Operating System Principle, OS

2025年秋. 操作系统原理

第9章 文件系统与实现

课 程 组：邹德清, 李珍, 李志, 苏曙光

企业教师：华为认证专家（鸿蒙方向）

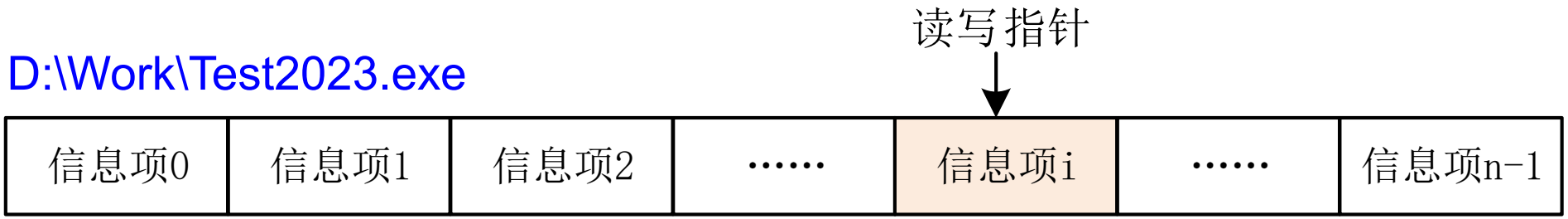
文件系统

- 内容
  - 文件和文件系统概念
  - 文件结构
  - 存储空间管理
  - 索引节点（inode）
  - 虚拟文件系统VFS
- 重点
  - 文件逻辑结构
  - 文件物理结构
  - 索引节点（inode）

9.1 文件和文件系统概念

文件和文件系统概念

- 文件
  - 文件是计算机系统存放信息的一种形式，由若干信息项有序构成。
  - 文件具有唯一的文件名。
  - 用户通过读写指针来存取文件的信息项。



文件和文件系统概念

- 文件分类
  - 按文件的用途
    - 系统文件
    - 库文件
    - 用户文件
  - 按文件的操作权限
    - 只读文件
    - 只写文件
    - 可执行文件
    - 可读可写文件
    - 不保护文件
  - 按文件的存储时间
    - 永久文件
    - 临时文件

文件和文件系统概念

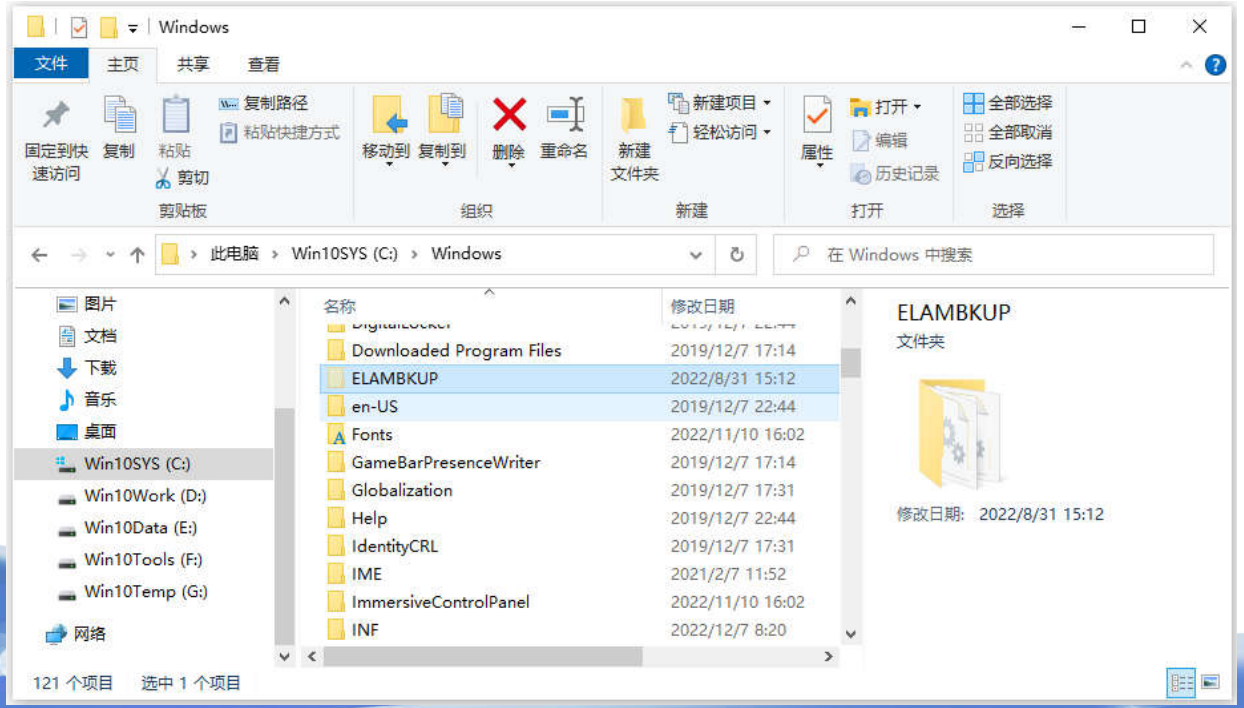
- 按文件的性质
  - 普通文件
  - 目录文件
  - 设备文件

susg : bash

```
[susg@localhost ~]$ ls -l /dev/
total 0
crw-------. 1 root root     10, 235 5月    1 17:06 autofs
drwxr-xr-x. 2 root root          280 5月    1 17:06 block
drwxr-xr-x. 2 root root           80 5月    2  2017 bsg
crw-------. 1 root root     10, 234 5月    1 17:06 btrfs-control
drwxr-xr-x. 3 root root           60 5月    2  2017 bus
lrwxrwxrwx. 1 root root            3 5月    1 17:06 cdrom -> sr0
drwxr-xr-x. 2 root root         3400 5月    1 17:06 char
crw-------. 1 root root      5,    1 5月    1 17:06 console
lrwxrwxrwx. 1 root root           11 5月    2  2017 core -> /proc/kcore
crw-rw----. 1 root video    29,    0 5月    1 17:06 fb0
lrwxrwxrwx. 1 root root           13 5月    2  2017 fd -> /proc/self/fd
```

文件和文件系统概念

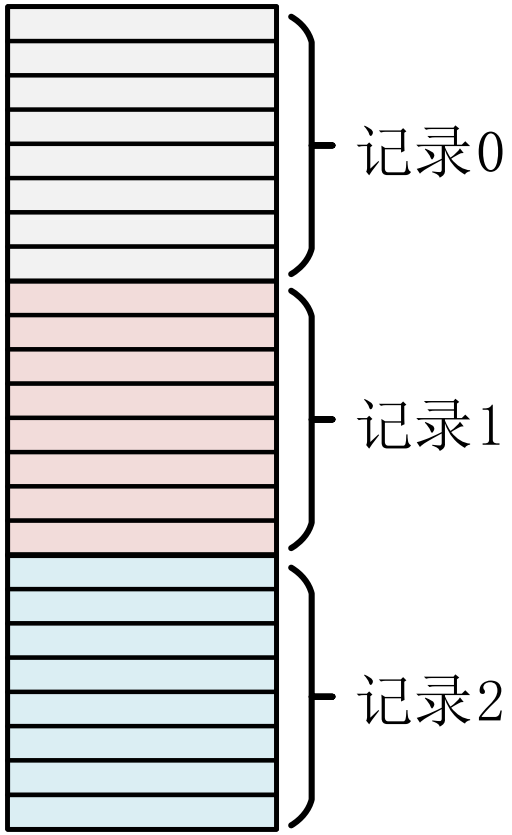
- 文件系统
  - 管理文件的机构称为文件系统。
    - ◆实现文件的创建、撤消、读写、修改、复制和存取控制等
      - □方便用户以文件名存取文件
    - ◆管理文件存储设备的空间和存取。
      - □高效利用存储空间和缩短存取文件时间

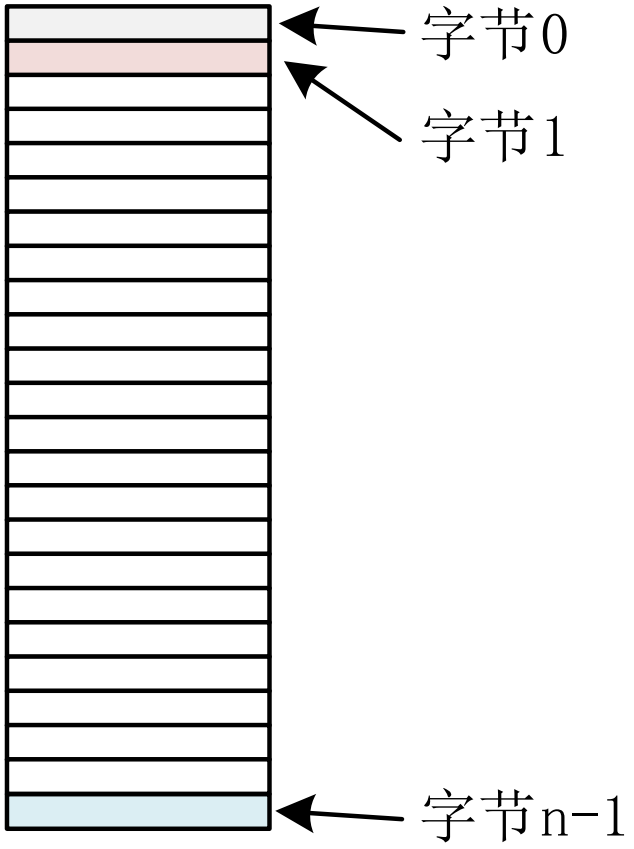


9.2 文件结构

文件的结构

- 文件的逻辑结构
  - 记录式文件
  - 流式文件



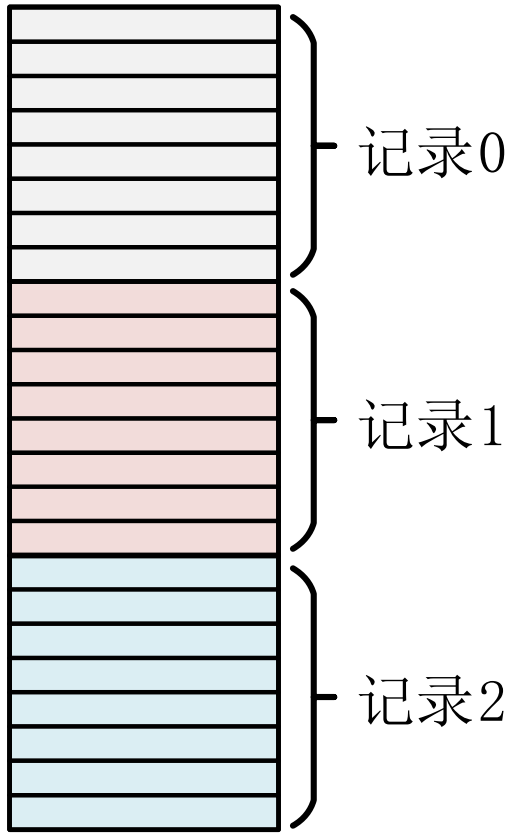


文件的结构

- 文件的逻辑结构
  - ■记录式文件
    - ◆信息项是记录，记录包含若干成员。
      - □例如：学生花名册.txt
        - ▲记录：姓名，学号，性别，成绩
    - ◆特点
      - □文件
      - □浪费
    - ◆分类
      - □定长记录文件
      - □不定长记录文件

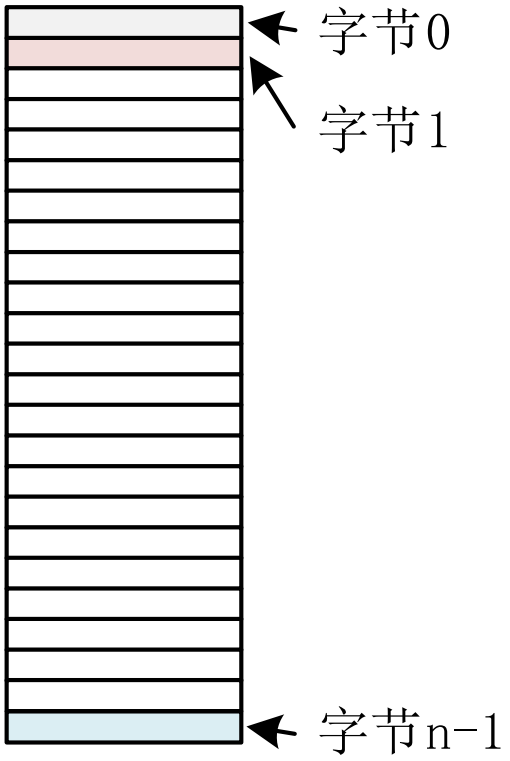
记录长：19；记录数：5			
王宇强	U202010445	男	90
郝新梅	U202054221	女	67
于世强	U202023090	男	79
刘欣悦	U202054882	女	88
张杉杉	U202033528	男	92

信息



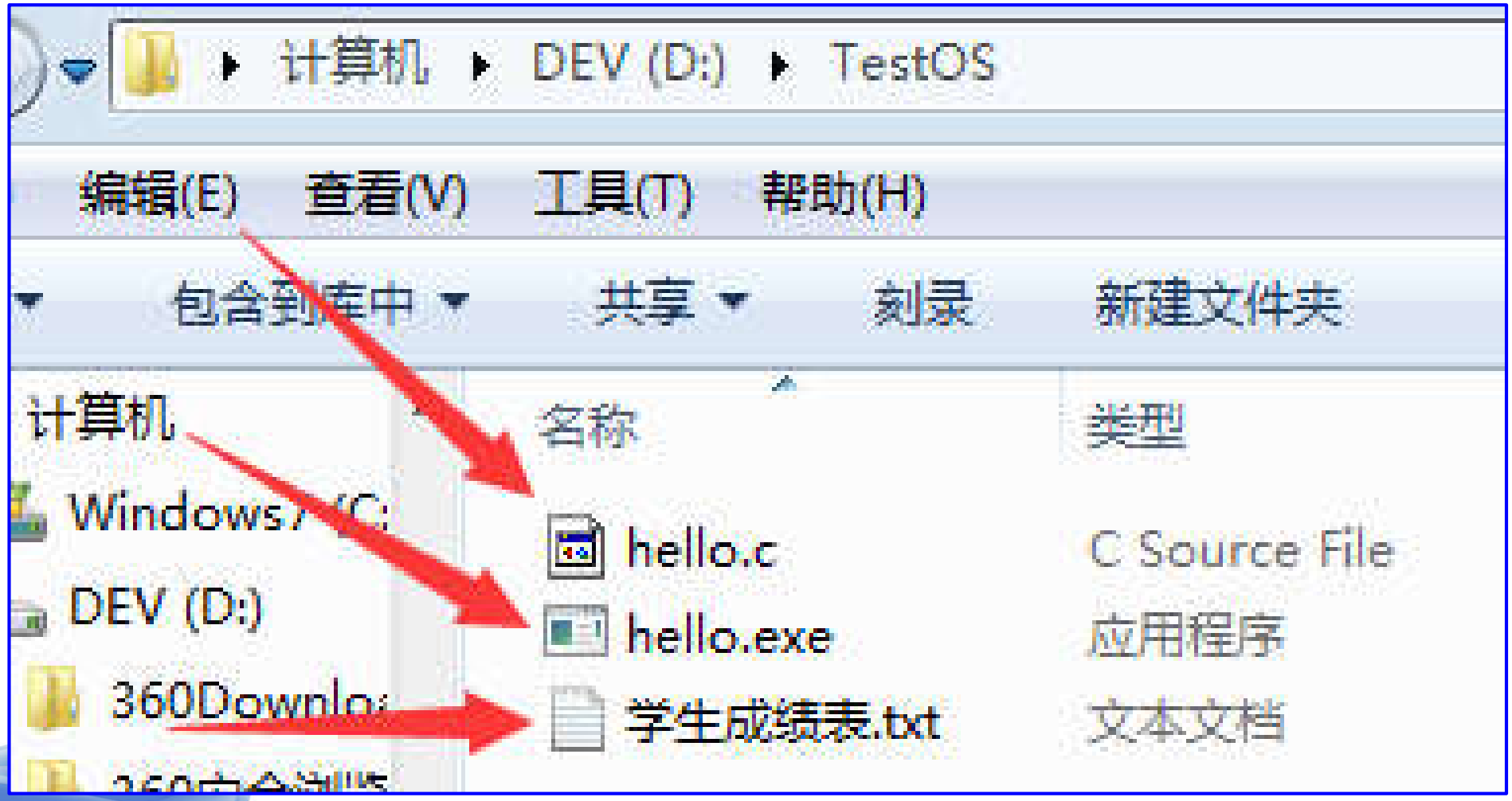
文件的结构

- 文件的逻辑结构
  - 流式文件
    - 信息项是字节
    - 特点
      - 文件长度=字节的数量
      - 无需额外说明/控制信息



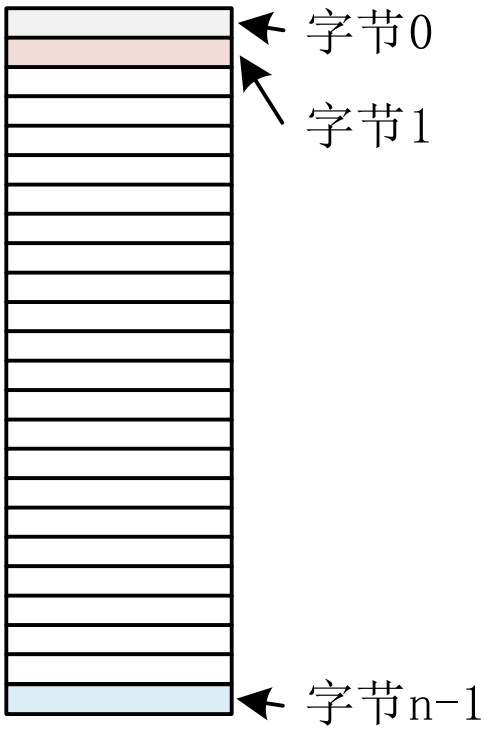
文件的结构

- 文件的逻辑结构
  - ■记录式文件 | 流式文件



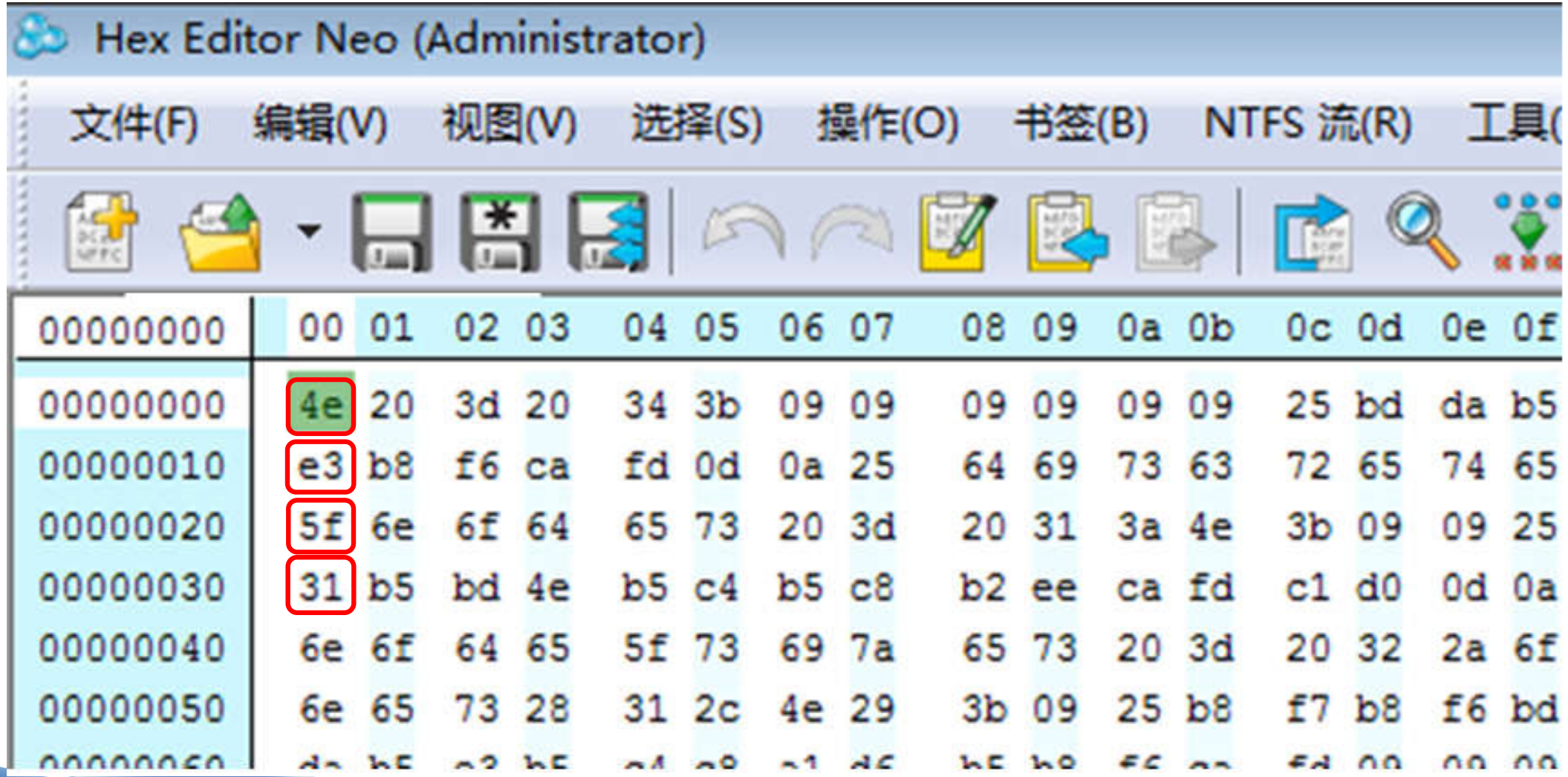
文件的结构

- 文件的逻辑结构
  - 流式文件
    - 信息项是字节。
    - 特点
      - 文件长度就是字节的数量
      - 文件无需额外说明信息或控制信息
  - 现代OS把文件当流式文件，由应用解释



