

# 操作系统原理

## 第9章 文件系统

华中科技大学计算机学院 谢美意

# 文件系统概述

# 1. 文件

## (1) 什么是文件

文件是在逻辑上具有完整意义的信息集合，它有一个名字以供标识，文件名是有若干约束的字符串。

**特点：**可以长期保存、可以在进程间共享、具有内部结构。

## (2) 构成文件的基本单位

信息项（单个字节或字符）、记录

## (3) 文件的其它描述

- ① 文件是具有符号名的信息(数据)项的集合
- ② 文件是具有符号名的记录的集合

## (4) 文件分类

- ① 按文件的性质和用途分类：系统文件、程序库文件、用户文件
- ② 按文件保护级别分类：不保护文件、执行文件、只读文件、读写文件
- ③ 按文件流向分类：输入文件、输出文件、输入输出文件

## (5) 文件名与属性

- ① 文件名：每个文件有一个给定的名字，包括文件符号名和内部标识符。
- ② 文件扩展名：表示文件的使用特征，如：.c .obj .lib 等。
- ③ 文件属性：表示文件类别、保护级等信息。如：
  - 文件大小、文件所有者、文件创建时间、最后修改时间
  - Linux的文件权限属性：rwxrwxrwx 0666的含义？

## 2. 文件系统

### (1) 什么是文件系统

文件系统是操作系统中负责管理和存取文件信息的软件机构。

### (2) 文件系统的组成

- ① 管理文件所需的数据结构：如目录表、文件控制块、存储分配表
- ② 管理程序
- ③ 一组操作

### (3) 文件系统的功能

- ① 从用户角度看 —— 文件系统实现了“按名存取”的功能。
- ② 从系统角度看 —— 文件系统提供的功能包括：**辅存空间管理、构造文件结构、文件共享、存取文件的方法、文件保护、一组文件操作命令。**

## 2. 文件系统

### (4) 文件系统的特点

#### ① 使用简单

使用文件名、一组文件操作命令。

#### ② 安全可靠

提供防护措施，在文件遭受破坏时，能及时修复。

全量备份、增量备份、动态备份、远程备份

#### ③ 既能共享，又能保密

身份验证、存取权限验证。



## 3. 文件组织的两种结构

### (1) 文件的逻辑结构

- 从用户角度看到的文件面貌，即用户对信息进行逻辑组织形成的文件结构。
- 为用户提供一种逻辑结构清晰、使用简便的逻辑文件形式。
- 用户按文件的逻辑结构形式去存储、检索和加工文件中的信息。

### (2) 文件的物理结构

- 信息在物理存储器上的存储方式，是数据的物理表示和组织。
- 选择工作性能良好、设备利用率高的物理文件形式。
- 系统按照文件的物理结构形式和外部设备打交道，控制信息的传输。

## 3. 文件组织的两种结构

### (3) 逻辑记录与物理记录 (磁盘块)

#### ① 逻辑记录

文件中按信息在逻辑上的独立含义来划分的信息单位，逻辑记录是对文件进行存取操作的基本单位。

#### ② 物理记录

在存储介质上，由连续信息所组成的一个区域称为块，也叫物理记录。

#### ③ 逻辑记录与物理记录的区别与联系

- 一个是逻辑的概念，一个是物理的概念。
- 逻辑记录最终要存放到物理记录上。



# 文件的逻辑结构与存取方法

# 1. 文件的逻辑结构

## (1) 流式文件

- ① **定义**：流式文件是相关的有序字符的集合，是无结构的。
- ② **存取方法**：流式文件是按信息的个数或以特殊字符为界进行存取的。

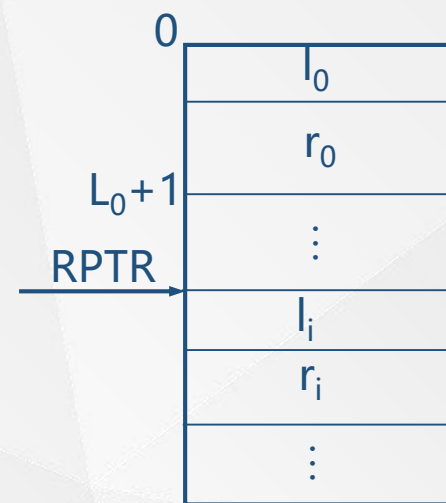
## (2) 记录式文件

- ① **定义**：记录式文件是一种有结构的文件。  
这种文件在逻辑上总是被看成一组连续顺序的记录的集合。

### ② 定长记录与变长记录



定长记录文件



变长记录文件

### (3) 两种逻辑结构的比较

- 流式文件就象给一张白纸给用户，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。
- 记录式文件就象给一张表格给用户，用户要按表规定的格式填信息。

## 2. 文件存取方法

### (1) 顺序存取

- 后一次存取总是在前一次存取的基础上进行的。
- 顺序存取时不必给出具体的存取位置。

### (2) 随机存取

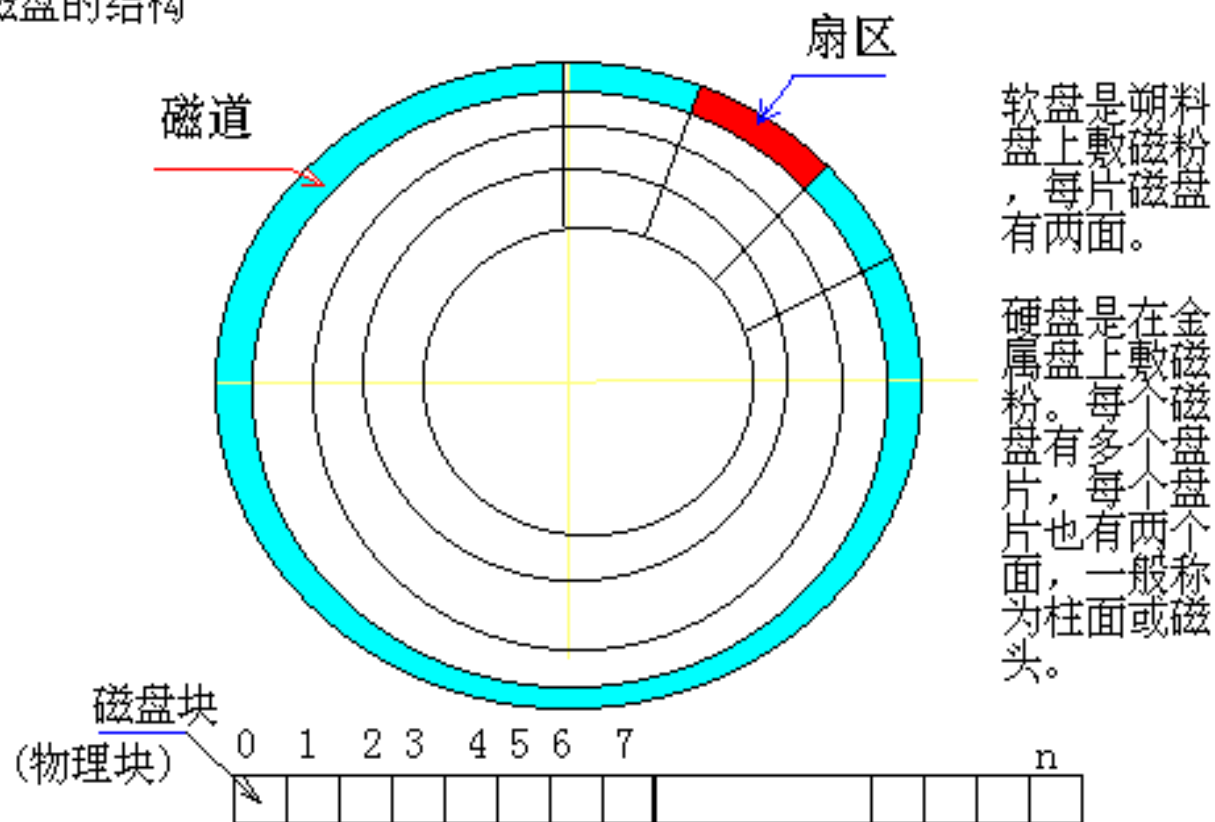
- 用户以任意次序请求某个记录。
- 随机存取时要指出起始存取位置（例如记录号）。

# 文件的物理结构

# 基本概念

- **块 (block)** 是存储介质上连续信息所组成的一个区域，也叫物理记录。
- 间隙是块之间不记录用户代码信息的区域。
- 块是主存和辅存间**信息交换的物理单位**，每次交换一块或整数块。
- 不同类型的存储介质，块的大小常常各不相同；同一类型的存储介质，块的大小也可以不同。
- 块的大小要考虑到用户使用方式、数据传输效率和存储设备类型等多种因素。

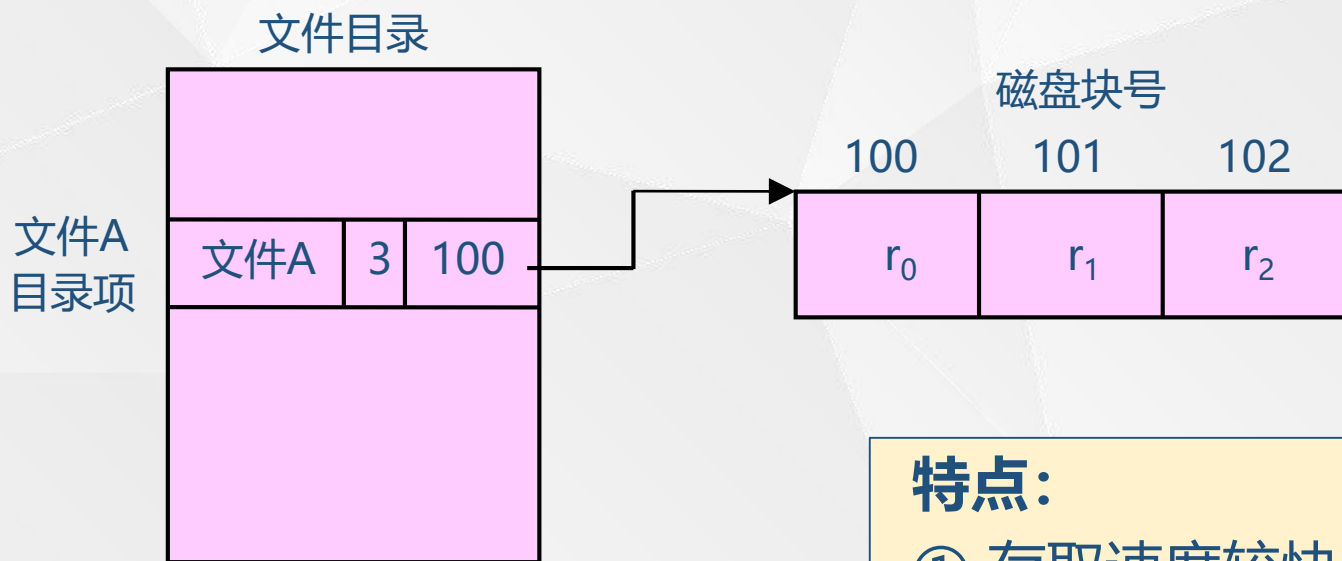
磁盘的结构





# 1. 连续文件

- 连续文件结构是由一组分配在磁盘连续区域的物理块组成的。
- 例：**文件A有三个记录 (逻辑记录与物理块大小相等, 都为512 B), 采用连续文件结构,  $r_0$ 存放在块号为100的磁盘块上。



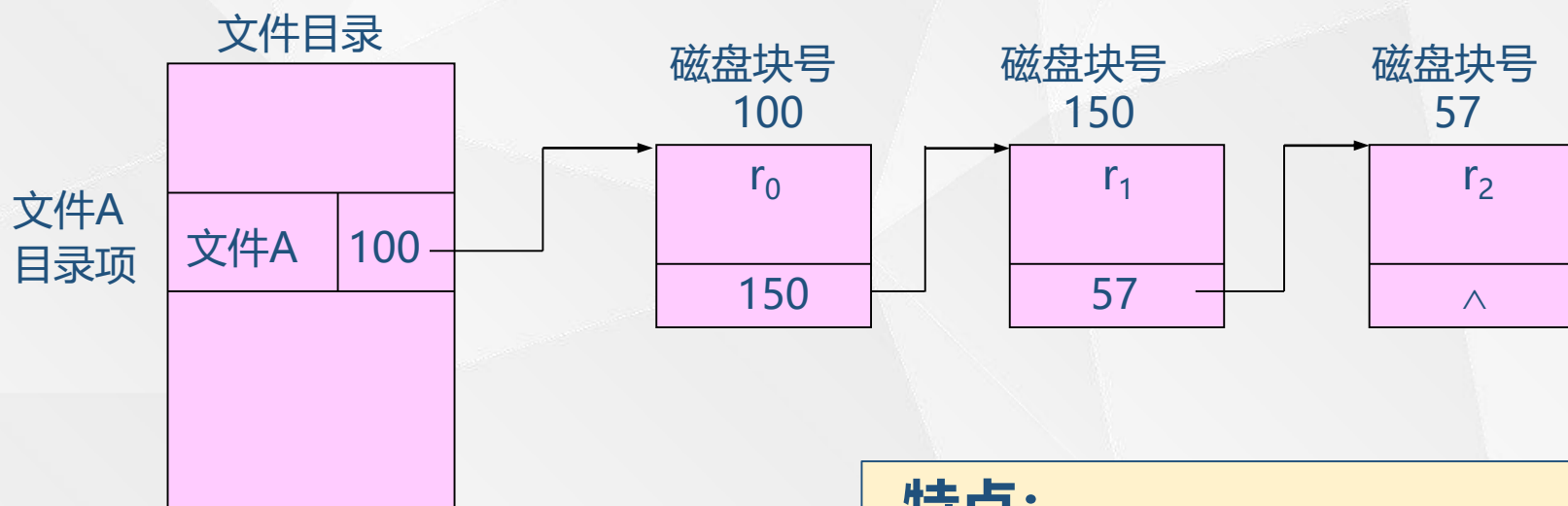
**问题：**在连续文件结构下，存取记录 $r_i$ 时，应如何操作？

## 特点：

- ① 存取速度较快
- ② 文件长度一经固定便不易改变
- ③ 文件的增生和扩充不易

## 2. 串联文件

串联文件结构由按顺序串联的块组成，即文件的信息存于若干个物理块中，每个物理块的最末一个字作为链接字，指出后继块的物理地址。文件的最后一块的链接字为结束标记“^”，它表示文件至本块结束。



**问题：**在串联文件结构下，存取记录 $r_i$ 时，应如何操作？

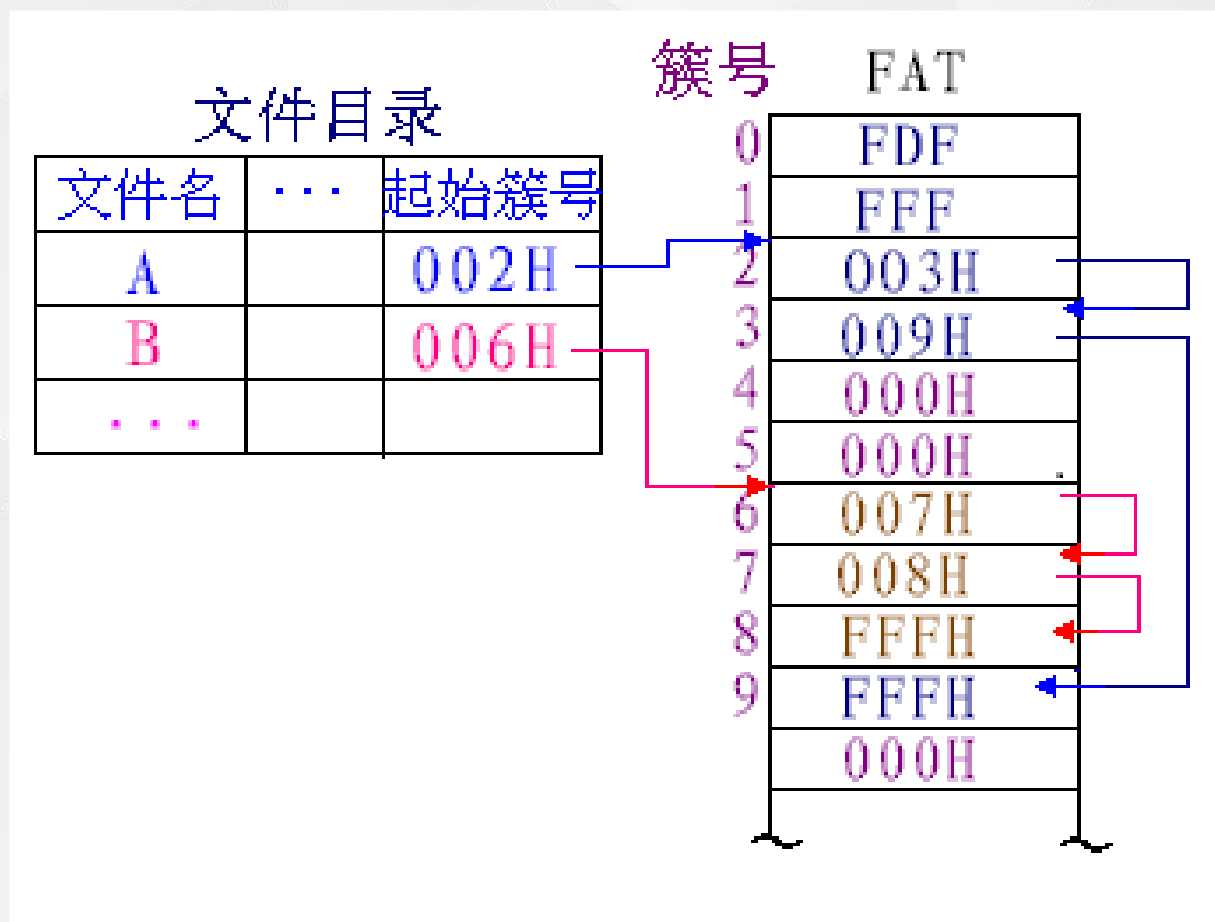
### 特点：

- ① 能较好地利用辅存空间
- ② 易于对文件进行增生和扩充
- ③ 连续存取速度快，随机存取速度慢

- 文件映照把串联文件中的链接字集中在一个结构中，既保持了串联文件的优点，也克服了其随机存取速度慢的缺点。DOS、WINDOWS系统就采用了这样文件结构，称为FAT文件。
- **文件分配表FAT**是以链接方式存储文件的系统中记录磁盘分配和跟踪空白磁盘块（簇）的数据结构。
- 该表在文件系统格式化后产生，共包含N个表项，每个表项对应一个簇，编号从0开始直至N-1（N为磁盘中簇的总数）。
- 每个表项中的内容为存放文件数据的下一个簇的簇号。
- 文件的首地址（第一个簇号）存放在目录中。因此，从目录中找到文件的首地址后，就能找到文件在磁盘上的所有存放地址。

# FAT12

- 磁盘格式化后建立，从磁盘的第二个扇区开始，有**两个**相同的FAT。
- 磁盘的类型：
  - FDF，双面
  - FCH，单面（9扇区/磁道）
  - F8H，硬盘
- 000H：空闲簇
- FFFH：文件的结尾簇
- XXXH：文件的下一簇



