



西安电子科技大学
XIDIAN UNIVERSITY



计算机科学与技术学院
SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY
国家示范性软件学院
NATIONAL PILOT SCHOOL OF SOFTWARE ENGINEERING

操作系统原理

第六章 文件管理

主讲：黄伯虎



基本内容	<ol style="list-style-type: none">1. 基本概念：地址重定位2. 连续分配方式：<ul style="list-style-type: none">• 分区存储管理方案3. 离散分配方式<ul style="list-style-type: none">• 页式存储管理方案• 段式存储管理方案• 段页式存储管理方案4. 内存扩充技术<ul style="list-style-type: none">• 交换和覆盖技术• 虚拟存储管理方案<ul style="list-style-type: none">• 请求分页管理方式• 页面置换算法：FIFO , LRU
重难点	<ul style="list-style-type: none">• 程序的地址空间（虚拟地址空间）• 地址重定位（动态重定位）• 页式、段式管理方案原理• 页面置换算法



一、基本概念



文件(file)



磁带



磁盘



光盘



优盘



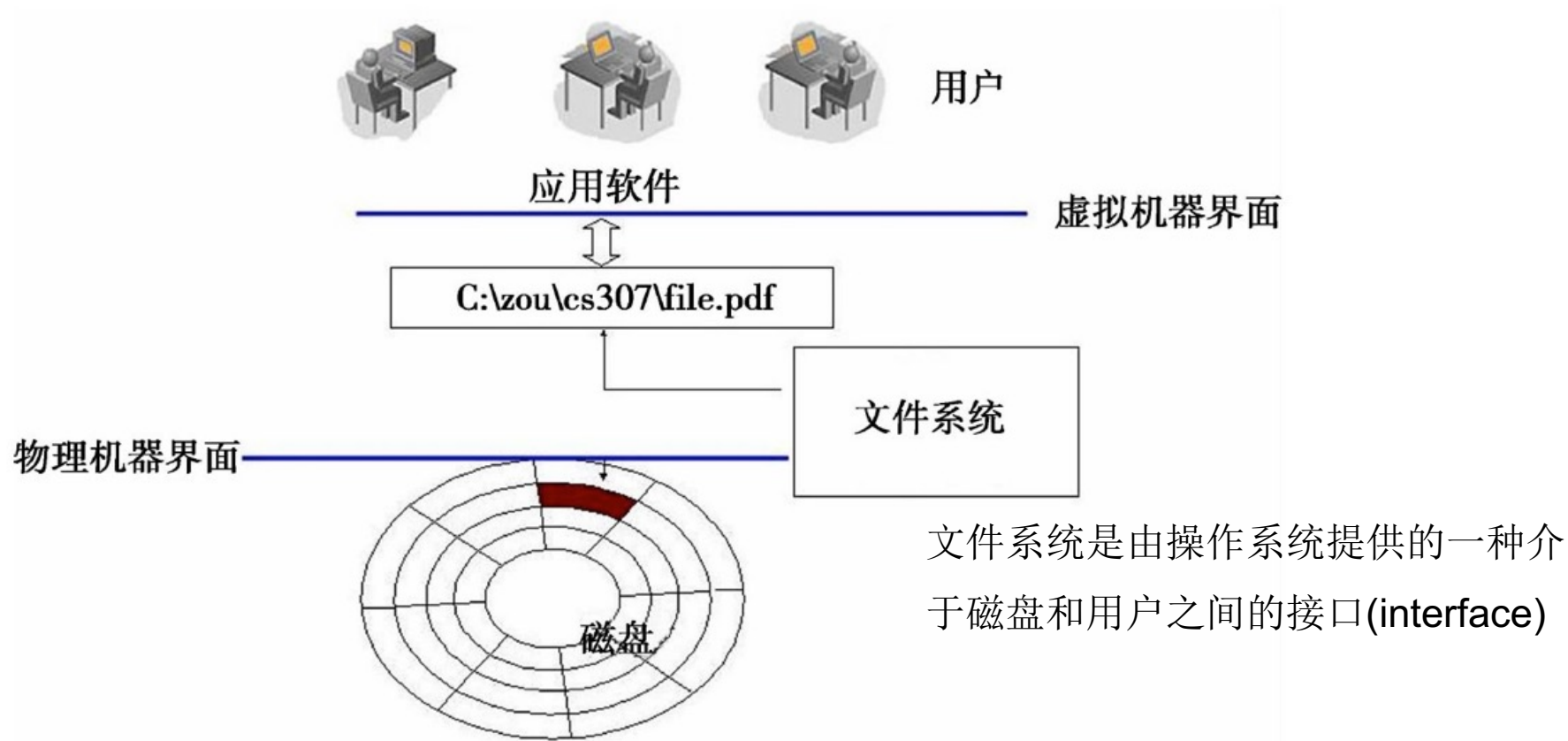
固态硬盘

定义：是记录在**外存**上的，具有**符号名**的，在逻辑上具有完整意义的一组相关信息项的集合。

从用户的角度看，文件是**逻辑外存**的**最小分配单元**，即信息(数据)只能以文件的形式写入外存。



一、基本概念



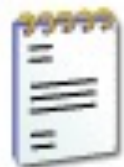
文件和文件系统本身就是对物理外存的抽象!



一、基本概念



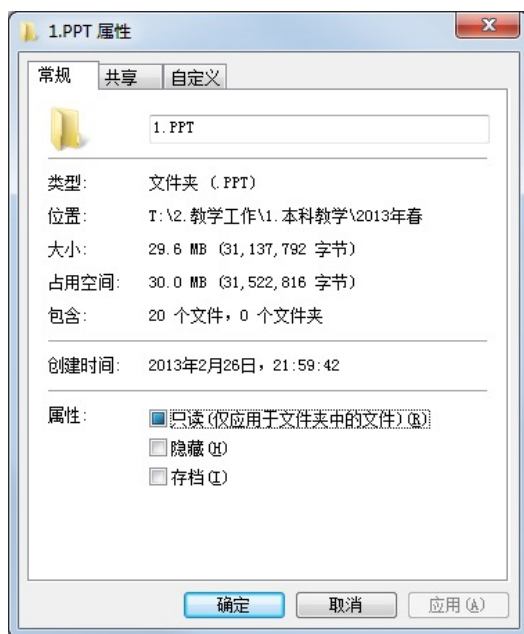
文件的组成部分



sort.c

文件体：文件真实的内容。

文件说明(属性)：操作系统为了管理文件所用到的信息。



- 文件名称
- 文件内部标识符
- 文件类型
- 文件存储位置
- 文件大小
- 访问权限
- 时间、日期、用户标识
-

文件目录



一、基本概念

文件的类型

❖ 分类的目的:

➤ 对不同文件进行区分和管理，提高系统效率

文件的后缀

exe, com, bin: 可执行文件
c, cc, cpp, java, pas: 程序源代码
bat, sh: 命令文件
txt, doc: 文本数据，文本文档
lib, dll: 库文件
zip, rar: 压缩文件
mpeg, mov, rm: 多媒体文件
.....

按照文件性质和用途

系统文件
库文件
用户文件

信息保存期限:

临时文件
档案文件
永久文件

文件的保护方式

只读文件
读写文件
可执行文件
不保护文件



一、基本概念



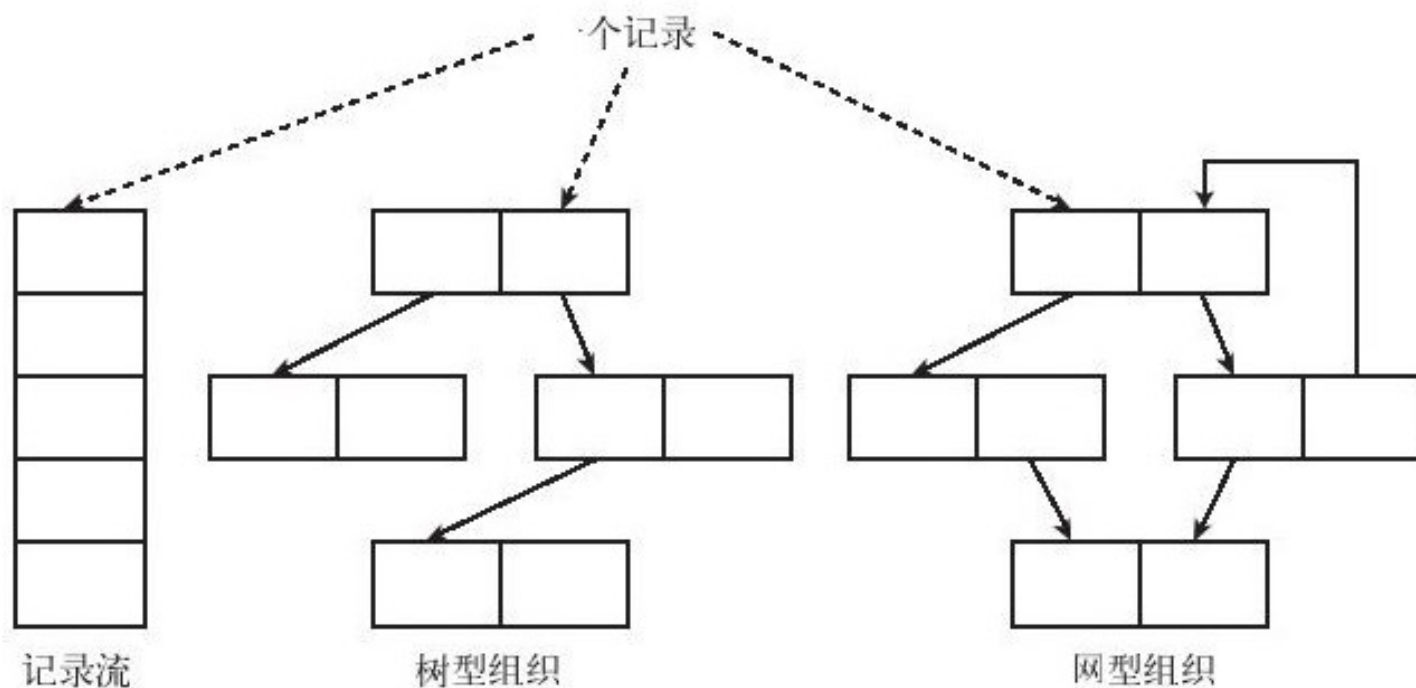
文件系统的类型

FAT16	MS-DOS,win95操作系统使用的文件系统。文件名采用8.3格式，支持分区最大2G，单个文件2GB。
FAT32	Windows 95以上版本支持的文件系统，支持分区最大128GB，单个文件4GB，增加了对长文件名（最多到255个字符）支持。
NTFS	Windows NT操作系统使用的文件系统。它具有较强的安全性，最大支持分区容量2TB，单个文件2TB。
Ext2/Ext3	Linux 系统中的文件系统，其特点为存取文件的性能极好，对于中小型的文件更显示出优势。单一文件大小上限为1TB，而文件系统的容量上限为4TB。
HPFS	OS/2操作系统使用的文件系统。突破了FAT文件系统的一些限制，极大地改善了文件在大目录下的访问时间，支持2TB硬盘容量。
CDFS/UDF	大部分的光盘的文件系统，只能在CD-R或CD-RW上读取。
.....	

二、文件的结构

逻辑结构

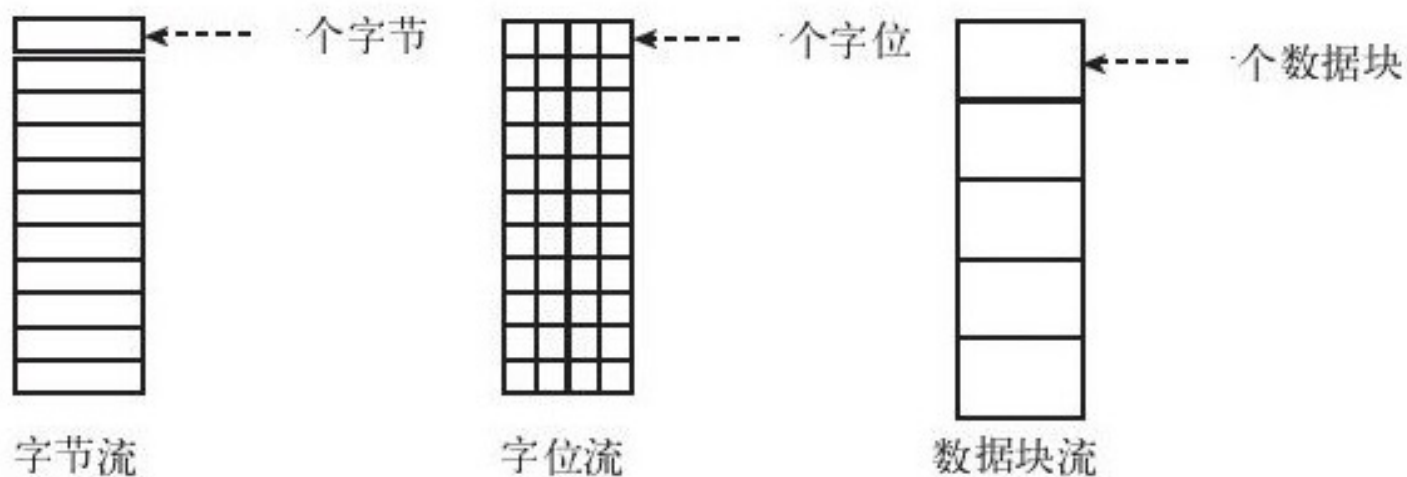
- ❖ 定义：是用户所观察到的文件内容组织形式，它独立于物理存储设备。
- ❖ 分类：
 - 有结构的文件（关系导向型结构）





二、文件的结构

➤ 无结构的文件（非关系导向型结构）



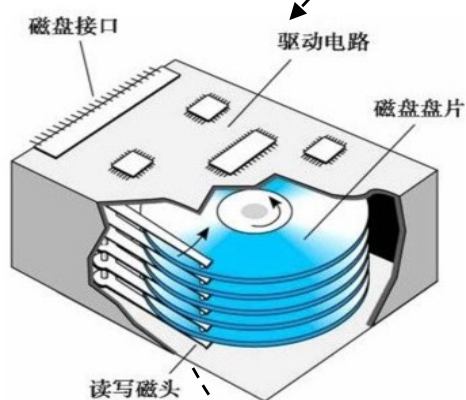
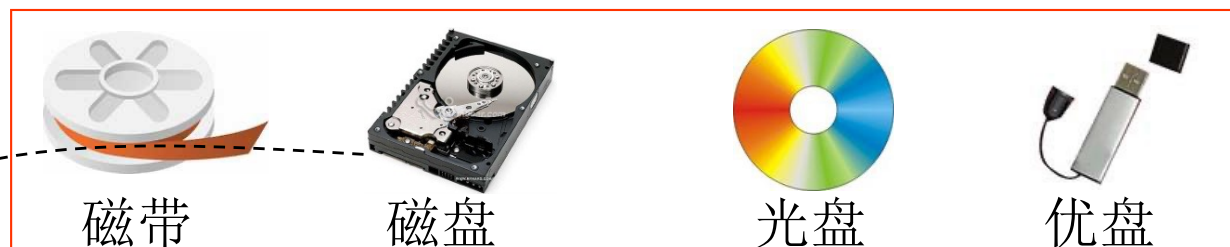


二、文件的结构

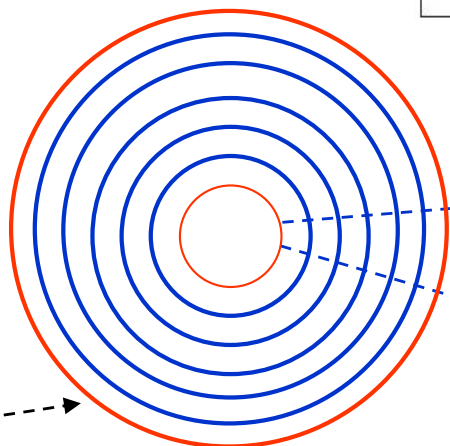


物理结构

- ❖ 定义：是指文件的内部组织形式，即文件在物理存储设备上的存放方法。



标题 (10B)	数据部分 (512B)	ECC纠错信息 (12B~16B)
----------	-------------	-------------------



