第8章 磁盘存储器的管理（4学时）

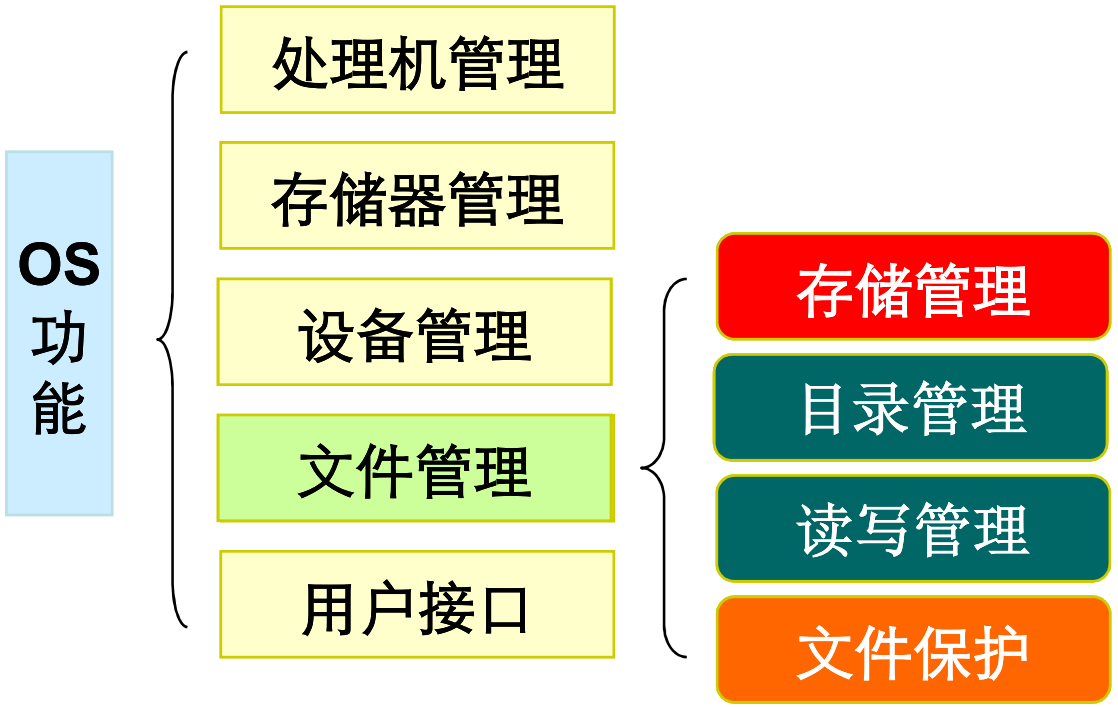
主讲教师：张春元

联系电话：13876004640

课程邮箱：haidaos@126.com

邮箱密码：zhangchunyuan

本章内容所处位置



本章主要内容

- 8.1 外存组织（分配）方式
- 8.2 文件存储空间的管理
- 8.3 提高磁盘I/O速度的途径
- 8.4 提高磁盘可靠性的技术
- 8.5 数据一致性控制

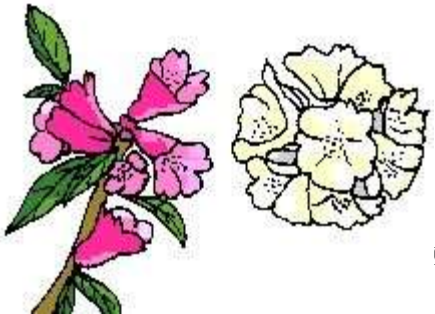
8.1 外存组织方式

❖ 文件的物理结构

* 文件的物理结构是指逻辑文件在外存上的存储组织形式，它与存储介质的存储特性有关，还与所采用的外存组织方式有关。
* 物理块是存储和传输信息的基本单位，物理块与外存设备有关。
* 文件在逻辑上都可看作是连续的，但在物理设备上存放时却有不同的方式，如连续组织、链接组织、索引组织等。

8.1 外存组织方式

- 8.1.1 连续组织
- 8.1.2 链接组织
- 8.1.3 FAT和NTFS技术
- 8.1.4 索引组织

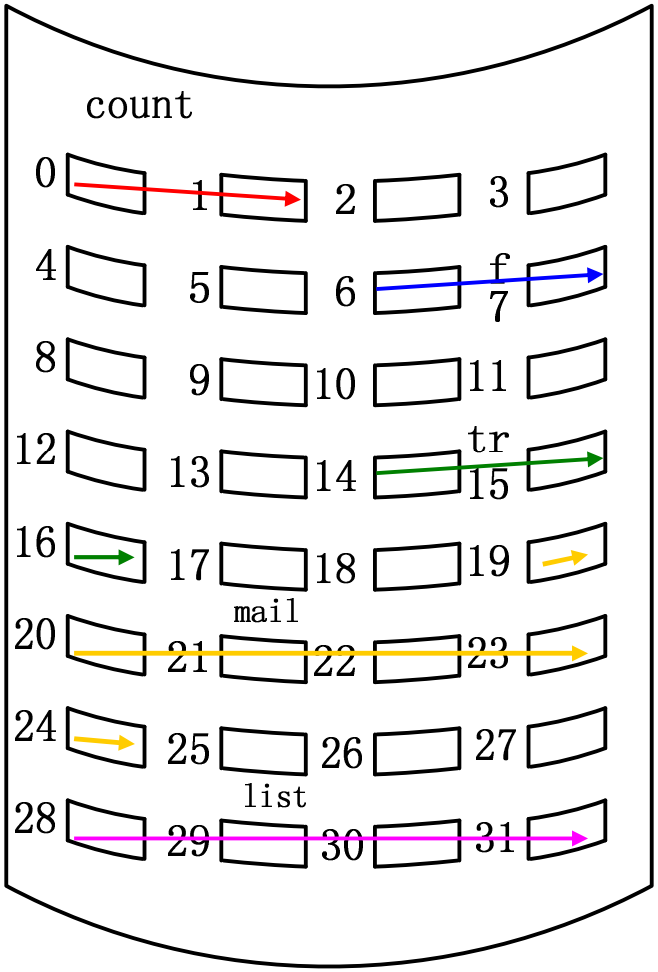


8.1.1 连续组织

❖ 1、连续组织方式

* 连续组织(Continuous Allocation)要求为每一个文件分配一组相邻接的盘块。一组盘块定义了磁盘上的一段线性地址。
* 在采用连续组织方式时，可把逻辑文件中的记录顺序地存储到邻接的各物理盘块中，这样所形成的文件物理结构称为顺序文件结构，此时的物理文件称为顺序文件。
* 文件对应的目录项（属性）中包含：始址、总块数、最后一块字节数（便于计算文件的总字节数）。

