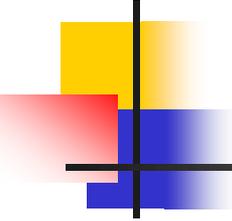


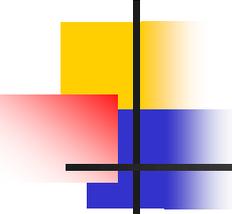
# 第6章 文件管理

外存是计算机保存数据的场所，而文件又是数据存储的主要形式。文件管理其实就是对外存空间的管理，本章主要讨论文件管理中的相关概念和主要管理策略。



# 教学要求

- ◆掌握文件、文件系统、文件目录的概念。了解文件系统的功能。
- ◆掌握文件的逻辑组织和物理组织的概念，以及相应的组织形式。
- ◆掌握目录的基本组织方式，了解文件存储空间的管理。
- ◆理解路径名和文件链接的概念。
- ◆掌握实现文件共享的办法。



# 教学内容

---

6.1 文件与文件系统

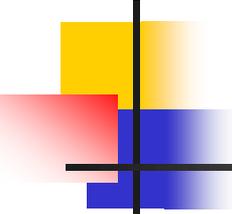
6.2 文件结构与存储设备

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

6.6 文件系统实例分析

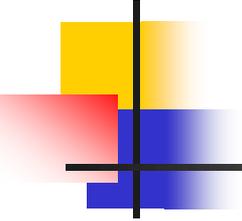


# 6.1 文件和文件系统

## 6.1.1 文件的概念

### 1、文件的定义

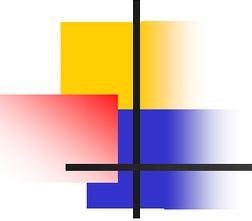
文件是指可保存的、具有标识名的一组相关信息的集合。



## 组成文件的信息单位：

---

- **字节**：是一种无结构的文件，或称流式文件。如：UNIX，MS-DOS中的文件形式。
- **记录**：记录由一组相关**数据项**组成。如：每个学生情况可视为一个记录，它由姓名，出生年月，性别，籍贯等数据项组成。所有学生登记表组成一个学生文件。

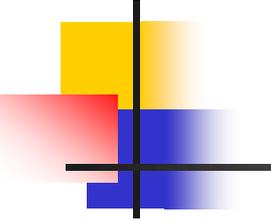


## 6.1.1 文件的概念（续）

---

### 2、文件的属性

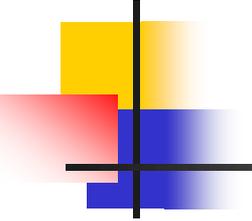
指文件的类型、大小、创建时间、操作特性和存取保护等信息。文件的属性一般存放在文件的目录项中。



## 6.1.2 文件的类型

### 1、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
- 库文件：各种标准函数，标准过程和实用程序等
- 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

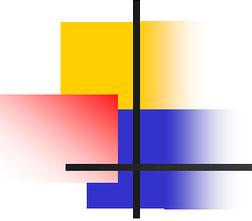


## 6.1.2 文件的类型（续）

---

### 2、按文件的存取控制属性

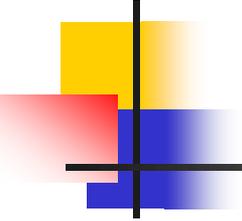
- 只读文件：只允许进行读操作
- 读写文件：允许进行读写操作
- 执行文件：允许执行



## 6.1.2 文件的类型（续）

### 3、按文件的组织形式分类

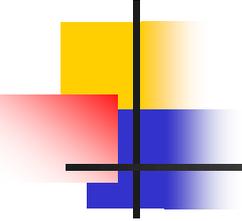
- 普通文件：指一般的用户文件和系统文件。
- 目录文件：指由文件目录项组成的文件。
- 特别文件：设备作为文件来统一管理和使用，把设备称为特别文件。



## 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

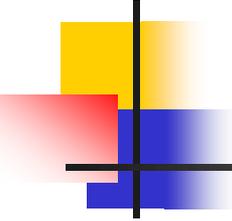


## 6.1.4 文件系统

操作系统中负责管理文件的软件机构称为文件系统。

文件系统的功能：

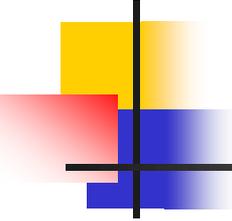
- 文件的按名存取
- 文件目录建立和维护
- 文件存储空间的分配和管理
- 文件的共享、保护和保密
- 文件操作



## 6.2 文件结构与存储设备

研究文件结构有两种观点：

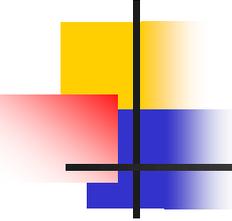
- ◆ 用户的观点：文件的逻辑结构（逻辑文件），可将文件看作字节的集合，或者记录的集合。
- ◆ 系统的观点：文件的物理结构（物理文件），主要研究驻留在存储介质上的文件是如何存放的。



## 6.2.1 文件的逻辑结构

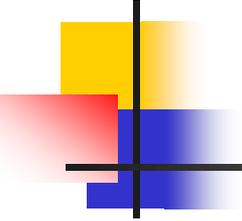
分为流式文件和记录式文件两种：

- **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。
- **优点：**节省存储空间，无需额外的说明和控制信息。



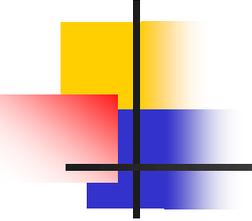
## 6.2.1 文件的逻辑结构（续）

- **记录式文件**:是一种有结构文件。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。
- ◆ **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。
- ◆ **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。



