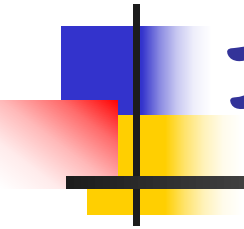


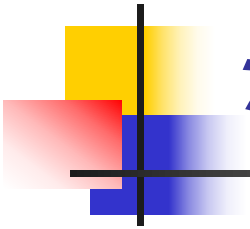
第7章 文件管理

第7章 文件管理

- 在计算机系统中要用到大量的**程序和数据**。由于内存容量有限且不能长期保存这些数据，一般情况下把它们以**文件**的形式存放在**外存**中，待需要的时候才把它们调入内存。
- 为了便于对文件进行管理，一般的操作系统都有文件管理的功能。
- **文件**和**文件系统**是操作系统的重要组成部分。

第7章 文件管理

- 7.1 文件系统
- 7.2 文件的逻辑结构
- 7.3 文件的物理结构
- 7.4 文件目录
- 7.5 文件操作
- 7.6 文件共享



7.1 文件系统

- 7.1.1 文件与文件系统的概念
- 7.1.2 文件的分类、属性及文件系统的功能

7.1 文件系统

7.1.1 文件与文件系统的概念

1.文件的定义

文件是具有标识符（文件名）的一组相关信息的集合。标识符是用来标识文件的，不同的系统对标识符的规定有所不同（dos,windows普遍采用的8+3格式，后来扩展成长文件名，Unix则大大弱化了文件扩展名）

7.1 文件系统

7.1.1 文件与文件系统的概念

2.文件系统的定义

文件系统是操作系统中负责存取和管理文件信息的机构。

它由管理文件所需要的数据结构（如文件控制块、存储分配表等）、相应的管理软件、访问文件的一组操作组成。

2. 文件系统模型

文件　　存放文件的辅存　　对文件进行组织的文件目录

7.1 文件系统

7.1.2 文件的分类、属性及文件系统的功能

1.文件的分类

为了便于管理和控制文件，将文件分为若干种类，下面是常见的几种分类方法：

- **按用途分类：**

（1）**系统文件。**是由系统软件构成的文件，对用户不直接开放，只允许用户调用

（2）**用户文件。**用户委托系统保存的文件。如源代码、目标程序等。

（3）**库文件。**由标准子程序和常用的应用程序组成的文件。只允许调用，不许用户修改。

7.1 文件系统

7.1.2 文件的分类、属性及文件系统的功能

- 按文件中的数据形式分类：

（1）源文件。它是由原程序和数据构成的文件。从终端输入或输出，一般由ASCII代码或汉字组成。

（2）目标文件。是由相应的编译程序编译而成的文件，由二进制码组成，扩展名为.obj。

（3）可执行文件。是由目标程序链接成的文件。文件的扩展名一般为.exe。

7.1 文件系统

7.1.2 文件的分类、属性及文件系统的功能

- **按操作保护分类**

（1）只读文件。仅允许对其进行读的文件。

（2）读写文件。允许用户对其进行读或写操作的文件。

（3）执行文件。允许用户调用执行，但不允许读，也不允许写的操作。

（4）什么也不许做的文件。

7.1 文件系统

7.1.2 文件的分类、属性及文件系统的功能

- **按文件的性质分类**

有些系统，如UNIX系统把文件分为三类：

（1）普通文件。一般的系统文件及用户文件。

（2）目录文件。由文件目录组成的文件。

（3）特殊文件。将设备看作特殊文件。

7.1 文件系统

7.1.2 文件的分类、属性及文件系统的功能

2.文件的属性

通过文件的属性，可进一步认识它，文件的属性可包括：

（1）文件类型。可从不同的角度确定它的类型

（2）文件长度。文件的当前长度

（3）文件的位置（文件物理地址）

（4）文件的读取控制。文件的读、写、执行等控制

（5）文件的建立时间、最后修改时间等

7.1 文件系统

7.1.2 文件的分类、属性及文件系统的功能

3.文件系统的功能

一个文件系统应具有以下功能：

（1）用户可执行创建、修改、删除、读写文件的命令

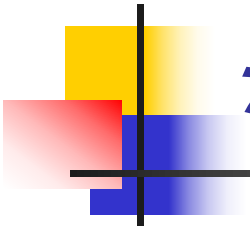
（2）用户能以合适的方式构造其它的文件

（3）用户能在系统的控制下，共享其它用户的文件

（4）允许用户用文件名访问文件

（5）系统应有转存和恢复文件的能力，以防意外情况的发生

（6）系统应提供可靠保护及保密措施



7.2 文件的结构

- 7.2.1 逻辑结构
- 7.2.2 存取方法

7.2 文件的逻辑结构

- 人们常从两种不同的角度研究文件的结构：

（1）从用户的角度。主要研究观察到的文件组成形式，用户可以直接处理其中的结构和数据，常称之为逻辑结构；

（2）从实现的观点。主要研究存储介质上的实际文件结构，是指文件在外存上的存储组织形式，常称为物理结构或存储结构。

7.2.1 逻辑结构

- 文件的逻辑结构分为以下两类：

1.有结构的文件

- 有结构的文件是指由若干个相关的记录构成的文件，又称**记录式文件**。
- 在文件中的记录一般有着相同或不同数目的数据项，根据记录的长度，记录式文件又分为等长记录文件和变长记录文件。

7.2.1 逻辑结构

- 下面是一个典型的记录式文件中的内容：

090601010 罗小宁 女 1988年5月 江苏连云港

090601011 王朔 男 1989年8月 江苏南京

…… …… …… …… ……

7.2.1 逻辑结构

2.无结构文件

又称流式文件，组成它的基本信息单位是字节或字，其长度是文件中所含字节的数目。如大量的源程序、库函数等采用的就是流式文件。

7.2.2 存取方法

- 对文件的逻辑结构的存取方法有两种：

（1）顺序存取。最简单的一种存取方法。严格按文件信息单位排列的顺序依次存取，后一次的存取总是在前一次存取的基础上进行，所以不必给出具体的存取位置

（2）随机存取。又称直接存取，在存取时必须先确定进行存取的起始位置（如记录号等）

7.2.2 存取方法

- 在对实际设备上的文件内进行存取时可采取不同的方式。
- 如磁带一般只采用顺序存取方式，
- 而对于磁盘、磁鼓上面的文件，既可采取顺序存取方式，也可采取随机存取方式。

7.2.2 存取方法

- 用户所看到的是逻辑文件，处理的是逻辑记录，是按照逻辑文件的形式去存储、检索和组织文件信息的。
- 但无论如何，这种逻辑上的文件总要以某种方式保存到存储介质上。
- 此时的文件称为物理文件。