对于一个FAT16文件系统，已知：

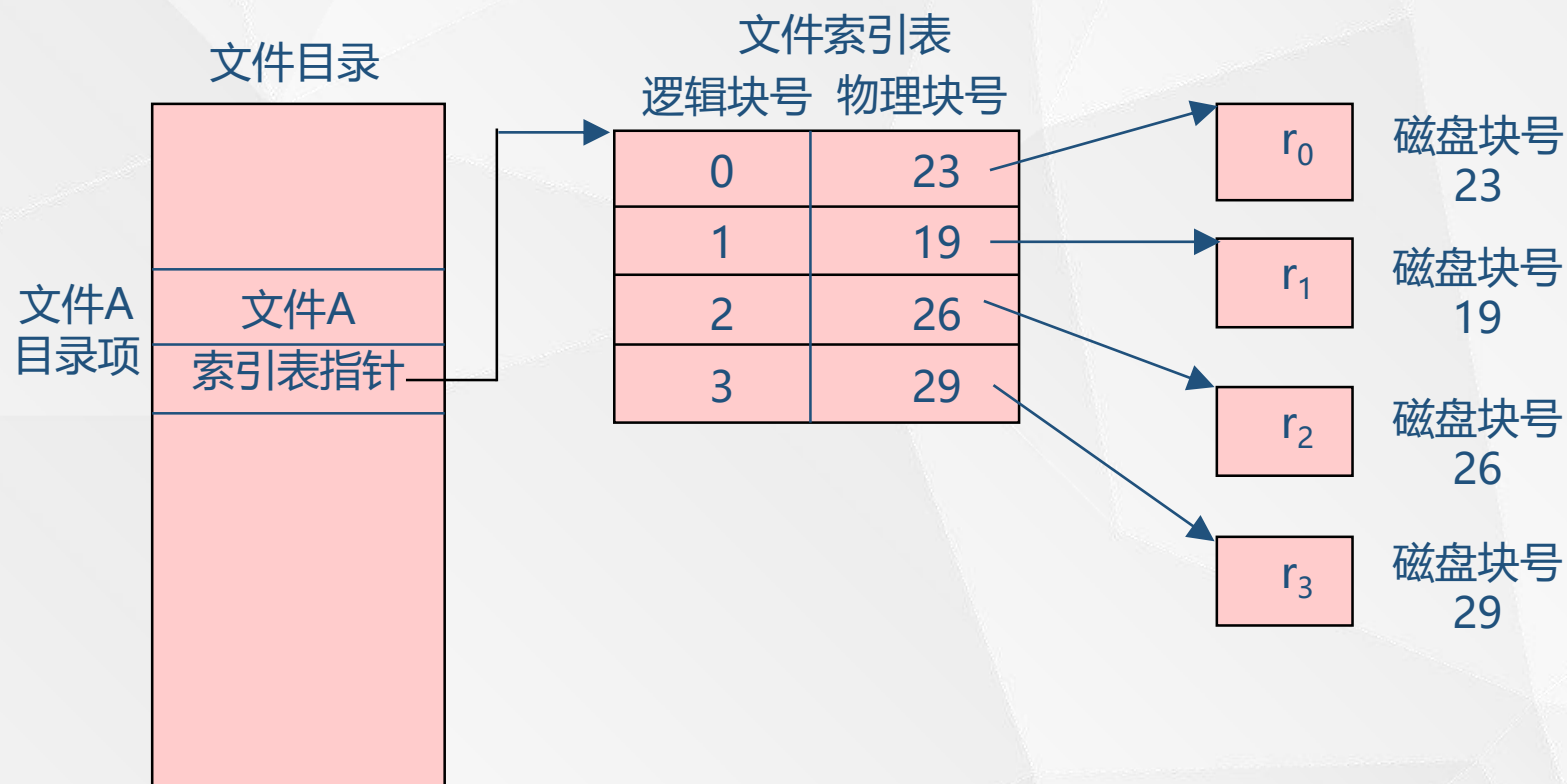
- 系统采用16位来表示簇号；
- 每簇由64个逻辑相邻的扇区组成；
- 扇区大小为512字节；

问：

- ① 该FAT16文件系统能够管理的最大逻辑磁盘大小？
- ② 若一个文件的逻辑内容为10个字节，该文件的物理大小是多少？

### 3. 索引文件

- 系统为每个文件建立逻辑块号与物理块号的对照表，这张表称为该文件的索引表。
- 索引文件由数据块和索引表构成。





## 3. 索引文件

- **索引文件的操作**

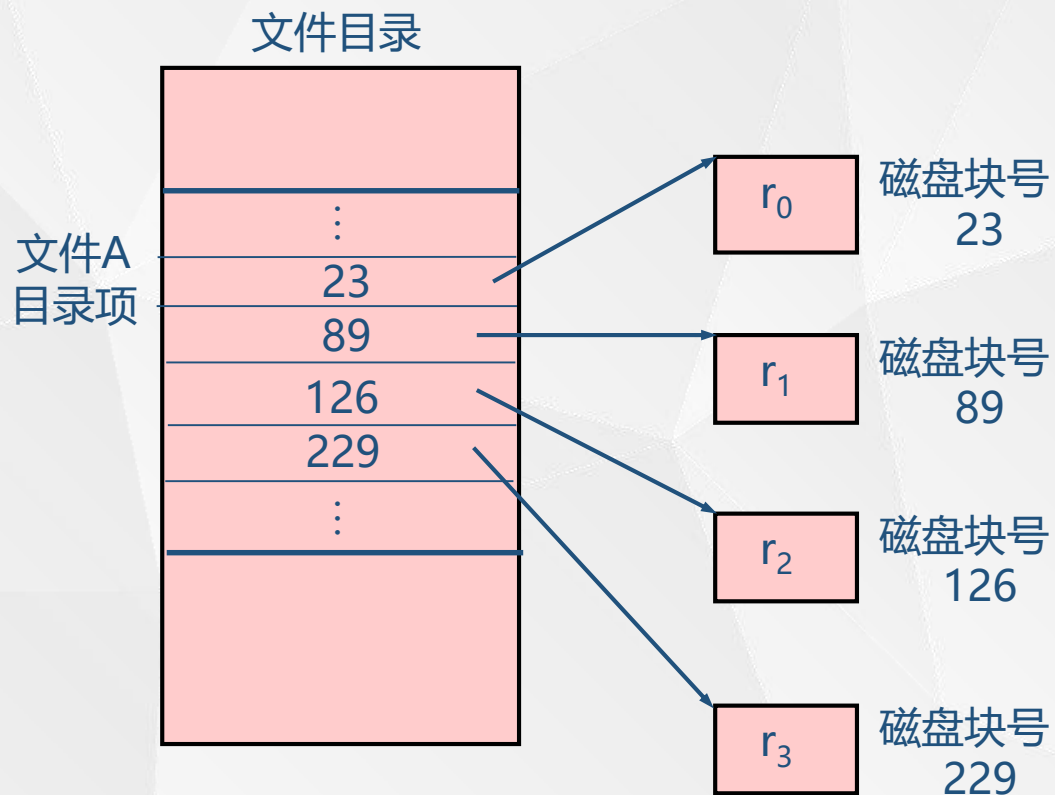
- 索引文件在存储区中占两个区
  - 索引区：存放索引表
  - 数据区：存放数据文件
- 访问索引文件的操作（两步）
  - 查文件的索引表，由逻辑块号查得物理块号
  - 由物理块号而获得所要求的数据

- **索引文件的特点**

- ① 易于文件的增删
- ② 直接读写任意记录

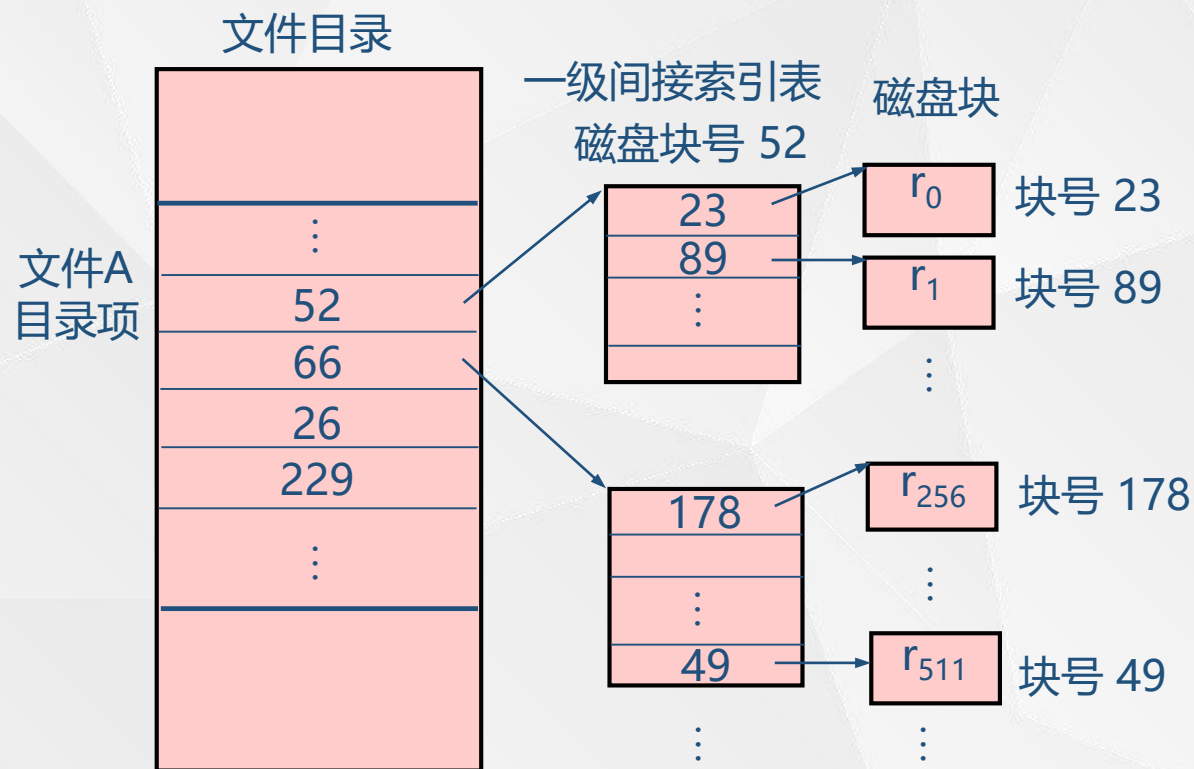
**Q：代价是什么？**

## (1) 直接索引



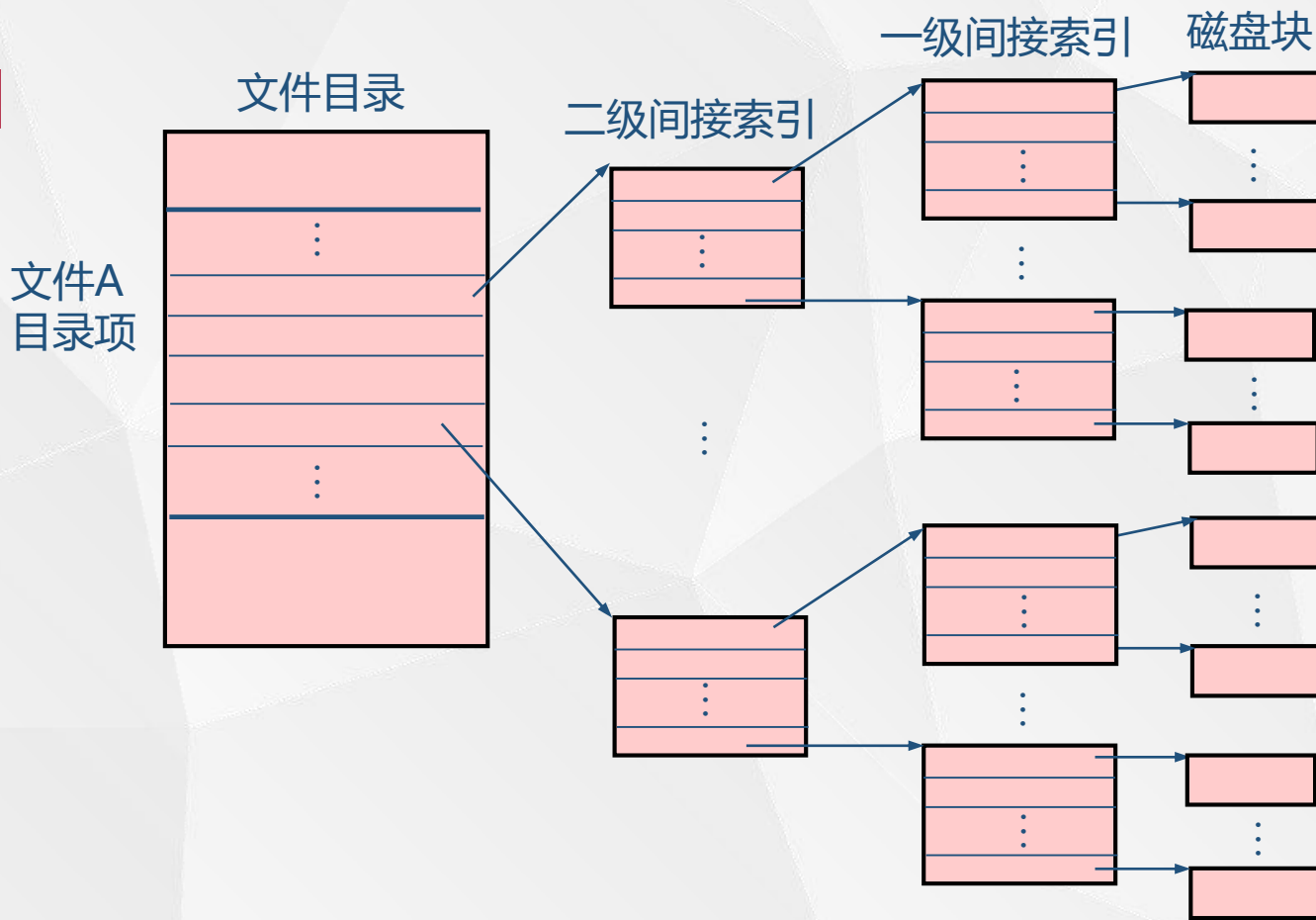
文件目录项中有一组表项用于索引，登记逻辑记录所在的磁盘块号。 (问题? )

## (2) 一级间接索引



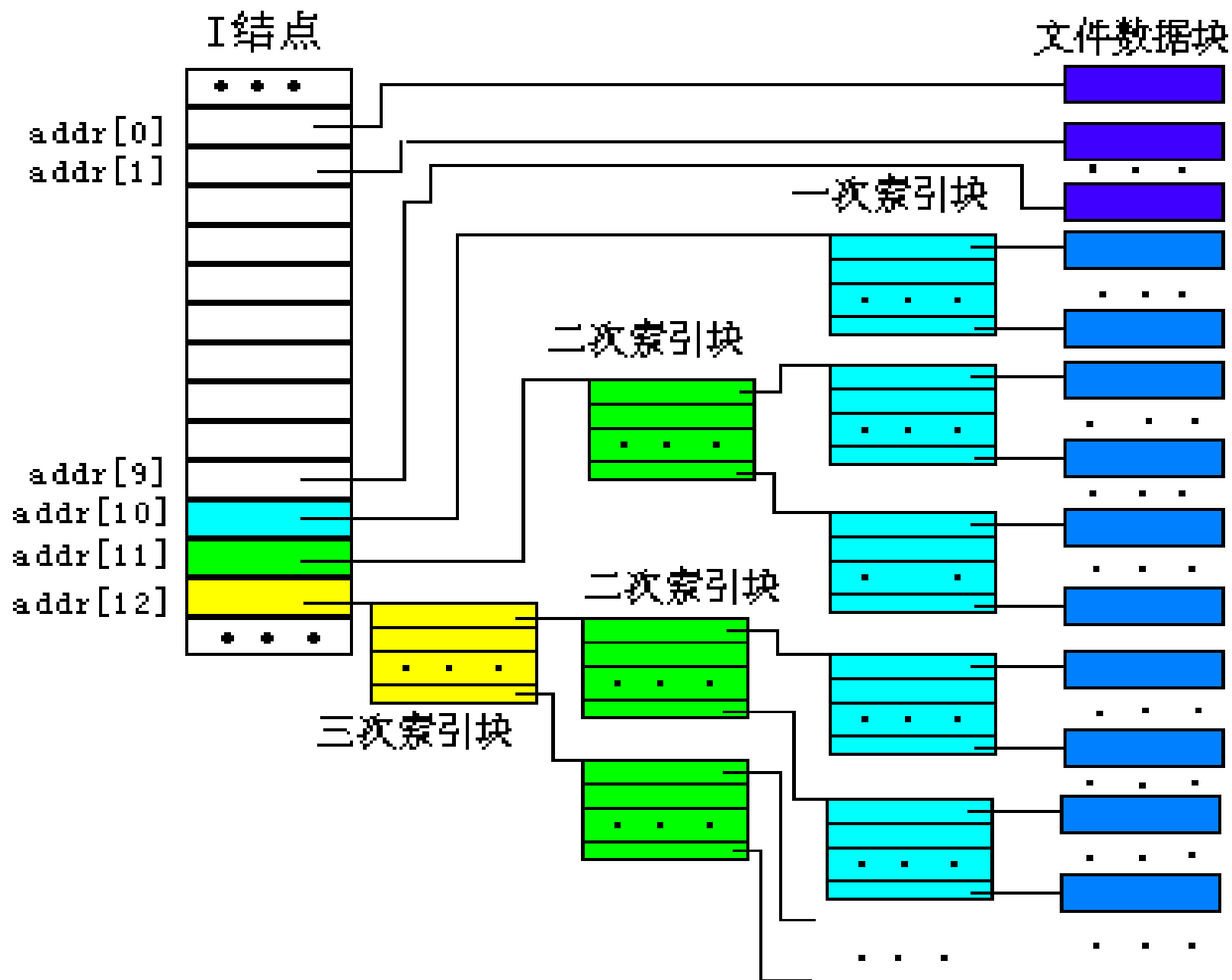
文件目录项中有一组表项，其内容登记的是第一级索引表块的块号。第一级索引表块中的索引表项登记的是文件逻辑记录所在的磁盘块号。

## (3) 二级间接索引



文件目录项中有一组表项，其内容登记的是第二级索引表块的块号。第二级索引表块中的索引表项登记的第一级索引表块的块号，第一级索引表项中登记的是文件逻辑记录所在的磁盘块号。

# UNIX 采用的改进索引结构



# || 几种文件物理结构的比较

## • 连续文件

- 优点：不需要额外空间开销，顺序访问和随机访问的效率都高。
- 缺点：动态增长困难；要求用户创建时提供文件大小；存储空间浪费较大。
- 适合长度固定不变的只读文件的存储。

## • 串联文件

- 克服了连续文件的不足；但随机访问的系统开销较大；适应于顺序访问的文件。
- DOS及WINDOWS系统中改造了串联文件结构，使其克服了串联文件的不足，但增加了系统的危险性。

## • 索引文件

- 既适应于顺序访问，也适应于随机访问；但索引表需要存储开销和访问时间开销。
- UNIX系统是使用索引结构成功的例子。



# 物理结构与存储介质之间的关系

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

# 文件存储空间的管理

- **功能：**负责辅助存储空间的有效分配和释放。
  - 创建和扩充文件时，**确定分配哪些磁盘块；**
  - 删除文件和缩短文件时，**回收磁盘块；**
  - 随着分配和回收，**可能会出现碎片需要整理。**
- **常用方法：**
  - 位示图
  - 空闲文件目录
  - 空闲块链

- 磁盘空间由固定大小的块组成，可方便地使用位示图管理。
- 每一位对应于一个物理块，1表示被占用，0表示空闲。
- **优点：**
  - 每个盘块仅需一比特来标识
  - 若盘块长为1KB，则位示图开销为0.012%
- **缺点：**对于大容量磁盘，位示图的存储和查找开销较大，需要建立辅助统计表来提高访问速度。
- **讨论：**FAT文件系统是如何管理空闲块的？

# 空闲文件目录

- 将空闲区存储块的位置及其连续空闲的块数构成一张表;
- 分配时, 依次扫描空闲区表, 寻找合适的空闲块并修改登记项;
- 删除文件并释放空闲区时, 把空闲位置及连续空闲区长度填入空闲区表, 出现邻接的空闲区时, 还需执行合并操作。
- 适用于连续文件结构。
- **缺点:** 碎片问题

序号	第一个空闲块号	空闲块个数	物理块号
1	5	4	(5,6,7,8)
2	23	5	(23,24,25,26,27)
3	30	6	(30,31,32,33,34,35)
4	.....	.....	.....