8.1.1 连续组织

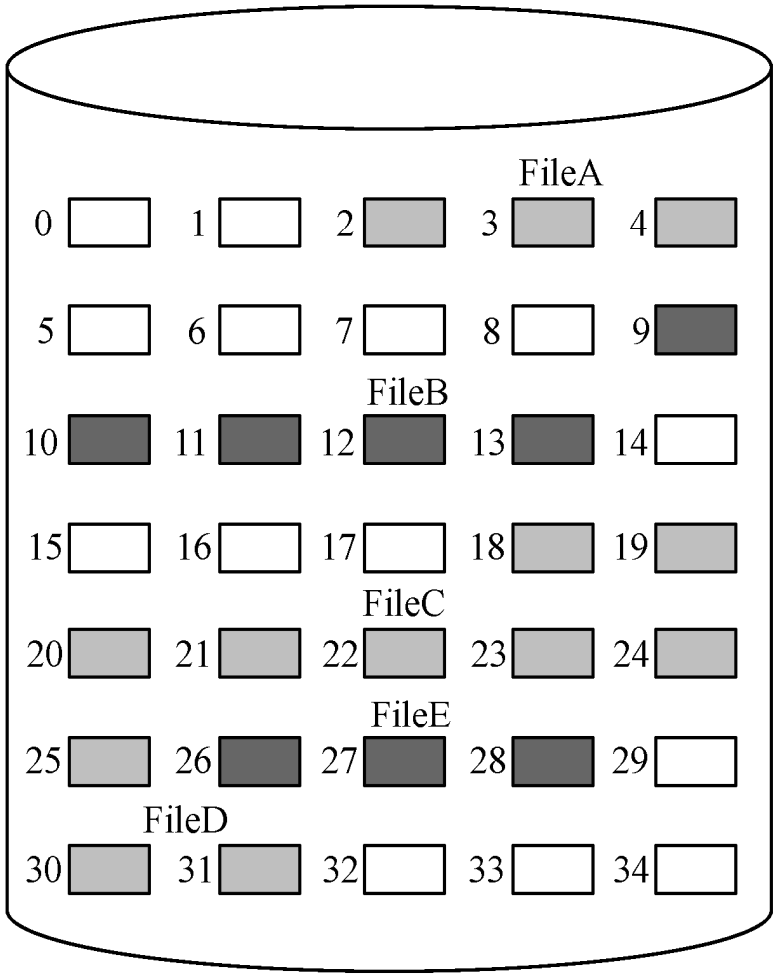


目录

file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

图8-1 磁盘空间的连续组织

8.1.1 连续组织



文件分配表

文件名	起始块	长度
FileA	2	3
FileB	9	5
FileC	18	8
FileD	30	2
FileE	26	3

图8-1 磁盘空间的连续组织

8.1.1 连续组织

❖ 2、连续组织的优缺点

* 优点

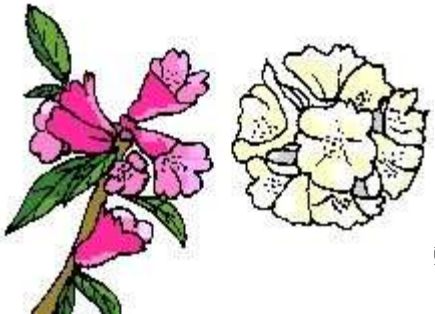
- 因磁头移动距离小，顺序访问容易且速度快。

* 缺点

- 要求连续空间，一段时间后需整理磁盘（采用紧凑方法）以消除外部碎片。
- 分配前必须事先知道文件长度
- 不能灵活地删除与插入记录。
- 不利于文件的动态增长。

8.1 外存组织方式

- 8.1.1 连续组织
- 8.1.2 链接组织
- 8.1.3 FAT和NTFS技术
- 8.1.4 索引组织



8.1.2 链接组织

1、链接组织方式(Chained Allocation)

* 链接组织将文件离散地分配于各盘块中。
* 在采用链接组织方式时，通过链接指针将同属于一个文件的多个离散的盘块链接成一个链表（链接指针如果分散存储在各盘块中，称为隐式链接；链接指针如果统一存储在一张表中，称为显式链接），把这样形成的物理文件称为链接文件。

8.1.2 链接组织

❖ 2、链接组织的类型

* 1> 隐式链接

- 特点：链接指针分散存储在各盘块中，另外文件的目录项中都须含有指向链接文件第一个盘块和最后一个盘块的指针。
- 一种改进办法：为了提高检索速度和减小指针所占用的存储空间，可以将几个连续的盘块组成一个簇，盘块分配时，以簇为单位进行。这种方法虽然可以减小指针所占用的存储空间，但却增大了内部碎片。