扇区：一般为600字节，其中512字节存放数据，其余存放控制信息。一个或多个扇区组成簇，称为物理块。

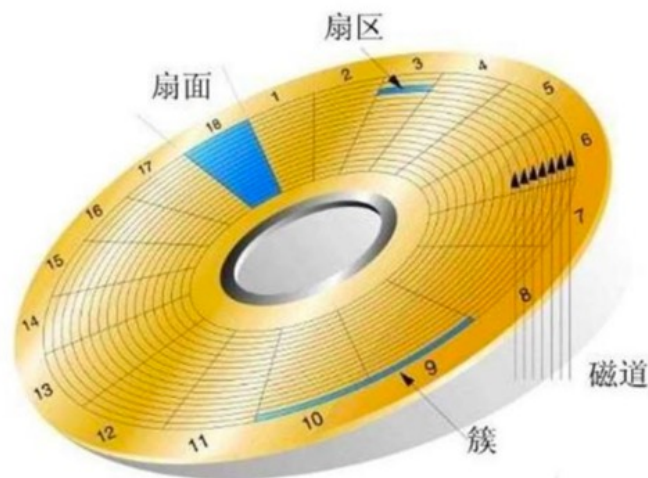


二、文件的结构

❖ 操作系统管理磁盘数据的基本单位

- 数据块 / 磁盘块 (block)
- 通常为扇区的整数倍 (2的幂次方倍)
- 操作系统给出的访问地址: 逻辑地址 (LBA, Logical Block Address)

设备号4	磁头号4	磁柱号16	扇面号8	扇面计数8
------	------	-------	------	-------



二、文件的结构

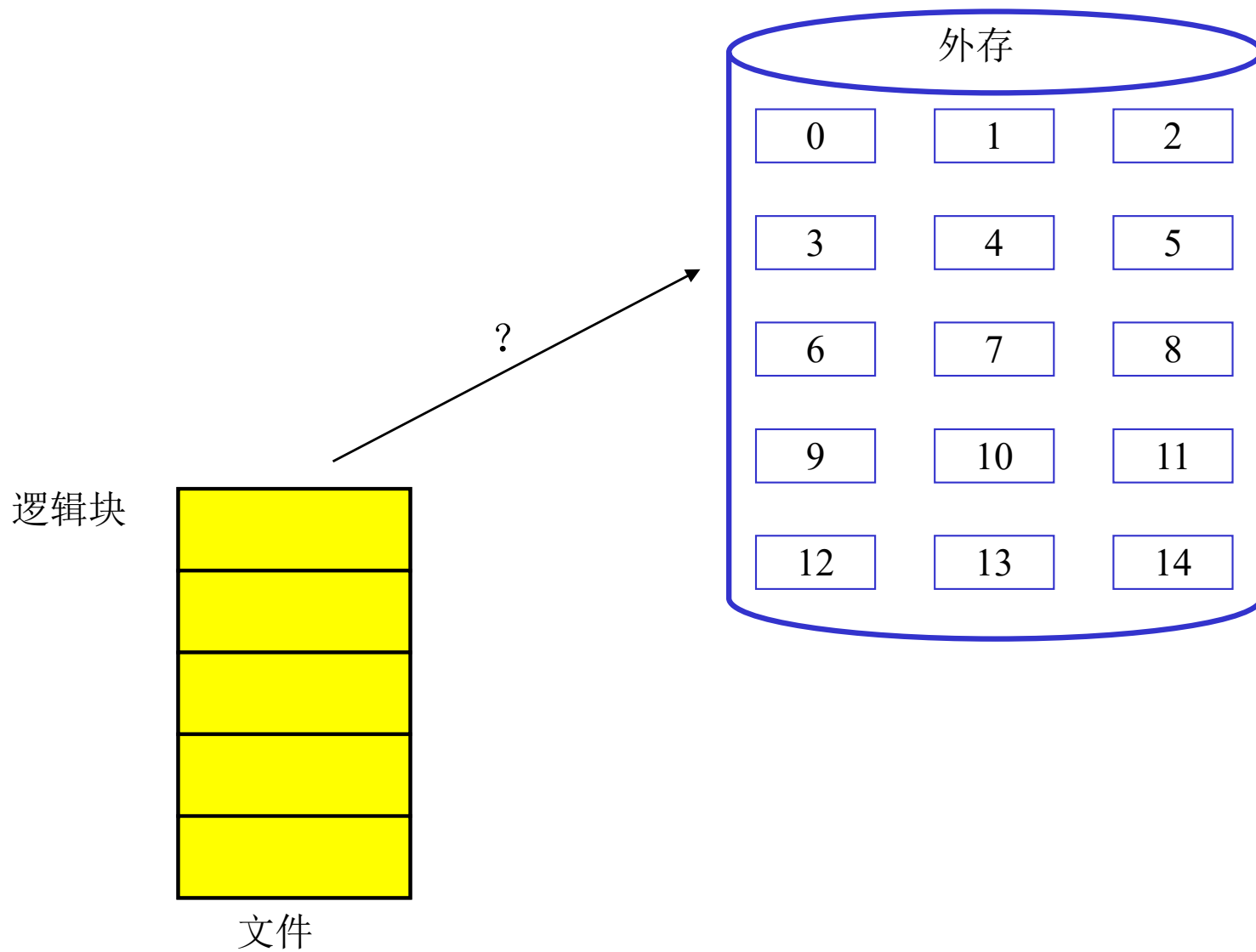


操作系统从磁盘读取数据的过程





二、文件的结构





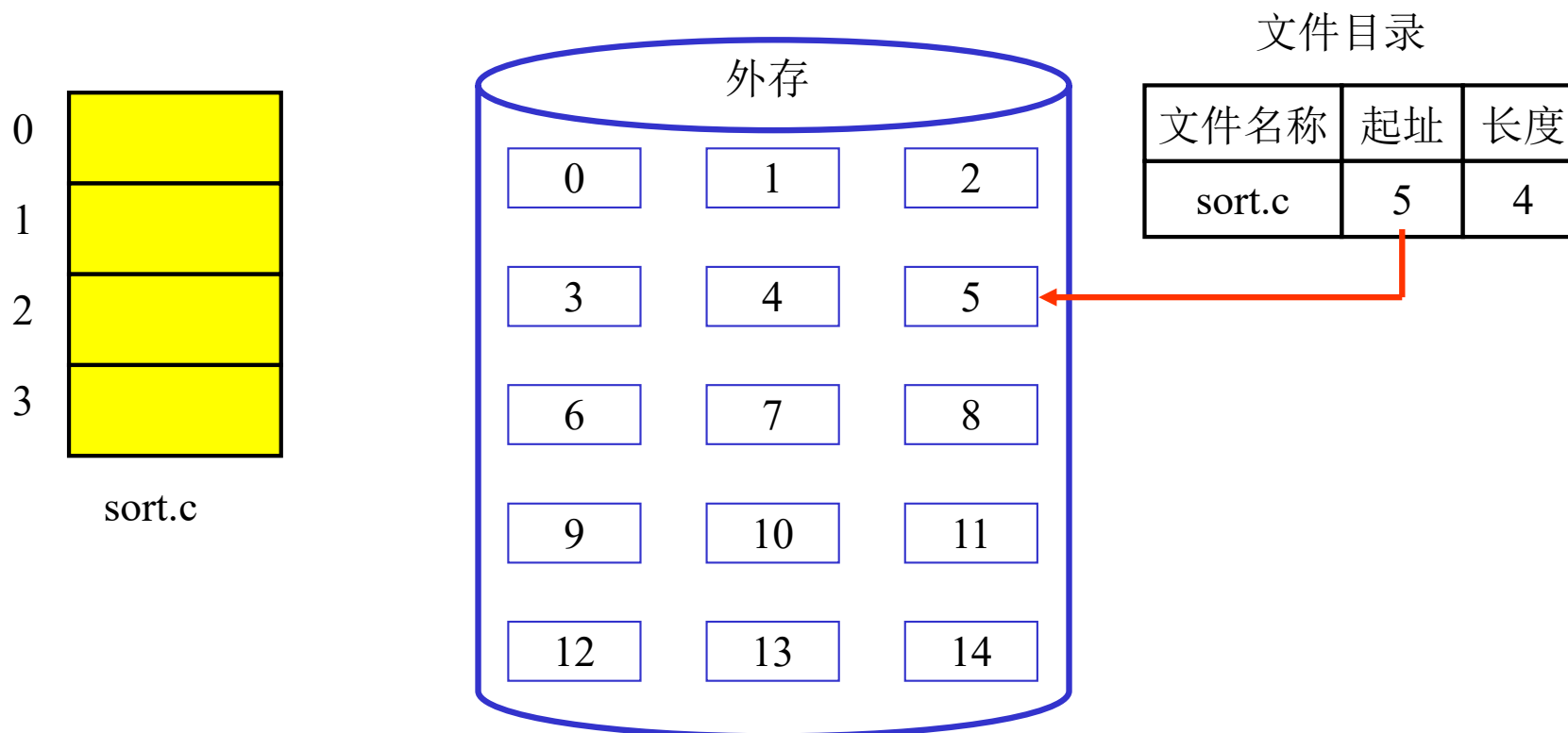
二、文件的结构



几种常见的物理存储方式

❖ 连续存储(顺序结构)