7.3 文件的物理结构（与存储介质密切相关）

- 7.3.1 连续结构（顺序结构）
- 7.3.2 链接结构
- 7.3.3 索引结构

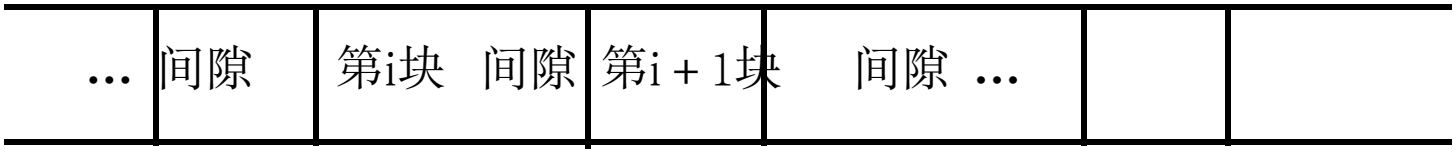
7.3 文件的物理结构

- 逻辑文件在辅存的组织结构称为文件的物理结构。
- 如何组织它们，则主要依赖于文件存储器（磁带、磁盘、光盘等）的物理特性，以及用户对文件的访问方式。

磁带

磁带是一种典型的顺序存取设备。由于磁带机的启动和停止要花费一定的时间，因此在磁带的相邻物理块之间设计有一段间隙将它们隔开，如下所示。

磁带



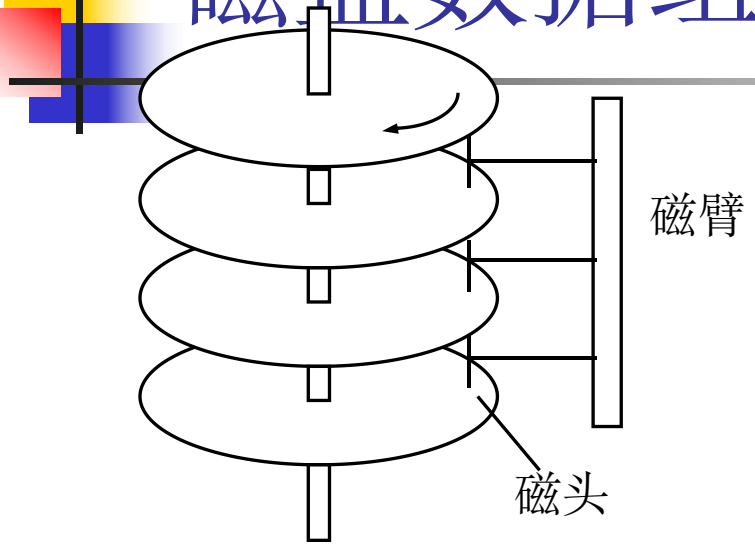
磁带（续）

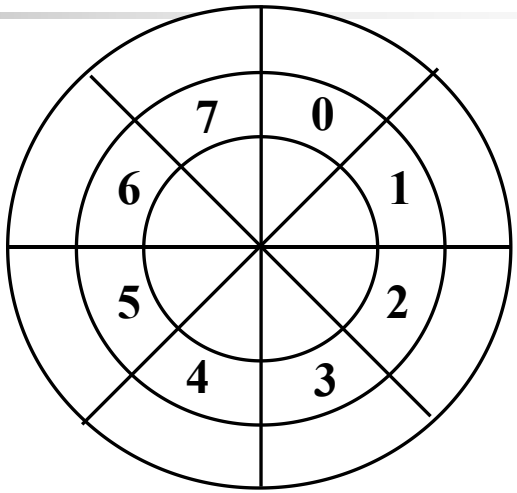
- 磁带的存取速度与信息密度（字符数/英寸）、磁带带速（英寸/秒）和块间间隙有关。
- 如果带速高、信息密度大且所需块间隙（磁头启动和停止时间）小，则磁带存取速度高。反之，若磁带带速低、信息密度小且所需块间隙（磁带启动和停止时间）大，则磁带存取速度低。

磁盘

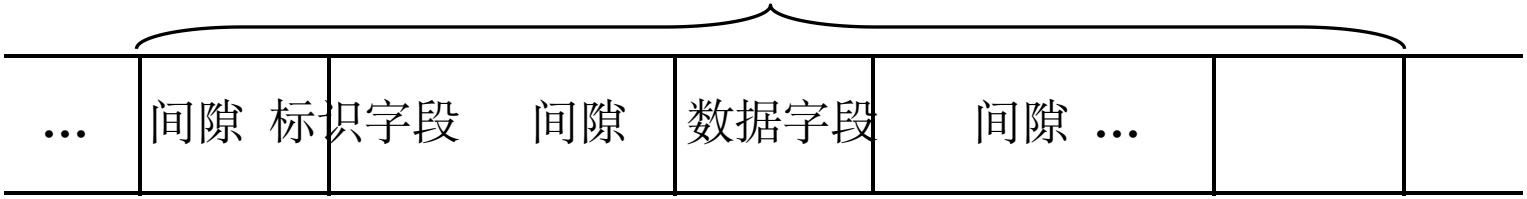
- 磁盘是典型的直接存取设备。
- 磁盘一般由若干磁盘片组成，可沿一个固定方向高速旋转。每个盘面对应一个磁头，磁臂可沿半径方向移动。
- 磁盘上的一系列同心圆称为磁道，磁道沿径向又分成大小相等的多个扇区，与盘片中心有一定距离的所有磁道组成一个柱面。
- 磁盘上的每个物理块可用柱面号，磁头号和扇区号表示。

磁盘数据组织和格式示意图





磁道 第i扇区



磁盘访问时间

- 磁盘访问时间由三部分组成：
  - 寻道时间：指将磁头从当前位置移动到指定磁道所经历的时间。由启动磁臂时间和磁头移动多条磁道的时间构成。
  - 旋转延迟时间：指扇区移动到磁头下面所经历的时间。平均旋转延迟时间是每转所需时间的一半。
  - 传输时间：指从磁盘上读出数据或向磁盘写入数据所经历的时间。
- 由于这三部分操作均涉及机械运动，故磁盘块的访问时间约为0.01～0.1s之间，其中寻道时间所占的比例最大。

存储设备、存取方法和物理结构间的关系

- 文件的物理结构与文件存储器的特性和存取方法密切相关。
- 磁带是一种顺序存取设备，适合采用顺序结构存放文件，相应的存取方法通常是顺序存取法。若采用其他文件结构或采用直接存取方式都不太合适。

7.3 文件的物理结构

- 在文件系统中，文件的存储设备通常划分为若干大小相等的物理块，每块大小为512B或1024B。
- 与此相对应，一般把文件信息也划分为与存储设备的物理块大小相等的逻辑块。
- 此时，块便成为分配和传输信息的基本单位。

7.3 文件的物理结构

- 一般来说，文件的物理结构有以下三种：

连续结构 → （辅存连续分配）

链接结构、

索引结构。 → 辅存的离散分配

7.3 文件的物理结构

7.3.1 连续结构（顺序结构）

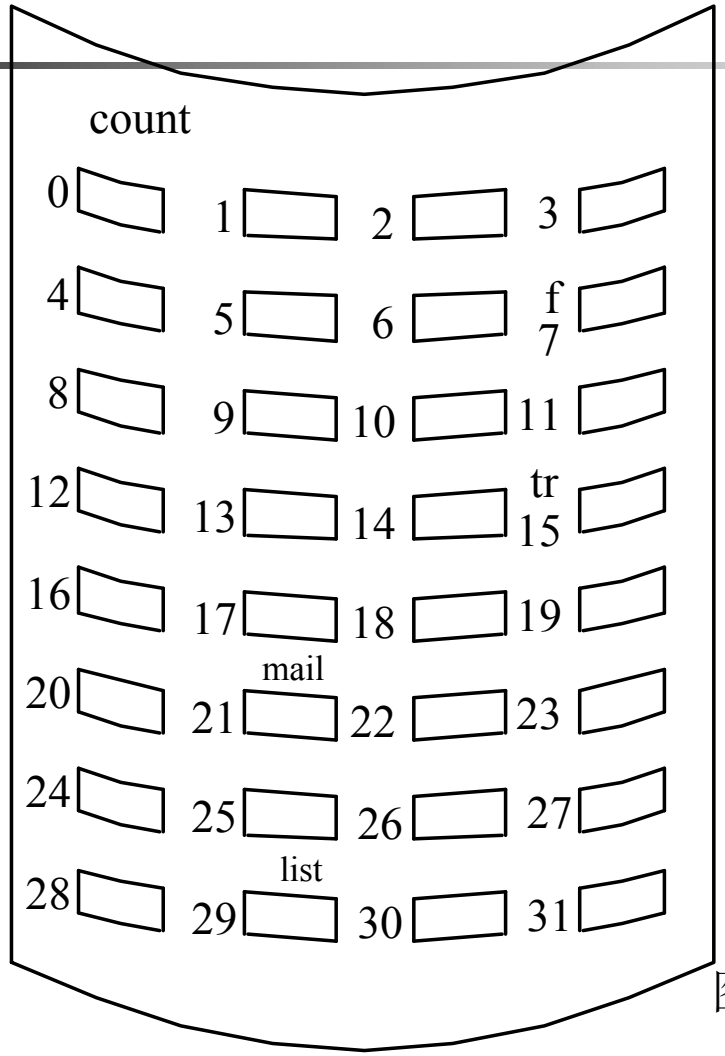
- 一个逻辑文件信息一次存放在外存的若干连续物理块中，这样的文件称为连续文件。
- 这种结构的文件，其文件在磁盘上的存放顺序与用户看到的逻辑记录是一致的
- 连续文件可采用顺序存取，也可以随机存取，就看你采用什么样的存储介质来存储文件。

7.3 文件的物理结构

7.3.1 连续结构（顺序结构）

- 物理结构为顺序结构的文件，若存放在顺序存储介质（如磁带）上，则适宜于顺序存取；
- 若存放于随机存储介质（如磁盘、磁鼓）上，则文件可顺序存取，也可以随机存取。

连续文件



目录

file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

图 6-7 磁盘空间的连续分配

7.3 文件的物理结构

7.3.1 连续结构（顺序结构）

连续文件的优点是：

- 知道文件在存储设备上的起始地址和文件长度就能快速存取；
- 顺序访问速度最快

缺点是

- 文件长度一经确定后不易改变，不利于文件的扩充和增生，
- 文件大小不一，每个文件都要求连续辅存。辅存中会产生碎片

7.3 文件的物理结构

7.3.1 连续结构（顺序结构）

- 所以物理结构为顺序结构的文件，只适宜于存放不修改的定长文件，如系统文件等。
- 因此又引入了文件的链接结构

7.3 文件的物理结构

7.3.2 链接结构

- 链接结构文件也叫串联文件。
- 它是把一个逻辑上连续的文件离散地存放到不连续的物理块中。
- 为了表示其对应的逻辑块次序，为各物理块设置一个指针，该指针指向下一个逻辑块所对应的物理块。
- 文件的最后一个物理块，其指针指向NULL。