8.1.2 链接组织

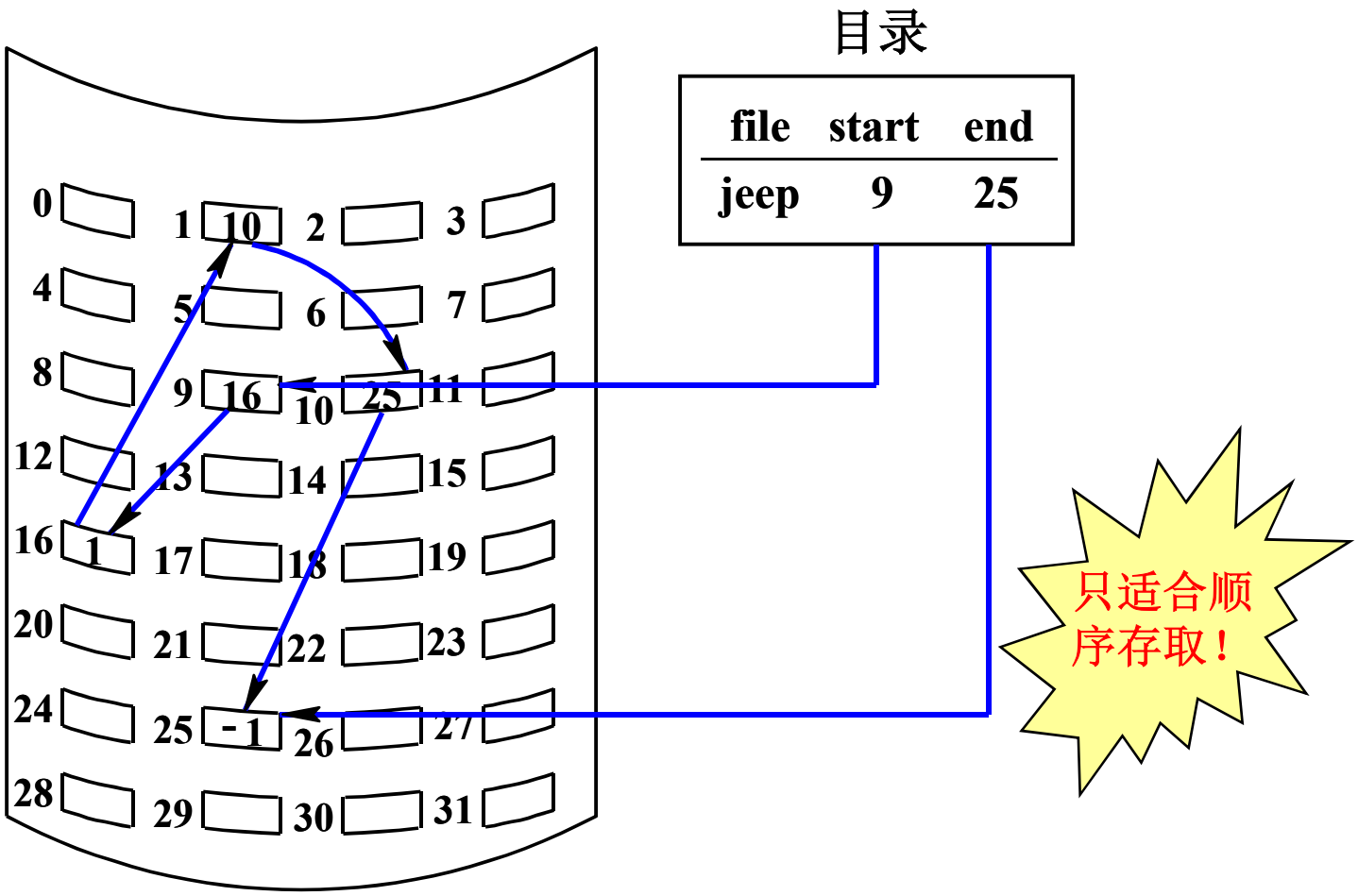


图8-2 磁盘空间的隐式链接组织

8.1.2 链接组织

* 2> 显式链接

  - 特点：链接指针统一存储在一张文件分配表FAT中，并将文件的首个物理块地址登记在它的目录项（也称文件控制块FCB）中。
  - FAT：在一个文件卷(逻辑盘)中仅设置一张FAT（实际上配有两个文件分配表：FAT1和FAT2。FAT1用于日常工作，FAT2备用，因此，在FAT1损坏时，可用FAT2表修补），其每个表项的序号为对应的物理块号，而表项中的内容则是分配给文件的下一个物理块的指针，即下一个盘块号。
  - 优点：FAT常驻内存，检索速度快，可解决隐式链接查找时需多次访问磁盘的问题。

8.1.2 链接组织

目录

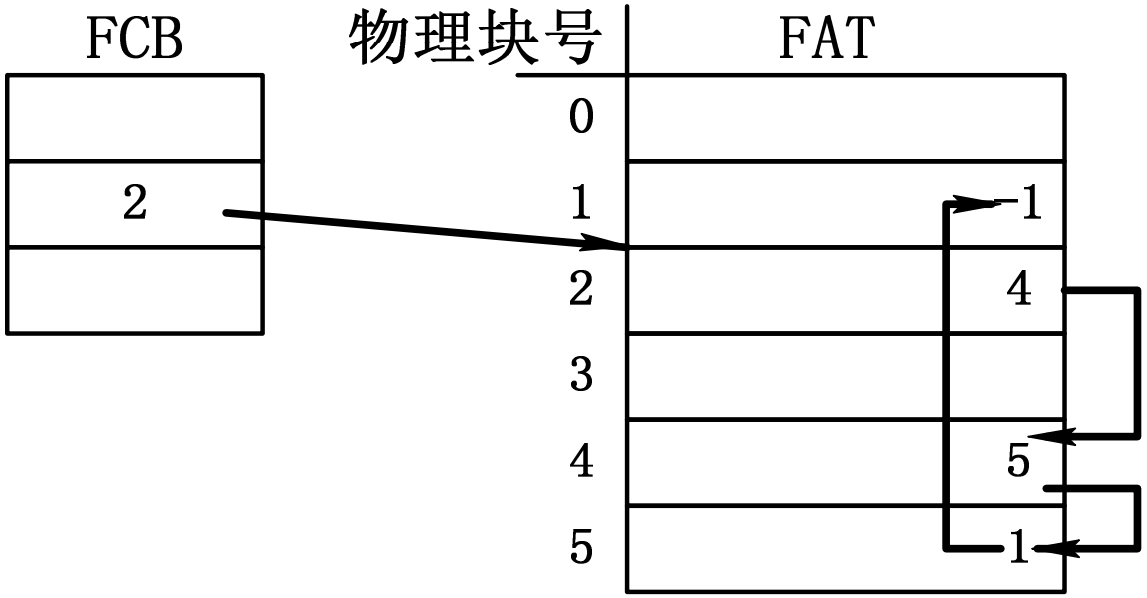


图8-3 显式链接结构

8.1.2 链接组织

❖ 3、链接组织的优缺点

* 优点

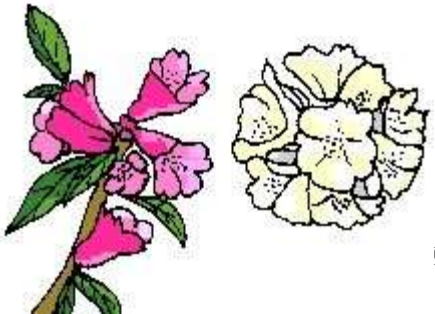
- 消除了外部碎片，提高外存利用率。
- 文件动态增长时，可动态地为它分配盘块。
- 文件的增删改方便(这种文件结构不要求连续存放)。
- 文件创建时用户不必指出文件的大小。

* 缺点

- 隐式链接：只适用于顺序存取，随机存取效率太低。若要查找文件中的某一块必须从头开始；如果访问文件的最后的内容，实际上是要访问整个文件；可靠性差。若某一块出错，则整个链断开，文件不完整。
- 显式链接：每个盘块需一个链接指针，采用FAT需占用较大内存。