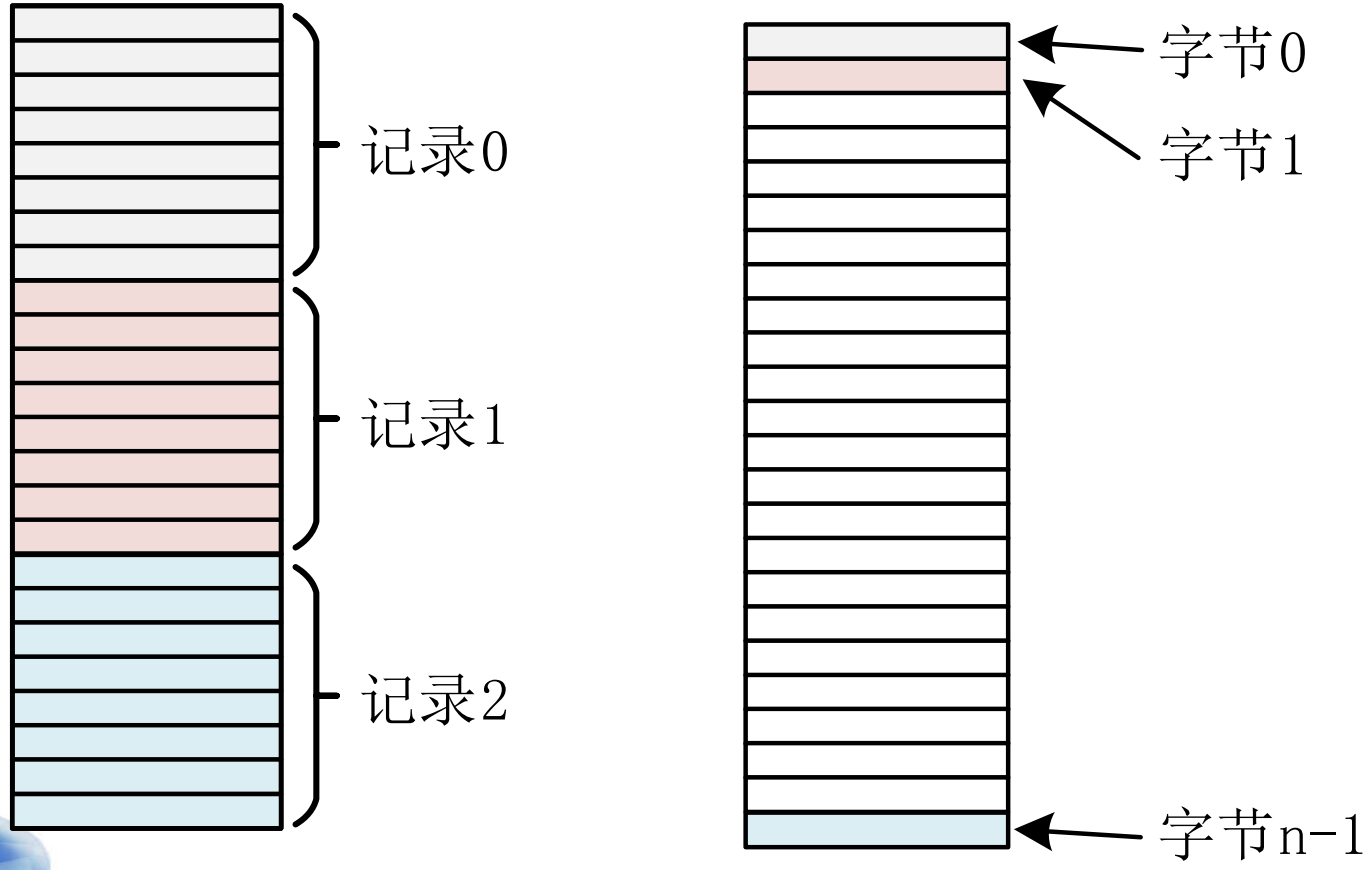
文件的结构

- 流式文件 //示例：DrawCircleDlg.cpp



文件的结构

- 文件的存取方法
  - 顺序存取
  - 随机存取

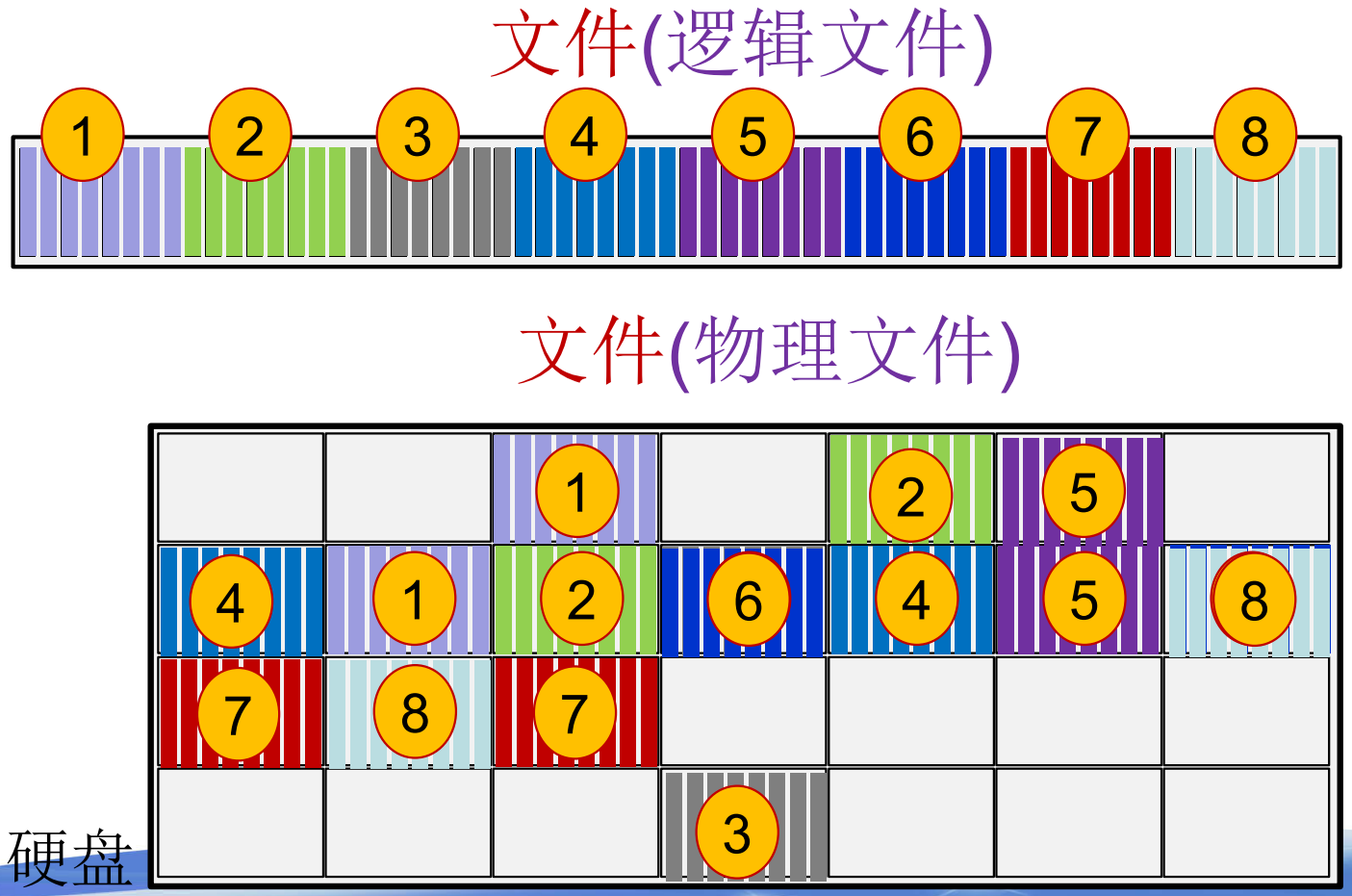


文件的结构

- 文件的存取方法
  - 顺序存取
    - 按从前到后的顺序依次对文件信息项进行读/写，直到定位到目标信息项为止。
  - 随机存取/直接访问
    - 直接定位到文件目标信息项进行读/写。
    - 适合流式文件或定长记录文件。

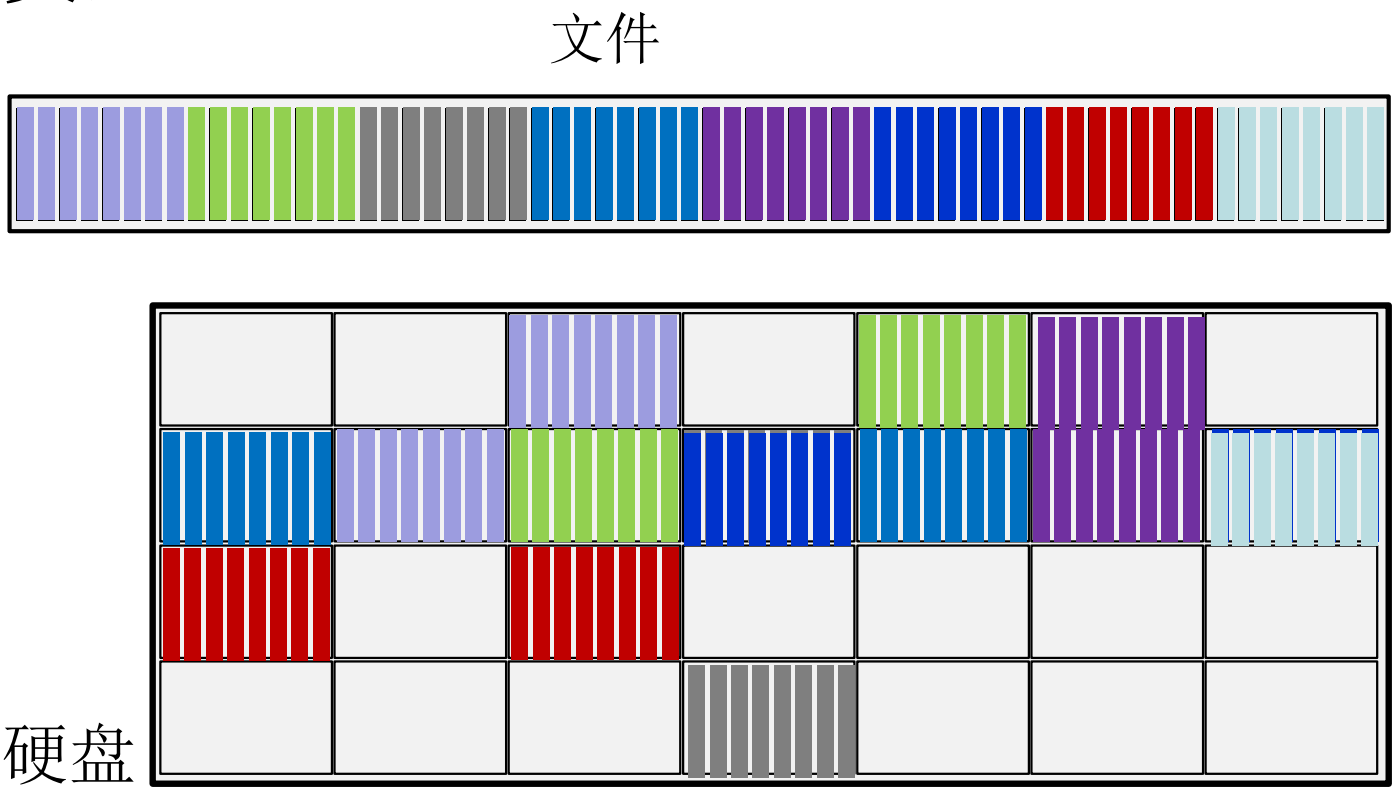
文件的结构

- 文件的物理结构
  - 文件在存储设备上的存储结构
  - 文件的逻辑块 | 文件的存储块（物理块）



文件的结构

- 文件的物理结构
  - 文件在存储设备上的存储结构
  - 强调合理利用储存空间，缩短I/O时间。
- 文件的物理结构类型
  - 连续文件
  - 串联文件
  - 索引文件

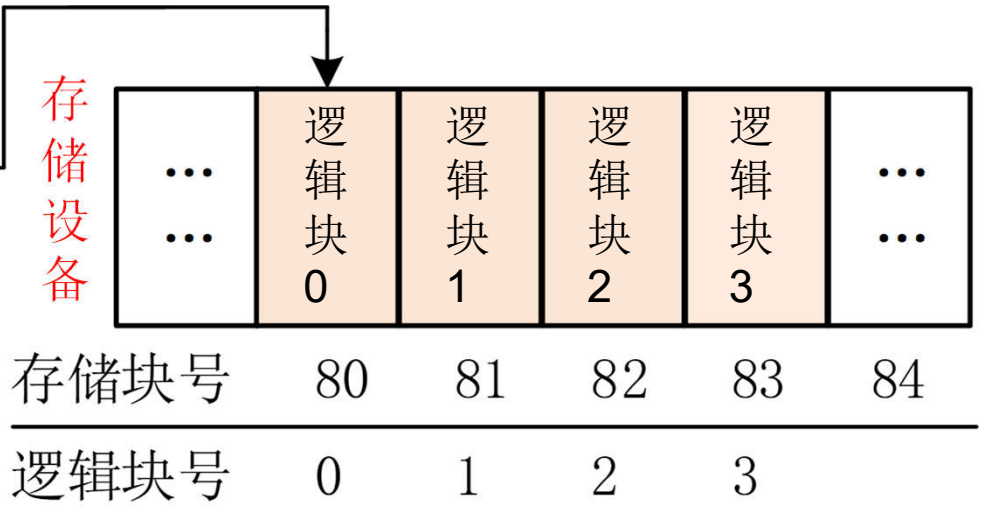


文件的物理结构

- 连续文件
  - 连续文件指文件存放在连续的存储块中。
    - 文件的存储块顺序与逻辑块顺序一致且连续
  - 文件目录记录文件长度(块数)和首个存储块号

文件目录

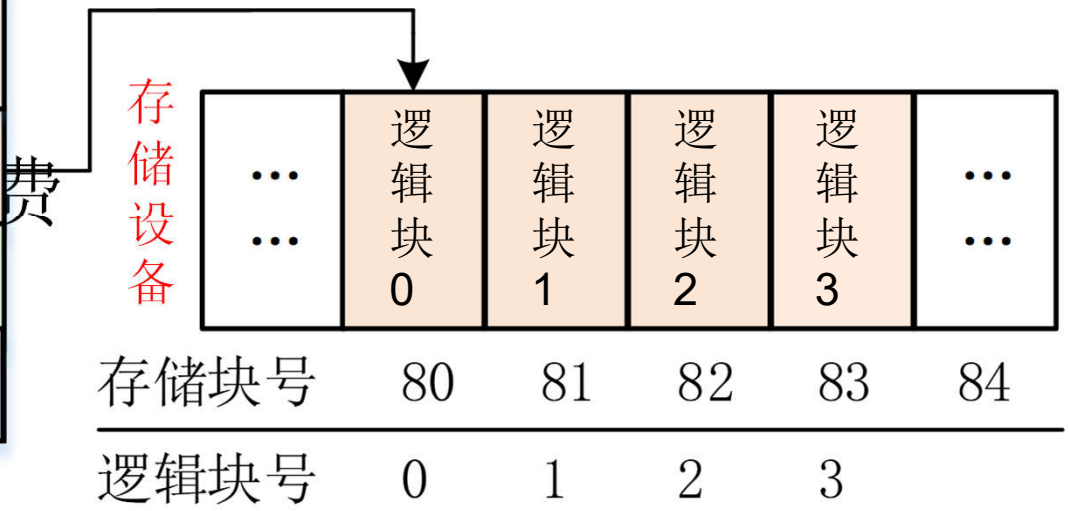
文件名	文件长度	首个存储块号
FileA	4	80
FileB	6	90
FileC	2	66



文件的物理结构

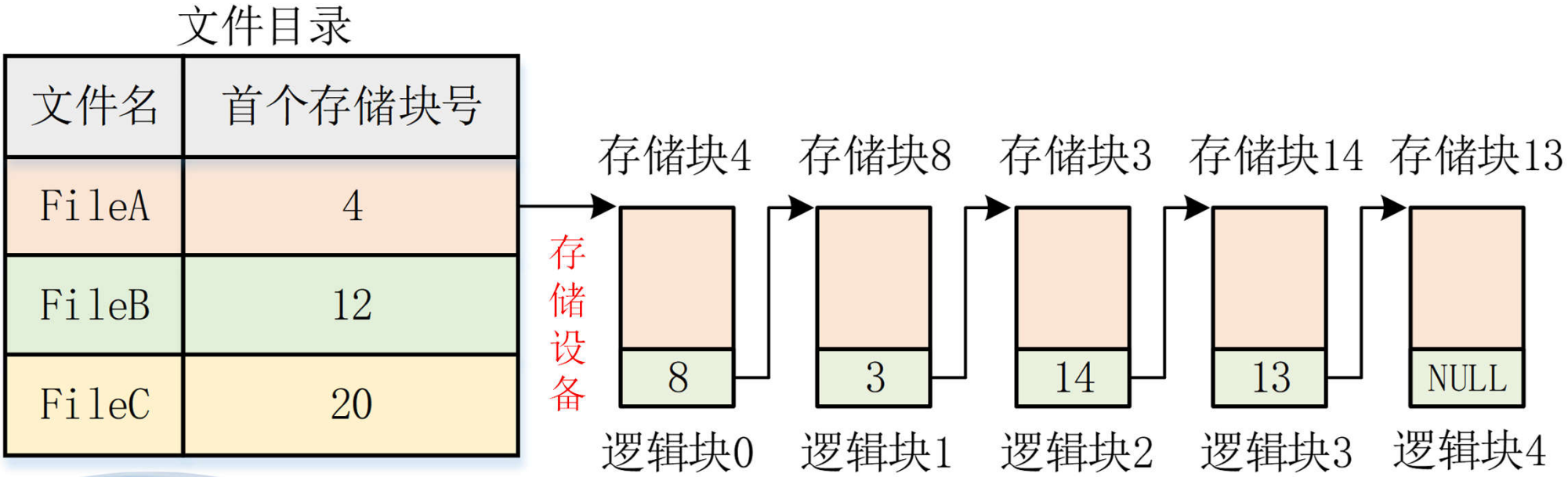
- 连续文件
  - ■特点
    - ◆文件建立时给出文件最大长度和文件起始位置。
    - ◆支持顺序存取和随机存取
      - □顺序存取速度快(寻道次数和寻道时间最少)
  - ■缺点 文件目录