➤ 它将逻辑上连续的文件信息依次存放在编号连续的物理块上。

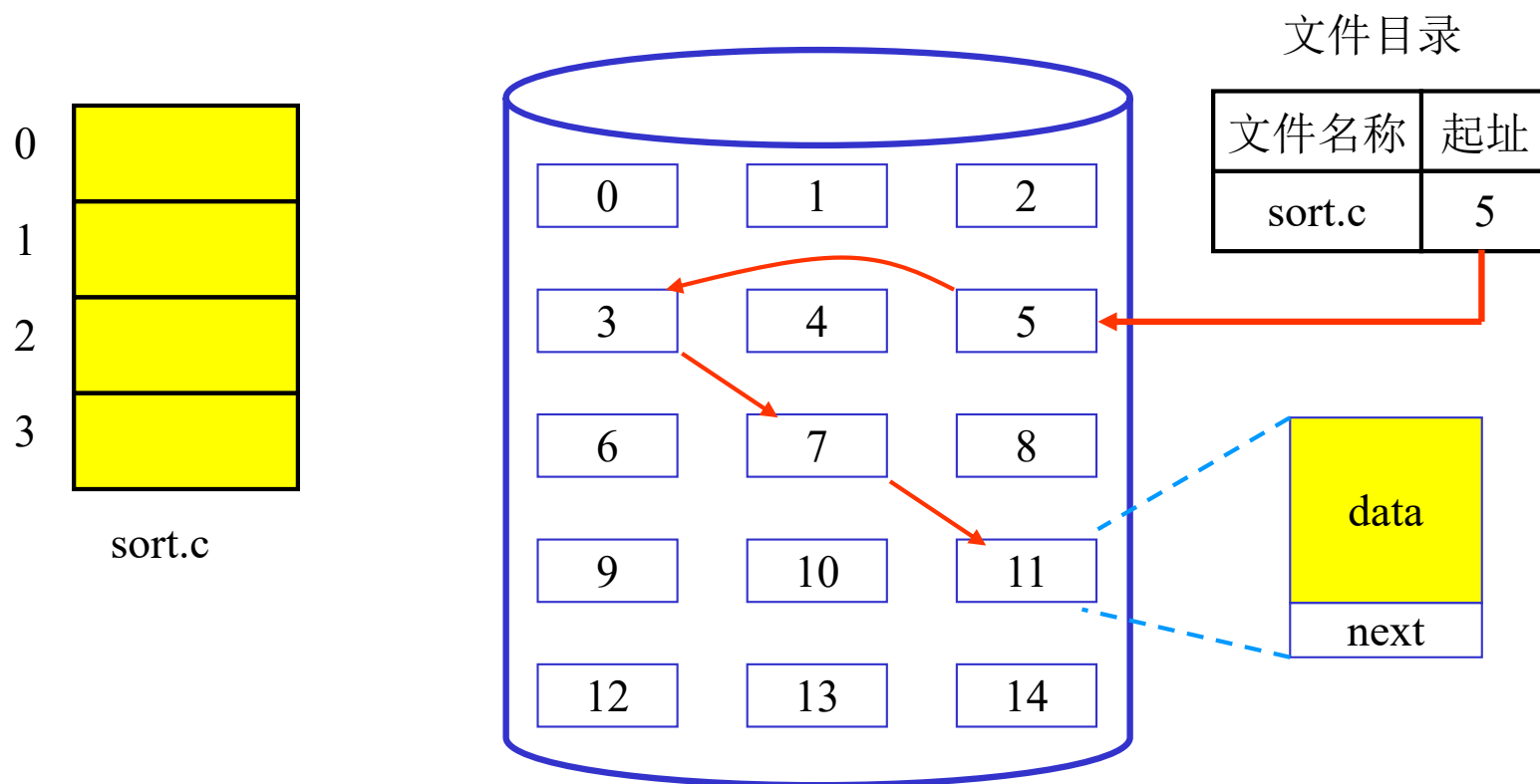




二、文件的结构

❖ 链接结构

- 将逻辑上连续的文件信息存放在不连续的物理块上，每个物理块设有一个指针指向下一个物理块。

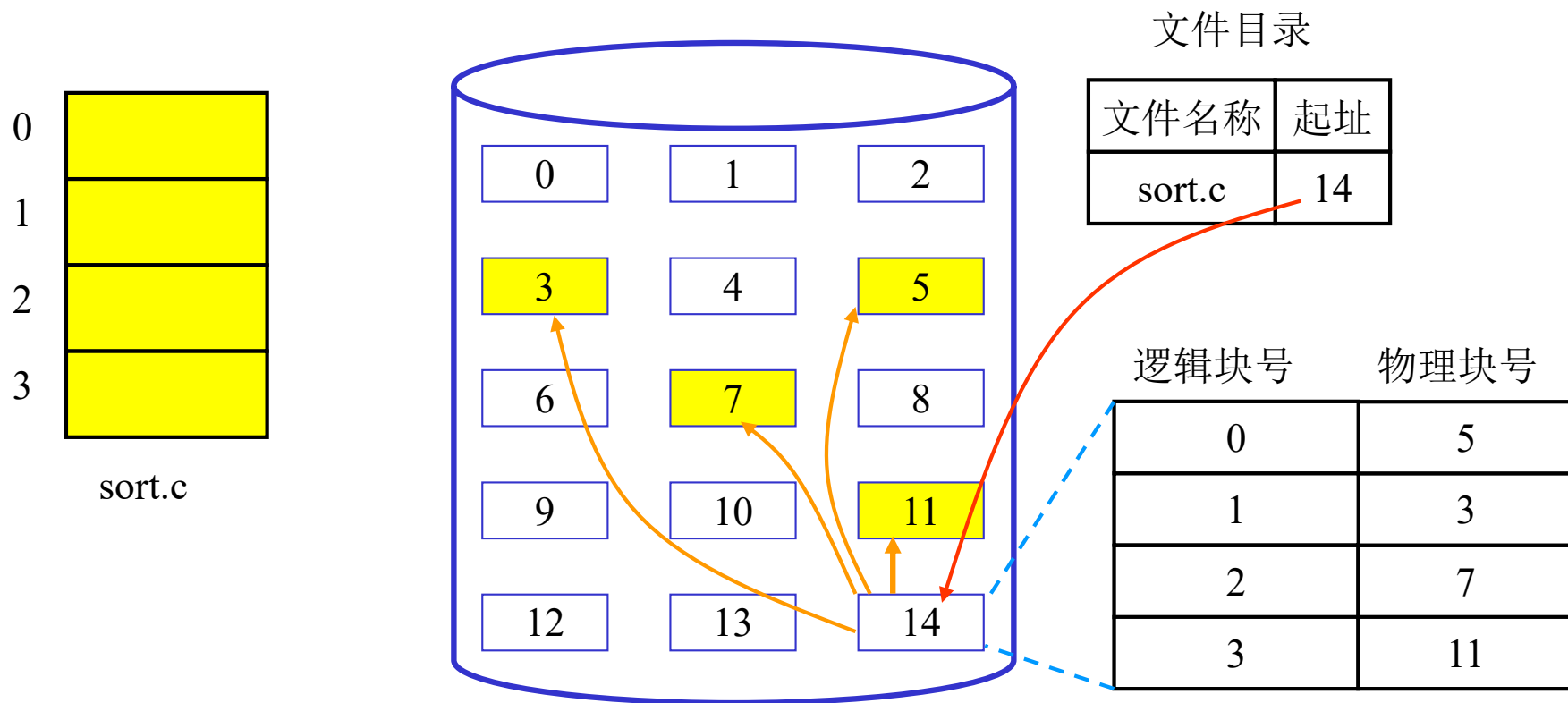




二、文件的结构

❖ 索引结构

- 将逻辑上连续的文件信息(记录)存放在不连续的物理块中，系统为每个文件建立一个专用数据结构——**索引表**，索引表中存放文件的逻辑块号和物理块号的对应关系。



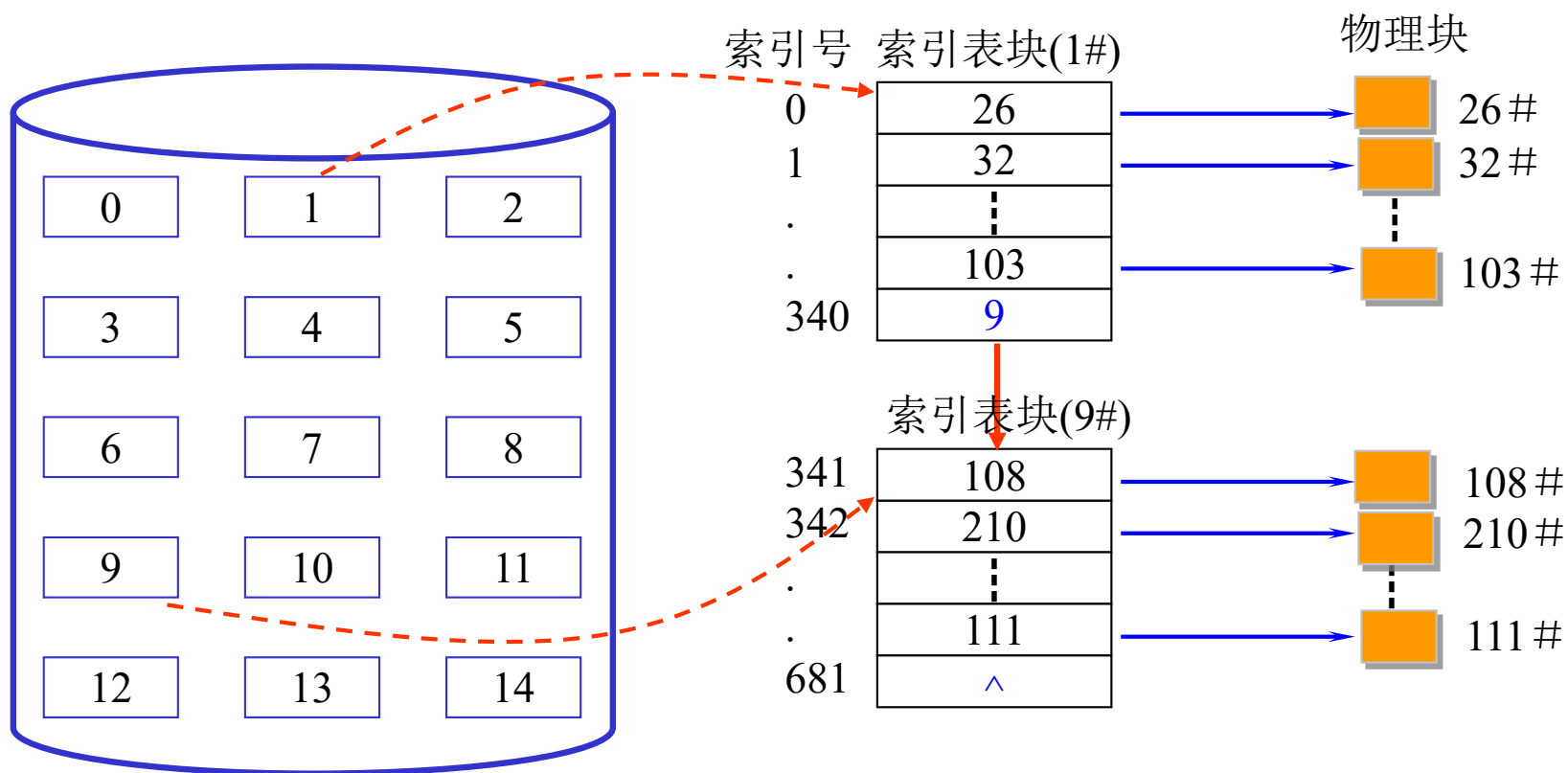


二、文件的结构



索引表的组织方式

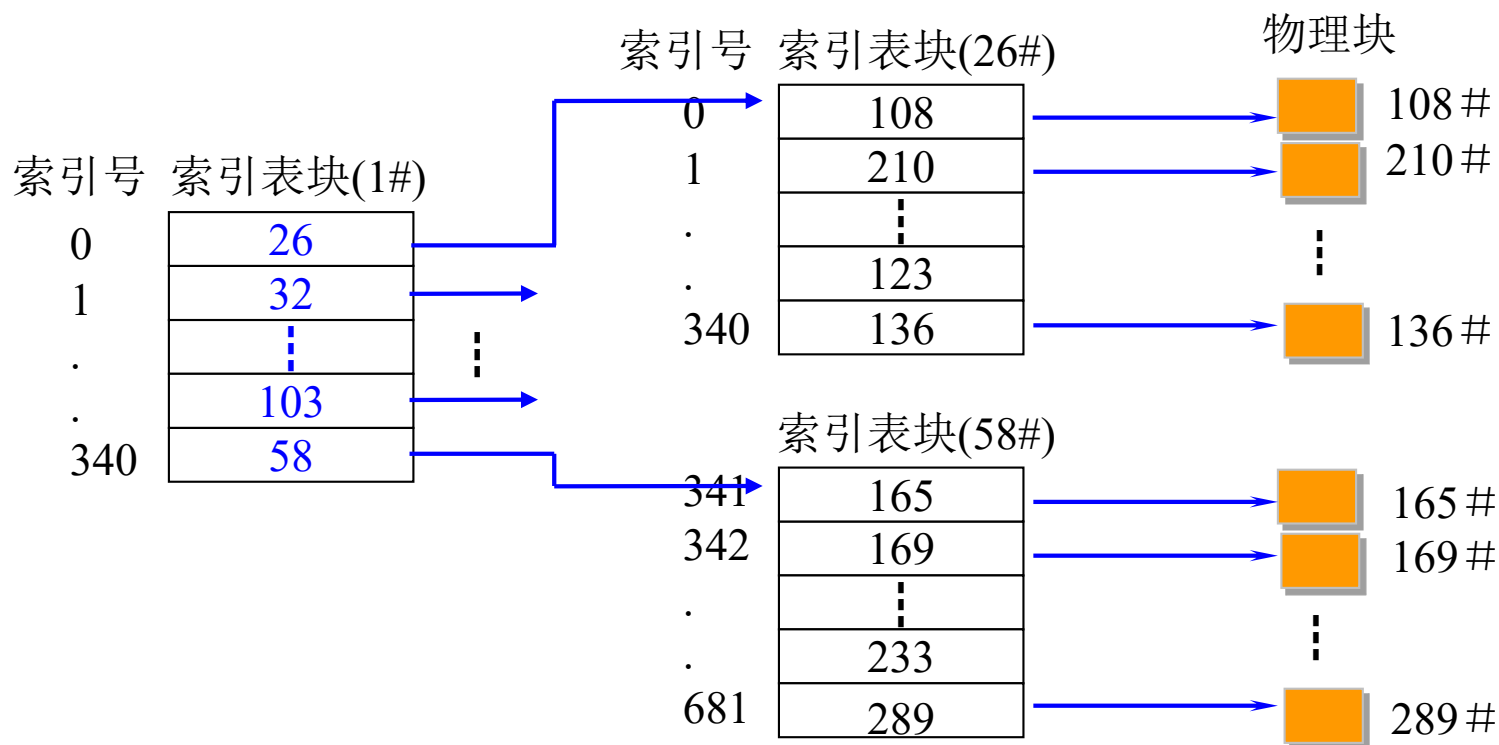
- ❖ 链接文件方式：将多个索引表块按链接文件的方式串联起来。





二、文件的结构

- ❖ 多重索引方式：将一个大文件的所有索引表的地址放在另一个索引表中。





❖ 非对称多重索引实例——UNIX文件系统

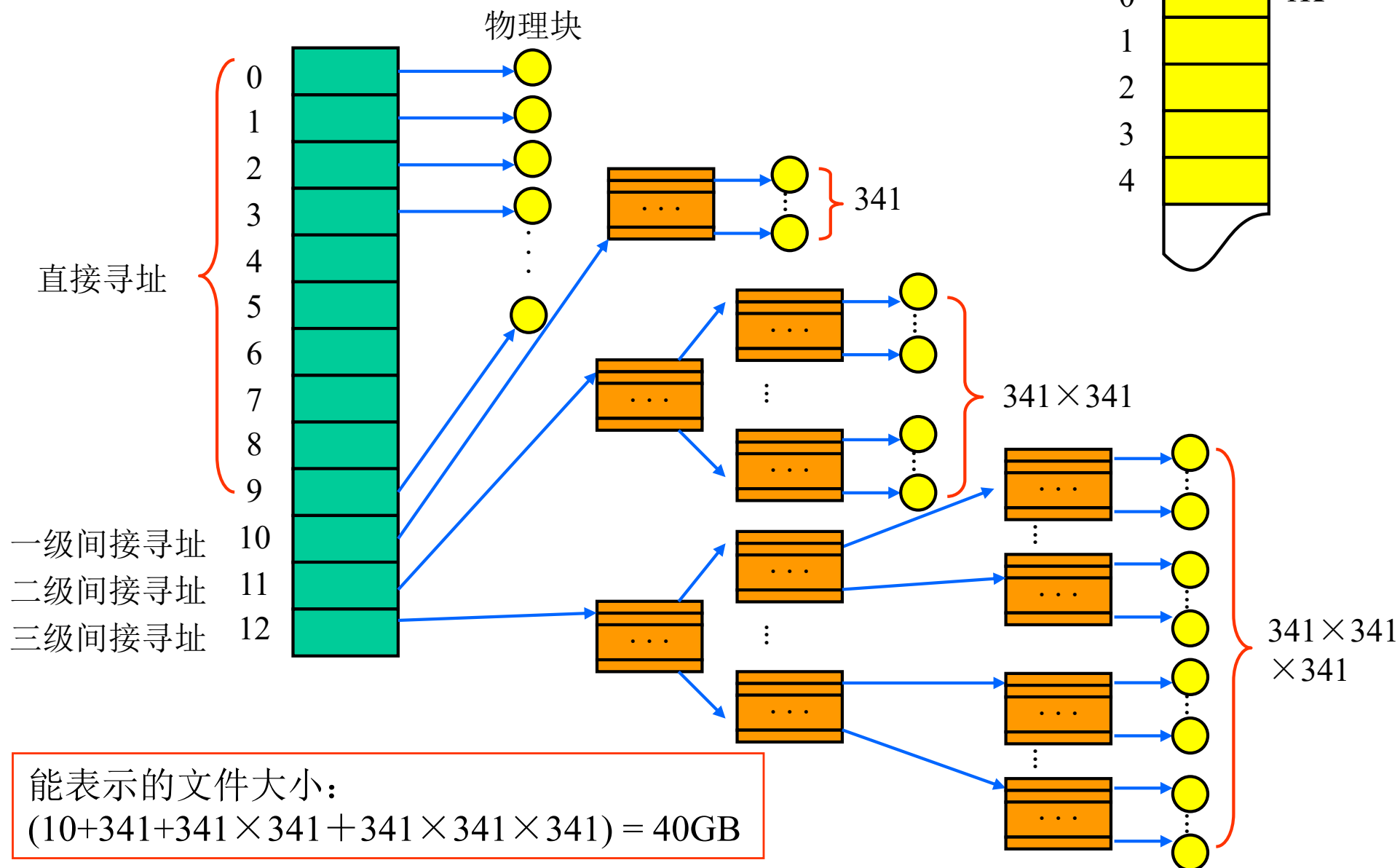
- UNIX文件系统采用三级索引结构。为了支持文件的共享和提高目录的检索速度，UNIX文件系统中设置了一类特殊的索引结点 (Index node)，称为 inode。inode 中包含了文件的控制信息，格式如下：

```
struct dinode{
    ushort    di_mode;          /*文件控制模式*/
    short     di_nlink;         /*文件的链接数*/
    ushort    di_uid; /*文件主用户标识*/
    ushort    di_gid;          /*文件主同组用户标识*/
    off_t     di_size;          /*文件长度，以字节为单位*/
    char      di_addr[40];      /*文件索引表，存放文件的物理盘块号*/
    time_t    di_atime;         /*文件最近一次访问时间*/
    time_t    di_mtime;        /*文件最近一次修改时间*/
    time_t    di_ctime;         /*文件创建时间*/
}
```



二、文件的结构

char di_addr[40]: 每3个字节组成一个单元，记录文件的物理块号

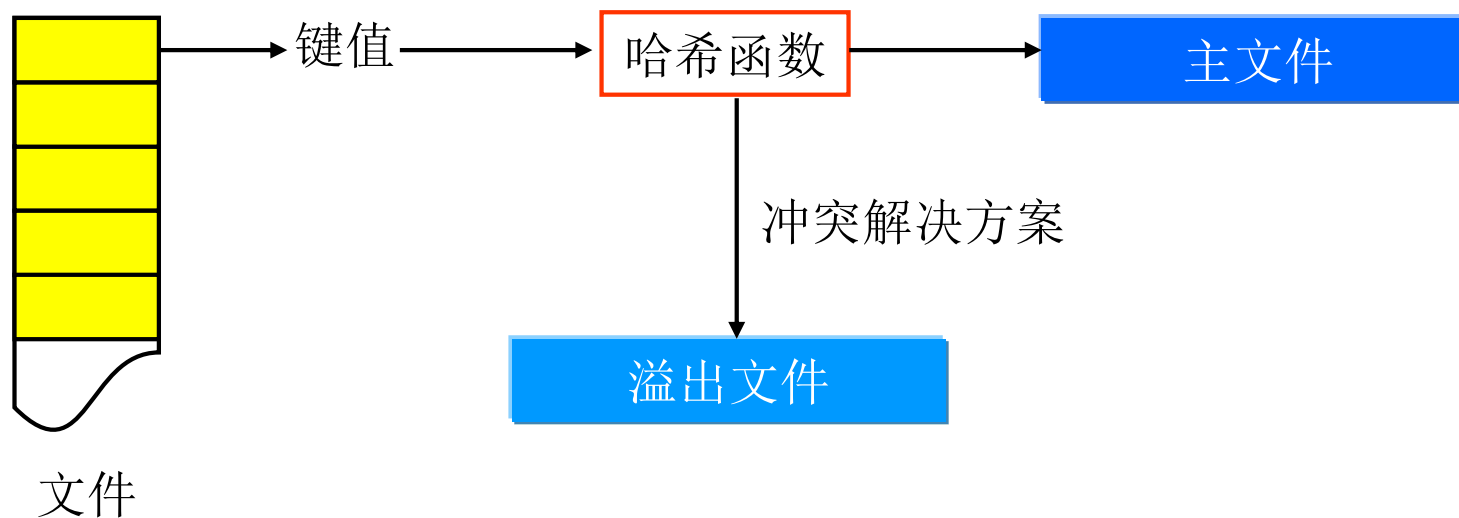




二、文件的结构

❖ Hash文件

- 采用计算寻址结构，它由主文件和溢出文件组成。





文件的存取方式

- ❖ 文件的存取方式是指读写文件存储器上的一个物理块的方法。
- ❖ 分类：
 - 顺序存取：指对文件中的信息按顺序依次读写的方式。
 - 随机读取
 - ① 直接存取法：允许用户随意存取文件中任意一个物理记录。
 - ② 按键存取法：根据文件中各记录的某个数据项内容来存取记录的，这种数据项称之为“键”。



二、文件的结构

❖ 文件结构、文件存取方式与文件存储介质的关系

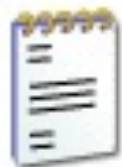
存储设备	磁带	磁盘		
文件结构	连续	连续	串联	索引
存取方式	顺序	顺序、直接	顺序	顺序、直接



三、文件的目录



文件目录的内容



sort.c

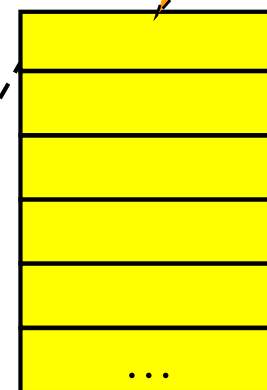
文件体：文件真实的内容。

文件说明(属性)：操作系统为了管理文件所用到的信息。

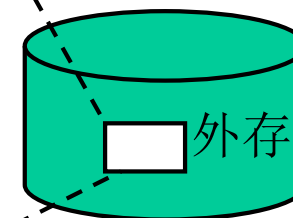
- 文件名称
- 文件标识符
- 文件类型
- 文件存储位置
- 文件大小
- 访问权限
- 时间、日期、用户标识

.....