8.1 外存组织方式

- 8.1.1 连续组织
- 8.1.2 链接组织
- 8.1.3 FAT和NTFS技术
- 8.1.4 索引组织



8.1.3 FAT和NTFS技术

❖ 1、FAT和NTFS技术的发展

* **FAT12→FAT16→FAT32→NTFS**

  (MS-DOS)　　(Win95/98)　(Win NT/2000/XP…)

* 卷（分区）
  - 一个或多个物理磁盘可以分成一个或多个逻辑磁盘，每个逻辑磁盘就是一个卷。
  - 每个卷都是一个能够被单独格式化和使用的逻辑单元，供文件系统分配空间时使用。
  - 一个卷包含了文件系统信息(目录、FAT表)、一组文件以及空闲空间。

8.1.3 FAT和NTFS技术

❖ 2、FAT12

* 每个FAT表项占12位，即FAT12表最多有2^{12}=4096个表项。
* 1> 以盘块为基本分配单位
  - FAT的每个表项存放一个盘块号，FAT的实际表项数目由磁盘的物理块（扇区）数决定。
  - 若每个盘块（扇区）大小一般为512B，则每个磁盘分区容量最大为4096*512B=2MB；如果一个物理磁盘最多支持4个分区，则FAT12文件系统可管理的最大磁盘容量为8MB。

8.1.3 FAT和NTFS技术

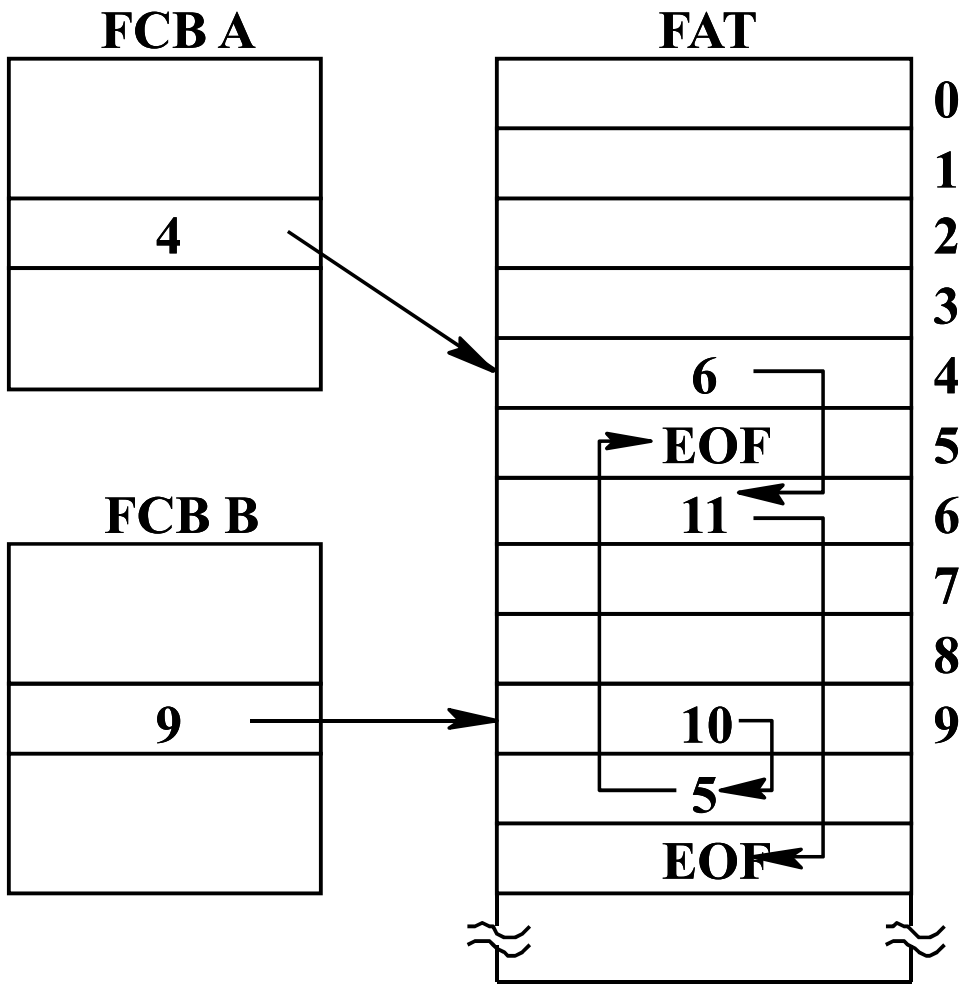
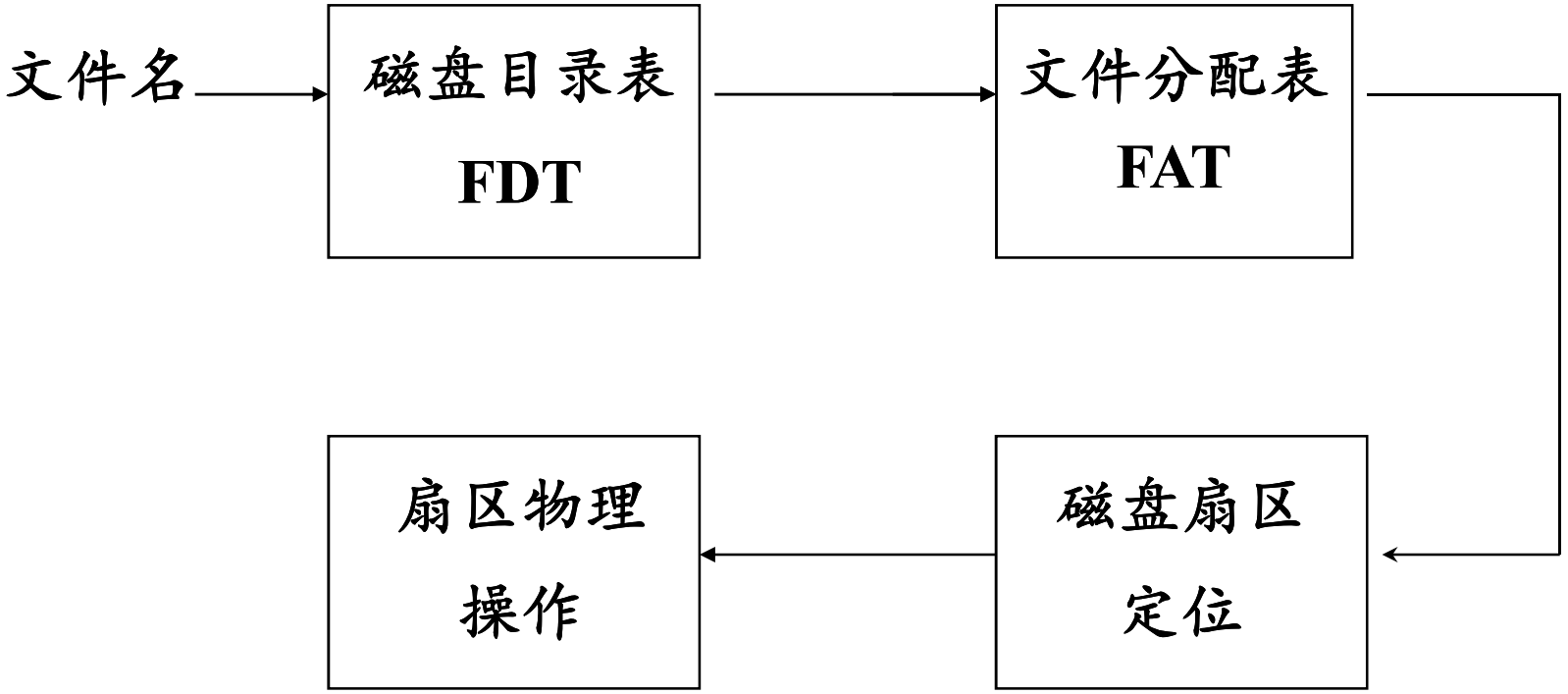


