

第5章 文件系统

- Ø 1 文件的概念及实现
- Ø 2 文件系统
- Ø 3 Windows FSD体系结构
- Ø 4 Windows文件系统概述
- Ø 5 NTFS文件系统

Ø 文件系统：操作系统中与管理文件有关的软件和数据称为文件系统。它负责为用户建立文件，撤消、读写、修改和复制文件，按名存取和进行存取控制。

Ø 文件系统是计算机组织、存取和保存信息的重要手段。讨论文件的组织结构、存取结构、保护以及文件系统空间管理等问题。

Ø 文件：具有一定名称的一组相关数据的集合。

Ø 目录是由文件说明索引组成的用于文件检索的特殊文件。

文件概念与实现

Ø文件命名

Ø文件提供了一种将数据保存在外部存储介质上以便于访问的功能。

Ø用户就不必关心文件存储方法、物理位置以及访问方式等，而可以接通过文件名来使用文件。

文件属性

- Ø 文件名称:
- Ø 文件内部标识:
- Ø 文件物理位置:
- Ø 文件拥有者:
- Ø 文件权限:
- Ø 文件类型:
- Ø 文件长度:
- Ø 文件时间:

按文件的用途进行分类：

- 系统文件：包括操作系统内核、系统应用程序等。
- 库文件：这包括标准的和非标准的子程序库。标准的子程序库通常称为系统库，提供对系统内核的直接访问，而非标准的子程序库则是提供满足特定应用的库。库文件又分为两大类：一类是动态链接库，另一类是静态链接库。
- 用户文件：用户自己的文件，如用户的源程序、可执行程序 and 文档等。

按文件的性质进行分类:

- Ø 普通文件: 主要是系统所规定的普通格式的文件, 例如字符流组成的文件、库函数文件、应用程序文件, 等等。
- Ø 目录文件: 包含普通文件与目录的属性信息的特殊文件, 这主要是为了更好地管理普通文件与目录。
- Ø 特殊文件: 在UNIX系统中, 所有的输入输出设备都被看作是特殊的文件, 甚至在使用形式上也和普通文件相同。通过对特殊文件的操作可完成相应设备的操作。

按文件的保护级别进行分类:

- Ø 只读文件: 允许授权用户读, 但不能写。
- Ø 读写文件: 允许授权用户读写。
- Ø 可执行文件: 允许授权用户执行, 但不能读写。
- Ø 不保护文件: 所有用户都有一切权限。

按文件数据的形式进行分类:

Ø源文件: 源代码和数据构成的文件。

Ø目标文件: 指的是源程序经过编译程序编译, 但尚未链接成可执行代码的目标代码文件。

Ø可执行文件: 编译后的目标代码由连接程序连接后形成的可以运行的文件。

文件存取

- Ø 文件顺序访问是按从前到后的顺序对文件进行读写操作。这种存取方式最为简单。有的存储设备如磁带只能支持顺序访问。
- Ø 文件随机访问，也称为直接访问，可以按任意的次序对文件进行读写操作。有的存储设备如磁盘能支持随机访问
- Ø 文件索引访问，也称按键访问，这种方式对文件中的记录按某个数据项(通常称为键)的值来排列，从而可以根据键值来快速存取。如索引表很长，则可以将索引表再加以索引，以形成具有层次结构的多级索引。

文件操作

Ø文件创建:

Ø文件删除:

Ø文件截断:

Ø文件读:

Ø文件写:

Ø文件的读写定位:

Ø文件打开:

Ø文件关闭:

文件结构

Ø 文件结构是指文件的组织形式。文件结构分为文件的逻辑结构和文件的物理结构

Ø 文件的逻辑结构

Ø 记录式文件

Ø 流式文件

Ø 前者是指用户把每个文件分为若干记录单位，存取文件是以记录为单位来进行的

Ø 后者则是指文件由字符流组成，文件内部的信息不再划分单位。

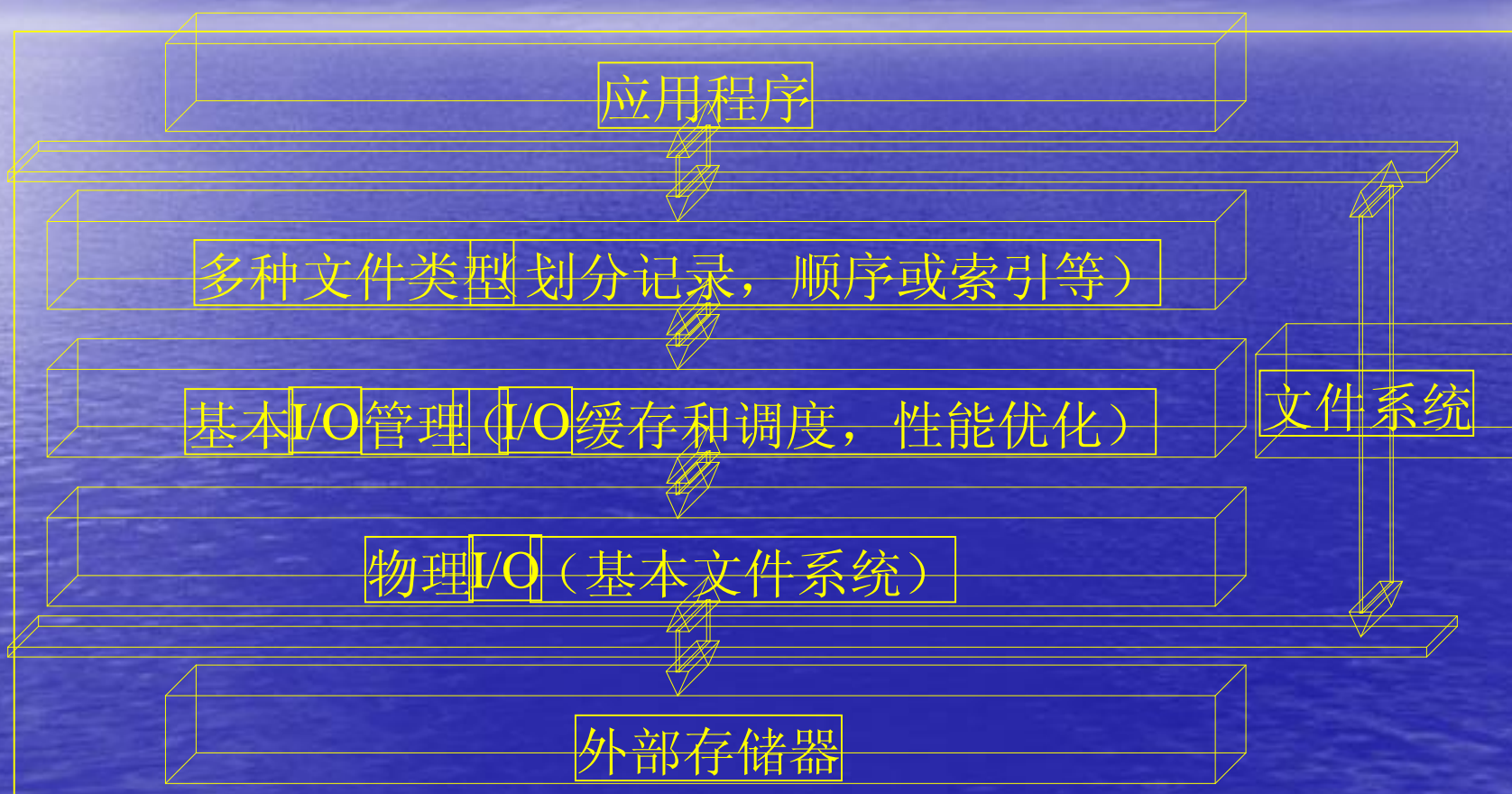
Ø 文件的物理结构则是指文件在外部存储介质上如何存放，也叫文件的存储结构。它对文件的存取方法有较大的影响

