



课程主要内容

👉 操作系统引论（第1章）

👉 进程管理（第2-3章）

👉 存储器管理（第4-5章）

👉 设备管理（第6章）

👉 文件管理（第7-8章）

👉 操作系统接口（第9章）

👉 UNIX操作系统



第6章 文件管理

□ 文件系统的功能/需解决的问题

❖ 从系统角度看：

负责为用户建立、删除、读、写、修改和复制文件。

❖ 从用户的角度看：

实现了按名存取

文件系统的功能：提供高效、快速、方便的信息存储和访问功能。



第6章 文件管理

- 文件和文件系统
- 文件逻辑结构
- 目录管理
- 文件共享与文件保护
- 外存分配方式（文件的物理结构）
- 文件存储空间的管理
- 数据一致性控制
- UNIX系统的文件管理

本章练习



6.1 文件和文件系统

- 文件、记录和数据项（域或字段---数据的基本单位）
- 文件类型和文件系统模型
- 文件操作

一、数据的组成

例如：用于描述学生的基本数据项-学号、姓名、年龄等

- 数据项（最低级的数据组织形式）
 - 基本数据项（最小的逻辑数据单位）
 - 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单位
 - 组合数据项
 - 由若干个基本数据项组成，简称组项
- 记录
 - 是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
- 文件

工资是个组项，由基本工资、工龄工资和奖励工资等基本项组成



文件

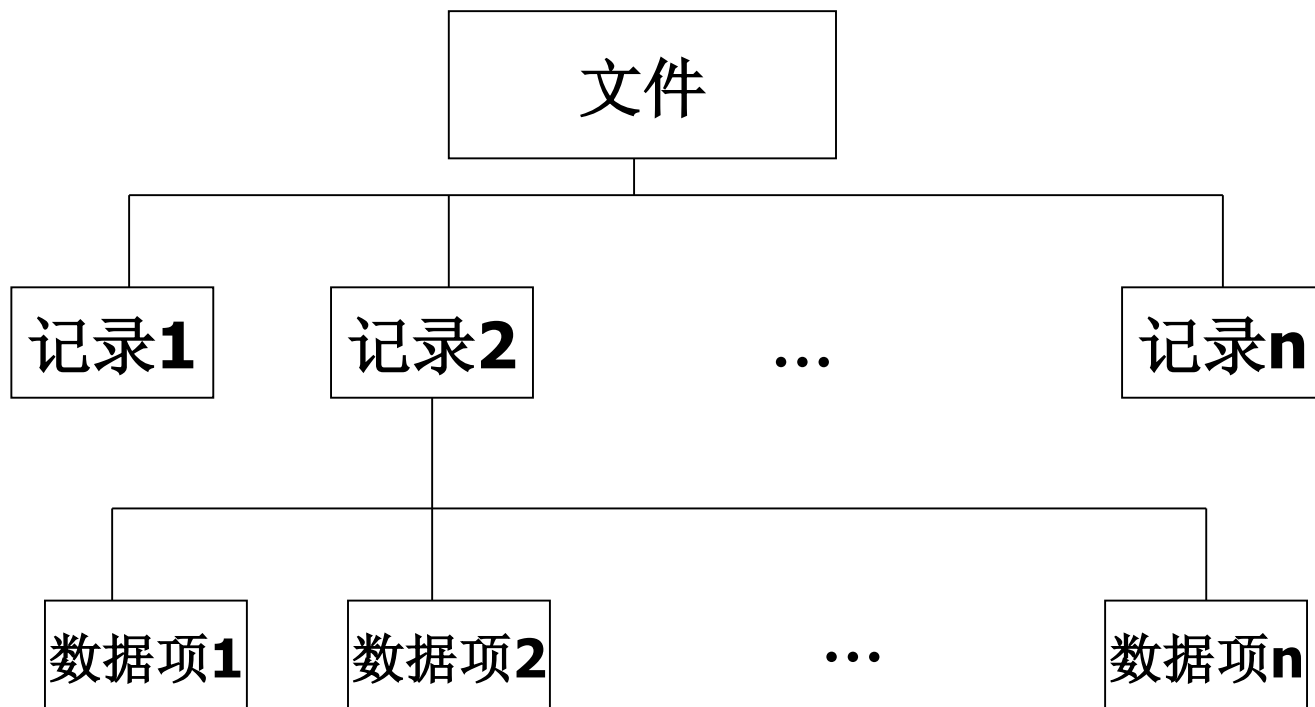
■ 文件

- 在文件系统中是一个最大的数据单位
- 是指记录在外存上的具有文件名的一组相关信息的集合。可分为**有结构文件**和**无结构文件**两种。有结构文件由若干个相关记录组成，而无结构文件则被看成一个字符流。

■ 文件属性

- 文件名、文件类型、文件长度、文件的物理位置、文件的建立日期以及用户对该文件的存取权限等

文件、记录和数据项间的层次关系





二、文件类型（1） ---文件名.扩展名

- 按用途分类
 - 系统文件-由系统软件构成的文件
 - 用户文件-用户的源代码、目标文件、可执行文件或数据
 - 库文件-由标准子例程和常用的例程构成
- 按数据形式分类
 - 源文件
 - 目标文件
 - 可执行文件
- 按存取控制属性
 - 只读文件
 - 读写文件
 - 只执行文件



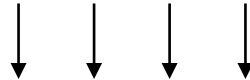
二、文件类型（2） ---文件名.扩展名

- 按组织形式和处理方式分类
 - **普通文件**—由ASCII码或二进制码组成的字符文件
 - **目录文件**—由文件目录组成，用来管理和实现文件系统功能的系统文件
 - **特殊文件**—特指系统中的各类I/O设备
- 按文件的逻辑结构分类
 - 有结构文件（记录式文件）
 - 无结构文件（流式文件）
- 按文件的物理结构分类
 - 顺序文件
 - 链接文件
 - 索引文件



三、文件系统模型（1）

用户（程序）



文件系统接口（命令接口、系统调用接口）

**对对象操纵
和管理的软
件集合**

对文件存储空间的管理
对文件目录的管理
将文件的逻辑地址转换为物理地址的机制
对文件的读和写管理
对文件的共享和保护

对象（文件、目录及磁盘存储空间）及其属性说明



三、文件系统模型（2）

1) 对象及其属性

文件管理系统管理的对象有：

- ① 文件：它作为文件管理的直接对象。
- ② 目录：为了方便用户对文件的存取和检索，在文件系统中必须配置目录。对目录的组织和管理是方便用户和提高对文件存取速度的关键。
- ③ 磁盘(磁带)存储空间：文件和目录必定占用存储空间，对这部分空间的有效管理，不仅能提高外存的利用率，而且能提高对文件的存取速度。



三、文件系统模型（3）

2) 对对象操纵和管理的软件集合

这是文件管理系统的核心部分。文件系统的功能大多是在这一层实现的，其中包括：

- 对文件存储空间的管理
- 对文件目录的管理
- 用于将文件的逻辑地址转换为物理地址的机制
- 对文件读和写的管理
- 对文件的共享与保护等功能



三、文件系统模型（4）

3) 文件系统的接口

为方便用户使用文件系统，文件系统通常向用户提供两种类型的接口：

① 命令接口。这是指作为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（系统调用接口）。这是指作为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。



四、文件操作

用户通过文件系统所提供的系统调用实施对文件的操作。最基本的操作有：

□ 对文件的操作

- 最基本的：创建、**打开**、**关闭**、删除、读、写、截断、设置文件的读写位置（将顺序存取改为随机存取）
- 其它的：文件属性类操作、目录类操作



四、文件操作

文件的“打开”和“关闭”操作

“打开”：系统将文件的属性（目录信息）从外存复制到内存打开文件表中，并返回该表目的编号给用户，建立了用户与文件间的联系。以后若再访问此文件，则利用编号直接在内存中检索，从而节省大量的检索开销，提高了文件的操作速度。

“关闭”：当用户不再需要对该文件的操作时，系统利用关闭文件将文件的属性从内存打开表中删除，从而切断用户与文件之间的联系。



四、文件操作

Linux操作系统对文件操作

1) 文件访问—系统调用

`open()`：打开一个文件，并返回这个文件的描述符。

文件描述符是值很小的整数，当系统开始运行时，一般会有3个已经打开的文件描述符：0 (`STDIN_FILENO`)、1 (`STDOUT_FILENO`)、2 (`STDERR_FILENO`)（在头文件`unistd.h`中定义）

`write()`：把缓冲区的数据写入文件（用文件描述符标识）中（包括写入磁盘、写入打印机等）。

`read()`：从文件（用文件描述符标识）中读出数据。

`close()`：关闭文件，终止文件描述符和文件之间的关联。



四、文件操作

Linux操作系统对文件操作

2) 文件访问—库函数（标准I/O库）

在标准I/O库中，与文件描述符对应的叫“流”，具体实现是指向结构FILE的指针（文件指针）。启动程序时，有三个文件流是自动打开的：stdin（标准输入）、stdout（标准输出）、stderr（标准错误输出）

fopen()：打开一个文件，返回一个FILE *指针。

fwrite()：把数据写到文件中。

fread()：从文件中读取数据。

fclose()：关闭指定的文件，会使所有未写出的数据都写出。

可使用int fileno(FILE *fp)把文件指针转换为文件描述符。



6.2 文件逻辑结构

文件系统设计的关键要素：**记录构成文件的方法**

将文件存储到外存的方法

对任一文件存在着两种形式的结构：

■ 文件的逻辑结构（文件的组织）

从用户观点出发，所观察到的**文件组织形式**，是用户可以直接处理的数据及结构，它独立于物理特性。

■ 文件的物理结构（文件的存储结构）

是指文件在外存上的**存储组织形式**，与存储介质的存储性能有关，还和所采用的外存分配方式有关。（分为顺序、链接及索引结构）

注：文件的逻辑结构和物理结构都将影响文件的检索速度。



6.2 文件逻辑结构

对文件的逻辑结构提出的基本要求：

提高检索速度；

便于修改；

降低文件存储费用。

■ 文件逻辑结构的类型

■ 顺序文件 (Sequential)

■ 索引文件 (Indexed)

■ 索引顺序文件 (Indexed Sequential)



一、文件逻辑结构的类型（1）

■ 有结构的记录式文件

✓ 文件构成： 由一个以上的记录构成。

✓ 记录长度： 分为定长和变长。

✓ 分类（按记录的组织方式）：

顺序文件-定长（也可以是变长）记录按某种顺序排列形成

索引文件-针对变长记录

索引顺序文件-为文件建立一张索引表，为每组记录的第一个记录设置一个表项

一、文件逻辑结构的类型（2）

好处：更加方便、OS代码更加可靠、灵活，用户编程也更加方便

■ 无结构的流式文件

✓ 文件构成：由字符流构成。大量的源程序、可执行文件、库函数等，所采用的就是无结构的文件形式，即流式文件。

有的计算机支持半字节寻址(Nibble Addressing)