7.3 文件的物理结构

7.3.2 链接结构

- 这样存放同一文件的物理块就链接成一个串联队列。
- 用户所见到的逻辑文件就被存放到这样一个串联的磁盘块序列中

7.3 文件的物理结构

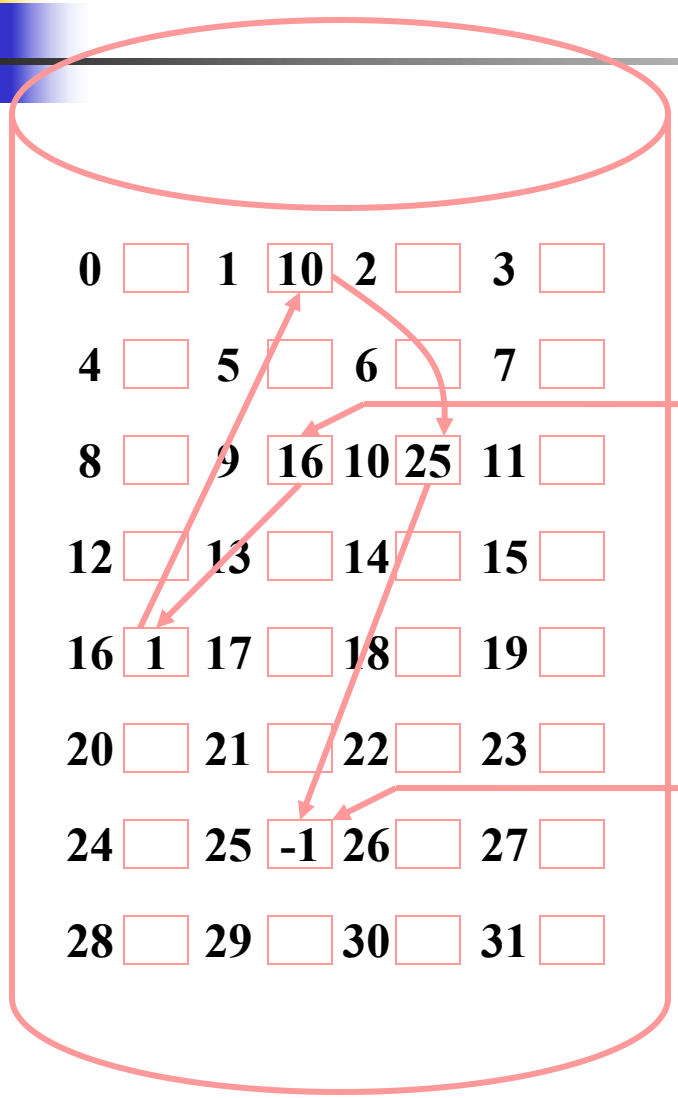
7.3.2 链接结构

- 根据链接方式的不同，链接结构又可分为隐式链接和显式链接两种。
- （1）隐式链接

在每个物理块中设有一个指针，指向其后续链接的物理块，从而使得存放同一文件的物理块链接成一个串联队列。

显然，使用隐式链接结构，不必在文件目录项中指明文件的长度，只需指明文件的起始块号就行了。

隐式链接文件



文件目录

文件名	始址	末址
jeep	9	25

磁盘空间的链接式分配

7.3 文件的物理结构

7.3.2 链接结构

- 由于隐式链接文件结构只能按队列中的链接指针顺序搜索，因此搜索效率低
- 而且由于在每个物理块中都包含一个指向后继的指针，万一某个块中的信息遭到破坏而无法存取时，该数据的其余块也就无法存取，因此隐式链接存在安全上的隐患(链比较脆弱，易断)。

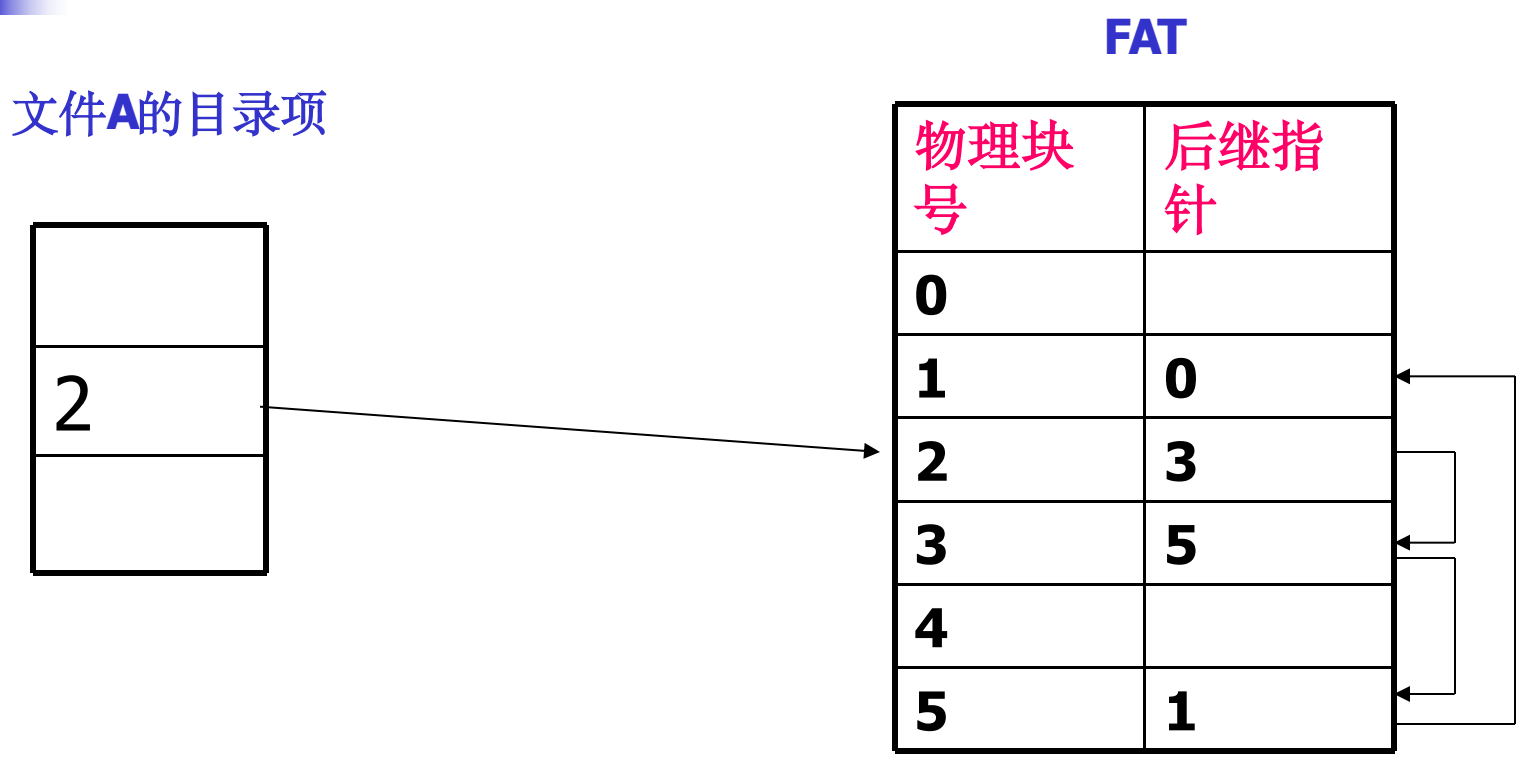
7.3 文件的物理结构

7.3.2 链接结构

- （2） 显式链接

- 为了提高链接文件安全性，提高链接文件的访问速度，还可以采用显式链接。
- 显式链接是指把用于链接文件各物理块的指针，显式地存放在磁盘上的一张链接表中，表中存放了磁盘中所有文件所占用的物理块号，因此该表称为文件分配表FAT。

显式链接文件结构示意图：



7.3 文件的物理结构

7.3.2 链接结构

- FAT中的序号为物理块号，编号从0到最大块号。
- 在该表中，属于某个文件的第一个物理块号，也就是每一条链的链首指针所指示的物理块号作为文件的地址被填入文件目录项的“物理地址”字段中。

7.3 文件的物理结构

7.3.2 链接结构

- 当存取文件的时候，系统会把FAT读入内存，这样查找记录的过程是在内存中进行的，因此不仅提高了检索速度，还大大减少了访问磁盘的次数。
- 在MS-DOS，Windows、OS／2中都采用了FAT。

7.3 文件的物理结构

7.3.2 链接结构

- 链接的优点是不需要指明文件的长度，只需指明文件的第一个块号即可，**易于扩展**；
- 且文件的逻辑记录可存放到不连续的物理块中，能较好地利用外存空间；**消除辅存碎片**