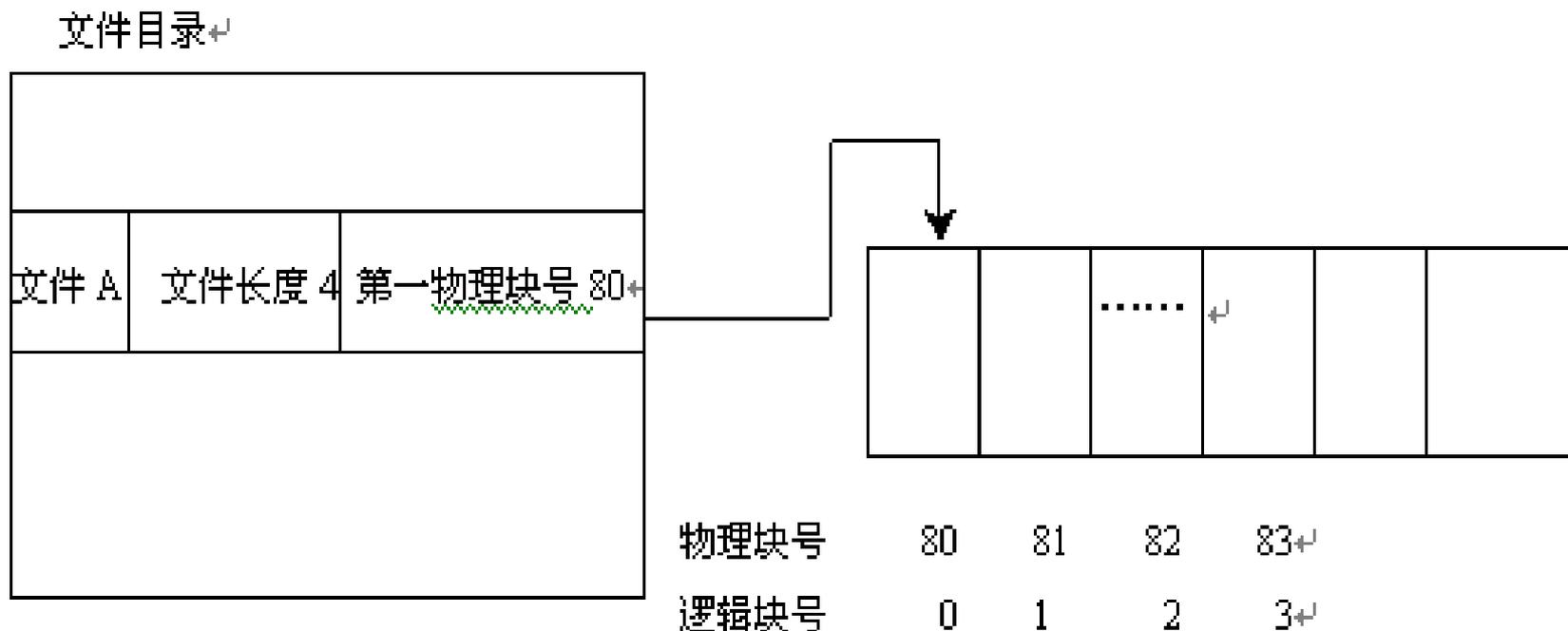
## 6.2.3 文件的物理结构

文件的物理结构有：连续结构、链接结构、索引结构和混合索引结构四种。

**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，即逻辑上连续的文件信息在物理上也连续。

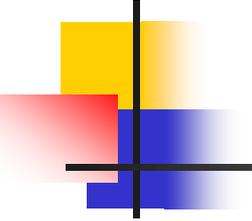
如存放在**磁带**上的文件，而存放在磁盘上的文件则可采用连续结构，也可采用别的结构。

# 连续结构



**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；



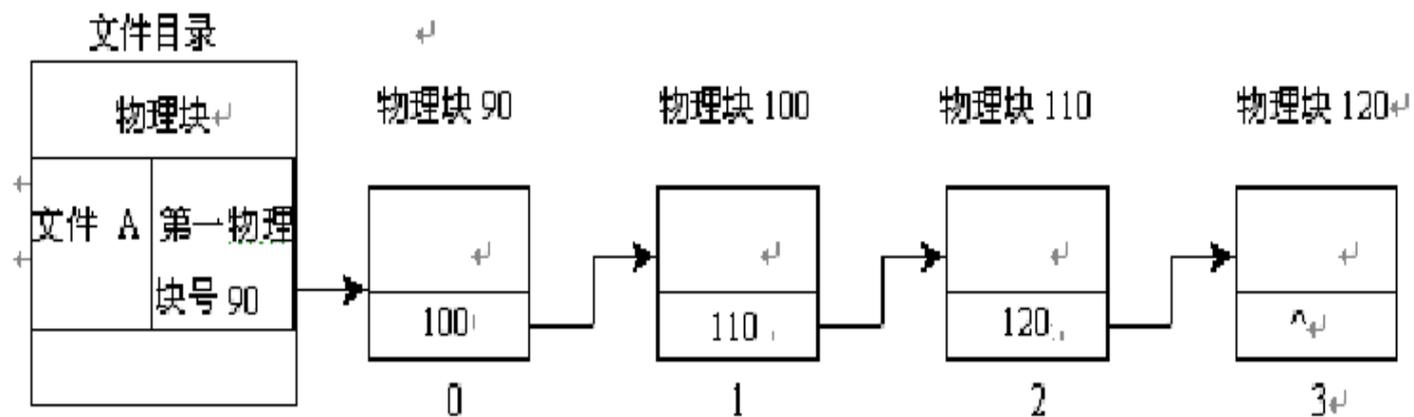
## 2. 链接结构

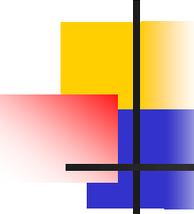
---

这是一种非连续的结构，存放文件信息的物理块不要求连续。又可分为**隐式链接**和**显式链接**两种：

- **隐式链接**：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。

# 隐式链接





## 隐式链接（续）

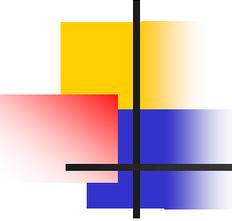
---

### 优点：

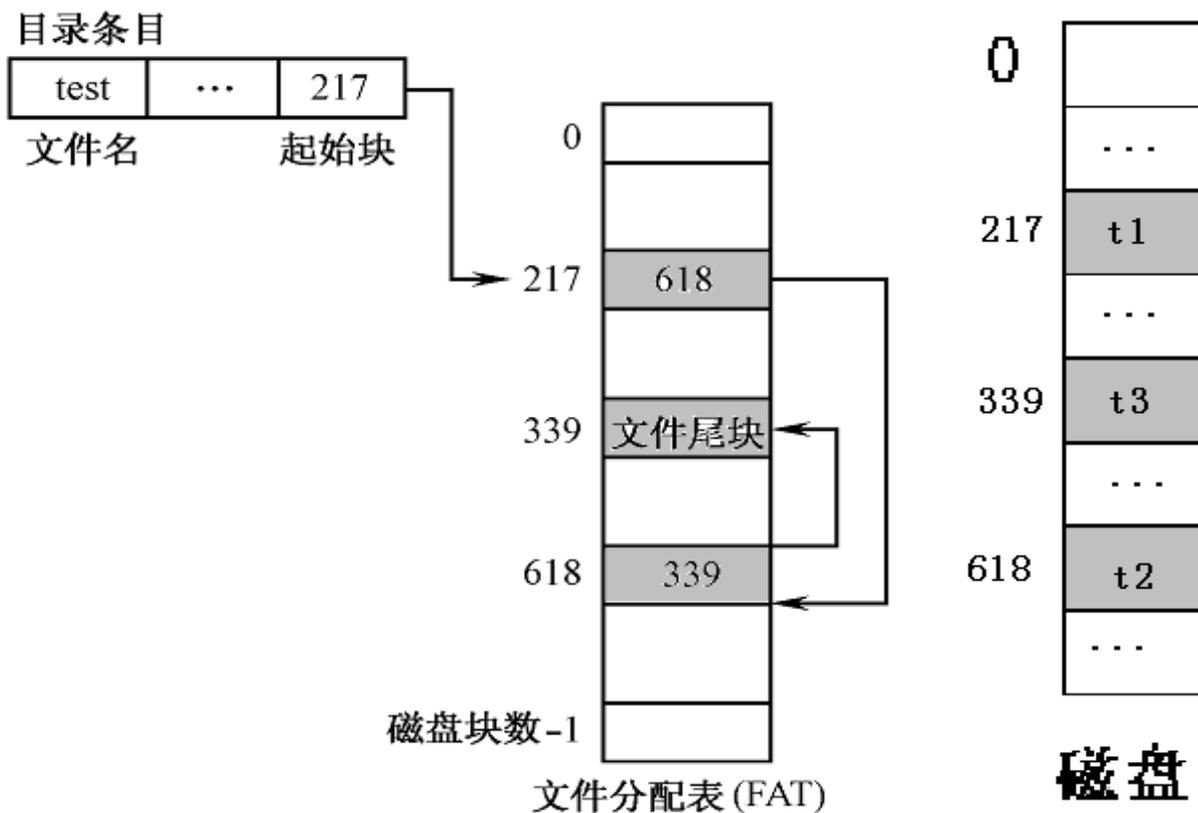
- 提高了磁盘空间利用率,不存在外部碎片问题
- 有利于文件插入和删除
- 有利于文件动态扩充