✓ 长度：以字节为单位（通用计算机寻址的最小单位）

✓ 访问：对流式文件的访问，则是采用读写指针来指出下一个要访问的字符。

✓ 注：在UNIX系统中，所有的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；系统不对文件进行格式处理。

✓ 可以把流式文件看作是记录式文件的一个特例。



二、顺序文件（1）

■ 逻辑记录的排序

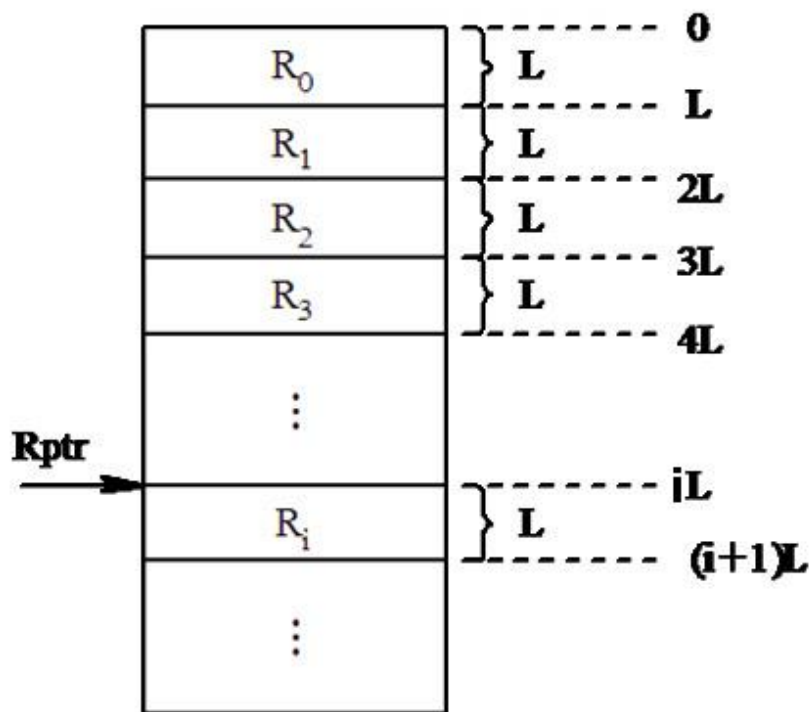
- ✓ 串结构：记录顺序与关键字无关，按存入时间的先后排列。最先存入的记录作为第一个记录，其次存入的为第二个记录， 依此类推。
- ✓ 顺序结构：所有记录按关键字（词）排列。可以按关键词的长短从小到大排序，也可以从大到小排序；或按其英文字母顺序排序。

二、顺序文件（2）

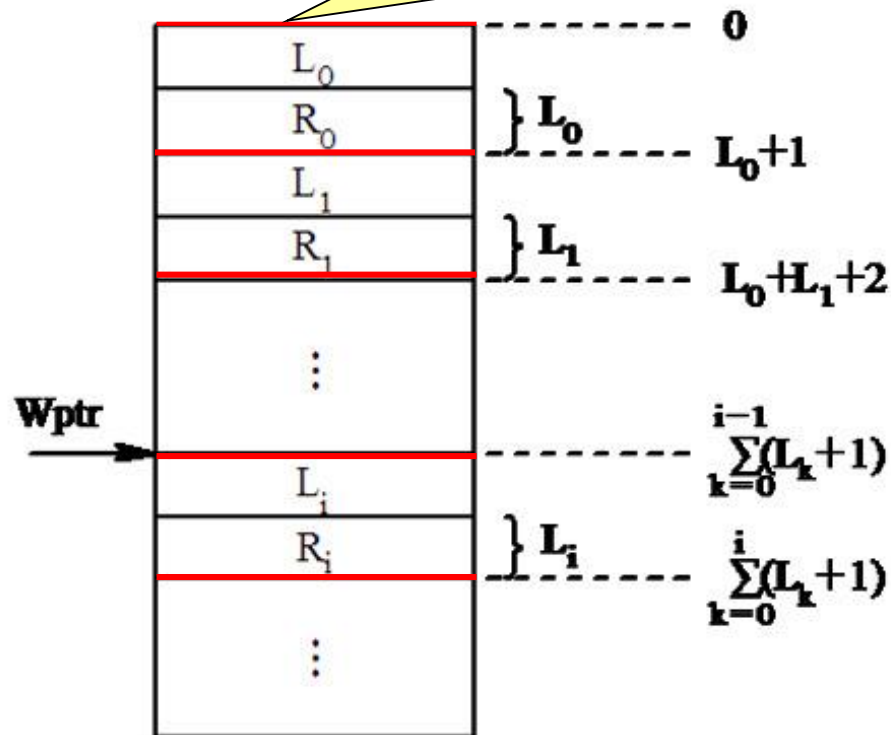
■ 对顺序文件的读、写操作

- ✓ 记录为定长的顺序文件
- ✓ 记录为变长的顺序文件

其中 L_i 占用一个字节，用来指明记录 i 的长度



(a) 定长记录文件



(b) 变长记录文件

图 定长和变长记录文件

顺序文件的优缺点

文件的访问方式：顺序访问和随机访问（直接访问）

■ 优点

- 顺序存取速度较快（批量存取）。
- 对定长记录，还可方便实现直接存取（随机存取）。
- 只有顺序文件才能存储在磁带上，并能有效地工作。

■ 缺点

- ✓ 对变长记录，直接存取低效。例如，有一个含有 10^4 个记录的顺序文件，如果对它采用顺序查找法去查找一个指定的记录，则平均需要查找 5×10^3 个记录；如果是可变长记录的顺序文件，则为查找一个记录所需付出的开销将更大，这就限制了顺序文件的长度。
- ✓ 不利于文件的动态增长。