7.3 文件的物理结构

7.3.2 链接结构

- 还易于对文件进行扩充：只要调整链接指针就可以在任何两个数据块之间删除或插入一个信息块。
- 所以适宜于存放变长文件

7.3 文件的物理结构

7.3.2 链接结构

- 链接结构的缺点是：只能按照队列中的指针顺序搜索，效率较低；
- 且其存取的方法只能顺序存取，不能随机存取。
- 为解决以上问题，又提出了文件的索引结构

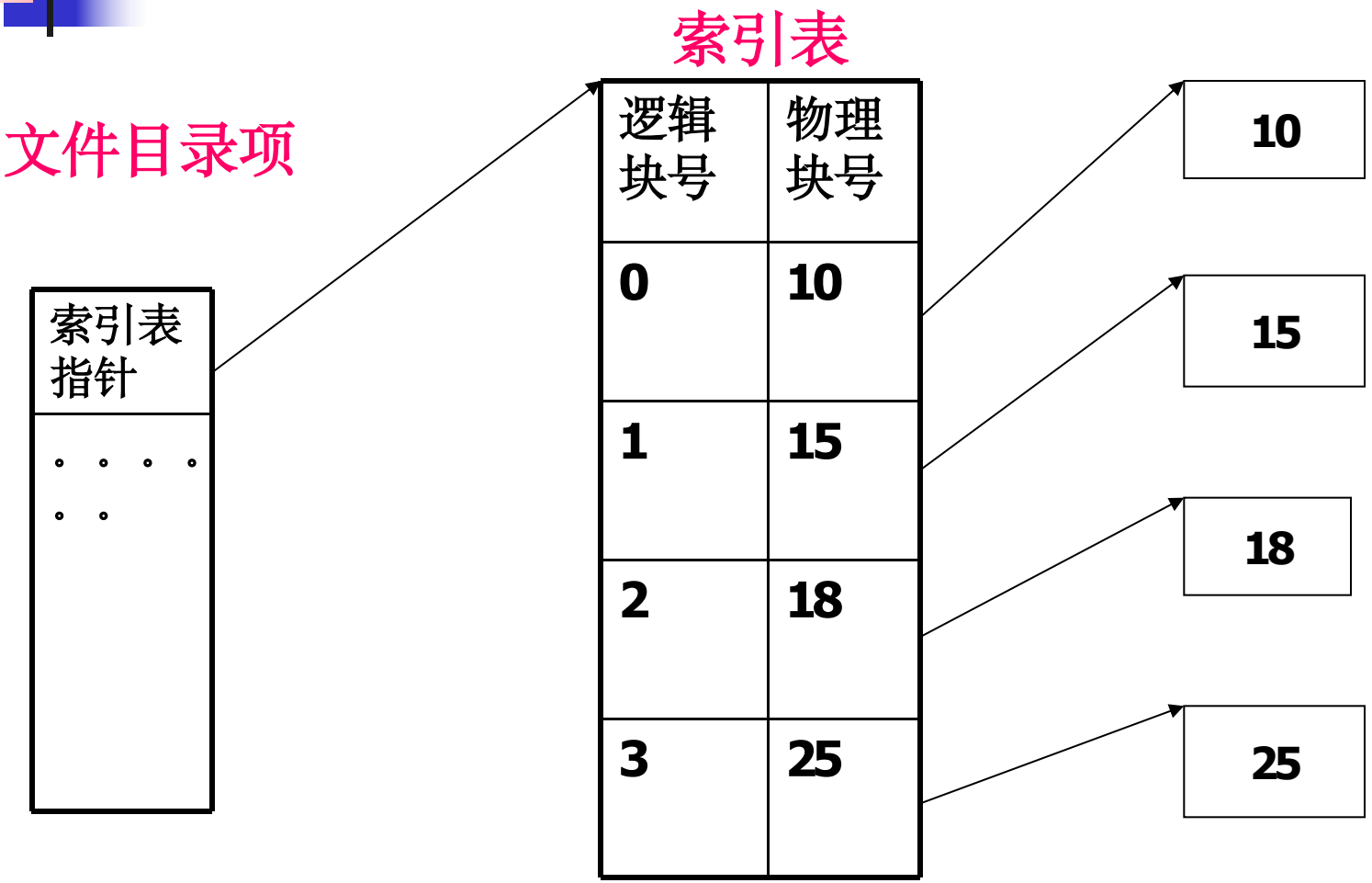
7.3 文件的物理结构

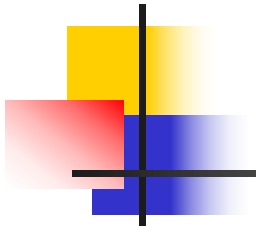
7.3.3 索引结构

- 索引结构要求系统为每一个文件建立一张索引表（与文件中的数据分开存放），表中每个栏目指出文件信息所在的逻辑块号和物理块号之间的对应关系
- 索引表本身的物理地址由文件说明信息(FCB)给出。

7.3 文件的物理结构

7.3.3 索引结构





- 索引文件结构既可以满足文件动态增长的需要，又可以比较方便、迅速地实现随机存取，因为逻辑块号和物理块的对应关系全在一个索引表中，而不是象链接文件那样分散在各个物理块中。