文件名	文件长度	首个存储块号
FileA	4	80
FileB	6	90
FileC	2	66



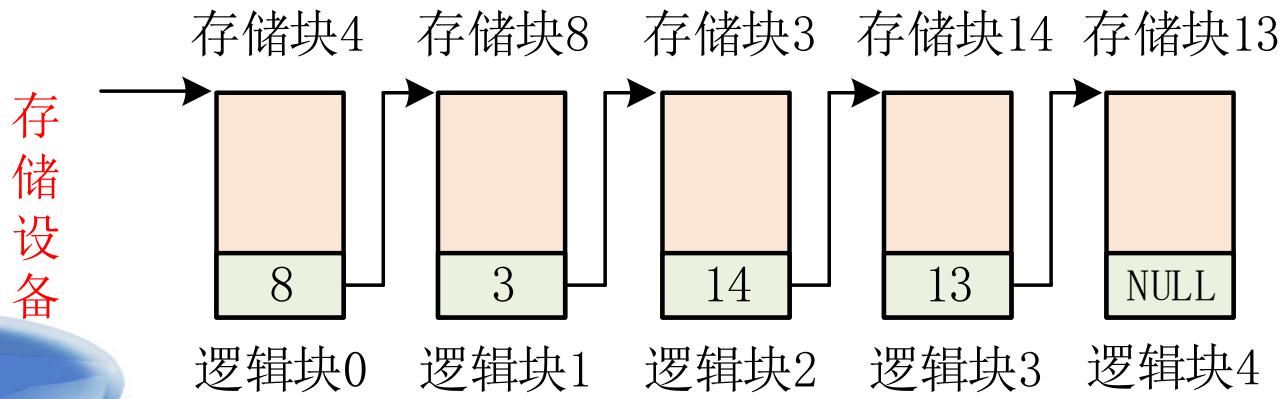
文件的物理结构

- 串联文件
  - 串联文件存放在离散的存储块中，每个存储块包含一个链接指针记录下一个存储块位置。
  - 文件目录记录文件首个存储块号



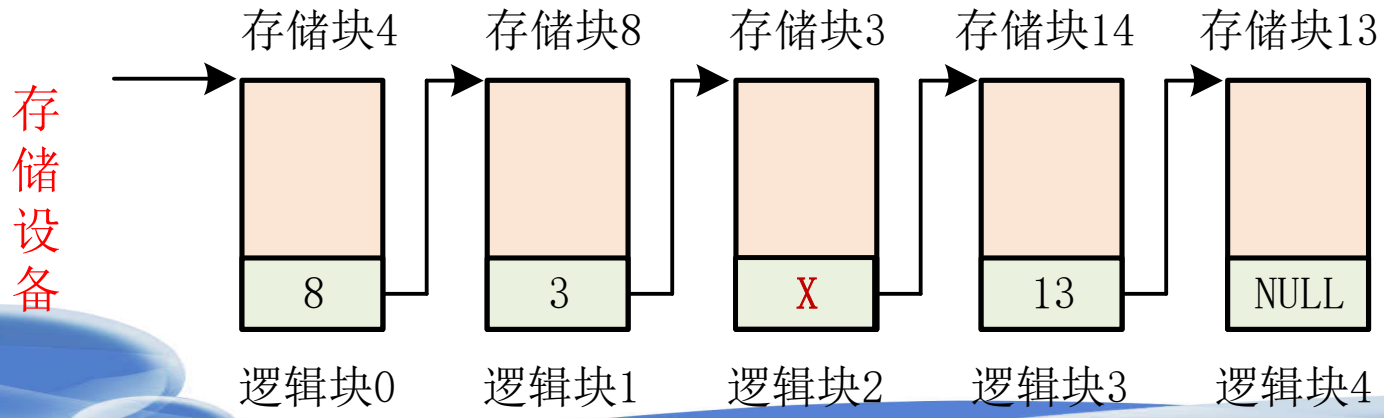
文件的物理结构

- 串联文件
  - 串联文件存放在离散的存储块中，每个存储块包含一个链接指针记录下一个存储块位置。
  - 特点
    - 串联文件可以显著消除存储碎片
    - 创建文件时无须知道文件长度
    - 文件动态增长时可动态分配存储块
      - 支持文件增、删、改等操作。



文件的物理结构

- 串联文件
  - 串联文件存放在离散的存储块中，每个存储块包含一个链接指针记录下一个存储块位置。
  - 缺点
    - 随机访问效率极低(较适合顺序访问方式）
    - 如果某个链接指针损坏，文件后面将无法访问。

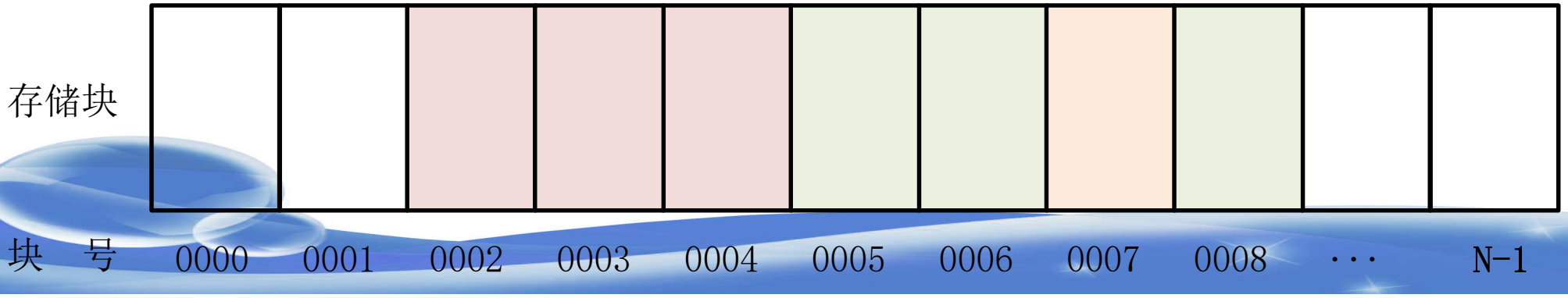


文件的物理结构

- 串联文件
  - 串联文件存放在离散的存储块中，每个存储块包含一个链接指针记录下一个存储块位置。
  - 实例：FAT文件系统
    - File Allocation Table，FAT，文件分配表
    - 文件分配表是一维数组，与存储设备空间对应，元素与存储块一一有序对应，元素存放下一个逻辑块的存储块块号或NULL结束符。文件目录中记录文件的首个存储块号。

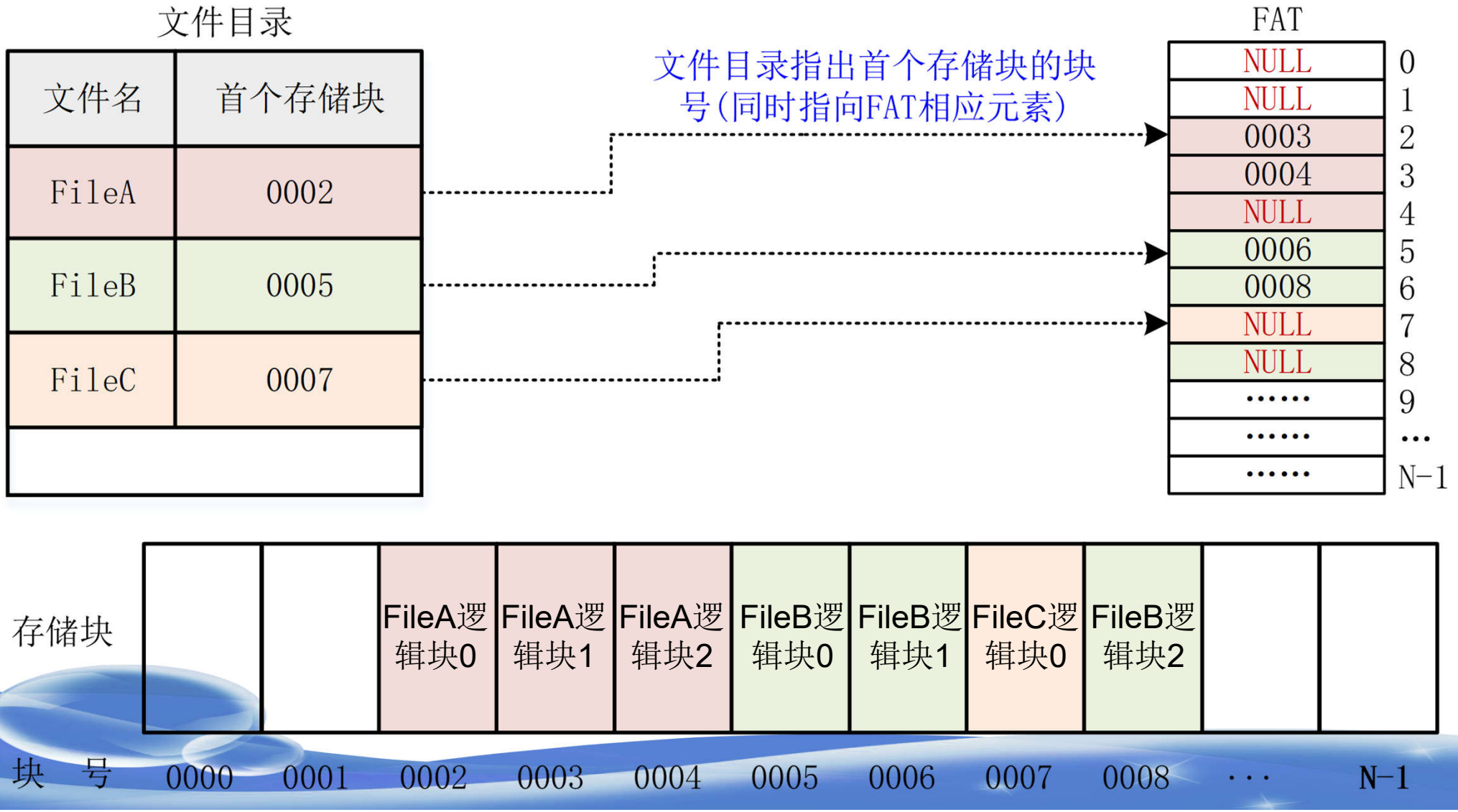
FAT

NULL	0
NULL	1
0003	2
0004	3
NULL	4
0006	5
0008	6
NULL	7
FFFF	8
……	9
……	…
……	N-1



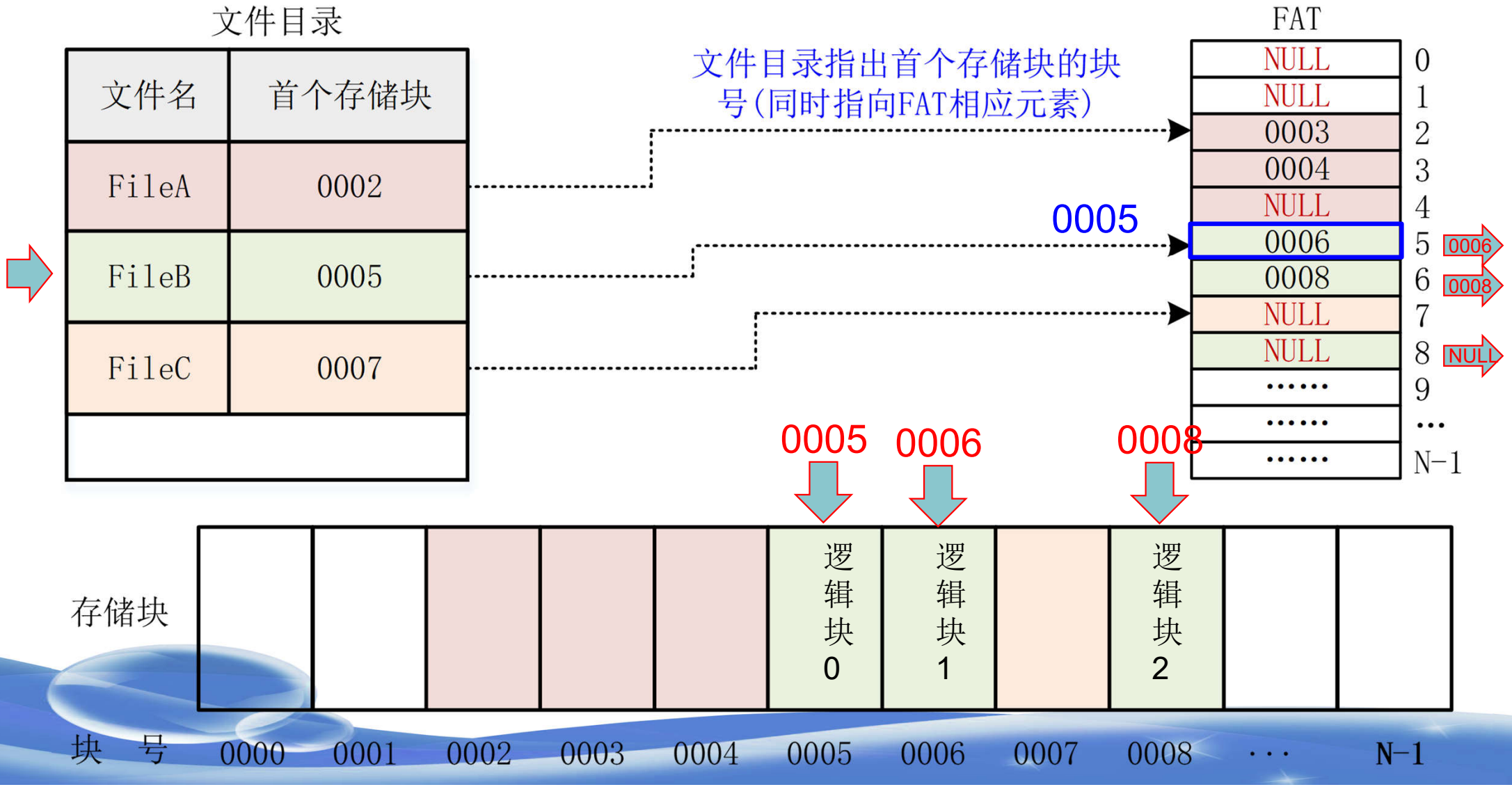
实例：FAT文件系统

- **FAT/ File Allocation Table，FAT**
  - ■文件分配表



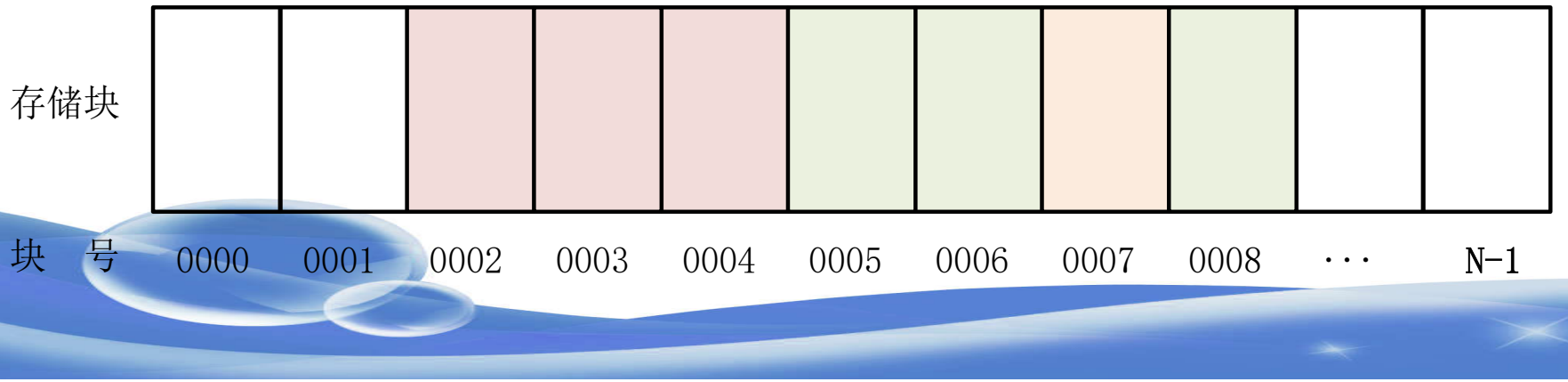
【不要动】实例：FAT文件系统

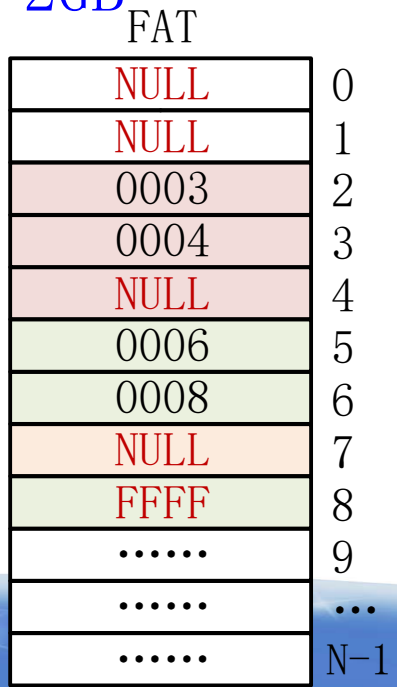
- **FAT/ File Allocation Table，FAT**
  - ■文件分配表



实例：FAT文件系统

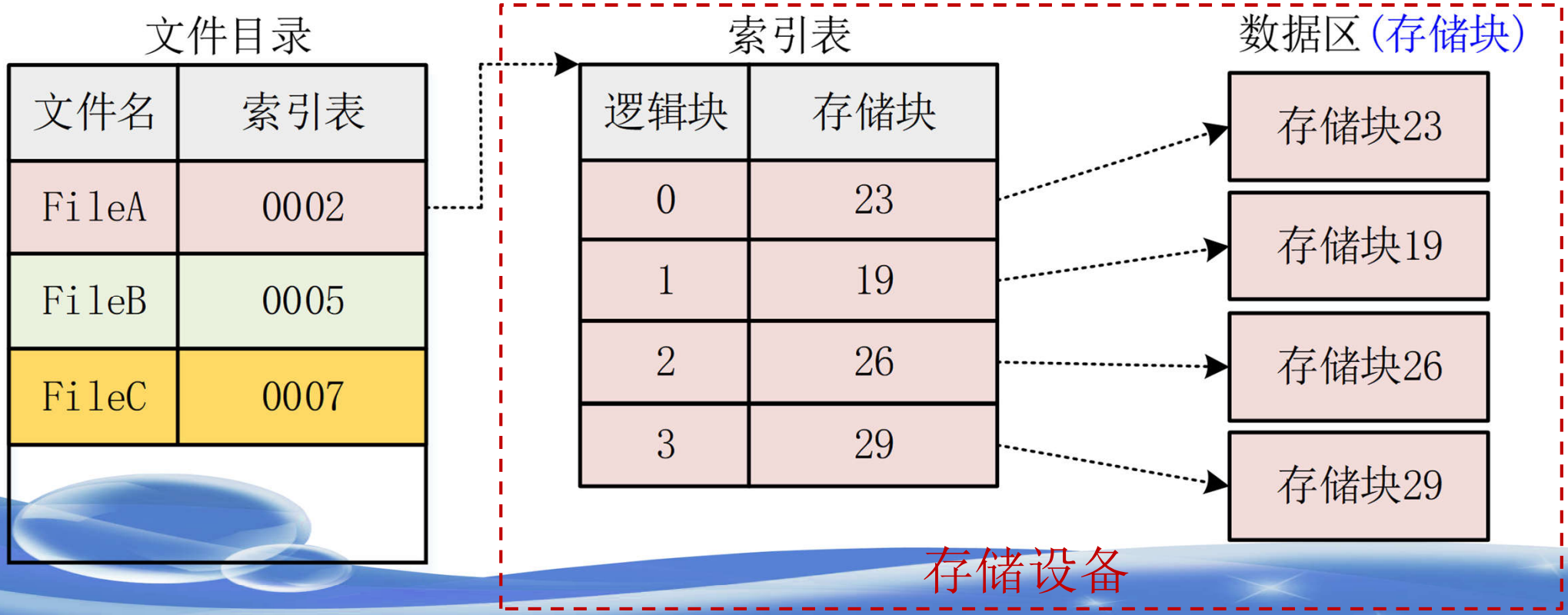
- **FAT/ File Allocation Table，FAT**
  - 容量越大或存储块越小，则FAT项越多(宽度/位数越大)；
  - 若FAT项越多(宽度/位数越大)，则支持越大容量或越小块；
  - FAT16文件系统
    - FAT元素16位宽，存储块(簇)数 ≤ 2^{16}簇
    - 存储块（簇） = 64个扇区 （可调：2^n个扇区）
    - 磁盘容量 ≤ 2^{16}簇 * 64扇区 * 512字节 = 2GB
  - FAT32文件系统
    - FAT元素32位宽