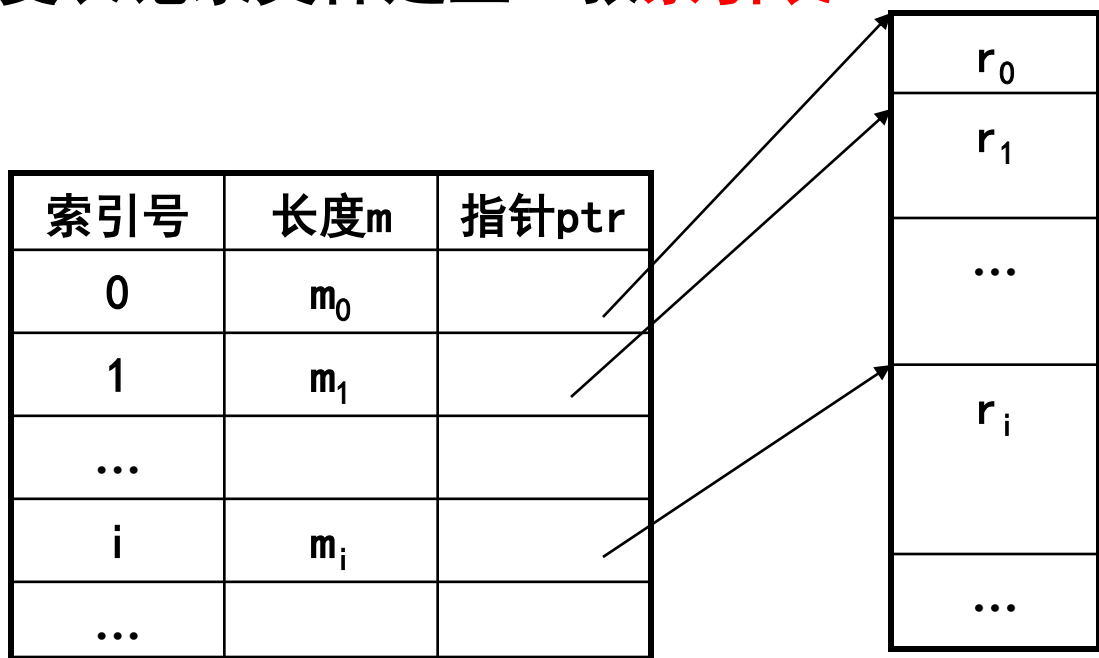
三、索引文件

■ 引入

为解决对变长记录文件难以进行直接存取的问题。

■ 索引文件

为变长记录文件建立一张**索引表**。



索引表

逻辑文件



索引文件的特点

■ 优点

- 通过索引表可方便地实现直接存取，具有较快的检索速度。
- 易于进行文件的增删。

■ 缺点

- 索引表的使用增加了存储费用；
- 索引表的查找策略对文件系统的效率影响很大。

■ 注：若索引表很大，可建立多级索引

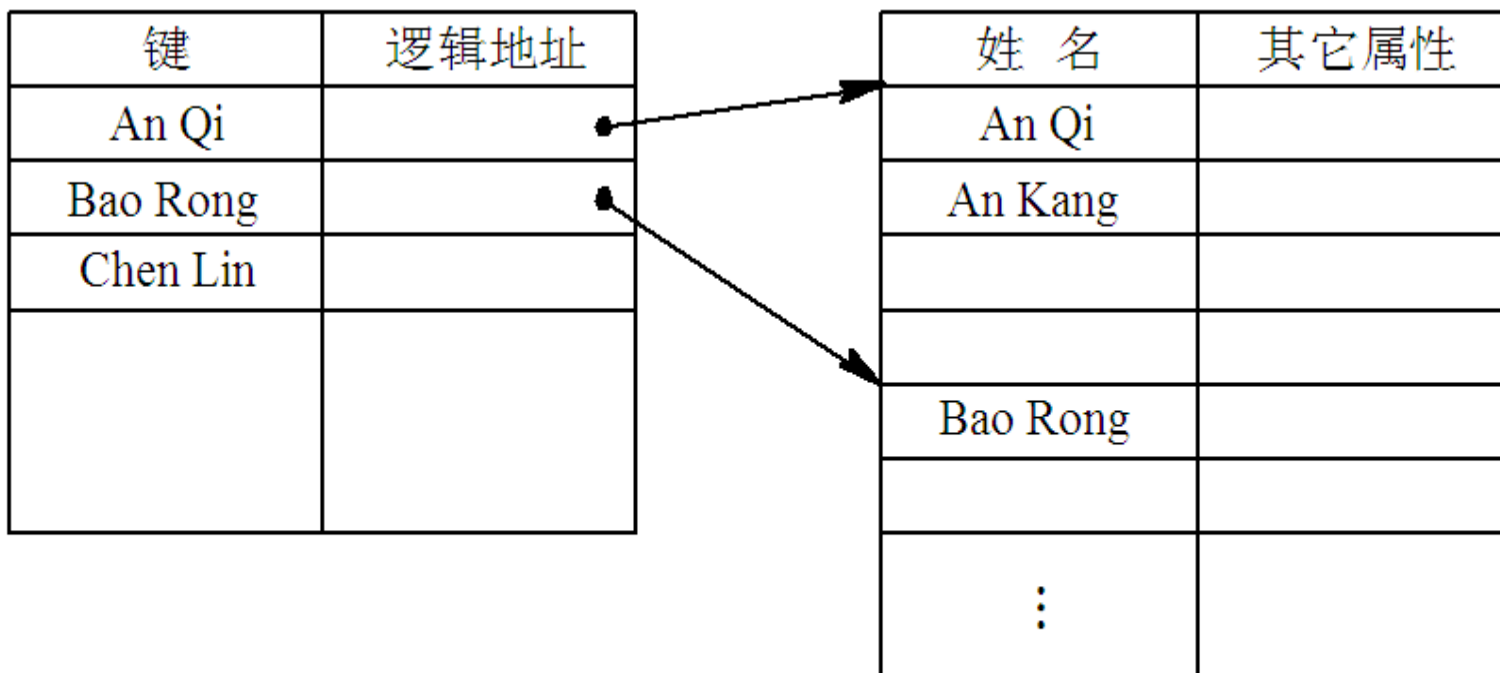
四、索引顺序文件

■ 引入

为解决变长记录文件的直接存取低效且存储费用增加的问题。

■ 索引文件

为顺序文件建立一张**索引表**。



逻辑文件



索引顺序文件的特点

■ 优点

- 通过索引表可方便地实现直接存取，具有较快的检索速度。（检索过程见P248）
- 易于进行文件的增删。

■ 缺点

- 索引表的查找策略对文件系统的效率影响很大。



6.3 目录管理

对文件目录的管理要求

- ✓ 实现“按名存取”
- ✓ 提高对目录的检索速度
- ✓ 文件共享
- ✓ 允许文件重名



6.3 目录管理

- 文件控制块和索引结点
- 单级目录结构
- 两级目录结构
- 树型目录结构
- 目录查询技术



文件控制块和索引结点

从文件管理角度看，文件由FCB和文件体（文件本身）两部分组成。

- 文件控制块（FCB-File Control Block）
 - 文件控制块是操作系统为管理文件而设置的数据结构，存放了文件的有关说明信息，记录了系统对文件进行管理所需要的全部信息。
 - FCB是文件存在的标志。
 - FCB保存在文件目录中，一个FCB就是一个目录项。



文件控制块和索引结点

- 文件控制块（FCB）（续）

- FCB中的信息

- ✓ 基本信息类：文件名、文件长度、类型、属性、文件物理位置
 - ✓ 存取控制信息类：文件存取权限、用户名、口令、共享计数
 - ✓ 使用信息类：文件的建立日期、最后修改日期、保存期限、最后访问日期



文件控制块（FCB）

■ 文件目录

把所有的FCB组织在一起，就构成了文件目录，即文件控制块的有序集合。

■ 目录项

构成文件目录的项目（目录项就是FCB）

■ 目录文件

为了实现对文件目录的管理，通常将文件目录以文件的形式保存在外存，这个文件就叫目录文件。

索引结点

■ 索引结点

■ 索引结点引入

设目录文件所占的盘块数是N，则查找一个目录项平均需要调入盘块 $(N+1)/2$ 次。假设一个FCB为64B，盘块大小为1KB，则每个盘块中只能存放 $1\text{KB}/64\text{B}=16$ 个FCB；若一个文件目录共有640个FCB，需占用 $640/16=40$ 个盘块，平均查找一个文件需启动磁盘20次。

文件名	索引结点编号
文件名1	
文件名2	
.....



索引结点

- 索引结点

- 索引结点引入

文件名	索引结点编号
文件名1	
文件名2	
.....

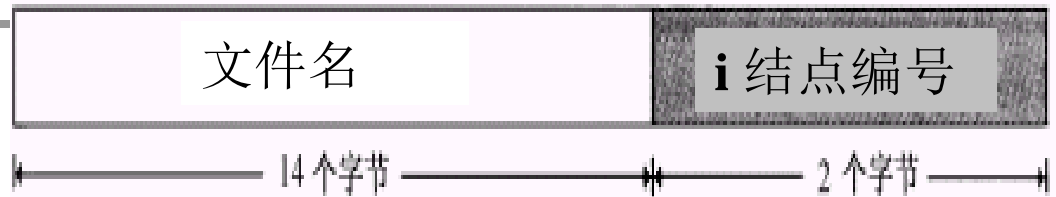
在目录检索的过程中，只用到了文件名，仅当找到一个目录项时，才需从该目录项中读出文件的物理地址。

UNIX系统中，采用了把文件名和文件描述信息分开的方法，把文件描述信息单独形成一个数据结构，叫**索引结点**。

索引结点

- 索引结点

- 索引结点引入



在文件目录中的每个目录项

仅由文件名和指向该文件所对应的i结点（即索引结点）的指针构成。

UNIX系统中，一个目录项只占16个字节（14字节为文件名，2个字节为指向i结点的指针）。在1KB的盘块中可存放 $1\text{KB}/16\text{B}=64$ 个目录项，如果一个文件目录中共有640个目录项，只占用 $640/64=10$ 个盘块，平均启动磁盘 $10/2=5$ 次，减少到原来（平均启动20次）的1/4，大大节省了开销。



索引结点

- 索引结点

- 磁盘索引结点

存放在磁盘上的索引结点，每个文件有唯一的磁盘索引结点。

(主标识、类型、存取权限、物理地址、长度、连接计数、存取时间)

- 内存索引结点

存放在内存上的索引结点。当文件被打开时，要将磁盘索引结点复制到内存的索引结点中，**增加**以下内容：

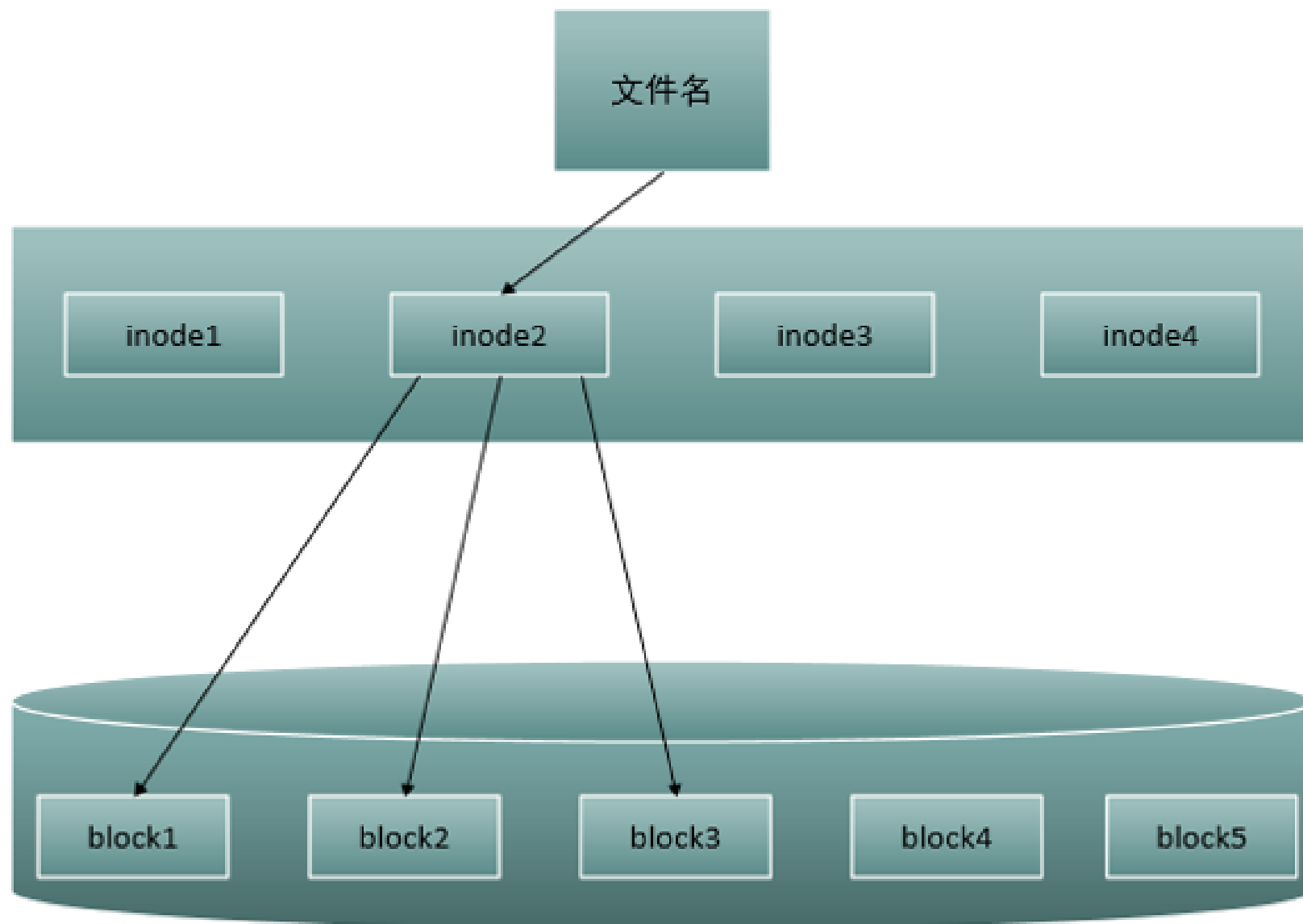
(索引结点编号、状态、访问计数、逻辑设备号、链接指针)

文件名	索引结点编号
文件名1	
文件名2	
.....



Information in a UNIX Disk-Resident Inode

File Mode	16-bit flag that stores access and execution permissions associated with the file. 12-14 File type (regular, directory, character or block special, FIFO pipe) 9-11 Execution flags 8 Owner read permission 7 Owner write permission 6 Owner execute permission 5 Group read permission 4 Group write permission 3 Group execute permission 2 Other read permission 1 Other write permission 0 Other execute permission
Link Count	Number of directory references to this inode
Owner ID	Individual owner of file
Group ID	Group owner associated with this file
File Size	Number of bytes in file
File Addresses	39 bytes of address information
Last Accessed	Time of last file access
Last Modified	Time of last file modification
Inode Modified	Time of last inode modification



Inode-block数据访问示意图



单级目录结构

- 在整个系统中只建立**一张目录表**

文件名	状态位	物理地址	文件其它属性
Alpha			
Report			
Text			
.....			