FCB, 文件控制块



目录文件



文件目录结构的组织方式直接影响到文件的存取速度，关系到文件共享性和安全性，因此组织好文件的目录是设计文件系统的重要环节！

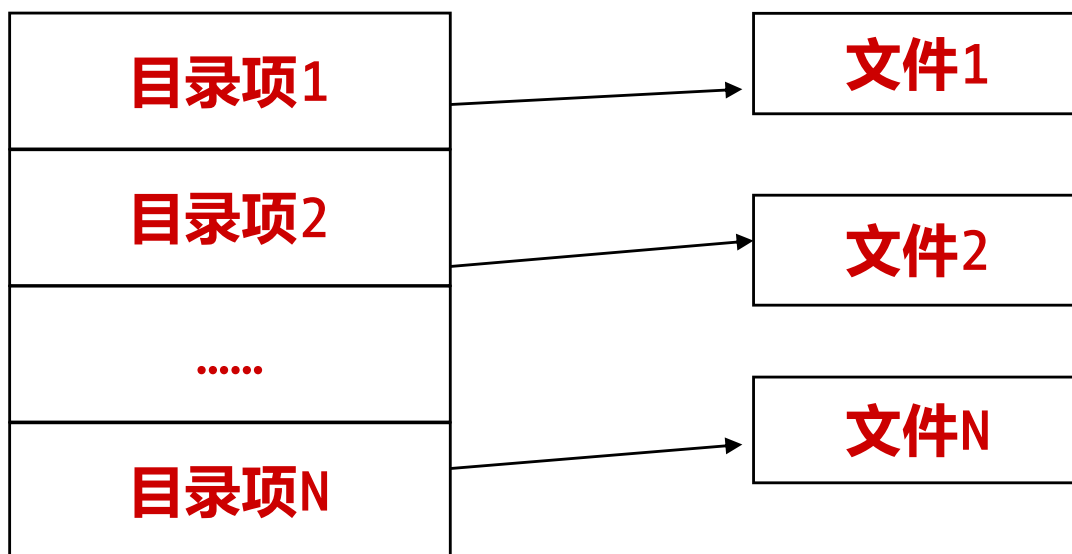


三、文件的目录



文件的目录结构

❖ 一级目录结构



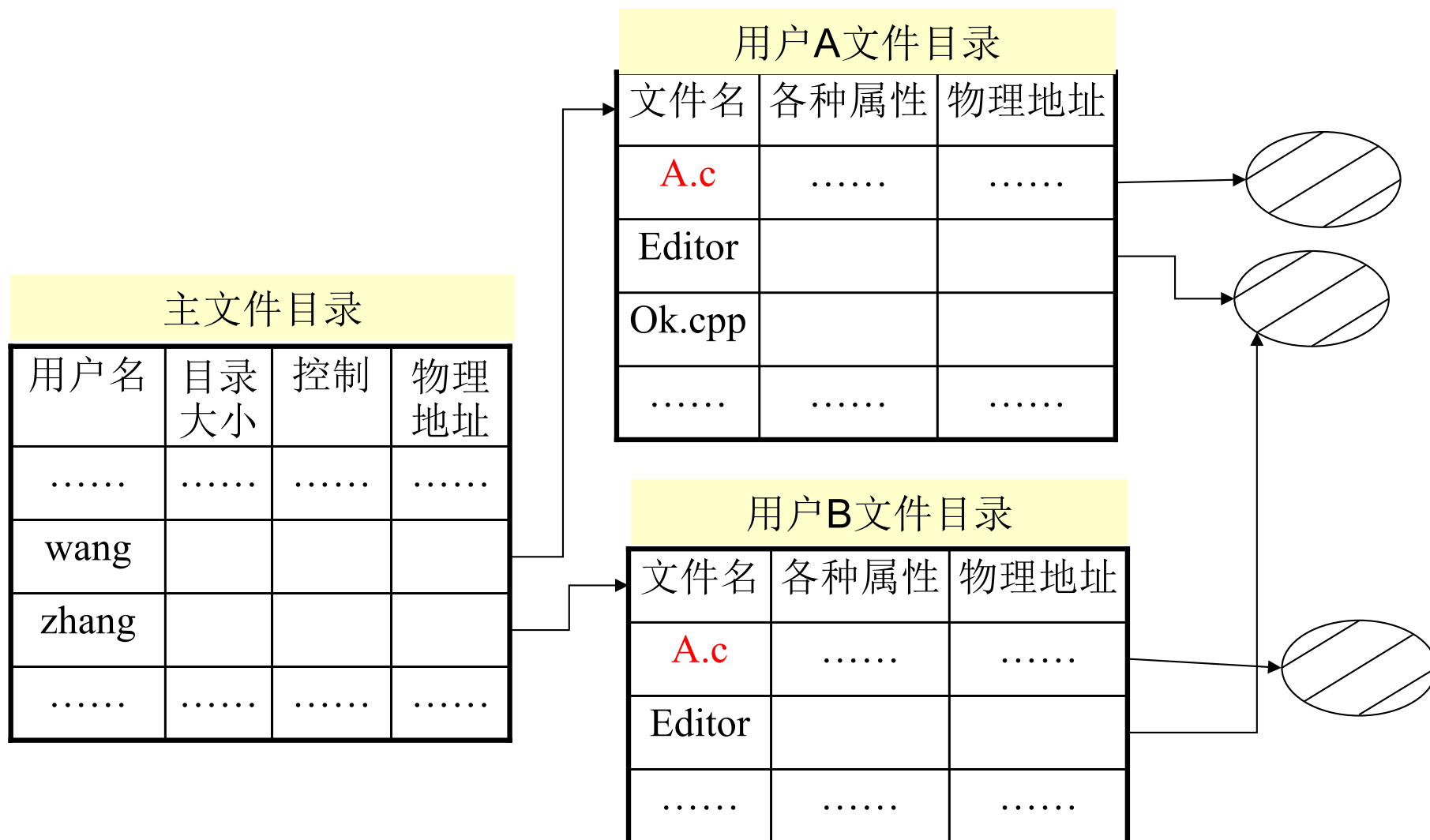
缺点:

- 重名问题
- 只支持单用户
- 文件共享问题



三、文件的目录

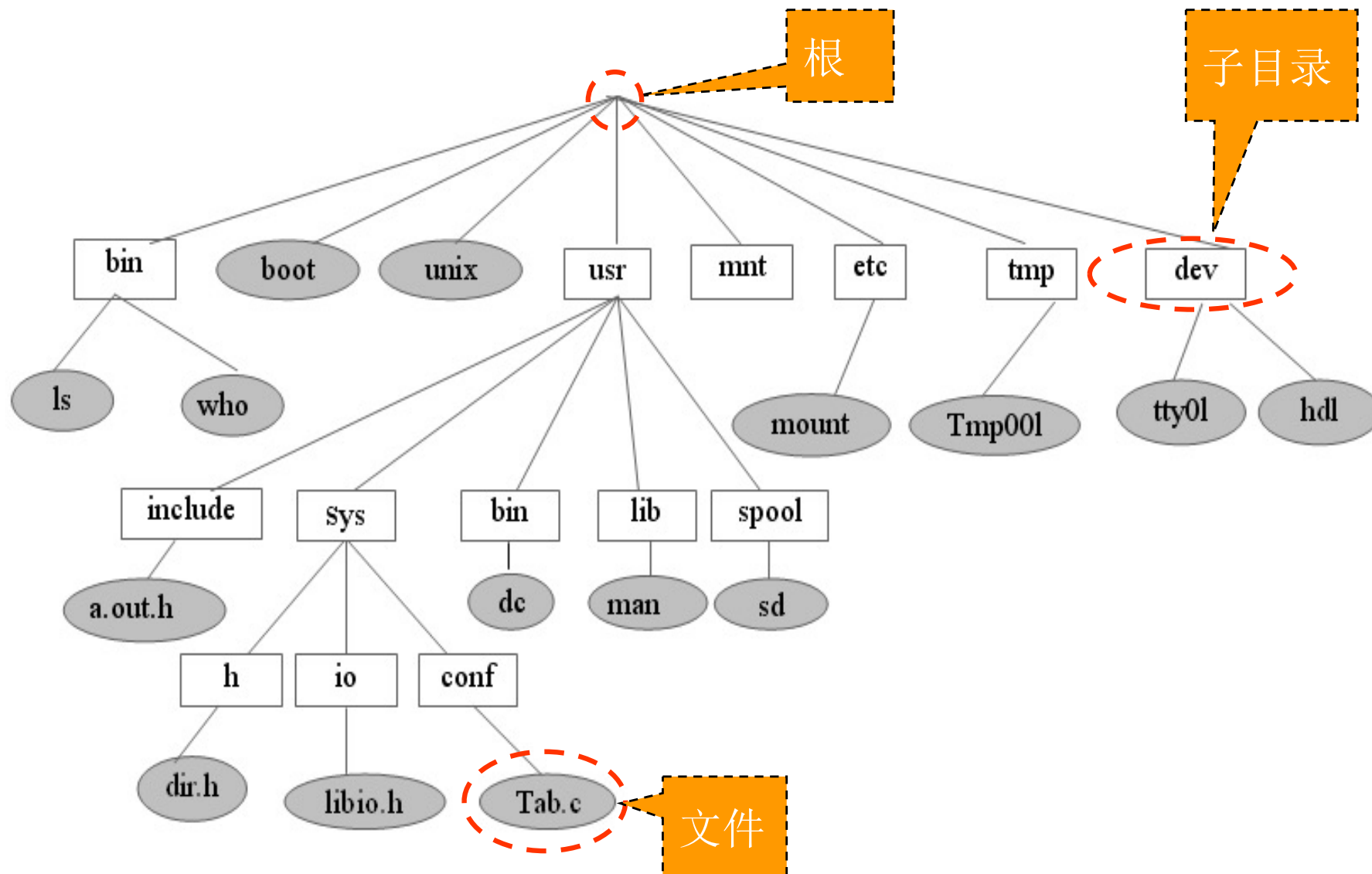
❖ 二级目录结构





三、文件的目录

❖ 多级目录结构（UNIX树形）





三、文件的目录



UNIX文件目录

目录文件

文件1	文件存放位置
文件2	文件存放位置
.....
文件n	文件存放位置

FCB

目录文件

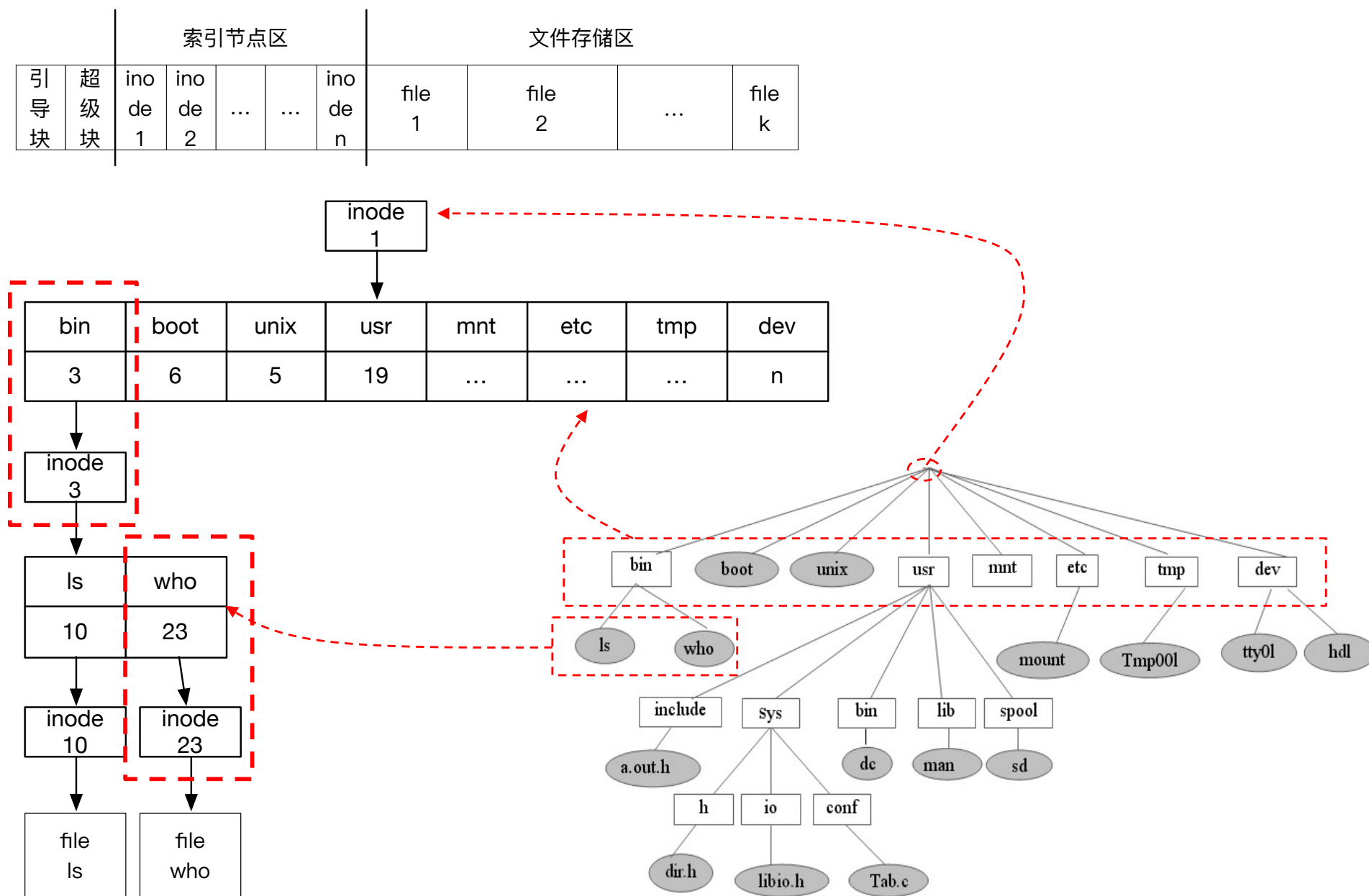
文件1	8	文件存放位置
文件2	2	文件存放位置
.....
文件n	k	文件存放位置