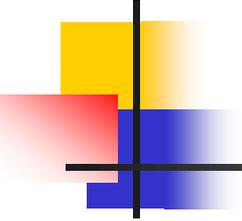
### 缺点：

- 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

- 
- 
- **显式链接**将盘块中的链接字按盘块号的顺序集中起来，构成盘文件映射表/文件分配表（FAT）。利用FAT可方便地进行随机存取。

# 显式链接

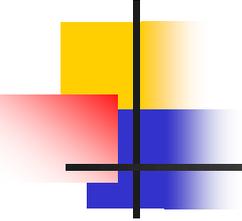




## FAT文件系统特性

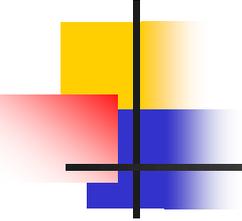
---

FAT也要占用一定的存储空间，若盘的容量较大，也可能占用较多的存储空间。在进行文件访问时，可能在内存中装不下整个FAT，这样就会造成若要读某块文件信息时，还要读盘块映射表的操作，影响使用效率。



在MS-DOS和Windows系统中，文件的物理结构使用的是FAT结构。通常将磁盘空间划分为块，每块大小为扇区的整数倍。在FAT文件系统中块称为簇。

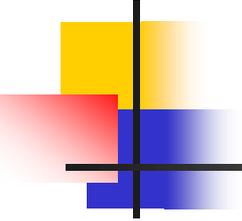
一个磁盘分区能分为多少簇则FAT就有多少表项



## 【思考题】

---

- 1、什么叫**FAT16**、**FAT32**？
- 2、在**FAT16**中一簇最大**64**个扇区，为什么**FAT16**能管理的磁盘分区为**2G**？
- 3、**FAT32**同**FAT16**相比有什么优点？
- 4、对于**FAT16**文件系统，若一个磁盘分区的大小为**512M**，问一个簇最少要为多少个扇区？

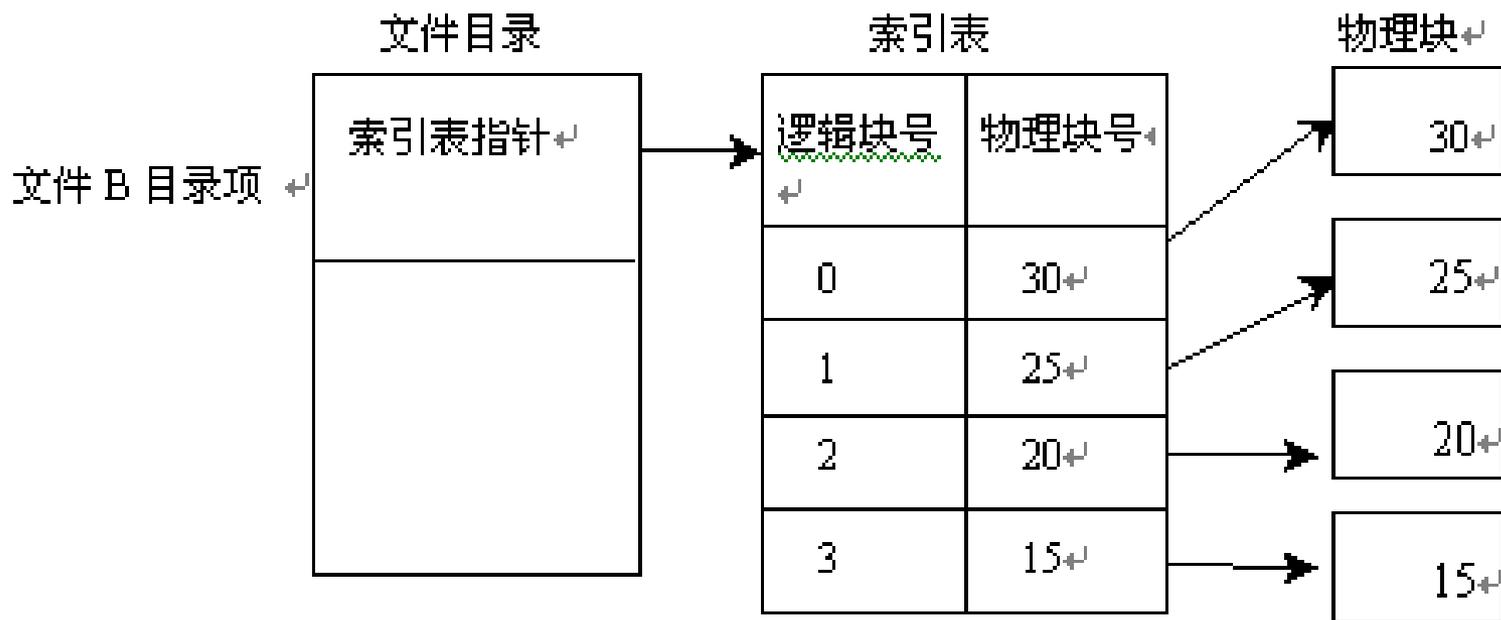


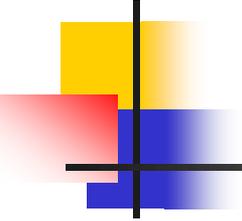
### 3. 索引结构

---

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。

# 索引结构（续）



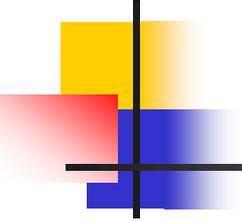


## 索引结构（续）

---

### 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；



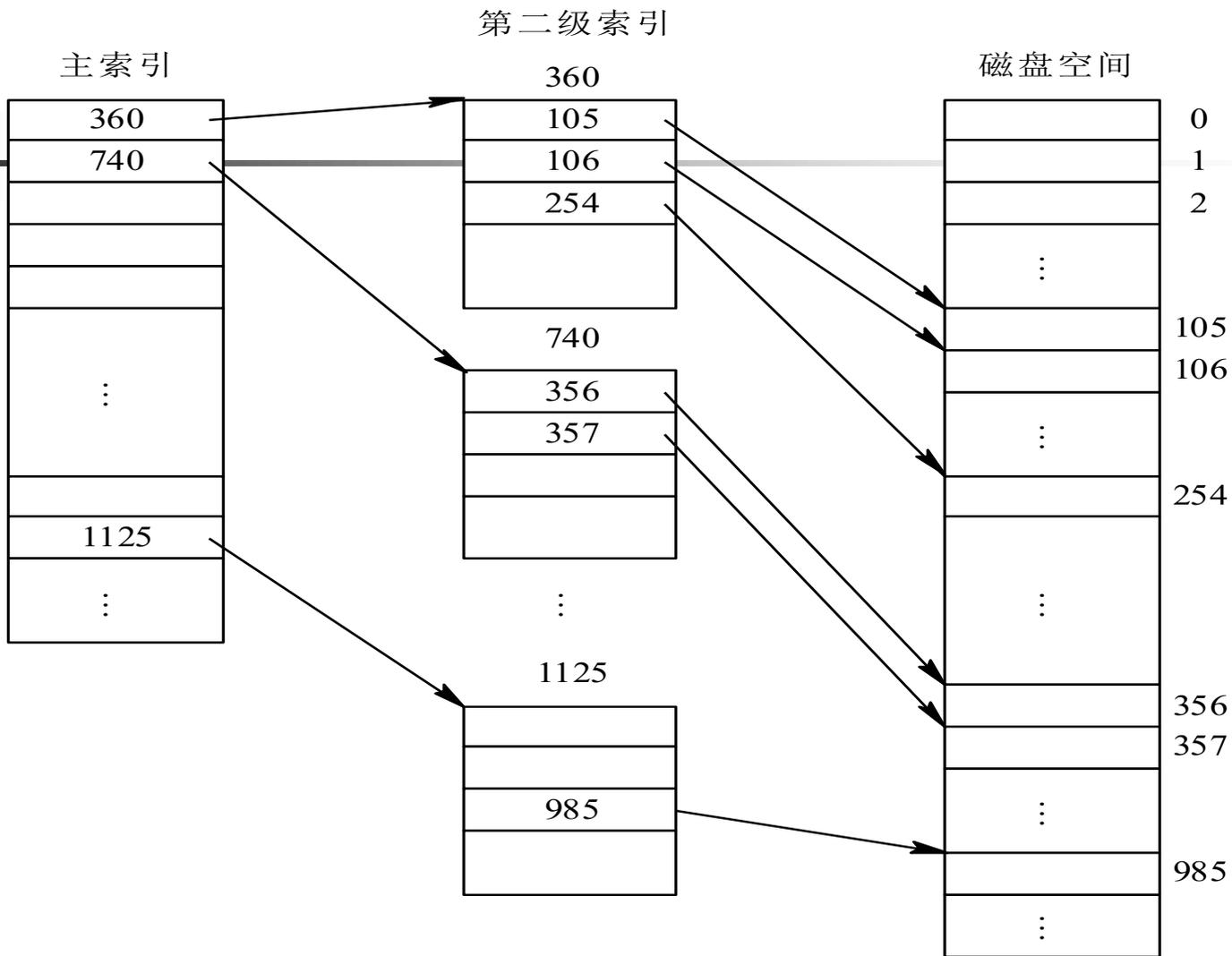
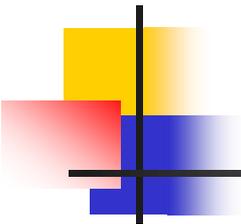
## 索引结构（续）