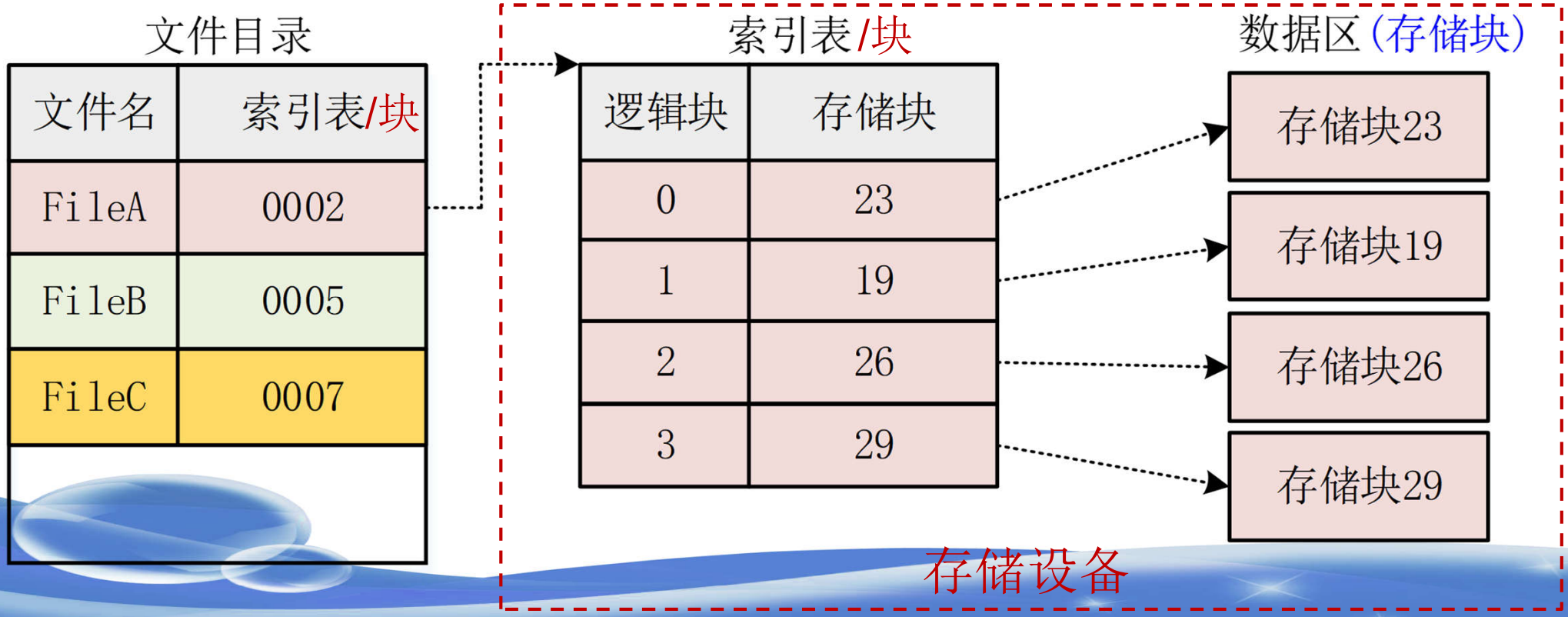
文件的物理结构

- 索引文件
  - 文件存放在不连续的存储块中，系统建立索引表记录文件逻辑块和存储块的对应关系。
    - 索引文件 = 索引表 + 数据区(存储数据的存储块)
  - 文件目录记录文件名和对应的索引表。



文件的物理结构

- 索引文件特点
  - 读取索引文件时先读取索引表
  - 索引表本身占据额外的存储区域/缺点
  - 支持顺序和随机存取；支持文件动态增长、插入、删除等
  - 实例：minix文件系统，ext文件系统(inode索引节点文件)

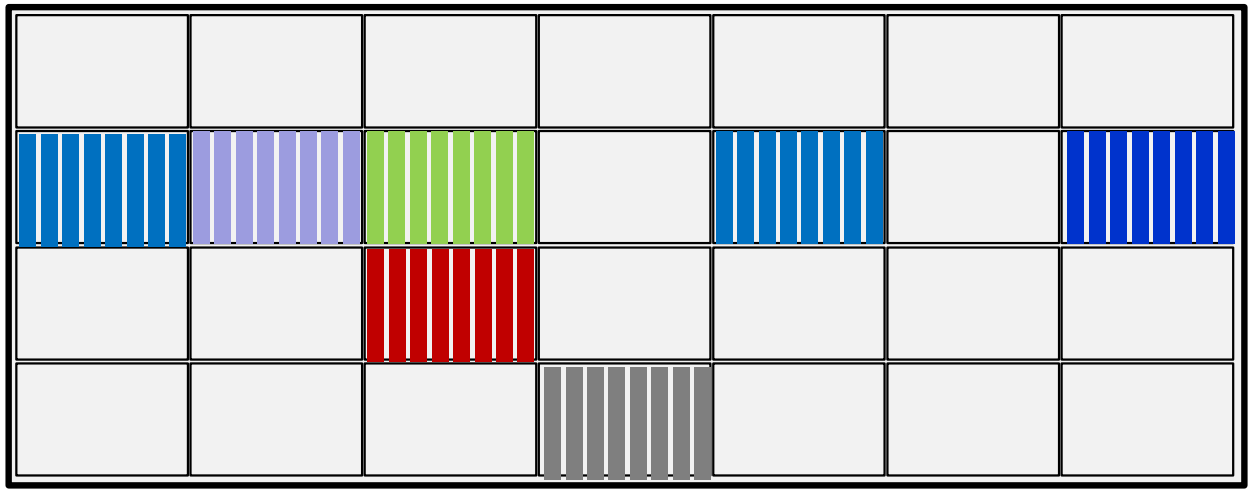


9.3 磁盘存储空间管理

磁盘存储空间管理

- 磁盘存储空间管理
  - 管理和记录磁盘空间使用情况。
- 磁盘空闲存储块管理方法
  - 空闲文件目录
  - 空闲块链
  - 位示图

磁盘（灰色块空闲）

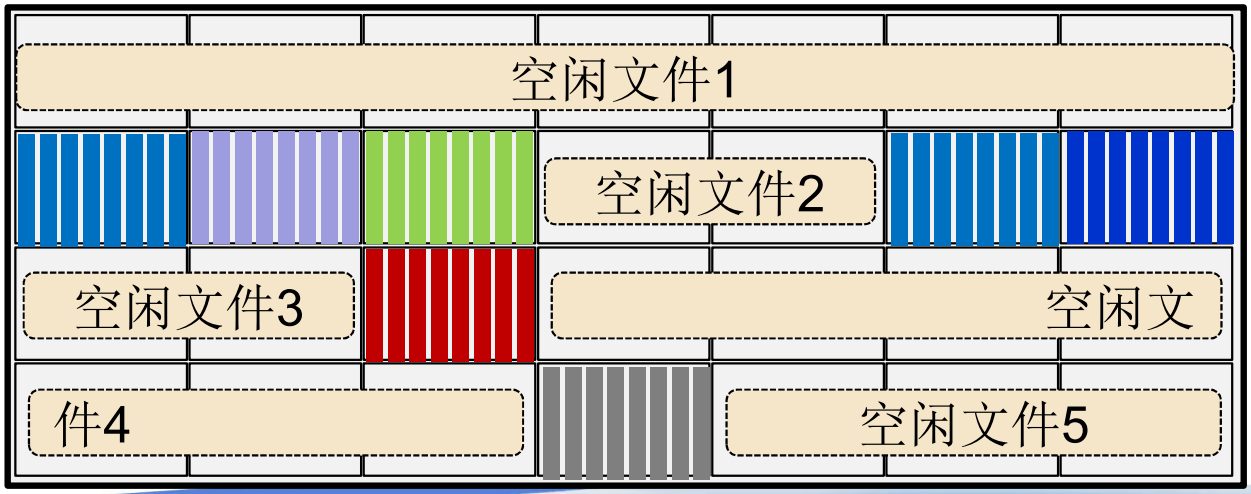


磁盘空闲存储块管理方法

- 空闲文件目录
  - ■空闲文件：连续的空闲存储块组成的特殊文件。
    - ◆存储设备上所有的空闲文件就代表了存储设备上的全部空闲空间。
  - ■空闲文件目录：为所有空闲文件建立的目录
    - ◆记录空闲文件的首块号和存储块数(或其他方式)

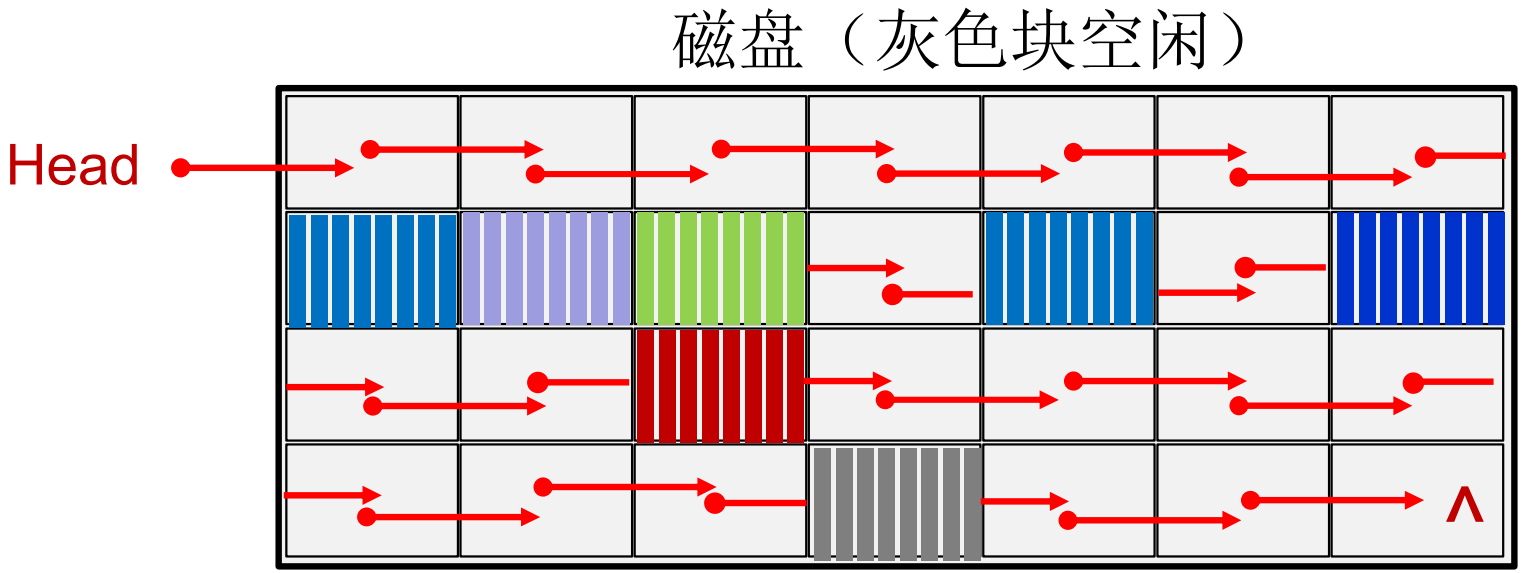
空闲文件	首块号	块数
空闲文件1	0	7
空闲文件2	10	2
空闲文件3	14	2
空闲文件4	17	7
空闲文件5	25	3

磁盘（灰色块空闲）



磁盘空闲存储块管理方法

- 空闲块链
  - 把所有空闲存储块用链表链接在一起。
    - 当申请空闲块时，从链表头部摘取空闲块
    - 当回收存储块时，把空闲块加在链表尾部。

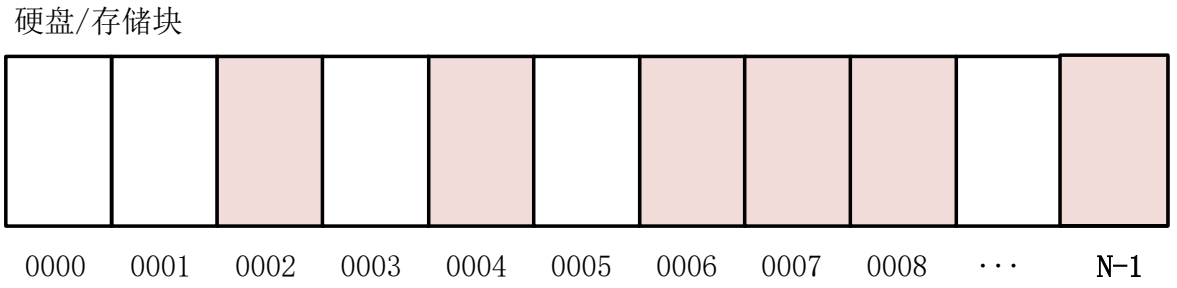


磁盘空闲存储块管理方法

- 位示图
  - 一块特殊内存区域，每一位(bit)对应一个存储块，值1表示存储块空闲，0表示已占用。

例：N=64块

2000H	1	1	**0**	1	**0**	1	**0**	**0**
	0	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1
2007H	1	1	1	1	1	1	1	0



9.4 文件目录

文件目录

- 文件目录的功能
  - 实现“按名存取”：系统根据文件名能找到指定文件。
    - ◆结构：文件目录项（directory entry）
      - □记录文件的文件名、存放地址以及相关属性。

文件目录

文件名	文件长度	首个存储块号
FileA	4	80
FileB	6	90
FileC	2	66

文件目录

文件名	首个存储块
FileA	0002
FileB	0005
FileC	0007

文件目录

文件名	索引表
FileA	0002
FileB	0005
FileC	0007

文件目录

- 文件目录项（directory entry）
  - 描述文件基本信息、使用信息和存取控制信息等。
    - 基本信息：文件名、存储地址(存储块号)等
    - 使用信息：属性、大小、建立时间、修改时间等
    - 存取控制信息：文件存取权限

文件目录

文件名	文件长度	首个存储块号
FileA	4	80
FileB	6	90
FileC	2	66

文件目录

文件名	首个存储块
FileA	0002
FileB	0005
FileC	0007

文件目录

文件名	索引表
FileA	0002
FileB	0005
FileC	0007

文件目录

- 文件目录项（**directory entry**）
  - 示例：MS-DOS的文件目录项(FAT文件系统，32B)

8字节	3字节	1字节	10字节	2字节	2字节	2字节	4字节
文件名	扩展名	属性	保留	时间	日期	首块号	大小
文件名	扩展名	属性	保留	时间	日期	首块号	大小
文件名	扩展名	属性	保留	时间	日期	首块号	大小

  - 属性(1字节)
    - 示例1：00000001 只读文件
    - 示例2：00000010 隐含文件

文件目录

- 文件目录项（**directory entry**）
  - ■示例：i_node索引节点的文件目录项(Minix文件)
    - ◆包含文件名和索引表块号（即索引节点号）
      - □检索文件时根据索引节点号访问索引节点，索引节点中包含文件的索引表和文件属性。

```
1  struct dir_entry {// Linux 0.11 (Minix文件)
2      unsigned short inode;
3      char name[NAME_LEN];
4  };
```

文件名	索引节点号
test.c	204901
os.doc	755407

索引表/块

逻辑块	存储块
0	23
1	19
2	26
3	29

文件属性：读/写/...

文件目录

- 文件目录项（**directory entry**）
  - ■示例：i_node索引节点的文件目录项(ext2文件系统)
    - ◆包含文件名和索引表块号(即索引节点号)

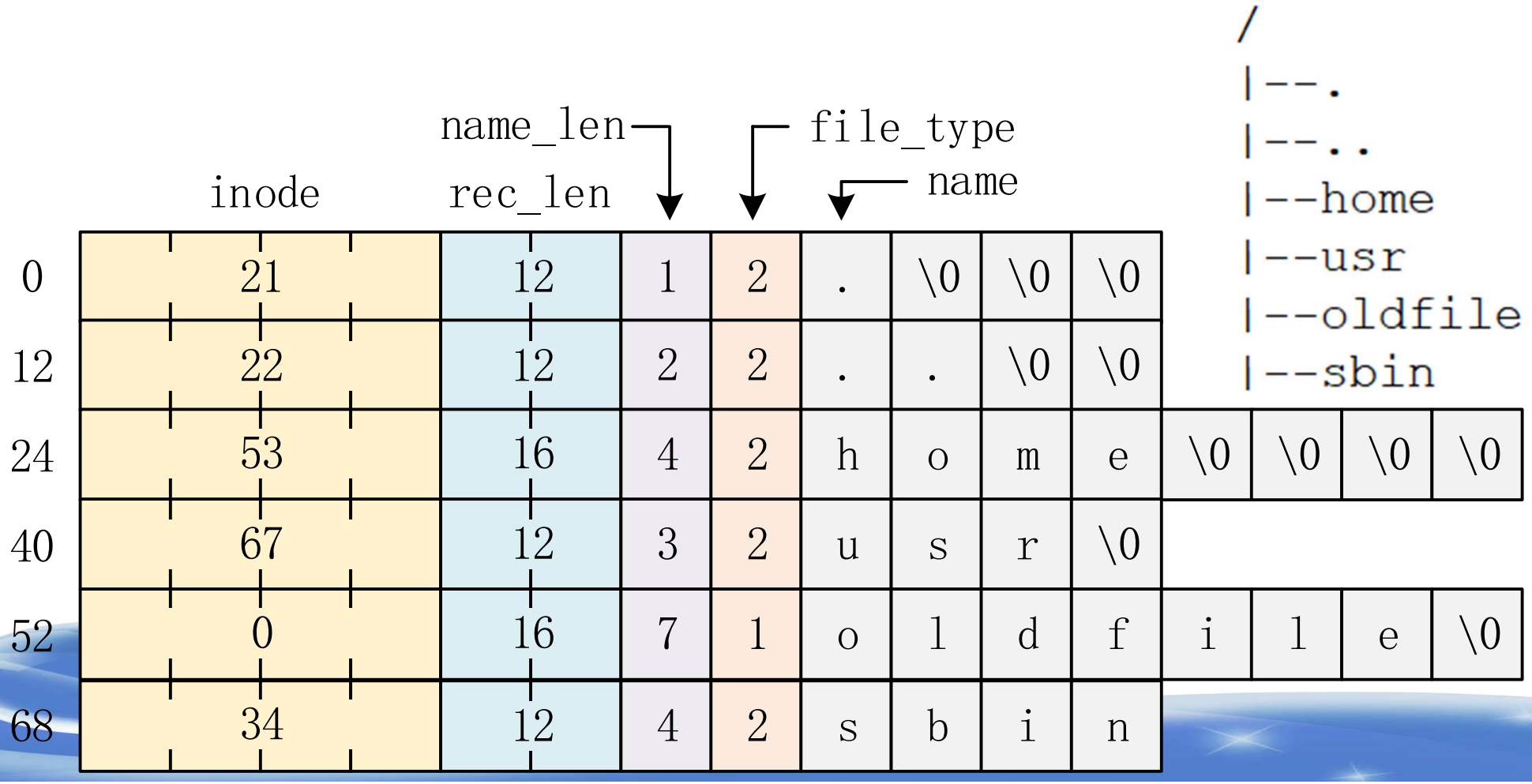
```
1  struct ext2_dir_entry_2
2  {
3      __le32  inode;    /* Inode number */
4      __le16  rec_len;  /* Directory entry length */
5      __u8    name_len;  /* Name length */
6      __u8    file_type;
7      char    name[EXT2_NAME_LEN];  /* File name */
8  };
```