iNode

引导块	超级块	i 节 点 区	文 件 存 储 区
-----	-----	---------	-----------



三、文件的目录



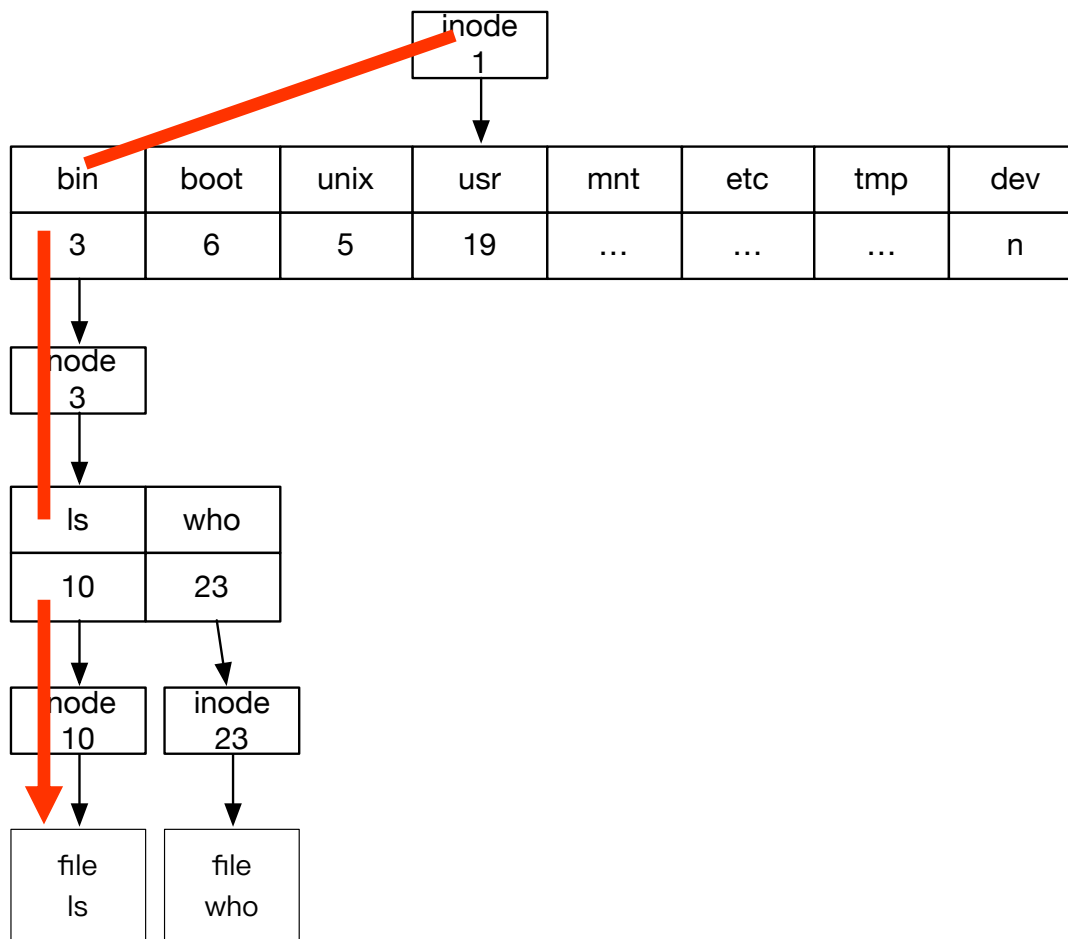


三、文件的目录

文件的查找——线性检索法

- ❖ 为实现“按名存取”，UNIX系统按照如下步骤查找所需资源

例：查找/bin/l





四、文件共享

共享的含义

- ❖ 文件共享是指不同用户或进程使用同一文件，实际上文件的实体只有一个。

文件共享类型 (UNIX)

- ❖ 硬链接(hard link)
- ❖ 软链接(symbolic Link)

❖ 可使用ln命令实现

- ```
struct inode{
 ushort di_mode;
 short di_nlink;
 ushort di_uid;
 ushort di_gid;
 off_t di_size;
 char di_addr[40];
 time_t di_atime;
 time_t di_mtime;
 time_t di_ctime;
}
```







### ❖ 硬链接执行效果

```
ln /bin/ls /usr/dir
ls -l /bin/ls
-r-xr-xr-x 2 root wheel 3554 sep 26 14:30 /bin/ls
ls -l /usr/dir
-r-xr-xr-x 2 root wheel 3554 sep 26 14:30 /use/dir
```

硬链接两大局限：

- 不能用于连接目录
- 也不能跨越文件系统的范围



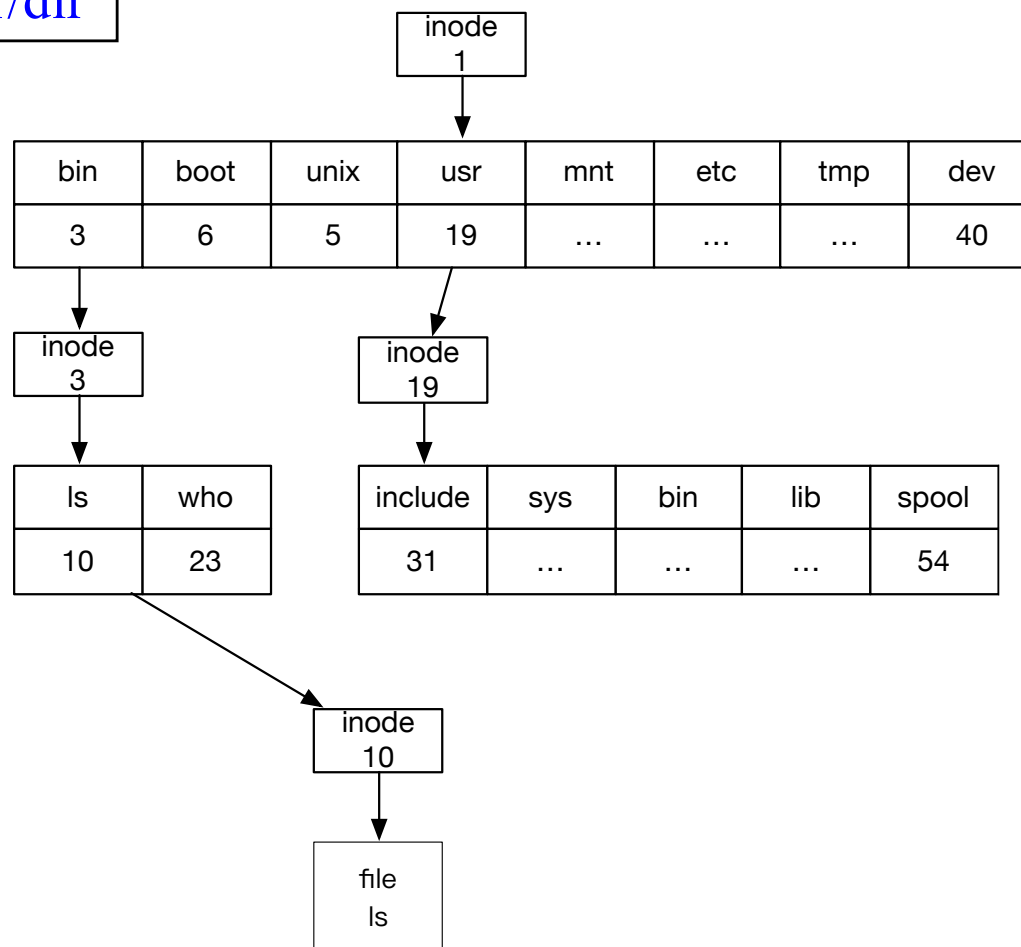
## 四、文件共享



### 软链接

- 也称**符号链接**，符号链接文件的内容为被链接文件的路径名。

例：# ln -s /bin/ls /usr/dir





### ❖ 软链接执行效果

```
ln -s /bin/ls /usr/dir
ls -l /bin/ls
-r-xr-xr-x 1 root wheel 3554 sep 26 14:30 /bin/ls
ls -l /usr/dir
lr-xr-xr-x 1 root wheel 5 oct 27 20:40 /usr/dir -> /bin/ls
```

软链接两大优势：

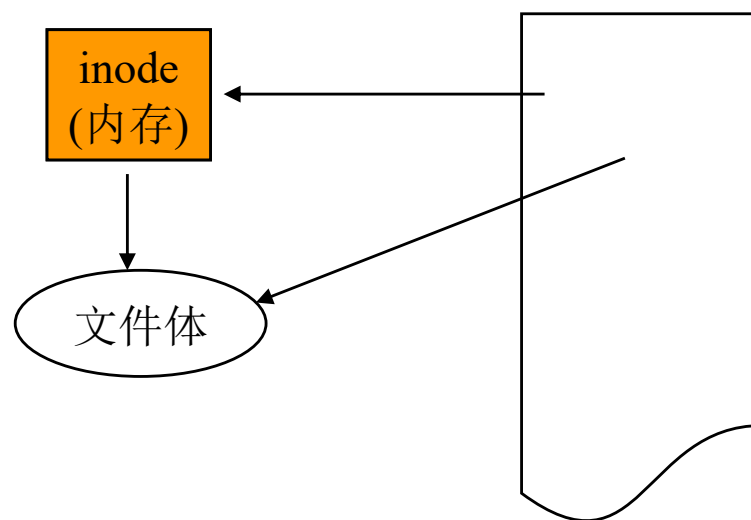
- 可用于连接目录
- 可能跨越文件系统甚至网络建立链接



### 打开文件机构

- ❖ 操作系统在内存设置了一个精炼的文件机构，用于实现灵活方便高效的文件操作，这套机构称为**打开文件机构**。
- ❖ 打开文件机构的组成：
  - 内存文件控制块（内存索引节点）
  - 系统打开文件控制块
  - 用户打开文件表

### ❖ 内存文件控制块



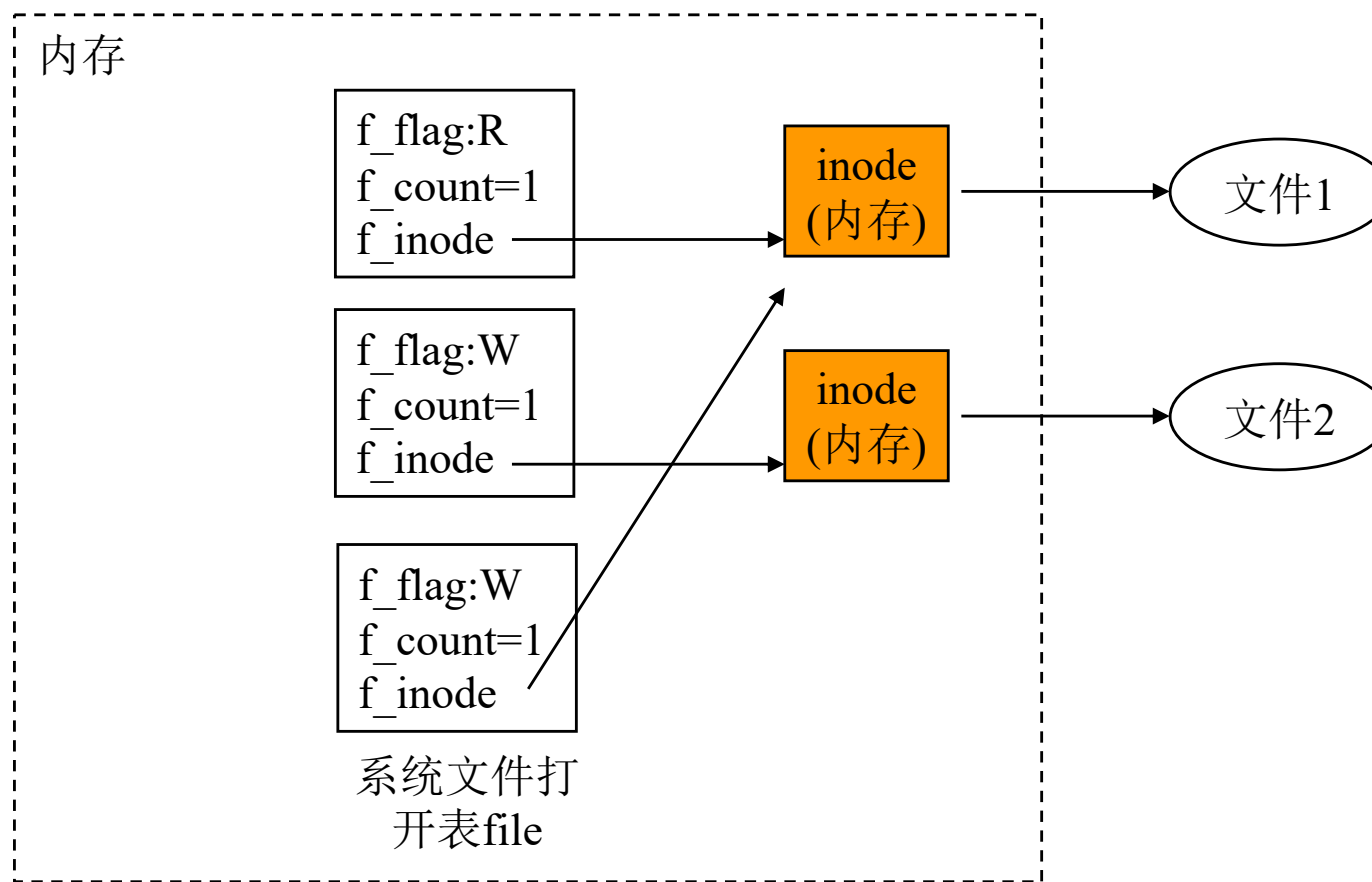
- 当打开某一个文件时，如果找不到其相应的内存inode，就在内存中将外存inode中的主要部分复制进去（适当变化），形成内存inode。
- 当需要查询、修改文件的控制信息时，直接在内存inode中进行。当关闭文件时，如果内存inode被修改过，则更新对应的外存inode信息。



## 五、文件的操作

### ❖ 系统打开文件控制块

- 系统打开文件表用于记录所有打开文件的控制信息。

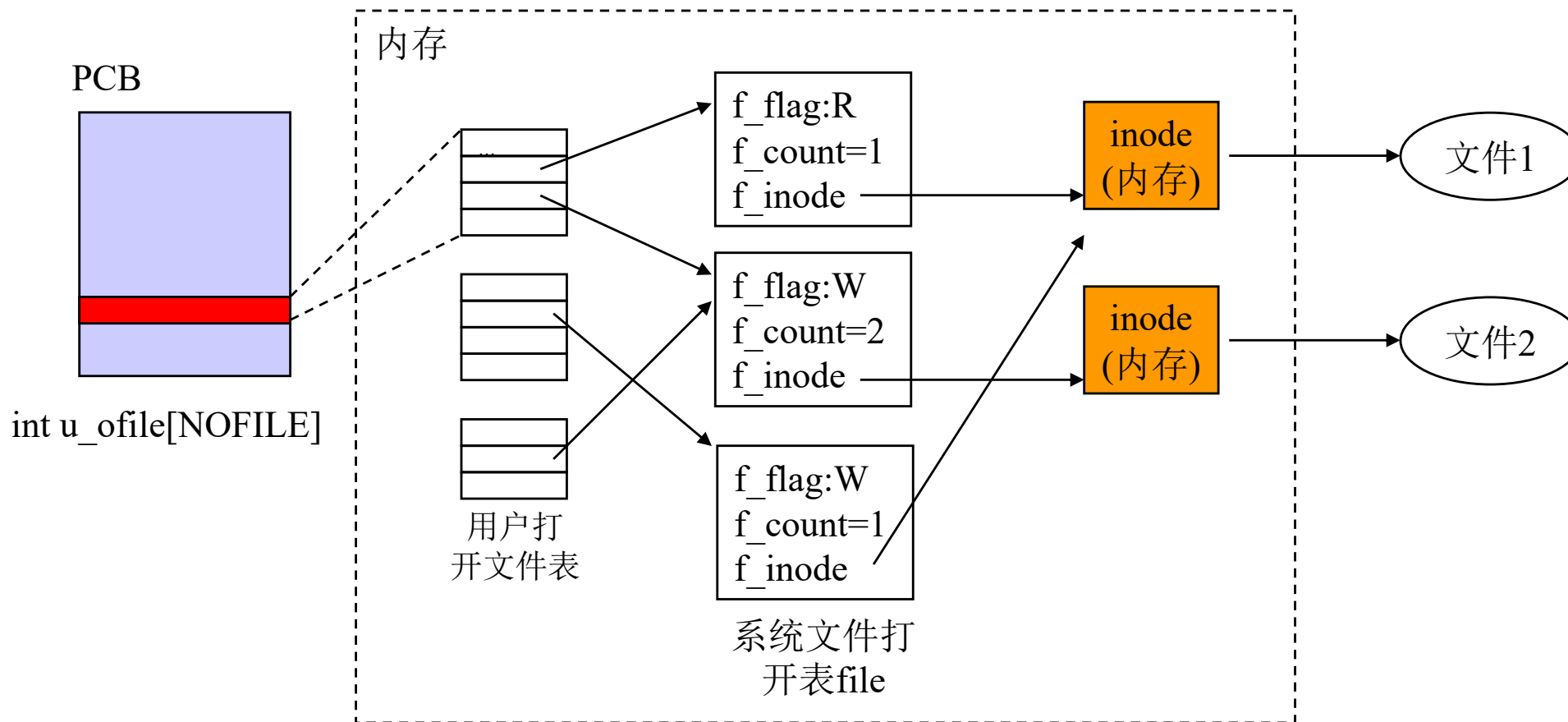




## 五、文件的操作

### ❖ 用户打开文件表

- 文件通常是通过进程来操作的，因此每一个文件都有一个用户(进程)。同样每一个进程可打开多个文件。
- 在操作系统中，每一个进程都有一张打开文件表——用户打开文件表。