优点：简单，易实现按名存取

缺点：限制了用户对文件的命名（即易重名）

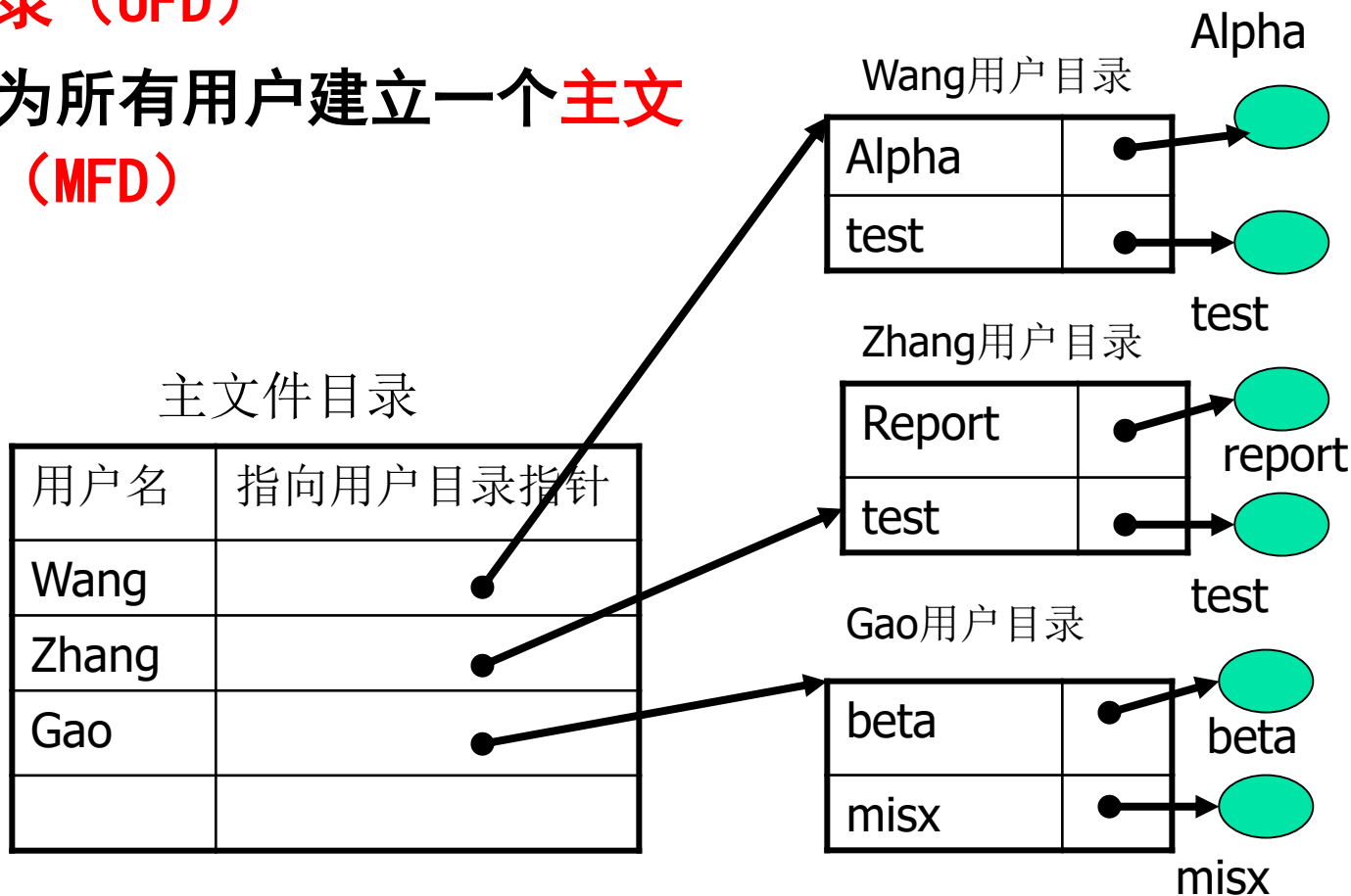
文件平均检索时间长(查找速度慢)

不便于实现文件共享

只适用于单用户环境

两级目录结构

- 在整个系统中建立两级目录
 - 为每个用户建立一个单独的用户文件目录（UFD）
 - 系统中为所有用户建立一个主文件目录（MFD）





两级目录结构

优点:

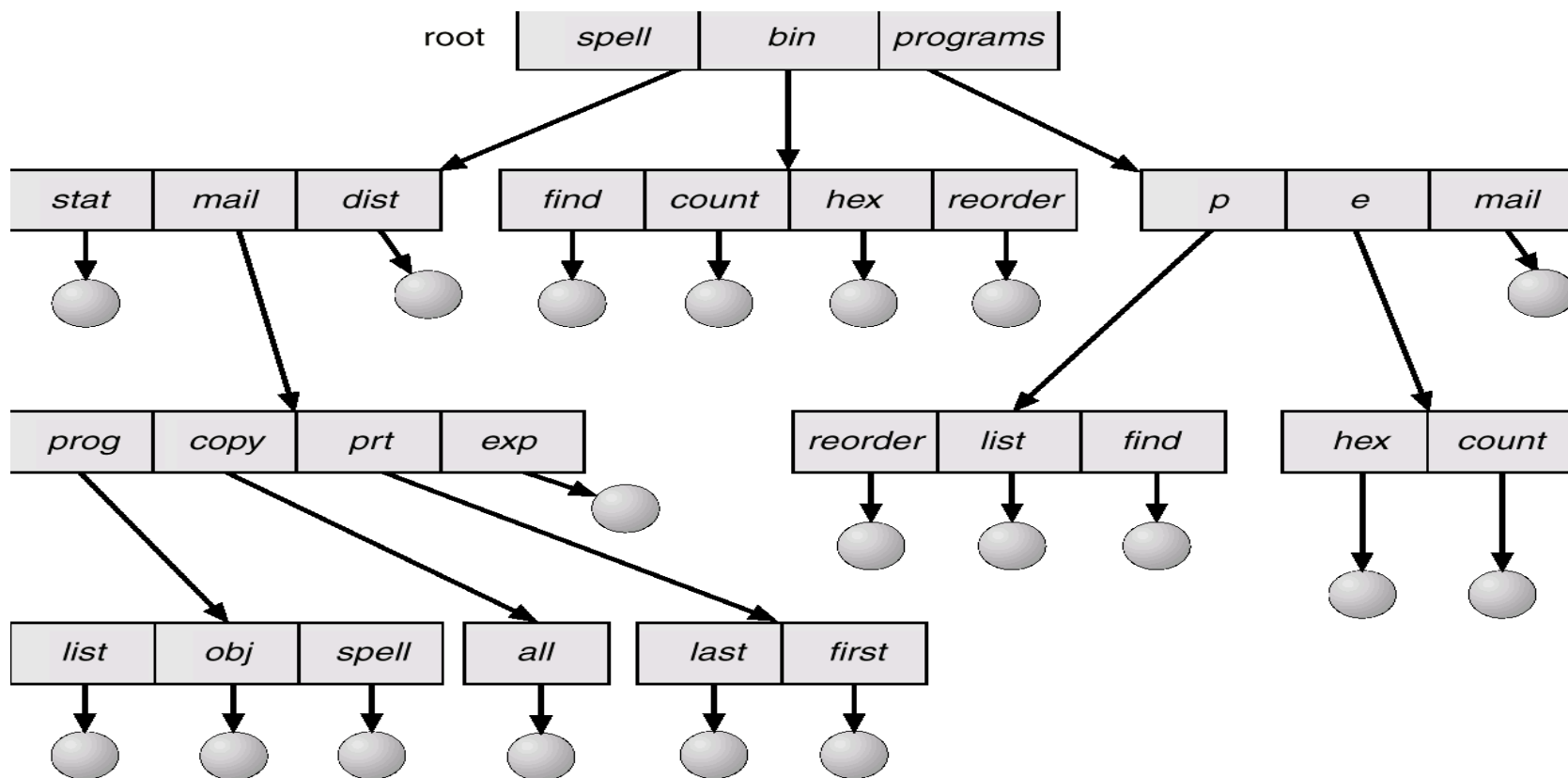
- 提高了检索目录的速度;
- 不同用户目录中可重名;
- 不同用户可用不同文件名来访问系统中一个共享文件

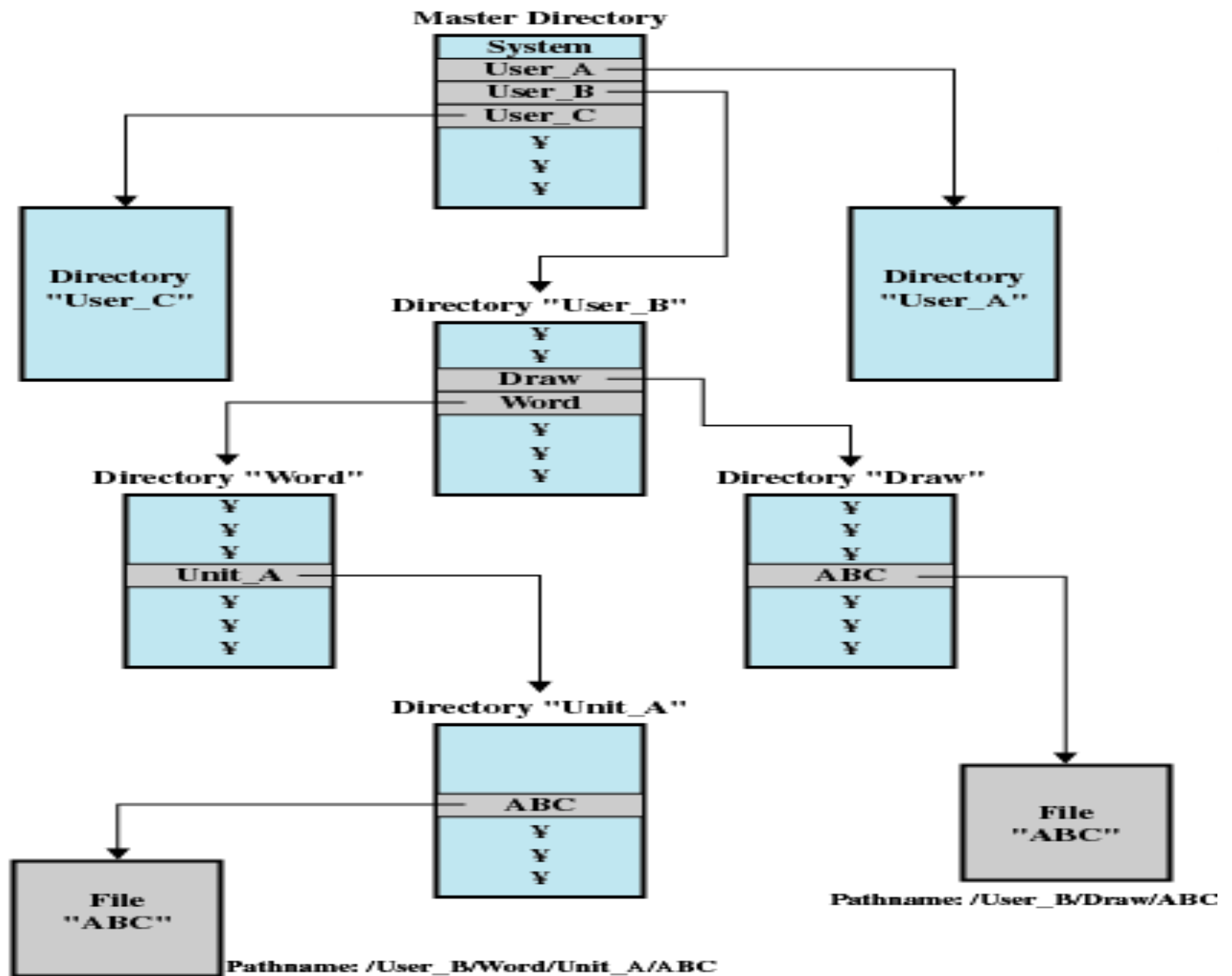
缺点:

