第6章 文件管理

6.1 文件与文件系统

6.2 文件结构与存储设备

6.3 文件目录管理

6.4 文件存储空间管理

6.5 文件的共享

文件管理引入

- 用户要求：
  - 信息要长期保留
  - 信息要按逻辑关系组织
  - 信息要方便被访问
  - 信息要安全可靠
- 为满足用户要求需要解决：
  - 硬件：提供存储介质
  - OS：解决信息如何组织，存放，如何快速找到、采取什么安全措施、如何共享、如何管理存储介质、如何提高访问效率、给用户提供常用的操作实现例程等

6.1 文件和文件系统

6.1.1 文件的概念

1、文件的定义

文件是计算机系统中信息存放的一种组织形式，下面给出两种具有代表性的解释:

- 文件是赋名的信息 (数据)项的集合。
- 文件是赋名的、有关联的信息单位 (记录)的集合。

6.1.1 文件的概念

这两种解释定义了两种文件形式:

- 前者说明文件是由字节组成，这是一种无结构的文件，或称流式文件。目前UNIX操作系统、MS-DOS系统均采用这种文件形式。
- 后者说明文件是由记录组成。而记录则是由一组相关信息项组成。例如每个学生的登记表可视为一个记录，它包括学生姓名，出生年月，性别，籍贯等信息项。所有学生登记表组成一个学生文件。

■ 定义：文件是指可保存的、具有标识名的一组逻辑上具有完整意义的信息集合。

6.1.1 文件的概念

2、文件的属性

- 每个文件都有一个名字和它所保存的信息，此外，OS还给每个文件附加了一些其它管理用的信息，这些信息称为文件的属性。
- 常见的一些文件属性：
  - 保护信息：谁可以对该文件进行何种操作；
  - 创建者：该文件是谁创建的；
  - 只读标志位：0表示可读/写，1表示只读；
  - 隐藏标志位：0表示普通文件，1表示隐藏文件；
  - 系统标志位：0表示普通文件，1表示系统文件；
  - 创建时间、最后访问时间、最后修改时间；
  - 文件长度。

6.1.2 文件的类型

1、以文件的性质和用途分类

- 系统文件: 指用操作系统的执行程序和数据组成的文件，这种文件不对用户开放，仅供系统使用。
- 库文件：是指系统为用户提供的各种标准函数，标准过程和实用程序等。用户只能使用这些文件，而无权对其进行修改。
- 用户文件: 由用户的信息组成的文件，如源程序文件，数据文件等。这种文件的使用和修改权均属于用户。

6.1.2 文件的类型

2、从按文件的存取控制属性分类

- 只读文件: 只允许进行读操作。
- 读写文件: 允许进行读写操作。
- 执行文件: 允许执行。

3、按文件的组织形式分类

- 普通文件: 指一般的用户文件和系统文件。
- 目录文件: 指由文件目录项组成的文件。
- 特别文件: 有的系统把设备作为文件统一管理和使用，并为区别起见，把设备称为特别文件。

文件类型示例

File name	extension	description
Executable	exe, com, bin	ready-to-run machine-language program
Object	obj, o	complied, machine language, not linked
Source code	c, pas, F77, asm java	source code in various languages
Batch	bat, sh	commands to the command interpreter
Text	txt, doc	textual data documents
Word processor	wp, tex, rrf, etc.	various word-processor formats
Library	lib, DLL	libraries of routines
Print or view	ps, dvi, gif	ASCII or binary file
Archive	arc, zip, tar	related files grouped into one file, sometimes compressed.

UNIX文件分类

- 普通文件
  - 内容可以是程序、数据、图象等，
- 目录文件
  - 由若干文件属性描述表构成，保存在磁盘块中
- 特殊文件
  - 设备（块设备文件、字符设备文件）

设备作为文件管理的好处：

- 界面统一，使用文件与使用设备命令相同，如申请设备open, 释放close, 读read, 写write，… …
- 利用文件保护功能可以保护设备

6.1.3 文件的操作

1．建立文件
2．打开文件
3．读文件
4．写文件
5．关闭文件
6．删除文件
7．文件指针定位

6.1.4 文件系统

- 文件系统：OS中负责处理文件相关事宜的程序和数据结构。包括文件的查找、存放、保护、共享、命名、文件常用操作的实现以及文件存储器的管理等等。功能包括：
  - 实现文件的按名存取，即实现名字到外存地址的映射
  - 实现文件的信息组织、存放
  - 实现文件的安全与保护
  - 实现文件的共享
  - 给用户提供文件的操作例程
  - 管理文件存储器

6.2 文件结构与存储设备

6.2.1 概述

- 研究文件结构有两种观点：

1.用户的观点（文件的逻辑结构）：主要研究用户思维中的抽象文件，为用户提供一种逻辑结构清晰、使用简便的逻辑文件。用户将按这种形式去存取、检索和加工文件。例如用户可将文件看作字节的集合，也可以将文件看作记录的集合。

2. 实现的观点（文件的物理结构）：主要研究驻留在存储介质上的文件的结构。

6.2.2 文件的逻辑结构

1. 逻辑结构

- 流式文件: 基本信息单位是字节或字，其长度是所含字节的数量。这种文件的优点是节省存储空间。在这种文件中无需额外的说明和控制信息。
- 记录式文件: 记录式文件是一种结构文件。由若干个记录组成，文件中的记录可按顺序编号为记录1，记录2，……，记录n。
  - 如果文件中所有记录的长度相等，则称为定长记录文件，文件的长度为记录个数与记录长度的积。
  - 若文件中的记录长度不相等，则称为变长记录文件。文件长度为所有记录长度之和。

6.2.2 文件的逻辑结构

文件逻辑结构的类型

1．有结构文件

在记录式文件中，每个记录都用于描述实体集中的一个实体，各记录有着相同或不同数目的数据项。记录的长度可分为定长和不定长两类。

(1) 定长记录。这是指文件中所有记录的长度都是相同的，所有记录中的各数据项都处在记录中相同的位置，具有相同的顺序和长度。文件的长度用记录数目表示。对定长记录的处理方便、开销小，所以这是目前较常用的一种记录格式，被广泛用于数据处理中。

6.2.2 文件的逻辑结构

(2) 变长记录。这是指文件中各记录的长度不相同。产生变长记录的原因，可能是由于一个记录中所包含的数据项数目并不相同，如书的著作者、论文中的关键词等；也可能是数据项本身的长度不定，例如，病历记录中的病因、病史；科技情报记录中的摘要等。不论是哪一种，在处理前，每个记录的长度是可知的。

6.2.2 文件的逻辑结构

2．无结构文件

如果说大量的数据结构和数据库是采用有结构的文件形式的话，则大量的源程序、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。其长度以字节为单位。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。可以把流式文件看做是记录式文件的一个特例。在UNIX系统中，所有的文件都被看做是流式文件，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。

6.2.2 文件的逻辑结构

2.文件的存取方法

- 顺序存取：严格按文件信息单位排列的顺序依次存取。当打开文件时，文件的存取指针指向第一个信息单位，如第一个字节或第一个记录，每存取一个信息单位存取指针加1指向下一个信息单位，如此类推。
- 随机存取：也称直接存取，每次存取操作时必须先确定存取的位置。对流式文件或定长记录的文件比较容易确定存取位置。对不定长的记录式文件比较麻烦。解决的方法是建立索引。文件的索引可以作为文件的一部分，也可以单独建立索引文件。
- 按键存取：根据记录中的关键字的值进行存取。常用于记录式文件中。

访问文件中的信息例

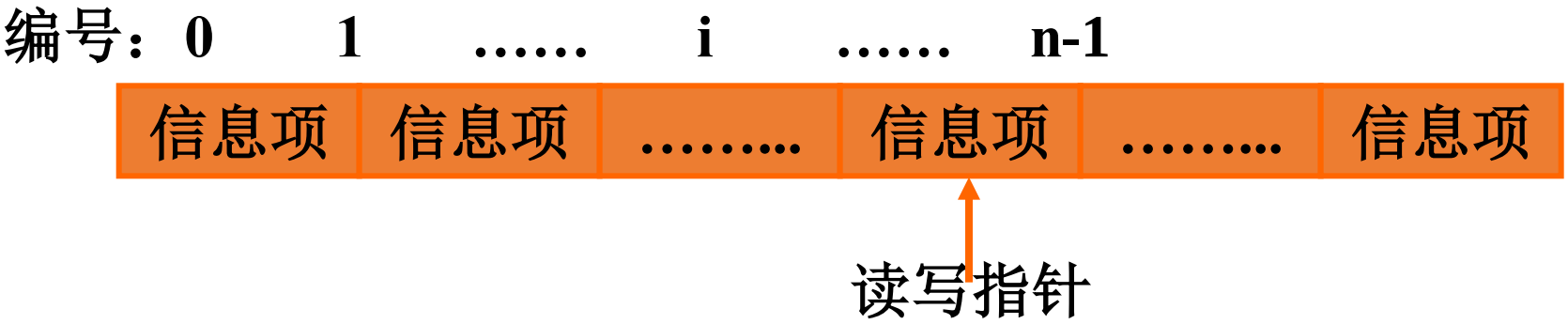
- Sequential Access（顺序存取）：
  - *read next*
  - *write next*
  - *reset*

适合顺序设备和直接设备

- Direct Access（直接存取）：
  - *read n*
  - *write n*
  - *position to n*
  - *read next*
  - *write next*

直接设备

n =记录号



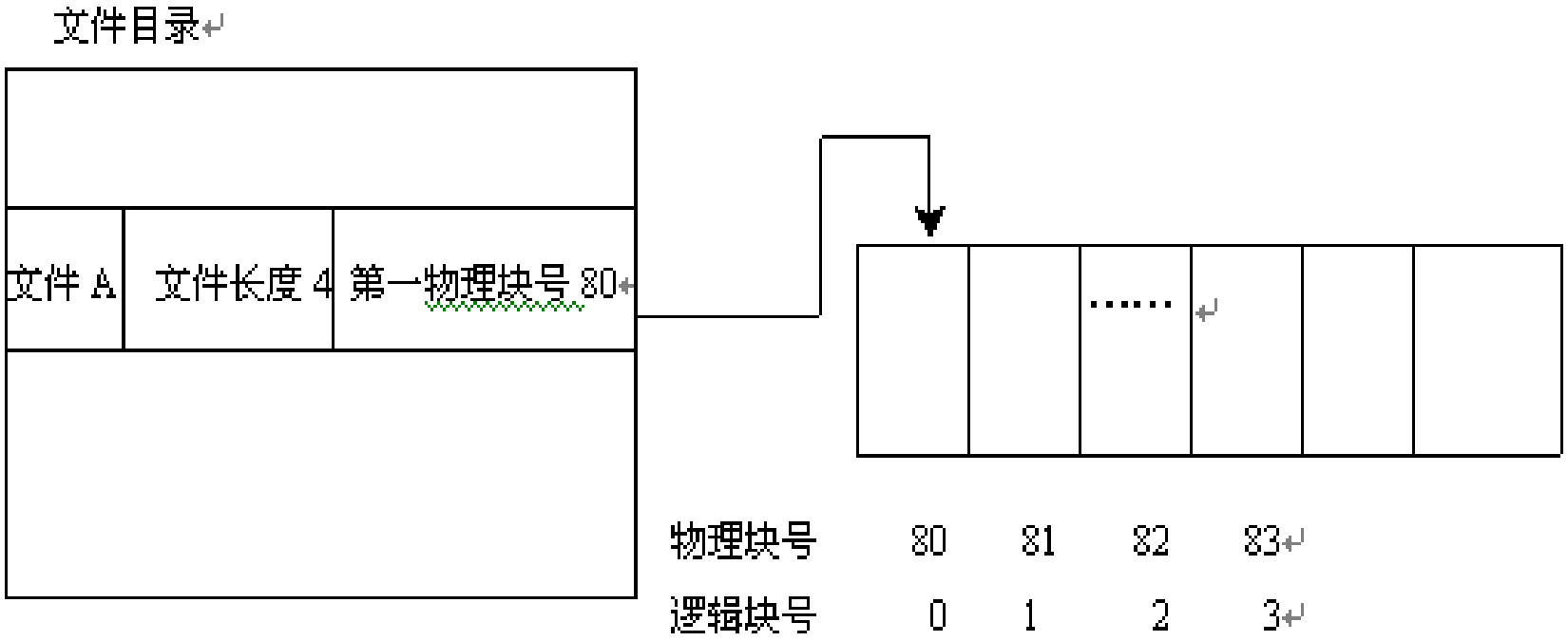
6.2.3 文件的物理结构（物理组织）

- 文件的物理结构是指文件在文件存储器中的存储方式。分为连续结构、串联结构、索引结构：

1.连续结构：文件的全部信息存放在外存的一片连续编号的物理块中，也称连续文件。

存放在磁带上的文件一般采用连续结构，即序号为i+1的物理块一定在i物理块之后。而存放在磁盘上的文件则可采用连续结构，也可采用别的结构。建立连续文件时要求用户给出文件的最大长度，以便系统为文件分配足够的存储空间，并在相应表格中登记文件的起始位置和长度。

