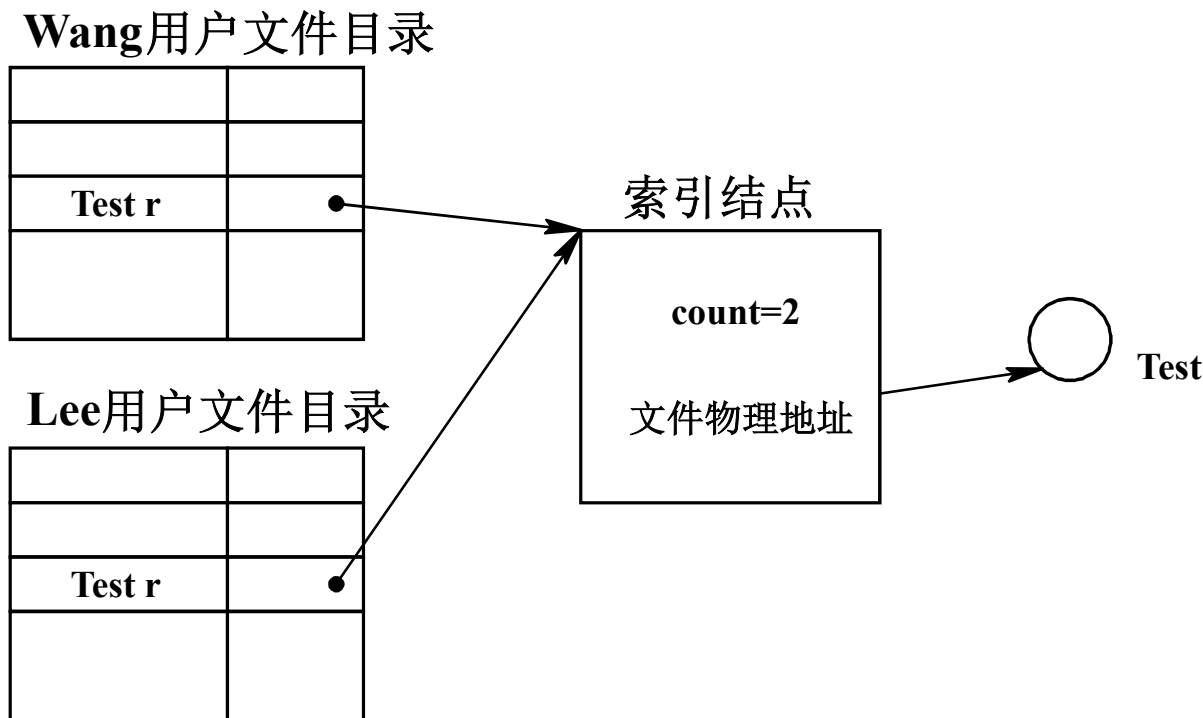
图7-13 包含有共享文件的文件系统



## 7.4.1 基于索引结点的共享方式

### ❖ 2、基于索引结点的文件共享

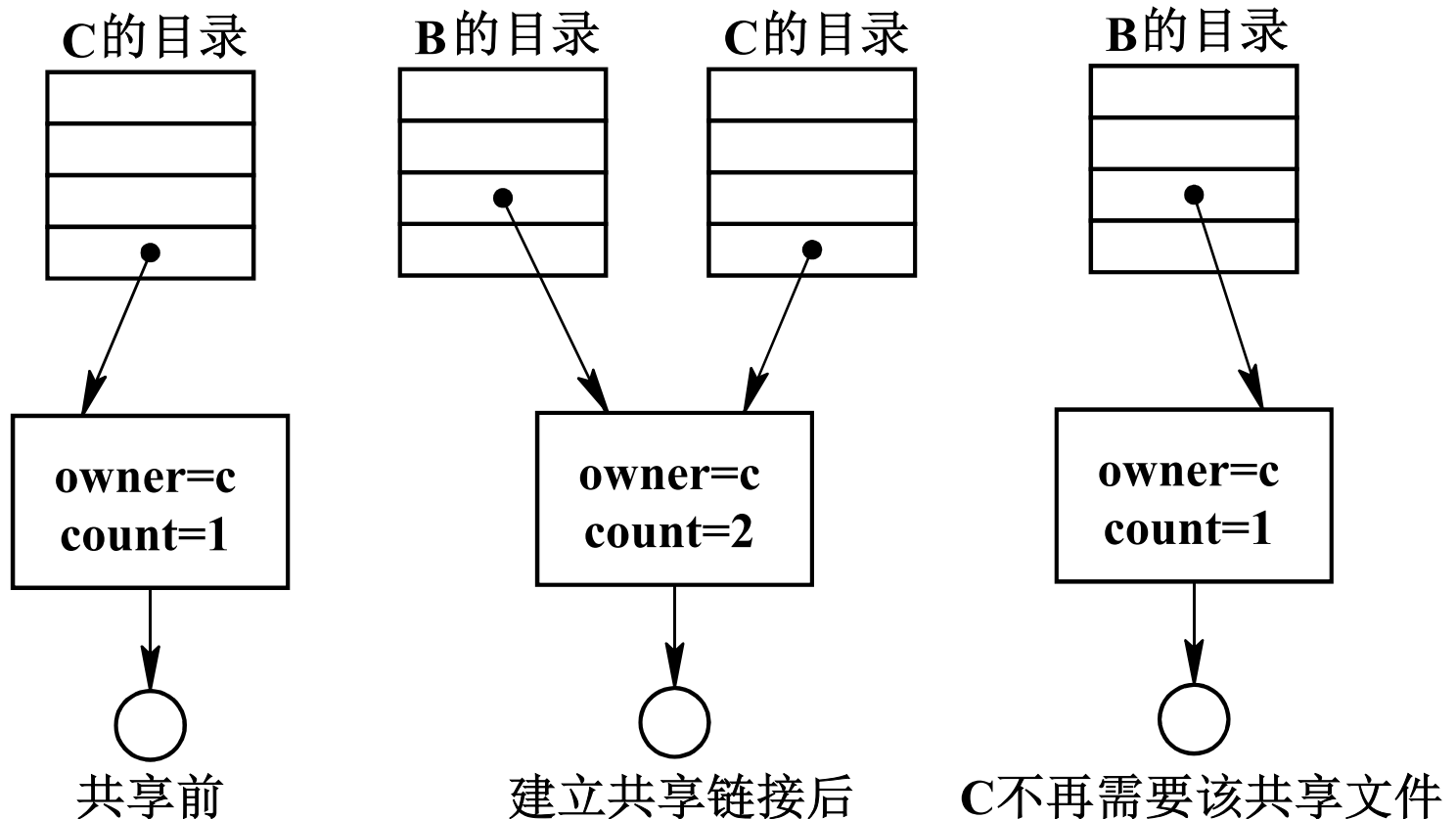
- \* **基本思想**：将诸如文件的物理地址和其它文件属性等信息放在索引结点中，**文件目录中只设置文件名及指向相应索引结点的指针**。





## 7.4.1 基于索引结点的共享方式

- \* **共享文件的删除：**当 $\text{count} > 1$ 时，这时文件主也不能删文件；否则，指针悬空。





## 7.4.1 基于索引结点的共享方式

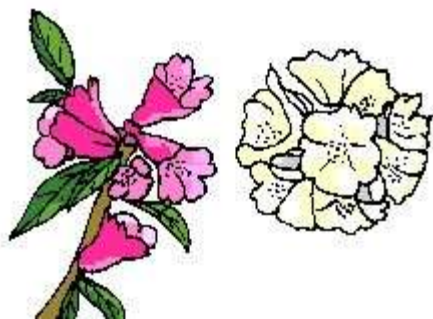
### ❖ 3、补充内容

- \* **记帐**：是指记录、汇总和报告有关系统活动和资源使用等相关信息。
- \* **UNIX常见的记帐**
  - 进程记帐
  - 连接时间记帐
  - 磁盘使用情况记帐
  - 打印机使用情况记帐
  - 费用记帐