图 8-4 MS-DOS的文件物理结构

8.1.3 FAT和NTFS技术



补充 DOS磁盘访问操作流程

8.1.3 FAT和NTFS技术

* 2> 以簇为基本分配单位
  - FAT的每个表项存放一个簇号。
  - 簇一般为2n个盘块大小；在MS-DOS中，簇的容量可以有2个(1KB)、4个(2KB)或8个扇区(4KB)等；当簇包含8个盘块，磁盘分区最大容量为16MB。
  - 簇的优点：可减少FAT表中的项数（在相同磁盘容量下，FAT表中的项数与簇大小成反比），FAT表占更少存储空间，提高文件系统效率。
  - 簇的缺点：存在簇内碎片。
* 3> FAT12存在的问题
  - 磁盘容量存在严重限制，虽然可通过增加簇的大小来提高磁盘容量，但与之同时簇内碎片也成倍增加。
  - 只支持8+3格式文件名。

8.1.3 FAT和NTFS技术

3、FAT16

* 每个FAT表项占16位，FAT表最多有2^{16}=65536个表项。
* 簇的大小可为4、8、16、32或64个盘块数
* FAT16的分区最大容量＝ 2^{16}×64×512B＝2GB
* 缺点：FAT16对FAT12改善有限，磁盘容量依旧受限严重，且只支持8＋3格式文件名；Win95以后的系统对FAT16进行了扩展，文件名长度可达255个字符。

8.1.3 FAT和NTFS技术

4、FAT32

* 每个FAT表项占32位，FAT表最多可有2^{32}个表项。
* 簇的大小固定为8个盘块，即4KB
* FAT32的分区最大容量＝ $2^{32}\times8\times512B=2TB$
* 优点：支持更多磁盘容量；簇内碎片小，空间利用率高；支持长文件名。
* 缺点：FAT表比较大，运行速度较FAT16格式慢；单个文件大小不能大于4GB(可采用exFAT解决这一问题，是微软公司对FAT32的一种扩展文件分配表)；分区最小容量不得小于512MB（不得少于65537个簇）；不能向下兼容。

8.1.3 FAT和NTFS技术

❖ 5、NTFS（New Technology File System）

* 1> NTFS新特征

- 使用了64位磁盘地址，磁盘分区理论最大容量2^{64}个簇。
- 支持长文件名（单个文件名<=255个字符，全路径名<= 32767个字符）。
- 具有系统容错功能。
- 提供了数据一致性控制。
- 提供了文件加密、文件压缩功能。

8.1.3 FAT和NTFS技术

* 2> NTFS磁盘组织
  - NTFS以簇作为磁盘空间分配和回收的基本单位。
  - 卷因子：卷上的簇的大小称为“卷因子”。卷因子在磁盘格式化时确定，可以为512B、1KB、2KB、…、64KB，大多情况下使用4KB大小。
  - 逻辑簇号LCN：将整个卷的所有簇按顺序编号。
  - 虚拟簇号VCN：以文件为单位，将属于某个文件的簇按顺序从头到尾进行编号，以便于引用文件内的数据。
  - 簇的定位（VCN、LCN地址映射）
    - 文件数据的LCN=文件开始簇号+ VCN
    - 物理磁盘地址=卷首地址+LCN*卷因子

8.1.3 FAT和NTFS技术

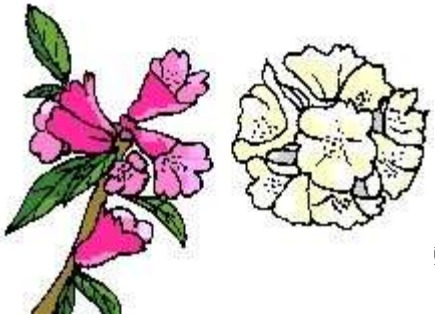
* 3> NTFS文件组织
  - NTFS以卷为单位，将一个卷中所有文件信息、目录信息以及未分配空间信息以文件记录（1个文件1条记录，记录大小为1KB）形式统一存储在一张主控文件表MFT中。
  - 如果文件很小，将其直接存储在MFT的记录中；如果文件较大，则将文件的真正数据存放在卷中其它可用簇中。
  - 文件在存储过程中，数据往往连续存放在若干相邻的簇中，仅用一个指针记录这几个相邻的簇即可。

8.1.3 FAT和NTFS技术

* **4> NTFS的缺点**
  - **NTFS格式只能被Win NT以上版本识别。**
  - **兼容性差。NTFS文件系统可以存取FAT格式文件，但NTFS格式文件却不能被FAT文件系统存取。**

8.1 外存组织方式

- 8.1.1 连续组织
- 8.1.2 链接组织
- 8.1.3 FAT和NTFS技术
- 8.1.4 索引组织



8.1.4 索引组织

❖ 1、单级索引组织

* 链接组织问题

- 不能高效地直接存取
- FAT需占较大的内存空间

* 索引组织

- 为每个文件分配一个索引块（文件较大时，需多个索引块），把分配给该文件的所有盘块号都记录在该索引块中；并在文件的目录项中填上指向该索引块的指针。
- 优点
  - 支持直接存取访问；不会产生外部碎片