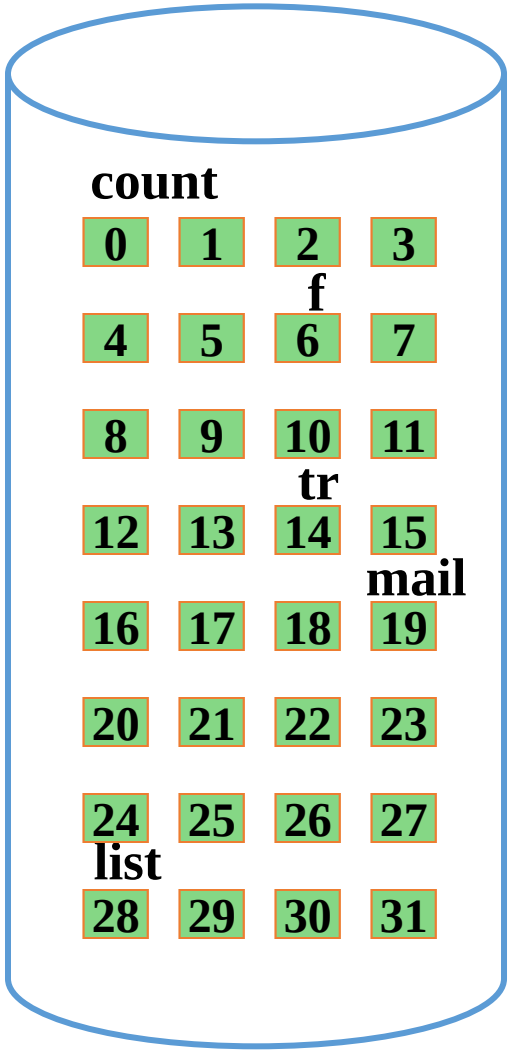
连续文件结构



优点：简单 ；支持顺序存取和随机存取； 顺序存取速度快，所需的磁盘寻道次数和寻道时间最少；

缺点：文件不易动态增长，预留空间浪费；不利于文件插入和删除；存在外部碎片问题；

顺序文件示例



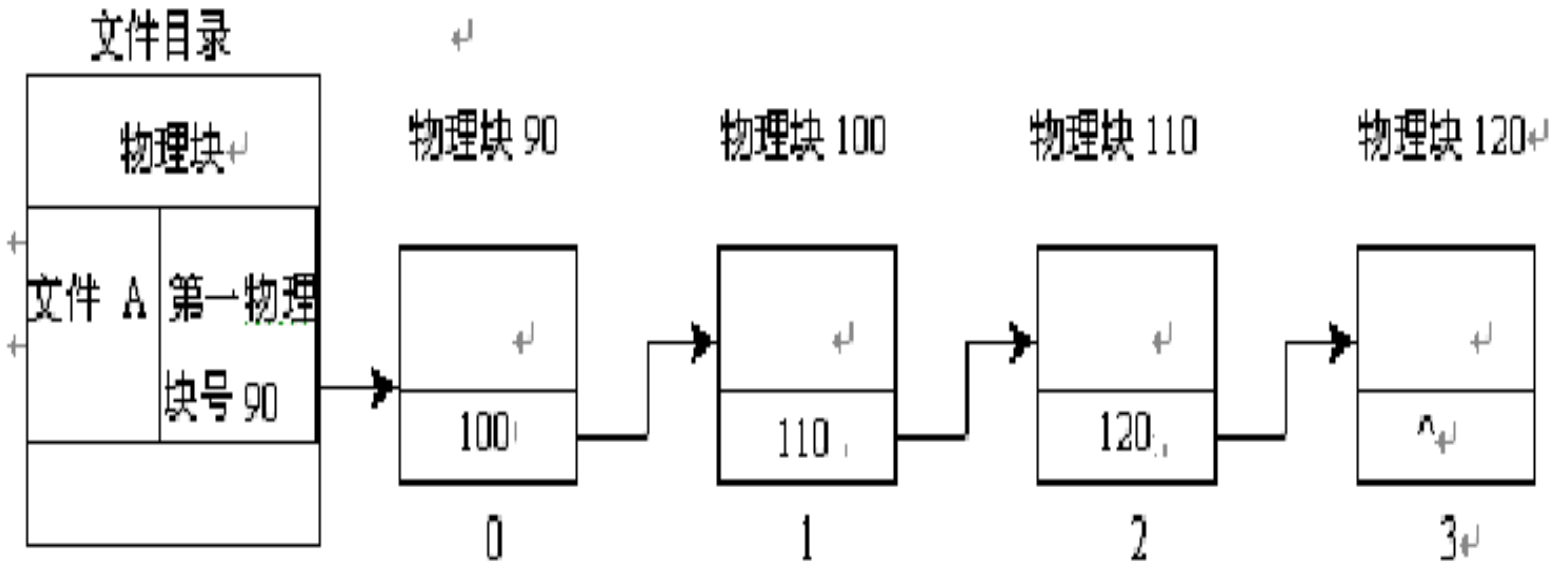
Directory

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

6.2.3 文件的物理结构

2.串联结构

- 也称链接结构。这是一种非连续的结构，存放文件信息的每一物理块中有一个指针，指向下一个物理块，这个指针的长度由物理设备的容量决定，通常放在该物理块的开头或结尾。