文件名	文件类型	名字长	记录长	索引节点号
test.cpp	2	8	16	204901
os.doc	1	6	14	755407

7 6 5 4 3 0

文件目录

- 文件目录项（**directory entry**）
  - ■示例：i_node索引节点的文件目录项(ext2文件系统)
    - ◆包含文件名和索引表块号(即索引节点号)



文件目录

- 目录文件
  - 目录文件是文件目录的实现，由文件目录项构成。

文件目录

文件名	文件长度	首个存储块号
FileA	4	80
FileB	6	90
FileC	2	66

文件目录

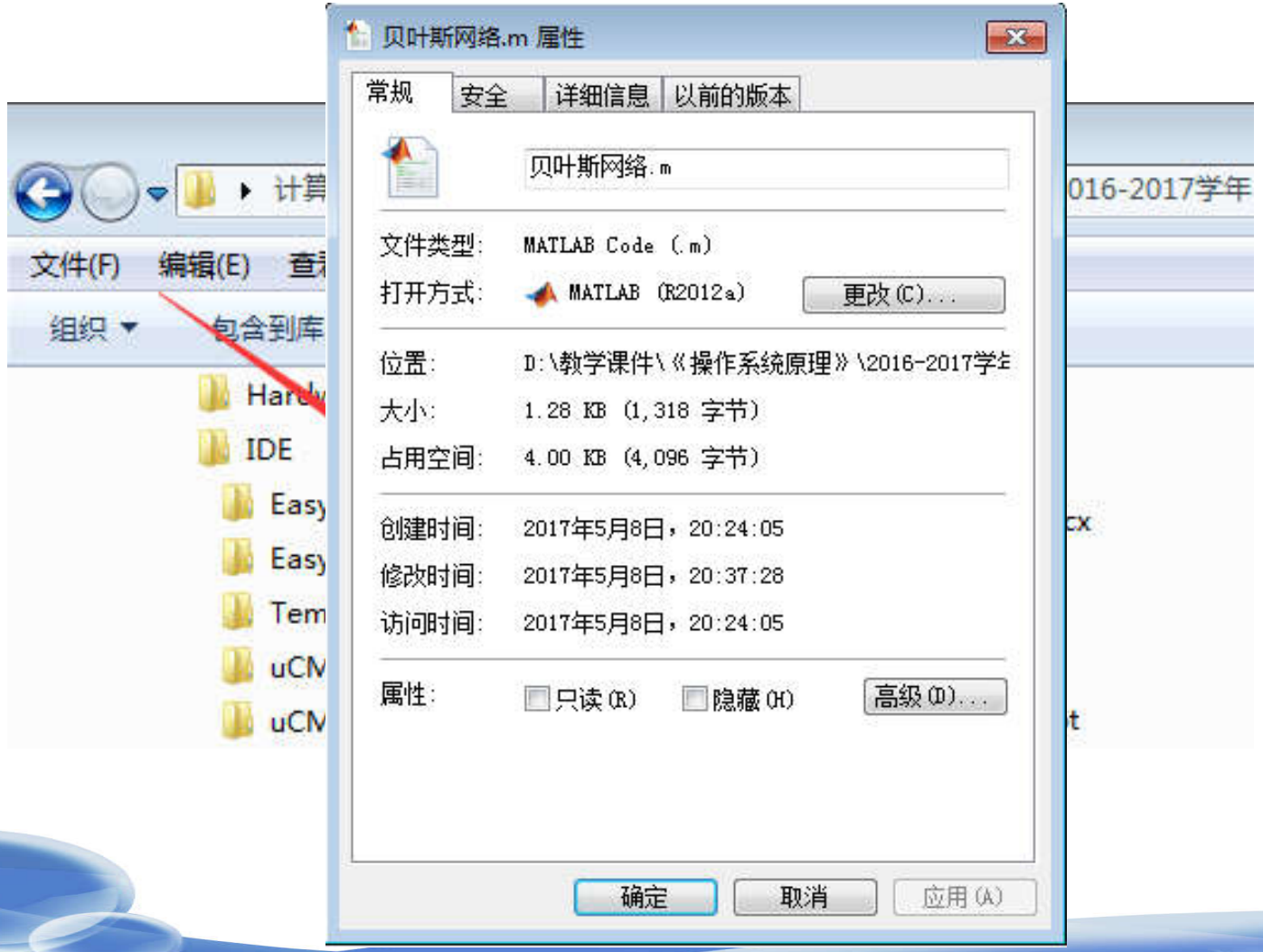
文件名	首个存储块
FileA	0002
FileB	0005
FileC	0007

文件目录

文件名	索引表
FileA	0002
FileB	0005
FileC	0007

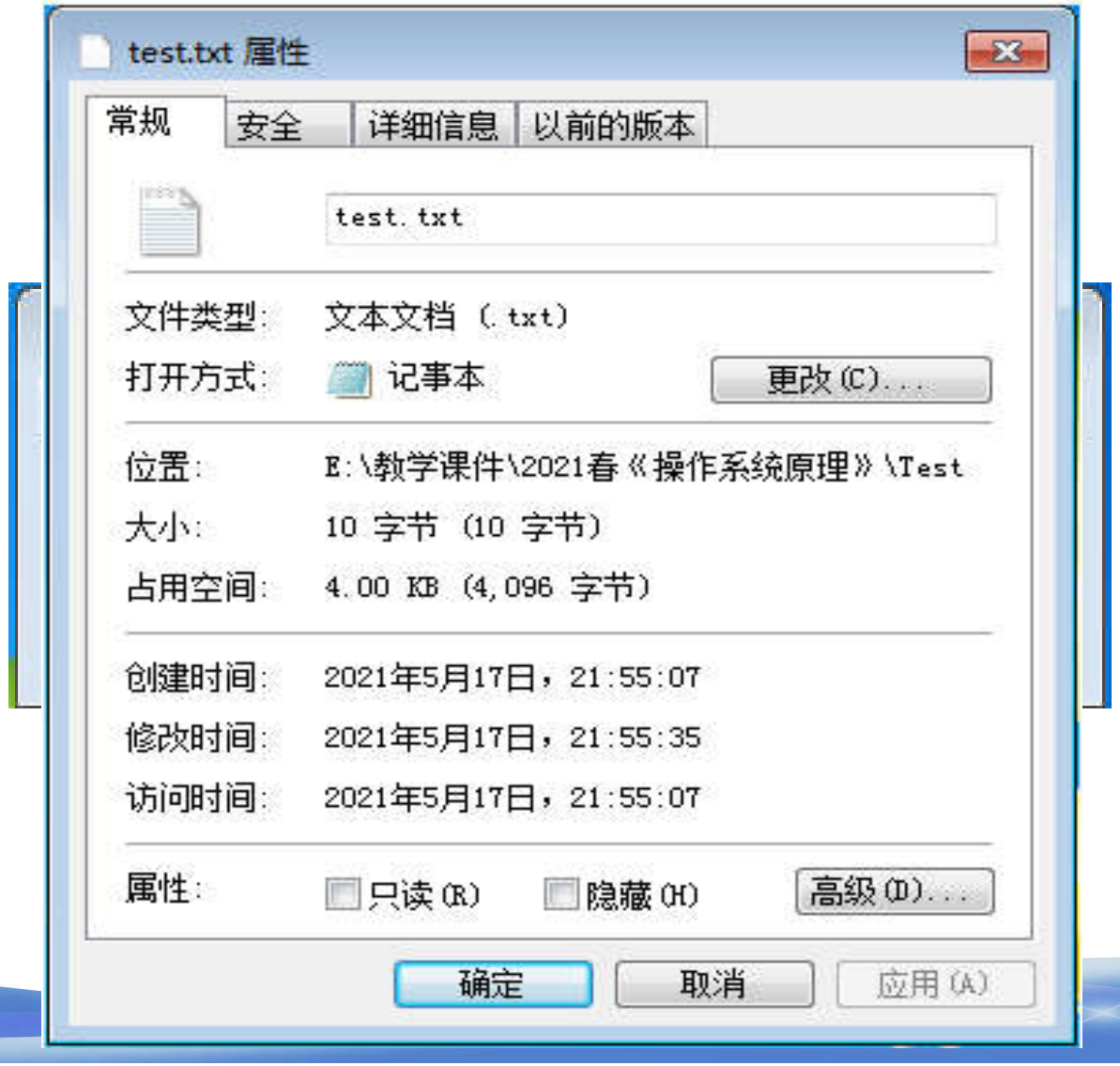
讨论：文件目录与文件属性

- 文件属性
  - ■指定文件的类型、操作特性和存取保护等信息。



讨论：文件目录与文件属性

- 文件属性
  - ■指定文件的类型、操作特性和存取保护等信息。



讨论：文件目录与文件属性

- 文件属性
  - ■指定文件的类型、操作特性和存取保护等信息。
  - ■文件的属性一般存放在文件的（目录 /文件）中。

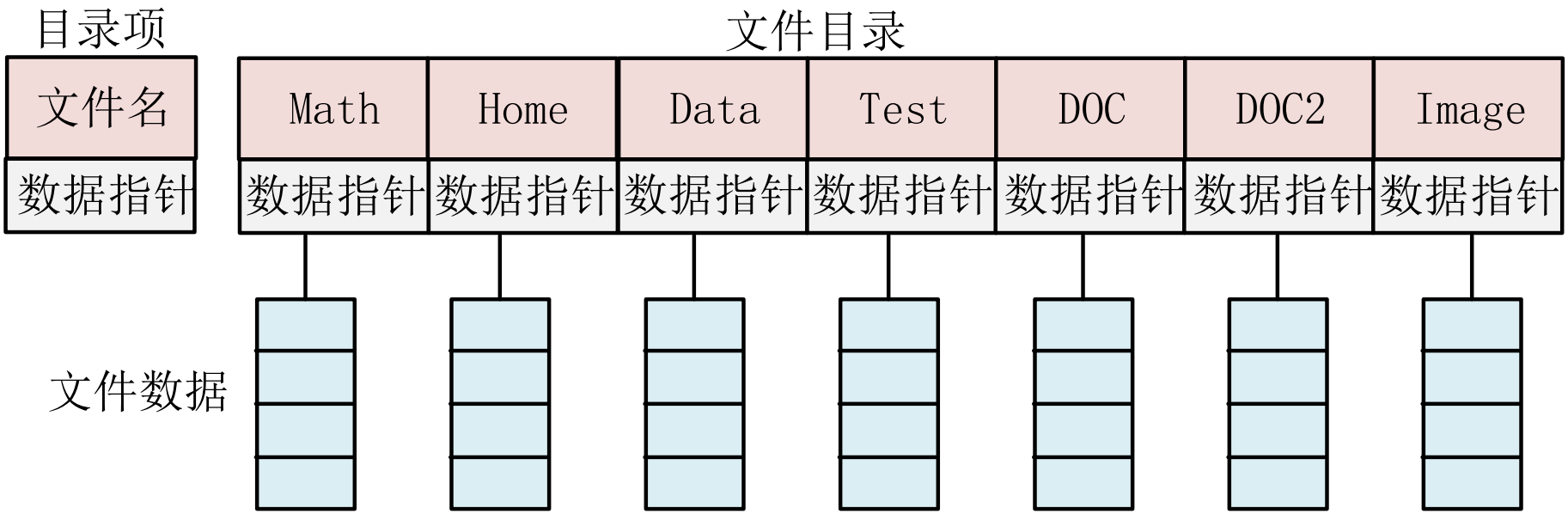
  - ■讨论：文件属性的传递过程
    - ◆c:/test.txt：只读，2024.02.19
    - ◆copy c:/test.txt d:/test.txt
      - □d:/test.txt：只读，2024.02.19

文件目录

- 目录结构
  - 单级目录
  - 二级目录
  - 多级目录（树型目录）
- 文件全名

文件目录

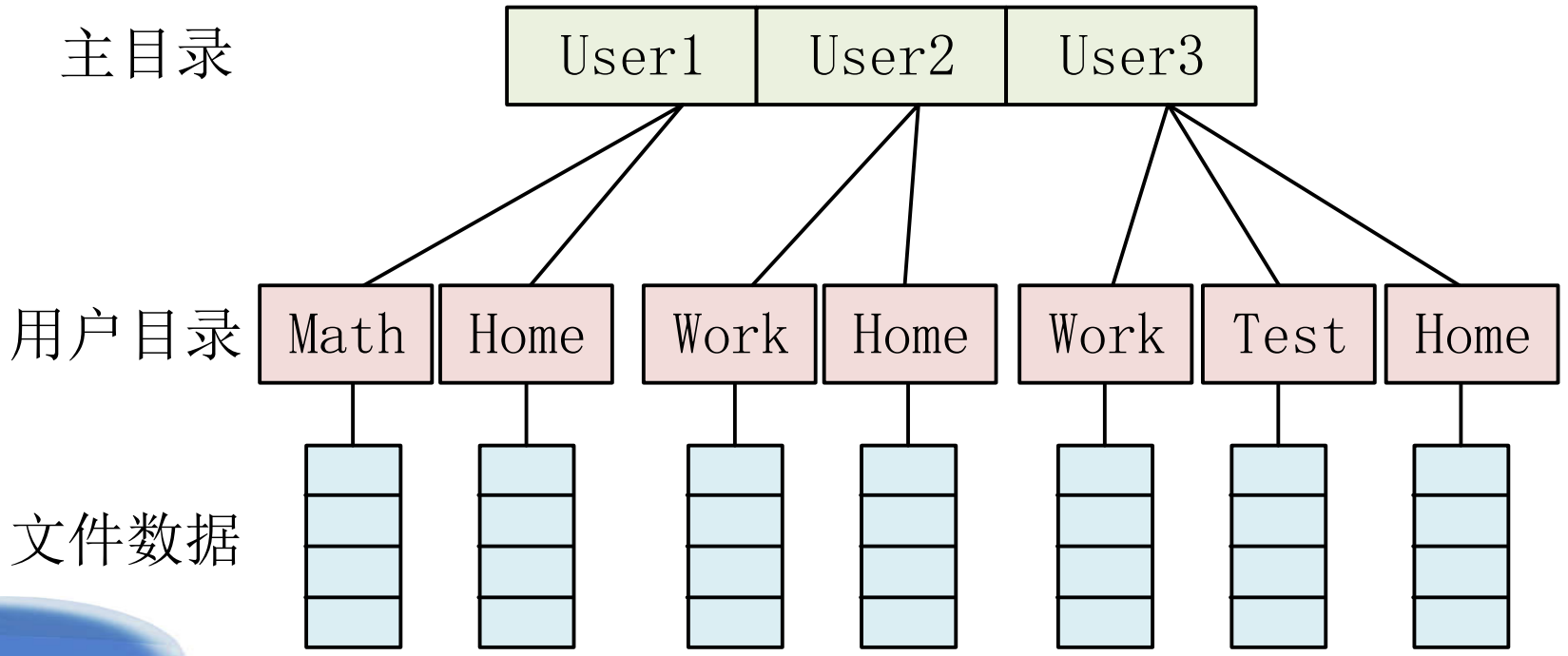
- 单级目录
  - 最简单的目录结构，全部文件都登记在同一目录中。



  - 特点
    - 简单、易于理解和实现
  - 缺点
    - 查找速度慢 | 不允许重名 | 不便于文件共享

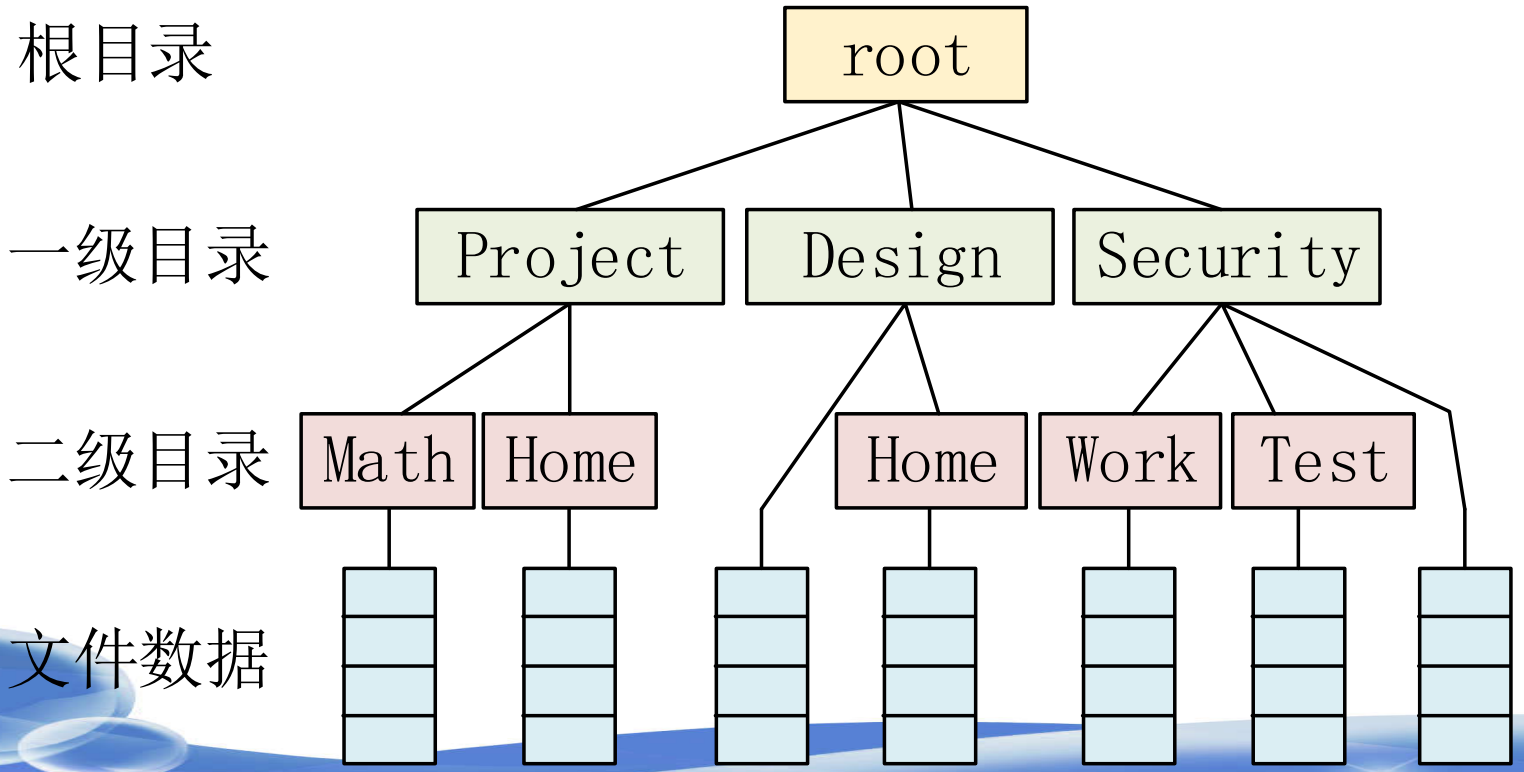
文件目录

- 二级目录
  - 第一级称为主目录(MFD)，第二级称为子目录或用户目(UFD)。
    - 每个用户有一个子目录(用户目录)
  - 优点
    - 解决文件重名的问题，不同用户可以使用相同名字。