- 限制了各用户对文件的共享
- 增加了系统开销, 缺乏灵活性, 无法反映真实世界复杂的文件结构形式。

树型目录结构 (1)

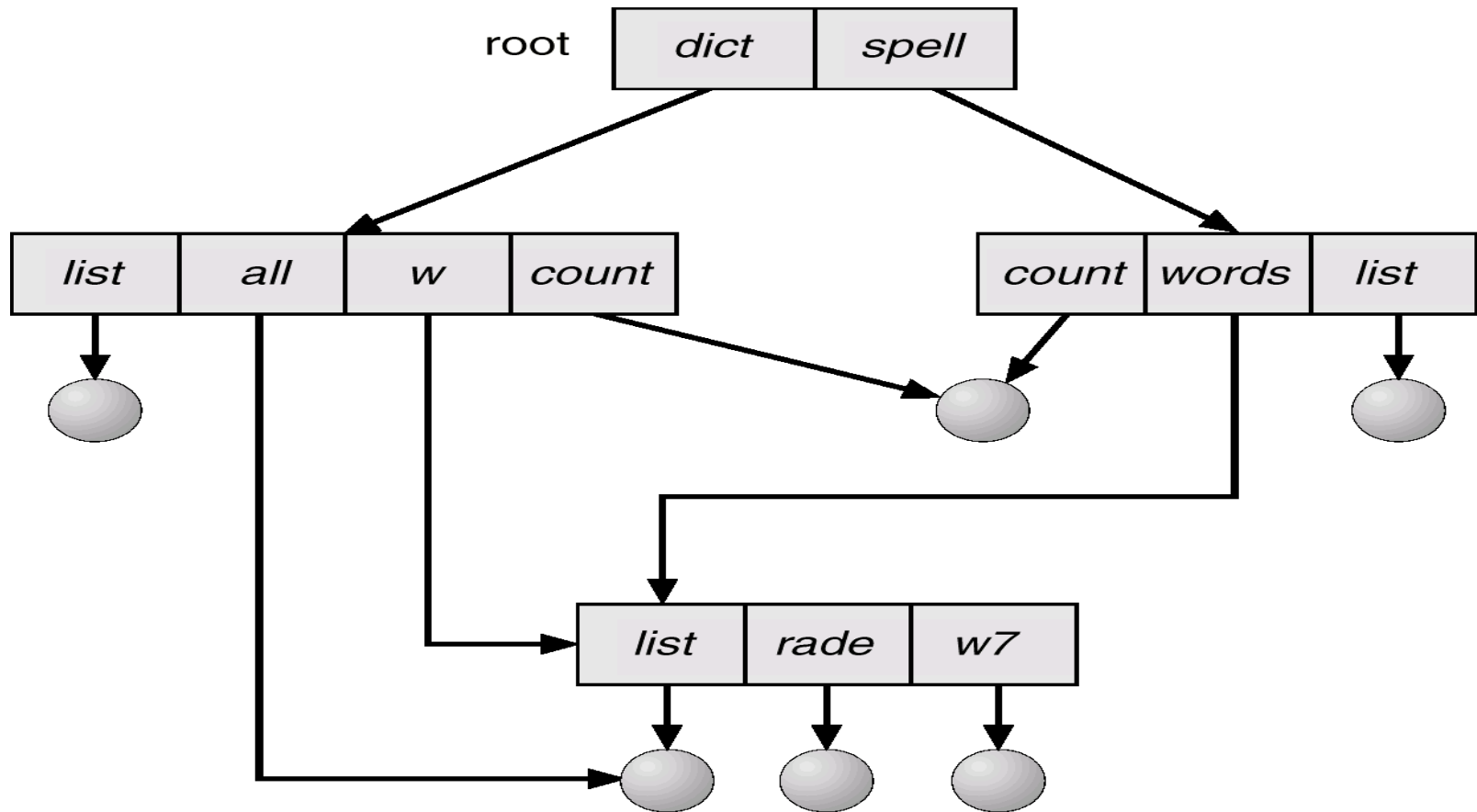
- 在两级目录中若允许用户建立自己的子目录，则形成3级或多级目录结构（即树型目录结构）





Example of Tree-Structured Directory

有共享的子目录和文件





树型目录结构

路径名

- 访问数据文件的一条路径。
- 绝对路径、相对路径

当前目录

增加和删除目录

优点

- 层次结构清晰, 实现分组, 便于管理和保护;
- 解决重名问题;
- 查找速度加快

缺点

- 查找一个文件按路径名逐层检查, 由于每个文件都放在外存, 多次访盘影响速度



目录查询技术

数据文件（按名存取）的查询步骤

- 根据用户提供的文件名，对文件目录进行查询，找到该文件的FCB（索引结点）
- 根据FCB（索引结点）所记录的磁盘盘块号，换算出文件在磁盘上的物理位置
- 启动磁盘驱动程序，读该数据文件至内存中。

对目录进行查询的方式

- 线性检索法（顺序检索法）
- Hash方法

目录查询技术-线性检索法（顺序检索法）

/usr/ast/mbox

根目录

1	.
1	..
4	bin
7	dev
14	lib
9	etc
6	usr
8	tmp

根据 usr 在根目录
中查找到 6 号结点

132 号盘块是
/usr 的目录

132

索引结点 6
(/usr 的索引结点)

496 号盘块是
/usr/ast 的目录

496

索引结点 26
(/usr/ast 的索引结点)

26	.
6	..
64	grants
92	books
60	mbox
81	minik
17	src

/usr/ast/mbox 的索引结点是 60

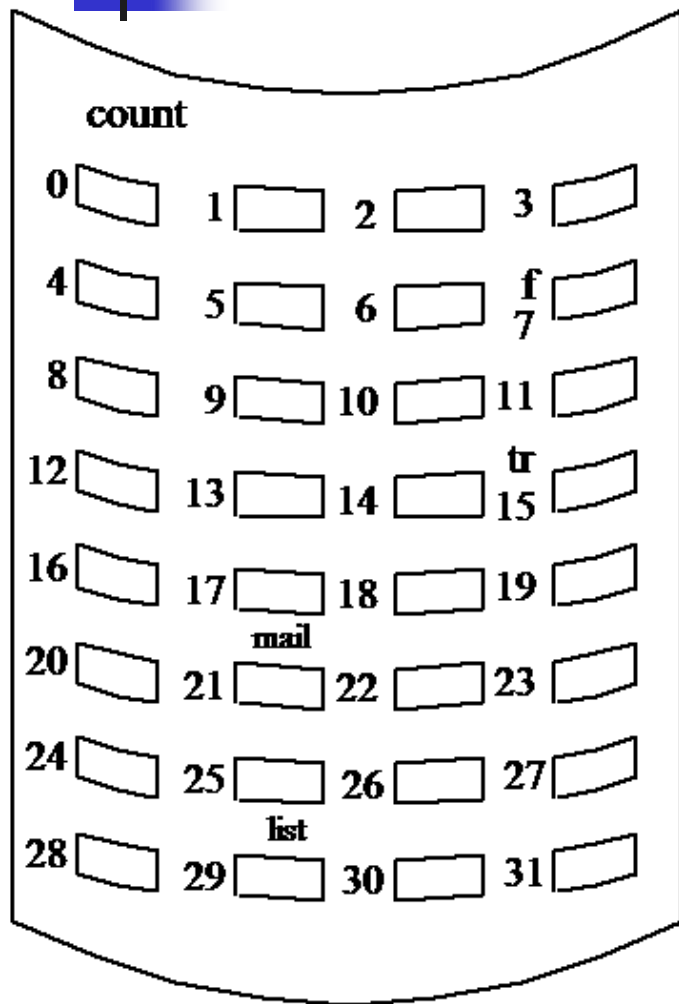
图 查找/usr/ast/mbox的步骤



6.6 外存分配方式

- 文件的物理结构即文件的外存分配方式，是从系统的角度来看文件，从文件在物理介质上的存放方式来研究文件。
- 要考虑的主要问题：
 如何有效地利用外存空间
 如何提高对文件的访问速度
- 目前常用的外存分配方法：
 - (1) 连续分配（顺序分配） —> 顺序文件结构
 - (2) 链接分配 —> 链接文件结构
 - (3) 索引分配 —> 索引文件结构

(1) 外存分配方式-连续/顺序分配 (1)



目录

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

采用连续分配方式时，可把逻辑文件中的记录顺序地存储到邻接的各个物理盘块中，这样所形成的文件结构叫做顺序文件结构，此时的物理文件称为顺序文件。

图 磁盘空间的连续分配



(1) 外存分配方式-连续/顺序分配 (2)

- 为每一个文件分配一片连续的磁盘块
- 只需要起始块号和长度，适用于预分配方法
- 可以随机存取
- 文件不能增长
- 从逻辑地址映射到物理地址较简单
- 浪费空间：动态存储分配问题
- 可以通过紧缩(compact)将外存空闲空间合并成连续的区域。



连续/顺序分配的主要优缺点

■ 主要优点

- ✓ 顺序访问容易
- ✓ 顺序访问速度快

■ 主要缺点

- ✓ 要求有连续的存储空间
- ✓ 必须事先知道文件的长度

(2) 外存分配方式-链接分配 (1)