6.2.3 文件的物理结构

- 串联结构的文件适用于顺序存取。因为要获得某一块的块号，必须读取上一物理块，因此要随机地存取信息就较为困难。

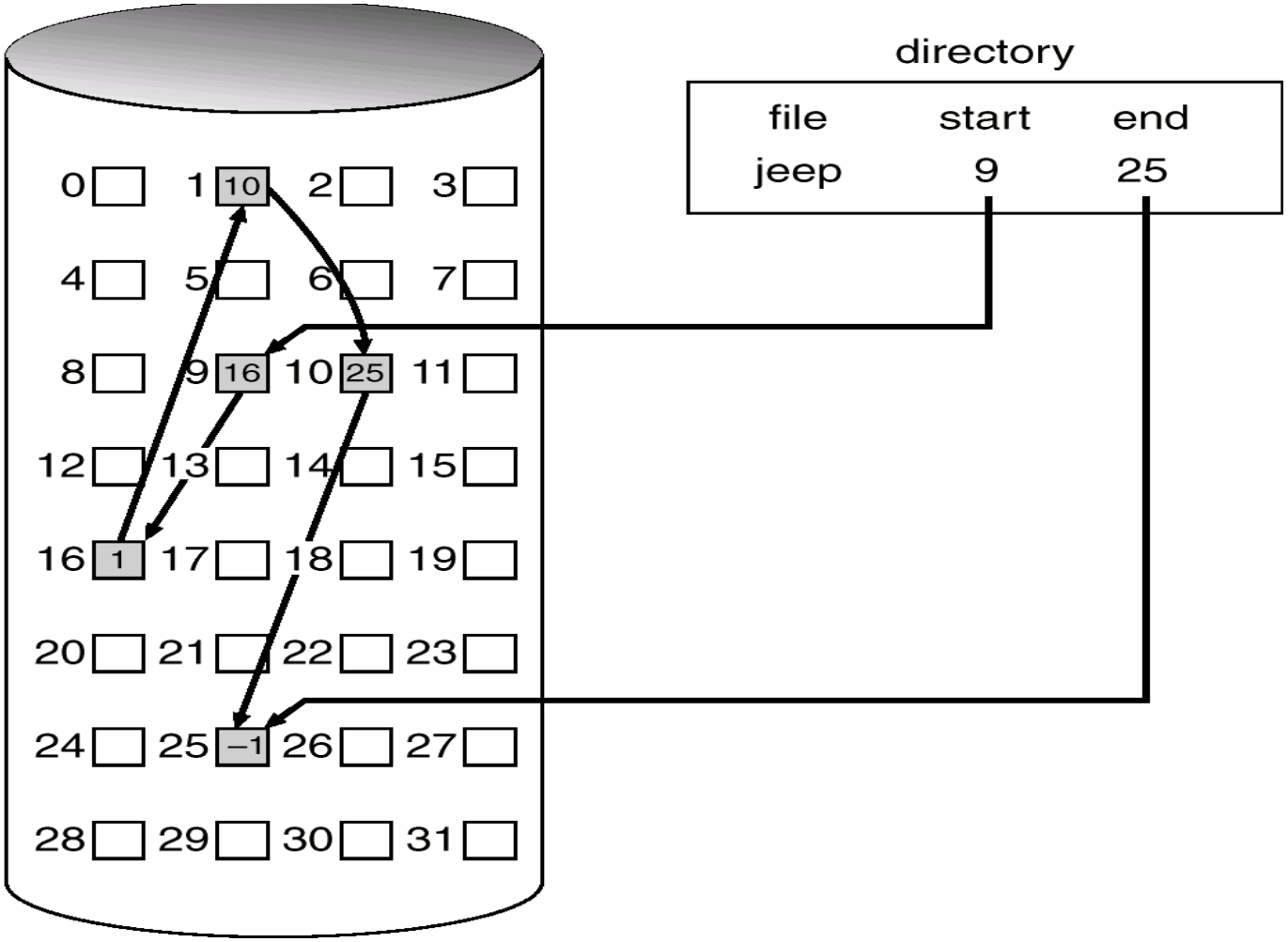
优点：

- 提高了磁盘空间利用率,不存在外部碎片问题
- 有利于文件插入和删除
- 有利于文件动态扩充

缺点：

- 存取速度慢，不适于随机存取
- 链接指针占用一定的空间
- 可靠性问题，如指针出错

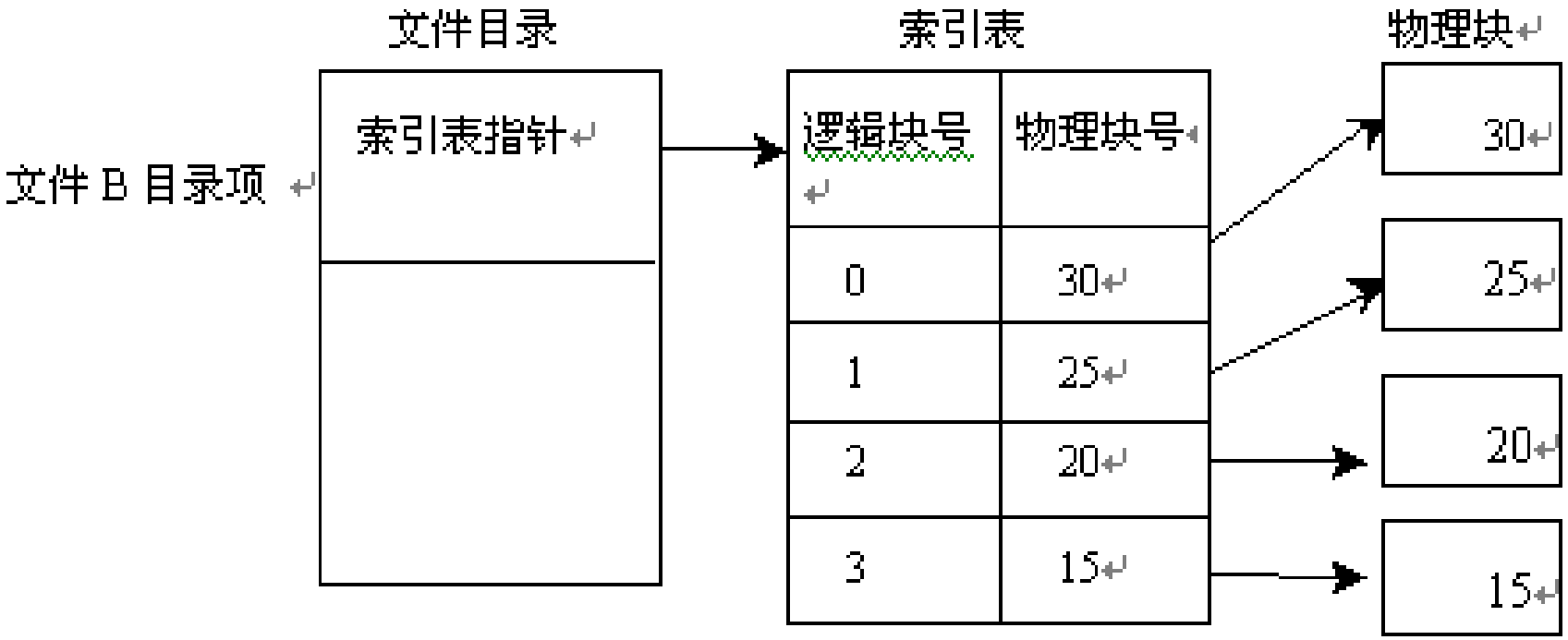
链接文件示例


directory
file
start
end
jeep
9
25
0
1
10
2
3
4
5
6
7
8
9
16
10
25
11
12
13
14
15
16
1
17
18
19
20
21
22
23
24
25
−1
26
27
28
29
30
31

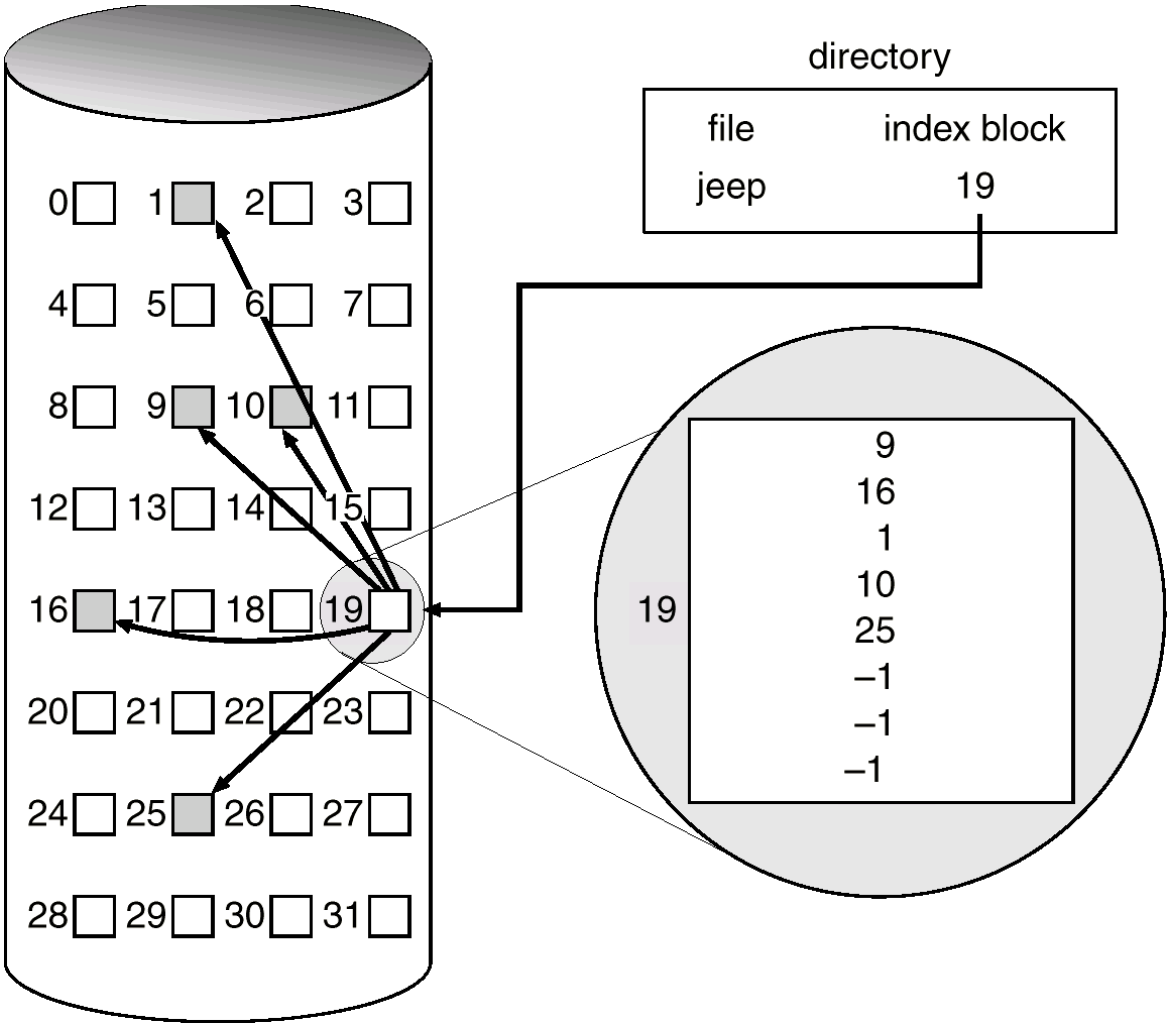
6.2.3 文件的物理结构

3.索引结构

一个文件的信息存放在若干不连续物理块中，系统为每个文件建立一个专用数据结构 - 索引表，并将这些块的块号存放在索引表中。一个索引表就是磁盘块地址数组,其中第i个条目指向文件的第i块。



索引文件示例


directory
file
index block
jeep
19
0 1 2 3
4 5 6 7
8 9 10 11
12 13 14 15
16 17 18 19
20 21 22 23
24 25 26 27
28 29 30 31
19
9
16
1
10
25
−1
−1
−1

6.2.3 文件的物理结构

优点：

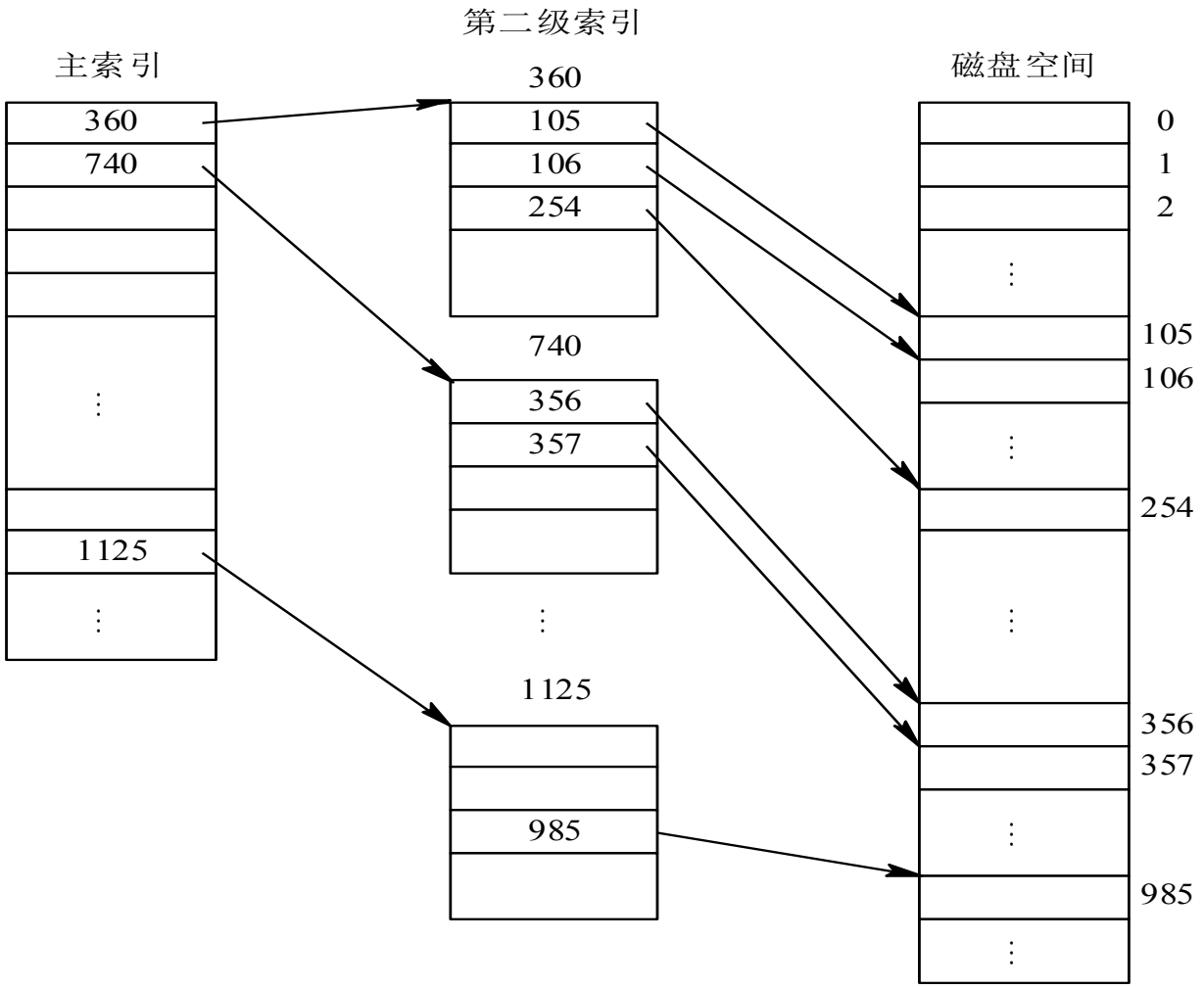
- 保持了链接结构的优点，又解决了其缺点，能顺序存取，又能随机存取；
- 满足了文件动态增长、插入删除的要求；
- 能充分利用外存空间；