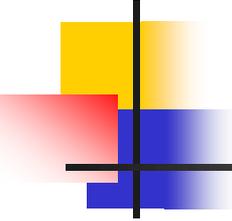
---

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





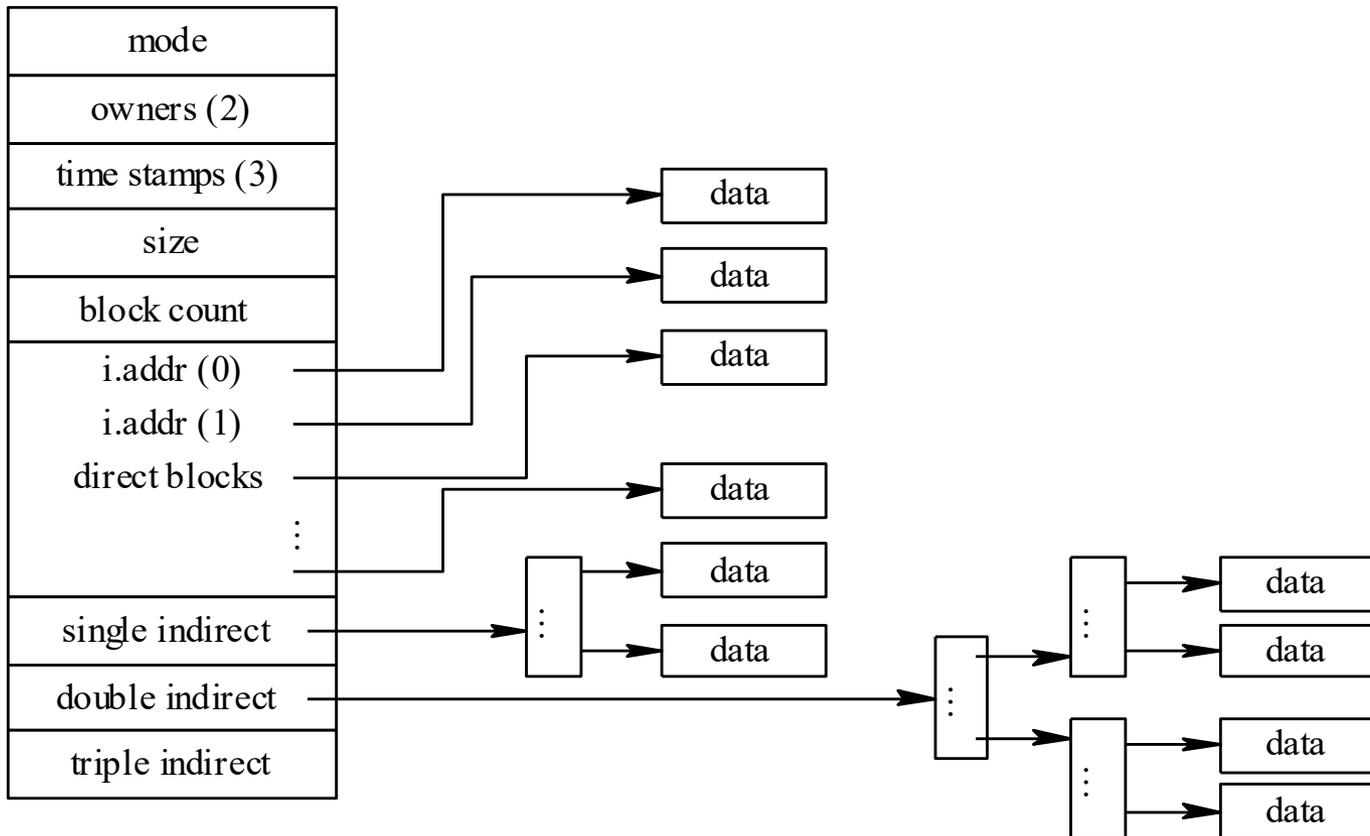
## 4. 混合索引结构

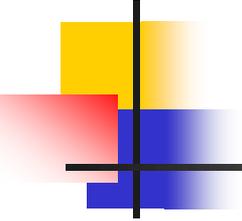
---

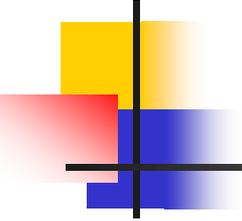
在**UNIX/Linux**系统中采用，考虑到文件长度的实际情况，可以把索引表的头几项设计成直接块寻址方式，即这几项所指的物理块中存放的是文件内容，而索引表的后几项设计成多重索引方式，即间接块寻址方式。

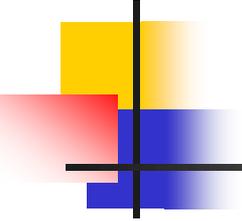
在**UNIX**系统中，在索引节点中有**13**个地址项（即为混合索引表）来存放文件的直接或间接块号。

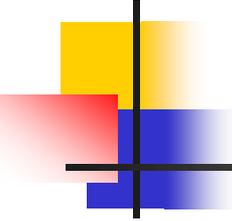
# 混合索引结构（续）



- 
- 
- **直接地址**：在索引节点中可设置10个直接地址项，即用*iaddr(0)~iaddr(9)*来存放直接地址。即这里的每项中所存放的是该文件数据的盘块的盘块号。

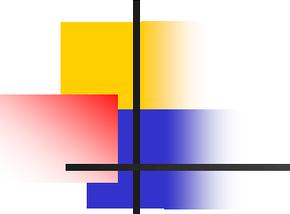
- 
- **一次间接地址**：对于大、中型文件，只采用直接地址是不现实的。为此，可再利用索引节点中的地址项*iaddr(10)*来提供一次间接地址。其实质就是一级索引分配方式。

- 
- **多次间接地址**：当文件长度很大时，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。同理，地址项*iaddr(12)*作为三次间接地址。



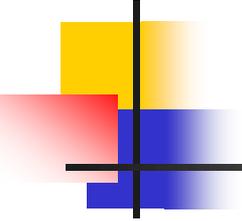
【例】存放在某个磁盘上的文件系统，采用混合索引分配方式，其 **FCB**中共有**13**个地址项，第**0~9**个地址项为直接地址，第**10**个地址项为一次间接地址，第**11**个地址项为二次间接地址，第**12**个地址项为三次间接地址。如果每个盘块的大小为**512**字节，若盘块号需要**3**个字节来描述。问：