8.1.4 索引组织

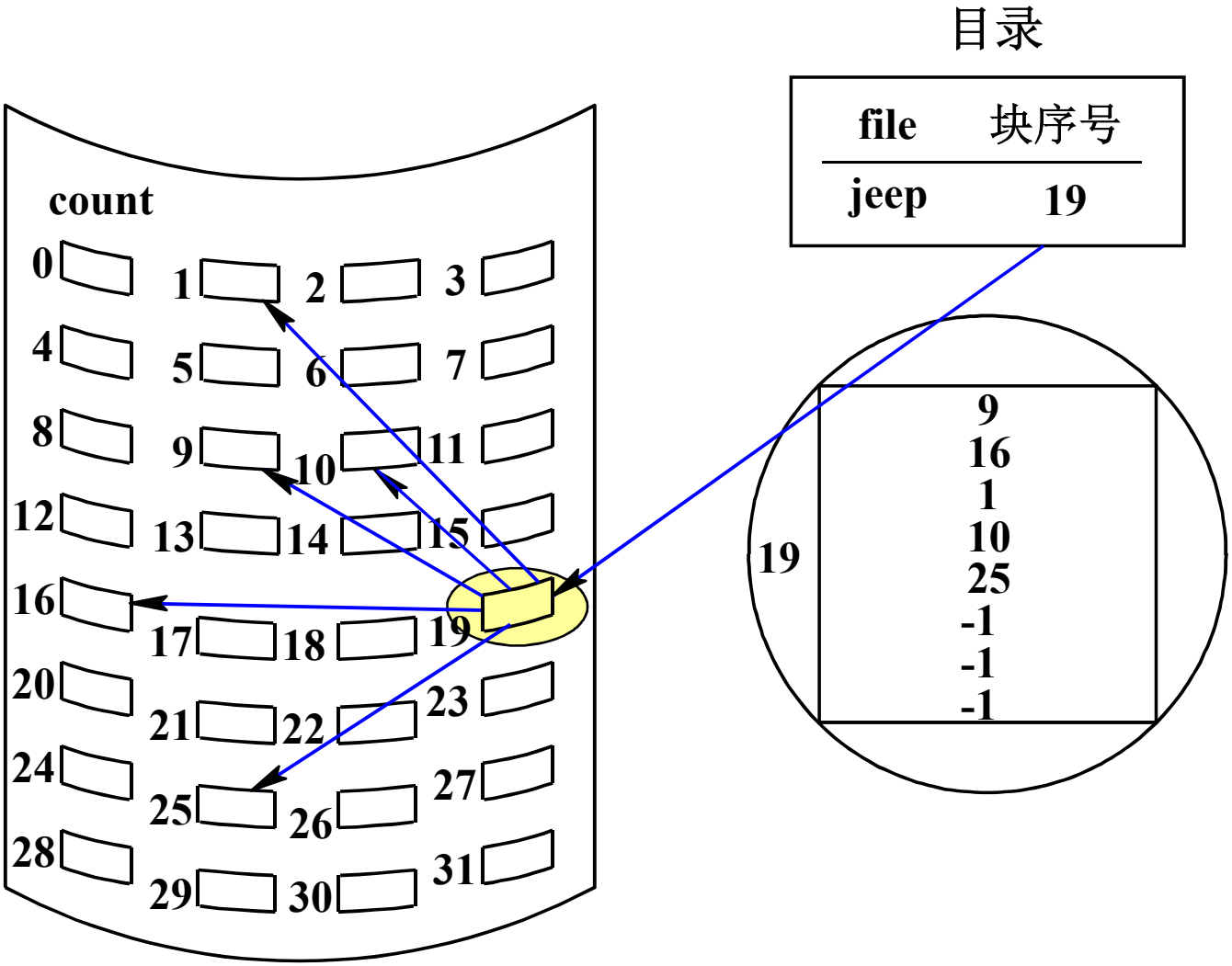


图8-6 单级索引组织方式

8.1.4 索引组织

* 问题
  - 文件较小时外存空间浪费严重（同样需为小文件建索引块，索引块利用率低）
  - 当文件较大时，索引块太多，查找速度减慢
    - 解决办法：建立多级索引

8.1.4 索引组织

❖ 2、多级索引组织

* 当文件较大、索引块较多时，可为索引块建立多级索引来加快存取访问速度。
* 【例3】设一个盘块大小为4kB，每个盘块号占4B，则一个索引块可存放1024个盘块号，如采用单级索引则最大允许文件长度（大小）为4MB；如采用二级索引，则存放的文件盘块号总数可达1024*1024=1M，最大允许文件长度（大小）为4GB 。