缺点：

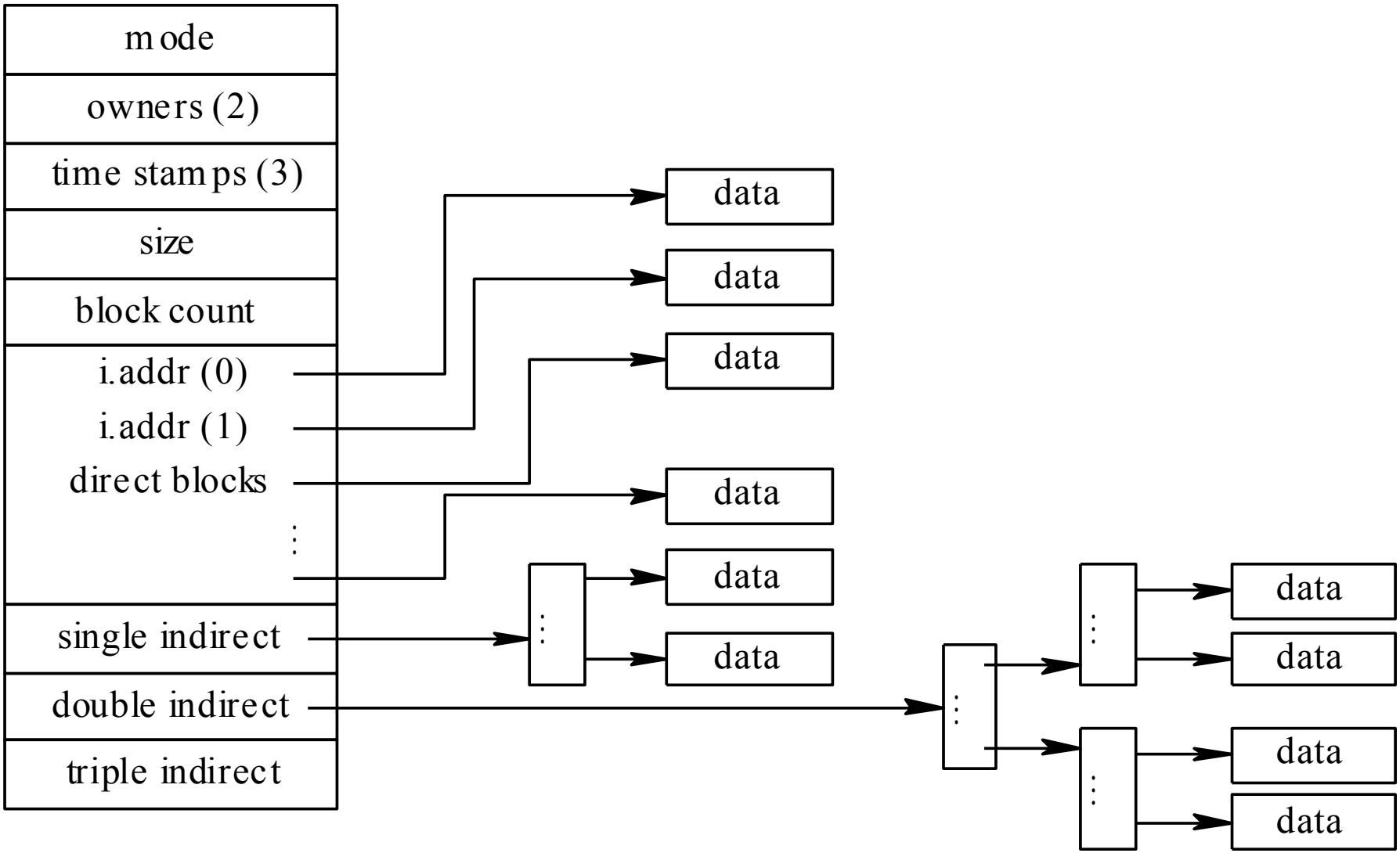
- 索引表本身带来了系统开销，如：内外存空间，存取时间开销等；

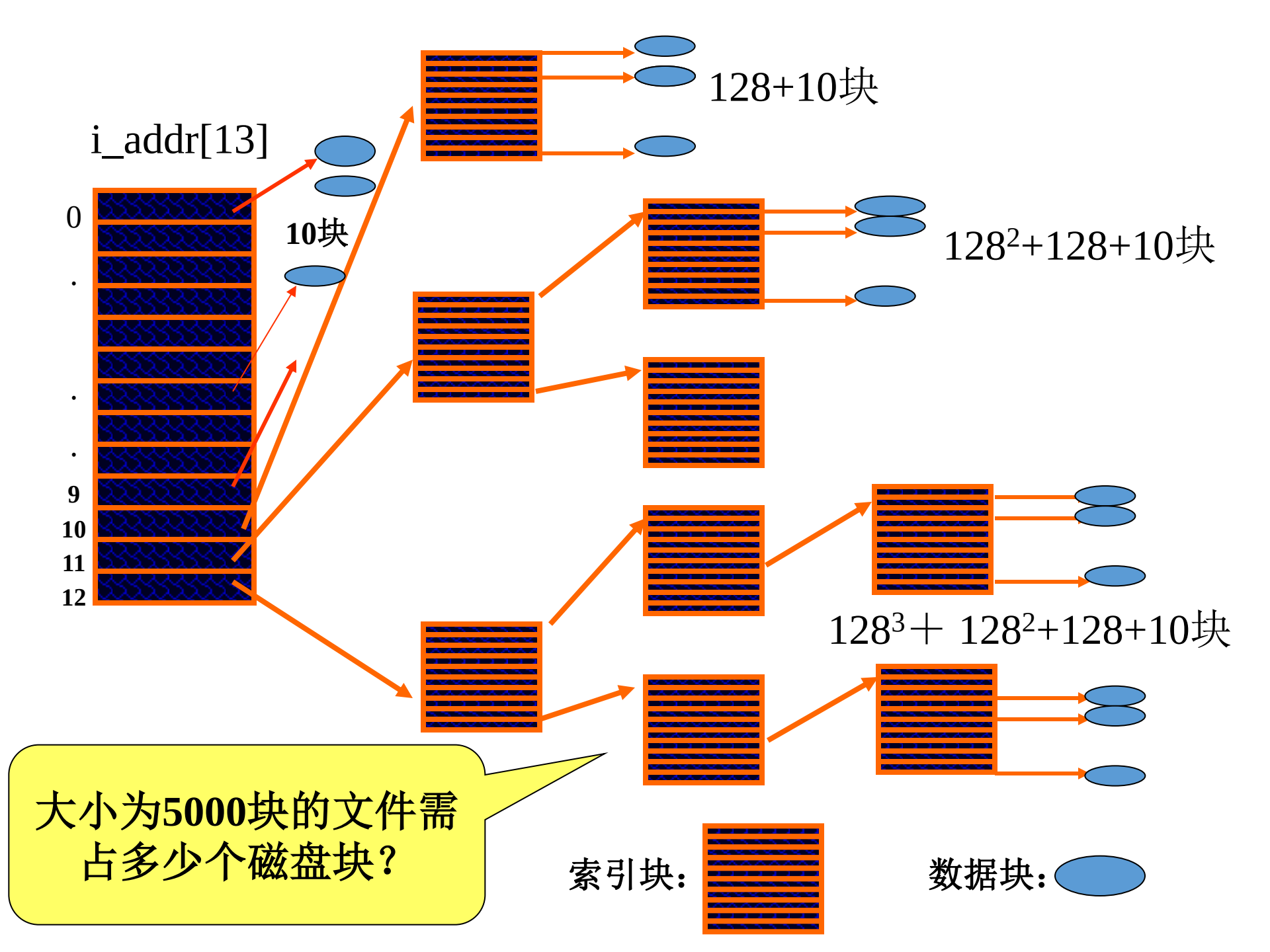
当索引表本身很大时，可考虑建立多级索引

多级索引


第二级索引
主索引
磁盘空间
360
740
1125
360
105
106
254
740
356
357
1125
985
0
1
2
105
106
254
356
357
985

4.混合索引结构（多重索引,unix采用）




i_addr[13]
0
.
.
.
9
10
11
12
10块
128+10块
128²+128+10块
128³＋ 128²+128+10块
大小为5000块的文件需
占多少个磁盘块？
索引块:
数据块:

6.2.3 文件的物理结构（物理组织）

(1) 直接地址

为了提高对文件的检索速度， 在索引节点中可设置10个直接地址项，即用i_addr(0)~i_addr(9)来存放直接地址。换言之，在这里的每项中所存放的是该文件数据的盘块的盘块号。假如每个盘块的大小为 4 KB，当文件不大于40 KB时，便可直接从索引结点中读出该文件的全部盘块号。

6.2.3 文件的物理结构（物理组织）

(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。为此，可再利用索引节点中的地址项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， 因而允许文件长达4 MB。

6.2.3 文件的物理结构（物理组织）

(3) 多次间接地址

当文件长度大于4 MB+40 KB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4 GB。 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4 TB。

6.2.4 文件的存储设备

1、存储设备类型

主要有磁带、磁盘、光盘、闪存（U盘）等。

其中，磁带属于顺序存取设备，只能存储连续文件；而磁盘、光盘等为直接存取设备，可存储任何格式的文件，对文件既能顺序存取，又能随机存取。

6.2.4 文件的存储设备

文件存储器

- 种类: 硬软磁盘/ 磁带/ 光盘/ 磁鼓/ 纸带/ 卡片/ 磁光盘/ 光盘库
- 外存性能指标:成本/容量/密度/速度/便携性/可写性
- 磁带：顺序设备、可拆卸、可读写、容量大，速度慢、作为脱机存档设备用
- 磁盘（硬盘）：直接存储设备、不可移动（可移动）、可读写、容量最大、速度快、作为联机设备用
- 光盘：直接顺序存储设备、可拆卸、可读（可写）、容量较大、作为脱机存档设备用