7.3 文件的物理结构

7.3.3 索引结构

- 即当文件的物理结构为索引结构时，即可顺序存取，也可随机存取。

7.3 文件的物理结构

7.3.3 索引结构

- 但是当文件很大时，其索引表也会很大。
- 如果索引表的大小超过了一个物理块，那么必须象处理其它文件的存放那样决定索引表的物理存放方式（二级、多级索引）。

二级索引分配

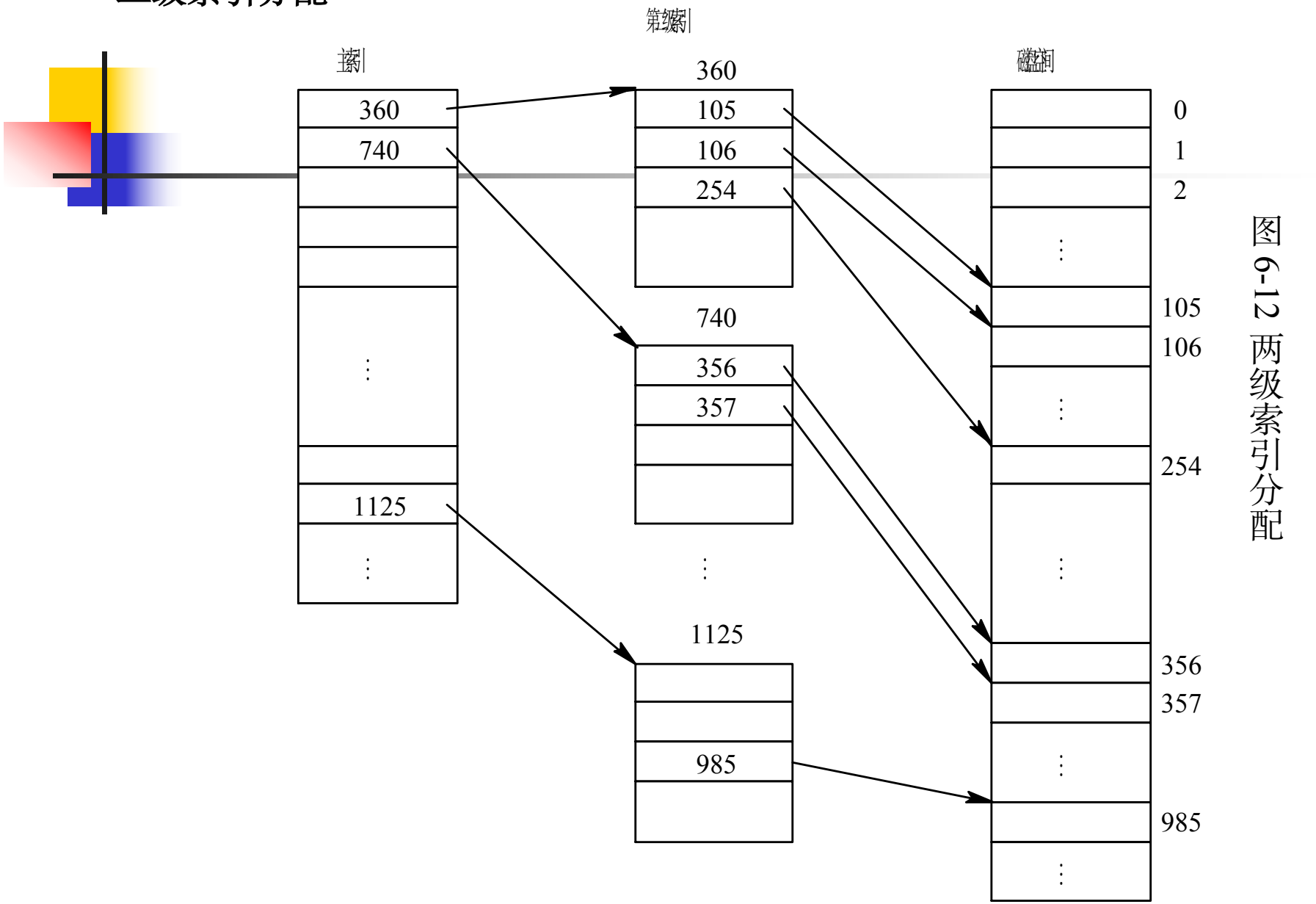


图 6-12 两级索引分配

7.3 文件的物理结构

7.3.3 索引结构

- 一种较好的方法是采用多重索引（间接索引）结构，即为索引表再建立索引。也就是说，在索引表所指的物理块中存放的不是文件内容，而是装有文件信息的物理块号。

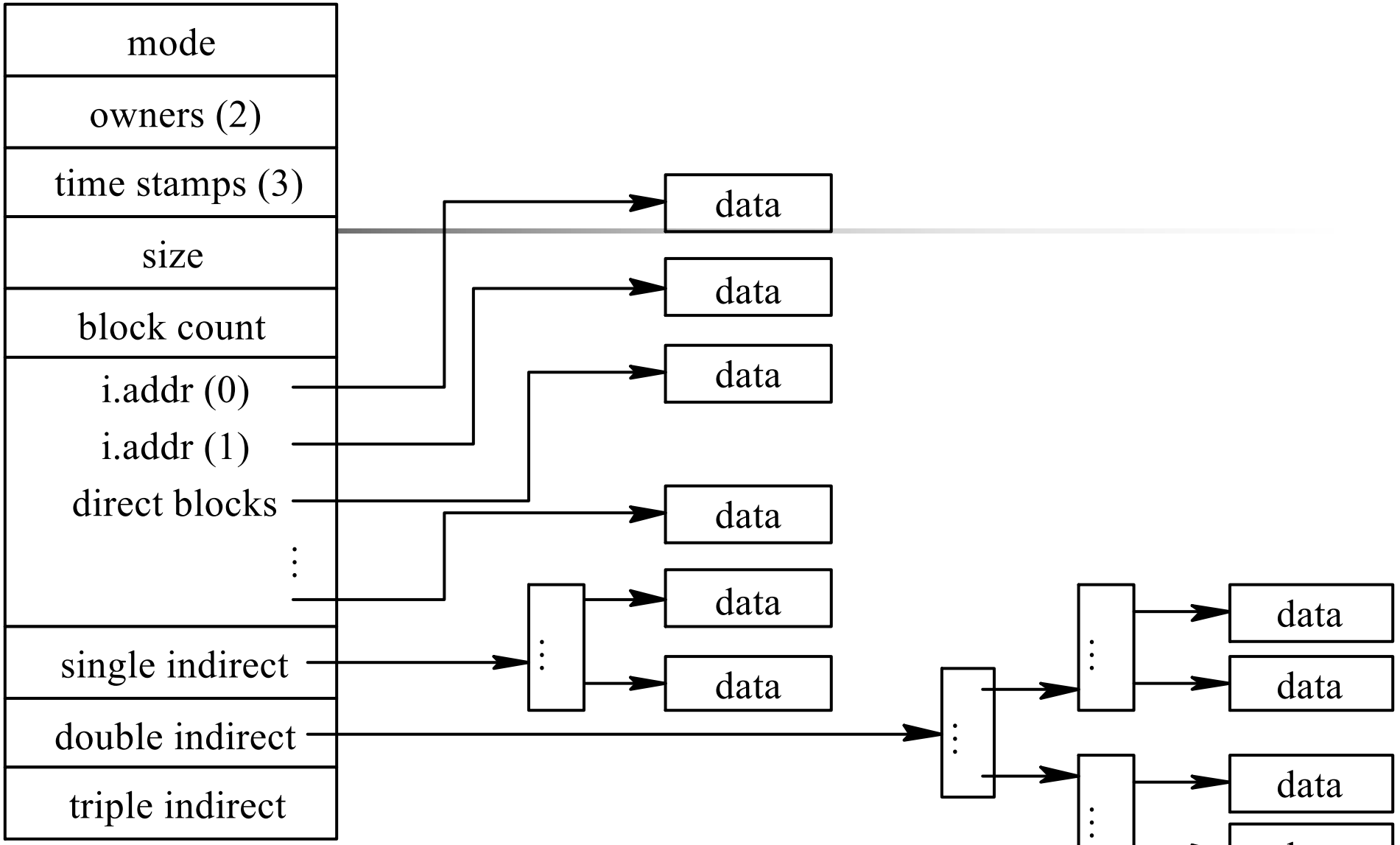
多级索引性能分析:

多级索引比二级索引（二级比单级索引）能够表示的文件更大（包含的物理块更多），但是随着索引级数的增加，文件的访问速度变慢。

注意：在采用多级索引的文件系统中，所有文件的索引块采用相同的结构。例如在三级索引文件中，即使是只有一个块的文件，也采用三级索引。

如何在文件尺寸和访问速度之间进行更好的平衡?

有没有可能小文件采用一级索引，大文件采用多级索引?



unix的混合索引方式

7.4 目录管理

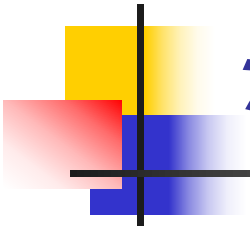
对目录管理的要求如下：

(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。



7.4 目录管理

- 7.4.1 文件控制块和文件目录
- 7.4.2 索引节点
- 7.4.3 目录结构

7.4 文件目录

- 在计算机系统中通常都要存储大量的文件。
- 为了对这些文件进行有效管理，需要将它们进行妥善组织，目前主要是通过文件目录来实现。

7.4.1 文件目录项和目录文件

- 为了能对文件进行正确的存取，文件系统为每个文件都建立了一个用于描述和控制文件的数据，并永久性存放在[illegible]目录项。
- 相关的文件的目录项集中[illegible]文件中，这个文件称为目录文件。

相关信息的集合

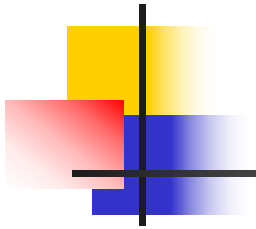
- 文件管理通常借助目录项中的信息，对文件进行各种操作。
- 目录项与文件是一一对应的，是文件存在的唯一标志。

7.4.1 文件目录项和文件目录

一个典型的文件目录项应包括如下信息：

（1）基本信息

- 包括文件名（逻辑名）、文件类型、
- 文件的物理位置（物理名）（如文件所在的设备号、文件在外存上的起始磁盘块号、文件大小等）、
- 文件的逻辑结构、文件的物理结构等。



（2）存取控制信息

包括文件所有者的存取权限、授权用户的存取权限，以及其他用户的存取权限等。

7.4.1 文件控制块和文件目录

（3）文件使用信息

- 包括文件的建立日期与时间；
- 文件最近修改或访问过的日期和时间；
- 当前的使用信息（如当前以打开文件的进程数、文件的使用状态、文件是否已被修改而尚未保存到磁盘等）。

7.4.1 文件控制块和文件目录

- 文件目录实际上就是文件目录项的有序集合，即把所有文件目录项有机地结合起来，就构成了文件目录。
- 专放文件目录项的文件被称为目录文件。
- 构成目录文件的基本单元通常称为目录项——它是每个文件的FCB的子集。

——————注意目录项和FCB的区别

7.4.1 文件控制块和文件目录

- 目录项和文件控制块在概念上是等同的，只不过当文件处于非活动状态时，常用**目录项**对文件的基本信息进行描述；
- 而当文件进入内存，处于活动状态时，常用文件**控制块**对文件进行管理与控制。

7.4.1 文件控制块和文件目录

引入目录文件后：

（1）可实现对文件的“按名存取”。

- 当用户要求存取某个文件时，系统首先找到目录文件，通过比较文件名就可以找到文件的目录项；
- 然后通过目录项中的文件起始位置，就能一次存取文件信息。

7.4.1 文件控制块和文件目录

（2）能实现对文件的共享。

- 在多用户系统中，应允许多个用户共享一个文件。
- 通过对某个目录的共享，就能实现对文件的共享。

（3）二级目录和多级目录结构能允许文件重名。

使用户能按自己的习惯给文件命名和使用文件。

7.4.2 索引节点
——索引节点的引入

- 通常文件目录是放在磁盘上的，当文件数量很多的时候，文件目录也要占用较多的磁盘块。
- 在查找目录的过程中，先把存放目录文件的第1个磁盘块装入内存，然后把用户要访问的文件名与目录项一一比较；
- 若未找到，则需将下一个磁盘块中的目录项调入内存……

7.4.2 索引节点
——索引节点的引入

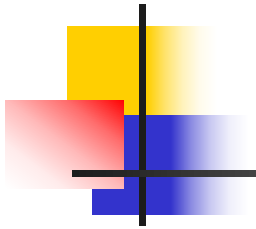
- 假设文件目录所占的磁盘块数为M，按此方法查找，则查找一个目录项平均需要调入磁盘块（M+1）／2次。
- 进一步，假设一个目录项为64B，磁盘块大小为1KB，则一个磁盘块上只能存放16个目录项；

若一个目录文件共有640个目录项，则该目录文件需要占用40个磁盘块。

经计算，平均查找一个文件需要启动磁盘20次。

7.4.2 索引节点
——索引节点的引入

- 经分析又发现，在检索文件目录时，是通过文件名进行搜索的，其它的一些描述信息在检索目录时一概不用。
- 只有在某个目录项中找到与查找的文件名相匹配的时候，才从该目录项中读出文件的物理地址。