文件管理的目的

- Ø方便的文件访问和控制：以符号名称作为文件标识，便于用户使用；
- Ø并发文件访问和控制：在多道程系统中支持对文件的并发访问和控制；
- Ø统一的用户接口：在不同设备上提供同样的接口，方便用户操作和编程；
- Ø多种文件访问权限：在多用户系统中的不同用户对同一文件会有不同的访问权限；
- Ø优化性能：存储效率、检索性能、读写性能；
- Ø差错恢复：能够验证文件的正确性，并具有一定的差错恢复能力；

文件系统的结构和功能元素

文件系统的结构



Ø 文件系统的功能:

Ø (1) 用户可执行创建、修改、删除读写文件的命令。

Ø (2) 用户能以合适的方式构造他的文件。

Ø (3) 用户能在系统的控制下，共享其他用户的文件。

Ø (4) 允许用户用符号名访问文件。

Ø (5) 系统应有转存和恢复文件的能力，以防止意外事故的发生。

Ø (6) 系统应提供可靠保护及保密措施。

存取方法

- (1) 顺序存取方式: 其中的记录是按序排列的, 记录的存取也是按顺序进行的。
- (2) 直接存取方式: 用户对记录的存取是不按顺序的, 即用户可以直接指定某一记录进行存取。
- (3) 按键存取方式: 用户对文件内容的访问不是根据记录的编号或地址, 而是根据记录的某项内容(关键字)来进行的。

文件的组织

文件的组织是指从用户观点出发讨论文件内部的逻辑结构(logical structure)或用户访问模式；它可以独立于在外存上的物理存储。

文件的逻辑结构

逻辑文件有两种形式

一种是无结构的流式文件——有序相关信息项的集合，其基本单位是字节或字；

另一种是有结构的记录式文件——数据记录的集合，其基本单位是逻辑记录，记录的长度有等长或变长之分。

文件的物理结构

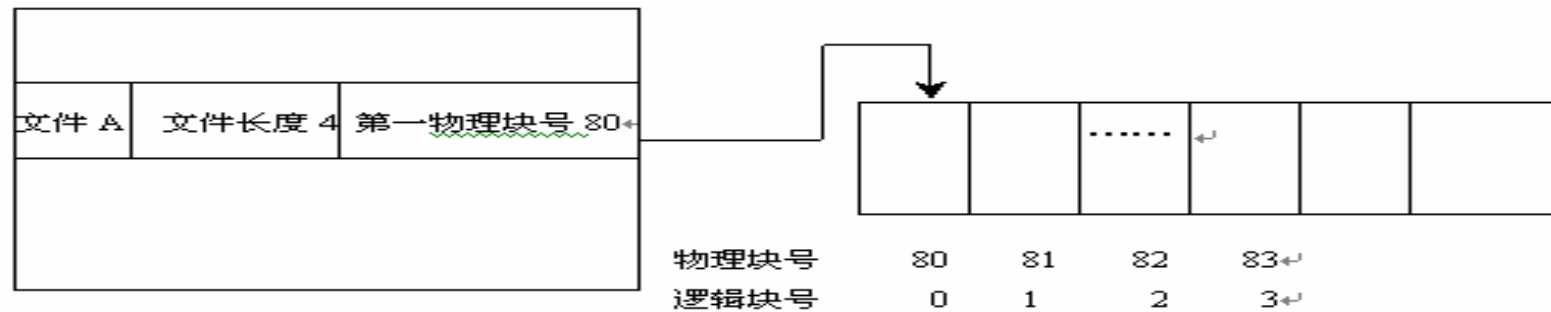
- Ø (1) 顺序结构: 是把一个在逻辑上连续的记录构成的文件分配到连续的物理块中。
- Ø (2) 链接结构: 把文件信息存放在非连续的物理块中, 在每个物理块中均设有一个指针指向其后续连续的另一个物理块, 从而使得存放同一文件的物理块链接成一个串联队列。
- Ø (3) 索引结构: 也称随机结构, 指为每个文件建立一个索引表, 其中每一个表项指出文件记录所在的物理块号, 表目按逻辑记录编写顺序排列或按记录内某一关键字顺序排列。