## 7.4 文件共享

- ❖ 7.4.1 基于索引结点的共享方式（硬链接）
- ❖ 7.4.2 利用符号链实现文件共享（软链接）





## 7.4.2 利用符号链接实现文件共享

类似windows  
中的快捷方式

### ❖ 基于符号链接的文件共享

- \* **基本思想**：为使用户B能共享用户C的一个文件F，可以由系统创建一个**Link类型的新文件**，也取名为F，在新文件中只包含被链接文件F的路径名，以实现B的目录与文件F的链接。
- \* **优点**：仅原文件指向索引结点，其它链接文件仅包含原文件的路径名，文件主可对原文件删除等；符号链接可用于计算机网络上共享文件。
- \* **缺点**：目录检索开销大，需多次启动磁盘；需为每一共享用户建立一个符号链接**文件**，需要耗费一定的磁盘空间；每一共享文件有多个不同的共享文件名，遍历、转储时会造成重复。



## 7.7.2 利用符号链接实现文件共享

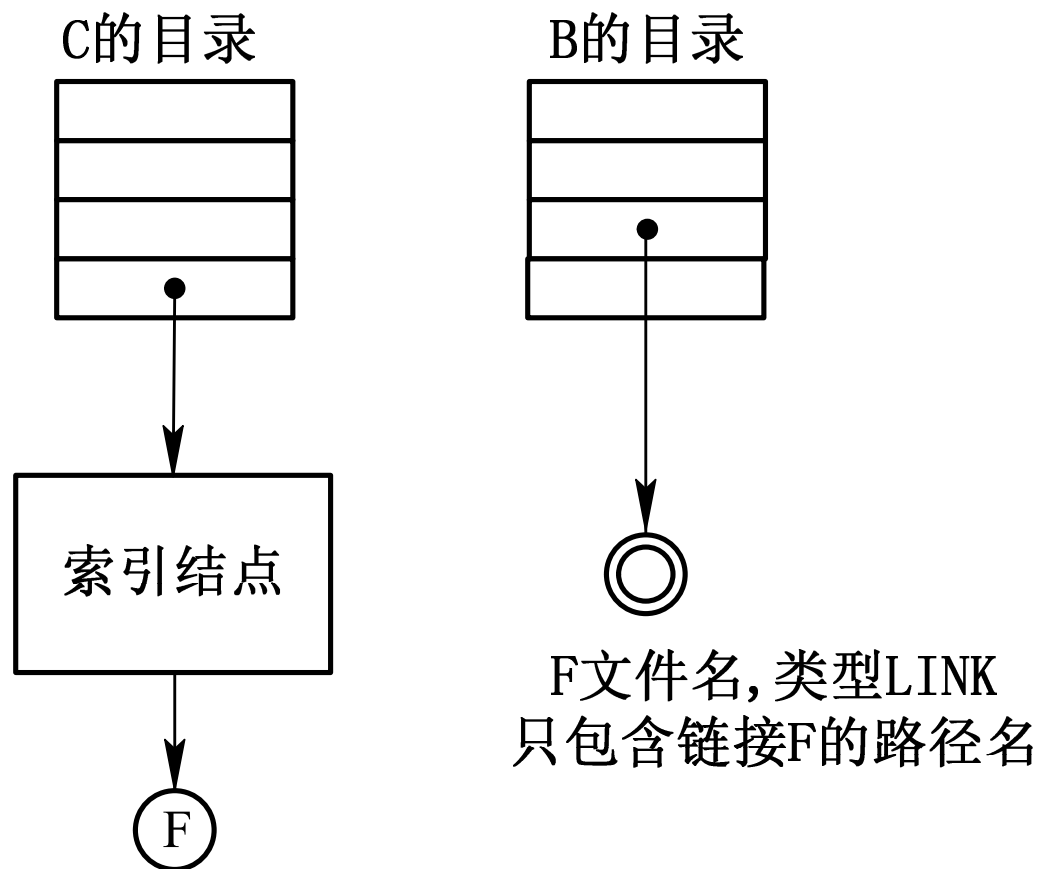


图 利用符号链实现文件共享



# 本章主要内容

- ❖ 7.1 文件和文件系统
- ❖ 7.2 文件的逻辑结构
- ❖ 7.3 目录管理
- ❖ 7.4 文件共享
- ❖ 7.5 文件保护







## 7.5 文件保护

### ❖ 文件保护

- \* 1> 通过存取控制机制来防止由人为因素所造成的文件不安全性。
  - 7.5.1 保护域
  - 7.5.2 访问矩阵
  - 7.5.3 访问矩阵的修改
  - 7.5.4 访问矩阵的实现