- 把所有空闲块连接在一起，系统保持指针指向第一个空闲块，每一空闲块中包含指向下一个空闲块的指针；
- 申请一个空闲块时，从链头取一块并修改系统指针；
- 释放占用块时，将其挂到空闲链的链头，并修改系统指针指向它。
- **优点：**几乎不需要占用空间
- **缺点：**分配多块时效率低（需要I/O操作）
- UNIX文件系统对这种方法进行了改进，在一个空闲块中保存一组空闲块指针，称为成组链接法。



# 文件目录及其结构

- **文件控制块(FCB)**是文件系统为每个文件建立的唯一的管理数据结构，一般包括：

- **文件标识和控制信息**

- 文件名、用户名、文件存取权限、访问控制权限、文件类型等；

- **文件逻辑结构信息**

- 记录类型、记录个数、记录长度、成组因子等；

- **文件物理结构信息**

- 文件所在设备名、文件物理结构类型、记录存放在辅存中的块号或文件信息首块盘块号，文件索引的位置等；

- **文件使用信息**

- 共享文件的进程数，文件修改情况，文件最大长度和当前大小；

- **文件管理信息**

- 文件建立日期，最近修改日期，最近访问日期等。

- 文件系统基于FCB来实现文件的**按名存取**
  - 创建文件时，为其建立FCB，用来记录文件的属性信息；
  - 存取文件时，先找到其FCB，再找到文件信息盘块号或首块物理位置。
- 为了加快文件的查找速度，通常将FCB集中起来，组成**文件目录**。
- 文件目录包含两种目录项：普通文件的FCB、子目录的**目录文件**的FCB
- 目录文件至少包含两个目录项：
  - 当前目录项 **.**
  - 父目录项 **..**
- 文件目录的基本功能
  - 将文件名转换成此文件信息在磁盘上的物理位置
  - 有效地组织和管理文件

由目录项构成的文件称为目录文件

# 一级文件目录

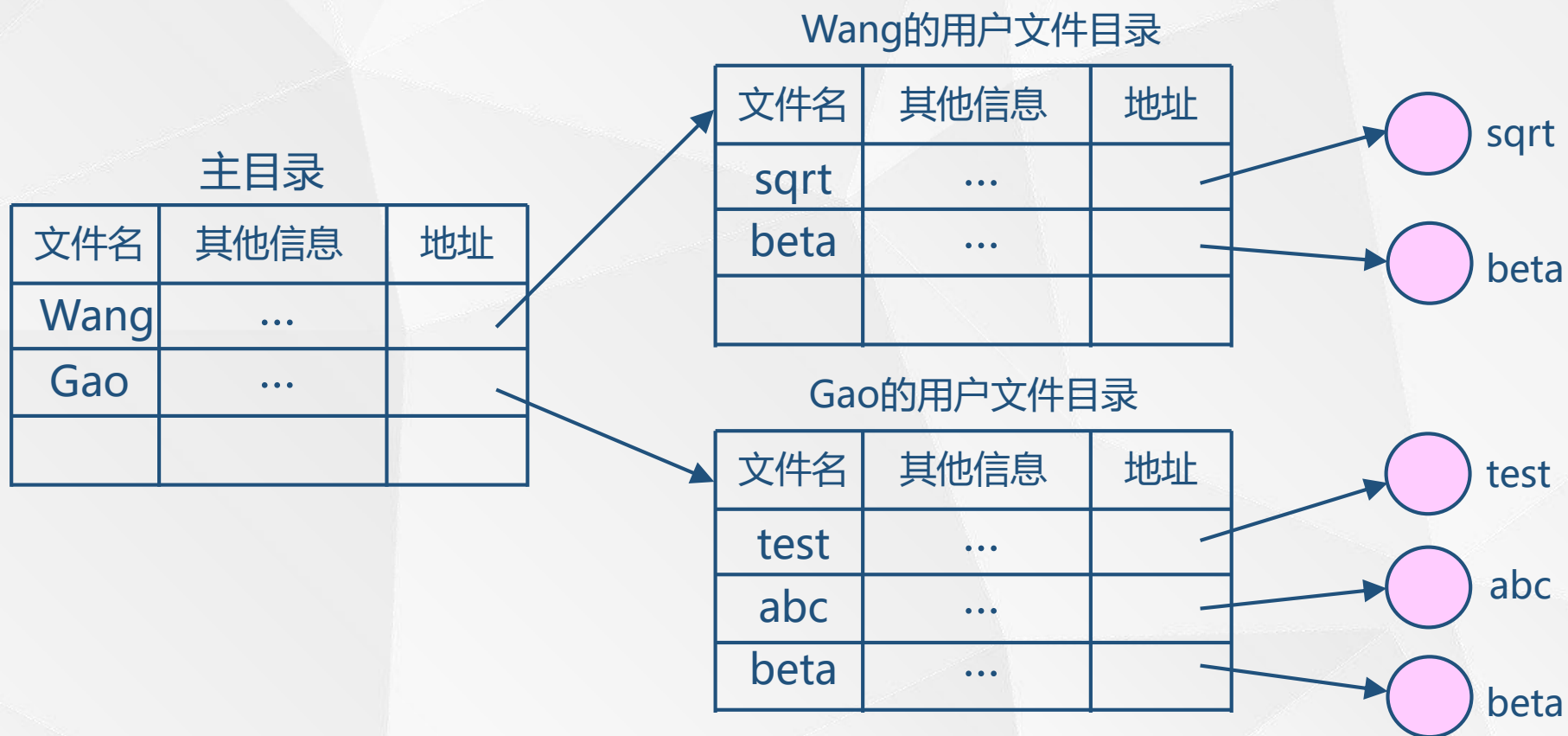
系统将已建立的所有文件的文件名、存放地址及有关的说明信息放在一张表中，这张表称为一级文件目录。（整个系统中只有一个目录文件）

文件 名	物 理 地 址	其 他 信 息
pa		
test		
compiler		
assembler		
abc		
wang		

- 特点：
  - 实现简单；
  - 不允许两个文件有相同的名字；
  - 在**多用户**环境中，易出现重名问题。
- **重名问题**又称命名冲突，是指不同用户对不同文件起了相同的名字，即两个或多个文件只有一个相同的符号名。
- 为解决命名冲突、获得更灵活的命名能力，文件系统必须采用多级目录结构。

# 二级文件目录

- 将文件目录分成**主目录**和**用户文件目录**两级。
- 每个用户建立一个用户文件目录，登记该用户建立的所有文件的相关信息。
- 主目录登记系统中各个用户文件目录的相关信息。

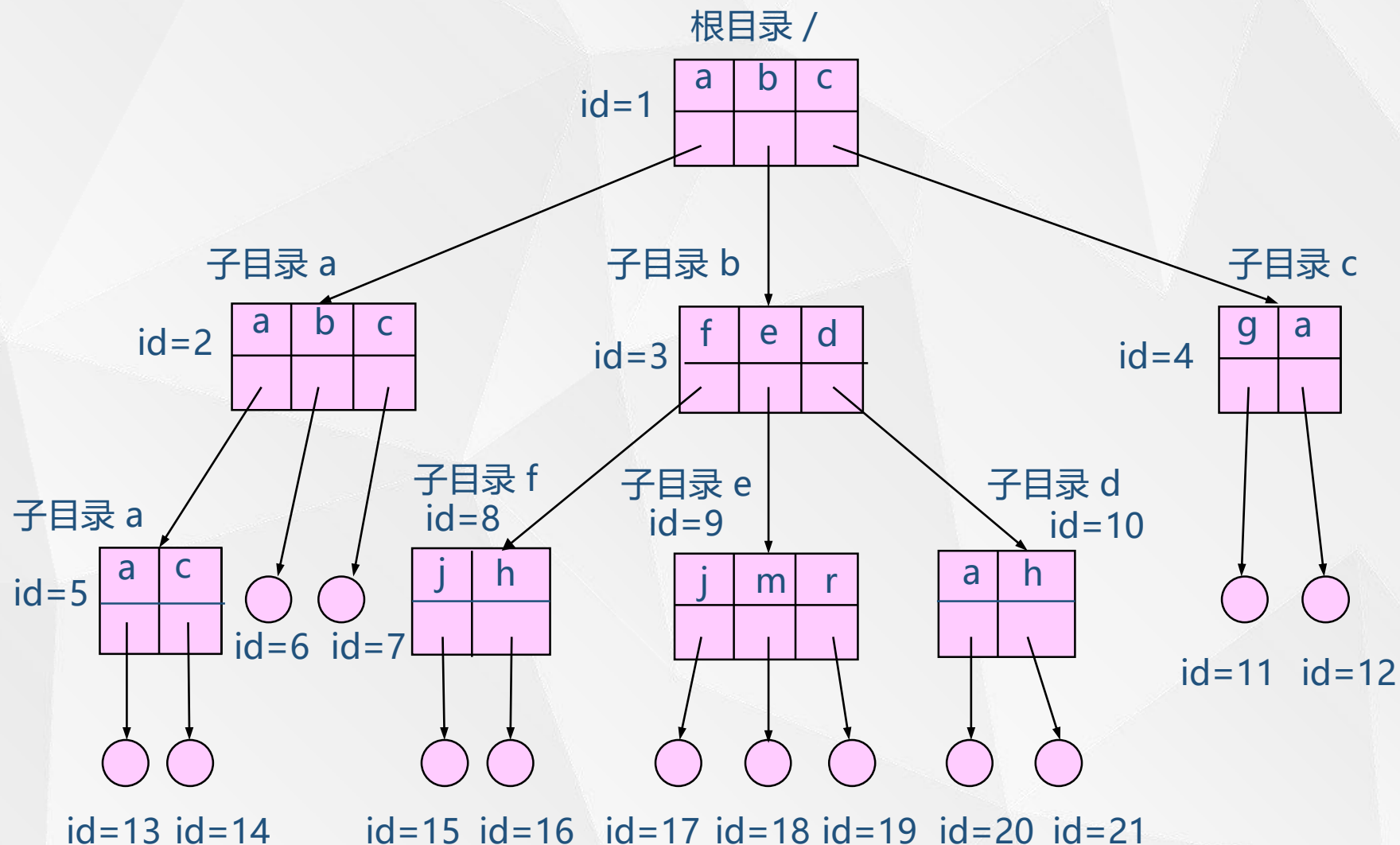




- **特点：**
  - 解决了命名冲突，允许不同用户目录下的文件名可以相同。
- **方法：**
  - 存取一个文件时给出文件路径名。
- 在二级文件目录下，文件的路径名是由用户名和文件名拼起来得到的。如：
  - 用户wang的文件beta的路径名为 “wang:beta”
  - 用户gao的文件beta的路径名为 “gao:beta”

# 树型文件目录

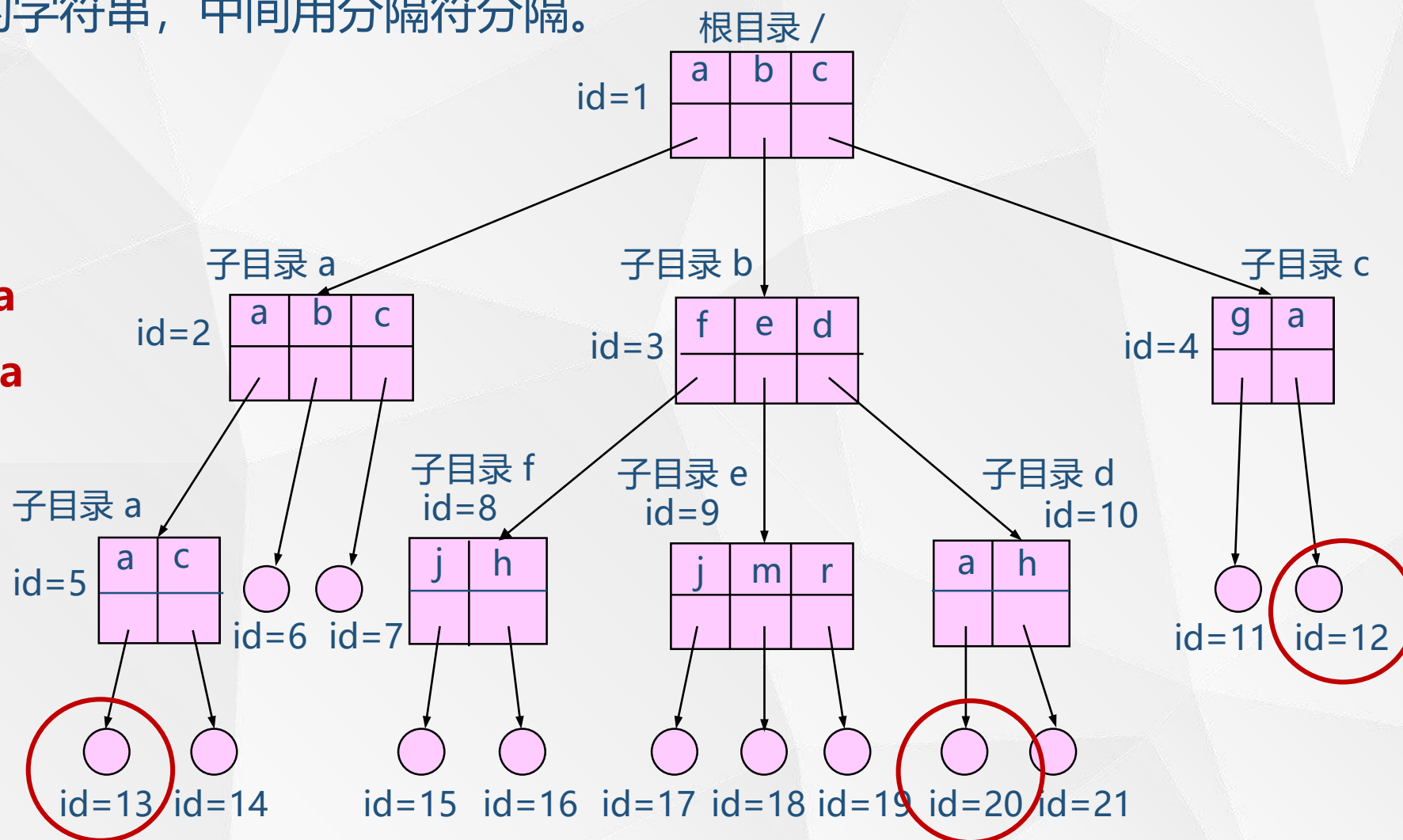
在多级目录系统中 (除最末一级外), 任何一级目录的录项可以描述一个目录文件, 也可以描述一个非目录文件 (数据文件), 而数据文件一定在树叶上。这样就构成了一个树形层次结构。



# 文件路径名

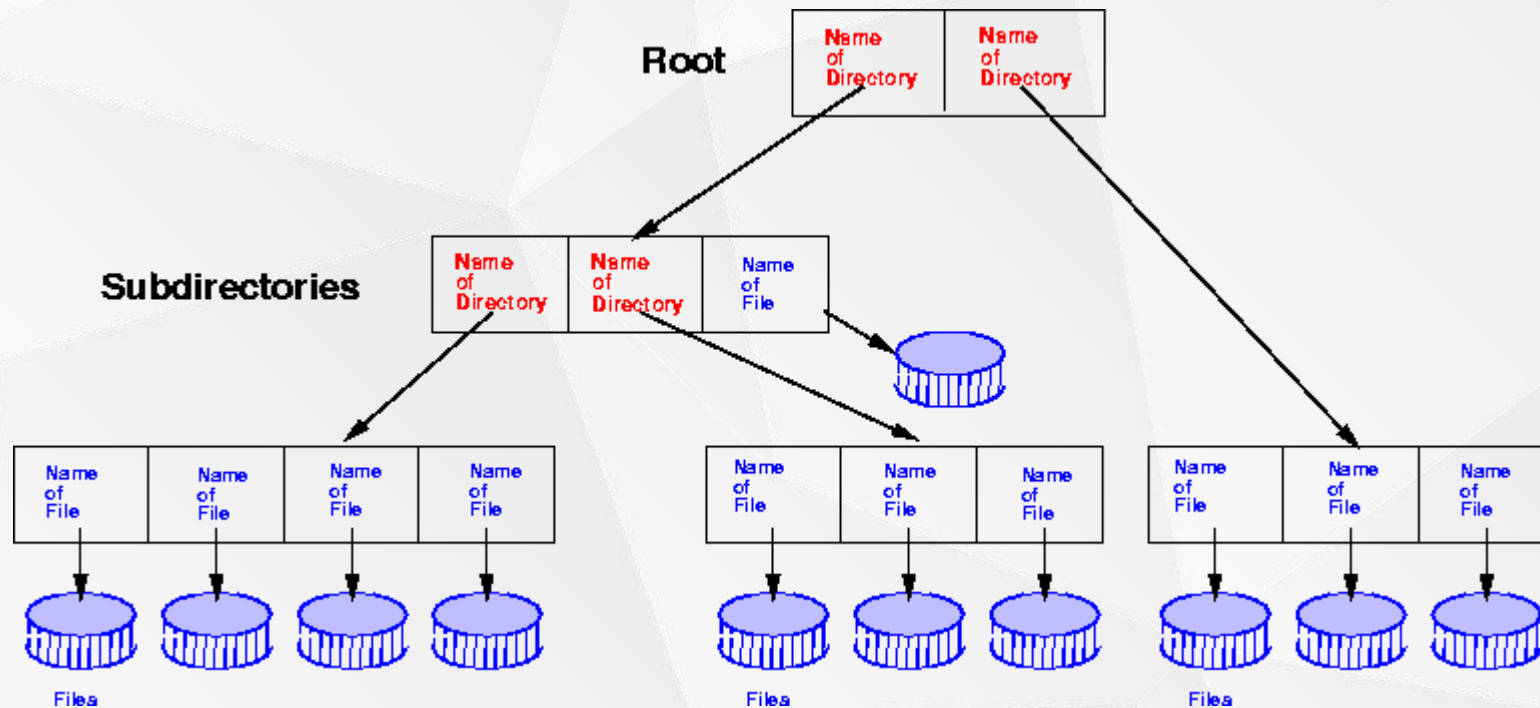
多级目录中文件的路径名，是由根目录到该文件的通路上所有目录文件的符号名和该文件的符号名组成的字符串，中间用分隔符分隔。