文件的连续结构

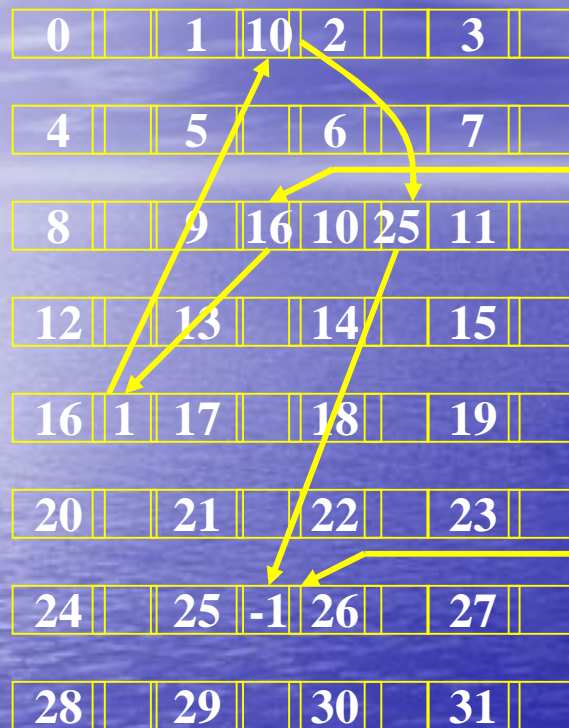


文件名	始址	块数
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

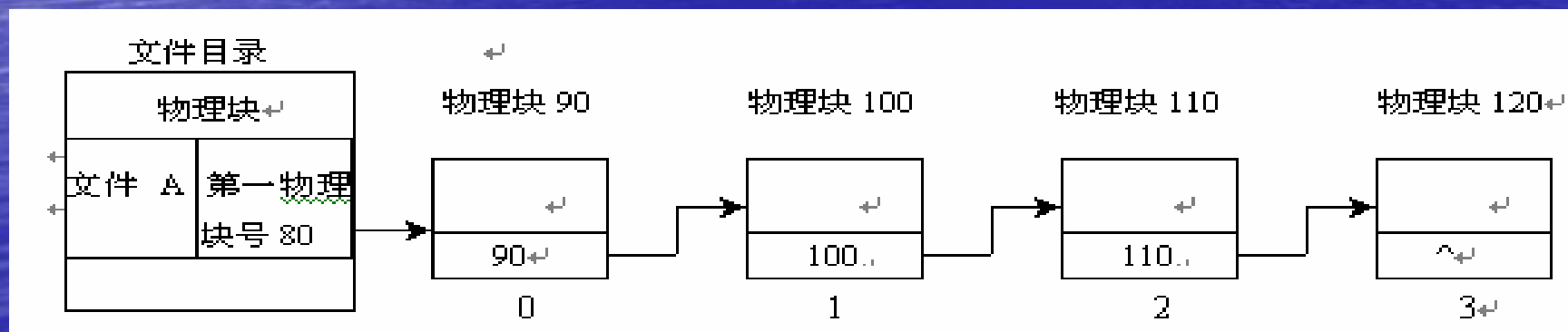
文件目录



文件的链接结构



文件名	始址	末址
jeep	9	25

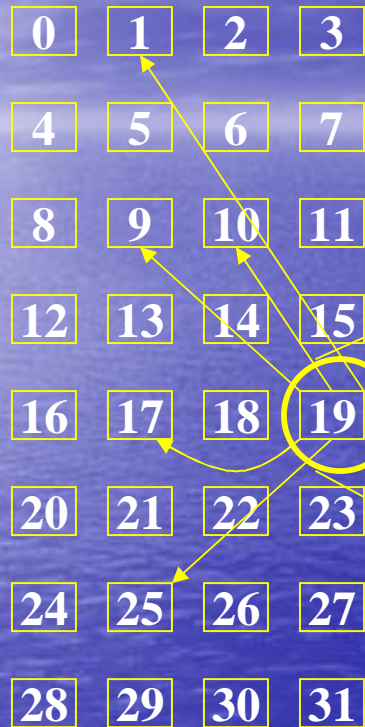


文件的索引结构

文件名 索引表地址

Jeep

19



19

9
16
1
10
25
-1
-1
-1

文件目录

索引表

物理块

文件 B 目录项

索引表指针

逻辑块号

物理块号

0

30

30

1

25

25

2

20

20

3

15

15

文件目录

目录是由文件说明索引组成的用于文件检索的特殊文件。文件目录的内容主要是文件访问的控制信息（不包括文件内容）。

- 1 目录内容
- 2 目录结构类型
- 3 文件别名的实现

目录内容

- Ø 文件名：字符串，通常在不同系统中允许不同的最大长度。可以修改。有些系统允许同一个文件有多个别名(alias)；
- Ø 别名的数目；
- Ø 文件类型：可有多种不同的划分方法，如：
 - Ø 有无结构（记录文件，流式文件）
 - Ø 内容（二进制，文本）
 - Ø 用途（源代码，目标代码，可执行文件，数据）
 - Ø 属性attribute（如系统，隐含等）
 - Ø 文件组织（如顺序，索引等）

地址信息

- Ø 存放位置：包括哪个设备或文件卷，以及各个存储块位置；
- Ø 文件长度：以字节、字或存储块为单位。可以通过写入或创建、打开、关闭等操作而变化。

访问控制信息

- 文件所有者：通常是创建文件的用户，或者改变已有文件的属主；
- 访问权限（控制各用户可使用的访问方式）：如读、写、执行、删除等；

使用信息

Ø 创建时间

Ø 最后一次读访问的时间和用户

Ø 最后一次写访问的时间和用户

目录结构类型

- Ø 一级目录：整个目录组织是一个线性结构，系统中的所有文件都建立在一张目录表中。它主要用于单用户操作系统。它具有如下的特点：
 - Ø 结构简单；
 - Ø 文件多时，目录检索时间长；
 - Ø 有命名冲突：如重名(多个文件有相同的文件名)
- Ø 二级目录：在根目录下，每个用户对应一个目录（第二级目录）；在用户目录下是该用户的文件，而不再有下级目录。适用于多用户系统，各用户可有自己的专用目录。