1. 隐式链接

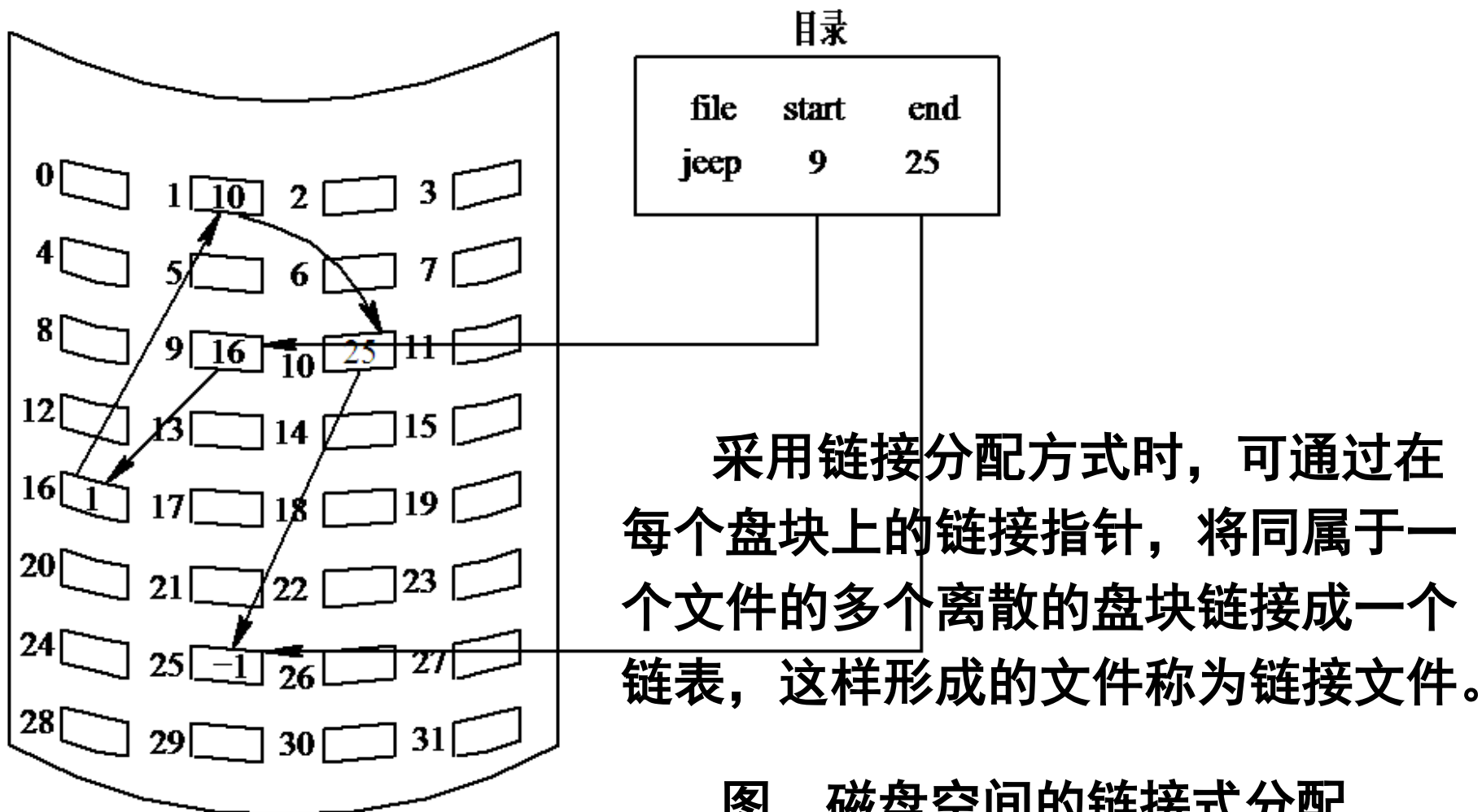


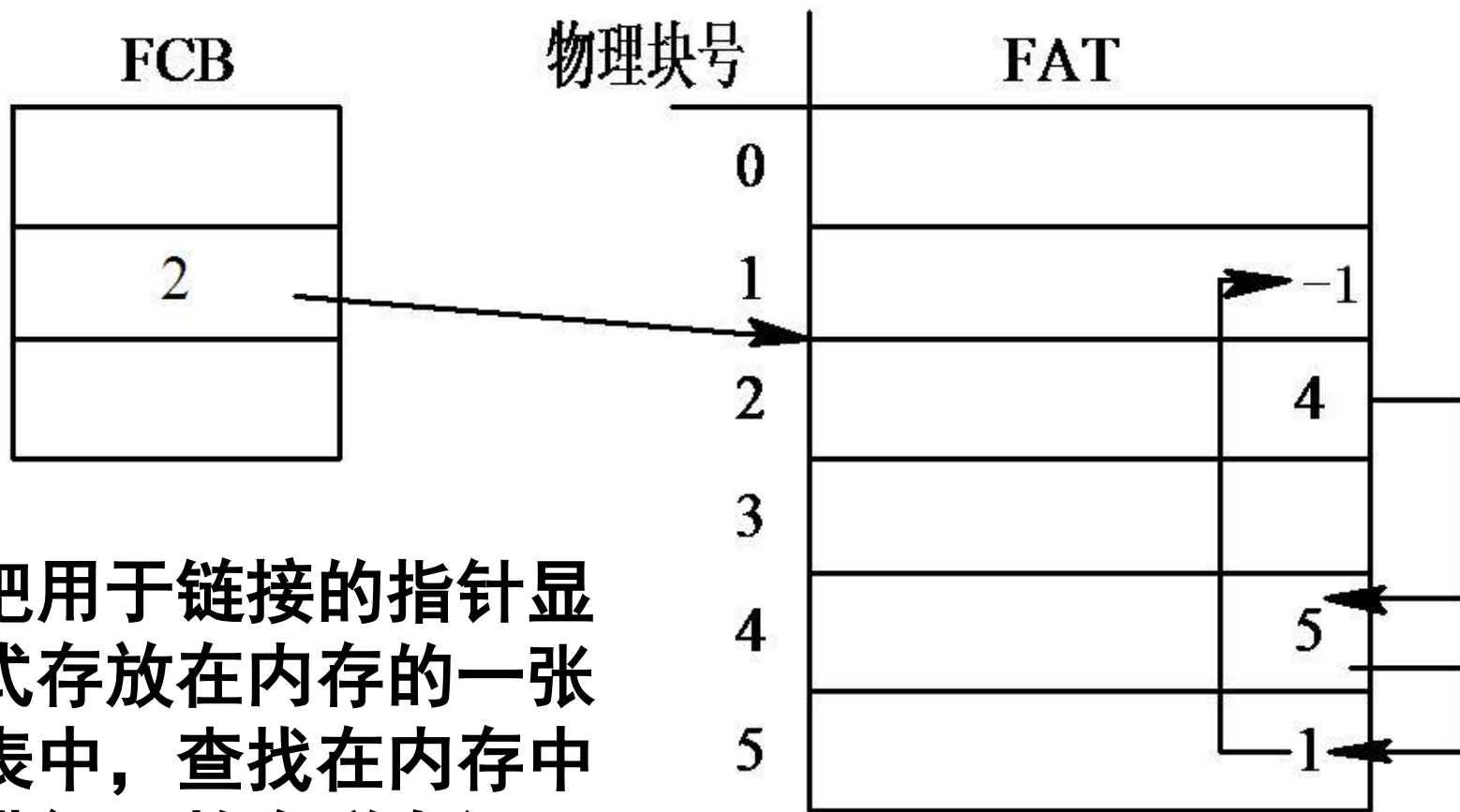
图 磁盘空间的链接式分配

(2) 外存分配方式-链接分配 (2)

- 每个文件是磁盘块的链接列表：块可以分散在磁盘各处
 - 按所需分配磁盘块，链接在一起
 - 在每个块中有指向下一个块的指针
 - 只需要起始地址
- Block(512B)=
- | |
|---------------|
| Pointer(4B) |
| (508B) |
- 可以通过合并(consolidation)将几个盘块组成一个簇，以簇为单位进行盘块分配，以提高I/O访问性能。

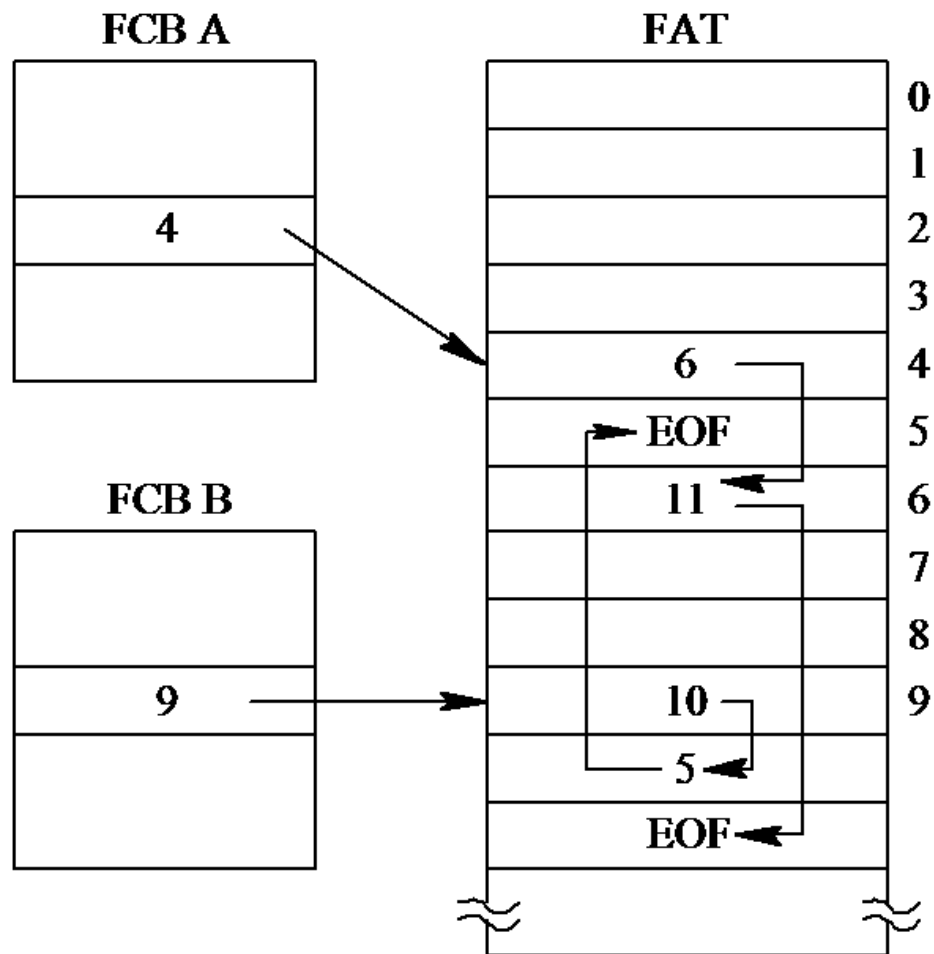
(2) 外存分配方式-链接分配 (3)