- (1)该文件系统允许文件的最大长度是多少？
- (2)将文件的字节偏移量**4000**、**14000**、**140000**转换为物理块号和块内偏移量。
- (3)假设某个文件的**FCB**已在内存，但其他信息均在外存，为了访问该文件中的某个位置的内容，最少需要几次磁盘访问，最多又需要几次？



## 6.2.4 文件的存储设备

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。



## 6.3 文件目录管理

---

对文件目录管理的要求如下：

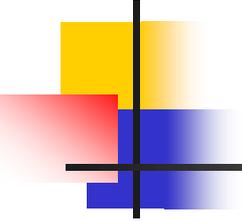
- (1)实现“按名存取”。
- (2)提高对文件（含目录）的检索速度。
- (3)文件共享。
- (4)允许文件重名。

## 6.3.1 文件控制块和文件目

### 1、文件控制块

文件控制块（**DOS**、**Windows**中也称目录项），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。文件控制块包括：

- 基本信息
- 存取控制信息
- 文件使用信息

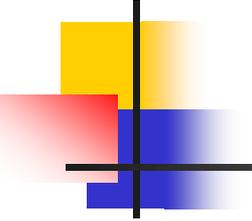


## 文件目录管理（续）

---

文件名	扩展名	属性	备用	时间	日期	第一块号	盘块数
-----	-----	----	----	----	----	------	-----

### MS-DOS中的目录项



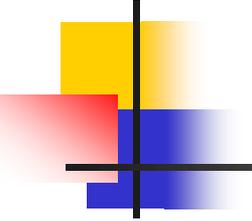
## 2、文件目录

---

文件目录是文件控制块（目录项）的有序集合。

文件目录作用：

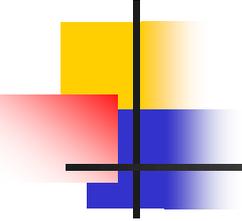
- 可实现文件的“按名存取”。当用户要求存取某个文件时，系统首先查找文件目录，通过比较文件名就可找到所查找文件的目录项。然后，通过目录项指出的文件存储的起始位置等就能依次存取文件信息。
- 能实现文件共享。



## 6.3.2 索引节点

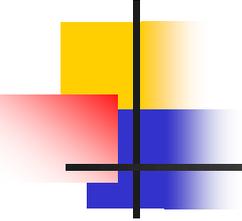
### 1、索引节点的引入

目录项中的其它的描述信息在目录检索时，根本没有必要装入内存。**UNIX/Linux**系统中，采用了**文件名**和**文件描述信息**分开存储的方法，使文件描述信息单独形成一个数据结构，该数据结构被称为索引节点(index node)，简称*i*节点。



## UNIX/Linux的文件目录

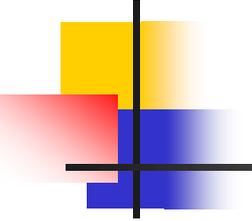
文件名	索引节点编号
文件名1	10
文件名2	11



## 2、磁盘索引节点

---

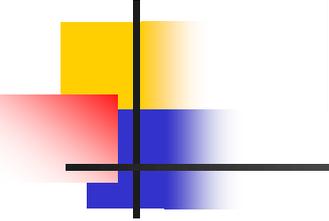
- (1) 文件主标识符
- (2) 文件类型
- (3) 文件存取权限
- (4) 文件物理地址
- (5) 文件长度
- (6) 文件连接计数
- (7) 文件存取时间



### 3、内存索引节点

---

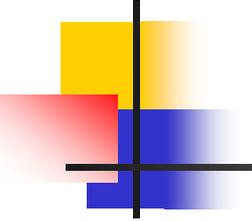
- (1) 索引节点编号：用于标识内存索引节点。
- (2) 状态：指示i节点是否上锁或被修改。
- (3) 访问计数：每当有一进程要访问此i节点时，将该访问计数加1，访问完再减1。
- (4) 文件所属文件系统的逻辑设备号。
- (5) 链接指针：设置有分别指向空闲链表和散列队列的指针。



## 6.3.3 目录结构

### 1. 单级目录

文件名	物理地址	文件说明	状态位
文件名1			
文件名2			



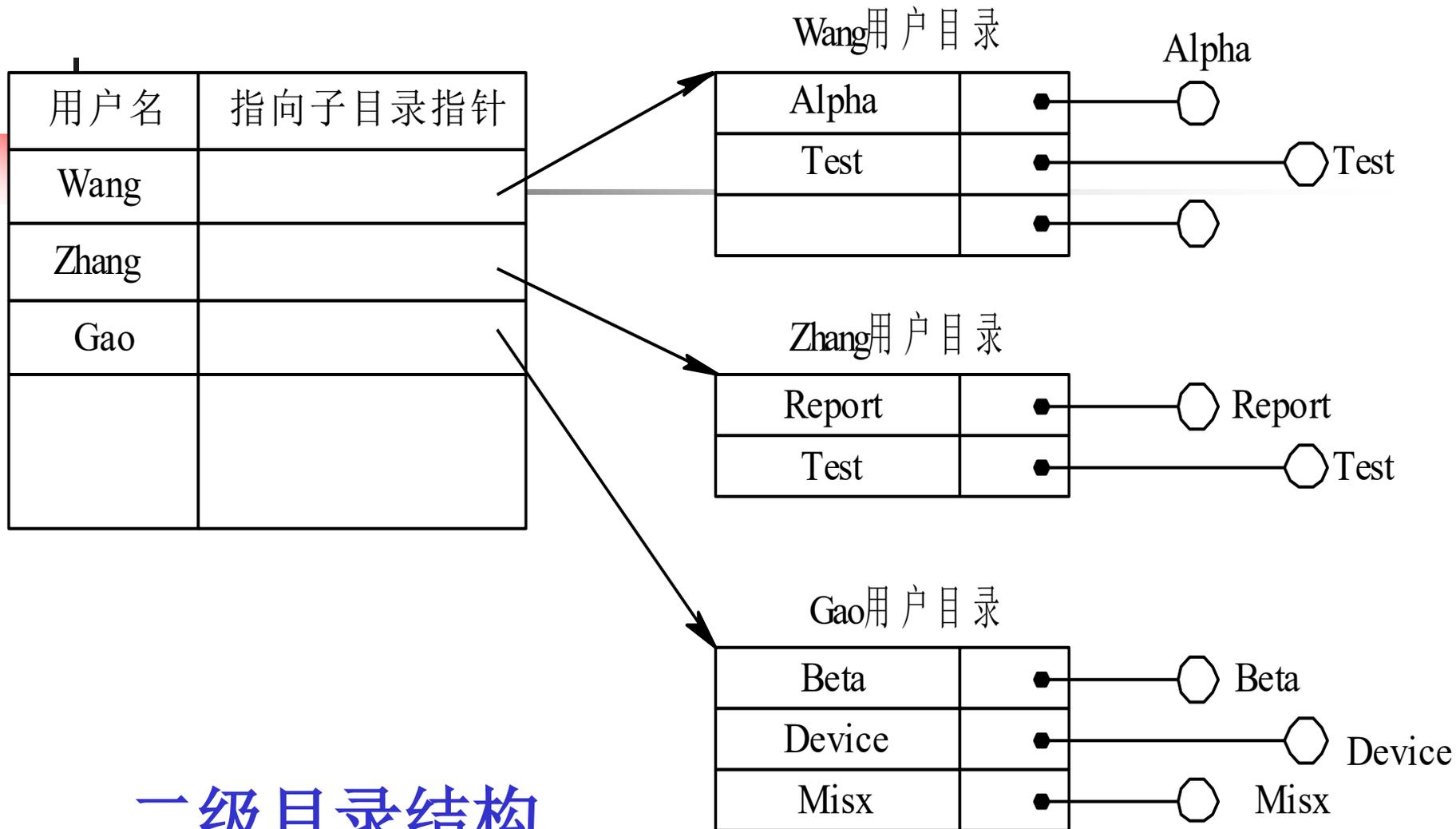
## 2. 二级目录

---

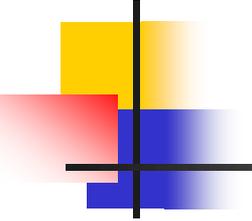
- 二级目录结构把目录分成主目录和用户文件目录两级。
- 主目录由用户名和用户文件目录首地址组成，用户文件目录中登记相应的用户文件的目录项。