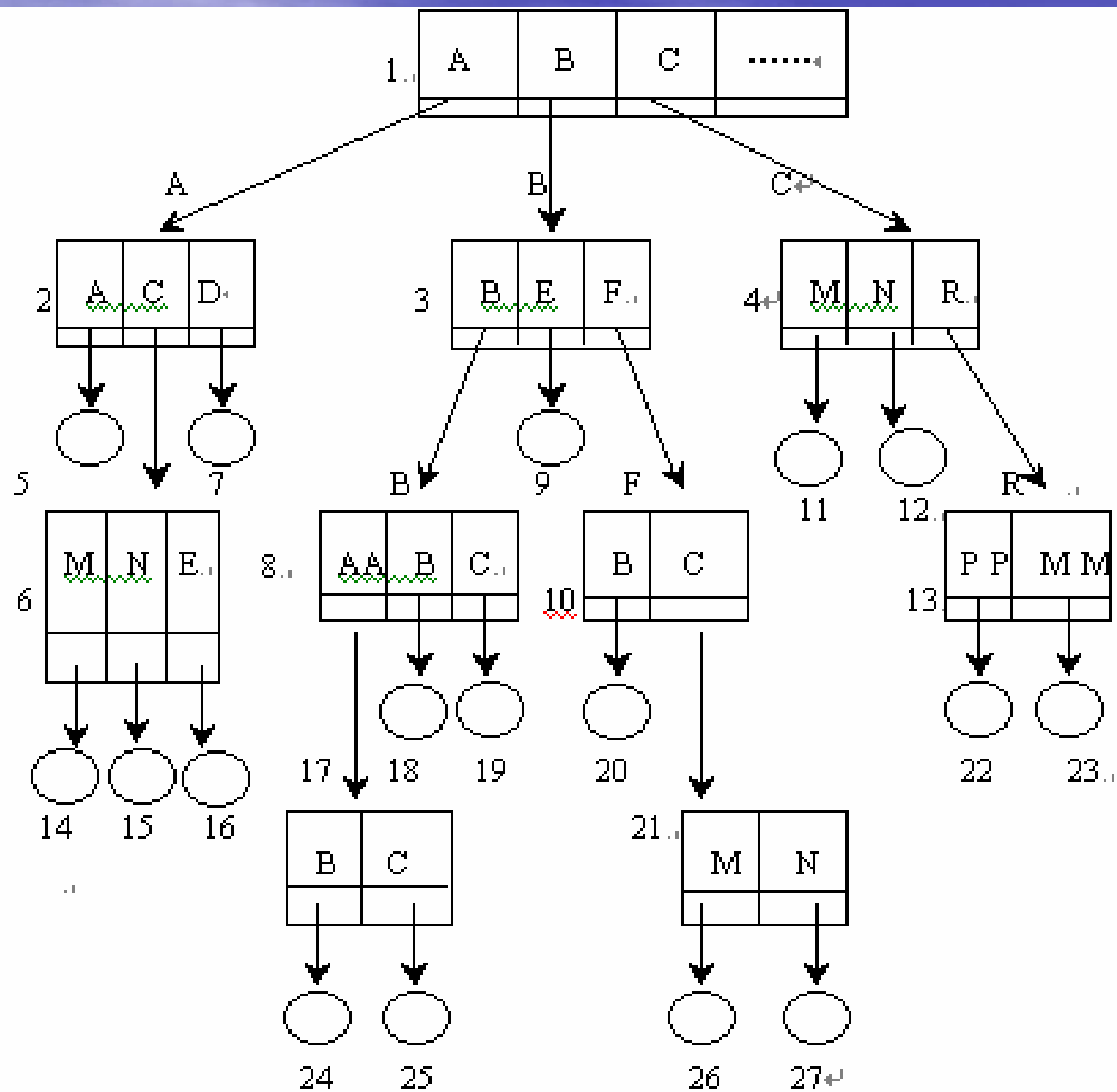
Ø 多级目录：或称为树状目录

Ø 目录名：可以修改。

Ø 目录树：中间结点是目录，叶子结点是目录或文件。

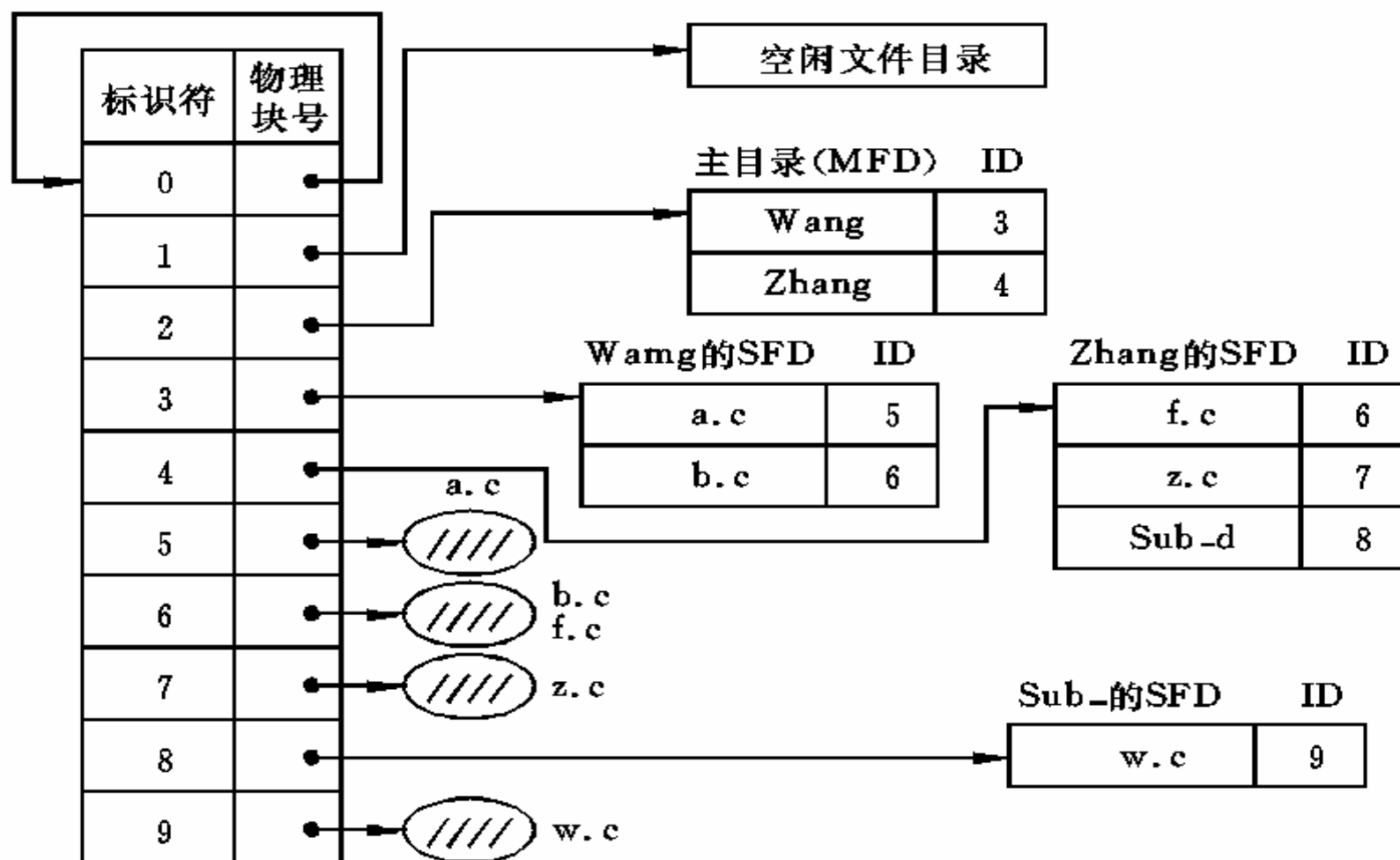
Ø 目录的上下级关系：当前目录、父目录、子目录、根目录等；

Ø 路径：每个目录或文件，可以由根目录开始依次经由的各级目录名，加上最终的目录名或文件名来表示；



改进的多级目录：为了提高目录检索速度，可把目录中的文件说明（文件描述符）信息分成两个部分：

- 符号文件目录：由文件名和文件内部标识组成的树状结构，按文件名排序；
- 基本文件目录（索引节点目录）：由其余文件说明信息组成的线性结构，按文件内部标识排序；



基本文件目录

文件的访问权限

设置文件访问权限的目的是为了在多个用户间提供有效的文件共享机制；

Ø 文件访问类型：

Ø 读：可读出文件内容；

Ø 写：可把数据写入文件；

Ø 执行：可由系统读出文件内容，作为代码执行；

Ø 删除：可删除文件；

Ø 修改访问权限：修改文件属主或访问权限

文件 用户	1	2	3	4	5	6	7	8
1	0	1	0	0	1	0	0	1
2	1	0	1	0	0	1	0	0
3	0	0	0	1	0	0	1	0
4	0	1	0	1	0	1	0	0
5	1	0	1	0	1	0	0	1
6	0	0	0	1	1	1	0	0
7	0	1	1	0	0	0	1	0