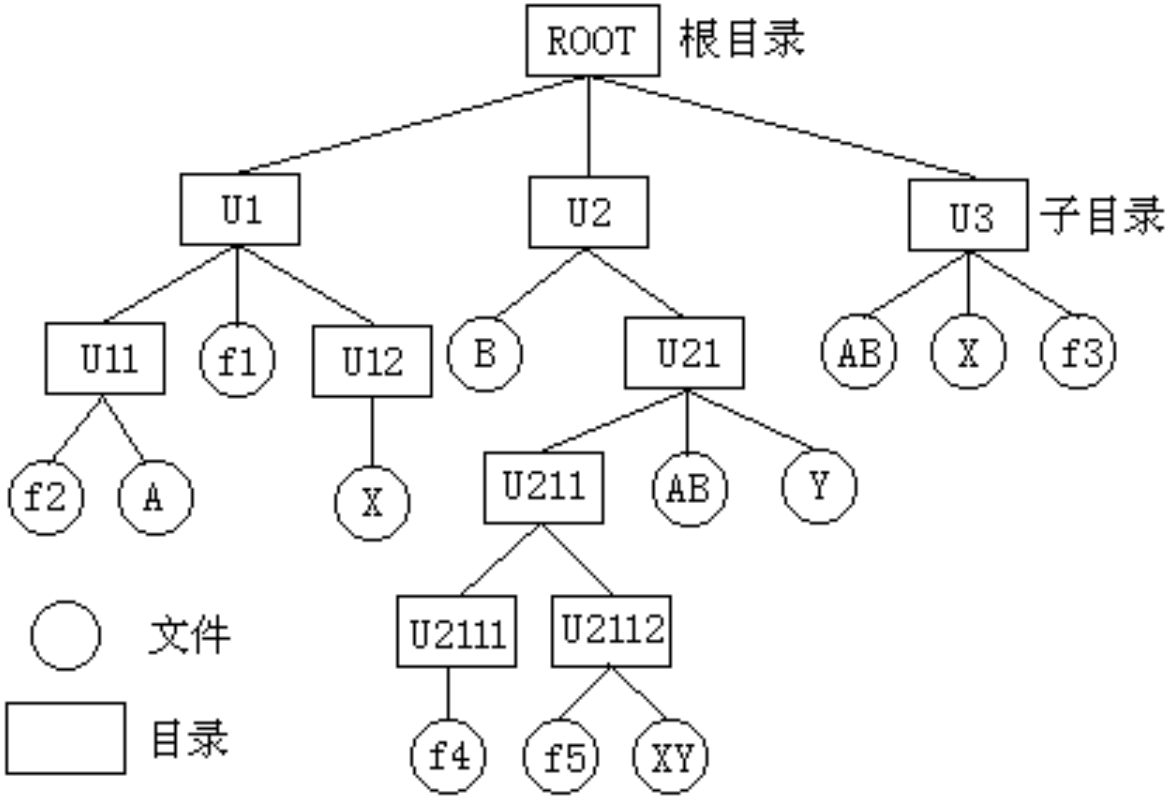
文件目录

- 二级目录
  - 第一级称为主目录(MFD)，第二级称为子目录或用户目(UFD)。
    - 每个用户有一个子目录(用户目录)
  - 优点
    - 解决文件重名的问题，不同用户可以使用相同名字。



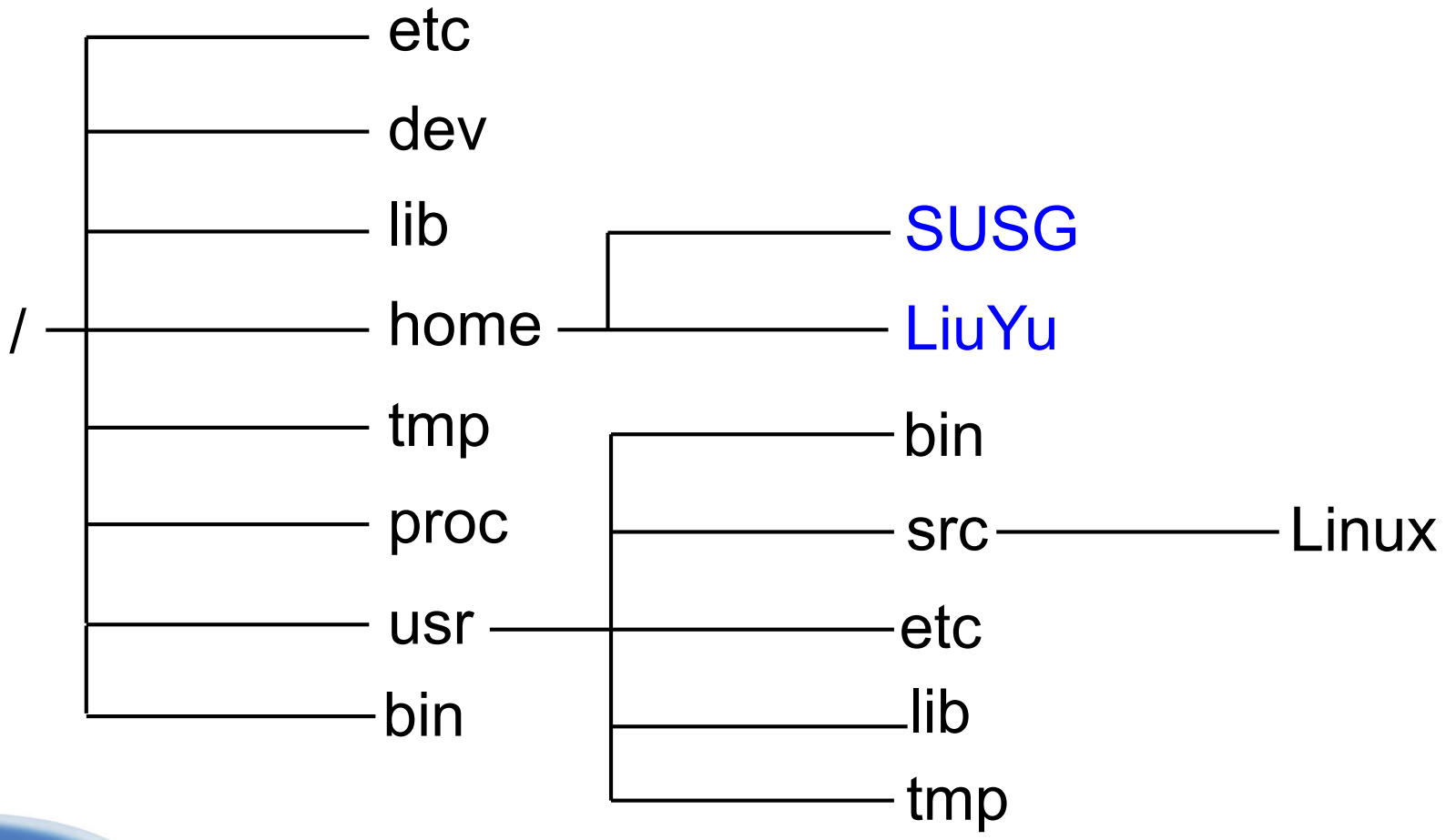
文件目录

- 树形目录
  - 多级目录结构，二级目录结构的扩充
  - 目录结构如同倒置的树，树根是主目录(根目录)，枝结点是子目录，树叶描述文件。



文件目录

- 树形目录
  - ■示例：Linux目录结构



文件目录

- 文件全名
  - 从根目录到文件为止整个通路上所有目录、子目录和文件的名字用"/"顺序连接构成的字符串称为文件全名。
    - 路径名：文件全名中由目录和子目录组成的部分。
      - 每个文件都有惟一的路径名。
    - 路径名的表达形式
      - 绝对路径名：从根目录直到文件的路径
      - 相对路径名：从指定目录到文件的路径

9.3 文件和目录的操作

文件和目录的读写

- 文件操作
  - 创建文件
  - 写文件
  - 读文件
  - 文件定位
  - 删除文件
  - 截短文件
  - 属性设置和读取
- 目录操作
  - 创建目录
  - 删除目录

Create
Delete
Rename
File_attribute
Open
Close
Write
Read
DIR_read
DISK_space
Link
Unlink
File_date

文件的保护

- 对文件的访问系统首先要检查访问权限
  - 仅允许执行 (E)。
  - 仅允许读 (R)。
  - 仅允许写 (W)
  - 仅允许在文件尾写 (A)
  - 仅允许对文件进行修改（U）
  - 允许改变文件的存取枚限（C）
  - 允许取消文件（D）
  - 权限可进行组合

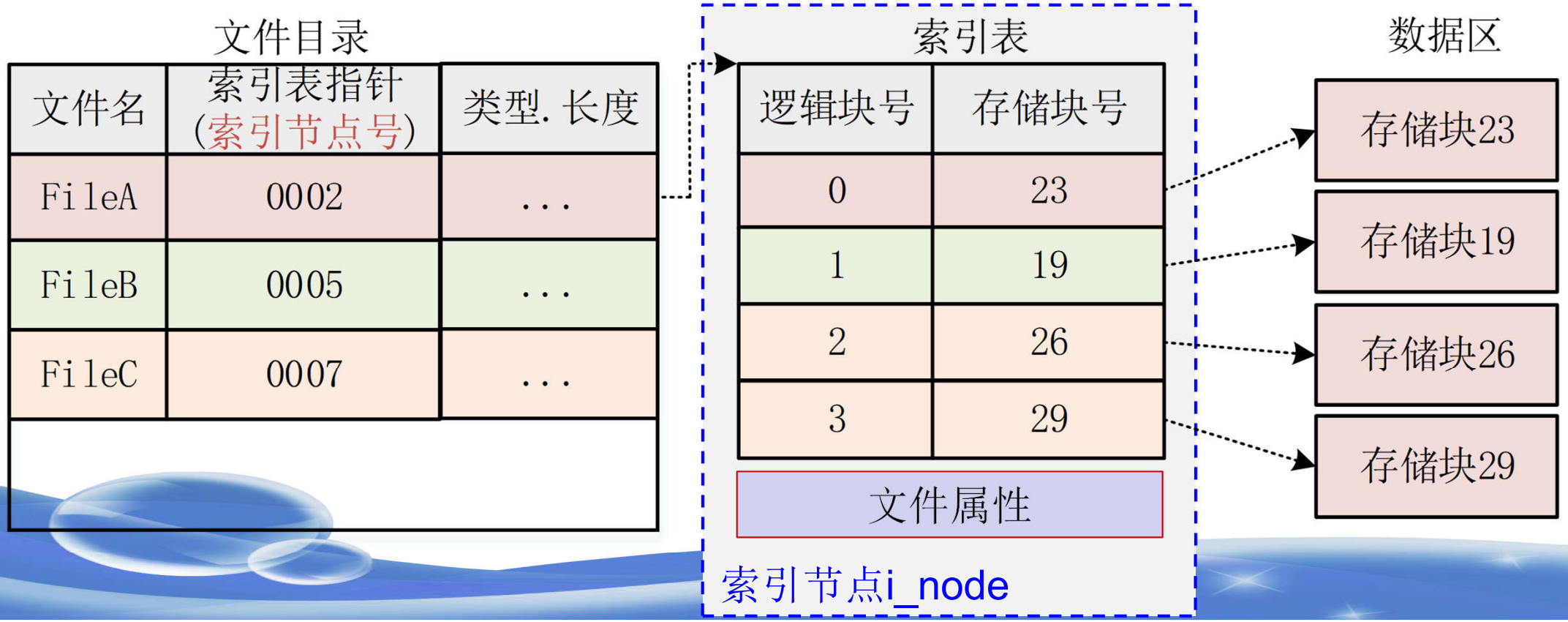
文件读写示例

```
1  //打开文件
2  FILE *pFile=fopen("MyTestFile.txt","rb");
3  char *pBuf;
4
5  fseek(pFile,0,SEEK END);
6
7  int len=ftell(pFile);
8  pBuf=new char[len];
9
10 rewind(pFile);// = fseek(pFile,0,SEEK_SET);
11
12 fread(pBuf,1,len,pFile);
13 fclose(pFile);
```

9.5 Linux索引文件/i_node[细节]

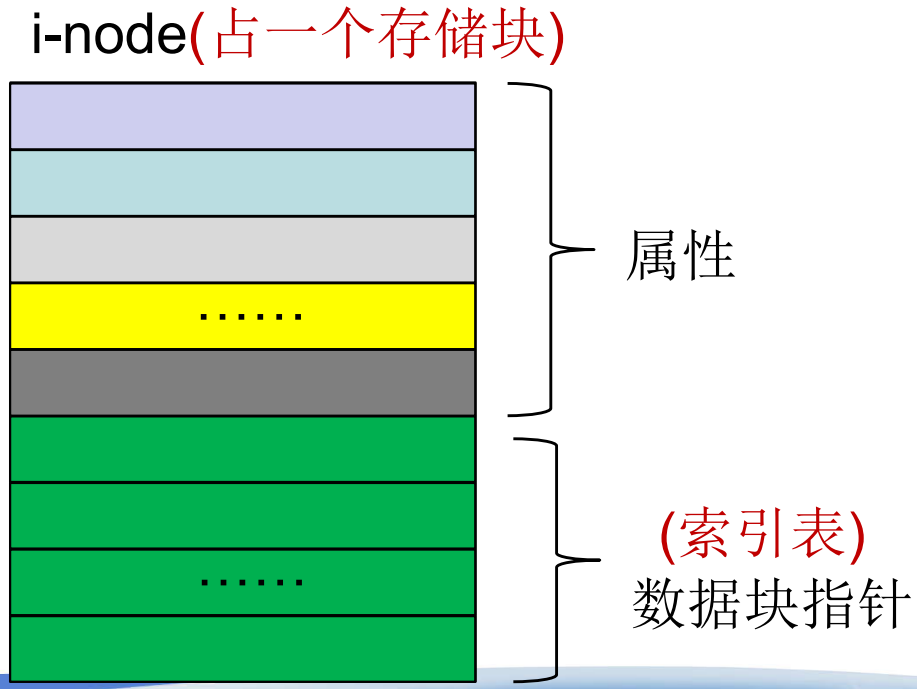
Linux索引文件ext2/i_node

- 索引文件
  - 索引文件 = 索引节点i_node + 数据区(若干存储块)
    - ◆索引节点i_node（占一个存储块）
      - □索引表
      - □属性：类型、权限、所有者、大小、时间戳等



Linux索引文件ext2/i_node

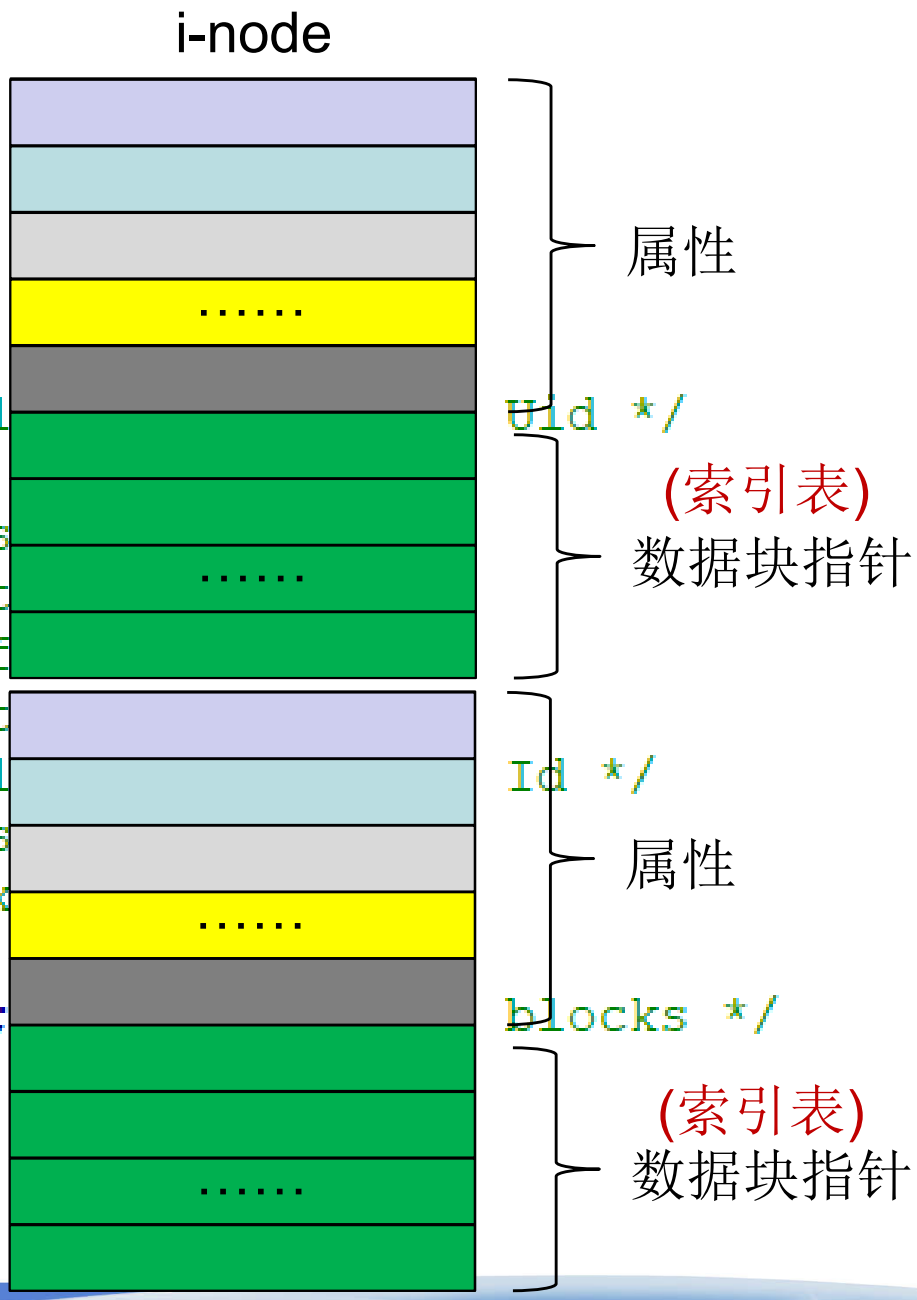
- 索引文件
  - 索引文件 = 索引节点i_node + 数据区(若干存储块)
    - 索引节点i_node（占一个存储块）
      - 索引表
      - 属性：类型、权限、所有者、大小、时间戳等



Linux索引文件ext2/i_node

- 索引节点i_node

```
1   struct ext2_inode    //ext2 = 32字节
2   {
3       __u16 i_mode;           /* File
4       __u16 i_uid;            /* Low 1
5       __u32 i_size;           /* Size
6       __u32 i_atime;          /* Acces
7       __u32 i_ctime;          /* Creat
8       __u32 i_mtime;          /* Modif
9       __u32 i_dtime;          /* Delet
10      __u16 i_gid;            /* Low 1
11      __u16 i_links_count;    /* Links
12      __u32 i_blocks;         /* Block
13      ......
14      __u32 i_block[EXT2_N_BLOCKS];
15      ......
16  };
```