## 五、文件的操作



### 举例

P1进程代码:

```
fd1=open("/etc/test", o_RDONLY);
fd2=open("pocal", o_WRONLY);
```

P2为P1的子进程, 代码如下:

```
fd3=open("/etc/testexa", o_RDONLY);
```

P3执行如下代码:

```
fd1=open("/etc/test", o_RDWR);
```



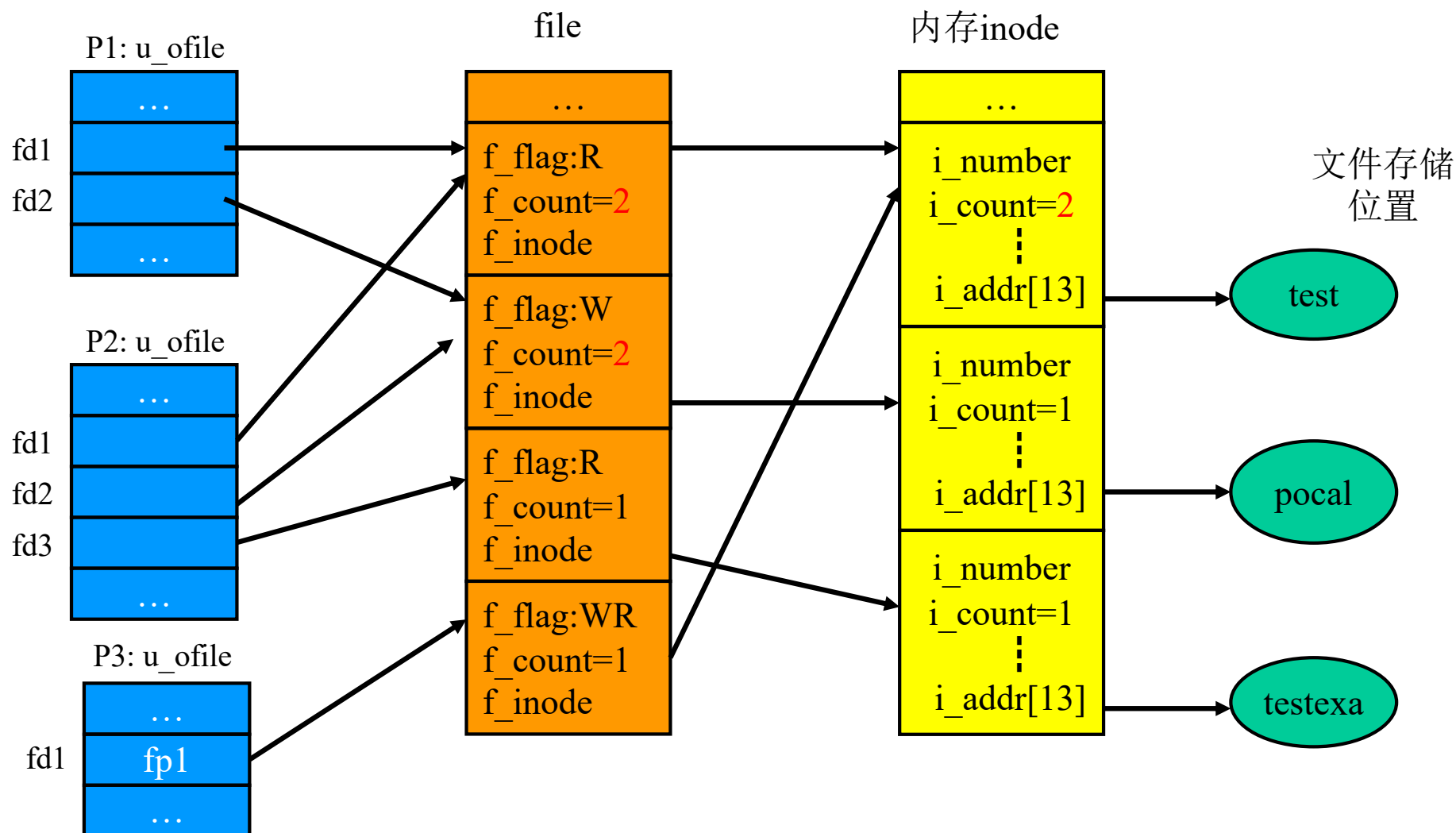


## 五、文件的操作

```
fd1=open("/etc/test", o_RDONLY);
fd2=open("pocal", o_WRONLY);
```

```
fd3=open("/etc/testexa", o_RDONLY);
```

```
fd1=open("/etc/test", o_RDWR);
```





### 概述

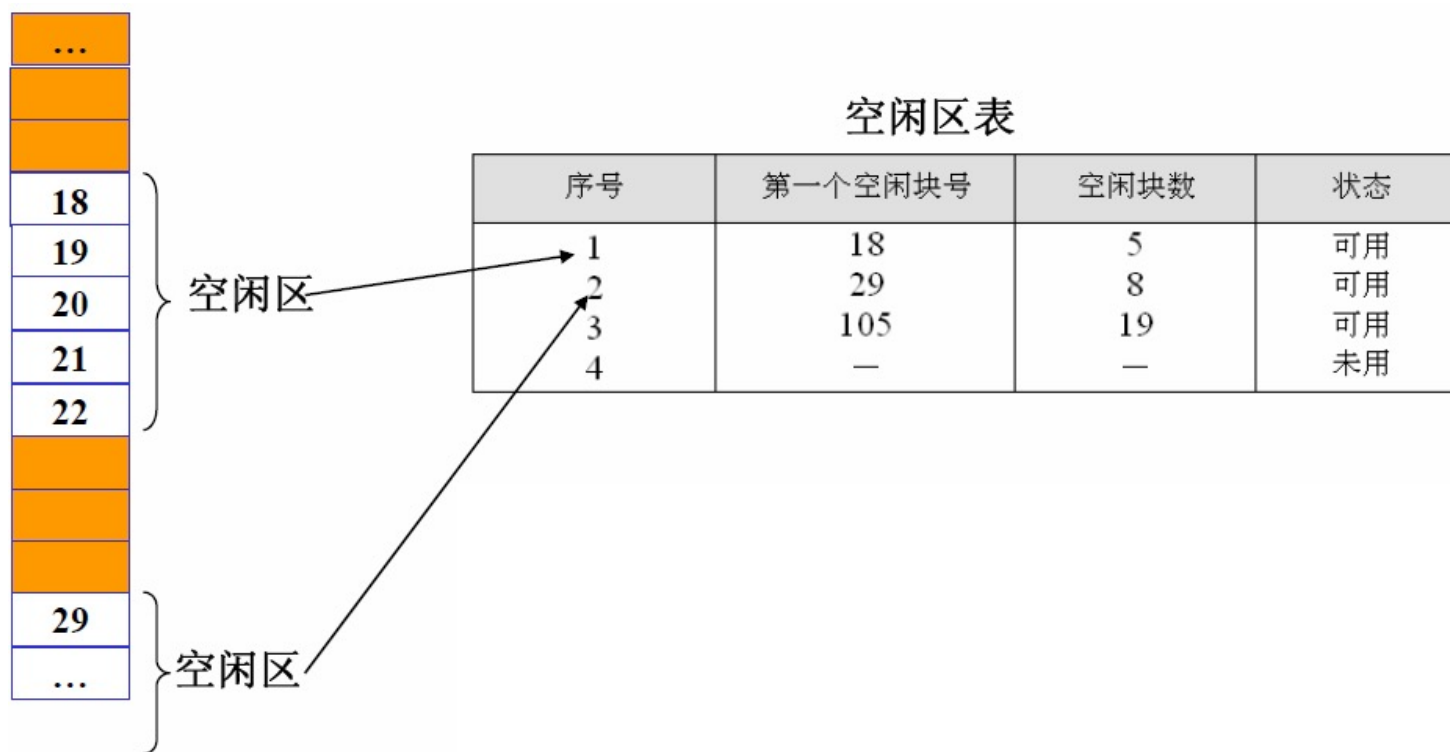
- ❖ 外存具有较大的存储空间，且被多用户共享，用户执行程序经常要在磁盘上存储、删除文件，因此，文件系统必须对磁盘空间进行有效管理。
- ❖ 外存空闲空间管理的数据结构通常称为磁盘分配表(Disk Allocation Table)。常用的空闲空间的管理方法有：
  - 空闲区表
  - 位示图
  - 空闲块链



## 六、外存空间管理

### 空闲区表

- ❖ 将外存空间上一个连续未分配区域称为“空闲区”。操作系统为磁盘外存上所有空闲区建立一张空闲表，每个表项对应一个空闲区，空闲表中包含序号、空闲区的第一个块号、空闲块的块数等信息。





### 位示图

- ❖ 这种方法是在外存上建立一张位示图(bitmap), 记录文件存储器的使用情况。每一位对应文件存储器上的一个物理块, 取值0和1分别表示空闲和占用。

```
10010011011100010101111100001010
00111001000101010111011101000011
00110111011101010101101101011110
11011111000100101100010110101111
00011111101010101101011101011011
10000011111101110101000011010010
```