存取控制矩阵

用户 文件	WW
文件名	RWE
B组	R
B组	W
C组	E
其他	NONE

文件存储单位：簇（cluster）

文件的存储空间通常由多个分立的簇组成，而每个簇包含若干个连续的扇区(sector)。

Ø 簇的大小

Ø 两个极端：大到能容纳整个文件，小到一个外存存储块；

Ø 簇较大：提高I/O访问性能，减小管理开销；但簇内碎片浪费问题较严重；

Ø 簇较小：簇内的碎片浪费较小，特别是大量小文件时有利；但存在簇编号空间不够的问题（如FAT12、16、32）；

文件存储分配数据结构

- Ø 连续分配：只需记录第一个簇的位置，适用于预分配方法。可以通过紧缩将外存空闲空间合并成连续的区域。
- Ø 链式分配：在每个簇中有指向下一个簇的指针。可以通过合并将一个文件的各个簇连续存放，以提高I/O访问性能。
- Ø 索引分配：文件的第一个簇中记录了该文件的其他簇的位置。可以每处存放一个簇或连续多个簇（只需在索引中记录连续簇的数目）。

外存空闲空间管理方法

Ø位示图：每一位表示一个簇，取值0和1分别表示空闲和占用。

Ø空闲空间链接：每个空闲簇中有指向下一个空闲簇的指针，所有空闲簇构成一个链表。不需要磁盘分配表，节省空间。每次申请空闲簇只需取出链表开头的空闲簇即可。

Ø空闲空间索引：在一个空闲簇中记录其他几个空闲簇的位置。

文件卷

- Ø 磁盘分区：通常把一个物理磁盘的存储空间划分为几个相互独立的部分，称为"分区"。一个分区的参数包括：磁盘参数（如每道扇区数和磁头数），分区的起始和结束柱面等。
- Ø 文件卷：或称为"逻辑驱动器"。在同一个文件卷中使用同一份管理数据进行文件分配和外存空闲空间管理，而在不同的文件卷中使用相互独立的管理数据。
 - Ø 一个文件不能分散存放在多个文件卷中，其最大长度不超过所在文件卷的容量。
 - Ø 通常一个文件卷只能存放在一个物理外设上（并不绝对），如一个磁盘分区或一盘磁带。

Windows FSD体系结构

- 文件系统驱动程序FSD (File System Driver) 是Windows 2000/xp下的标准核心层驱动程序
- Windows 2000 的文件系统有两类：
 - 一类是本地文件系统FSD;
 - 一类是远程文件系统FSD。

- 文件系统驱动程序和内存管理器、高速缓存管理器有机的结合在一起,有着密不可分的关系.
- FSD 通常使用高速缓存(Cache) 管理器以高速缓存文件数据.
- FSD 还和内存管理器结合起来,以使内存映射文件能被正确实现

用户模式

内核模式

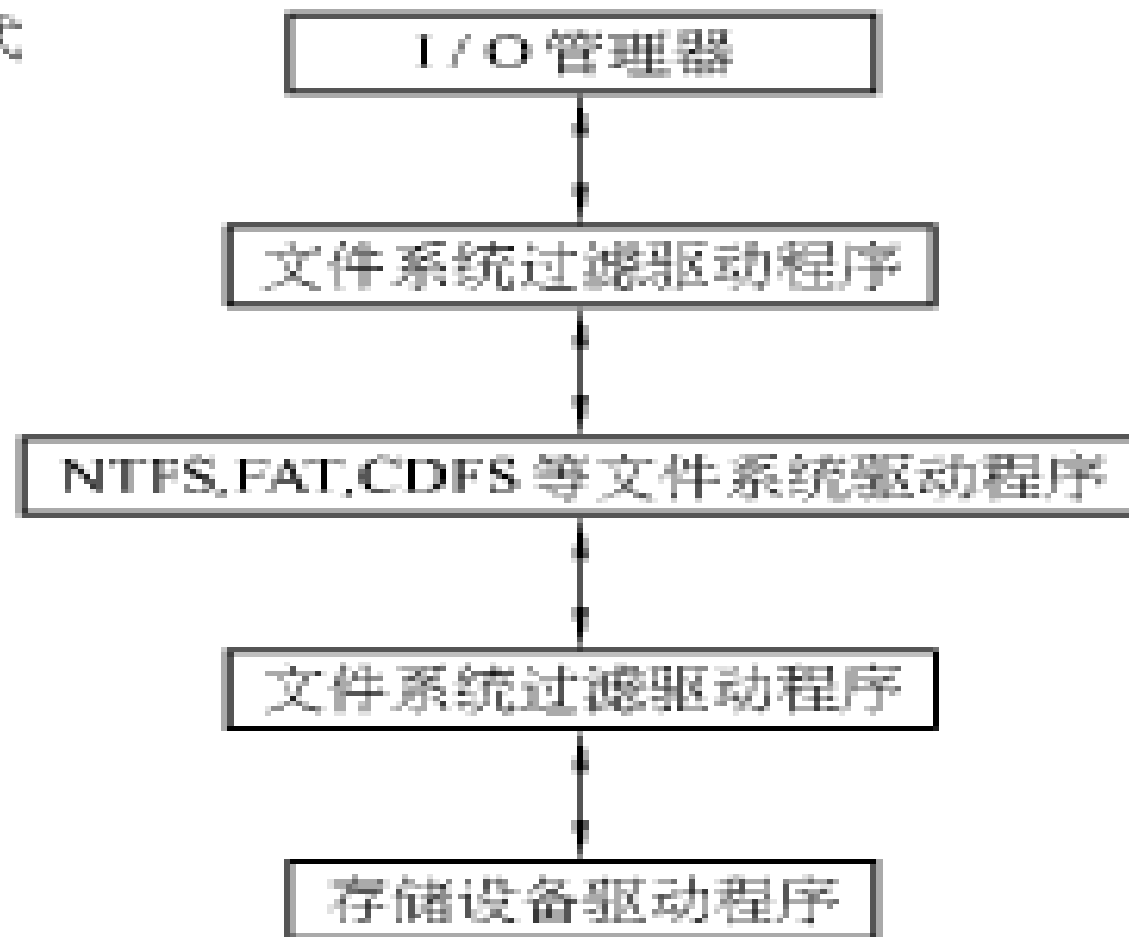
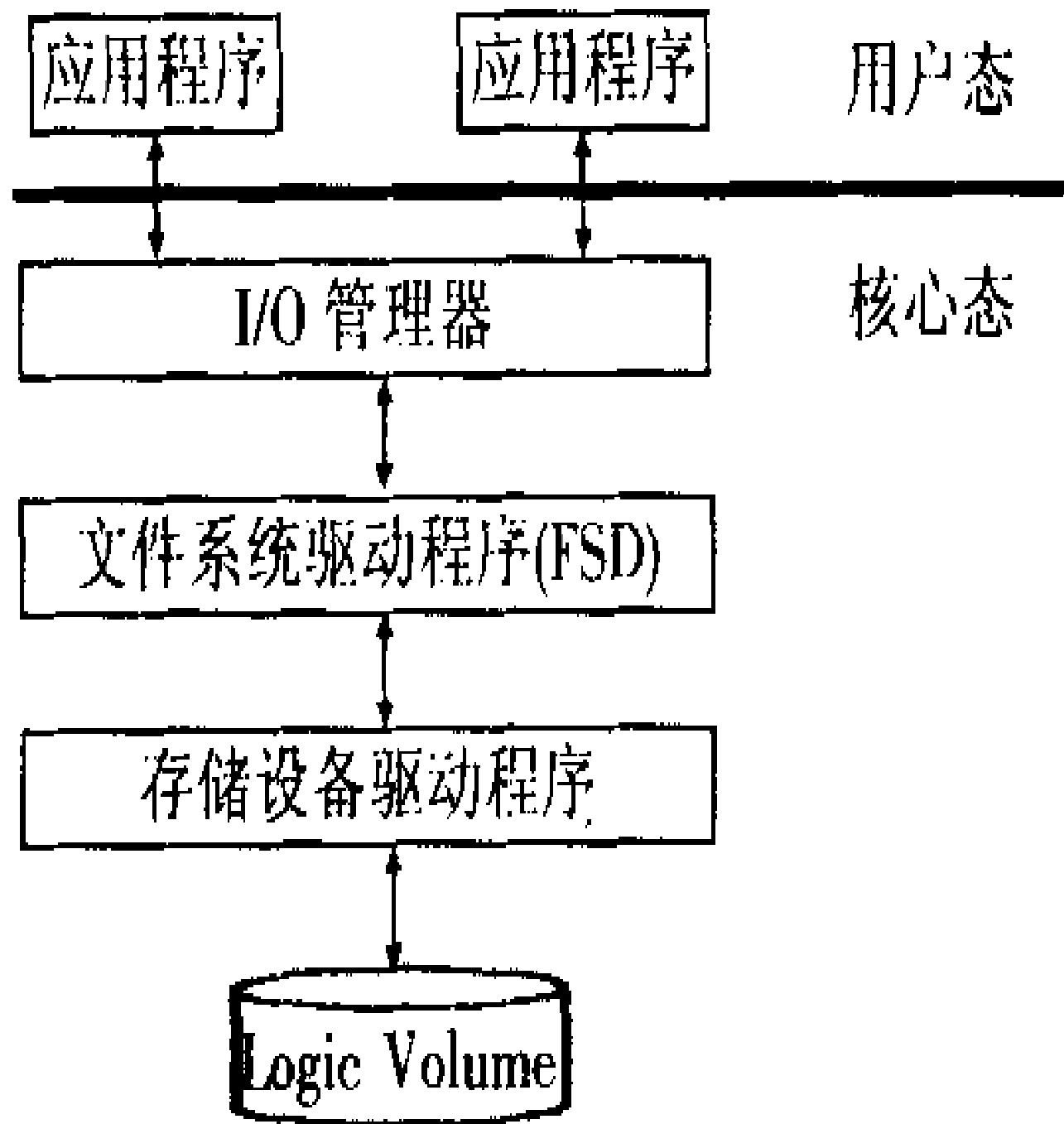