**优点：**较为简单，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。



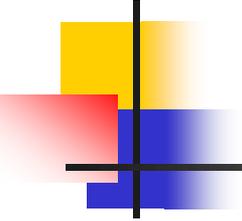
## 二级目录结构



### 3. 多级目录结构

---

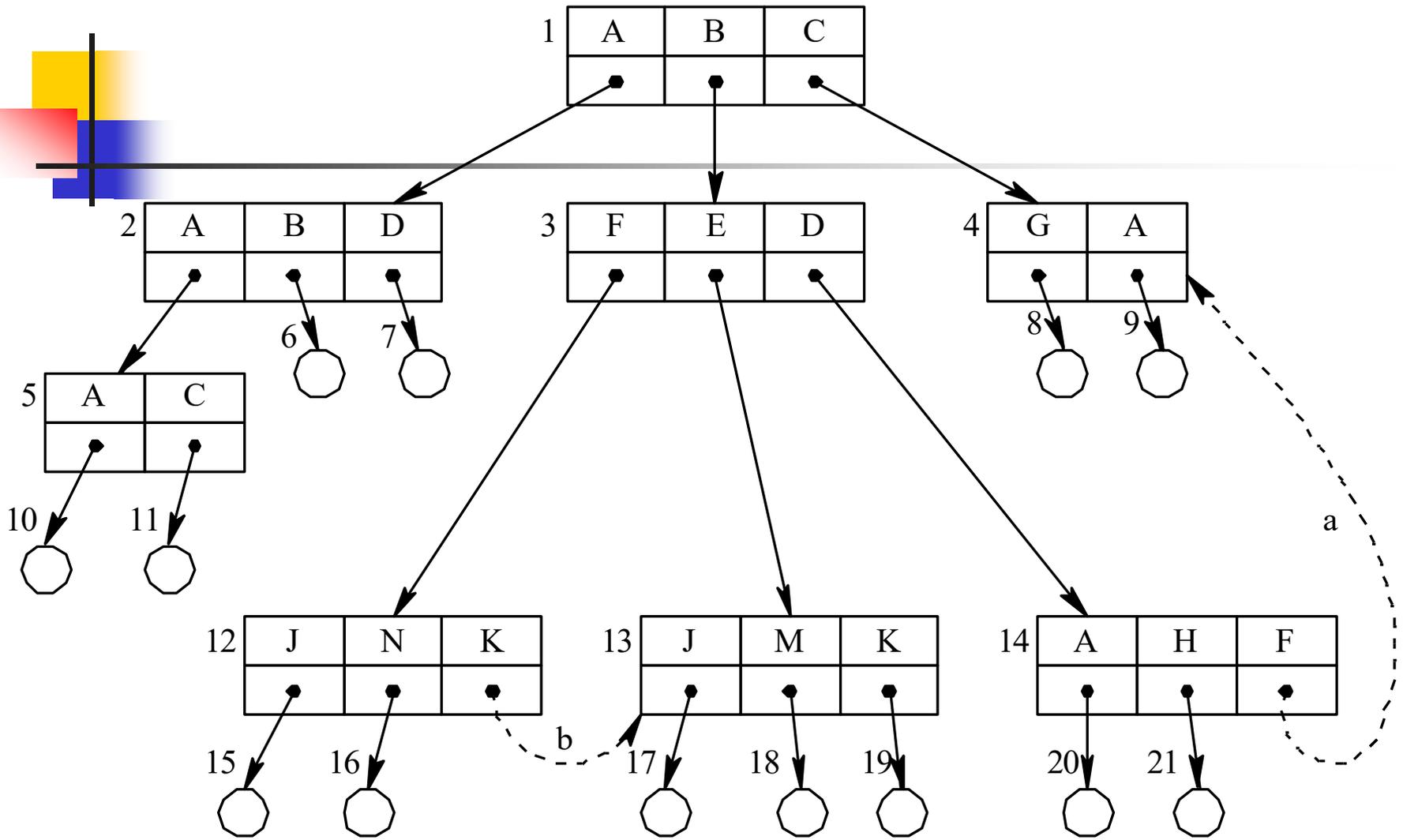
- 也称**树形目录结构**
- 多级目录结构由**根目录**和各级**子目录**组成，为管理上的方便，除根目录外，其它各级目录均以文件的形式组成目录文件。
- 早期的文件系统中，根目录中的目录项个数是有限的，而子目录中的目录项个数可以任意多个，仅受磁盘空间的限制。



# 路径名（绝对路径名，相对路径名）

---

- 在多级目录结构中一个文件的唯一标识不再是文件名，而是路径名。路径指的是访问一个文件所经过的路线。
- **绝对路径名**：从根结点开始，由经过的目录名和文件名以及分隔符组成，如：  
C:\WINNT\A.TXT, /d2/d5/f3, /f7
- **相对路径名**：从当前目录开始，由所经过的目录名和文件名以及分隔符组成。如：../d1/f1

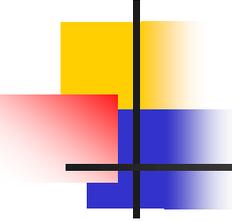


## 6.4 文件存储空间管理

### 6.4.1 空闲块表

也称为空闲文件目录

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



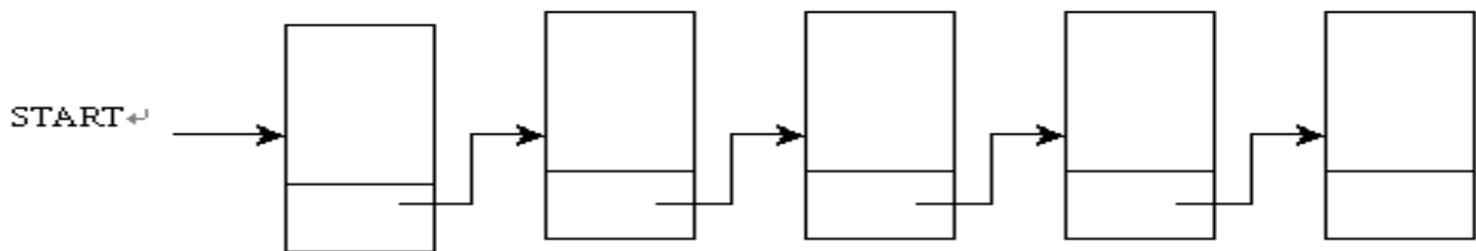
## 6.4.1 空闲块表（续）

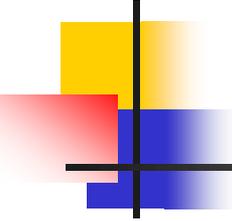
---

- 空闲盘块的分配与内存的动态分配类似，同样可以用首次、最佳、最坏适应法。盘块的回收也同内存的回收方式类似。

## 6.4.2 空闲块链

- 空闲块链是一种把所有空闲块链接在一起的结构。
- 分配：当系统建立文件需分配空闲块时，从链中摘取所需的空闲块，然后调整链首指针。
- 回收：反之，当回收空闲块时，把释放的空闲块逐个插入链首。