6.3 文件目录管理

对文件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。

6.3.1 文件控制块与文件目录

- 目录项(文件控制块，FCB ）：目录项是操作系统为管理文件而设置的数据结构，存放了为管理文件所需的所有有关信息。所有的目录项就构成了文件目录。包含信息如下：

（1）基本信息类

① 文件名 ； ② 文件物理位置 ； ③ 文件逻辑结构 ；④ 文件的物理结构

（2）存取控制信息类

（3） 使用信息类

MS-DOS中的目录项

文件名	扩展名	属性	备用	时间	日期	第一块号	盘块数

6.3.1 文件控制块与文件目录

文件名
文件号
文件主
文件类型
文件属性
共享说明
文件长度
文件地址
建立日期
最后修改日期
最后访问日期
口令
其它

FCB（File Control Block）

FCB创建：建立文件时

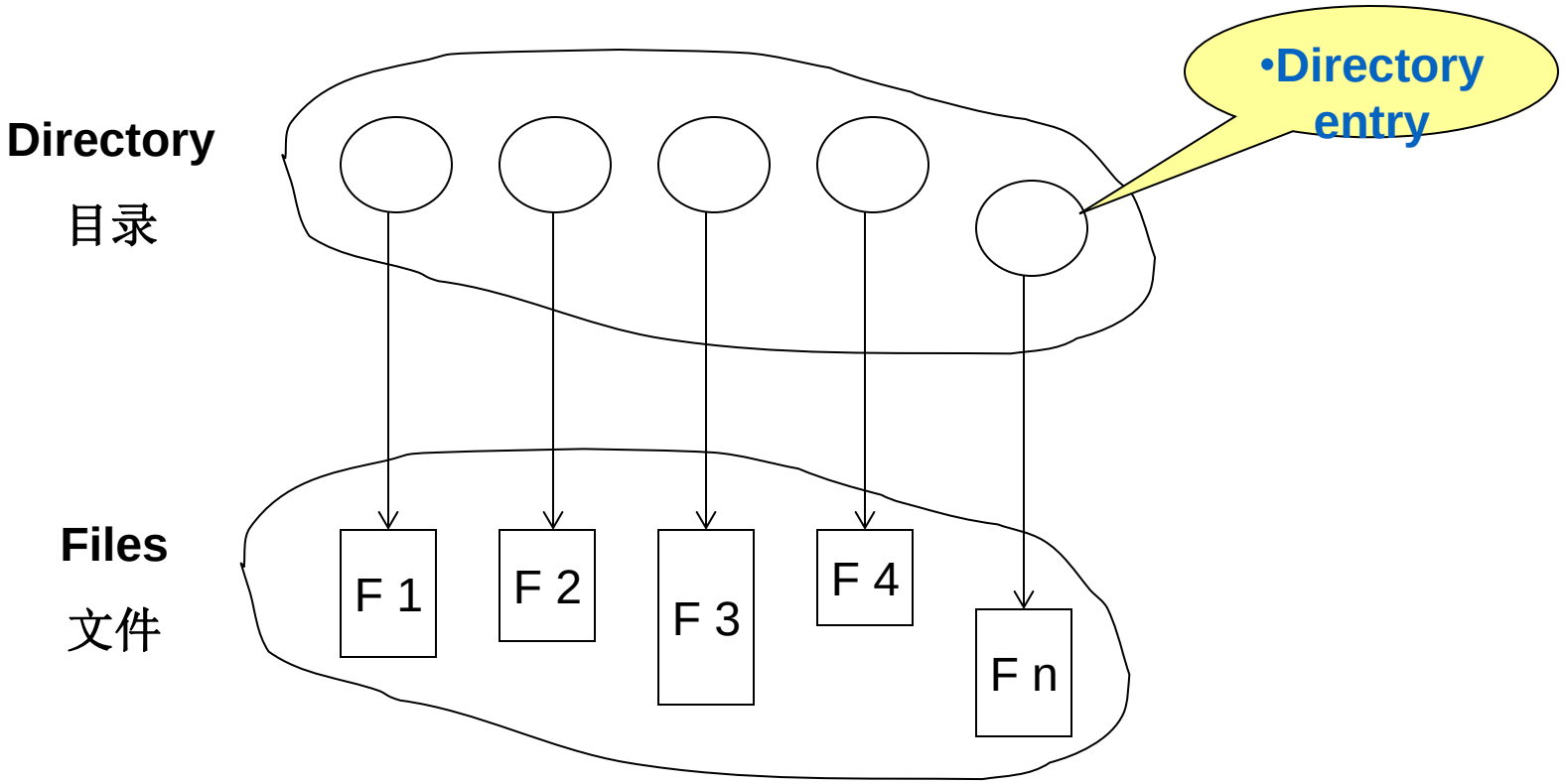
FCB撤消：删除文件时

FCB→内存：文件打开时

内存FCB写回外存：文件关闭时

6.3.1 文件控制块与文件目录

目录文件图例



UNIX/Linux的文件目录

1) 索引节点的引入

文件名	索引节点编号
文件1	10
文件2	11

目录文件

文件1 i节点

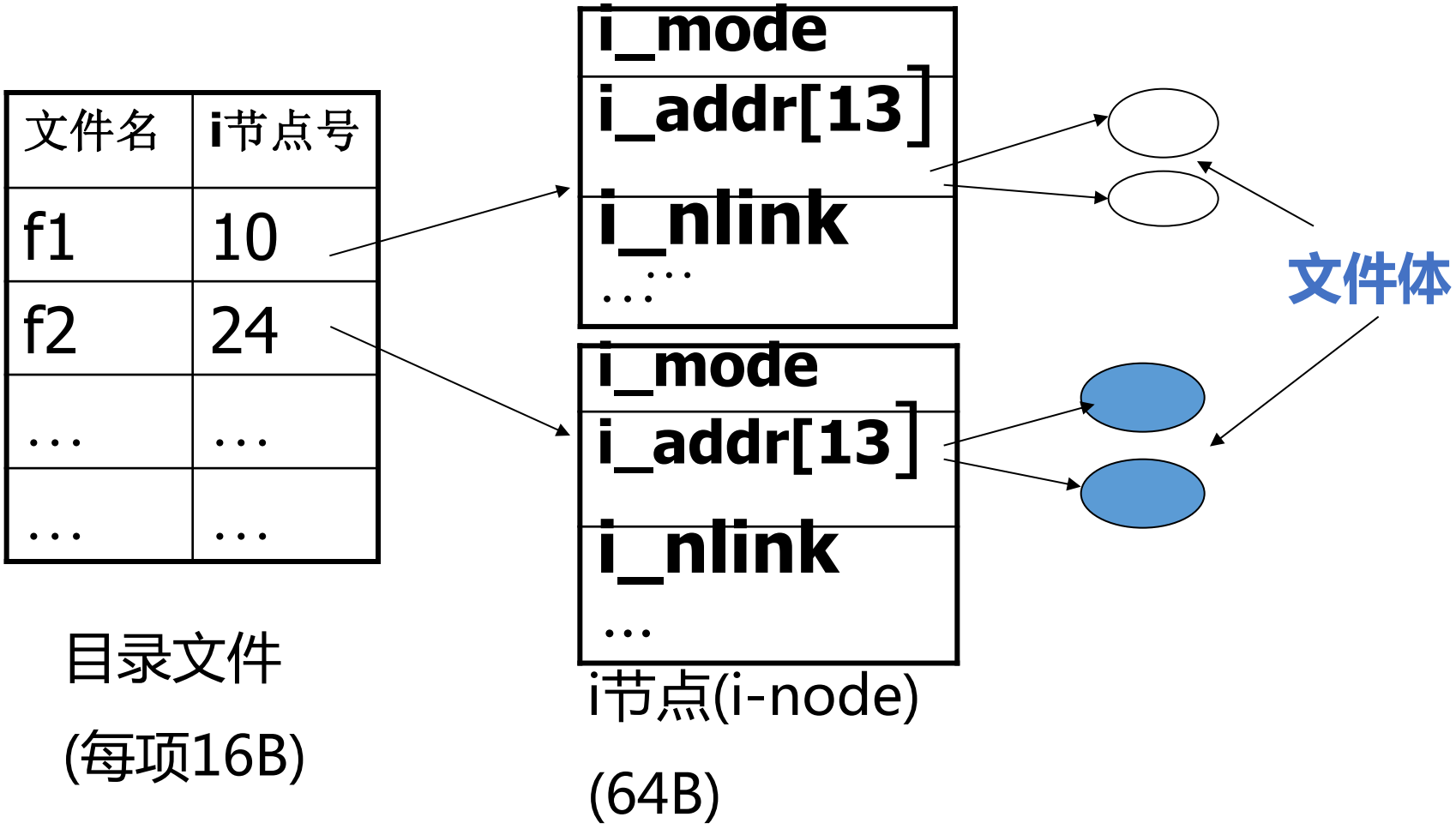
(1) 文件主标识符
(2) 文件类型
(3) 文件存取权限
(4) 文件物理地址
(5) 文件长度
(6) 文件连接计数
(7) 文件存取时间

文件2 i节点

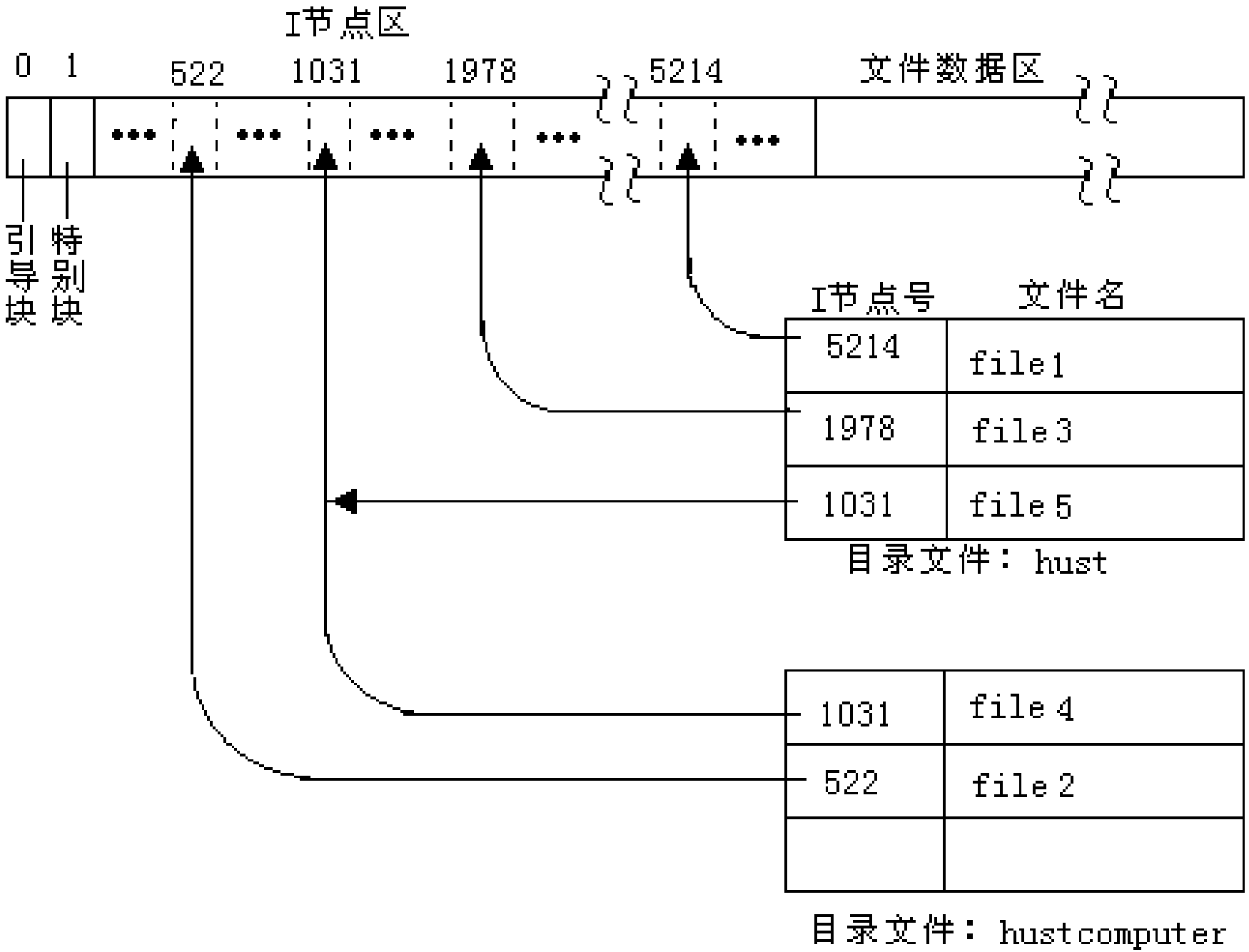
(1) 文件主标识符
(2) 文件类型
(3) 文件存取权限
(4) 文件物理地址
(5) 文件长度
(6) 文件连接计数
(7) 文件存取时间

一个FCB分两部分，真正的文件属性放在i_node中

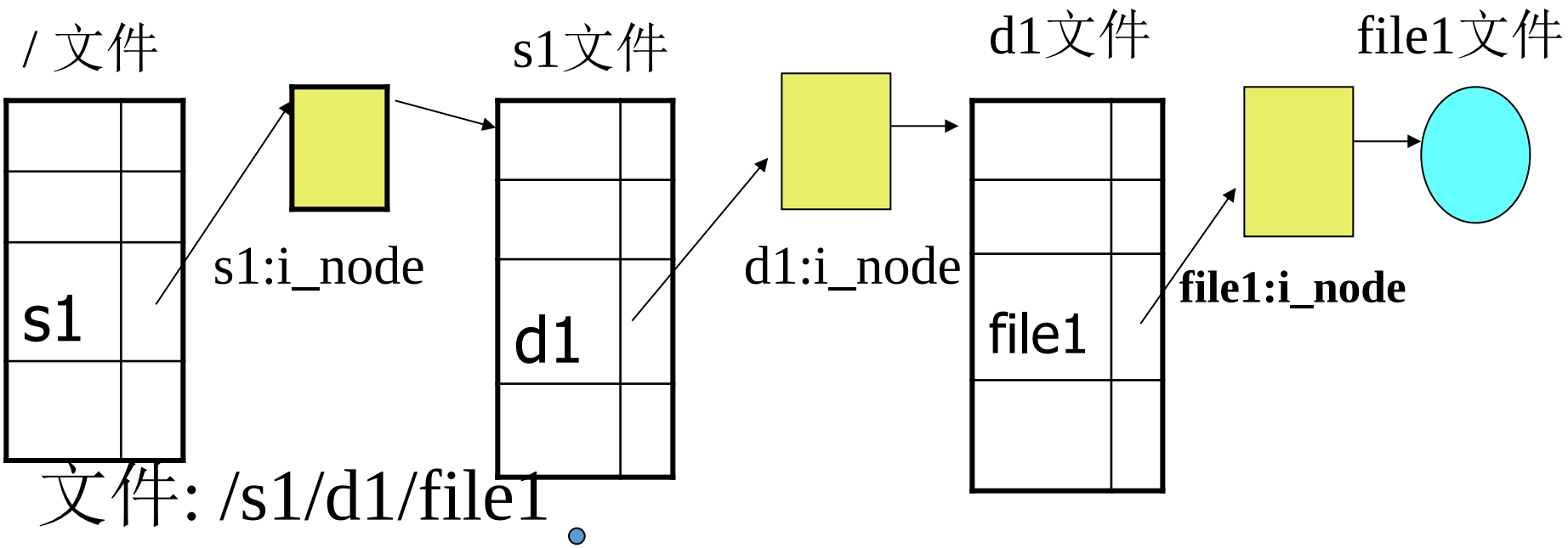
UNIX目录文件与索引节点（i节点）



一个FCB分两部分，真正的文件属性放在i_node中


I节点区
0
1
522
1031
1978
5214
文件数据区
引导块
特别块
I节点号
文件名
5214
file1
1978
file3
1031
file5
目录文件：hust
1031
file4
522
file2
目录文件：hustcomputer

unix目录文件与索引节点(续)