2. 显式链接



把用于链接的指针显式存放在内存的一张表中，查找在内存中进行。整个磁盘设置一张表。

图 显式链接结构





文件分配表FAT

- 用于链接文件各物理块的链接指针，显式地存放在内存的一张链接表中。
- 该表在整个磁盘仅设置一张。
- 表序号为整个磁盘的物理块号 ($0 \sim (n-1)$)
- 表项存放链接指针，即下一个块号。
- 文件的首块号存入相应文件的目录项 (FCB) 中。
- 查找是在内存的FAT中进行，故提高了检索速度，同时又减少磁盘的访问次数。
- 被MS-DOS和OS/2等所采用。



链接分配的优缺点

- 优点

- 1、无外部碎片，没有磁盘空间浪费
- 2、无需事先知道文件大小。文件动态增长时，可动态分配空闲盘块。对文件的增、删、改十分方便。

- 缺点

- 1、不能支持高效随机/直接访问，仅适合于顺序存取
- 2、需为指针分配空间。（隐式链接）
- 3、可靠性较低（指针丢失/损坏）
- 4、文件分配表FAT（显式链接）
FAT需占用较大的内存空间。



计算FAT表所占的内存空间

- 整个系统有一张FAT表，在FAT的每个表项中存放下一个盘块号。对于1.2M的软盘，每个盘块的大小为512B，在FAT表中共有2.4K个表项，每个FAT表项占12位（FAT12），所以FAT表占用3.6KB的存储空间。
- 一个物理磁盘能支持4个逻辑磁盘分区，每个逻辑分区设置一张FAT表，盘块（扇区）大小为512B，则每个逻辑分区大小为 $512\text{B} \times 2^{12} = 2\text{MB}$ ，则相应的磁盘容量仅为8MB
- 引入“簇”（一组相邻扇区）作为基本分配单位



计算FAT表所占的内存空间

- FAT12（只支持短文件名）、FAT16、FAT32（Windows98以及后续的Windows系统，支持长文件名，不支持4GB以上的单个文件；而MS-DOS使用FAT12和FAT16）
- Windows2000/XP以及后续的Windows OS使用NTFS（New Technology File System）
- 若磁盘大小为20GB，盘块大小为4KB，计算FAT表的大小（FAT32）。若给文件A依次分配了3、6、2、5四个物理块，画出分配后的FAT表。



(3) 外存分配方式-索引分配 (1)

- 链接分配方式虽然解决了连续分配方式所存在的问题，但又出现了另外两个问题， 即：
 - (1) 不能支持高效的直接存取。要对于一个较大的文件进行直接存取，必须首先在FAT中顺序地查找许多盘块号。
 - (2) FAT需占用较大的内存空间。
- 为每一个文件分配一个索引块（表），再把分配给该文件的所有块号，都记录在该索引块中。故索引块就是一个含有许多块号地址的数组。
- 该索引块的地址由该文件的目录项指出。
- 支持随机/直接存取。
- 不会产生外部碎片。
- 适用于文件较大时。

(3) 外存分配方式-索引分配 (2)

1. 单级索引分配

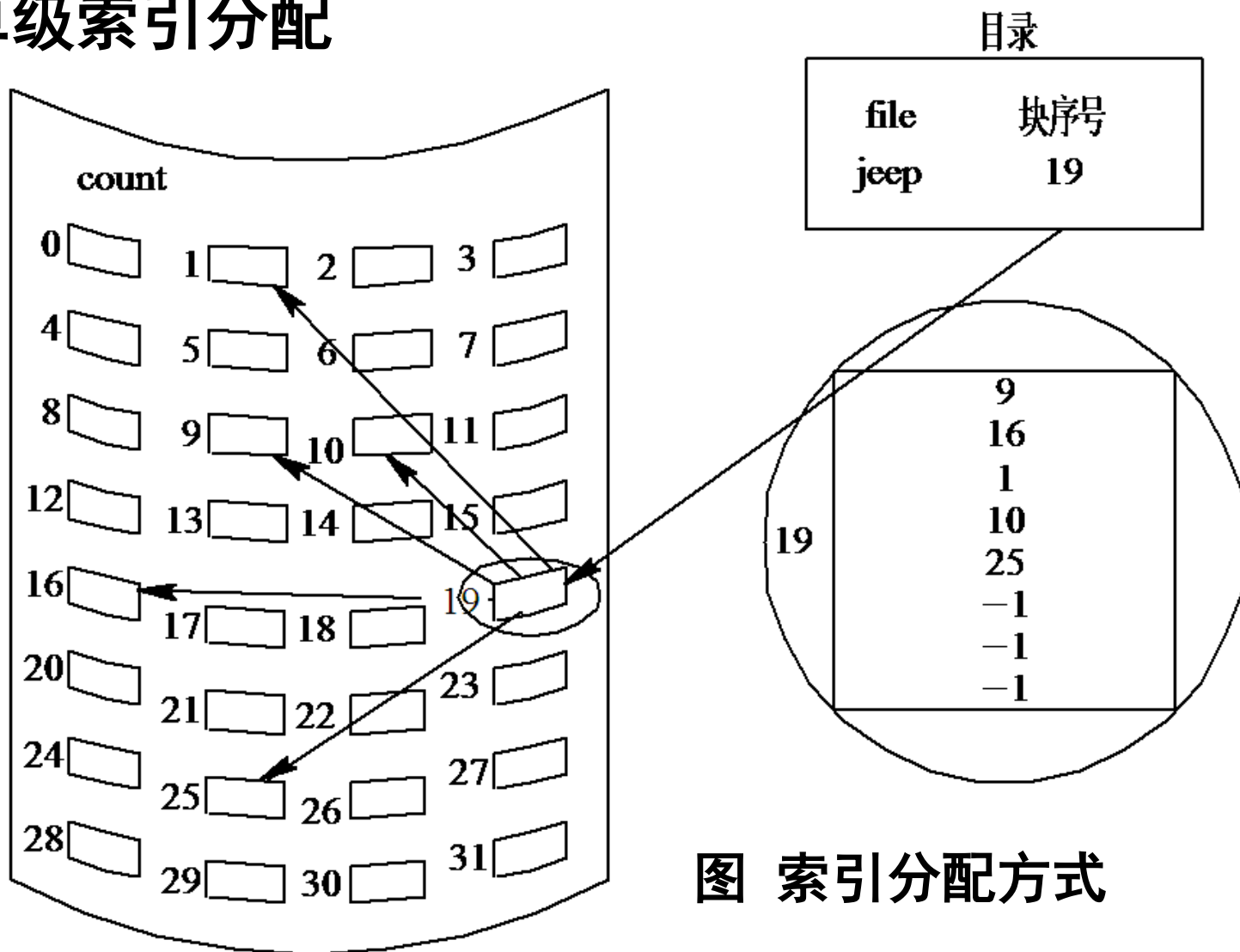
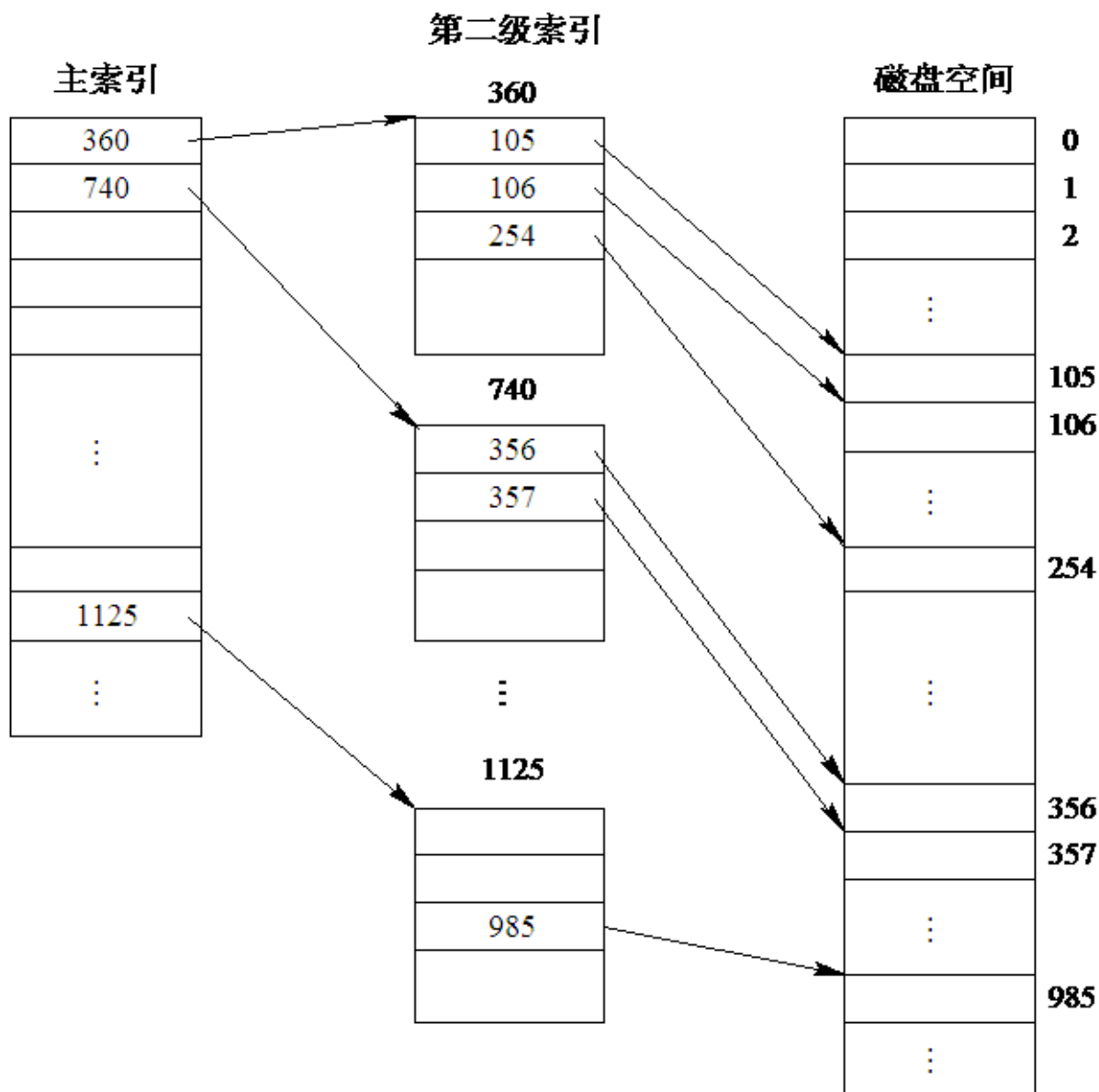