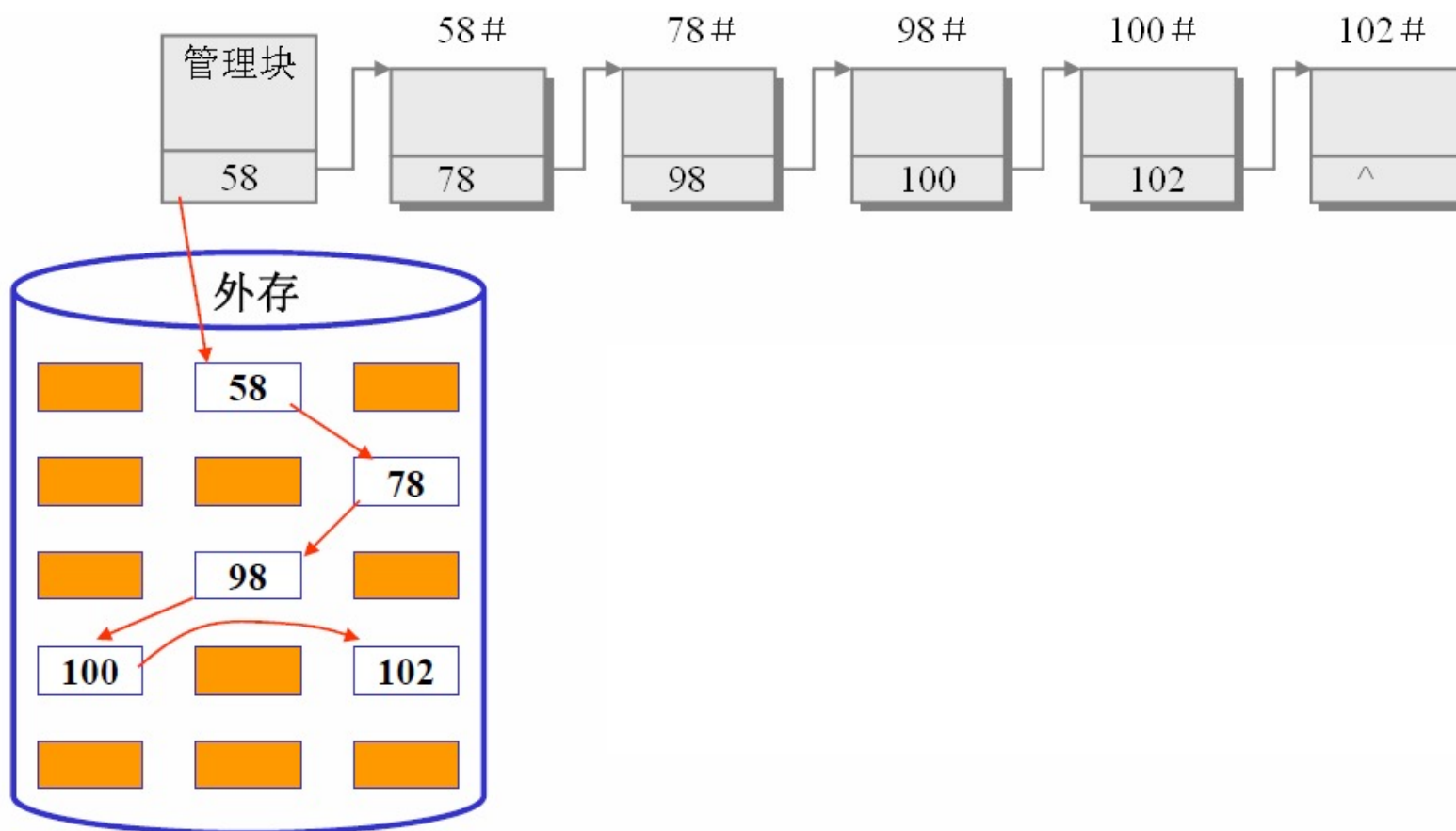


## 六、外存空间管理



### 空闲块链

- ❖ 每个空闲物理块中有指向下一个空闲物理块的指针，所有空闲物理块构成一个链表，链表的头指针放在文件存储器的特定位置上（如管理块中）。

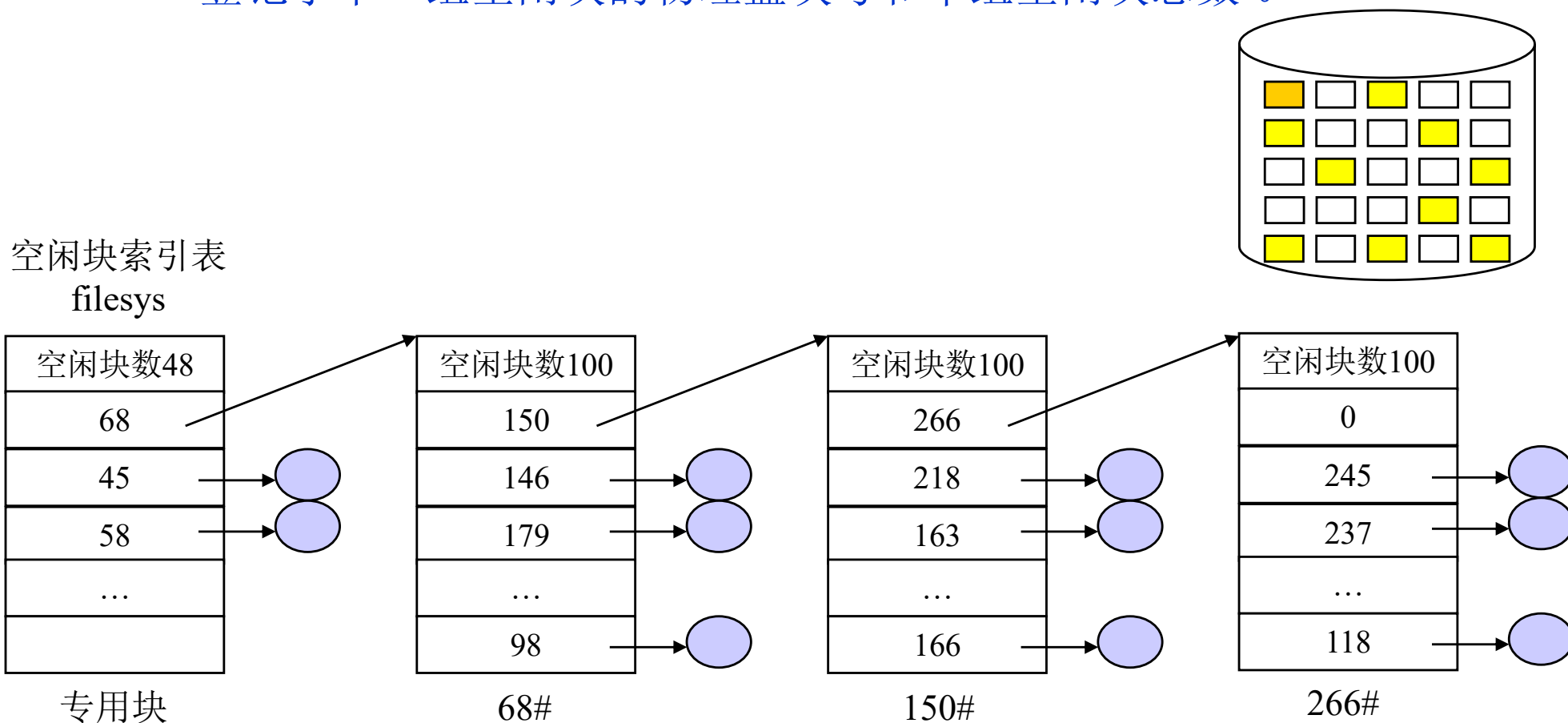




## 六、外存空间管理

### 成组链接法

- ❖ 将空闲块分成若干组，每100个空闲块为一组。每组的第一个空闲块登记了下一组空闲块的物理盘块号和本组空闲块总数。

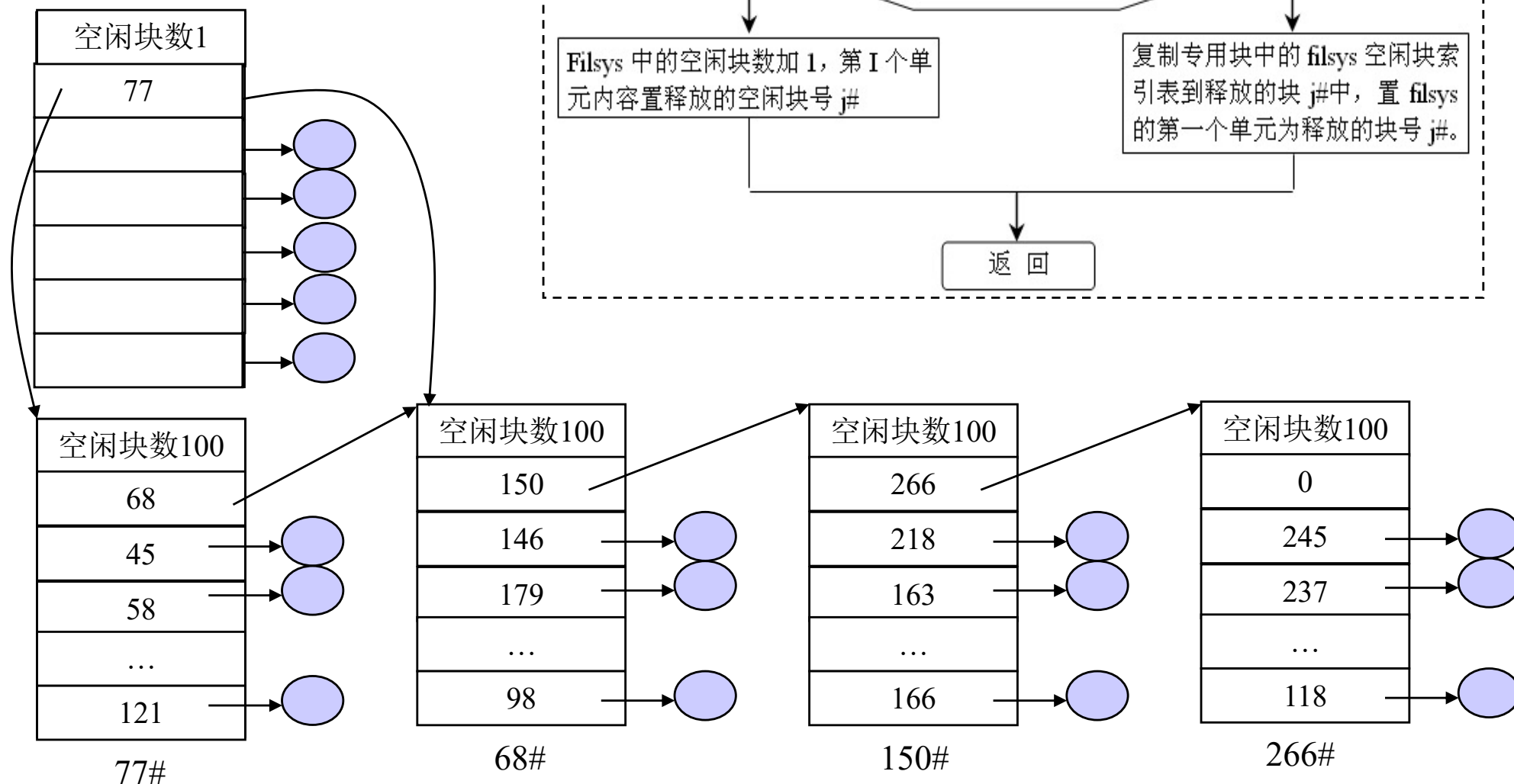




## 六、外存空间管理

### ❖ (进程)释放算法

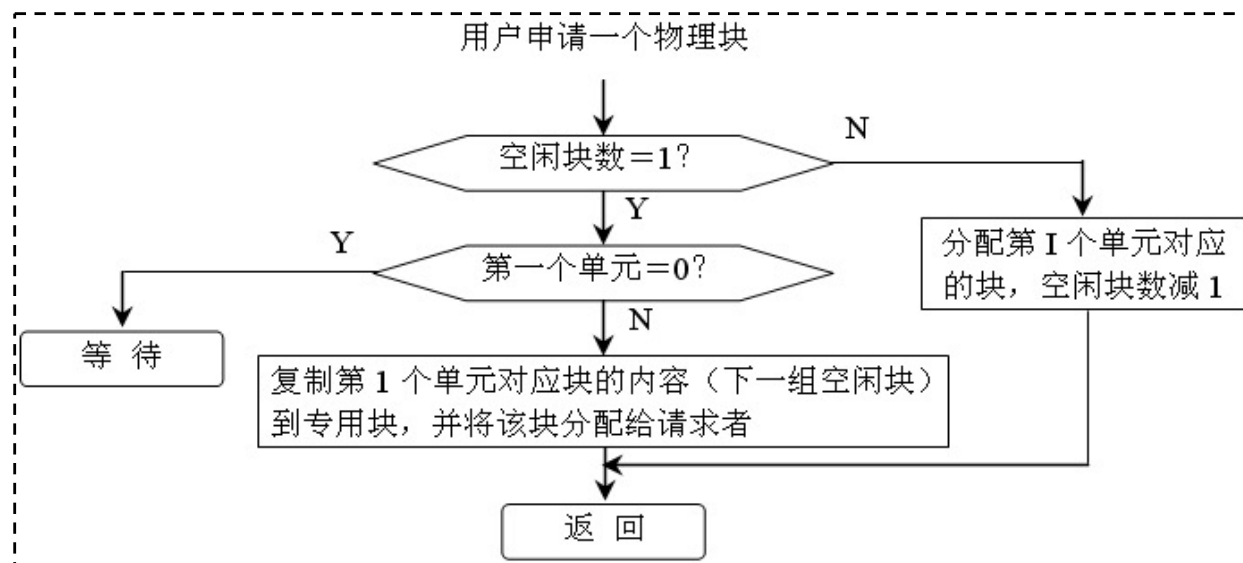
空闲块索引表  
filesys



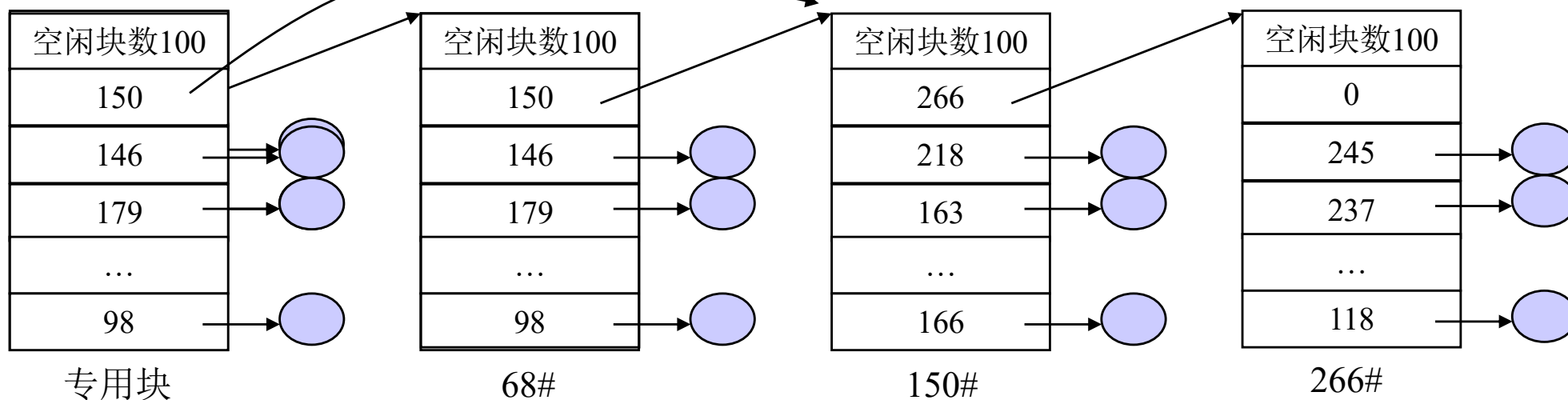


# 六、外存空间管理

## ❖ 分配算法



空闲块索引表  
filesys





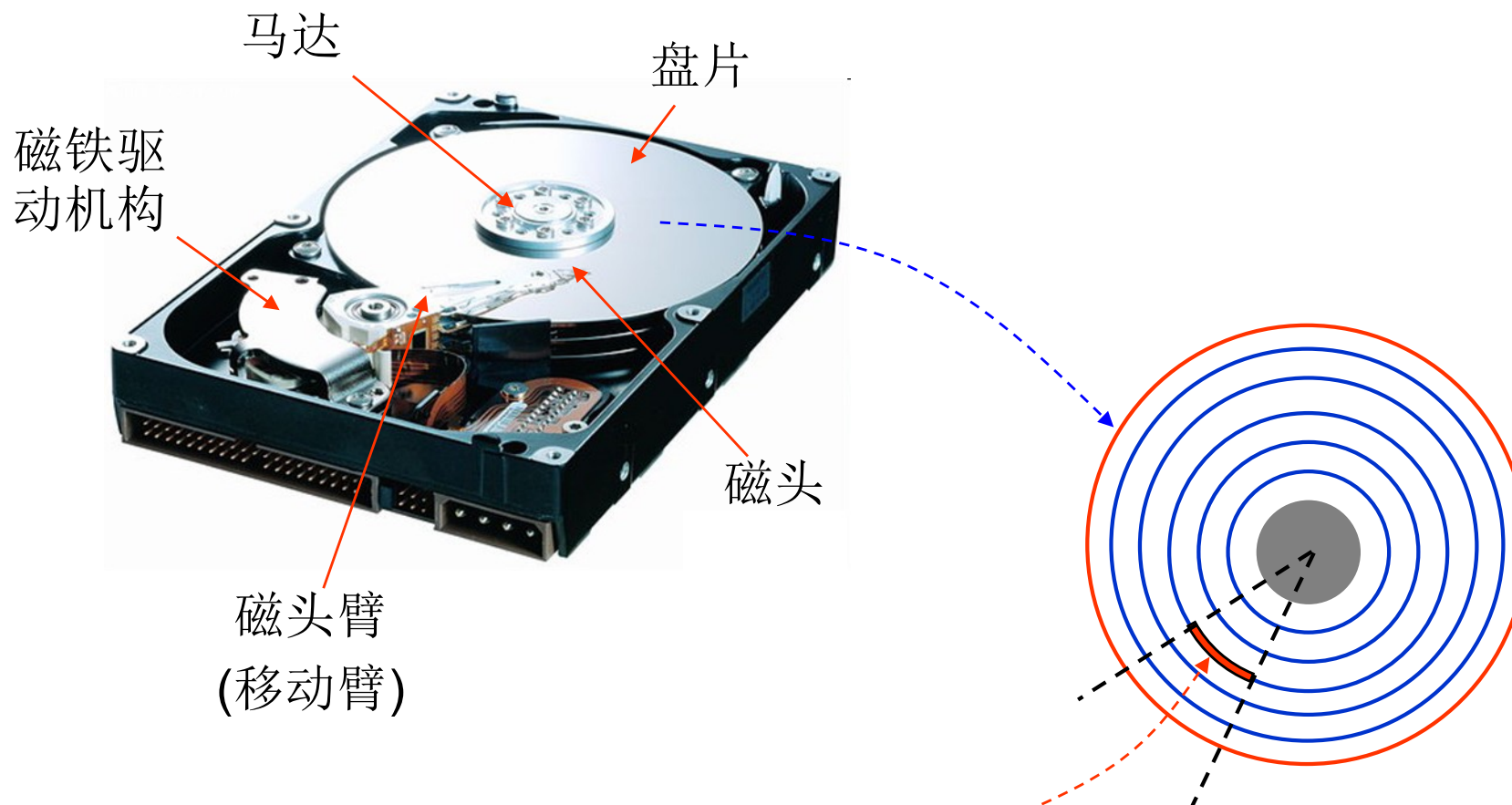
## 七、文件系统的性能分析



### 影响文件系统性能的因素？

- ❖ 存储介质
- ❖ 磁盘性能的好坏
- ❖ 磁盘调度算法的好坏
- ❖ 磁盘高速缓冲区的大小

# 磁盘的结构



扇区：一般为600字节，其中512字节存放数据，其余存放控制信息。



# 影响磁盘读写速度的因素

✚ 寻道时间(平均)

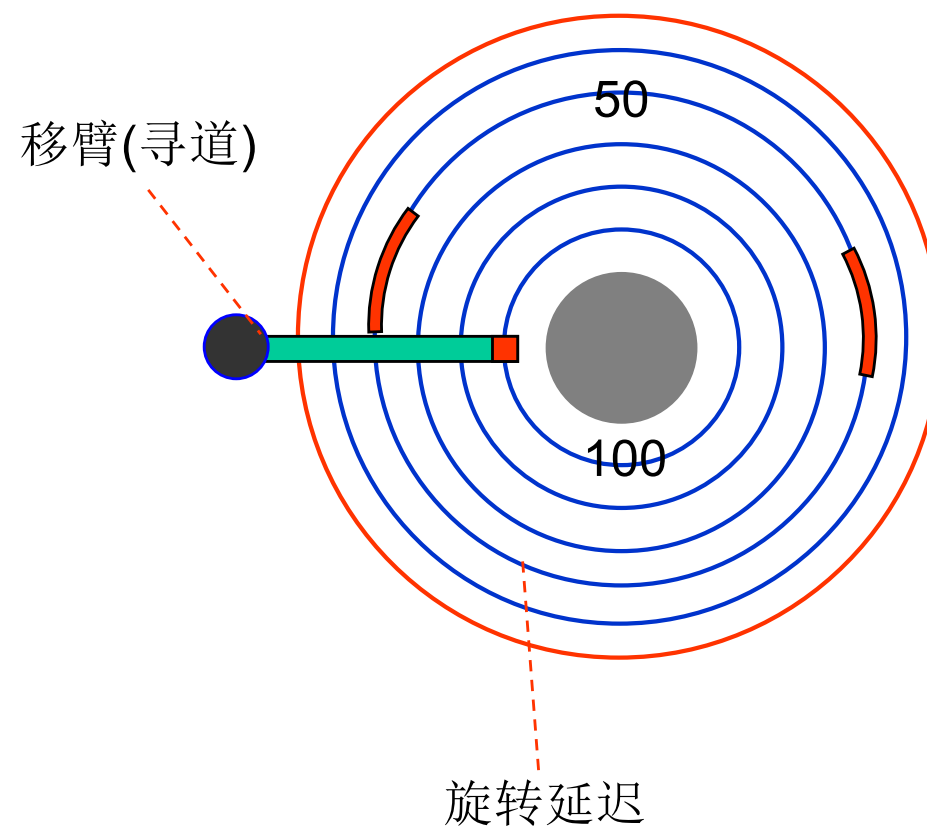
❖ 8~20ms

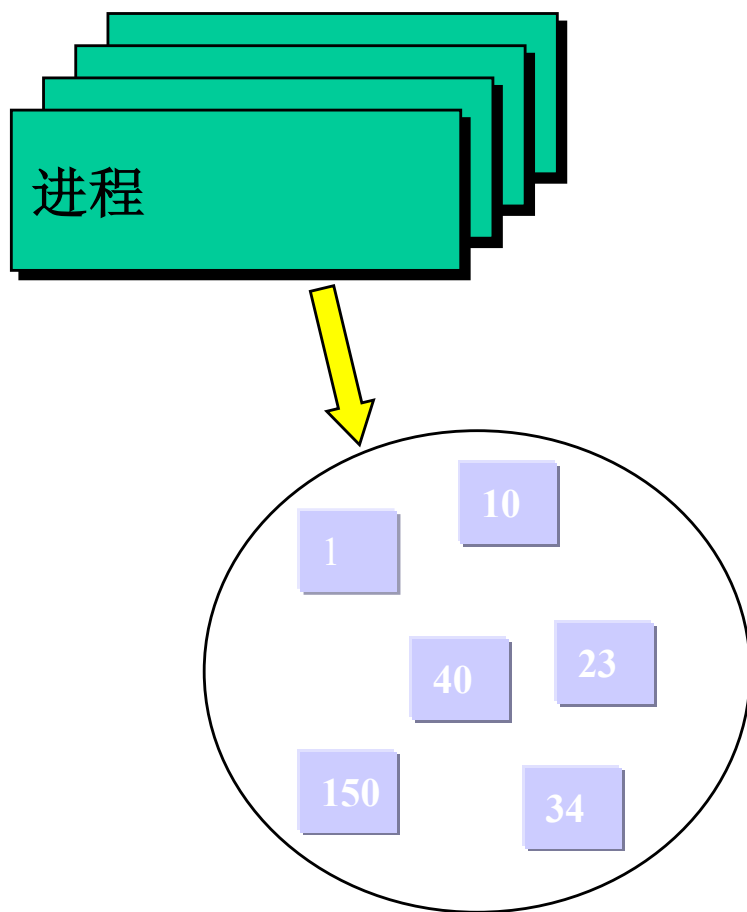
✚ 旋转延迟时间(平均)