设根目录i节点在内存，file1为5块，最少需读磁盘多少次？最多呢？

6.3.2 目录结构

1. 单级目录结构

文件名	物理地址	文件说明	状态位
文件名1			
文件名2			

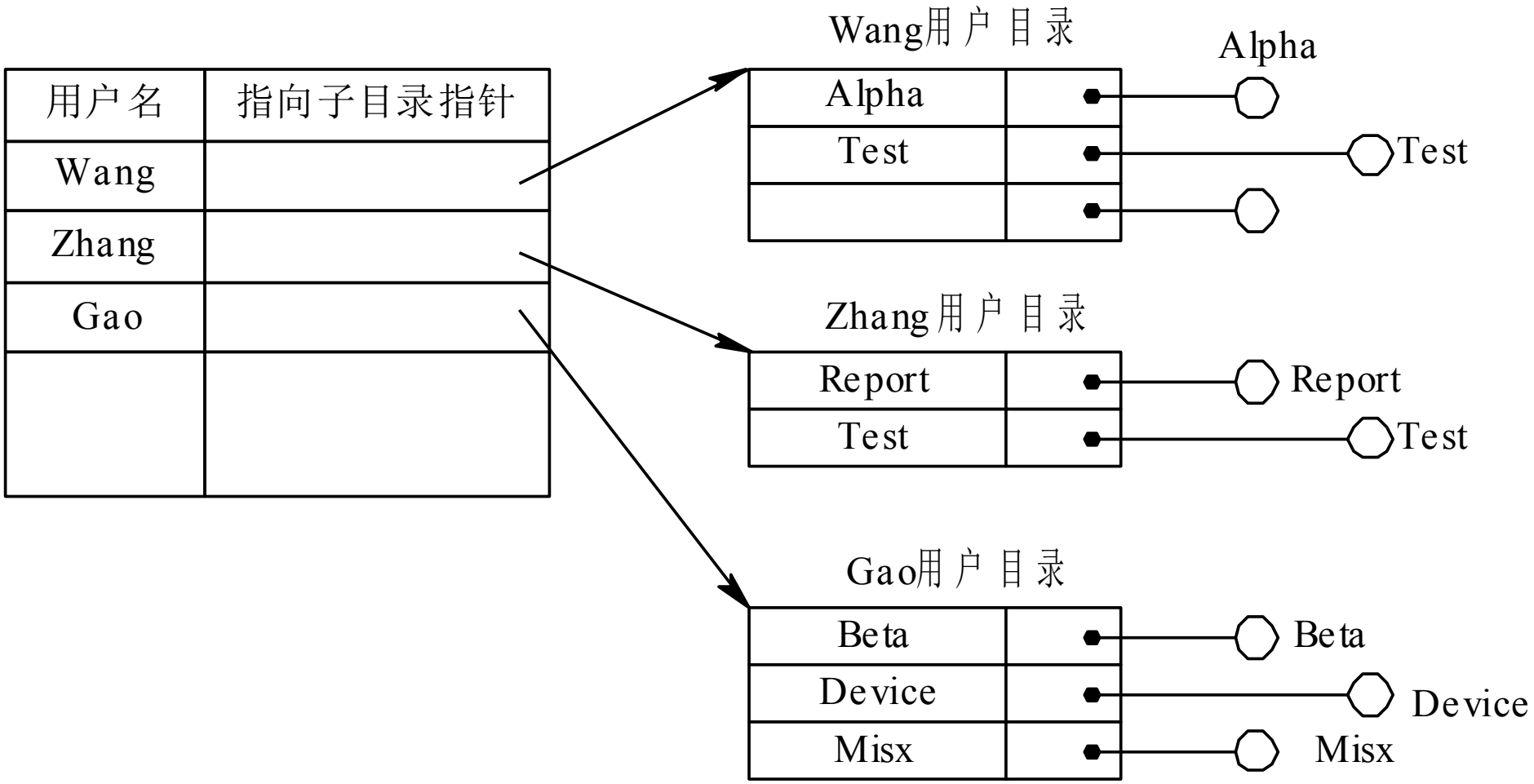
6.3.2 目录结构

2. 二级目录

- 二级文件目录结构把目录分成主目录和用户文件目录两级。
- 主目录由用户名和用户文件目录首地址组成，用户文件目录中登记相应的用户文件的目录项。

  - 优点：二级目录结构较为简单，也比较好地解决了重名的问题。
  - 缺点：缺乏灵活性，特别是不能反映现实世界中多层次的关系。

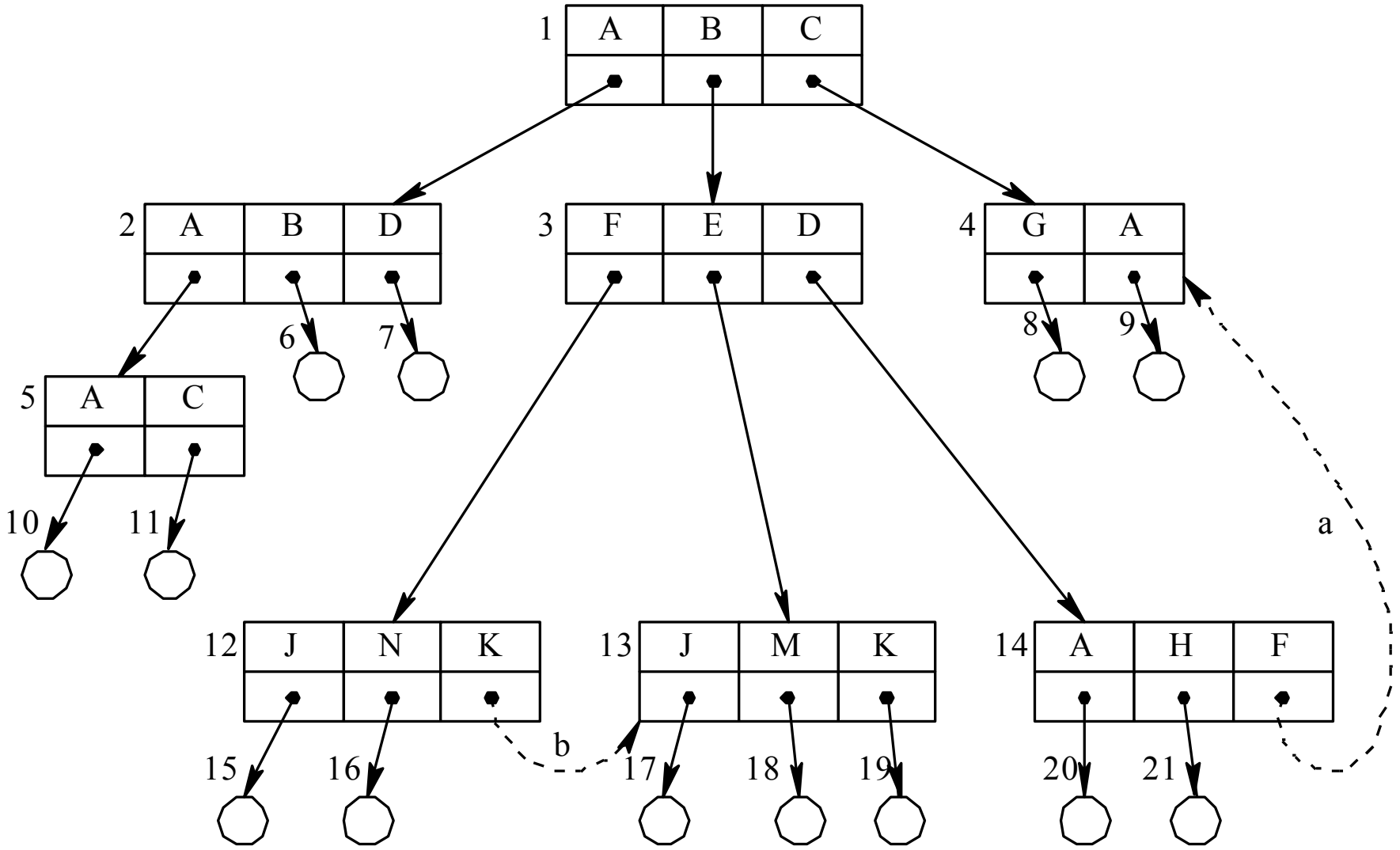
6.3.2 目录结构


用户名
指向子目录指针
Wang
Zhang
Gao
Wang用户目录
Alpha
Test
Alpha
Test
Zhang用户目录
Report
Test
Report
Test
Gao用户目录
Beta
Device
Misx
Beta
Device
Misx

6.3.2 目录结构

3. 多级目录结构

- 多级目录结构由根目录和各级目录组成，为管理上的方便，除根目录外，其它各级目录均以文件的形式组成目录文件。
- 根目录中的每个目录项可以对应一个目录文件，也可以对应一个数据文件，同样目录文件中的每个目录项可以对应一个目录文件。也可以对应一个数据文件。如此类推，就形成多级目录结构。
- 也称树形目录结构
- 绝对路径：从根目录开始给出文件名，如:/dir1/dir2/file1
- 相对路径：从当前目录开始给出文件名，如：dir2/file1