- 因此，在目录检索时，除了文件名，目录项中的其它描述信息根本没有必要装入内存。

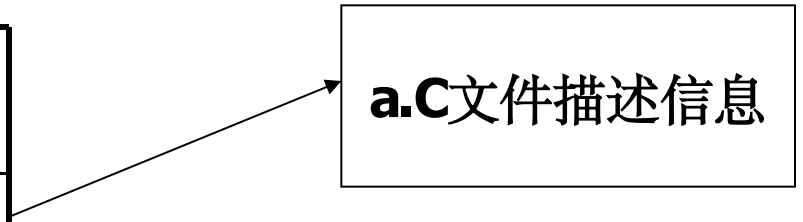
7.4.2 索引节点

——索引节点的引入

- 因此在有些系统中，如*Unix*／*Linux*，采用了文件名和文件描述信息分开存储的方法，使文件的描述信息单独形成一个数据结构。
- 该数据结构称为索引节点，简称*i*节点。
- 在文件的目录项中，只包含两个内容：文件名和指向*i*节点的指针。

具有i节点的文件目录结构：

文件名	索引节点（i节点）指针
a.c	19
b.txt	33
c.exe	20
……	45



7.4.2 索引节点
——索引节点的引入

索引节点又分为磁盘i节点和内存i节点两种。

磁盘i节点是指存放在磁盘上的i节点，其主要内容包括：

- 文件所有者标识符、
- 文件类型、存取权限、
- 文件物理地址、文件长度、
- 文件链接计数及文件存取时间等.

7.4.2 索引节点
——索引节点的引入

- 内存i节点存放在内存中的i节点，当打开文件的时候，需要将磁盘i节点上的内容复制到内存i节点，便于以后使用。
- 除了包含磁盘i节点的所有栏目外，内存i节点中还包括索引节点号、状态、访问计数、文件所属的文件系统逻辑设备号以及链接指针等。

7.4.3 目录结构

- 目录结构的组织，关系到文件系统的存取速度，也关系到文件的共享和安全。
- 因此，组织好文件的目录结构，是提高文件系统性能的重要环节。
- 常用的目录结构有单级目录（一级目录）、二级目录和多级目录（也叫树形目录）三种。

7.4.3 目录结构

1.单级目录结构

- 单级目录结构是最简单的目录结构，在整个文件系统中仅设置一个文件目录，每个文件占用一个目录项。
- 每个目录项包含文件名、扩展名、文件类型和长度、文件的物理地址及其它属性。

单级目录结构如表所示：

文件名	物理地址	文件描述信息	。。。
File1	…….	……	
File2	……	……	
…	……	……	。。。

7.4.3 目录结构

1.单级目录结构

- 单级目录结构的优点是简单，管理方便能实现文件系统的基本功能——按名存取。
- 但存在一些缺点：

（1） 无法解决重名问题

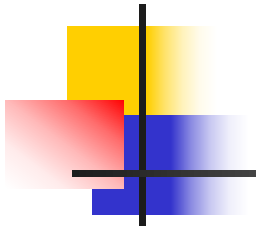
（2） 难以实现文件共享

（3） 查找速度慢

7.4.3 目录结构

2.二级目录结构

- 在单级目录中，要求文件名和文件之间有一一对应的关系，不允许多个文件具有相同的文件名。
- 但在多道系统中，不同用户对不同文件取相同的名字已经很普遍，单级目录显然不能满足该要求。



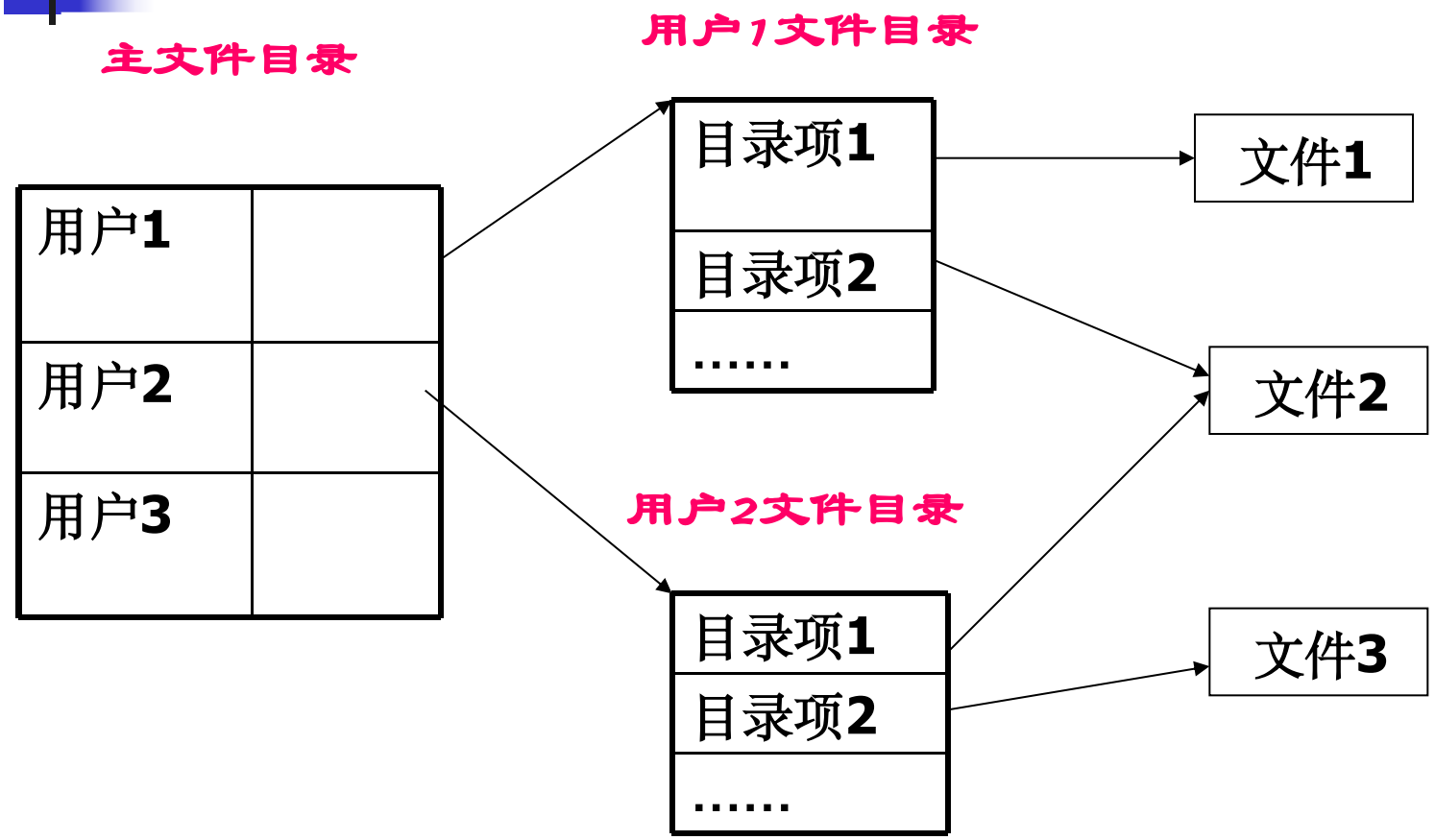
- 为此文件系统应具有更加灵活的命名能力。
- 二级目录就是为了实现文件能重名而提出的。

7.4.3 目录结构

2.二级目录结构

- 在二级目录中，第一级为主文件目录（MFD），它用于管理所有用户文件目录，它的目录项登记了每个用户名及其文件目录的地址。
- 第二级为用户文件目录（UFD），它为该用户的每个文件设置一个目录项，用户只允许查看他自己的目录项。

二级目录结构如下表：


主文件目录
用户1文件目录
用户1
用户2
用户3
目录项1
目录项2
……
文件1
文件2
用户2文件目录
目录项1
目录项2
……
文件3

7.4.3 目录结构

3.多级目录结构

- 多级目录（树型目录）结构是在二级目录的基础上发展起来的。
- 二级目录解决了不同用户不能采用同一个文件名的问题，但还是不能反映目录的层次关系。多级目录则更好地反映各个目录及文件的层次关系。
- 在多级目录结构中，对文件的访问是通过文件的路径名进行的。
- 路径名又分为绝对文件名和相对文件名。

7.4.3 目录结构

3.多级目录结构

- 多级目录有以下优点：

（1）层次清楚

（2）解决了重名问题

（3）检索速度快。可为每类文件建立一个子目录，从而很容易查找。

目前的Windows等多种现代操作系统采用的都是多级（树形）目录。

7.4.3 目录结构

3.多级目录结构

多级目录的缺点是：

由于是按路径名逐层查找文件，而每个文件都放在外存，因此查找过程中要多次访问磁盘，从而影响计算机的速度。

注意：

- 若每个用户都想让自己的文件保密，不被他人共享，如何才能做到？
- 那么必须在文件目录中增加一个”存取控制项”，在该目录项中对自己的文件提出保密或其它要求。

7.4.4 文件的操作

- 用户对文件的操作主要包括文件的建立、打开、关闭、删除、读、写和控制等。
- 在树型目录结构中还包括对目录文件的相关操作，如目录文件的建立、改变、显示和删除等。