图 索引分配方式

(3) 外存分配方式-索引分配 (3)

2. 多级索引分配

图 两级索引分配





(3) 外存分配方式-索引分配 (4)

假设每个盘块大小为1KB，每个盘块号占4个字节，则在一个索引块中可放256个文件物理块的盘块号。在两级索引时，最多可包含的存放文件的盘块的盘块号总数为 $256*256=64K$ 个盘块号。可以得出：采用两级索引时，所允许的文件最大长度为64MB。



(3) 外存分配方式-索引分配 (5)

3. 混合索引方式

UNIX文件系统采用的是混合索引结构(综合模式)。每个文件的索引表为13个索引项。最前面10项直接登记存放文件信息的物理块号(直接寻址)

如果文件大于10块,则利用第11项指向一个物理块,假设每个盘块大小为4KB,每个盘块号占4个字节,该块中最多可放1K个盘块号(一次间接寻址)。对于更大的文件还可利用第12和第13项作为二次和三次间接寻址。

(3) 外存分配方式-索引分配 (6)

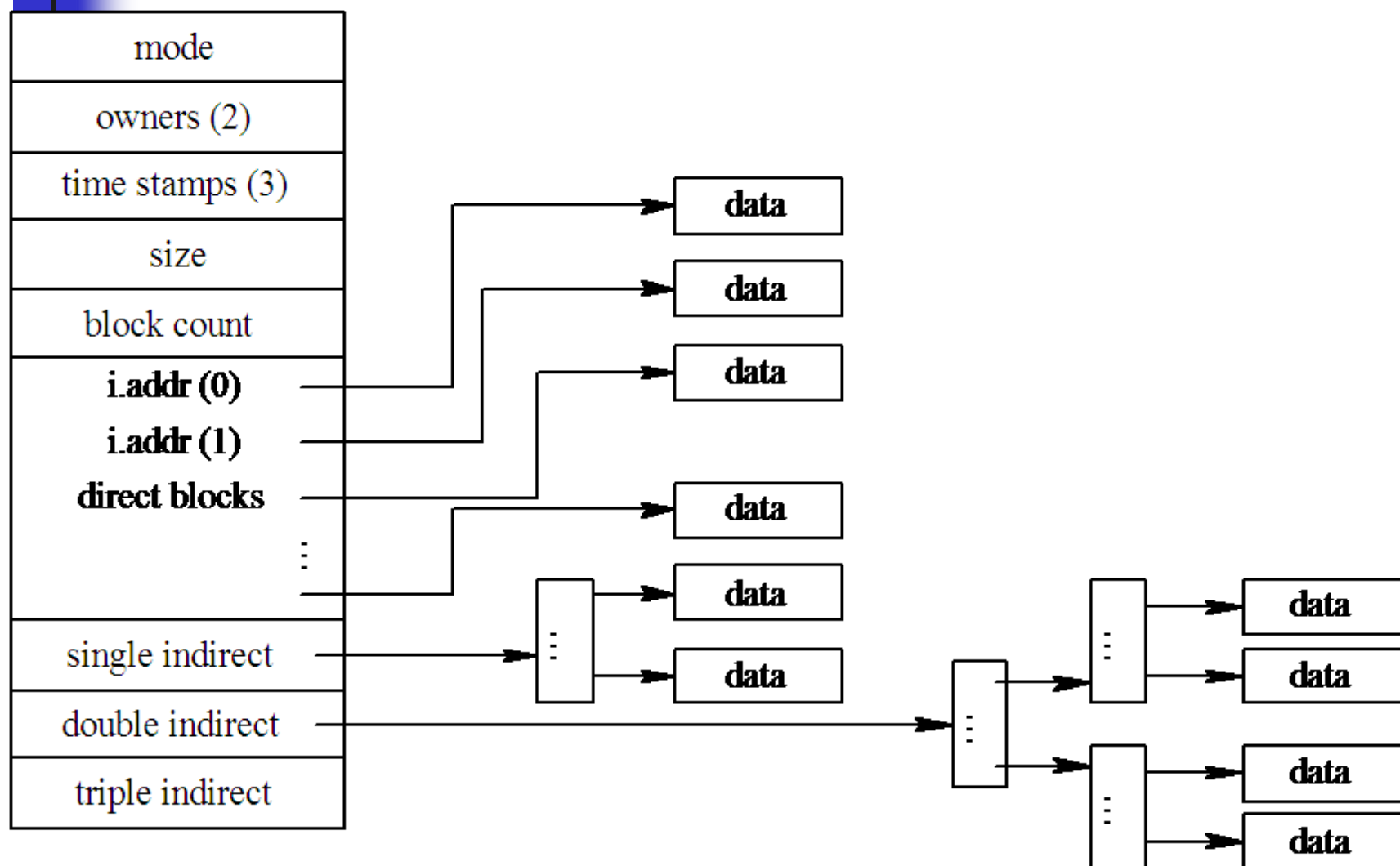


图 混合索引方式



(3) 外存分配方式-索引分配 (7)

(1) 直接地址

为了提高对文件的检索速度，在索引结点中可设置**10**个直接地址项，即用 **iaddr(0)~iaddr(9)** 来存放直接地址。即每项中所存放的是该文件数据所在盘块的盘块号。假如每个盘块的大小为 **4 KB**，当文件不大于**40 KB**时，就可以直接从索引结点中读出该文件的全部盘块号。



(3) 外存分配方式-索引分配 (8)

(2) 一次间接地址

对于大、中型文件，只采用直接地址是不现实的。可再利用索引结点中的地址项 $iaddr(10)$ 来提供一次间接地址。这种方式的实质就是一级索引分配方式。图中的一次间址块也就是索引块，系统将分配给文件的多个盘块号记入其中。在一次间址块中可存放1K个盘块号（4KB/4B），因而允许文件长达4 MB。

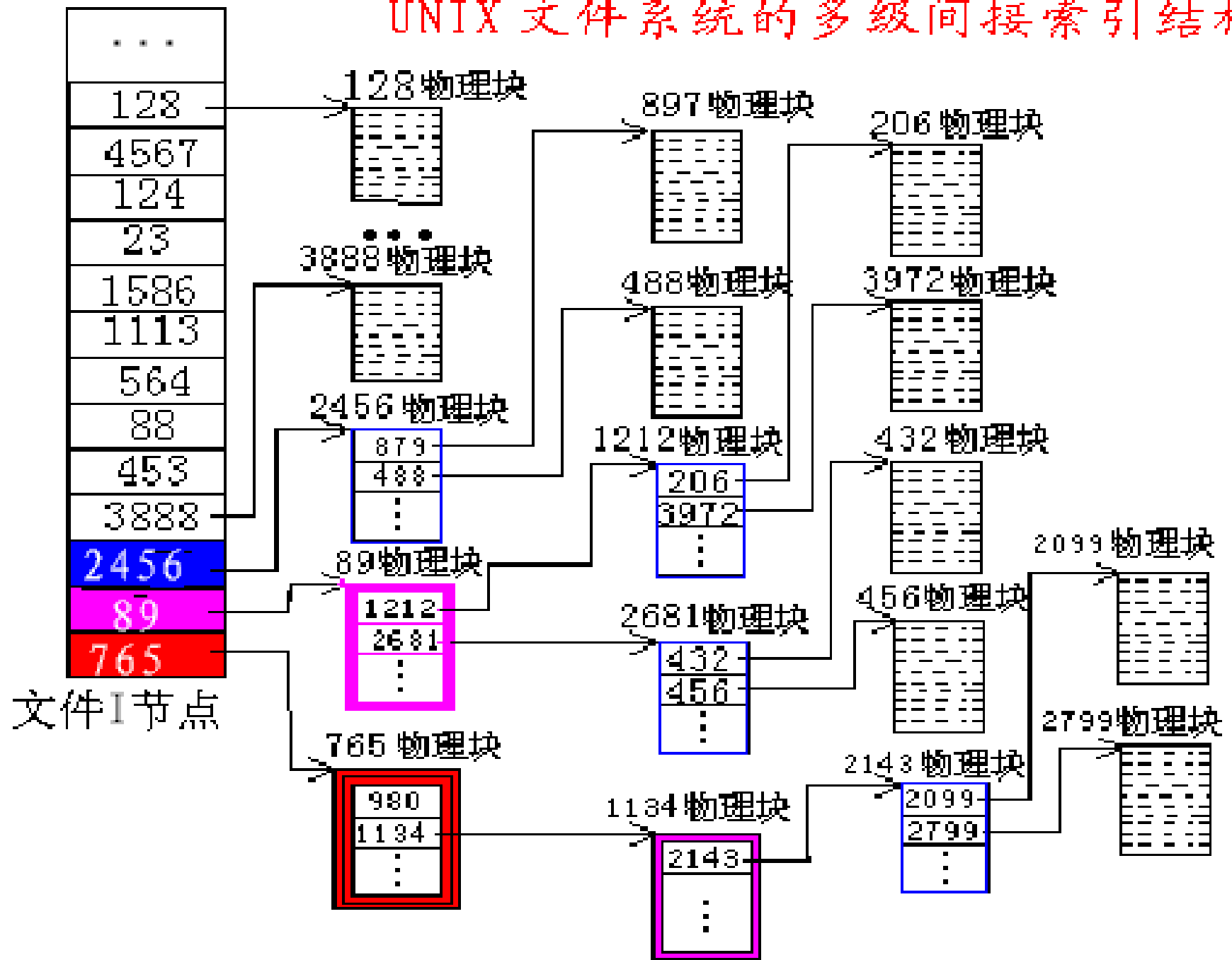


(3) 外存分配方式-索引分配 (9)

(3) 多次间接地址

当文件长度大于4MB+40KB时(一次间址与10个直接地址项)，系统还须采用二次间址分配方式。这时，用地址项 $iaddr(11)$ 提供二次间接地址。该方式的实质是两级索引分配方式。系统此时是在二次间址块中记入所有一次间址块的盘号。在采用二次间址方式时，文件最大长度可达4GB ($4KB * 1K * 1K$)。同理，地址项 $iaddr(12)$ 作为三次间接地址，其所允许的文件最大长度可达4TB ($4KB * 1K * 1K * 1K$)。

UNIX 文件系统的多级间接索引结构



文件的物理结构、存取方式与存储介质的关系

存储介质	磁带	磁盘		
物理结构	连续结构	连续	链接	索引
存取方式	顺序存取	顺序	顺序	顺序
		随机		随机