图 1 文件系统过滤驱动程序在系统中的位置

FSD 的功能

- (1) 接收由用户模式来的系统函数NtReadFile、NtWriteFile。
- (2) 判断读写I/ O 请求,如果是Cached I/ O 请求,则把它送到Cache 管理器,并且还调用CcInitilize2 CacheMap 进行Cache 初始化。
- (3) 如果是nonbuffered I/ O 请求则会直接发送到底层设备驱动程序。
- (4) 如果有请求信息,调用NtQueryInformation 函数,交给文件大小控制块,再由它发给Cache 管理器,由它访问Cache 中的文件的信息。

- (5) 如果是Fast I/ O(快速I/ O 是一种读取或写入告诉缓存的方法,通过此方法可以不会产生IRP) , FSD 直接调用Fast IORead 和Fast IOWrite。
- (6) 如果有缺页现象,由Cache 管理器报告给内存管理器,处理后返回FSD ,它的分页处理模块交给cached I/ O ,然后发送到下一层磁盘设备驱动程序



- I/O 管理器接收到请求,经过处理之后派发给文件系统驱动程序层。
- 文件在磁盘卷上有一定的结构,FSD 就负责创建各种文件系统格式及相应的数据结构
- FSD 就将请求发往设备驱动层,最后由存储设备驱动程序进行数据与逻辑盘之间的传送,从而实现了用户态程序对磁盘的读写操作。

卷参数块(VPB)

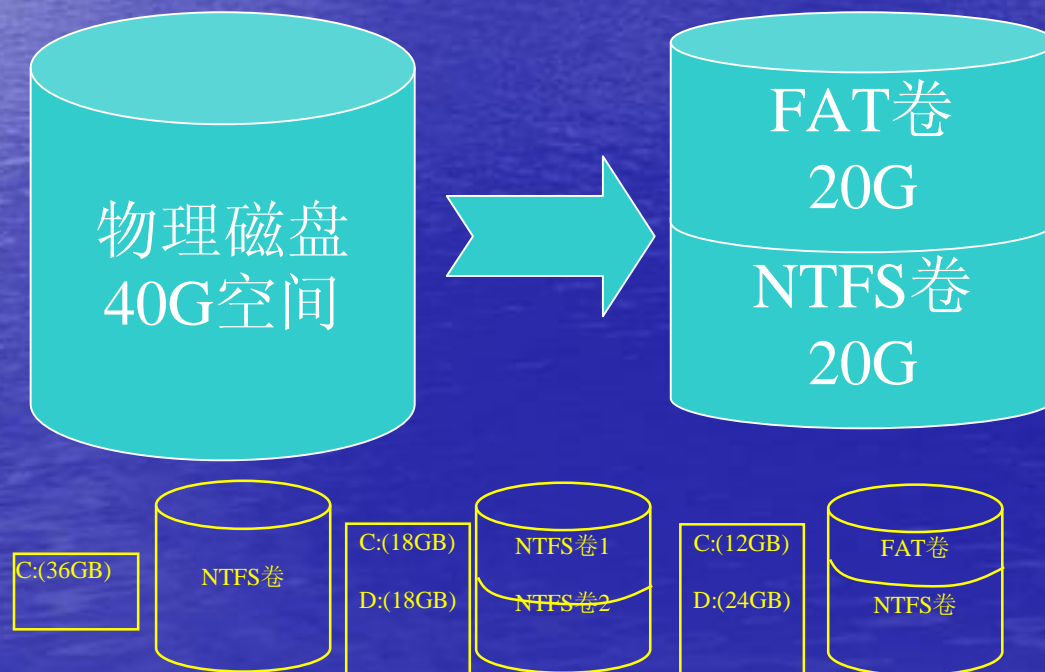
- 与两个设备对象联系在一起的就是VPB ,这个结构的作用是用来创建一个逻辑设备对象和物理设备对象之间的关联