7.5 文件的操作

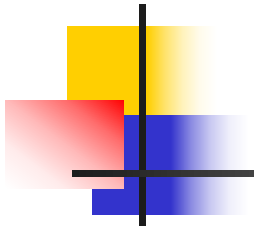
1.建立文件

- 在创建一个新文件时，系统首先要为新文件分配必要的存储空间，并在相应的文件目录中，为之创建一个目录项。
- 用户提供要创建的文件名及若干参数，系统根据用户提供的参数表及系统控制需要来填写目录项中的有关项。

7.5 文件的操作

2.打开文件

- 为了避免用户在每次访问文件时从外存中查找文件目录，系统提供了打开文件的命令。
- 该命令的功能是将待访问文件的目录信息读入内存活动文件表中，建立起用户和文件的联系。



- 一旦文件被打开就可以多次使用，直到文件被关闭为止。
- 有些系统也可以通过读/写命令，隐含地向系统提出打开文件的要求。

7.5 文件的操作

3.读文件

- 打开文件以后，就可以读取文件的信息。
- 在读一个文件时，应在相应的系统调用中给出文件名、读入的内存地址以及读取的字节数。
- 此时系统同样需要查找目录，找到相应的目录项，从而获得文件在外存上的位置。
- 在文件控制块中，有一个指针用于文件的读／写。

7.5 文件的操作

4.写文件

- 在写一个文件的时候，应在相应的系统调用中给出文件名、写入数据的内存地址以及写入的字节数。
- 为此系统也同样要查找目录，查找相应的目录项，从而获知文件在外存上的地址。
- 利用文件控制块中的读／写指针，把内存中指定单元的数据写入到指定文件中 。
- 写入数据时，系统应为其分配物理块，以便把记录信息写到外存上。

7.5 文件的操作

5.关闭文件

- 若文件使用完毕或暂时不用，应将其关闭。
- 关闭文件的功能是撤销内存中有关该文件的目录信息，切断用户与该文件的联系；
- 若在文件打开期间，该文件做过某种修改，还应将其写回辅存。

7.5 文件存储空间的管理

7.5.1 空闲表法

1. 空闲表法

序号	第一空闲盘块号	空闲盘块数
1	2	4
2	9	3
3	15	5
4	—	—

图 6-20 空闲盘块表

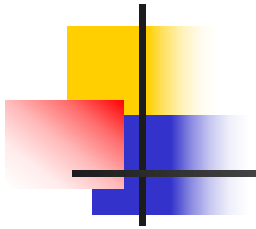
空闲表法的空闲空间管理

- 当请求分配存储空间时，系统依次扫描空闲文件目录，直到找到一个能满足要求的空闲文件为止。若该文件大小大于申请空间量则还要进行划分。
- 当回收存储空间时，也需要顺序扫描空闲文件目录，寻找一个空表目，并将释放空间的第一个物理块号以及释放空间的块数填到这个表目中。若释放空间与已有空闲文件邻接，则需进行合并。

空闲表法的特点

显然，只要将动态分区管理方法中的算法稍作修改，即可用于空闲表法。

- 特点：仅当文件存储空间中只有少量空闲文件时该方法有比较好的效果，否则空闲表变大导致其效率下降。该方法不仅适用创建于连续文件，也适合创建离散文件。



7.5.2. 空闲链表法

(1)　空闲盘块链。

(2) 空闲盘区链

空闲块链

- 空闲块链方法将文件存储设备上的所有空闲块（以块为单位）链接起来，并设置一个头指针指向空闲块链的第一个物理块。
- 当申请分配存储空间时，就按需要从链首依次取下几个物理块分配给文件。当回收存储空间时，将回收的空闲块依次链入空闲块链中。

空闲块链的特点及改进

- 特点：实现简单但工作效率低，因为在空闲块链上增加或移去空闲块时要进行链表操作。
- 一种改进方法是将空闲块分成若干组，再用指针将组与组链接起来，将这种管理空闲块的方法称为空间盘区链。这种管理方法在进行空闲块的分配与回收时要比空闲块链方法节省时间。

unix的成组链接法

1. 空闲盘块的组织

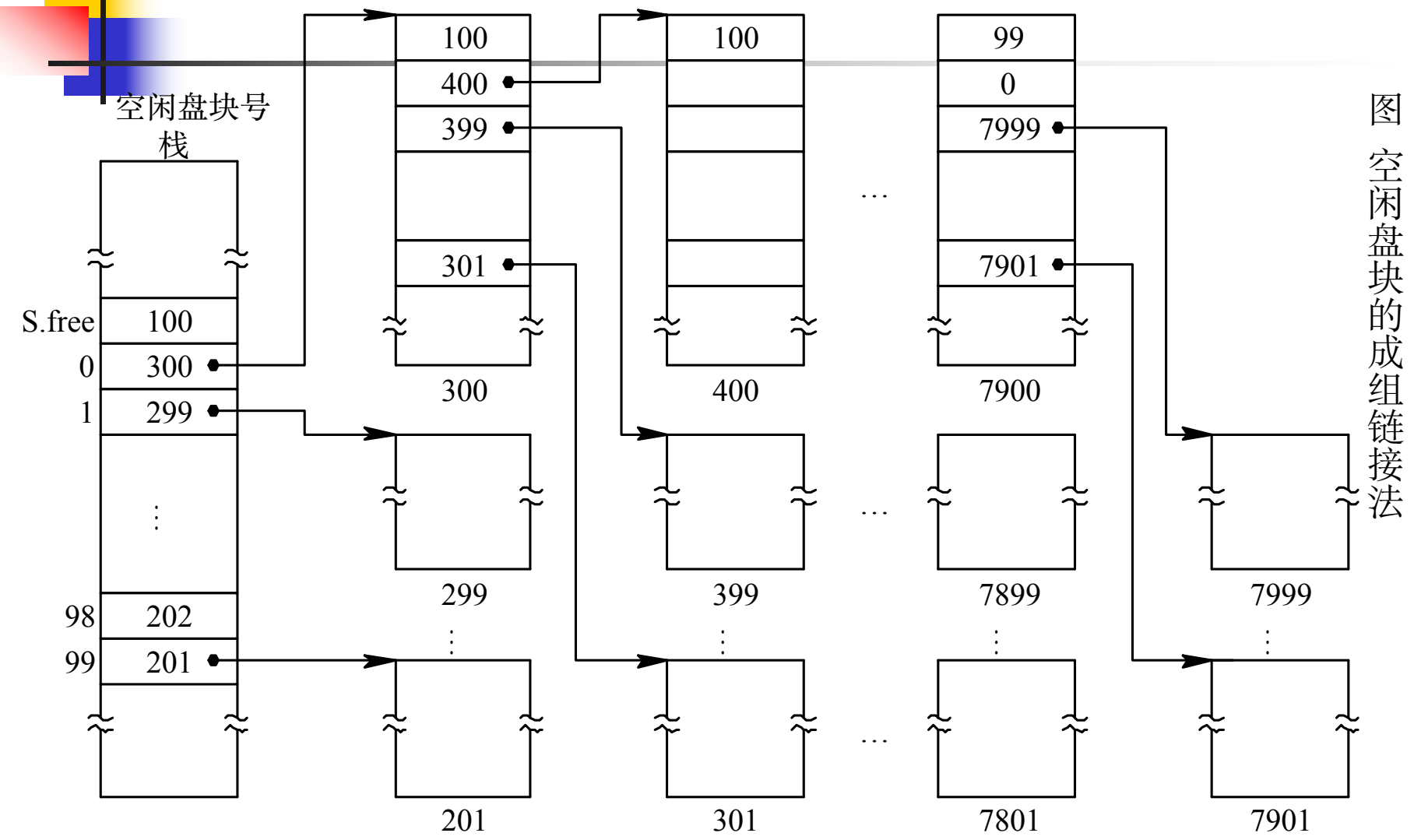
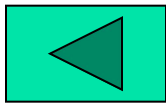


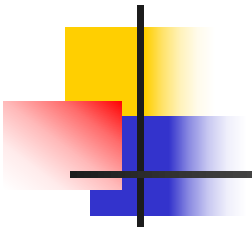
图 空闲盘块的成组链接法

2. 空闲盘块的分配与回收

当系统要为用户分配文件所需的盘块时，须调用盘块分配过程来完成。该过程首先检查空闲盘块号栈是否上锁，如未上锁，便从栈顶取出一空闲盘块号，将与之对应的盘块分配给用户，然后将栈顶指针下移一格。若该盘块号已是栈底， 即S.free(0)，这是当前栈中最后一个可分配的盘块号。由于在该盘块号所对应的盘块中记有下一组可用的盘块号，因此， 须调用磁盘读过程，将栈底盘块号所对应盘块的内容读入栈中，作为新的盘块号栈的内容，并把原栈底对应的盘块分配出去(其中的有用数据已读入栈中)。 然后，再分配一相应的缓冲区(作为该盘块的缓冲区)。最后，把栈中的空闲盘块数减1并返回。