❖ 3~8.57ms

✚ 数据传输时间

❖ 0.13ms





给出一种访问顺序，使得访问这些磁道总的花费时间尽可能的少？

磁盘调度算法：

- 先来先服务
- 最短寻道时间优先算法
- 电梯调度算法

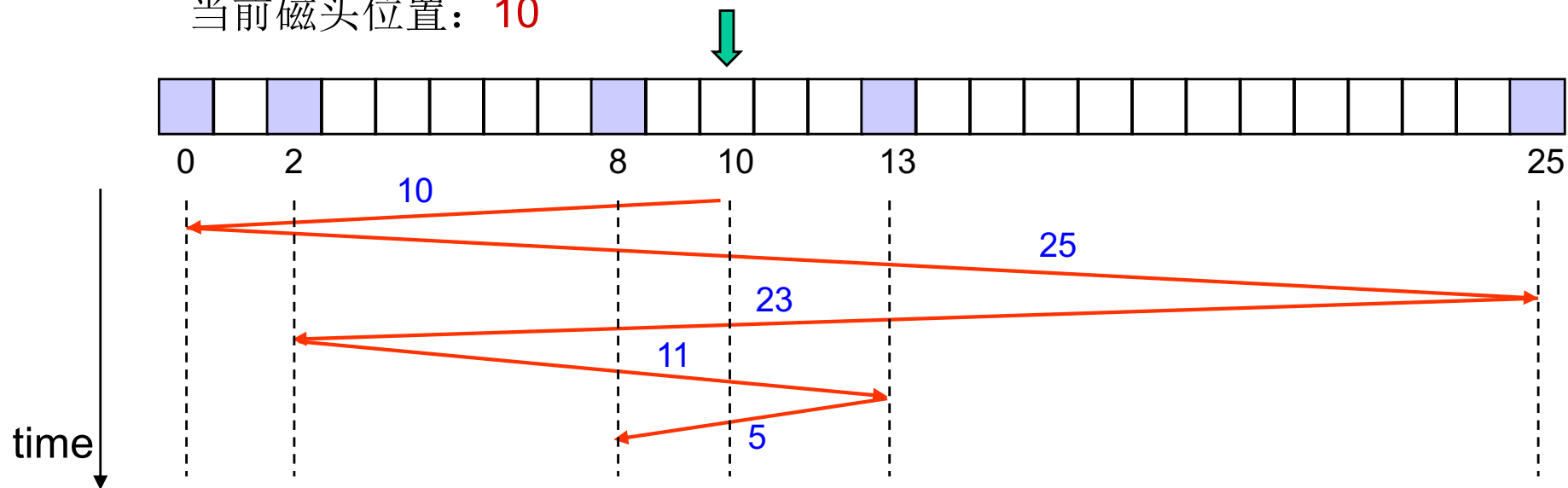


## 先来先服务算法(FCFS, First Come First Served)

- ❖ 思想：严格按照进程请求访问磁盘的先后次序进行调度，是一种最简单的磁盘调度算法。

访问请求的到达顺序：0、25、2、13、8

当前磁头位置：10



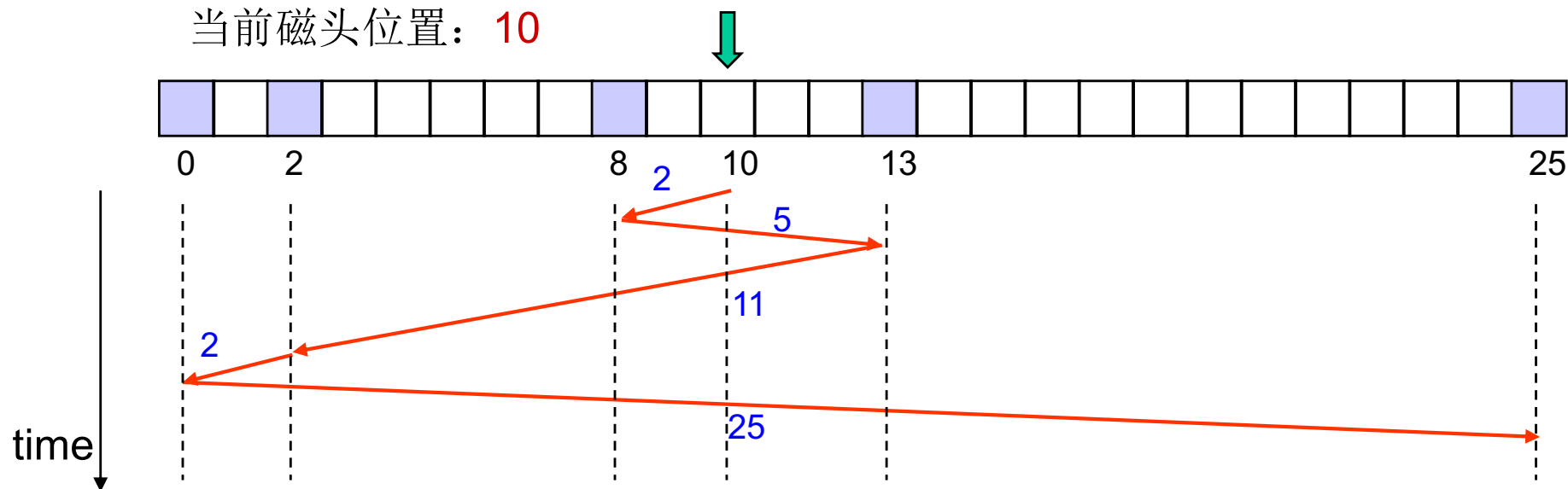
总寻道时间=10+25+23+11+5=74

## 最短寻道时间优先算法(SSF, Shortest Seek First)

❖ 思想：要求每次访问的磁道与当前磁头所在的磁道距离最近。

访问请求的到达顺序：0、25、2、13、8

当前磁头位置：10



总寻道时间=2+5+11+2+25=45

“饥饿”现象！



# 移臂调度算法

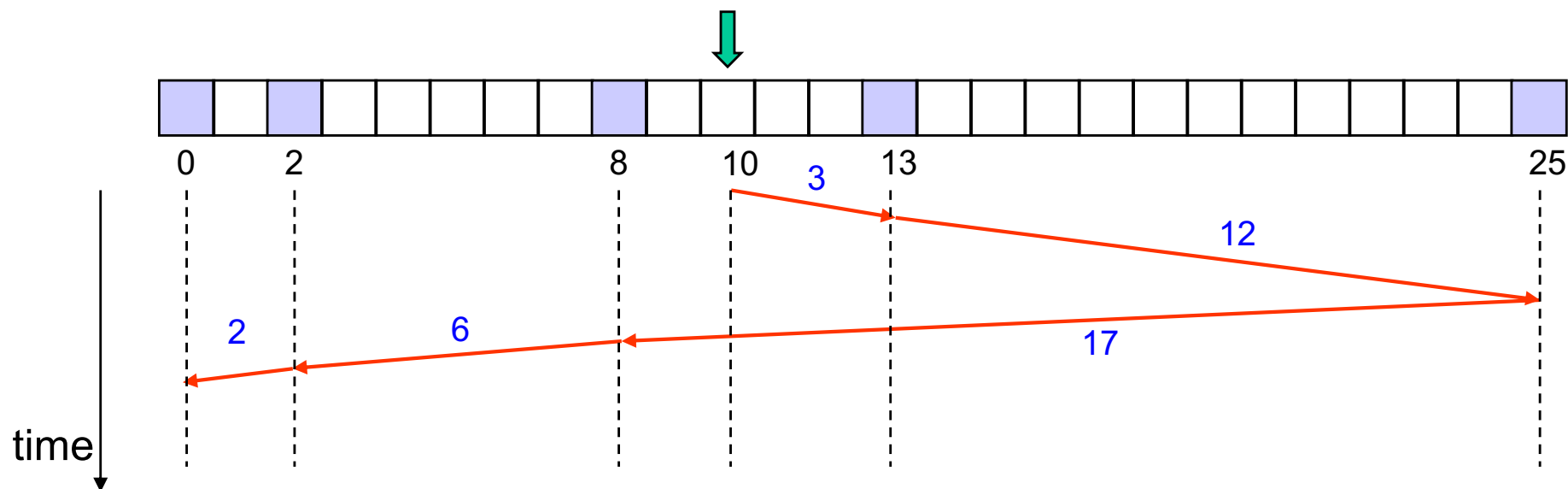


## 电梯调度算法(SCAN)

- ❖ 思想：不仅考虑到预访问的磁道与当前磁道的距离，更优先考虑磁头的当前移动方向。

访问请求的到达顺序：0、25、2、13、8

当前磁头位置：10，方向向右



总寻道时间=3+12+17+6+2=40



|           | FCFS | SSF | SCAN |
|-----------|------|-----|------|
| 平均花费时间    | 较大   | 较小  | 较小   |
| 稳定性       | 差    | 优   | 优    |
| 是否有“饥饿”现象 | 无    | 有   | 无    |

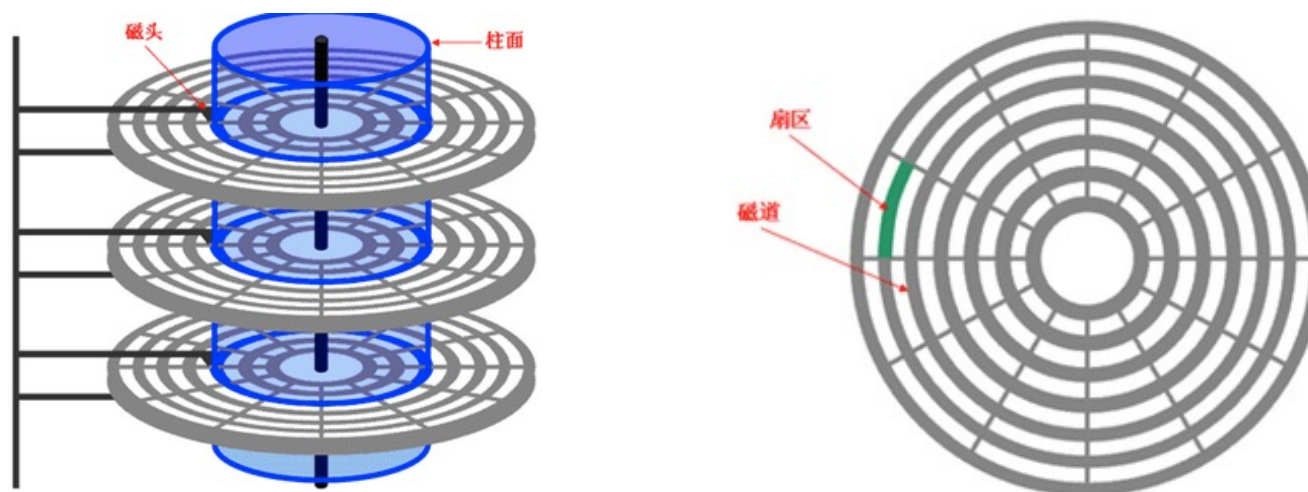




## 七、文件系统的性能问题

### 旋转调度算法

- ❖ 研究当移动臂定位后，如何访问数据的问题？可能出现的情况如下：
  - 进程请求访问的是同一磁道上的不同编号的扇区；
  - 进程请求访问的是不同磁道上具有相同编号的扇区。
  - 进程请求访问的是不同磁道上的不同编号的扇区；



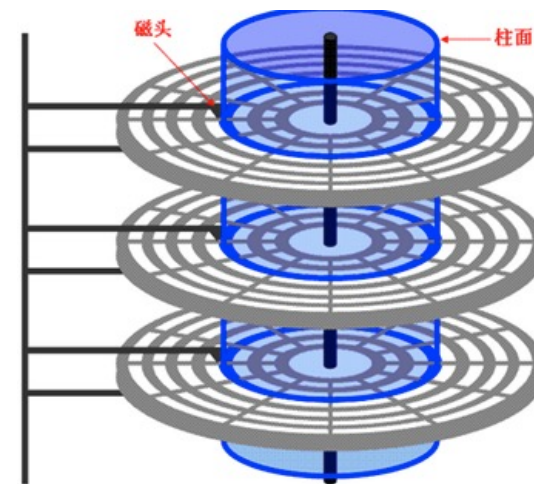
目标：在同一柱面读写信息时，花费最少的时间。



## 七、文件系统的性能问题

❖ 举例（假设当读写头总是从编号最小的扇区处开始读数据）

| 请求顺序 | 柱面号 | 磁道号 | 扇区号 |
|------|-----|-----|-----|
| ①    | 16  | 5   | 3   |
| ②    | 16  | 1   | 6   |
| ③    | 16  | 5   | 1   |
| ④    | 16  | 9   | 6   |
| ⑤    | 16  | 1   | 4   |
| ⑥    | 16  | 8   | 12  |



可能的访问顺序：③①⑤②⑥④，③①⑤④⑥②

结论：当一次移臂调度将移动臂定位到某一柱面后，还可能要进行多次旋转调度才能得到所有数据。



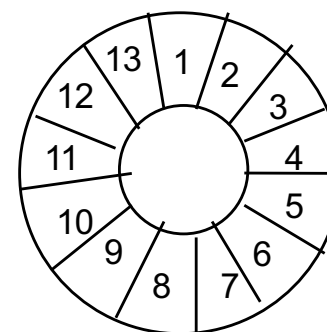
## 七、文件系统的性能问题



### 信息的分布问题

- ❖ 磁盘调度算法可以尽量减少访问时间，但是受到文件存储位置的限制，其性能不一定能达到最优。要想进一步提高读写效率，还需要合理的安排数据的存储位置。

例子：假设有13个记录存放在磁盘的某一磁道上，磁道被划分成13块，顺次存放这13个记录。如果磁盘旋转速度为30ms（毫秒）转1周，处理程序每读一个记录后花5ms进行处理。请问访问这13个记录总共需要花费多少时间？



$$\text{访问时间} = 13 * 30 + 5 = 395\text{ms}$$



## 七、文件系统的性能问题

### ❖ 信息的优化分布

为缩短处理时间应如何排列这些记录？重新排列记录后的总的处理时间是多少？