- \* 2> 通过磁盘容错技术来防止由系统因素特别是磁盘故障所造成的文件不安全性。
- \* 3> 通过“后备系统”来防止由自然因素所造成的不安全性。



## 7.5.1 保护域

### ❖ 保护域

- \* **访问权**：是指一个进程对**某一对象**进行操作的权力，访问权通常由资源的拥有者或管理者决定。
- \* **访问权的表示**：(对象名,权集)，例如(F1,{R,W})
- \* **保护域**：一个**进程或用户**对**一组对象**的访问权集合。
- \* **进程与保护域间的静态联系方式（一对一）**
  - 进程在整个运行过程中受限于同一个保护域，其可用资源是固定的。
- \* **进程与保护域间的动态联系方式（一对多）**
  - 进程在不同运行阶段使用不同的域。



## 7.5.1 保护域

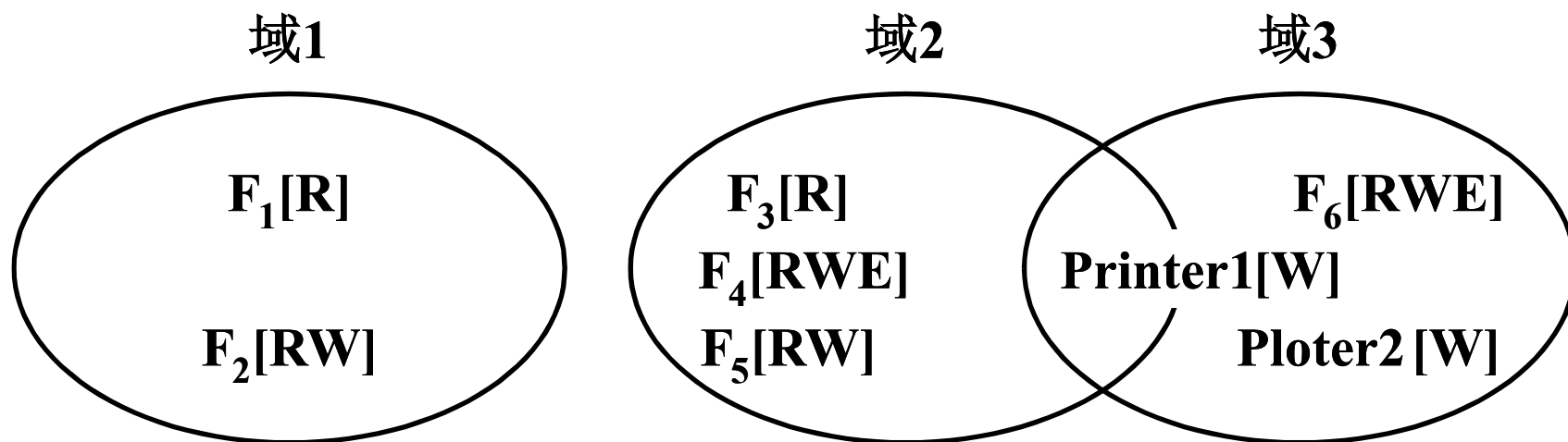


图7-17 三个保护域



## 7.5.2 访问矩阵

### ❖ 1、基本访问矩阵

- \* 访问矩阵是指描述系统访问控制的矩阵，其行代表一个域，列代表一个对象，矩阵中的每一项由一组访问权组成。

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>	打印机 1	绘图仪 2
D <sub>1</sub>	R	R, W						
D <sub>2</sub>			R	R, W, E	R, W		W	
D <sub>3</sub>						R, W, E	W	W