Windows2000的文件系统

Windows2000(W2K)支持许多文件系统，包括运行于Windows95、MS-DOS和OS / 2上的文件分配表(FAT)。W2K的开发者们还设计了一个新的文件系统——W2K文件系统(NTFS)，用于满足工作站和服务器的要求。

- Ø 支持NTFS文件系统之外，也支持FAT文件系统和CDFS文件系统。（注：CDFS是CD-ROM的文件系统，用于读取CD-ROM，是一种只读的文件系统，很少使用，这里不予介绍。）
- Ø NTFS文件系统同DOS下长期使用的FAT文件系统相比，有两个主要的方面优点：一是NTFS许可权限，即本地安全性，二是NTFS支持对单个文件或者文件夹进行压缩。

卷：建立在磁盘分区上。Windows2000常用
两种卷：FAT卷和NTFS卷



FAT12、FAT16、FAT32

文件分配表 (File Allocation Table, FAT)

每一种fat文件系统都用一个数字来标识磁盘上簇号的位数。

例如FAT12的簇标识为12位，单个分区最多存储 2^{12} (4096)个簇，FAT12簇大小在512B与8KB之间。FAT12卷至多有32MB.

用途:作为5.25英寸（1.2M）和3.5英寸（1.44MB）软盘标准格式。

Ø FAT16文件系统1982年应用于MS-DOS中，16位簇标识能够定位65536个簇

Ø FAT16簇大小为512B到64KB,卷大小为4GB,Windows限制在2GB

引导 区	文件 分配 表1	文件 分配 表2	根目 录	其他 文件 目录
---------	----------------	----------------	---------	----------------

Ø 文件分配表包含一个卷上所有簇的条目，而且有一个备份

文件分配表的条目还定义了文件分配链来连接文件和目录。

链上的链接是文件存储的下一个族的标号。

FAT32 : 16位簇标识扩充为32位, FAT32簇大小可达32KB, 理论上具有8TB的寻址能力。Windows2000/xp则限制它的卷大小为32GB

NTFS的重要特征：

可恢复性：之所以需要建立新W2K文件系统，就是为了具备从系统崩溃和磁盘故障中恢复

数据的能力。当发生这类故障时，NTFS能够重建文件卷，它是通过为文件系统的变化使用一个事务处理模型来实现这一点的。文件系统的每个重要的变化都被看作是一个原子动作，或者完全执行，或者根本就不执行。当发生故障时，每个正在处理的事务随后或者取消，或者完成。

NTFS支持基于每个文件、每个目录和每个卷的压缩。通常，NTFS压缩只能在用户数据上执行，而不能在文件系统元数据上执行。通过使用Win32中的GetVolumeInformation函数可以判断一个卷是否已被压缩；要得到一个文件的实际压缩大小，可用Win32中的GetCompressedFileSize函数；使用DeviceIoControl函数则可检查或改变一个文件或目录的压缩设置。

安全性：NTFS使用W2K对象模型来实施安全机制。一个打开的文件作为一个文件对象来实现，并且有一个定义它的安全属性的安全描述符。安全描述符作为文件的一部分存储在磁盘上。

大磁盘和大文件：NTFS比包括FAT在内的其他大多数文件系统能够更有效地支持非常大的磁盘和非常大的文件。

Ø多数据流：文件的实际内容被当作字节流处理。
在NTFS中可以为一个文件定义多个数据流，与文件相关的每个信息单元，包括文件名、文件拥有者、文件时间标记、文件内容等都当做NTFS对象属性。

ØWindows资源管理器也使用多数据流。右击NTFS文件并选择属性，可以为文件关联标题、主题、作者、关键词等，它们作为信息流加入文件中去

Ø通用索引功能：NTFS中，每个文件都有一组属性与之相关联。文件管理系统中文件描述的集合组织成一个关系数据库，因而文件可以建立关于任何属性的索引。

ØNTFS文件系统能够在在一个磁盘卷中索引文件属性。能够有效定位匹配某种条件的文件。

NTFS卷和文件结构

Ø NTFS使用下列磁盘存储概念：

Ø 扇区：磁盘中最小的物理存储单元。一个扇区中能存储的数据量字节数总是2的幂，并且通常为512个字节。

Ø 簇：作为磁盘空间分配和回收的基本单位。一个或多个连续的扇区(在同一磁道上一个接一个)。一个簇中扇区的数目也为2的幂。

Ø 卷：磁盘上的逻辑分区。由一个或多个簇组成，供文件系统分配空间时使用。

Ø 一个卷包括文件系统信息、一组文件以及卷中剩余的可以分配给文件的未分配空间。

Ø NTFS并不识别扇区，簇是最基本的分配单位。例如，假设每个扇区为512个字节，并且系统

Ø 为每个簇配置两个扇区(1簇=1KB)。如果一个用户创建了一个1600个字节的文件，则给该文件分配2簇。如果用户后来又把文件修改成3200个字节，则再分配另外2簇。分配给一个文件的簇不需要一定是连续的，即允许一个文件在磁盘上被分成几段。