8.1.4 索引组织

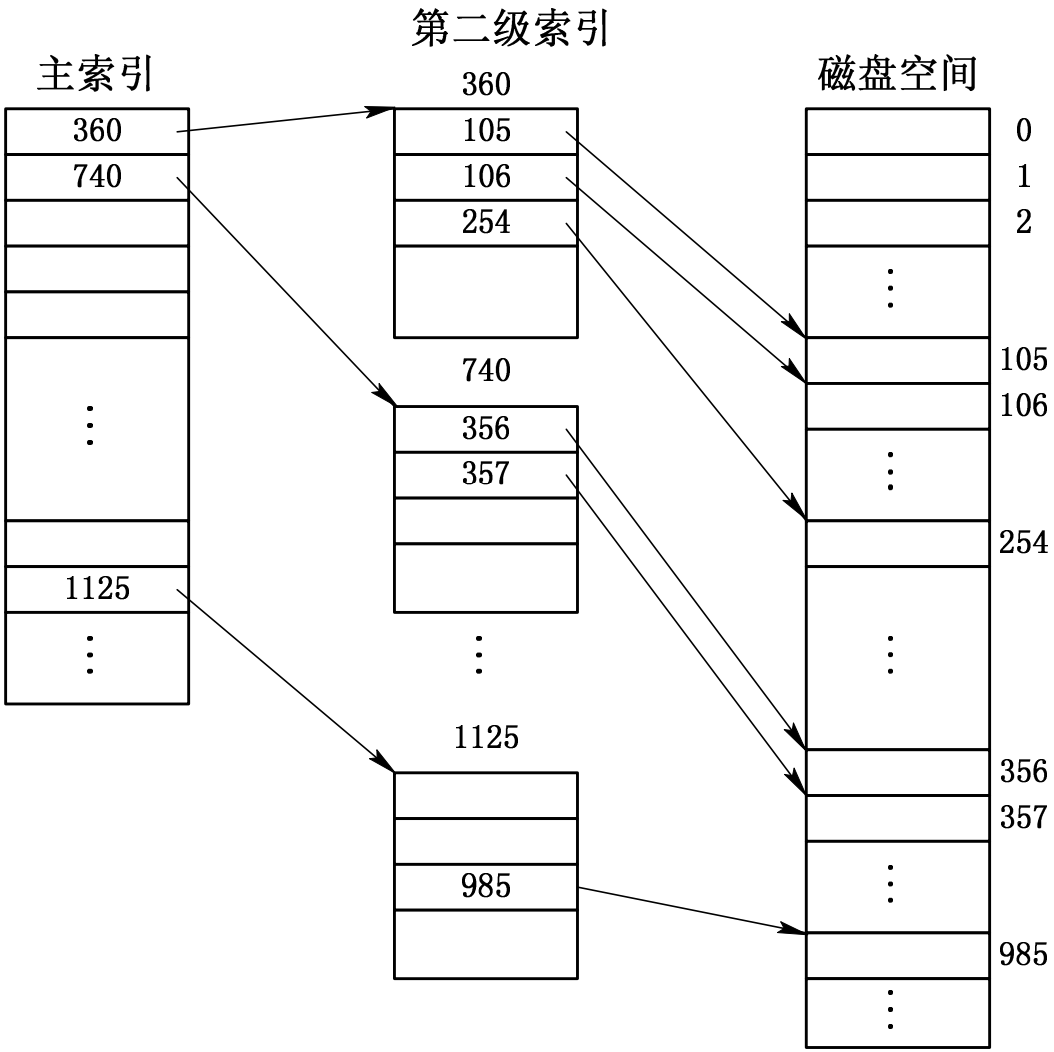


图8-7 两级索引组织

8.1.4 索引组织

❖ 3、混合索引组织

最大允许文件长度40KB+4MB+4GB+4TB

* Unix系统采用，其索引结点中可设置13个地址项：

✓ 直接地址（显式链接组织方式）：iaddr(0)-iaddr(9)直接存放文件数据所在盘块的盘块号。如盘块大小为4KB，则最大允许文件长度为40KB。

✓ 一次间接地址（单级索引组织方式）：iaddr(10)存放索引块盘块号，文件数据所在盘块的盘块号存储在索引块中，最大允许文件长度为4MB。

✓ 二次间接地址（二级索引组织方式）：iaddr(11)存放二级索引的主索引块盘块号，最大允许文件长度为4GB。

✓ 三次间接地址（三级索引组织方式）：iaddr(12)存放三级索引的主索引块盘块号，最大允许文件长度为4TB。

8.1.4 索引组织

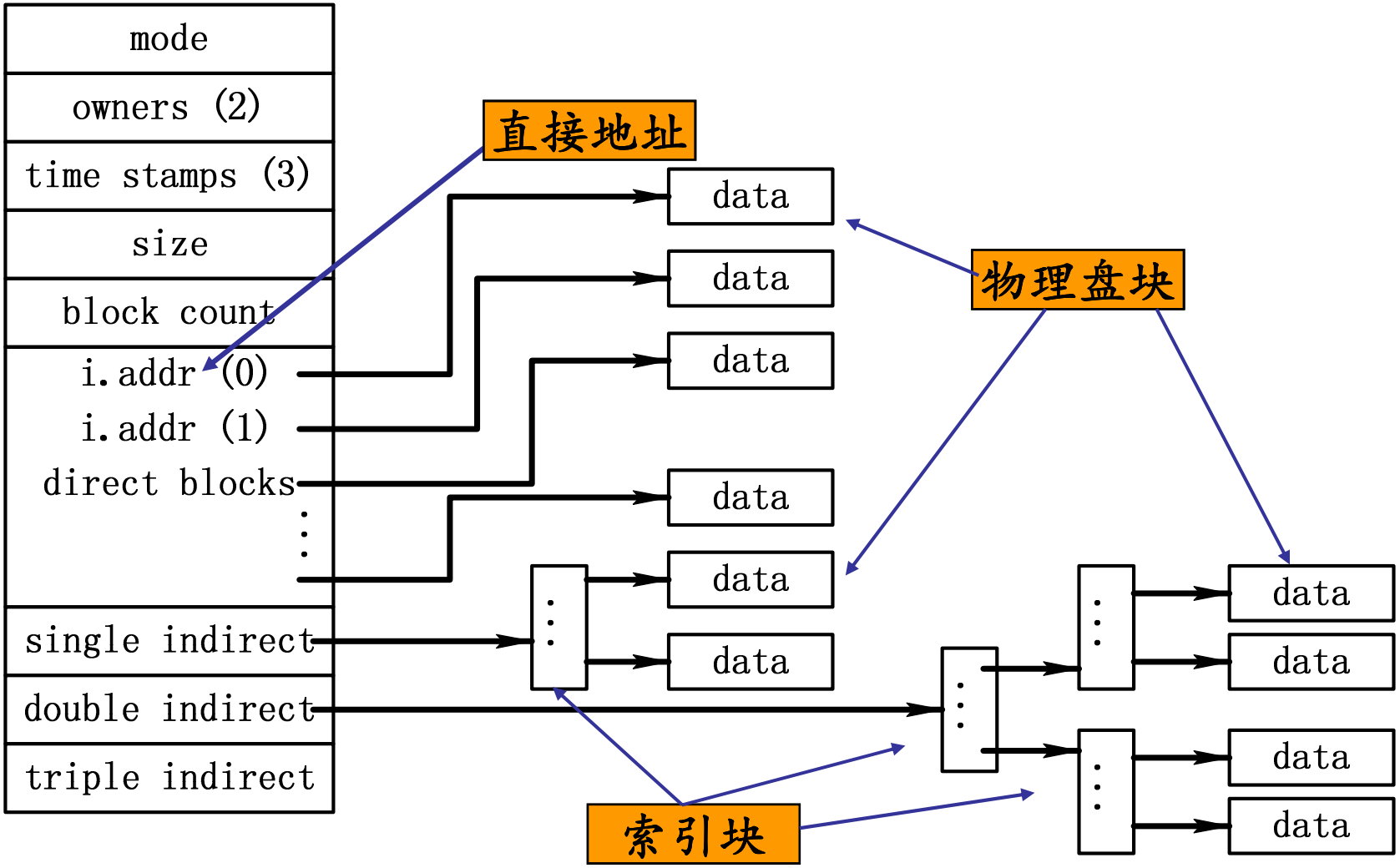


图8-8 混合索引索引组织

8.1 外存组织方式

- 【补充示例1】请分别解释在连续组织方式、隐式链接组织方式、显式链接组织方式和索引组织方式中如何将文件的字节偏移量3500转换为物理块号和块内偏移量（设盘块大小为1KB，盘块号需占4个字节）。