在系统回收空闲盘块时，须调用盘块回收过程进行回收。它是将回收盘块的盘块号记入空闲盘块号栈的顶部，并执行空闲盘块数加1操作。当栈中空闲盘块号数目已达100时， 表示栈已满，便将现有栈中的100个盘块号， 记入新回收的盘块中，再将其盘块号作为新栈底。





成组链接法的特点：

（1）组织的时候以组为单位（一组100块），链比较短

（2）分配和回收时，以块为单位，提高了辅存空间的利用率

7.5.3 位示图法

1. 位示图

常用的管理存储空间的办法是建立一张位示图，以反映整个存取空间的分配请况

用一串二进制位反映磁盘空间中分配使用情况，每个物理块对应一位，"1"表示对应的物理块已分配，"0"表示其对应的块未分配

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	0	0	0	1	1	1	0	0	1	0	0	1	1	0
2	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1
3	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0
4																
⋮																
16																

位示图

2. 盘块的分配

(1) 顺序扫描位示图，从中找出一个或一组其值为“0”的二进制位(“0”表示空闲时)。

(2) 将所找到的一个或一组二进制位， 转换成与之相应的盘块号。假定找到的其值为“0”的二进制位，位于位示的第i行、第j列（i和j都从0开始编号）

则其相应的盘块号应按下式计算：

$$b=n*i+j$$

式中， n代表每行的位数。

(3) 修改位示图， 令map［i,j］=1。

3. 盘块的回收

(1) 将回收盘块的盘块号转换成位示图中的行号和列号。 转换公式为：

i=(b)DIV n

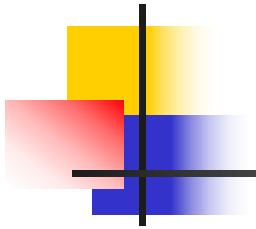
j=(b-1)MOD n

(2) 修改位示图。 令map [i,j] =1。

特点：因位示图比较小，可以保存在主存中，因此空间的分配与回收较快；但需要进行位示图中二进制所在位置与盘块号之间的转换。

7.6 文件的共享（重要）

- 在现代计算机系统中，必须提供文件的共享功能，即系统应允许多个用户（进程）共同使用同一个文件。
- 文件共享能使得被共享的文件在系统中只保留一个副本，从而避免了每个用户各占用一个副本的情况，节省了存储空间。



- 那么，如何能够做到在系统中只保持一个文件副本的情况下，达到多个用户共享该文件的目的呢？

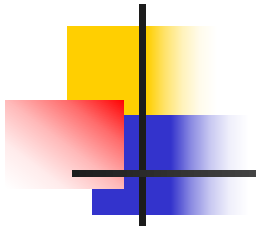
7.6 文件的共享（重要）

- 早在20世纪六、七十年代，就出现了许多实现文件共享的方法：如绕道法、连访法和基本文件目录法。
- 这些早期的方法，目前已逐步淡出历史舞台。
- 下面介绍两种当前常用的文件共享方法——符号链接法、基于索引节点的共享方法。

7.6 文件的共享（重要）

1.符号链接法：

- 快捷方式就是采用符号链接的方法来共享文件。
- 先观察一下Windows是怎样通过快捷方式链接到一个文件，以达到共享该文件的目的：
- 在当前目录下，”右击鼠标／建立快捷方式／通过”浏览“选择要共享的文件”，从而就建立了一个指向另一个文件的链接；



- 通过该链接（快捷方式），就可以在只保持一个文件副本的情况下，达到共享其它文件的目的。

1.符号链接法：

- 当用户想在某个目录下共享某个文件A的时候，系统为用户建立一个类型为LINK的新文件，该LINK类型的文件中只包含了**要共享的文件A的绝对路径名**；
- 当用户试图打开该LINK类型的文件的时候，操作系统首先读符号链接文件的目录项，读出符号链接文件的内容（对于**普通文件**至此对文件的读操作结束），对读出的符号链接文件的内容按照路径进行解析，依次读出路径上的各目录文件，直至读出要共享的文件的内容。

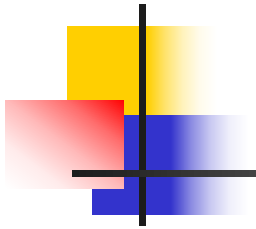
1.符号链接法特点：

- 优点：用符号链接法不仅能共享本机上的文件，而且能通过计算机网络共享到世界上任何一台联网计算机上的文件或文件夹。可以跨文件系统。
- 缺点：访问速度较普通文件慢。

7.6 文件的共享（重要）

2.基于索引节点的共享方式：

- 在前面文件目录中说过，为了提高查找文件目录的速度，人们在目录管理中引入了i节点：
- 即文件目录中只存放文件名和指向i节点的指针，文件的其它信息（物理结构、逻辑结构、物理地址等信息）都存放在指针指引的i节点里；



- 两个用户要共享同一个文件，只需让自己目录中的一个文件所对应的*i*节点指针指向同一个*i*节点即可。共享方法如图：

用户A的目录

文件名	索引节点（i节点）指针
a.c	19
b.txt	33
……	4

索引节点

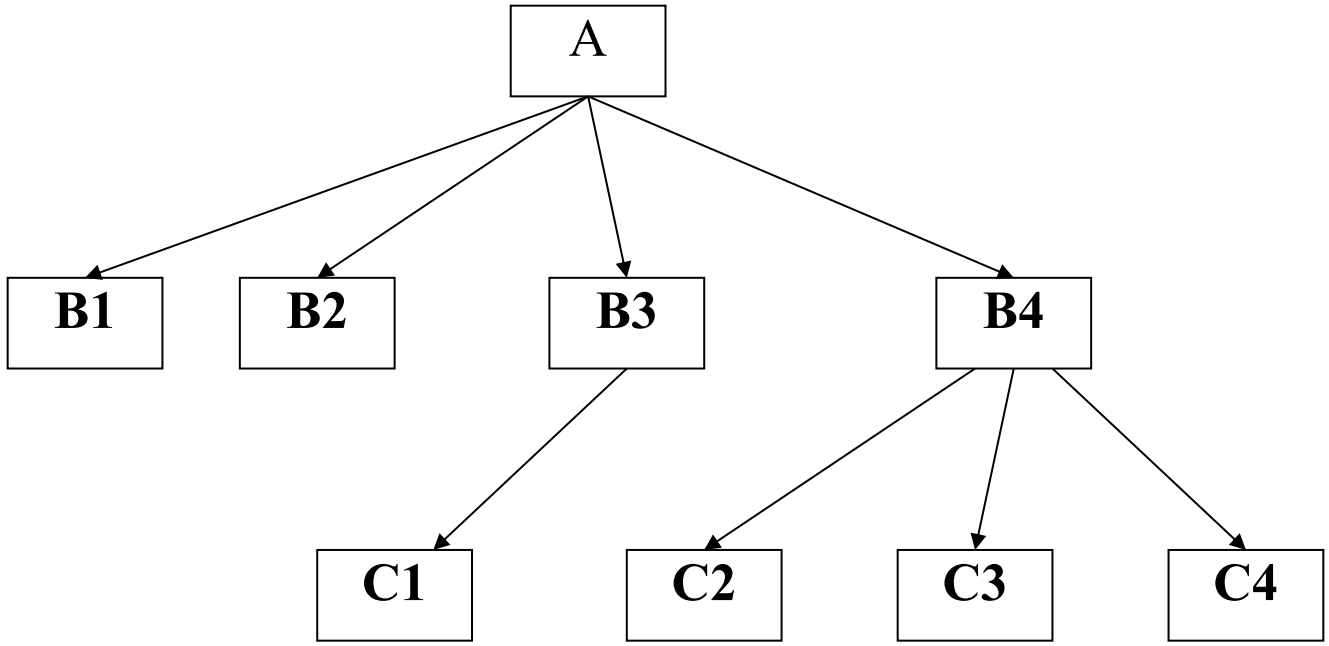
文件物理地址
连接用户数:2

文件

用户B的目录

文件名	i节点指针
c.doc	
b.c	19

例：如图所示，中间的各个节点是目录，叶子节点是文件。（1）能把B2改名为B1吗？（2）能把C1改名为B1吗？（3）能在A根目录下再增加一个目录B3吗？



(4) 若某用户想在自己的P目录下共享图中B3用户目录下的文件C1，该怎么办?

- **参考解答：**

（1）不能。在一个目录下文件不能重名。

（2）可以。因为不在一个目录下，可以重名。

（3）不能。

（4）该用户可以在自己的目录p下建立一个LINK类型的文件，该文件的内容就是一个指向B3用户目录中C1文件的指针。