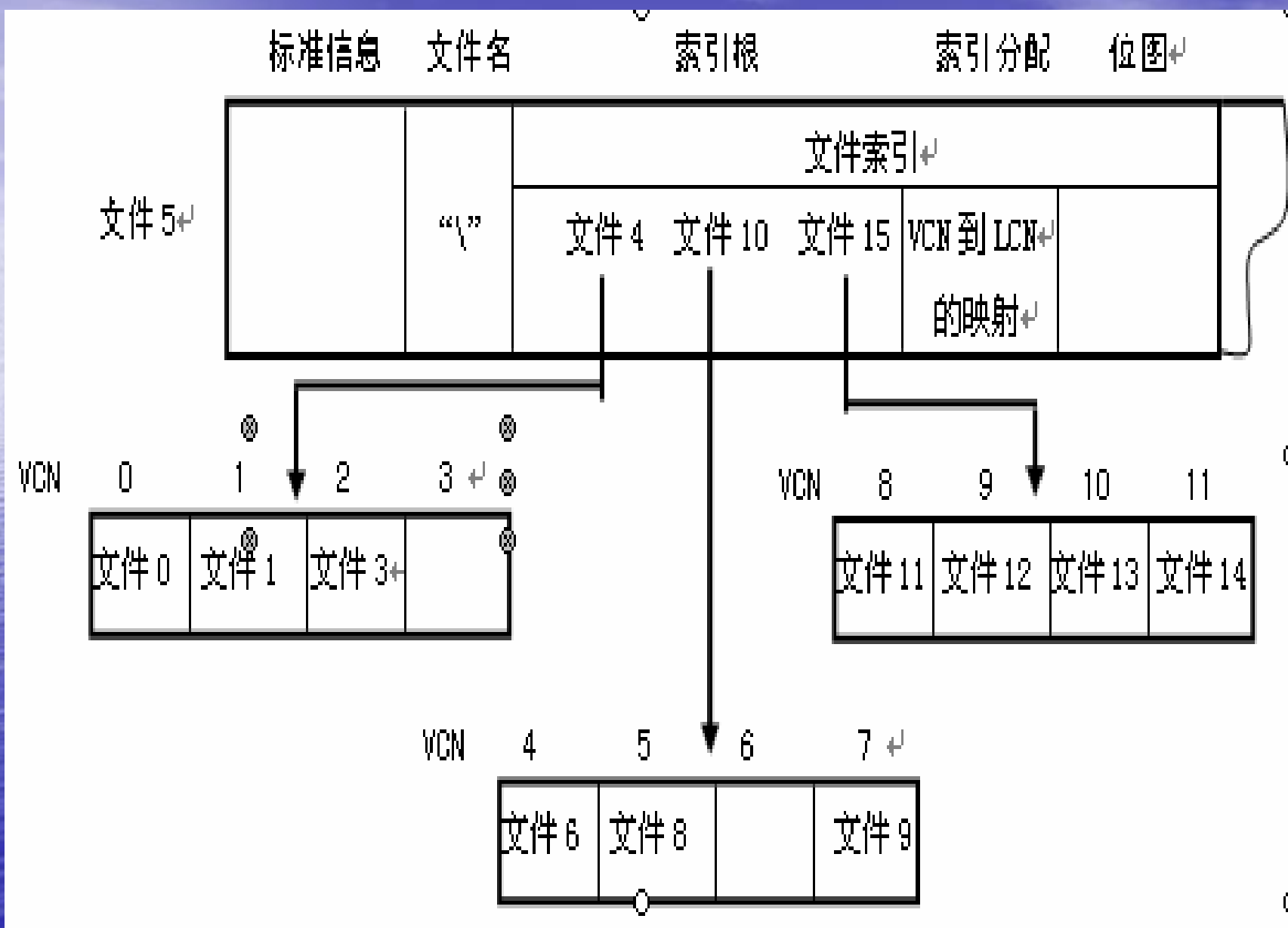
Ø 使用簇进行分配使得NTFS不依赖于物理扇区的大小。这使得NTFS能够很容易地支持扇区大小不是512个字节的非标准磁盘，并且能够通过使用比较大的簇有效地支持非常大的磁盘和非常大的文件

Ø NTFS使用逻辑簇号（LCN）和虚拟簇号（VCN）来进行簇的定位。

Ø 逻辑簇号：对整个卷从头到尾进行简单编号。簇大小*逻辑簇号就得到卷上的物理字节偏移量，从而得到物理磁盘地址。

Ø 虚拟簇号：对属于特定文件的簇从头到尾进行编号，以便引用文件中的数据。

Ø VCN可以映射成为LCN



NTFS卷布局

Ø1 分区引导扇区

Ø2 主文件表

Ø3 系统文件

Ø4 文件区域

分区引导扇区(partitionbootsector)(尽管它被称作一个扇区，但它可能有16个扇区那么长)，引导扇区包含卷的布局信息、文件系统的结构以及引导启动信息和代码。

主文件表MFT，主文件表包含关于在这个NTFS卷中所有文件和文件夹(目录)的信息以及关于可用的未分配空间的信息。事实上，MFT是这个NTFS卷中所有内容的列表，被组织成关系数据库结构中的许多行

Ø 下面简要描述一下MFT的元数据文件。

Ø MFT中的第1个记录就是MFT自身。为了确保文件系统结构的可靠性，系统专门为它准备了一个镜像文件(\$MftMirr)，也就是MFT中的第2个记录。

Ø 第3个记录是日志文件(SLogFile)。该文件是NTFS为实现可恢复性和安全性而设计的。当

Ø 系统运行时，NTFS就会在日志文件中记录所有影响NTFS卷结构的操作，包括文件的创建和改变目录结构的命令，例如复制，从而在系统失败时能够恢复NTFS卷。

第4个记录是卷文件(\$Volume)，它包含了卷名、被格式化的卷的NTFS版本和一个标明该磁盘是否损坏的标志位(NT-FS系统以此决定是否需要调用Chkdsk程序来进行修复)。

第5个记录是属性定义表(\$AttrDef)，其中存放了卷所支持的所有文件属性，并指出它们是否可以被索引和恢复等。

第6个记录是根目录(\)，其中保存了存放于该卷根目录下所有文件和目录的索引。在访问了一个文件后，NTFS就保留该文件的MFT引用，第二次就能够直接进行对该文件的访问。

第7个记录是位图文件(\$Bitmap)。NTFS卷的分配状态都存放在位图文件中，其中每一位代表卷中的一簇，标识该簇是空闲的还是已被分配了的。

第8个记录是引导文件(\$Boot)，它是另一个重要的系统文件，存放着Windows 2000 / XP的引导程序代码。该文件必须位于特定的磁盘位置才能够正确地引导系统。该文件是在Format程序运行时创建的，可以通过普通的文件I / O操作来修改。

第9个记录是坏簇文件(\$BadClus)，它记录了磁盘上该卷中所有的损坏的簇号，防止系统对其进行分配使用。

第10个记录是安全文件(\$Secure)，它存储了整个卷的安全性描述符数据库。NTFS文件和目录都有各自的安全描述符

第11个记录为大写文件(\$UpCase, uppercasefile)，该文件包含一个大小写字符转换表。

第12个记录是扩展元数据目录(\$Extendedmetadatadirectory)。

从第13到第16个记录为预留，尚未使用。

文件0	MFT
1	MFT副本（部分）
2	日志文件
3	卷文件
4	属性定义表
5	根目录
6	位图文件
7	引导文件
8	坏簇文件
	▪
	▪
	▪
16	用户文件和目录

系统文件，长度大约1M

包含：

- 1 关于MFT前三行的镜像，保证一个扇区失败仍可访问MFT
- 2 日志文件：事务处理，用于NTFS的恢复
- 3 簇的位映像：说明哪一簇正被使用
- 4 属性定义：每个文件包含一组属性

Ø 磁盘镜像是将一个硬盘与另一个硬盘作数据镜像的过程。镜像硬盘包含主磁盘上数据信息的精确复制。磁盘镜像的目的是为了保证，如果主硬盘失效，镜像硬盘能继续工作，而不致丢失任何服务。这是恢复信息的 fastest 手段，因为镜像硬盘包含了来自主磁盘的精确信息，不需要重新建立信息。

Ø 任何分区都可以镜像，包括系统分区和引导分区。唯一的要求是主分区和镜像分区必须大小相同