- **id=12:** /c/a
- **id=13:** /a/a/a
- **id=20:** /b/d/a

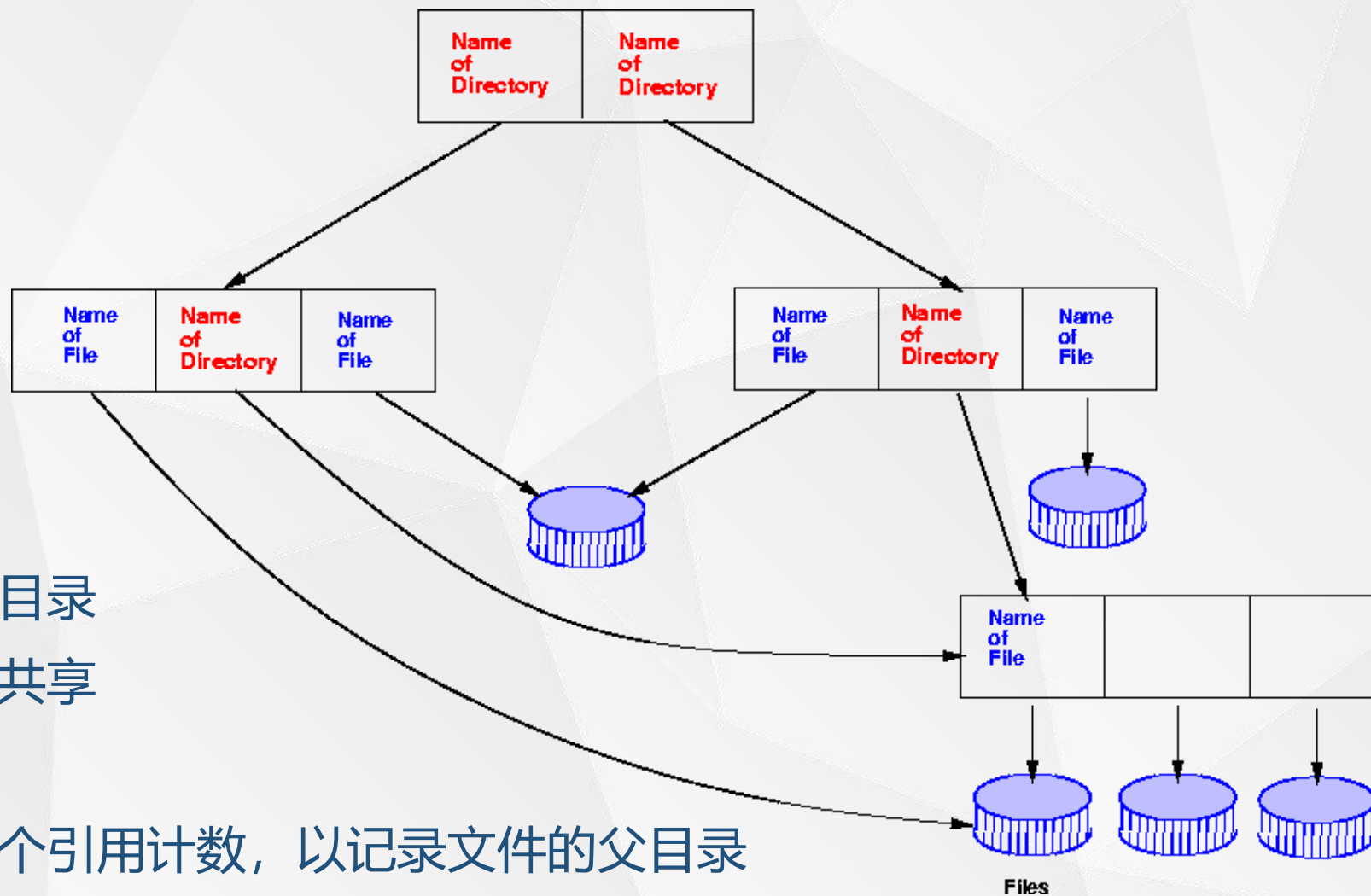


# 纯树形目录结构

- 目录可以有任意多的层次
- 目录可以包含子目录，也可以包含文件
- 每个文件只有一个父目录
- 文件共享较为困难



# DAG目录结构



- 一个文件可以有多个父目录
- 能较方便地实现文件的共享
- 目录结构的维护较复杂
- 需要为每个文件维护一个引用计数，以记录文件的父目录个数，仅当引用计数值为1时，删除操作才真正删除文件。



# 文件共享与安全



- 文件共享是指某一个或某一部分文件可以让事先规定的某些用户共同使用。
- 文件安全是指保护文件不得被未经文件主授权的任何用户存取，且对于授权用户也只能在允许的存取权限内使用文件。常见存取权限包括：
  - **无 (None)**：用户不知道该文件的存在；
  - **知情 (Knowledge)**：用户可以确定该文件的存在和文件的所有者；
  - **执行 (Execution)**：用户可以加载并执行一个程序文件；
  - **读 (Reading)**：用户可以读该文件；
  - **追加 (Appending)**：用户可以在文件末尾追加数据，但不能修改或删除内容；
  - **更新 (Updating)**：用户可以在文件中修改、删除或添加数据；
  - **变更保护 (Changing protection)**：用户可以变更授予其他用户的访问权限；
  - **删除 (Deletion)**：用户可以从文件系统中删除该文件。

- **如何进行文件的保护**

- 操作前先对用户的权限进行验证。
- 所谓存取权限的验证，是指用户存取文件之前，需要检查用户的存取权限是否符合规定，符合者允许使用，否则拒绝。

- **验证用户存取权限的方法**

- 访问控制矩阵
- 存取控制表
- 用户权限表
- 口令
- 密码

# 如何加快文件的查找

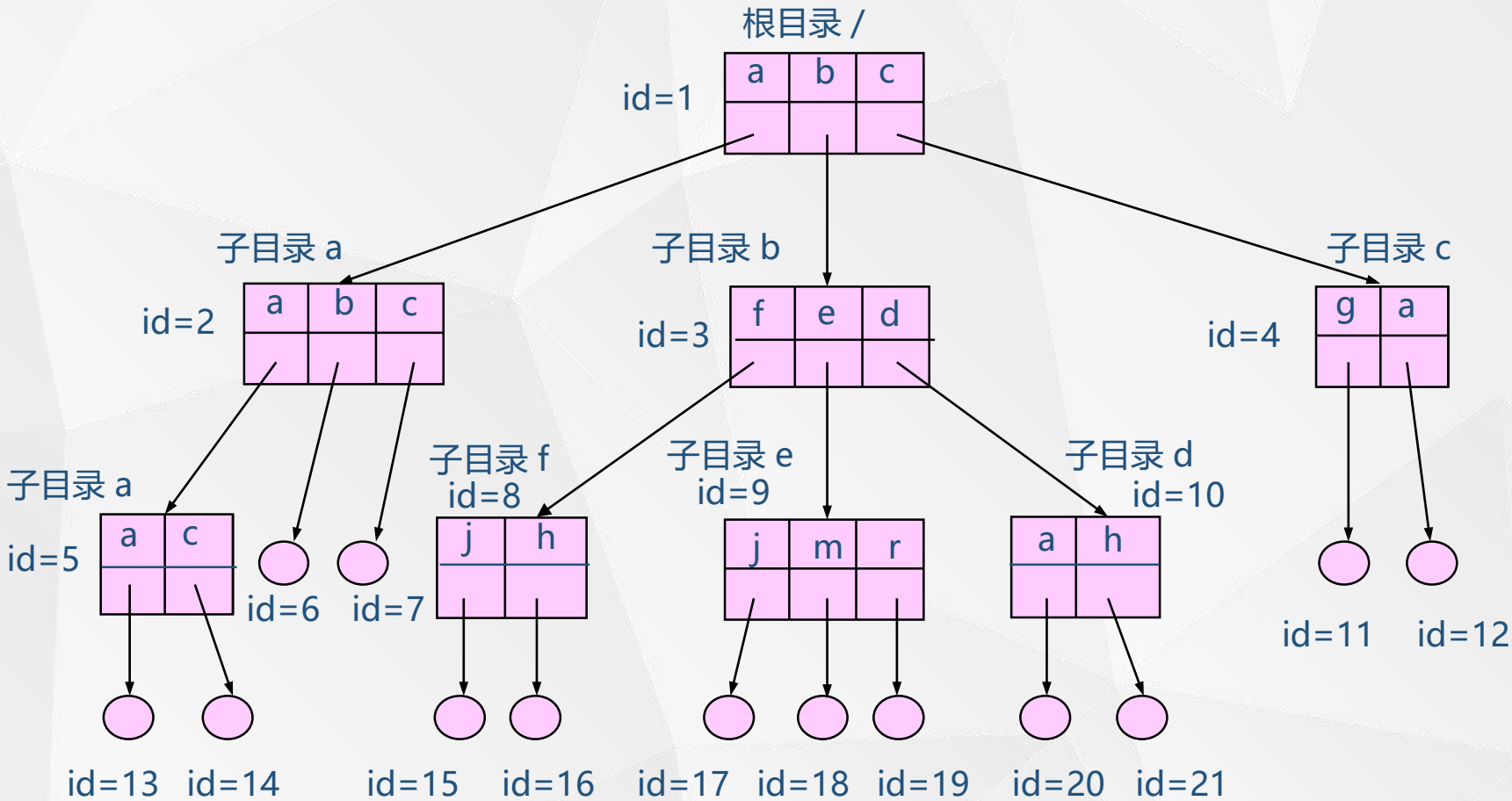
- 当前目录（又称值班目录）

当指定当前目录后，用户对文件的所有访问都是相对于当前目录进行的。

- 当前目录为id=8时，  
id=15的文件的  
路径名为：**j**

- 此时如果要访问id=12  
的共享文件a，  
则该文件的  
相对路径为：

**`*/*/c/a`**

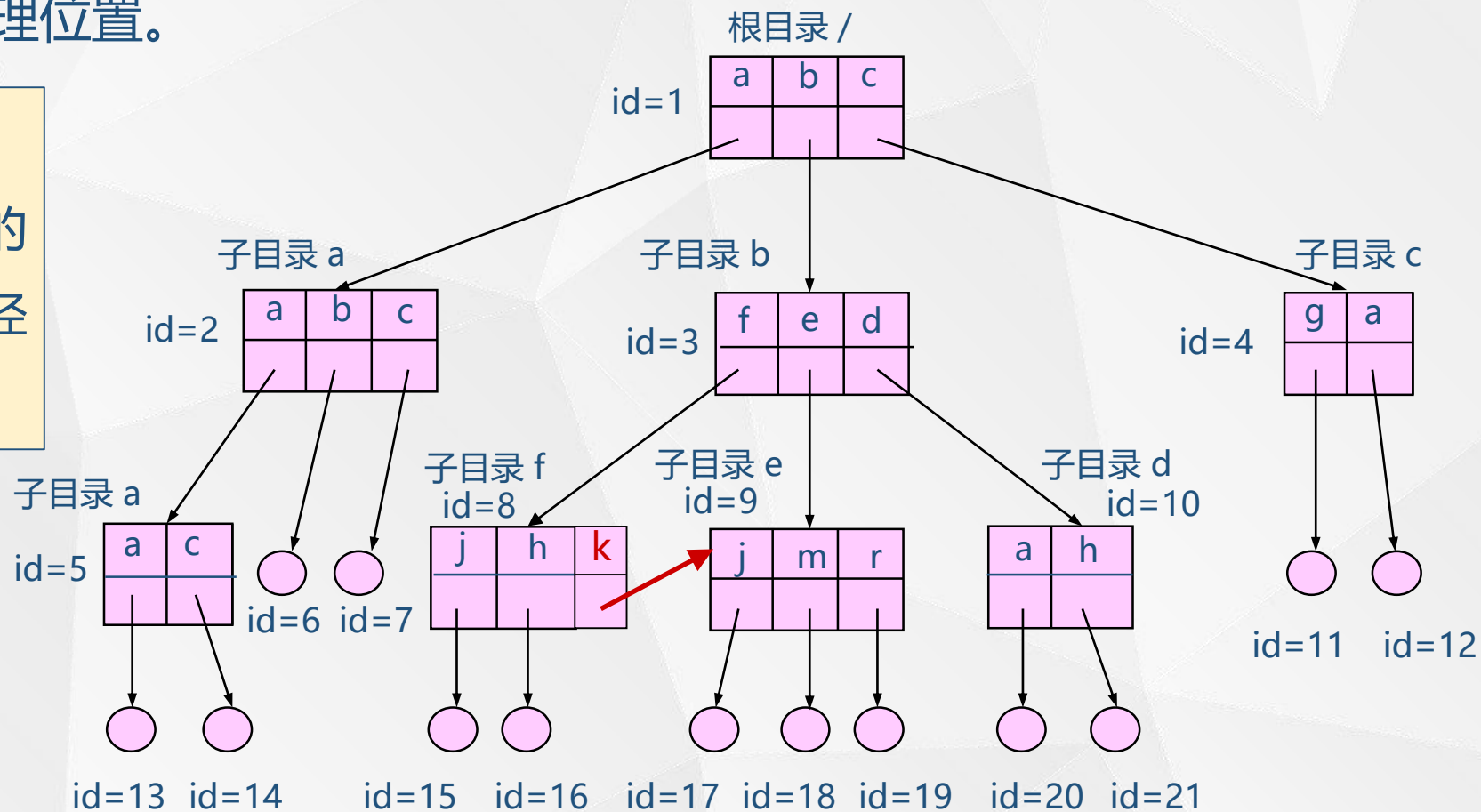


# 如何加快文件的查找

## • 链接技术

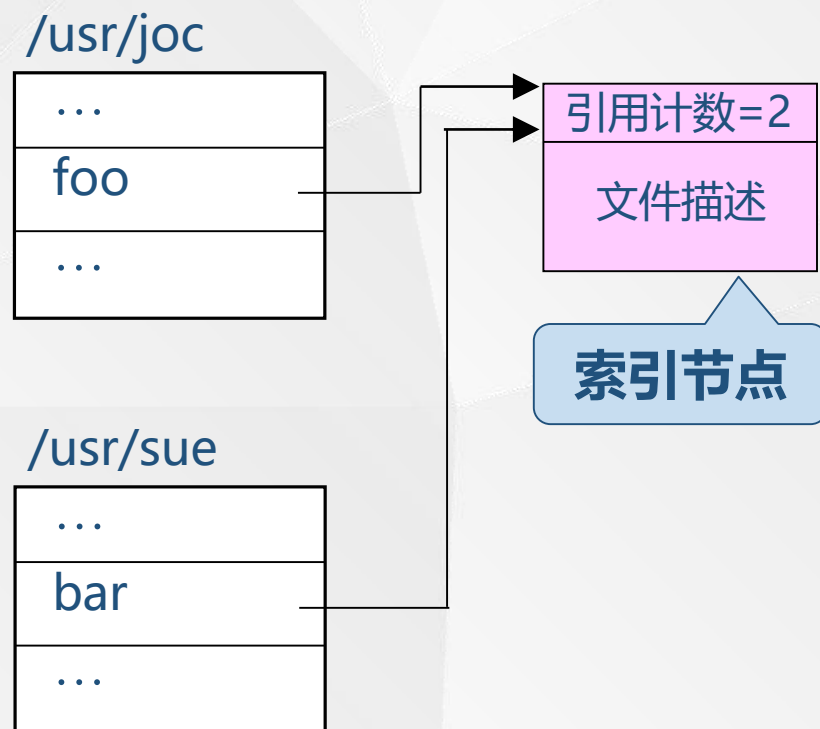
所谓“链接”，就是在相应目录表目之间进行链接，即个目录中的表目直接指向另一个目录表目所在的物理位置。

假定当前目录为id=8，  
共享子目录e中id=17的  
文件j，可使用文件路径  
名 **k** 直接存取该文件。

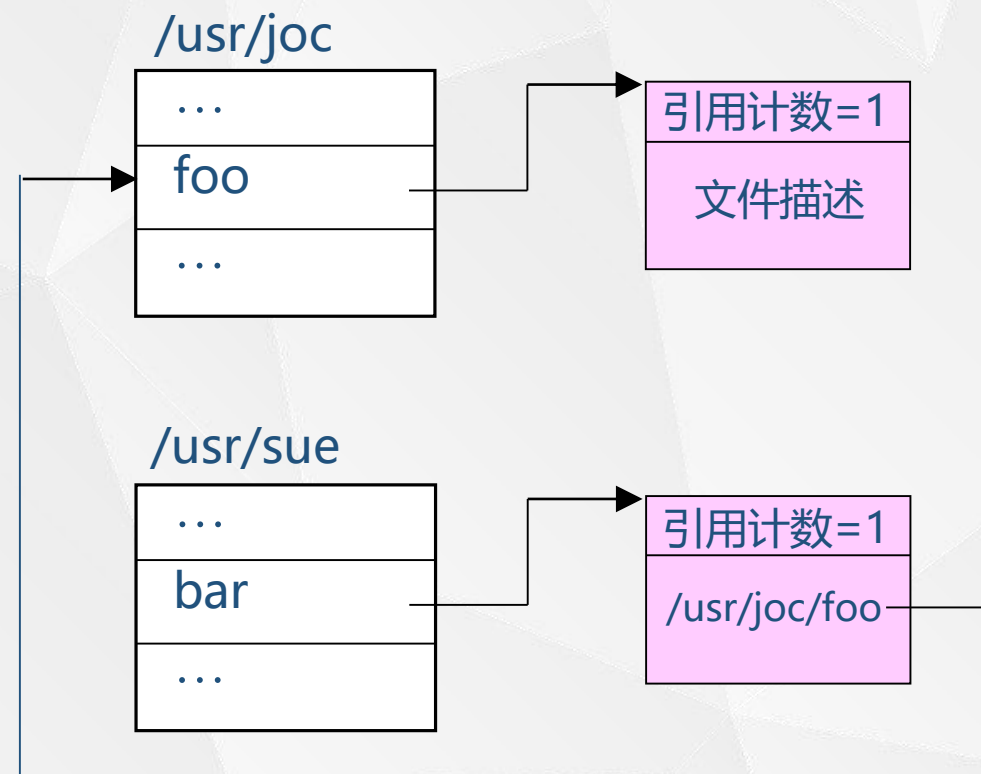


# UNIX/Linux 的链接文件

- UNIX/Linux 下有两种链接文件：**硬链接(Hard Link)** 和 **软链接(Soft Link)**，软链接又称**符号链接(Symbolic link)**。



硬链接示意图



软链接示意图



# 硬链接与软链接的区别

- 创建命令：
  - 硬链接: **ln /data/ln/src /data/ln/dst**
  - 软链接: **ln -s /data/ln/src /data/ln/dst**
- 硬链接与源文件等价；软链接文件中并不包括实际的文件数据，只包括了它指向文件的路径。
- 删除源文件后，硬链接文件可照常使用；软链接文件的操作则会失败。
- 硬链接不能链接目录文件；软链接既可以链接到普通文件也可以链接到目录。
- 硬链接只限于本文件系统；软链接可以链接到处于不同文件系统的文件及目录。
- 硬链接可以加快文件查找速度；软链接并不能。



# 文件操作与文件备份

# 1. 文件操作

## (1) 常用的文件操作命令

- create 创建一个新文件
- delete 从系统目录中撤消一个文件
- rename 在系统目录中改变文件的名字
- open 打开文件，在用户和文件(或设备)之间建立一个逻辑通路
- close 关闭文件，在用户和文件(或设备)之间撤消一个逻辑通路
- write 写到一个文件(或设备)上
- read 从一个文件(或设备)读入数据信息