注：R-读，W-写，E-执行

图7-18 访问矩阵示例



## 7.5.2 访问矩阵

### ❖ 2、具有域切换权的访问矩阵

- \* 在访问矩阵里添加进表示域切换的权项。
- \* 域切换权的表示：用S表示。

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>	打印机 1	绘图仪 2	域D <sub>1</sub>	域D <sub>2</sub>	域D <sub>3</sub>
D <sub>1</sub>	R	R, W								S	
D <sub>2</sub>			R	R, W, E	R, W		W				S
D <sub>3</sub>						R, W, E	W	W			

注：R-读，W-写，E-执行，S-切换

图7-19 具有切换权的访问控制矩阵



## 7.5.3 访问矩阵的修改

### ❖ 1、拷贝权

- \* 拷贝权：是指进程可将某个域中所拥有的访问权拷贝到同一列其它域中的权力。
- \* 拷贝权的表示：在有拷贝权的访问权上加\*表示。
- \* 限制拷贝：具有拷贝权的访问权被拷贝到其它域中后，被拷贝过来的访问权不能再次被拷贝。

## 7.5.3 访问矩阵的修改

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

(a)

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

(b)

图 7-20 具有拷贝权的访问控制矩阵



## 7.5.3 访问矩阵的修改

### ❖ 2、所有权

- \* **所有权**：是指进程拥有在不同域中对某对象的访问权进行**添加和删除**的权力。
- \* **所有权的表示**：用**O**表示。



## 7.5.3 访问矩阵的修改

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		

(a)

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

(b)

图7-21 具有所有权的访问矩阵



## 7.5.3 访问矩阵的修改

### \* 3> 控制权

- **控制权**：是指进程拥有对某一域中不同对象的访问权进行**添加和删除**的权力。
- **控制权的表示**：用**Control**表示。
- **控制权与所有权的区别**：
  - **控制权**是用来改变访问矩阵内**同一行**中的各项访问权
  - **所有权**是用来改变访问矩阵内**同一列**中的各域访问权



## 7.5.3 访问矩阵的修改

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>	打印机 1	绘图仪 2	域D <sub>1</sub>	域D <sub>2</sub>	域D <sub>3</sub>
D <sub>1</sub>	R	R, W									
D <sub>2</sub>			R	R, W, E	R, W		W				Control
D <sub>3</sub>						R, W, E	W	W			

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>	打印机 1	绘图仪 2	域D <sub>1</sub>	域D <sub>2</sub>	域D <sub>3</sub>
D <sub>1</sub>	R	R, W									
D <sub>2</sub>			R	R, W, E	R, W		W				Control
D <sub>3</sub>						R, W	W				

图7-22 具有控制权的访问矩阵



## 7.5.4 访问矩阵的实现

### ❖ 访问矩阵的实现

- \* 问题：访问矩阵是稀疏矩阵，存储及访问开销大。
- \* 1> 访问控制表
  - 将访问矩阵按列（对象）划分，为每一列（每个对象）建立一张访问控制表，并将控制表中的空项删除。
- \* 2> 访问权限表
  - 将访问矩阵按行（域）划分，为每一行（每个域）建立一张访问权限表，并将权限表中的空项删除。



# 本章小结

**\*\*掌握  
\*理解**

- ❖ 文件与文件系统的概念\*
- ❖ 文件的逻辑结构\*\*
  - \* 四类有结构文件：顺序、索引、索引顺序、直接
- ❖ 目录管理\*\*
  - \* 三种目录结构：单级、两级、多级
  - \* 两种目录查询：线性检索、Hash法
- ❖ 文件共享与保护\*\*
  - \* 两种共享方式：硬链接(索引结点)、软链接(符号链接)
  - \* 三种保护措施：访问控制、磁盘容错、后备系统(第8章)
- ❖ 重要概念：**FCB**、**i结点**、有结构文件、访问权、保护域



# 本章作业

## ❖ 作业内容:

- \* 操作系统第7章网络在线测试 (12月2日前完成)



本章课程结束！ 谢谢大家！