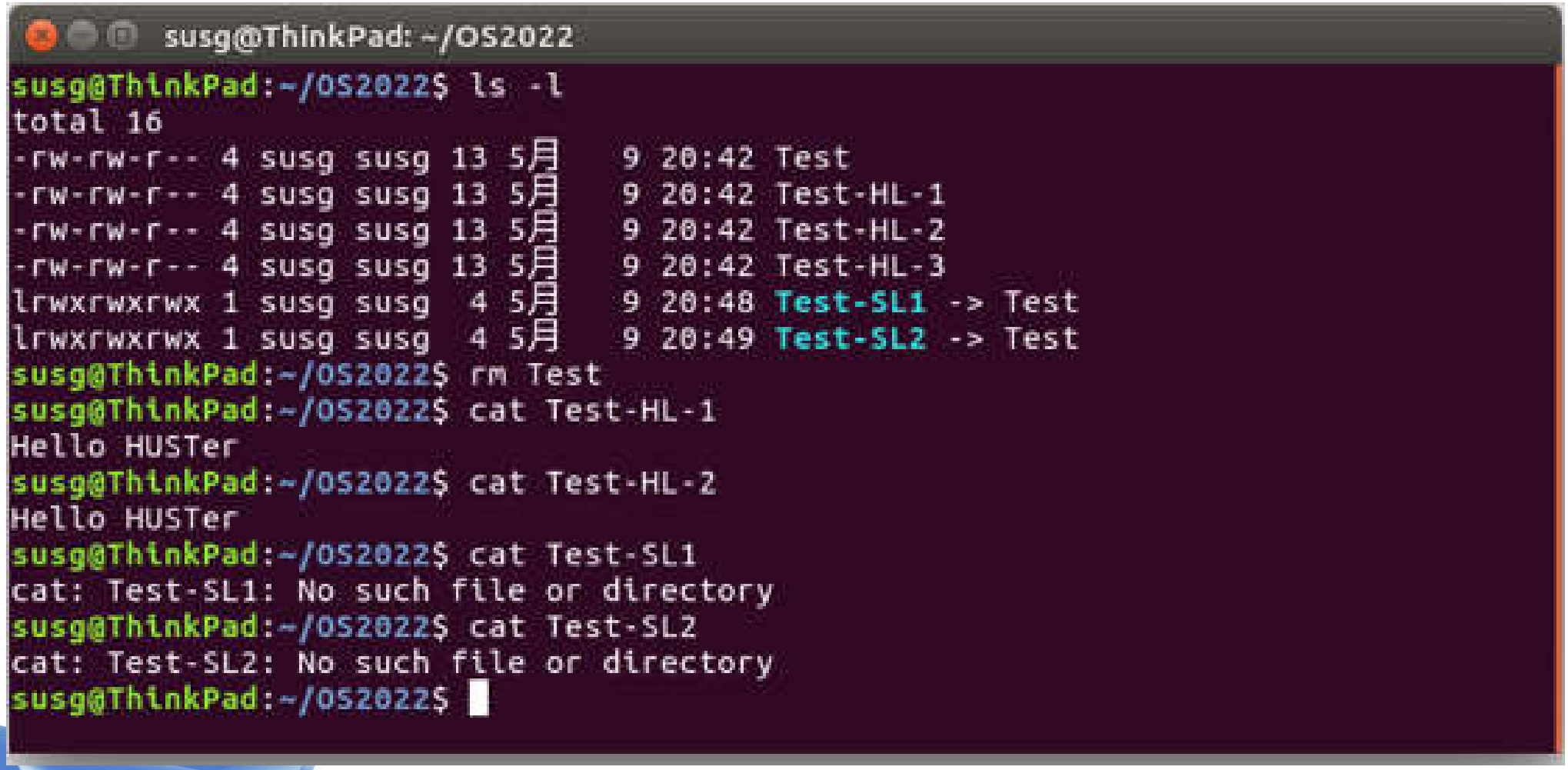
Linux索引文件ext2/i_node

- （1）i_mode字段
  - ■指定文件的类型和文件的访问权限

符号	类型	宏
-	Regular file/普通文件	S_ISREG(m)
d	Directory/目录	S_ISDIR(m)
c	Character Device/字符设备	S_ISCHR(m)
b	Block Device/块设备	S_ISBLK(m)
f	FIFO	S_ISFIFO(m)
s	Socket	S_ISSOCK(m)
l	Symbolic Link/符号链接	S_ISLNK(m)

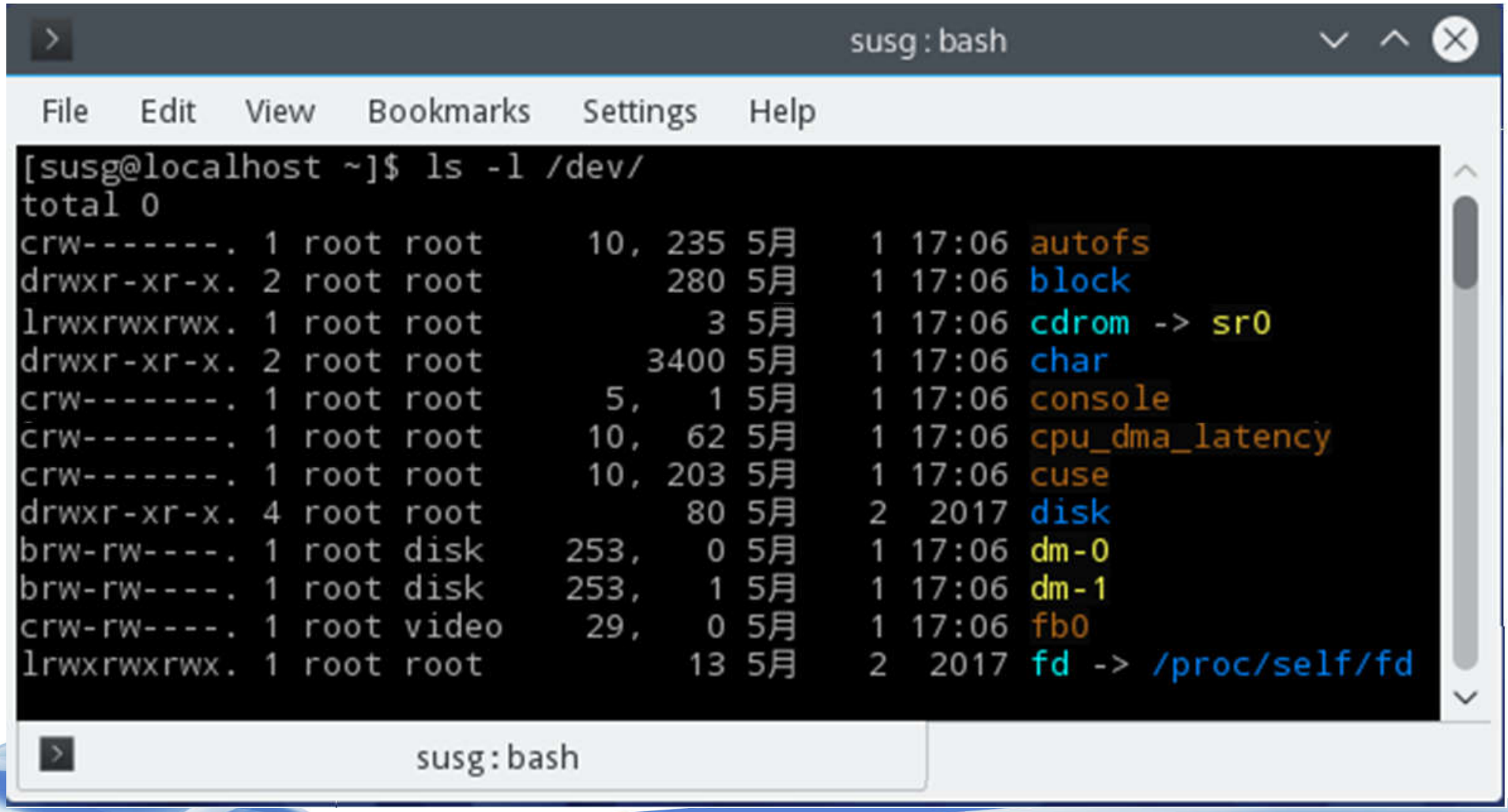
Linux索引文件ext2/i_node

- （1）i_mode字段
  - ■指定文件的类型和文件的访问权限



Linux索引文件ext2/i_node

- （1）i_mode字段
  - ■指定文件的类型和文件的访问权限



Linux索引文件ext2/i_node

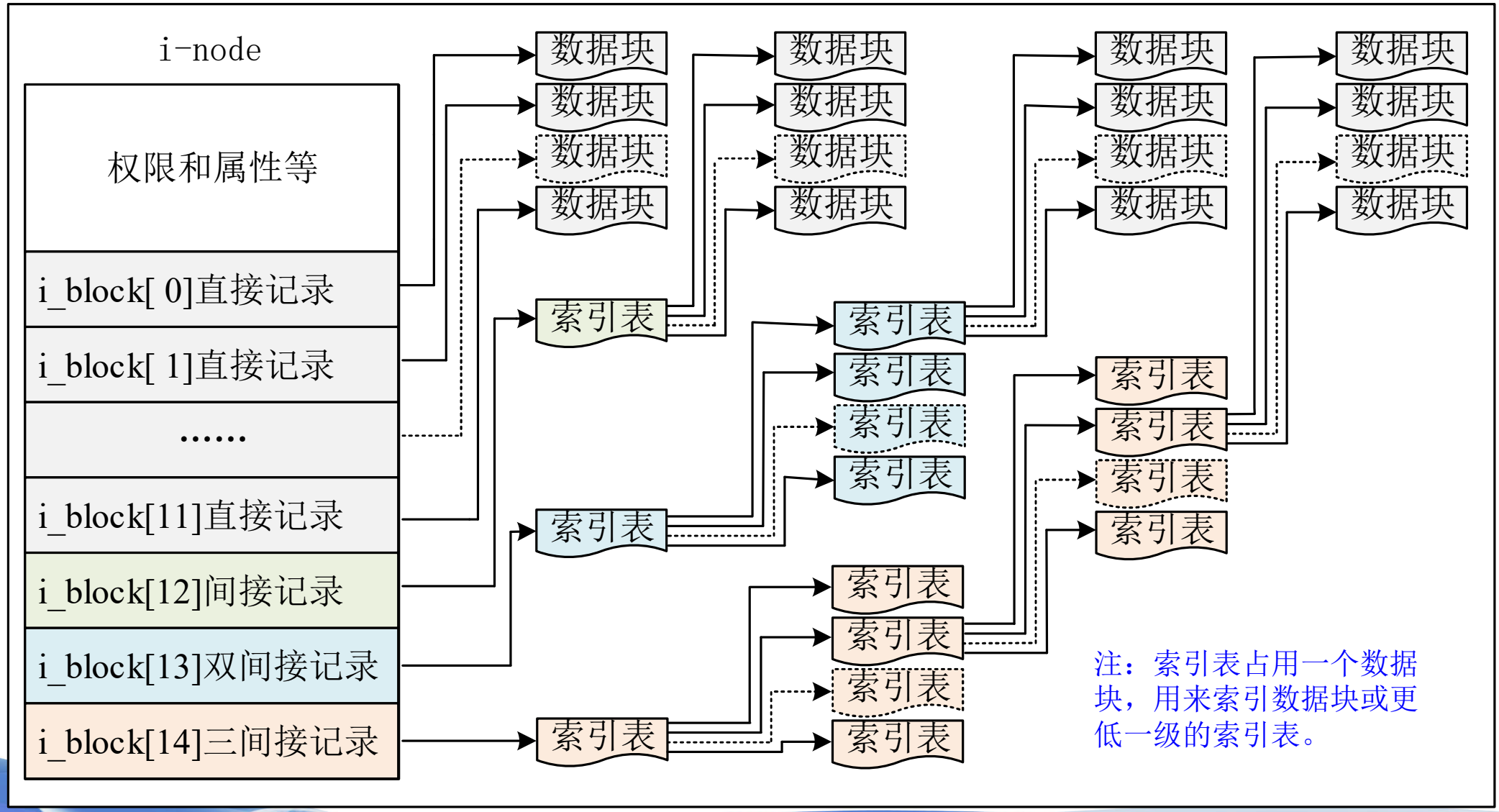
- （2）i_ atime、i_ctime、i_ mtime和i_ dtime
  - 访问、创建、修改和删除时间。
- （3）i_uid和i_gid
  - 文件所有者的用户ID和组ID
- （4）i_ size
  - 文件实际大小(单位字节)。
- （5）i_links_count
  - 硬链接的数量
- （6）i_blocks
  - 文件已使用数据块数量
- （7）i_block[EXT2_N_BLOCKS]
  - 数组，存储指向数据块的指针

索引表：EXT2_N_BLOCKS = 15 (默认)

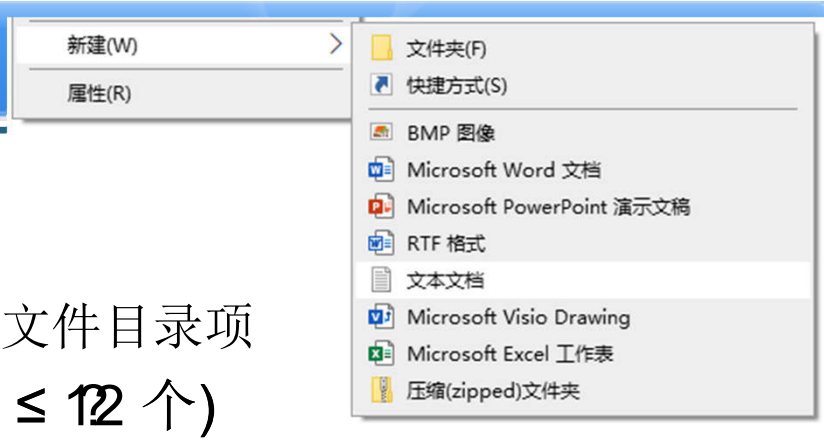
```
struct ext2_inode
{
    __u16 i_mode;          /* File
    __u16 i_uid;           /* Low 1
    __u32 i_size;          /* Size
    __u32 i_atime;         /* Acces
    __u32 i_ctime;         /* Creat
    __u32 i_mtime;         /* Modif
    __u32 i_dtime;         /* Delet
    __u16 i_gid;           /* Low 1
    __u16 i_links_count;   /* Links
    __u32 i_blocks;        /* Block
    ......
    __u32 i_block[EXT2_N_BLOCKS];
    ......
};
```