1
A
B
C
2
A
B
D
3
F
E
D
4
G
A
5
A
C
6
7
8
9
10
11
12
J
N
K
13
J
M
K
14
A
H
F
15
16
17
18
19
20
21
a
b

6.4 文件存储空间管理

- 分配与回收：算法 + 数据结构
  算法：时间、空间开销
  数据结构：空间开销

- 原因：外存海量空间，而管理的过程在内存进行，
  要考虑表格存放在内存时的开销。

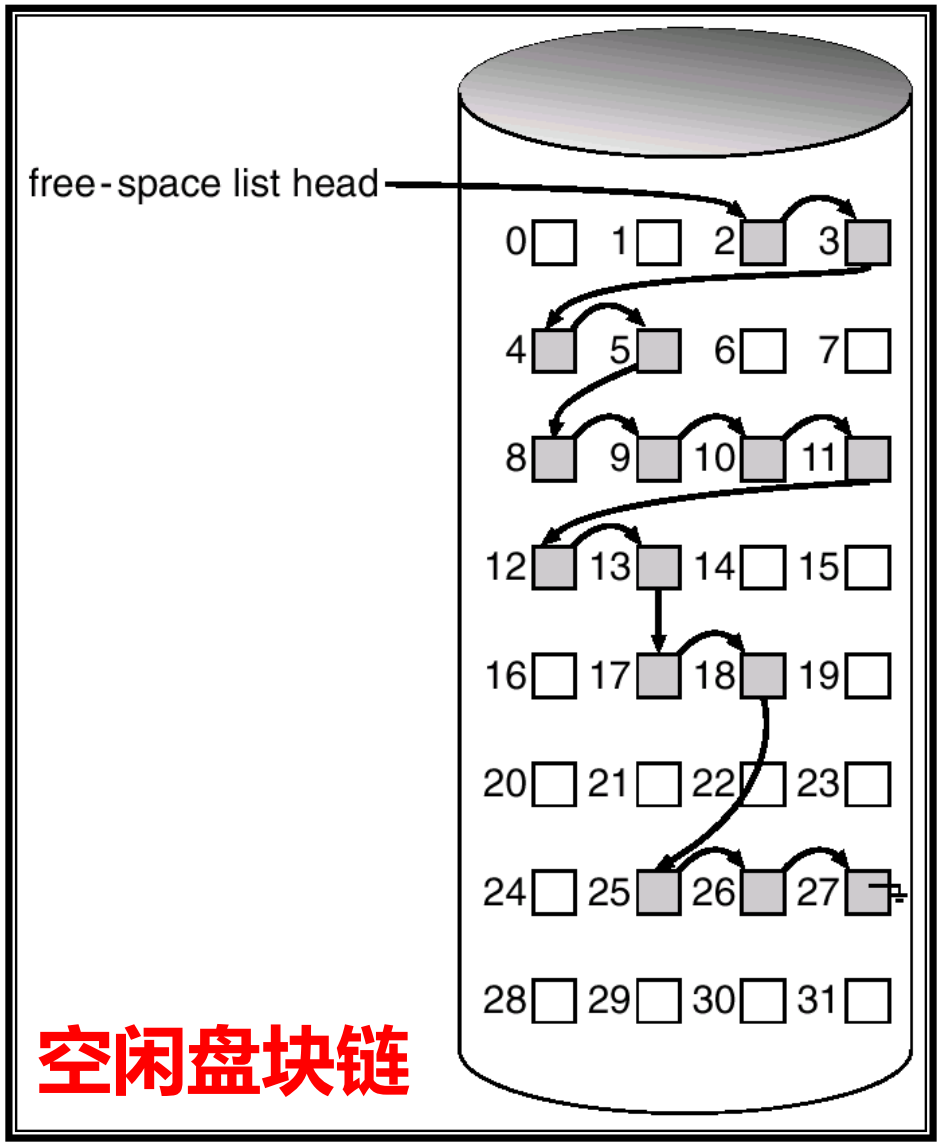
6.4.1 空闲表法

类似内存的动态分区分配：每个文件分配一块连续的存储空间，外存的所有空闲区建立一张空闲表

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

6.4.2 空闲链表法

- 用链表来表示磁盘的空闲空间列表：空闲盘块链和空闲盘区链
- 空闲盘块链
  - 每一个空闲物理块上都有一个指针，把所有的空闲块通过该指针链接起来，形成一个链表，文件系统只需记住该链表的首结点指针；
  - 当需要分配一个空闲物理块给某个文件时，即摘下链表的首结点
- 空闲盘区链：把所有的空闲盘区（可包含多个盘块）链接起来

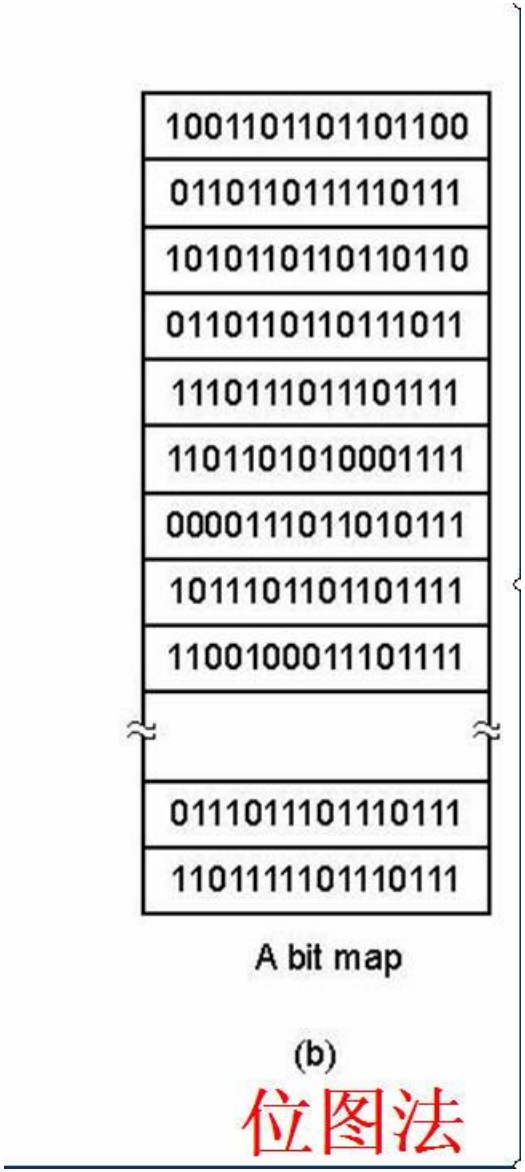


6.4.3 位示图

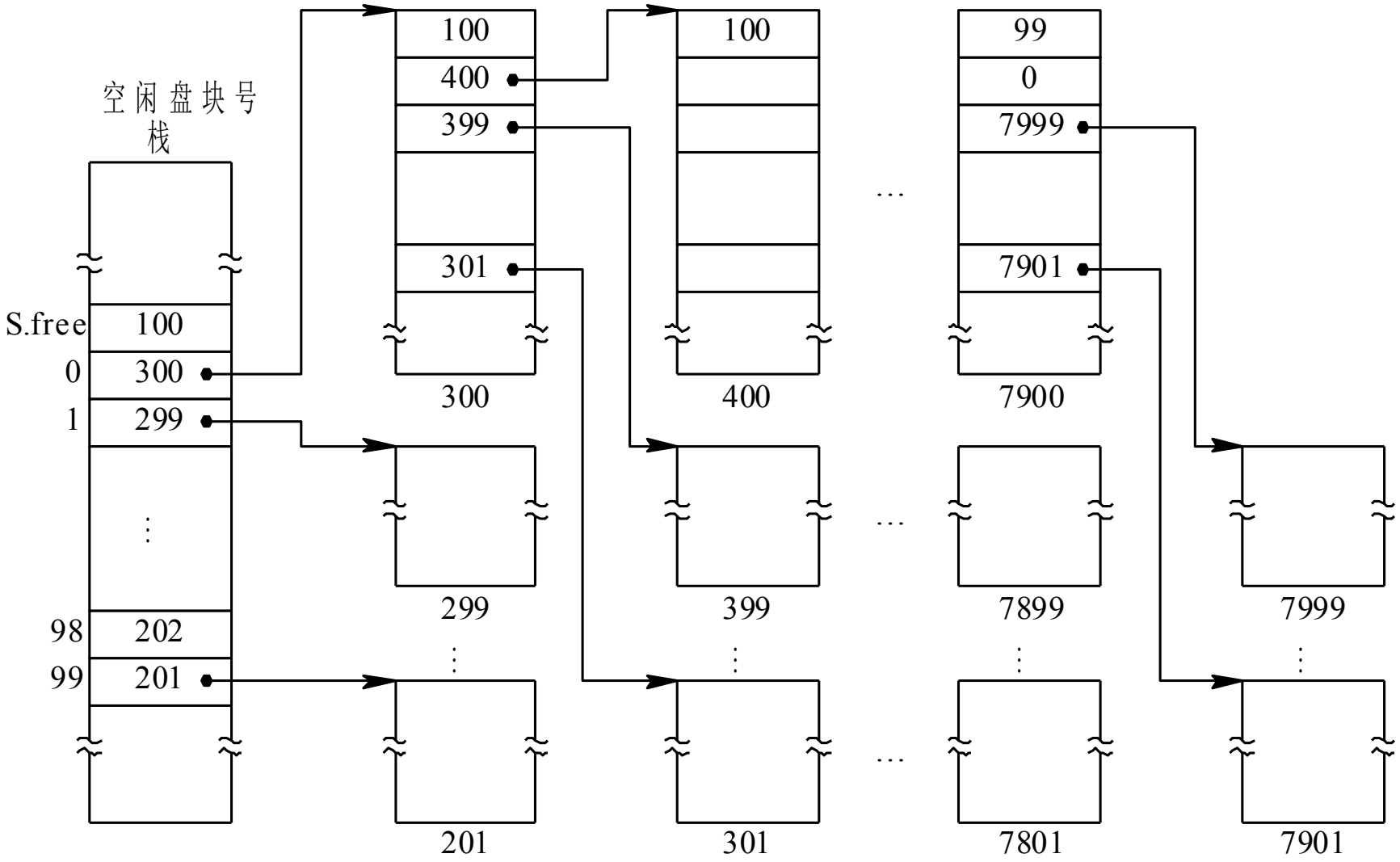
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	0	0	0	1	1	1	0	0	1	0	0	1	1	0
2	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1
3	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0
4																
⋮																
16																

6.4.3 位示图

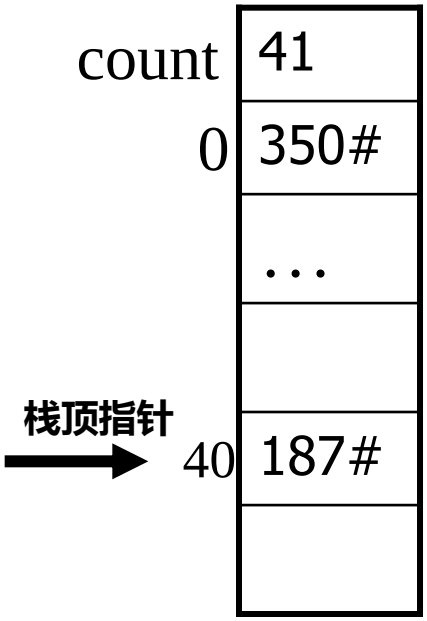
- 位示图本身存放在磁盘上，假设磁盘大小16G，物理块大小1KB，则需要$2^{34}/2^{10}=2^{24}$位，即位示图本身占2M字节、2048个物理块；
- 占据空间很大，因而适合小磁盘。
- 优点：分配给文件的物理块比较连续。



6.4.4 成组链接法（UNIX)



空闲块成组链接法分配与回收



内存分配一个空白块栈，栈顶指针p指向最后一块

- **分配算法：**

```
If ( (*p)==0) return error;
if(count==1)
   {将0号元素指定的盘块读到栈中；
    初始化栈顶指针p;重新执行该语句；}
else {分配p所指向的磁盘块；
        p --;
        count--; }
```

- **回收算法：设回收块号为n#**

```
if (count==50)
    {将空白快栈写到磁盘快n#中；
     将n#块填写到0号元素中；
      count＝1；
      p＝0；}
else {p=p+1;
      将n#块地址写到p所指定的单元中；
      count=count+1; }
```

文件的共享

- 文件共享分为外存（静态）共享和内存（动态）共享
- 外存共享（用户）：
  - 同一名字共享：通过权限表（ACL）实现
  - 不同名字共享：符号链接计数（unix：i_link)