## 空闲块链（续）

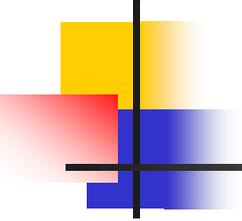
- 这种方法只需在系统中保留一个链首指针，令其指向第一个空闲块。
- 优点：简单，但工作效率较低，因为每次在链上增加和移出空闲块时，需要做I/O操作。

例如，把一空闲块插入链时，要把链首指针（原指向第一个空闲块）写该空闲块中，然后让链首指针指向该空闲块。从链中摘取空闲块时也要读取下一个空闲块的指针。

## 6.4.3 位示图

系统为磁盘建立一张位图，在位图中每个物理块占1位，按物理块的顺序排列。“1”表示对应的物理块已占用，“0”表示空闲。

	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																



# 1. 盘块的分配

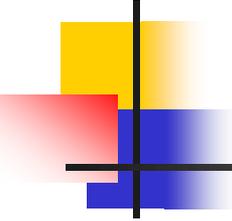
---

(1) 顺序扫描位示图，从中找出一个或一组其值为“0”的二进制位(“0”表示空闲时)。

(2) 将所找到的一个或一组二进制位，转换成与之相应的盘块号。假定找到的其值为“0”的二进制位，位于位示图的第*i*行、第*j*列，则其相应的盘块号应按下式计算：

$$b=n(i-1)+j$$

(3) 修改位示图，令  $\text{map}[i,j] = 1$ 。



## 2. 盘块的回收

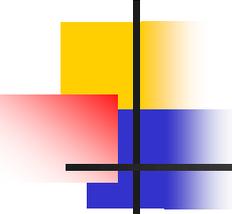
---

(1) 将回收盘块的盘块号转换成位示图中的行号和列号。转换公式为：

$$i=(b-1)DIV n+1$$

$$j=(b-1)MOD n+1$$

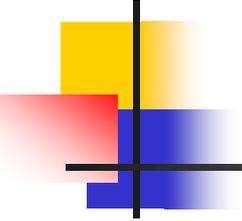
(2) 修改位示图。令  $map [i,j] = 0$ 。



## 6.4.4 成组链接法

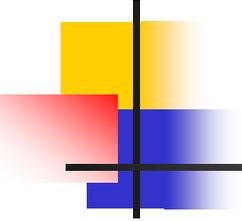
通常在UNIX/Linux系统中采用

实现方法是：将若干个空闲块归为一组，将每组中的所有空闲块号在其前一组的第一个空闲块号指示的磁盘块中，而将第一组中的所有空闲块号放入文件系统的超级块中的空闲块号表中。



## 6.5 文件的共享

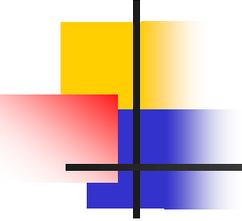
- 1、符号链接法
- 2、基于索引节点的共享方式
- 3、公共目录法
- 4、授权法



## 6.6 文件系统实例分析

### 6.6.1 Windows 2000/XP文件系统

- 支持传统的FAT12、FAT16 、 FAT32文件系统
- 支持只读光盘CDFS、通用磁盘格式UDF、高性能HPFS等文件系统。
- 支持一个全新的文件系统NTFS (New Technology File System)

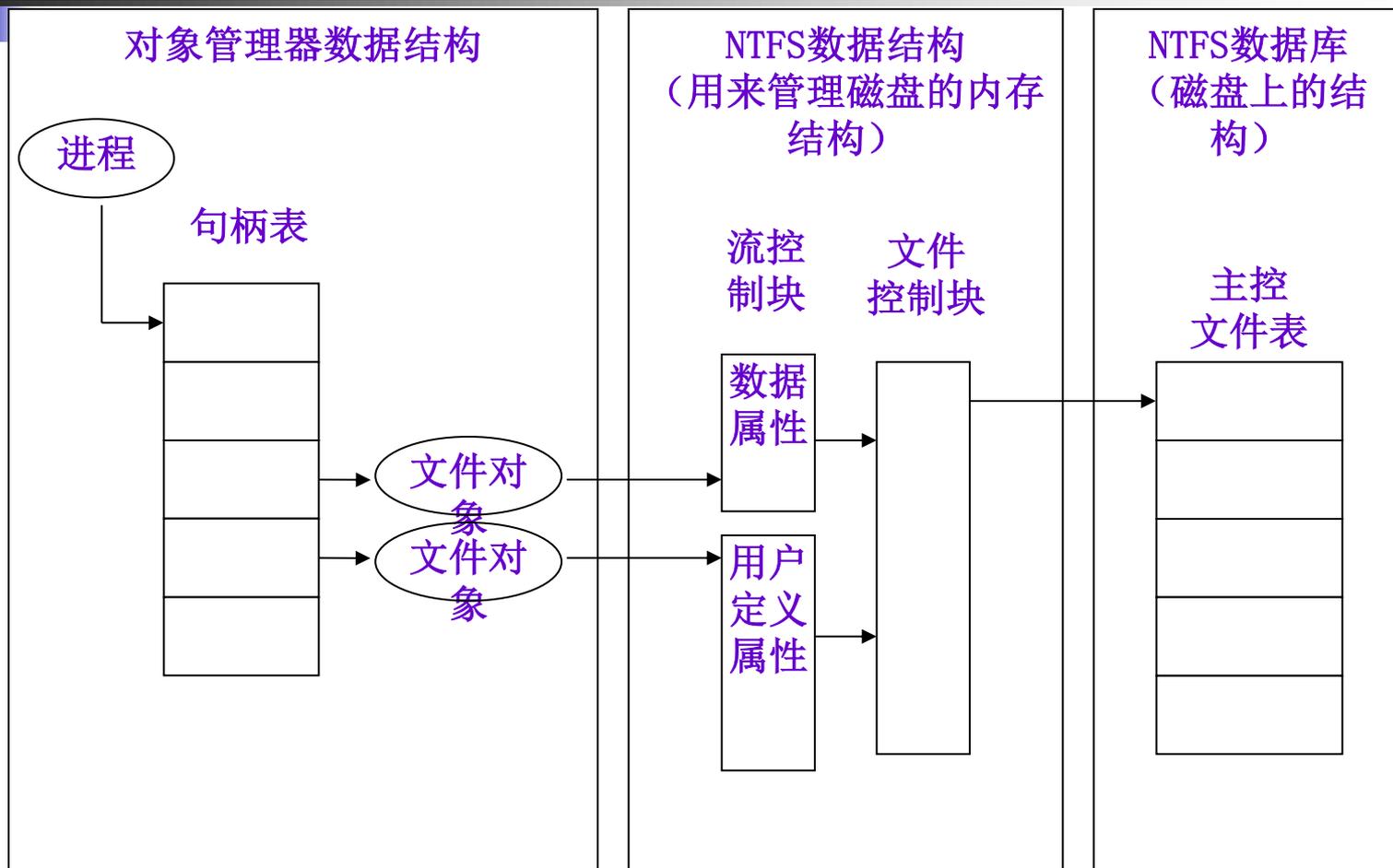


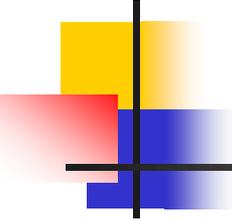
# 1、NTFS具有的特性

---

- 可恢复性：日志记录(logging)
- 安全性：
- 文件加密：
- 数据冗余和容错：
- 大磁盘和大文件：
- 数据流：
- 基于Unicode的文件名

## 2、NTFS数据结构

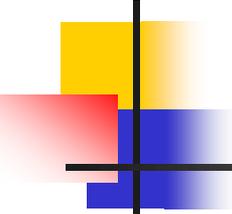




## 3、NTFS安全性支持

---

- NTFS卷上的每个文件和目录在创建时创建人就被指定为拥有者，拥有者控制文件和目录的权限设置，并能赋予其他用户访问权限。

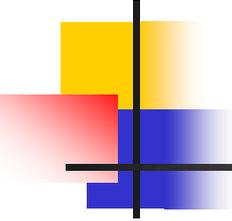


## NTFS安全性支持（续）

---

文件和目录的安全性权限设置规则：

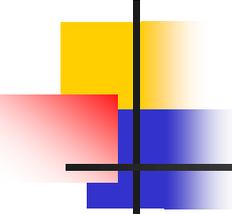
- 只有用户在被赋予其访问权限或属于拥有这种权限的组，才能对文件和目录进行访问。
- 权限是累积的，如果组A用户对一个文件拥有“写”权限，组B用户对该文件只有“读”权限，而用户C同属两个组，则C将获得“写”权限。