Linux索引文件ext2/i_node

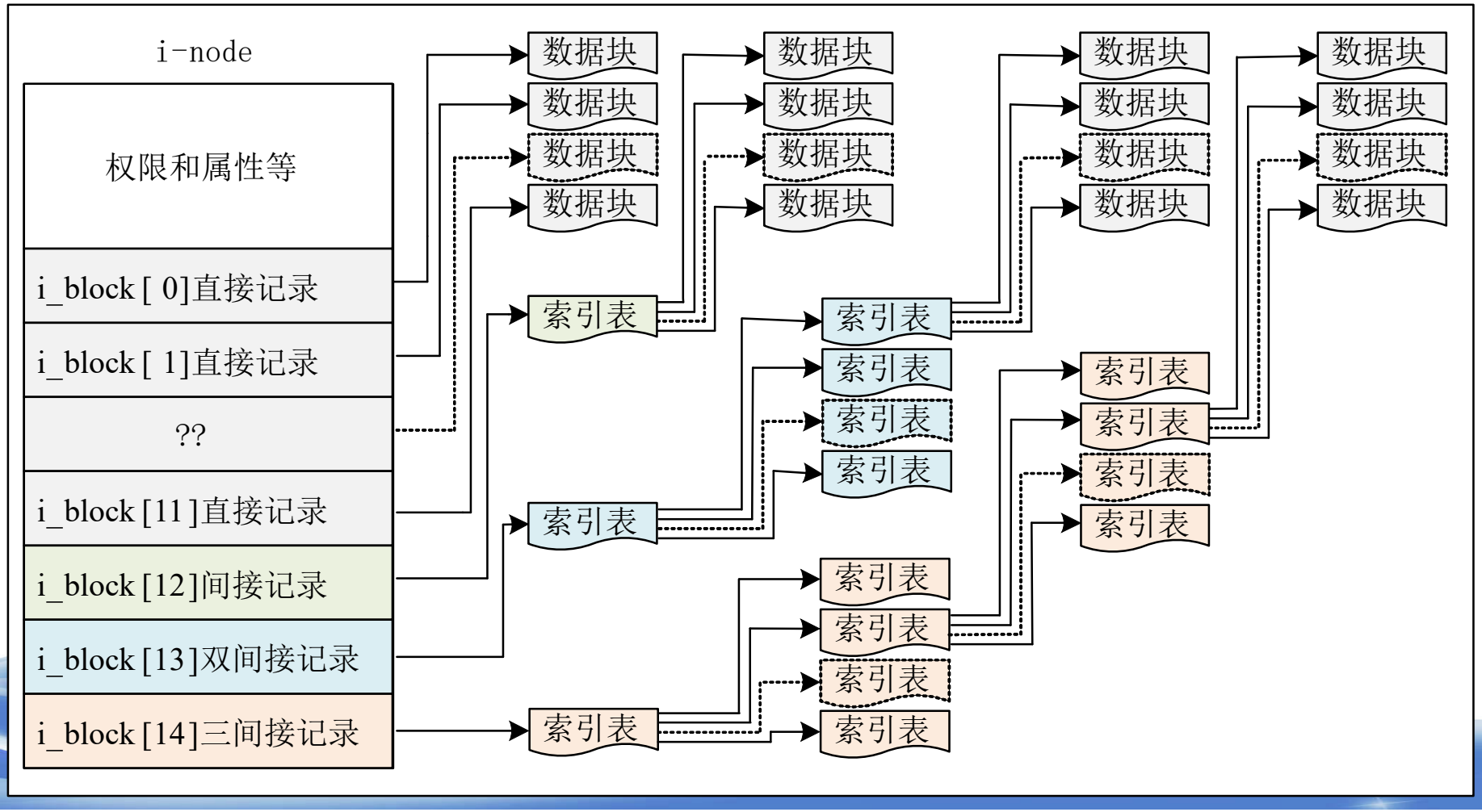
- （7）i_block[EXT2_N_BLOCKS] 【逻辑块⇔数据块】



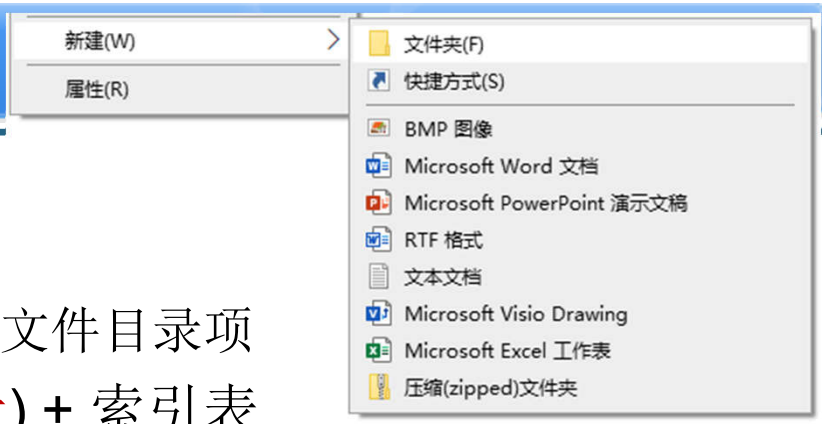
Linux索引文件ext2/i_node



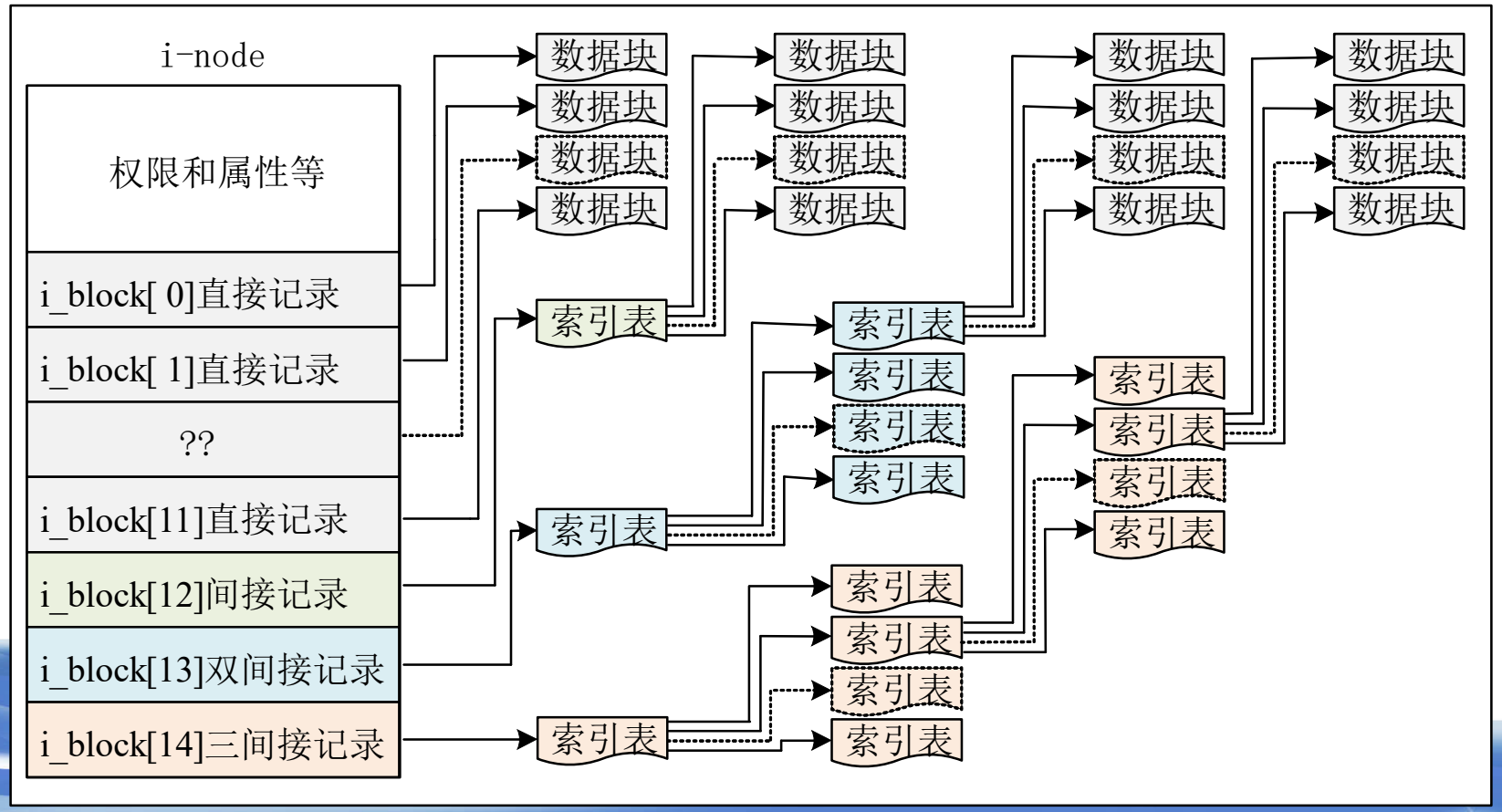
- 创建文件（**touch/create**）
  - 分配1个索引节点inode和1个存储块，增加1个文件目录项
    - ◆如果[未来]文件增大：分配更多的存储块（≤ 12 个）
      - ◘如果文件增加得更大：启用间接记录/双间接记录/三间接记录



Linux索引文件ext2/i_node



- 创建目录（**md**）/目录文件
  - 分配1个索引节点inode和1个存储块，增加1个文件目录项
    - ◆索引节点inode：文件属性(文件类型=目录) + 索引表
    - ◆存储块：文件目录项(文件名，索引节点号，...)
  - 如果[未来]目录中文件太多，文件目录项的数量更多，分配更多存储块



Linux索引文件ext2/i_node

- 示例(命令stat,md,echo)　　（课后验证）
  - ■创建目录（md）
  - ■创建文件（echo >）

Linux索引文件ext2/i_node

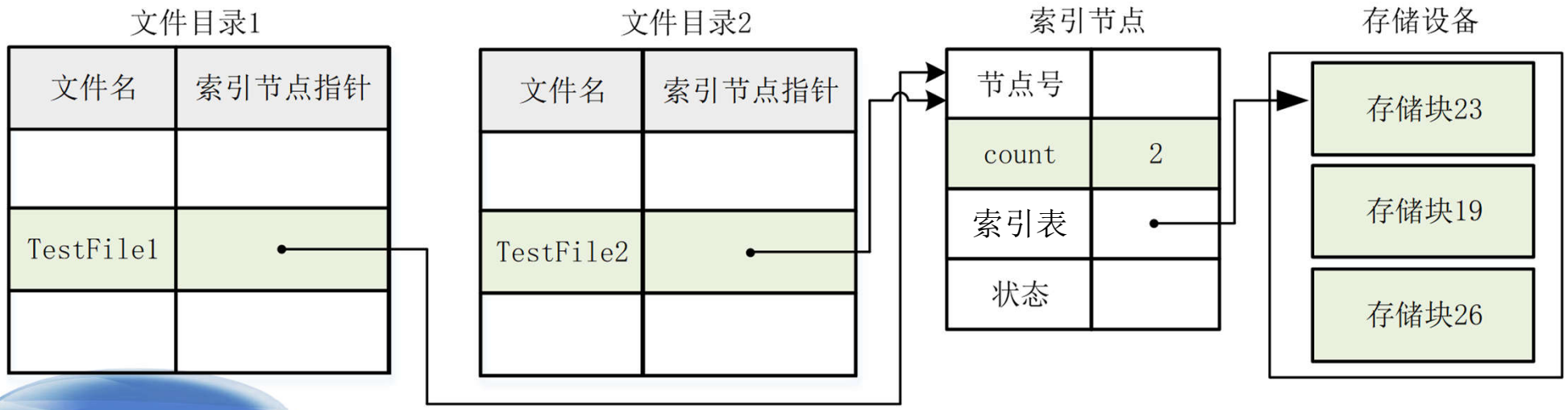
- 示例(命令stat,md,echo)　　（课后验证）
  - ■创建目录（md）
  - ■创建文件（echo >）

Linux索引文件ext2/i_node

- 示例(命令stat,md,echo)　　（课后验证）
  - ■创建目录（md）
  - ■创建文件（echo >）

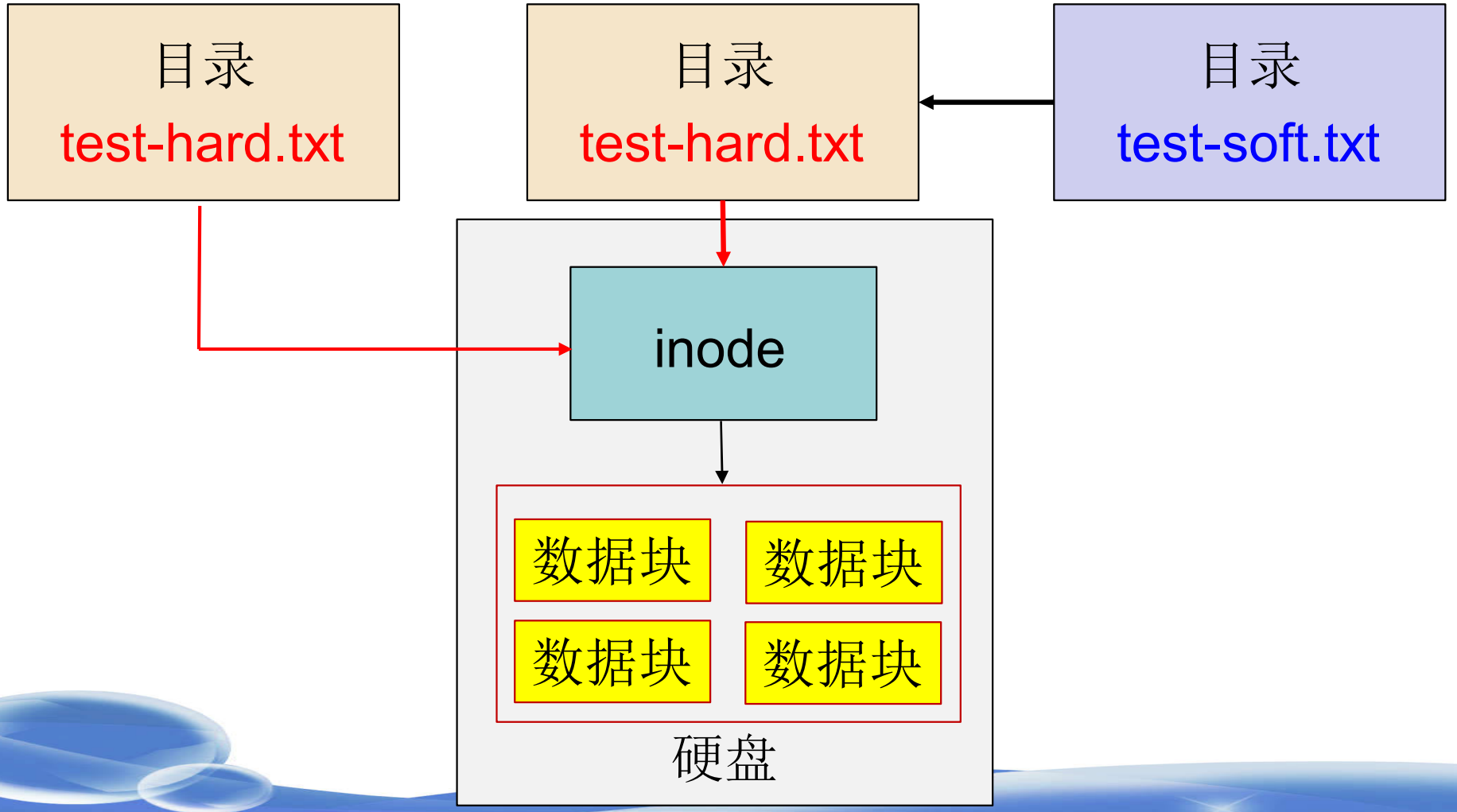
Linux索引文件ext2/i_node

- 硬链接 | 软链接
  - 硬链接
    - ◆硬链接是一个文件的入口路径（文件名）。
    - ◆例：TestFile1和TestFile1是同一文件的两个入口路径
  - 软链接
    - ◆软链接是指创建一个LINK类型的新文件，在新文件中只包含被共享文件的路径名以链接原文件。



Linux索引文件ext2/i_node

- 硬链接 | 软链接
  - 文件路径 | inode | 文件数据



Linux索引文件ext2/i_node

- 硬链接 | 软链接

（课后观察）

  - 硬链接创建命令
    - ln
  - 软链接创建命令
    - ln -s

```
> echo hello >> file
> ln file file-hardlink
> ln -s file file-softlink
```

Linux索引文件ext2/i_node

- 硬链接 | 软链接 （课后观察）
  - cat查看文件内容，得到的输出一样

```
> cat file
hello
> cat file-hardlink
hello
> cat file-softlink
hello
```

Linux索引文件ext2/i_node

- **ls –i 查看inode编号**

（课后观察）

  - 硬链接文件inode号与原文件相同
  - 软链接文件inode号与原文件不同

```
> ls -i file*
532670 file
532670 file-hardlink
532672 file-softlink
```

```
> cat file
hello
> cat file-hardlink
hello
> cat file-softlink
hello
```

- **ls –lj 查看文件详细信息**

```
> ls -l file*
-rw-rw-r-- 2 6 file
-rw-rw-r-- 2 6 file-hardlink
lrwxrwxrwx 1 4 file-softlink -> file
```

Linux索引文件ext2/i_node

- 删除(rm)原文件和之后...（课后观察）

```
> strace rm file
...
unlinkat(AF_FDCMD, "file", 0)
...
```

```
> cat file
hello
> cat file-hardlink
hello
> cat file-softlink
hello
```

■硬链接文件正常访问

■软链接文件访问失败

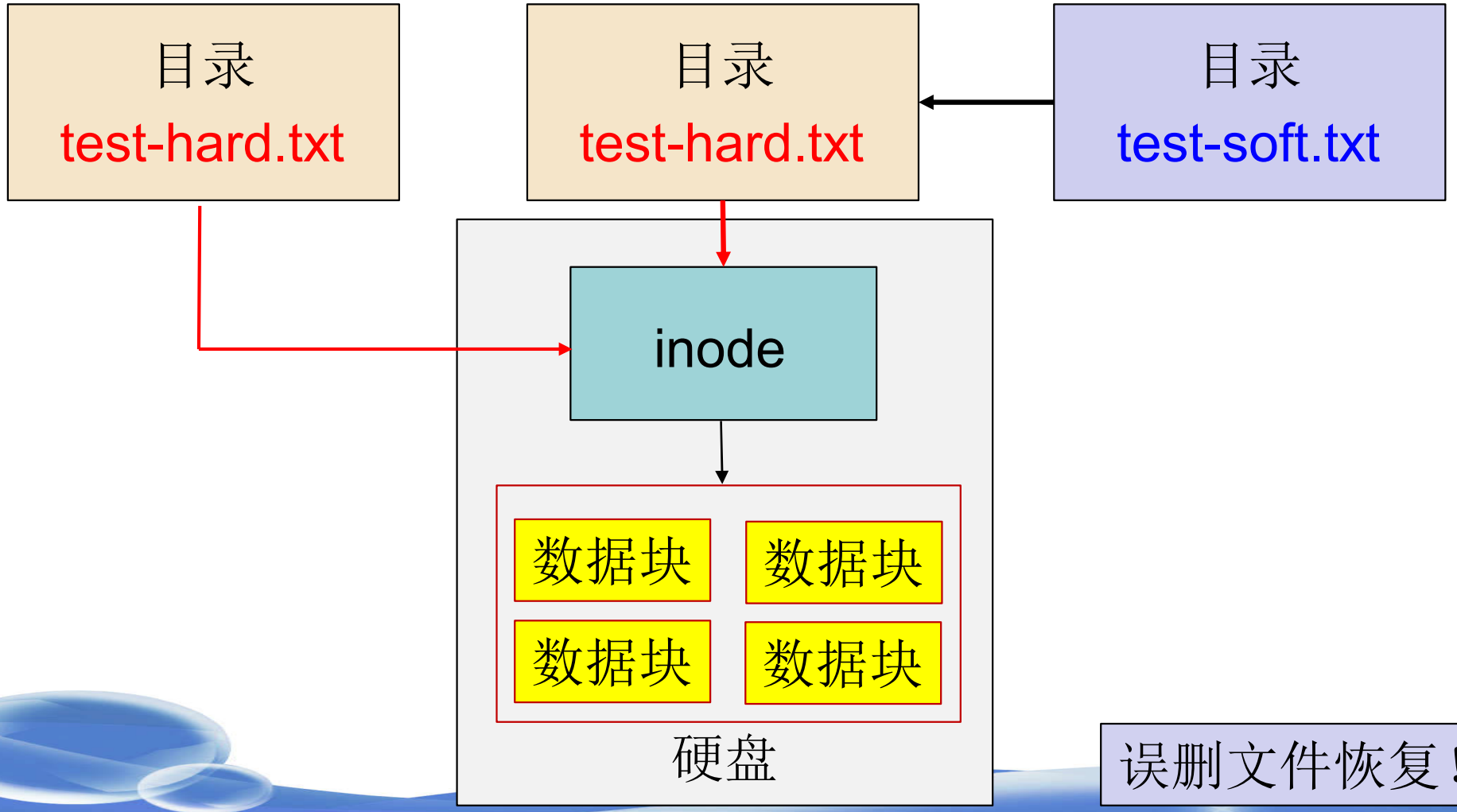
```
> cat file-hardlink
hello
> cat file-softlink
cat: file-softlink: No such file or directory
```

Linux索引文件ext2/i_node

- 硬链接 | 软链接

（课后观察）

  - 文件路径 | inode | 文件数据



误删文件恢复！？

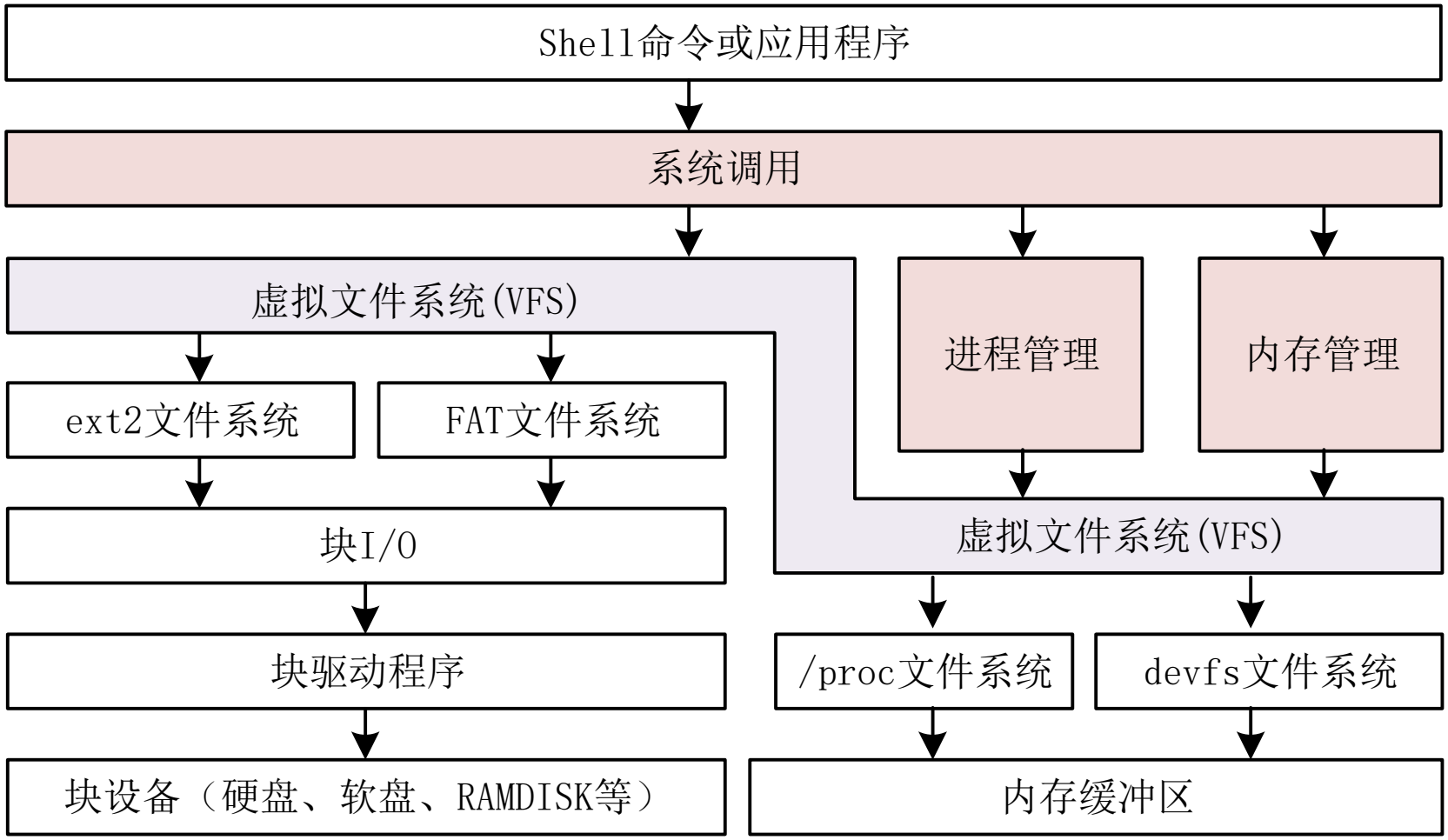
9.4 虚拟文件系统VFS

Linux虚拟文件系统

- **虚拟文件系统（Virtual File System，VFS）**
  - VFS是覆盖在逻辑文件系统之上面向操作系统的一个接口层，它对每个逻辑文件系统的实现细节进行抽象，使得不同的文件系统在Linux核心以及其他进程看来，都是相同的。

Linux虚拟文件系统

- 虚拟文件系统（Virtual File System，VFS）



Linux虚拟文件系统

- **VFS重要的数据结构**
  - super_block
  - file_system_type
  - file
  - inode
  - super_operations
  - file_operations
  - inode_operations