# 1. 文件操作

## (2) “打开文件”和“关闭文件”操作

### ① 打开文件操作

所谓打开文件就是把该文件的有关目录表目复制到主存中约定的区域，建立文件控制块，建立用户和这个文件的联系。

### ② 关闭文件操作

所谓关闭文件就是用户宣布这个文件当前不再使用，系统将其在主存中的文件控制块删去，因而也就切断了用户同这个文件的联系。

## 2. 文件备份

为了能在软、硬件失效的意外情况下恢复文件，保证文件的完整性、数据的连续可利用性，文件系统提供适当的机构，以便复制备份。

- **周期性转储**

按固定的时间周期把存储器中所有文件的内容转存到某种介质上，通常是磁带或磁盘。在系统失效时，使用这些转存磁盘或磁带，将所有文件重新建立并恢复到最后一次转存时的状态。

- **增量性转储**

只转储从上次转储以后已经改变过的信息；增量转储的信息量较小，故转储可在更短的时间周期内进行。

- **文件备份技术的发展**

动态备份，远程备份技术

# **UNIX文件系统的 主要结构及实现技术**

# 1. UNIX文件系统概述

## (1) UNIX文件的特点

- ① 树型文件目录结构
- ② 可安装拆卸的文件系统
- ③ 文件是无结构的字节流式文件
- ④ 将外部设备与文件一样对待



# 1. UNIX文件系统概述

## (2) UNIX文件的类型

### ① 普通文件

用户程序、数据文件

### ② 目录文件

用于组织和形成树型目录结构中的一个单位，由若干目录项组成。

### ③ 特别文件

与硬件设备有关的文件称为特别文件。包括块设备文件、字符设备文件。与计算机连接每一种输入输出设备都有一个特别文件。它是操作系统核心用于存取输入输出设备的通道，是用户与硬件设备联系的桥梁。

# 1. UNIX文件系统概述

## (2) UNIX文件的类型 (续)

### ④ 命名管道

管道是一种进程间通信机制。管道中缓存了从输入端接收的数据，这样从输出端读数据的进程就能以先进先出的方式来接收数据。

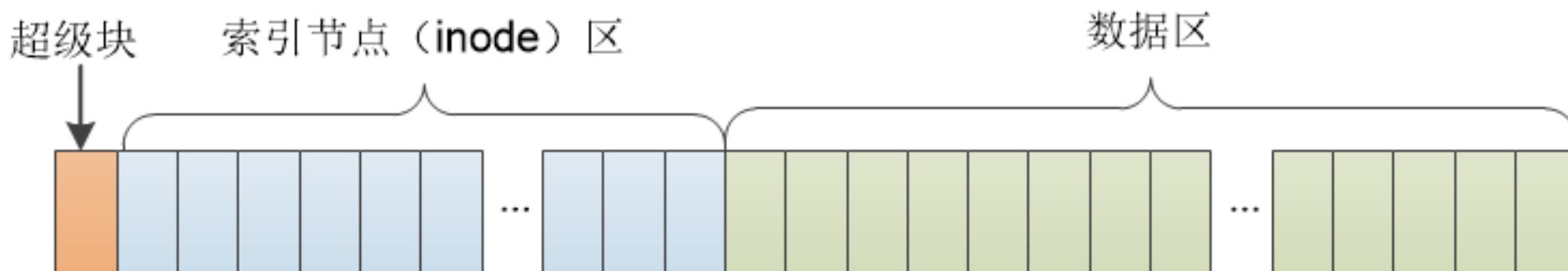
### ⑤ 链接文件

本质上，链接文件是一个已存在的文件的另一个文件名。

### ⑥ 符号链接

这是一个数据文件，文件中包含了其所链接的那个文件的名字。

## 2. UNIX系统的索引文件结构



### (1) 文件索引节点

UNIX系统把文件目录项中除了名字以外的信息全部存放到一个磁盘的数据块上，这种数据块称为磁盘索引节点（index node），简称 i 节点（inode）。

文件所有者标识	i_uid, i_gid
文件类型	i_type
文件存取许可权	i_mode
链接计数	i_ilink
文件存取时间	i_time
文件长度	i_size
地址索引表	i_addr[13]

- **文件所有者标识**  
定义对一个文件具有存取权的用户集合，分为文件所有者、用户组所有者
- **文件类型**  
分为正规文件、目录文件、字符特殊文件或块特殊文件等
- **文件存取许可权**  
按文件所有者、文件的用户组所有者及其他用户三个类别对文件施行保护。每类都具有读、写、执行该文件的存取权，并且能分别地设置。

文件所有者标识	i_uid, i_gid
文件类型	i_type
文件存取许可权	i_mode
链接计数	i_ilink
文件存取时间	i_time
文件长度	i_size
地址索引表	i_addr[13]

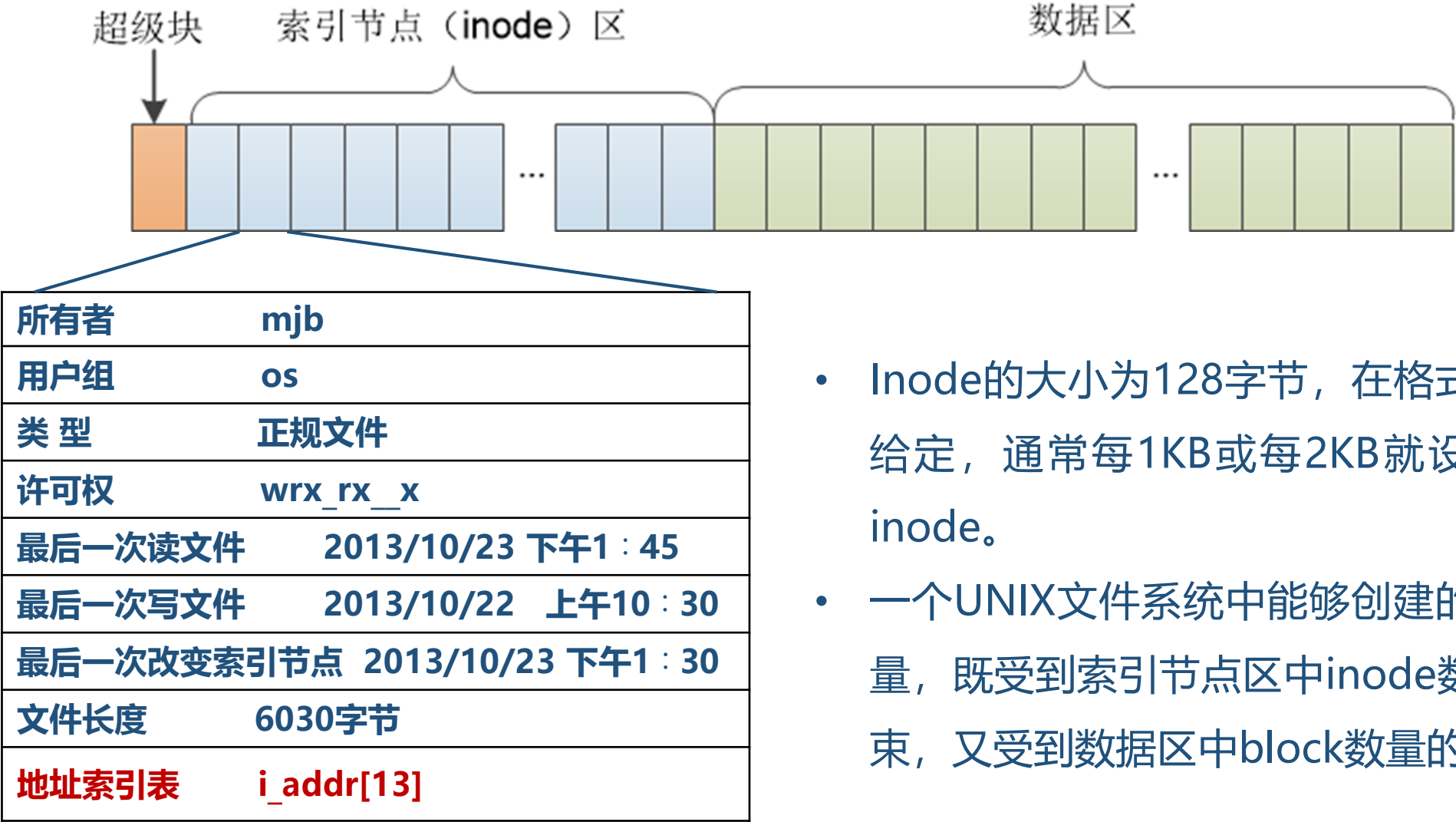
- **文件链接数目**

表示在文件目录结构中，有多少个文件名指向该文件。每当增加一个名字时，i\_ilink值加1，减少一个名字时其值减1。当其值减为0时，该文件才能真正删除。

- **地址索引表**

文件数据的磁盘地址信息，即地址索引表，在UNIX 第七版本用 i\_addr[8] 来描述。在UNIX system V 中用 i\_addr[13] 来描述。

# 磁盘索引节点示例



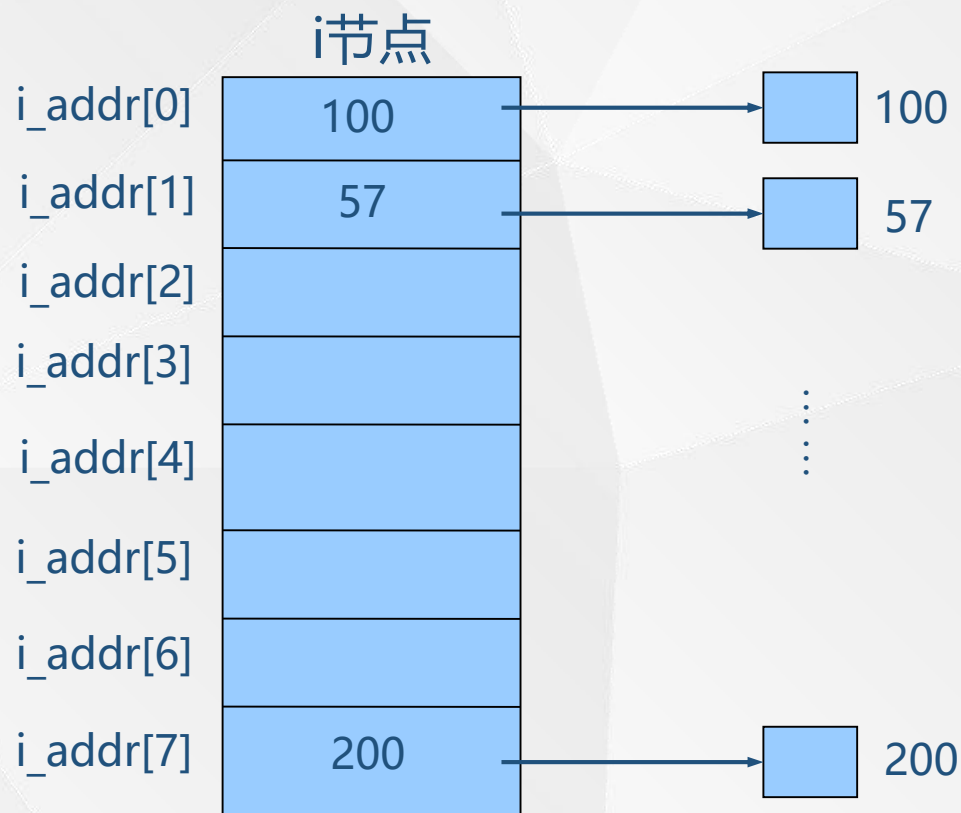
- Inode的大小为128字节，在格式化时就给定，通常每1KB或每2KB就设置一个inode。
- 一个UNIX文件系统中能够创建的文件数量，既受到索引节点区中inode数量的约束，又受到数据区中block数量的约束。



## 2. UNIX系统的索引文件结构

### ① UNIX第七版的文件索引结构

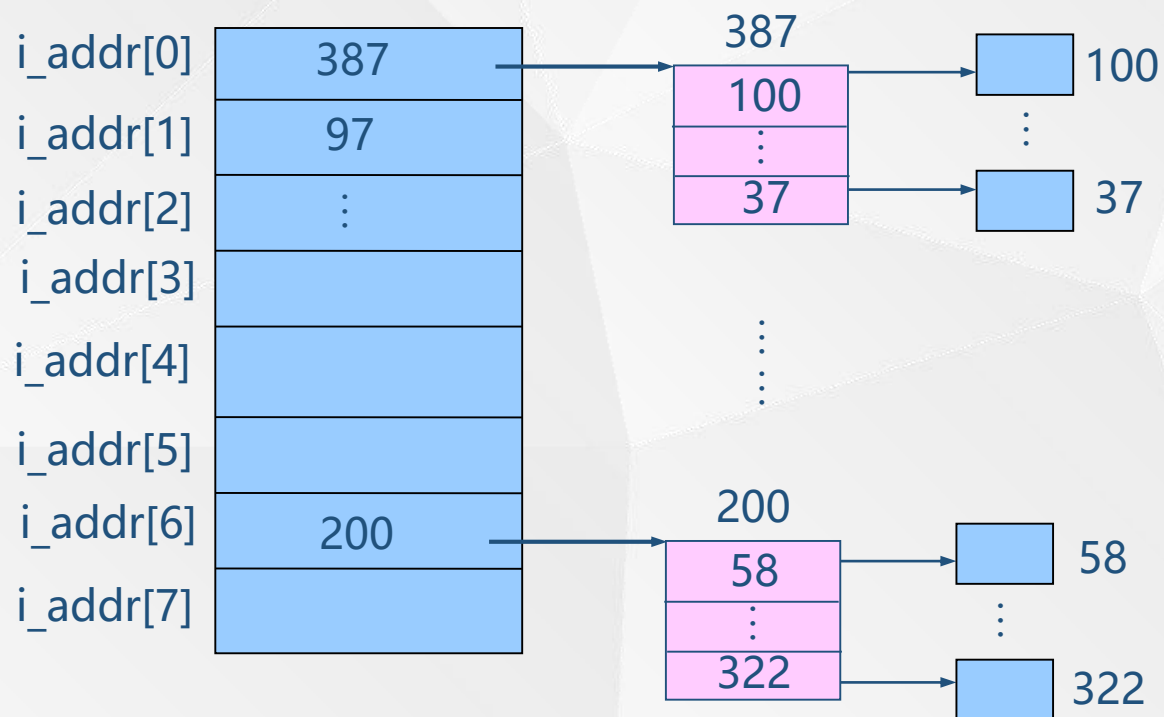
#### • 小型文件



- 在文件i节点中使用一个具有8个数据项的数组i\_addr[] 来描述文件物理结构;
- 构造小型文件时, 数组i\_addr[] 作为直接索引表;
- 小型文件最大为: **8×512B**

## 2. UNIX系统的索引文件结构

### • 大型文件

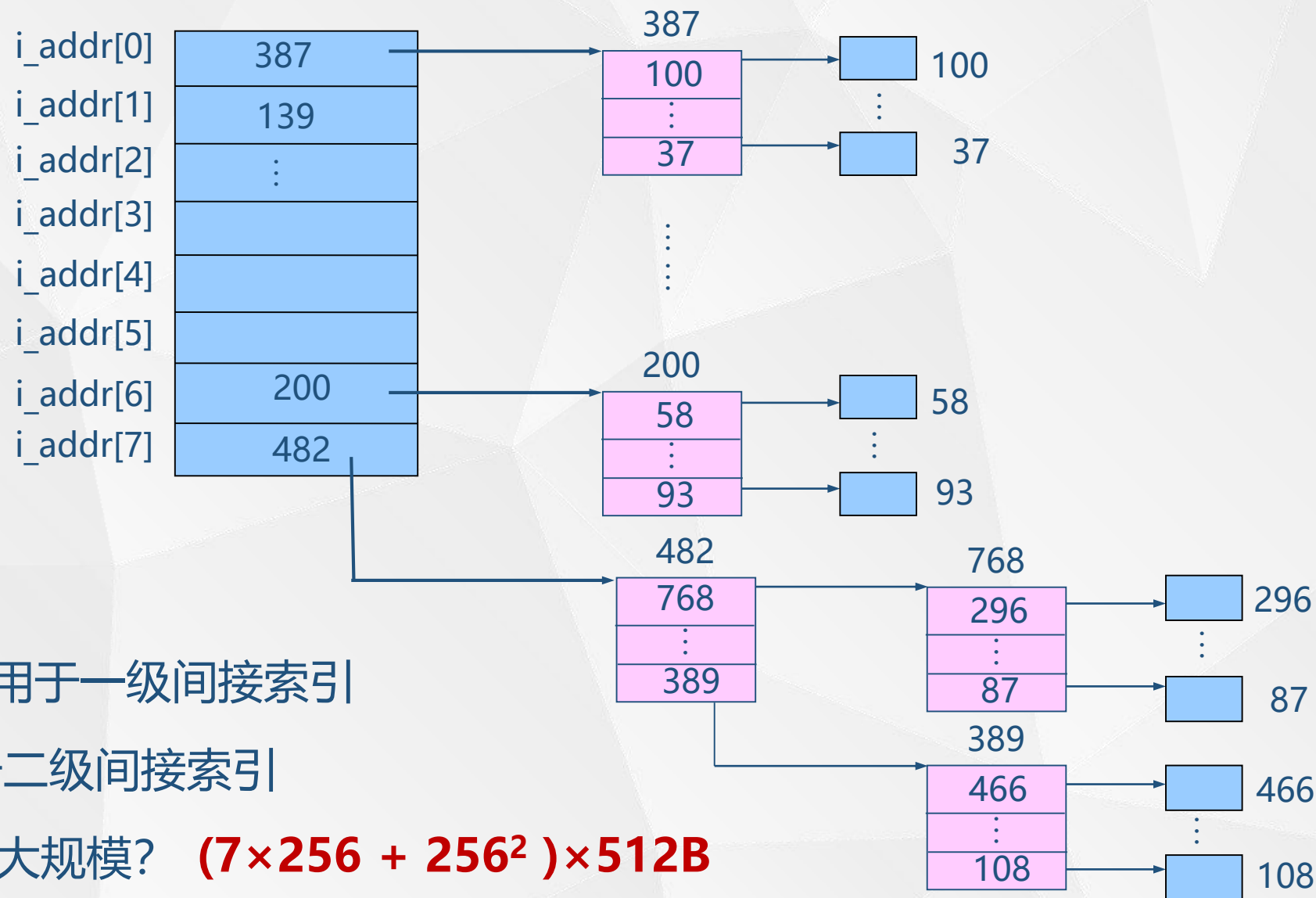


- 数组 `i_addr[]` 用于一级间接索引, 只使用 `i_addr[0] - [6]`
- 大型文件的最大规模:

**$7 \times 256 \times 512B$**

## 2. UNIX系统的索引文件结构

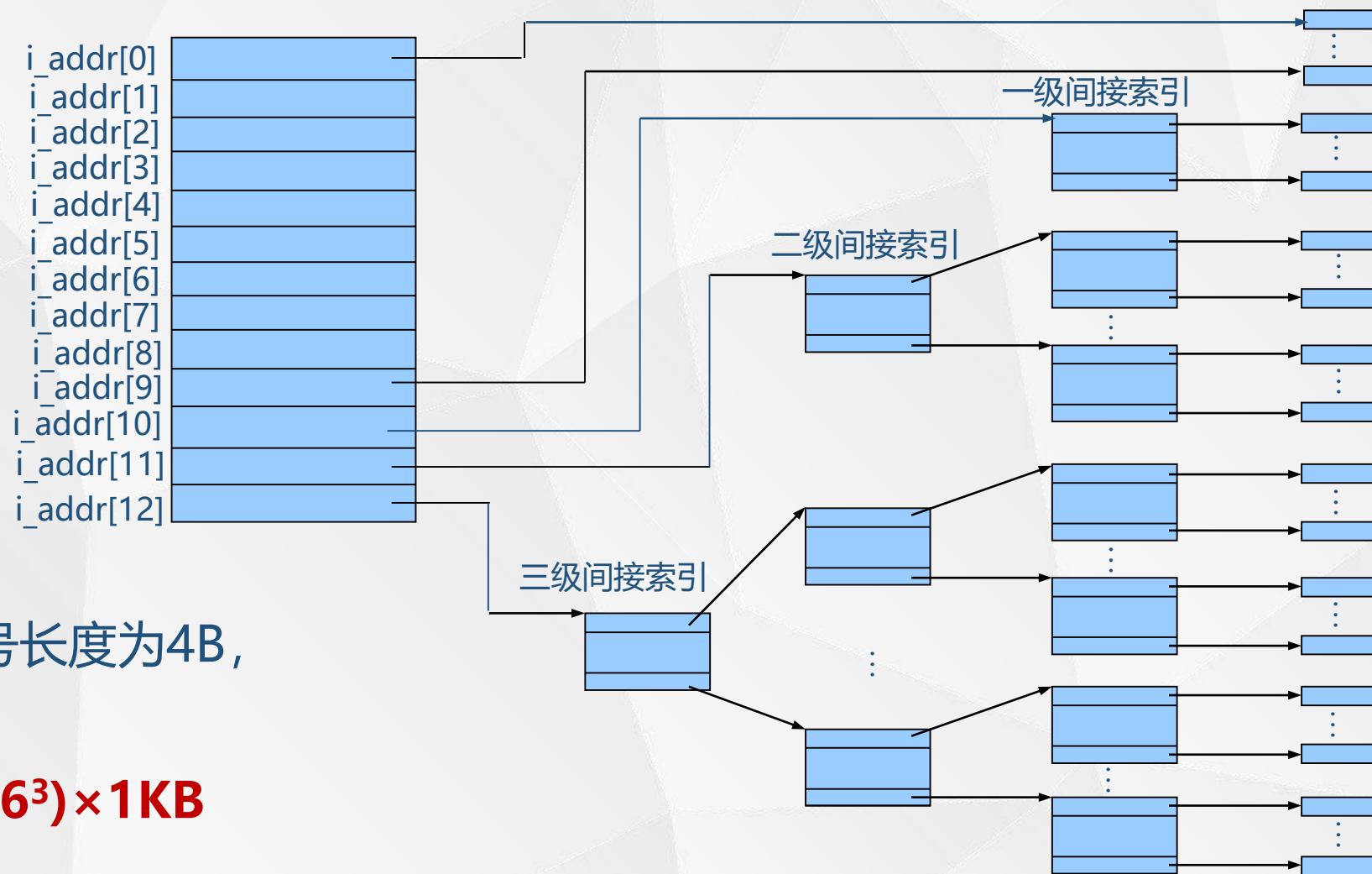
### • 巨型文件



- `i_addr[0]`-[6]用于一级间接索引
- `i_addr[7]`用于二级间接索引
- 巨型文件的最大规模?  **$(7 \times 256 + 256^2) \times 512B$**

## 2. UNIX系统的索引文件结构

### ② UNIX system v 的文件索引结构



若块大小为1KB，块号长度为4B，  
文件的最大规模？

$$(10 + 256 + 256^2 + 256^3) \times 1KB$$

### 3. UNIX系统文件目录结构

#### (1) 目录文件与目录项

##### ① 目录文件

每个目录表为一个目录文件。目录文件由目录项组成。

##### ② 目录项

每个目录项包含16个字节 (UNIX系统老版本)。在目录项中, 第1、2字节为相应文件的辅存i节点号; 后14个字节为文件名。一个辅存磁盘块 (512B)包含32个目录项。



**Q:** 这样设计有什么好处?



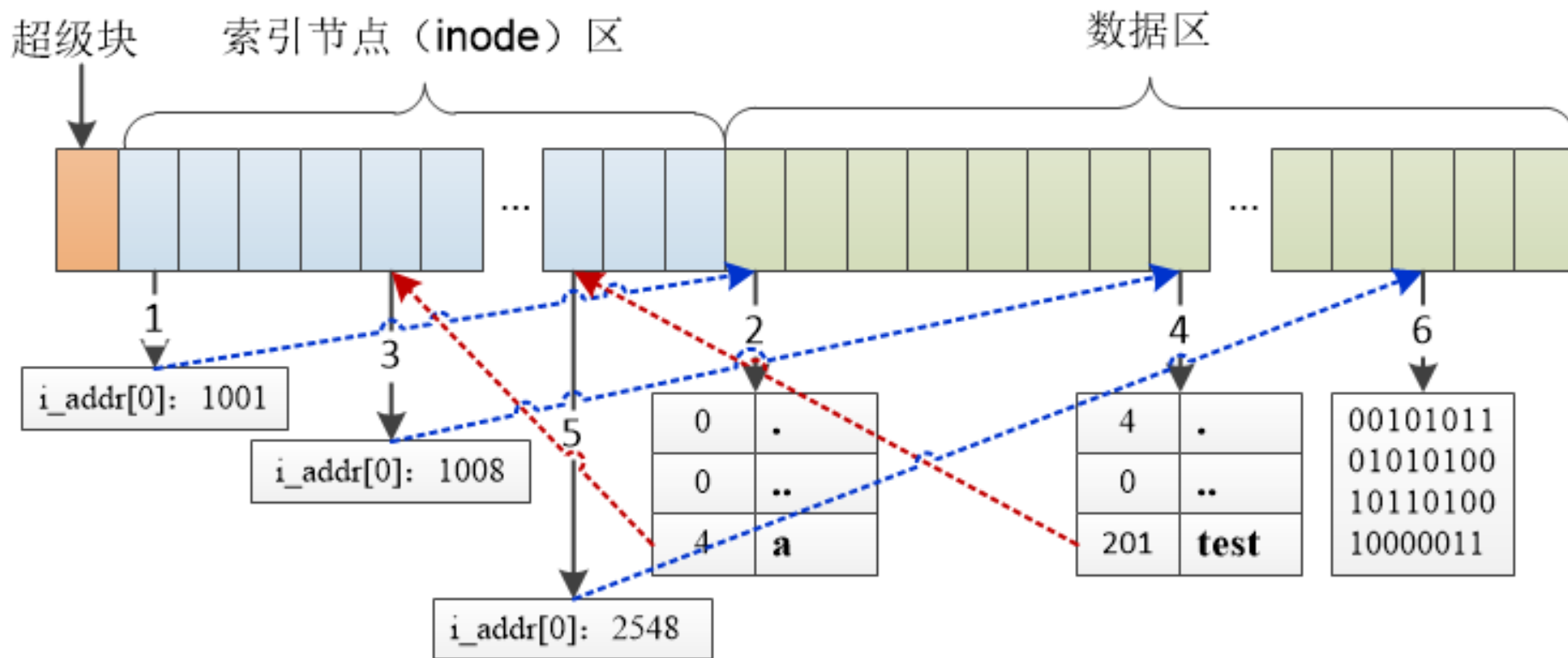
### 3. UNIX系统文件目录结构

#### (2) UNIX系统树型目录结构

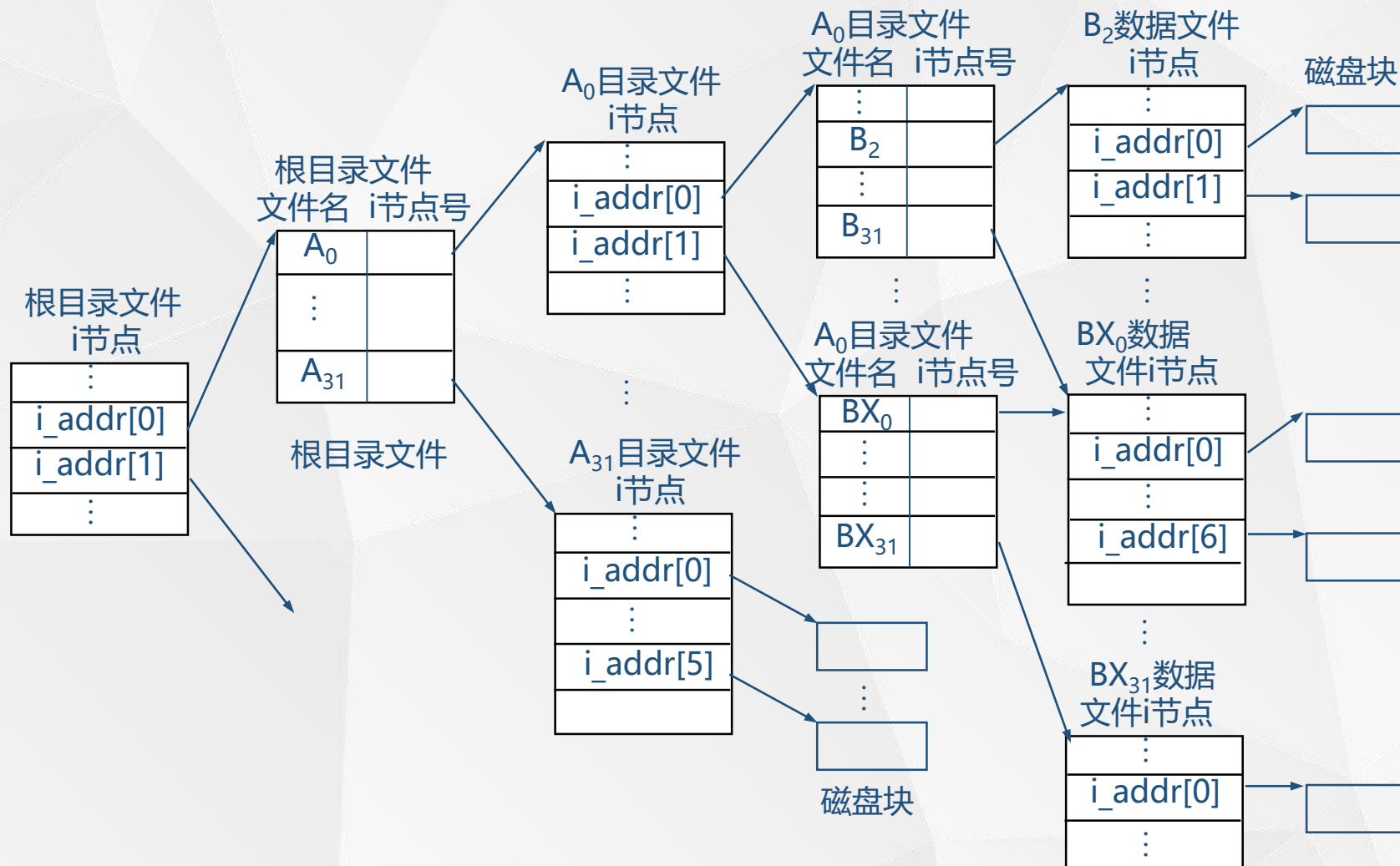
- ① 每个文件系统都有一个根目录文件，它的辅存i节点是相应文件存储设备上辅存索引区中的第一个。
- ② 打开某个文件时，从根目录的i节点可以找到根目录文件的索引结构，得到根目录文件的每个数据块。
- ③ 将待打开文件的路径信息与目录文件中的目录项逐一比较，可以得到下级目录的i节点号，并最终得到目标文件的i节点号。从i节点号中的索引表，得到数据文件的存储块号，实现对目标文件的随机存取。



# 文件 /a/test 查找过程示意图



# 读文件 /A0/B2

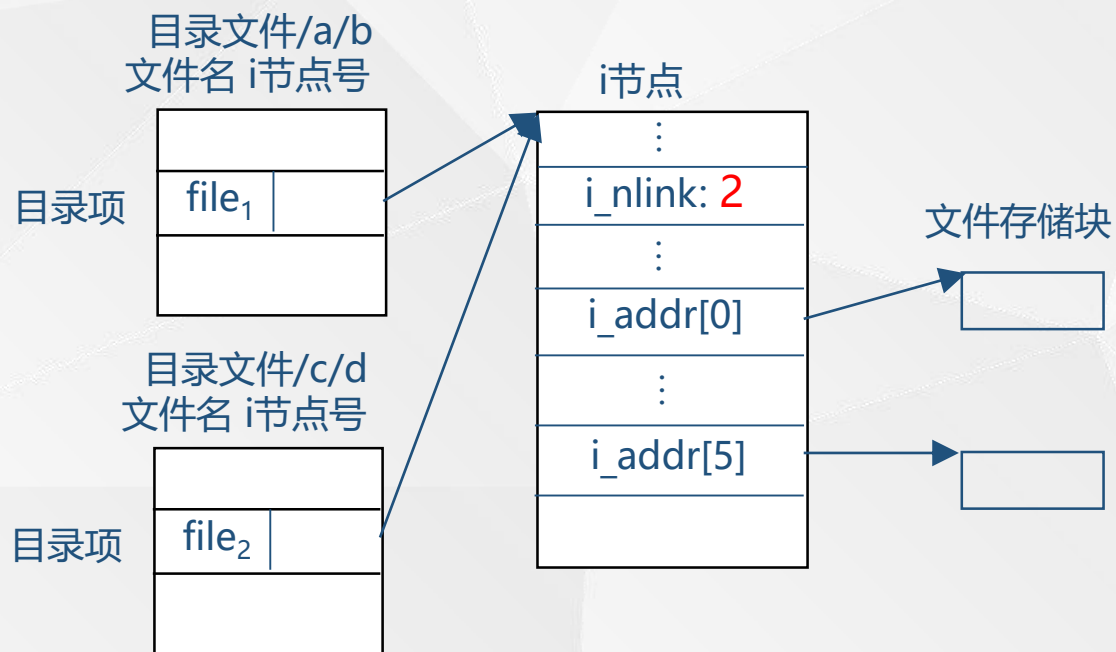


## 3. UNIX系统文件目录结构

### (3) 文件目录结构中的链接

- UNIX文件目录结构中带有交叉链接。用户可以用不同的文件路径名共享一个文件。
- 文件链接在用户看来是为一个已存在的文件另起一个路径名。
- 文件链接的结果表现为一个文件由多个目录项所指向。
- **Q: 如果一个文件被链接, 它有几个inode?**
- **Q: 一个目录文件的链接数至少是几?**

例：一个文件有两个名字：/a/b/file1      /c/d/file2  
这两个文件的目录项同时指向一个数据文件i节点。



**Q：**可能发生多个目录项同时指向一个目录文件的i节点的情况吗？为什么？

不能。所以对目录不能建硬链接。

## 4. UNIX系统的打开文件机构

- 为了提高系统效率，减少主存空间的占用，系统设置了打开文件和关闭文件操作。当打开一个文件时，建立用户与该文件的联系。
- 文件系统中管理这一工作的机构称为打开文件机构。打开文件机构由活动 i 节点表、打开文件表和用户文件描述符表组成。

### (1) 活动 i 节点表

当执行打开文件操作时，将文件辅存 i 节点的有关信息拷贝到主存，形成活动 i 节点（主存索引节点），若干个活动 i 节点组成活动 i 节点表。

**Q：**如果不将 i 节点驻留在内存，会有什么问题？

# 活动i节点（主存索引节点）的结构

主存索引节点状态	i_flag
设备号	i_dev
索引节点号	i_number
引用计数	i_count
文件所有者标识号	i_uid, i_gid
文件类型	i_type
文件存取许可权	i_mode
文件联结数目	i_nlink
文件长度	i_size
文件地址索引表	i_addr[13]

Q: 活动i节点中为什么没有读/写偏移量等信息?



## 4. UNIX系统的打开文件机构

### (2) 系统打开文件表

一个文件可以被同一进程或不同进程，用同一或不同路径名，相同的或互异的操作要求（读、写）同时打开。为了记录打开文件所需的附加信息，文件系统设置了一个内核结构——系统打开文件表。

读写标志	f_flag
引用计数	f_count
指向主存索引节点的指针	f_inode
读/写位置指针	f_offset

### (3) 用户文件描述符表

- 用户进程扩充控制块user中的一个数组u\_ofile[NOFILE] 称为用户文件描述符表，其中的每一项（指针）指向系统打开文件表的一个表项。
- 一个打开文件在用户文件描述表中所占的位置就是它的文件描述符（或称打开文件号）。进程可以打开不同的文件，也可以对同一文件以不同的操作方式打开。

## 4. UNIX系统的打开文件机构

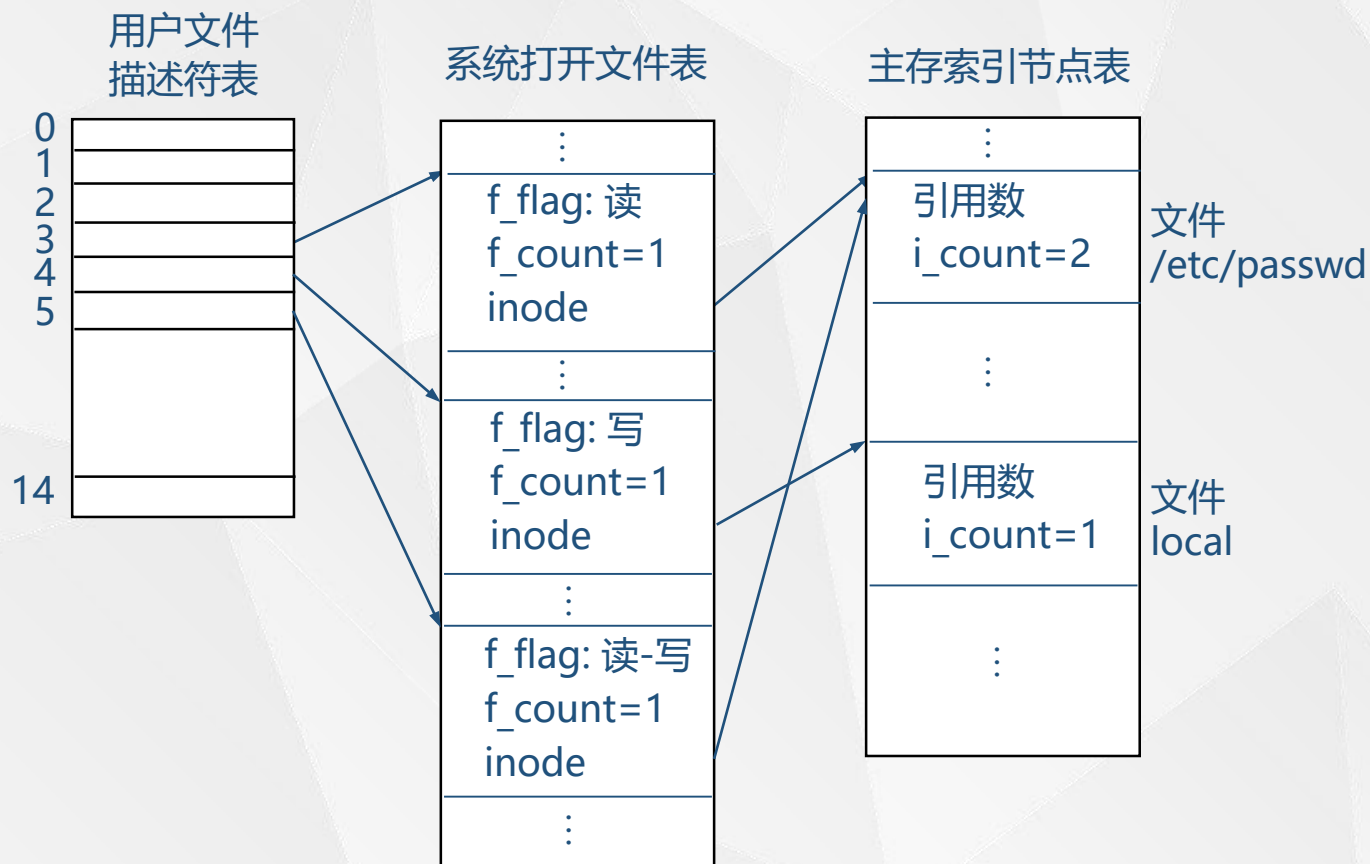
假定一个进程执行下列代码：

```
fd1=open ( "/etc/passwd" , O_RDONLY);
```

```
fd2=open ( "local" , O_WRONLY);
```