## NTFS安全性支持（续）

---

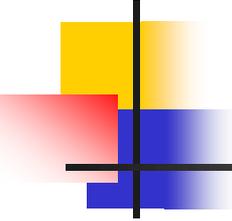
- “拒绝访问” 权限优先高于其他所有权限。如果组A用户对一个文件拥有“写”权限，组B用户对该文件有“拒绝访问”权限，那么同属两个组的C也不能读文件
- 文件权限始终优先于目录权限



## NTFS安全性支持（续）

---

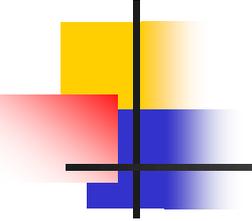
- NTFS的安全性支持——加密文件系统EFS (Encrypted File System)
- EFS加密技术是基于公共密钥的，它用一个随机产生的文件密钥FEK (File Encryption Key)，通过加强型的数据加密标准DESX (Data Encryption Standard) 算法对文件进行加密



## NTFS安全性支持（续）

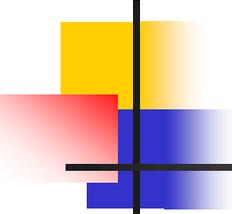
---

- EFS使用基于RSA (Rivest Shamir Adleman ) 的公共密钥加密算法对FEK进行加密，并把它和文件存储存一起，形成文件的一个特殊的EFS属性字段——数据加密字段DDF。
- 解密时，用户用自己的私钥解密存储在文件DDF中的FEK，再用解密后得到的FEK对文件数据进行解密，最后，得到文件的原文。



## 6.6.2 Linux文件系统

- Linux支持多种不同类型的文件系统，包括EXT/2/3、MINIX、UMSDOS、NCP、ISO9660、HPFS、MSDOS、NTFS、XIA、VFAT、PROC、NFS、SMB、SYSV、AFFS以及UFS等。



# 1、虚拟文件系统VFS

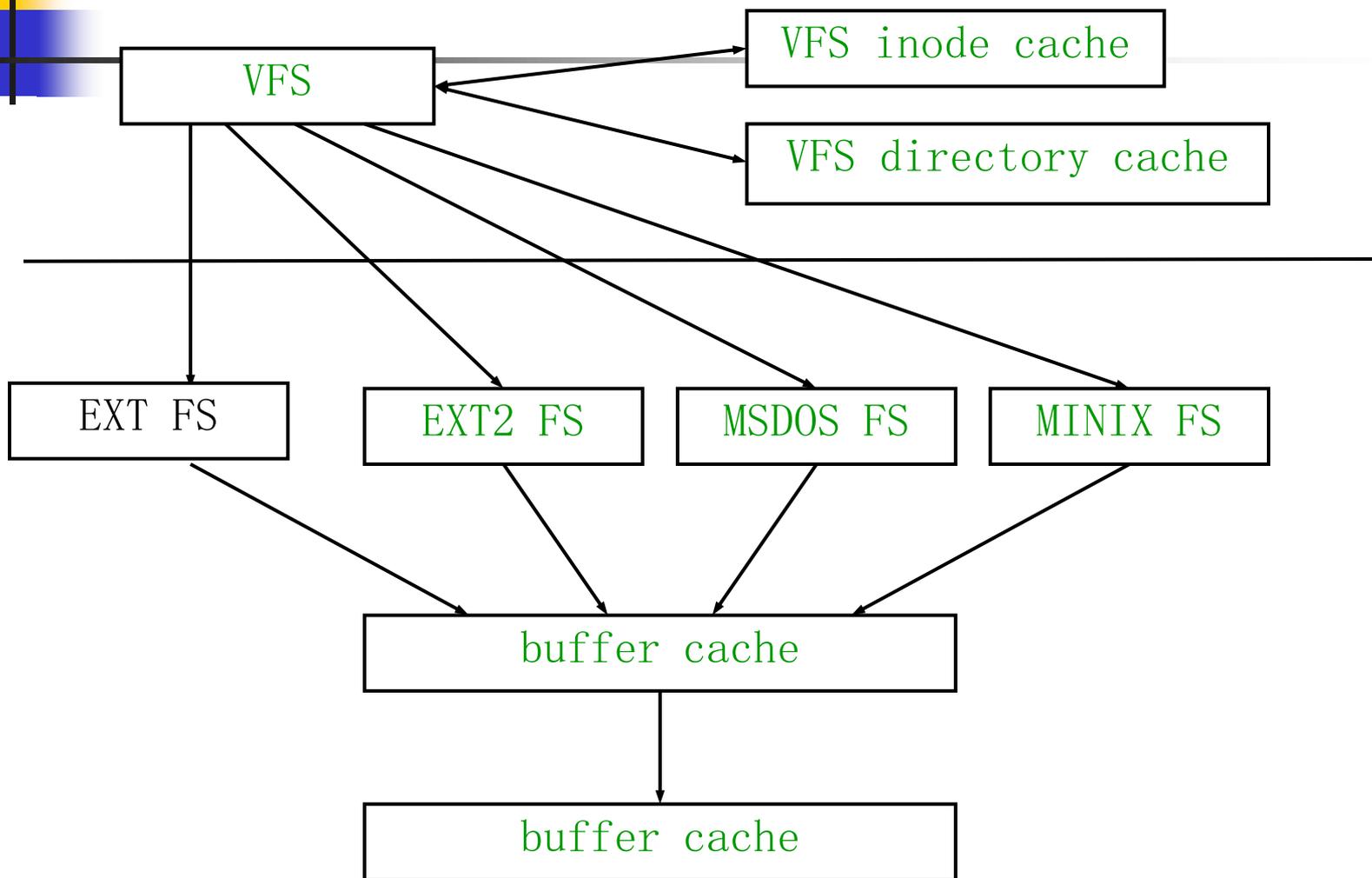
---

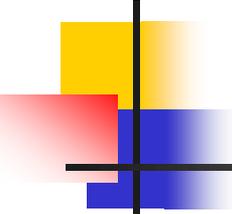
什么是VFS?

VFS的功能包括:

- 记录可用的文件系统类型
- 把设备与对应的文件系统联系起来
- 处理面向文件的通用操作
- 涉及具体文件系统的操作时，把它们映射到与控制文件、目录以及inode相关的物理文件系统

# 虚拟文件系统VFS





## 2、EXT文件系统

---

- EXT（1992年）和EXT2（1994年）是专为Linux设计的可扩展文件系统。
- 在EXT2中，文件系统组织成数据块的序列，数据块的长度相同，块大小在创建时被固定下来。
- EXT2把它所占用的磁盘逻辑分区划分为块组，每个块组依次包括超级块、组描述符表、块位图、inode位图、inode表以及数据块。