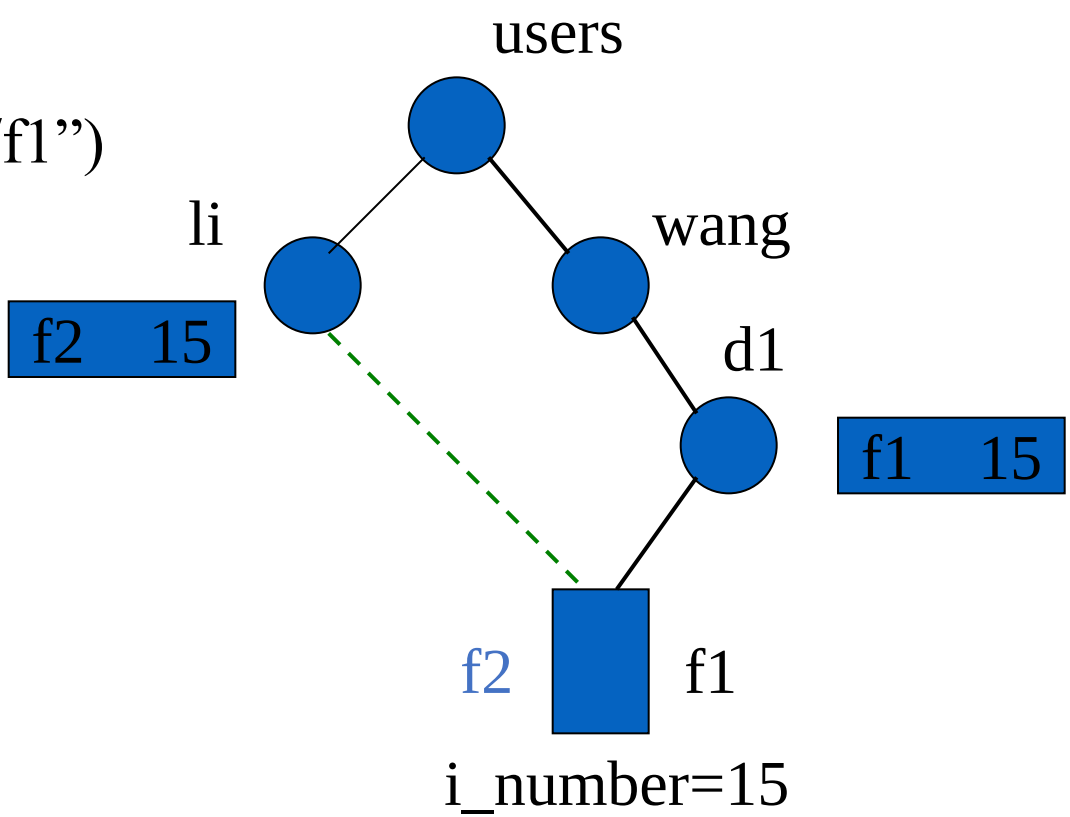
- 内存共享（进程）：
  - 共享内存中的文件FCB(unix: i_node中的i_count)
  - 共享文件的读写指针（unix: f_count ，f_offset)

文件的外存共享例（不同名字）

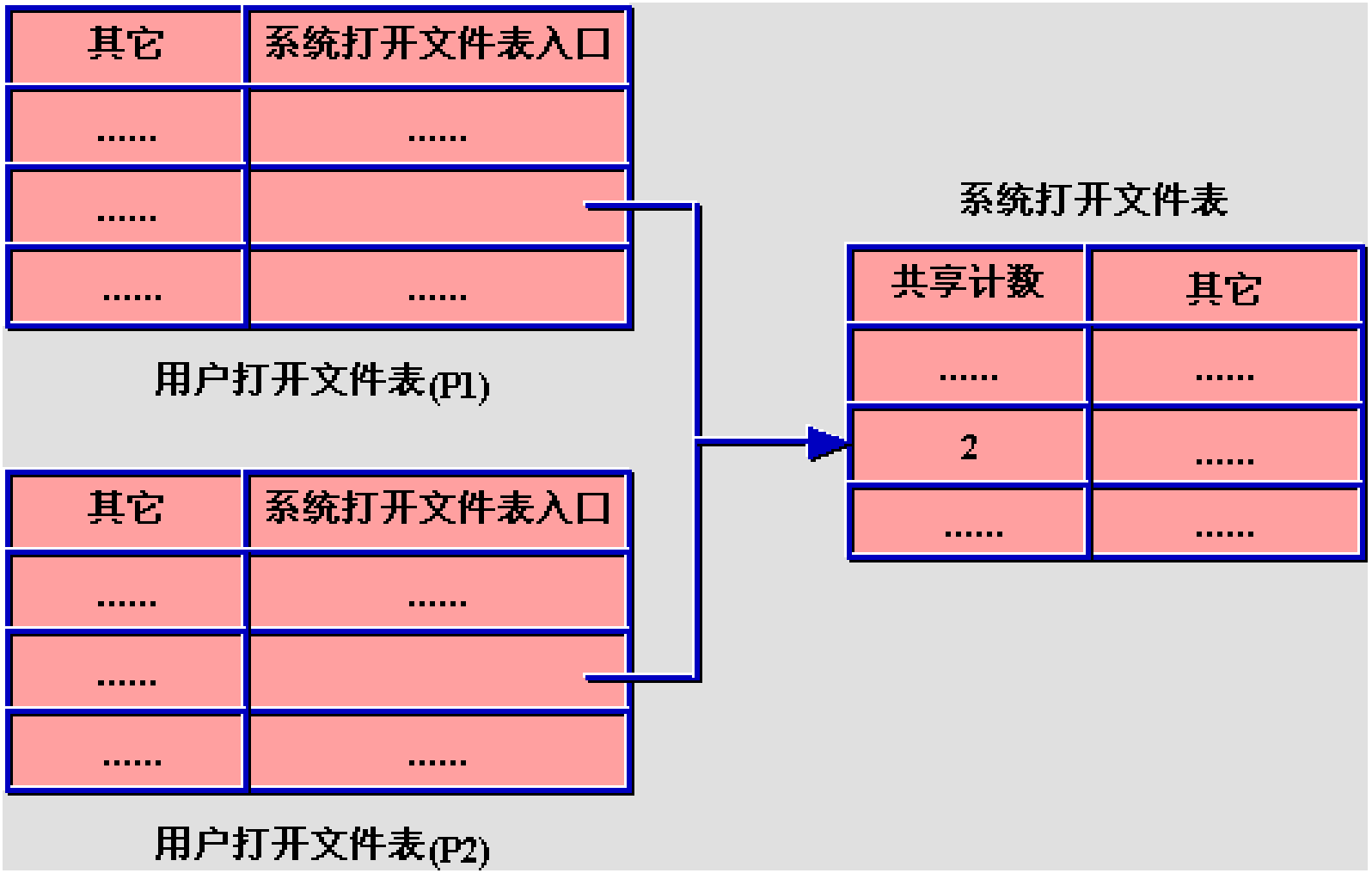
link(“/usr/users/wang/d1/f1”,

“/usr/users/li/f2”)

unlink(“/usr/users/wang/d1/f1”)



文件的外存共享例（不同名字）



文件的保护、保密与安全

- 保护
  - 防止用户对文件进行非授权的访问
- 保密
  - 防止文件内容泄露
- 安全
  - 防止文件被破坏
  - 自然因素
  - 人为因素

6.5.2 文件的存取控制

1、存取控制矩阵

文件 / 用户	ALPHA	BETA	REPORT	SQRT		
张三	RWX	---	R-X	---		
李四	R-X	---	RWX	R-X	…	
王五	---	RWX	R-X	R-X		
赵六	---	---	---	RWX		
.	.					

存取控制矩阵

存取控制矩阵在概念上是简单清楚的，但实现上却有困难。当一个系统用户数和文件数很大时，二维矩阵要占很大的存储空间，验证过程也费时。

6.5.2 文件的存取控制

2、存取控制表

- 存取控制矩阵由于太大而往往无法实现。一个改进的办法是按用户对文件的访问权限的差别对用户进行分类，由于某一文件往往只与少数几个用户有关，所以这种分类方法可使存取控制表大为简化。它把用户分成三类：文件主、同组用户和其它用户，每类用户的存取权限为可读、可写、可执行以及它们的组合。
  - 每个文件一张表：列出所有用户对该文件的访问权限。
  - 存放在文件的FCB中
  - 表的空间开销小，检索快
  - 大部分系统采用这种方式

访问权限控制表例（UNIX）

- 在UNIX/LINUX系统中，用ls长列表显示时每组存取权限用三个字母RWX表示，如读、写和执行中那一样存取不允许则用“-”字符表示，用ls -l长列表显示ls文件如下：

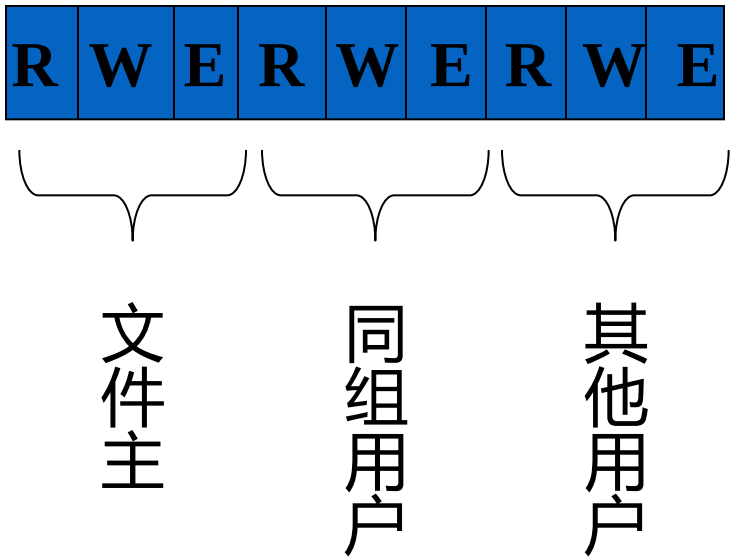
-r-xr-xr-t 1 bin bin 43296 May 13 2005 /opt/K/SCO/Unix/5.0.4Eb/bin/ls

显示前2-10共9个字符表示文件的存取权限，每3个字符为一组，分别表示文件主、同组用户和其它用户的存取权限。也可用十进制数来表示权限如下：

文件名：FILE1			
用户组	文件主	同组用户	其他用户
访问权限	7	4	4

访问权限控制表例（UNIX）i_mode

9个二进制位：



- 存放在i-node中，成员名： i_mode
- 文件主判别：访问进程u_uid=i_uid
- 同组用户判别：访问进程u_gid=i_gid
- i_mode在创建文件时给出，creat(filename, mode)
- 其后文件主可以修改：chmod(filename, new_mode)

6.5.3 文件的保密

1、 口令

使用口令的优点是：简便，节省空间。 其缺点有以下几点：

- 可靠性差。口令易被窃取。
- 存取控制不易改变。
- 保护级别少。

2、密码

密码技术除保密性强外，还具有节省存储空间的优点。但它必须花费大量的编码和译码时间，从而增加了系统的开销。

文件主要操作例程

- 建立文件：申请一个目录项、填表
- 打开文件：将文件的FCB拷贝到内存活动文件表中
- 读文件：将文件内容读到内存
- 写文件：将内存内容写到文件中
- 关闭文件：删除内存中文件的FCB
- 删除文件：删除文件目录项，回收文件所占空间
- 文件定位：修改文件指针
- ……

文件系统小结

1.概念：文件、文件系统、目录文件、目录项(FCB)、
i_node、活动i_node、空白块栈

2.文件系统功能、Unix文件分类

3.文件的逻辑结构，顺序、索引、索引顺序

4. 文件FCB内容、unix目录结构、目录文件与i_node关系、

5. 文件的物理结构及特点、unix多重索引结构的实现

6. 文件的保护、保密、安全措施

7. 文件的外存共享、内存共享

8. 文件存储器管理，unix空白块成组链接法的实现

9. 文件操作

作业

1. UNIX文件系统中有一个文件为：“/s1/s2/s3/f1”,设根目录的i_node在内存，打开该文件，至少访盘几次？若s3是当前目录（即其i_node在内存）则打开该文件，至少访盘几次？

2. UNIX文件系统的磁盘块大小是512字节。物理结构采用多重索引结构。设每个磁盘块物理地址用5个字节表示，则系统要为一长度为3600KB文件分配多少个磁盘块？

作业

3. 就以下几个方面，对文件的物理结构进行比较

	访问效率/碎片	随机访问	插入/删除	空间开销
顺序结构				
链接结构				
索引结构				