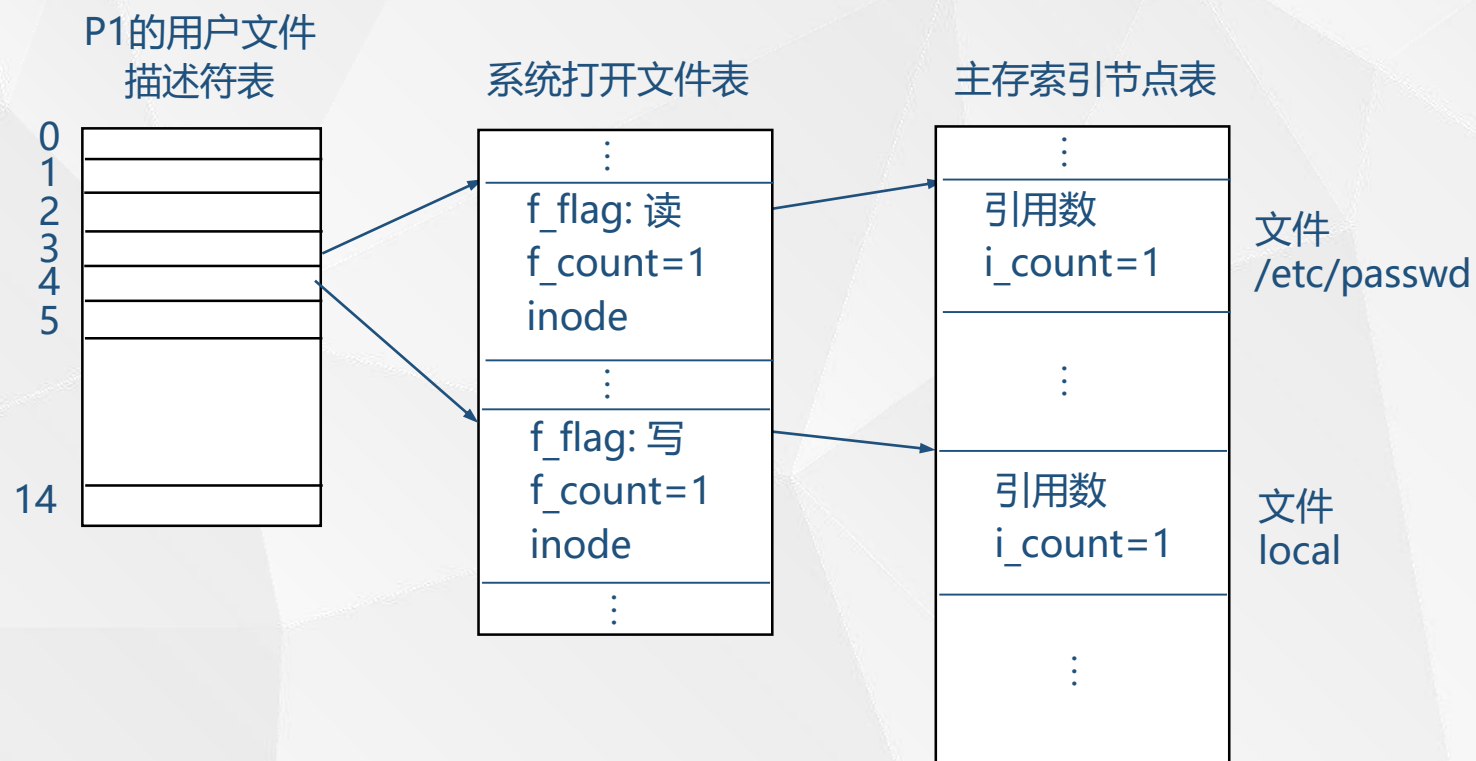
```
fd3=open ( "/etc/passwd" , O_RDWR);
```

**Q：**若此进程此后又创建了一个子进程，此图将有何变化？

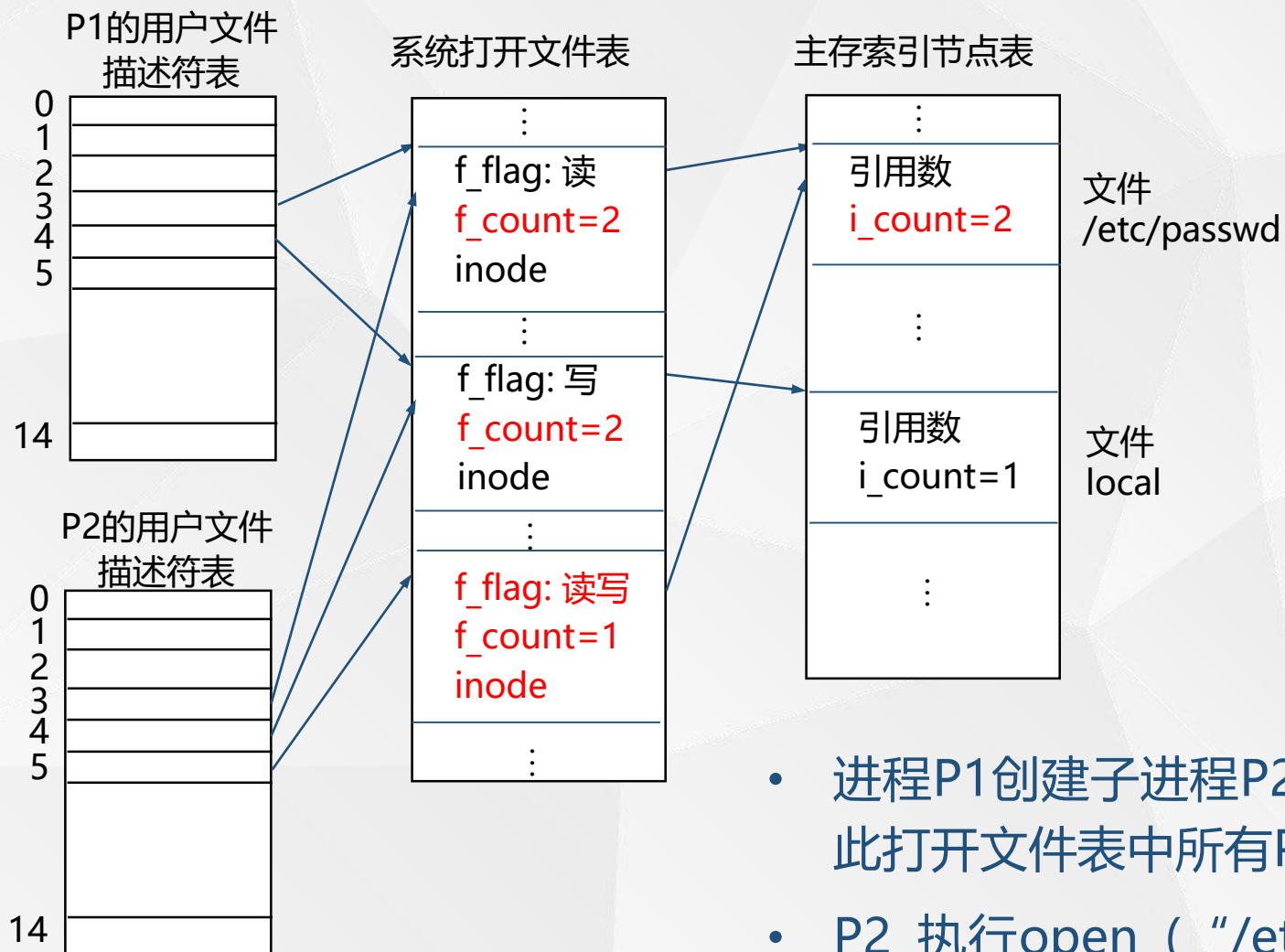


- 子进程共享父进程的“系统打开文件表项”，该表项的文件打开计数fcount加1，子进程直接使用父进程open()操作返回的fd即可访问该文件。
- 父进程的close()操作不影响子进程对该文件的使用，反之亦然。
- 父子进程独立运行后，各自open的文件就不再共享了，跟两个独立进程打开文件情况一样。

课堂练习：一个UNIX系统中，已知进程P1打开文件后的数据结构图如下：



若进程P1此后又创建了一个子进程P2，且P2又执行了打开文件操作 `open ( "/etc/passwd" ,O_RDWR);` 则上图将如何变化？说明原因。



- 进程P1创建子进程P2后，P2继承了父进程的打开文件，因此打开文件表中所有P1的打开文件项的f\_count值均加1。
- P2 执行open ( "/etc/passwd" ,O\_RDWR), 系统会为其在系统打开文件表中新建一个打开文件项。又因为文件/etc/passwd已经被打开过1次，所以只需对其主存索引节点的引用数加1。



## 5. 文件存储器空闲块的管理

### (1) 文件卷和卷管理块

一个文件系统就是逻辑设备，每个逻辑设备占用一片连续的磁盘存储空间。  
文件卷上存放UNIX文件系统。文件卷结构图如下。

引导块	管理块	索引节点区	数据区
-----	-----	-------	-----

- ① **引导块**      大小为一个磁盘块，包含引导程序。
- ② **索引节点区**   索引节点结构组成。
- ③ **数据区**      数据文件占用的区域。
- ④ **管理块**      记录文件系统各种数据，如文件系统大小、空闲块数目等。



```
struct filsys
```

```
{
```

```
    int s_isize;
```

```
    /* i节点区总块数 */
```

```
    int s_fsize;
```

```
    /* 文件卷总块数 */
```

```
    int s_nfree;
```

```
    /* 直接管理的空闲块数 */
```

```
    int s_free[100];
```

```
    /* 空闲块号栈 */
```

```
    int s_ninode;
```

```
    /* 直接管理的空闲i节点数 */
```

```
    int s_inode[100];
```

```
    /* 空闲i节点号栈 */
```

```
    char s_flock;
```

```
    /* 空闲块操作封锁标记 */
```

```
    char s_iloc;
```

```
    /* 空闲i节点分配封锁标记 */
```

```
    char s_fmod;
```

```
    /* 文件卷修改标记 */
```

```
    char s_ronly;
```

```
    /* 文件卷只读标记 */
```

```
    int s_time;
```

```
    /* 文件卷最近修改时间 */
```

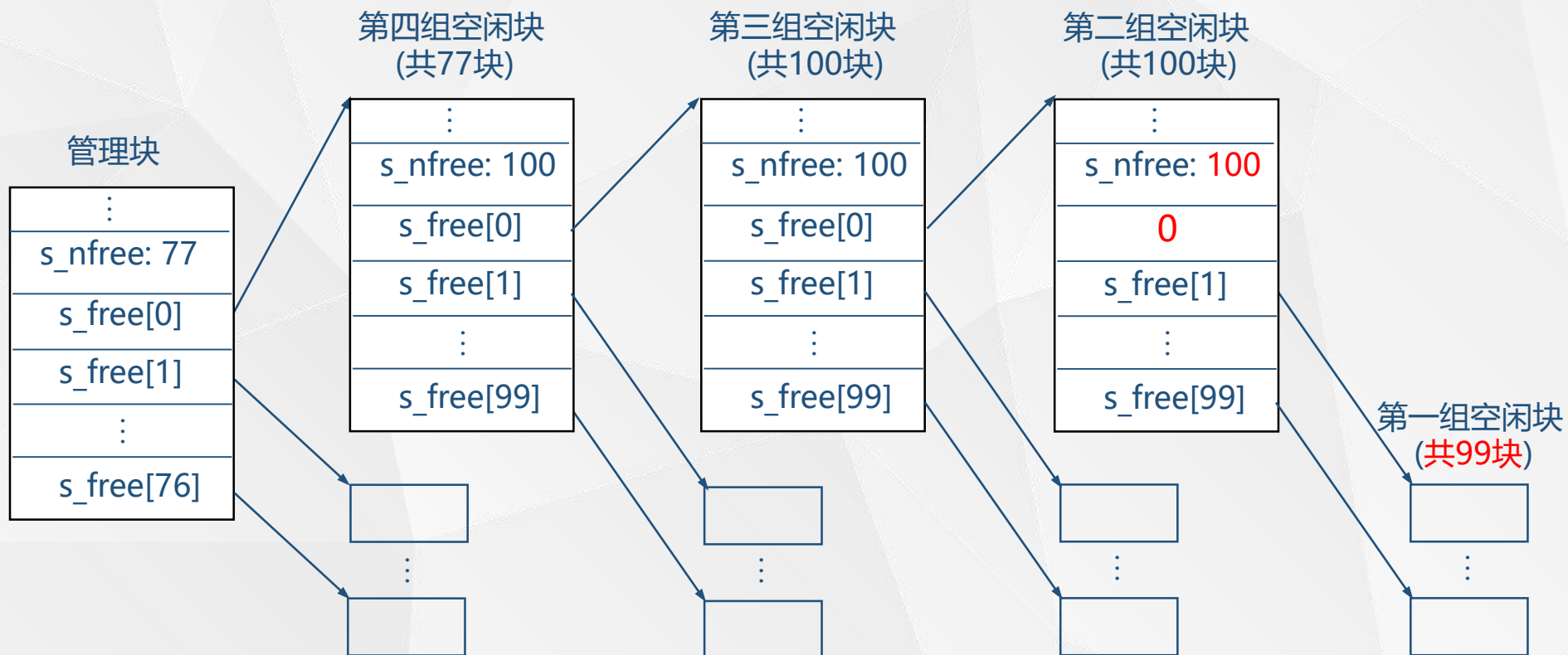
```
}
```

## 5. 文件存储器空闲块的管理

### (2) 空闲磁盘块的管理

- UNIX的空闲磁盘块管理采用**成组链接法**，即将空闲表和空闲链两种方法相结合。
- 系统初启时，文件存储区是空闲的。将空闲块从尾倒向前，每100块分为一组（最后一组为99块），每一组的最后一块作为索引表，用来登记下一组100块的物理块号和块数。
- 最前面一组（可能不足100块）的物理块号和块数存放在管理块的s\_free[100]和s\_nfree中。

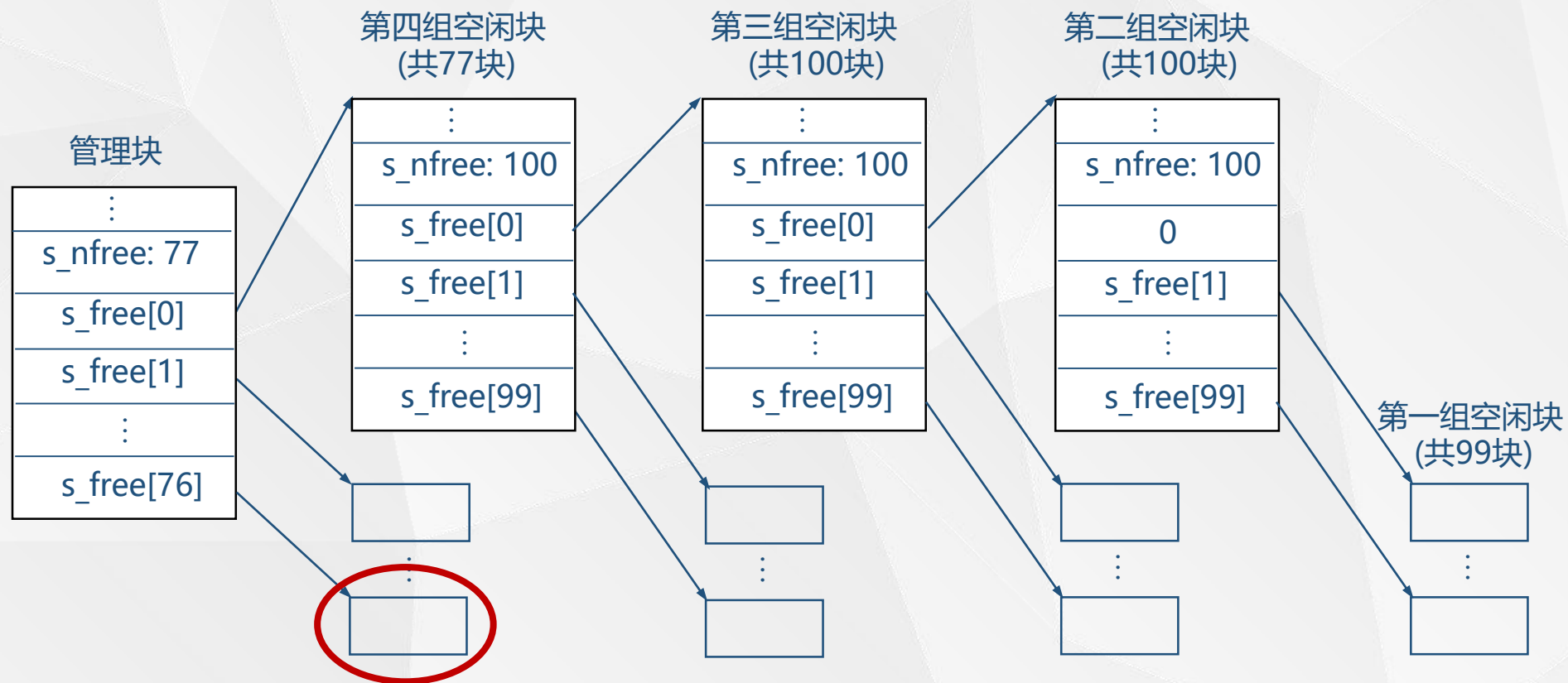
# 空闲磁盘块分组链接索引结构图示



**Q: 图中一共有多少个空闲块?**

- 检查管理块的空闲盘块号栈是否已上锁(临界资源)，若已上锁则进程睡眠等待；
- 给空闲磁盘块号栈上锁后，将s\_nfree减1，若s\_nfree仍大于0，即栈中还剩不止一个空闲盘块，则将s\_free[s\_nfree]中登记的空闲盘块分配出去；
- 若s\_nfree为0，即当前空闲盘块号栈中只剩下最后一个空闲盘块，必须先将该盘块的内容读入管理块的空闲盘块号栈中，然后再将该盘块分配出去。若s\_nfree为0，而且栈底登记的盘块号为0，则表示系统中无空闲盘块可分配。

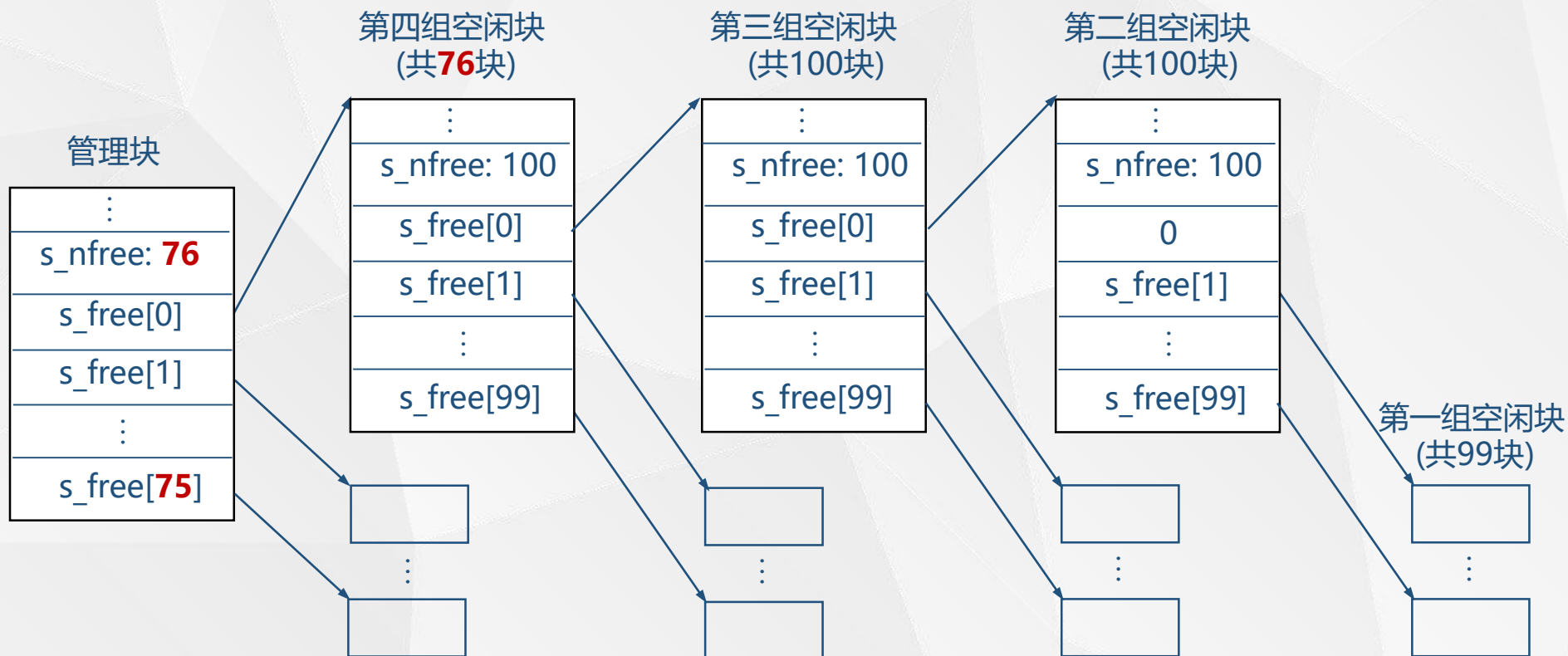
# || 示例：分配一个磁盘块



Q: 哪个空闲块被分出去?



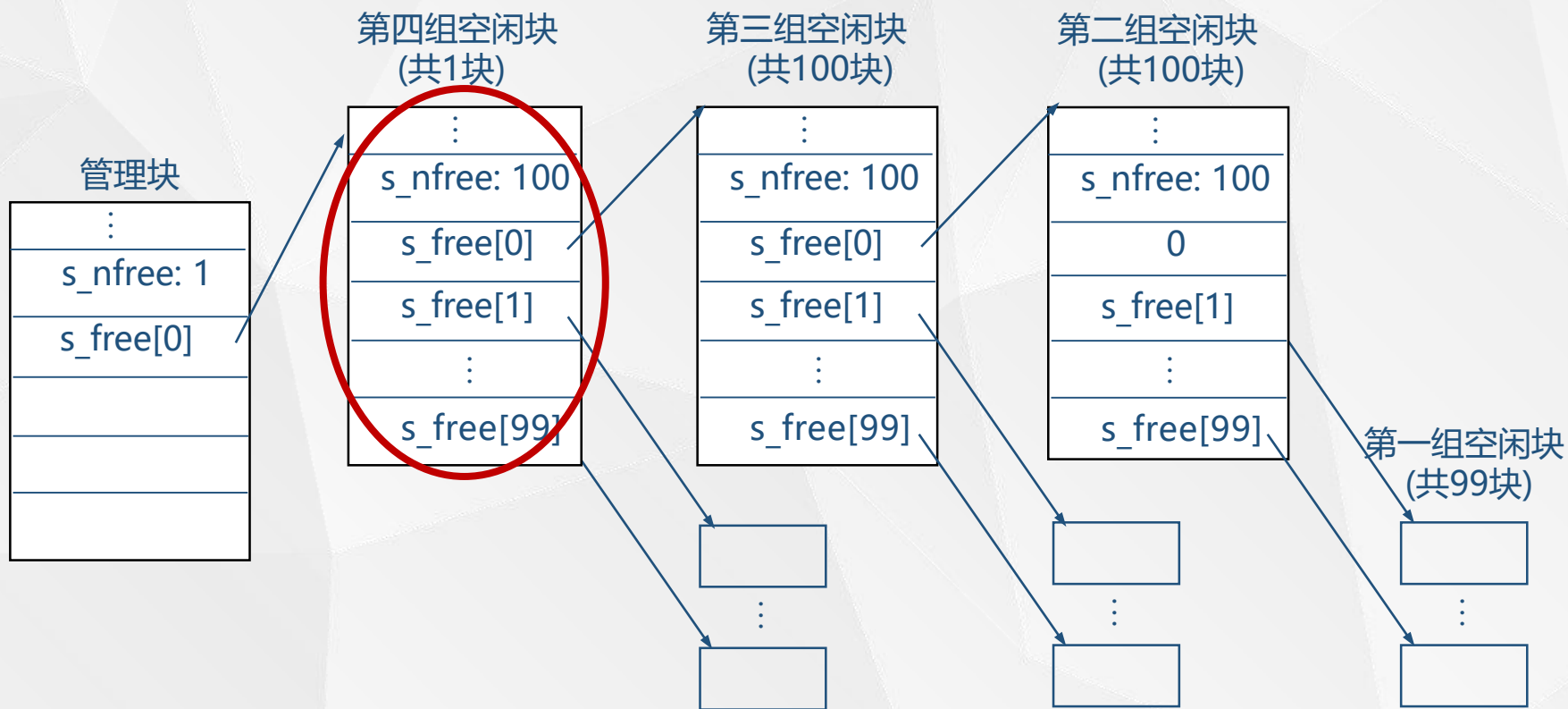
# || 示例：分配一个磁盘块



s\_free[76]被分配

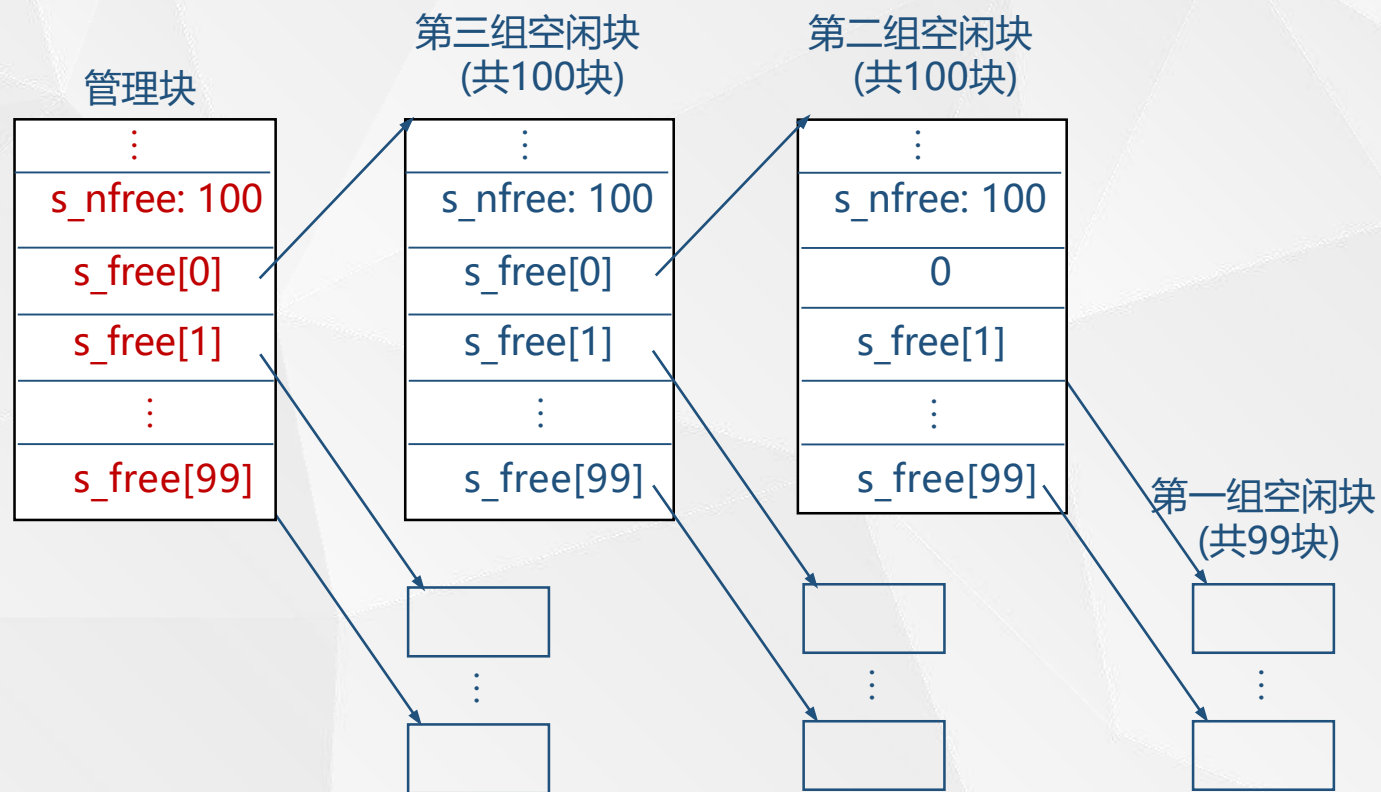


# || 示例：分配当前组最后一块



Q: 哪个空闲块被分出去?

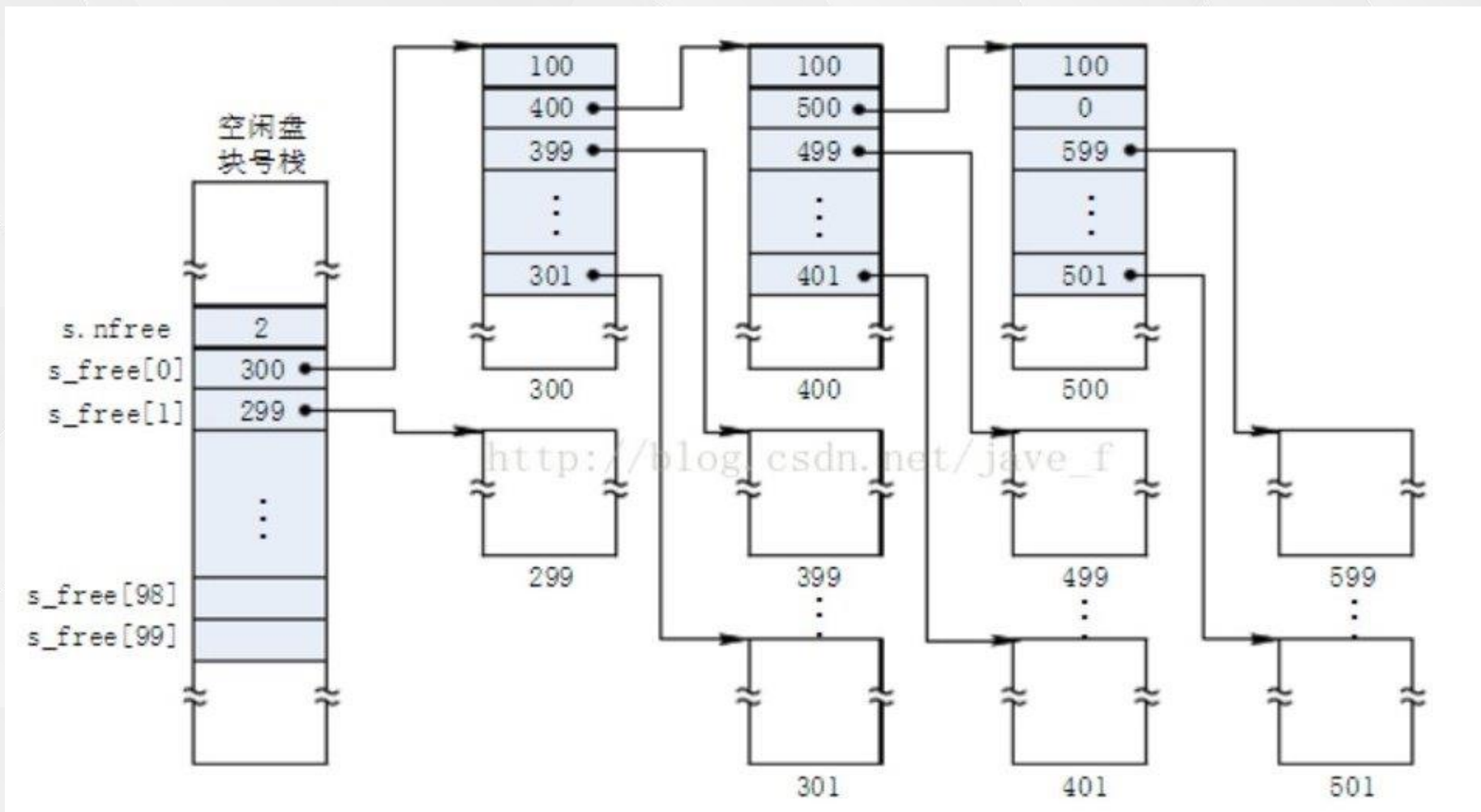
## || 示例：分配当前组最后一块



s\_free[0]被分配，第三组空闲块的信息进入管理块。

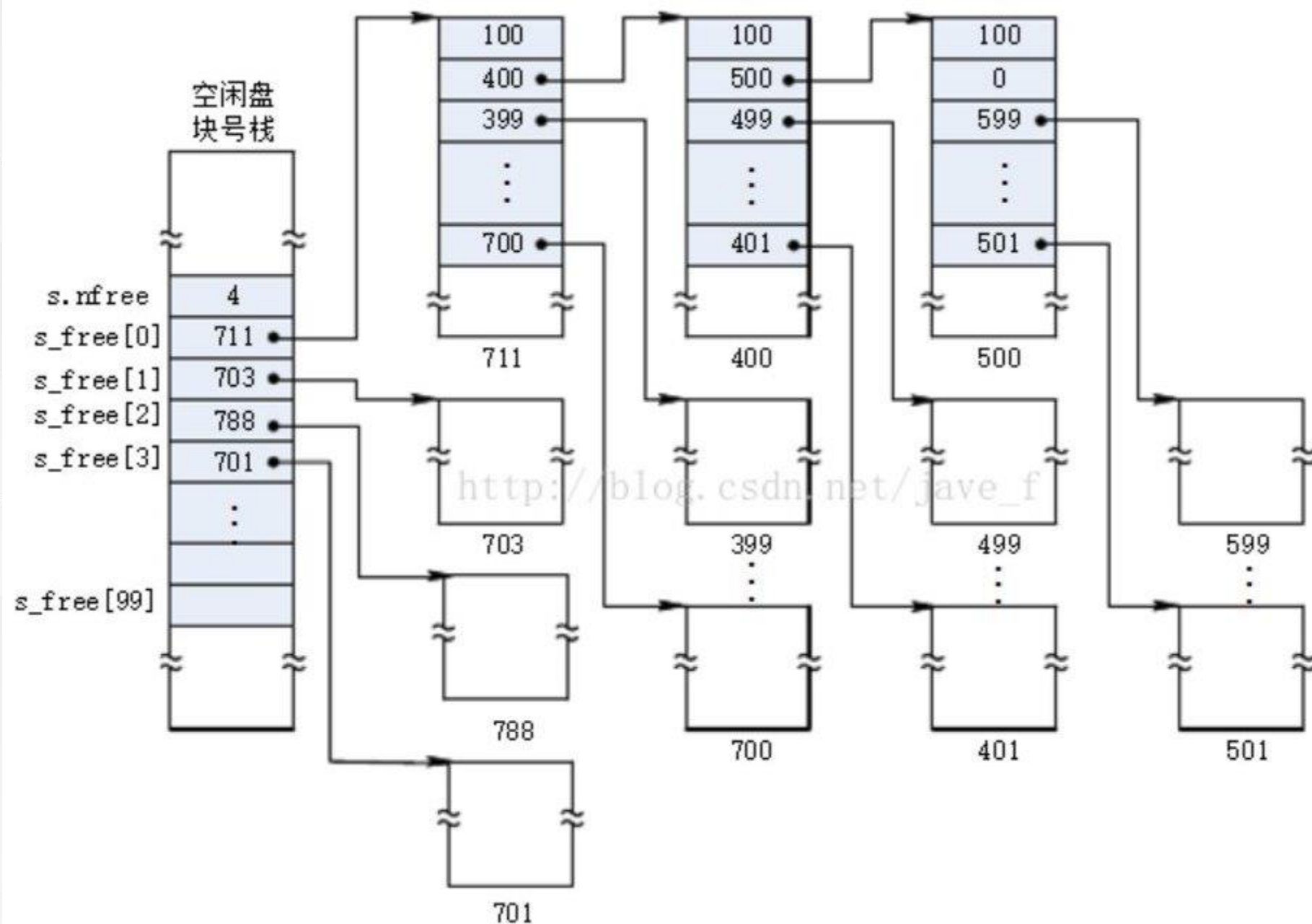
**Q：** 如果已经分配到最后一组的最后一块了呢？

- 在系统回收空闲盘块时，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。



某系统采用成组链接法来管理磁盘的空闲空间，目前磁盘的状态如图所示。

- (1) 该磁盘中目前还有多少个空闲盘块？
- (2) 在为某个文件分配3个盘块后，系统要删除另一文件，并回收它所占的5个盘块，它们的盘块号依次为700、711、703、788、701，请画出回收后的盘块链接情况。





## ■ 文件系统基本概念

- 文件、文件系统

## ■ 文件的逻辑结构及存取方法

- 文件的逻辑结构：流式文件、记录式文件
- 文件的存取方法：顺序存取、随机存取

## ■ 文件的物理结构

- 连续文件
- 串联文件
- 索引文件：直接索引、一级间接索引结构、二级间接索引结构



## ■ 文件目录

- 文件目录项内容
- 重名问题的解决办法
- 树型文件目录
- 文件路径名，当前目录

## ■ 文件共享与安全

- 文件共享：存取权限验证方法
- 文件安全的定义
- 用文件名加快文件的查找：建立当前目录、链接技术

## ■ 文件操作与文件备份

- 文件操作：常用的文件操作命令、“打开文件”与“关闭文件”
- 文件备份：周期性备份、增量转储

## ■ UNIX文件系统的主要结构及实现技术

- UNIX系统的索引文件结构：文件目录项组成、**文件索引节点**（磁盘索引节点）
- UNIX 7 的文件索引结构：小型文件结构、大型文件结构、巨型文件结构
- **UNIX system V 的文件索引结构**
- UNIX系统的文件目录结构：UNIX树型目录结构（支持不同文件路径名共享一个文件）
- UNIX系统的**打开文件结构**：活动i节点表、系统打开文件表、用户文件描述符表
- 文件存储器空闲块的管理
  - ◆ UNIX文件系统磁盘存储区(文件卷)的结构
  - ◆ 空闲磁盘块的管理：**成组链接法**