- 【解】3500/1024得商为3，余数为428，则逻辑块号为3，块内偏移量为428。
  - 1> 连续组织：从文件的目录项中得到起始物理盘块号，例如a0，则所求的物理盘块号为a0+3，块内偏移量为428。
  - 2> 隐式链接组织：由于每块需留4个字节存放下一个盘块号，因此逻辑块号为3500/1020的商3，块内偏移为440。从文件的目录项中可得该文件的首个（即第0个）盘块的块号，如b0；然后可从b0块得到第1个盘块号，如b1；再从b1得到第2个盘块号，如b2；从b2得到第3个盘块号，如b3，则所求物理盘块号为b3，块内偏移量为440。

8.1 外存组织方式

* 3> 显式链接组织：从文件的目录项中可得该文件的首个（即第0个）盘块的块号，如c0；然后从FAT的第c0项中得到分配给文件的第1个盘块的块号，如c1；再从FAT的第c1项中得到分配给文件的第2个盘块的块号，如c2；从FAT的第c2项中得到分配给文件的第3个盘块的块号，如c3；如此可得所求物理盘块号c3，块内偏移量为428。

* 4> 索引组织：从文件的目录项中可得该文件的索引块的地址；再从索引块的第3项（距离索引块首字节12字节的位置）可获得字节偏移量3500对应的物理块号，而块内偏移为428。

本章主要内容

- 8.1 外存组织（分配）方式
- 8.2 文件存储空间的管理
- 8.3 提高磁盘I/O速度的途径
- 8.4 提高磁盘可靠性的技术
- 8.5 数据一致性控制

8.2 文件存储空间的管理

文件存储空间管理的任务是为每个文件分配必要的外存空间，提高外存的利用率，并能有助于提高文件系统的工作速度。

由于文件存储设备是以块为单位进行管理的，因此，文件存储空间的管理实质上是空闲块的组织和管理问题，它包括空闲块的组织、分配与回收。

- 8.2.1 空闲表法和空闲链表法
- 8.2.2 位示图法
- 8.2.3 成组链接法

8.2.1 空闲表法和空闲链表法

❖ 1、空闲表法

* 为外存上的所有空闲区建立一张空闲表，每个空闲区对应于一个空闲表项，将所有空闲区按起始盘块号递增的顺序排列。
* 空闲表法用于外存的连续分配，与内存的动态分区分配相似，为每个文件分配一块连续的存储空间。
* 分配：首次适应算法、循环首次适应算法等。
* 回收：与邻接盘区进行合并。
* 优点：连续分配速度较快，对对换空间及小文件的外存分配适用。

8.2.1 空闲表法和空闲链表法

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

图8-9 空闲盘块表

8.2.1 空闲表法和空闲链表法

❖ 2、空闲链表法

* 1>空闲盘块链

- 概念：将磁盘上的所有空闲空间以盘块为单位拉成一条链。
- 分配：用户请求分配时，从链首开始依次摘下适当数目的空闲盘块分配给用户。
- 回收：放回至空闲盘块链末尾。
- 优点：分配和回收简单。
- 缺点：为一个文件分配盘块时可能需要重复操作多次，链表可能很长。

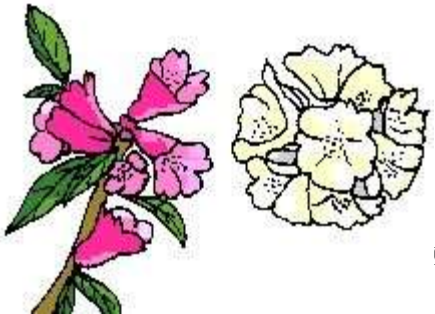
8.2.1 空闲表法和空闲链表法

* 2> 空闲盘区链

  - 概念：将磁盘上的所有空闲盘区(每区可有若干个连续盘块)拉成一条链。每个盘区上除含有用于指示下一个空闲盘区的指针外，还应有能指明本盘区大小的信息；也可采用显式链接方法在内存中设置一张空闲盘区链接表。
  - 分配：通常采用首次适应算法，根据用户的请求选择足够大的盘区，分割后分配给它。
  - 回收：与邻接盘区进行合并。
  - 优点：链表较短。
  - 缺点：分配回收复杂。

8.2 文件存储空间的管理

- 8.2.1 空闲表法和空闲链表法
- 8.2.2 位示图法
- 8.2.3 成组链接法



8.2.2 位示图法

❖ 1、位示图

* 概念：用二进制的一位来表示磁盘中一个盘块的使用情况，“0”表示盘块空闲，“1”表示盘块已分配，所有盘块对应的二进制位构成的一个集合称为位示图，通常可用m*n个位数来构成位示图，并使m*n等于磁盘总块数。

* 【例4】假定磁盘的块长为1KB，对于200MB的磁盘需有200K位来映射，即需要200×1024/8=25K字节，即需25个物理块来构成一个位示图。

8.2.2 位示图法

	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																

图8-10 位示图

8.2.2 位示图法

❖ 2、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(当“0”表示盘块空闲时)；
* 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，若每行共有n位，则其相应的盘块号b应按下式计算：

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

* 3> 修改位示图，令map［i，j］=1 。

8.2.2 位示图法

❖ 3、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：

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

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

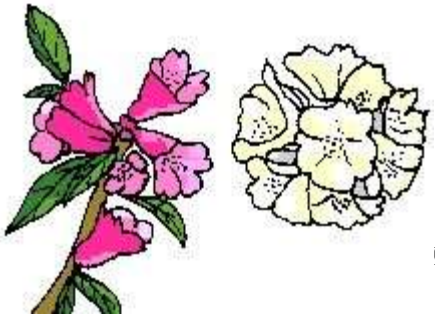
* 2> 修改位示图，令map［i，j］=0。

❖ 4、位示图法优缺点

* 简单，特别适于连续分配；不占空间，可放入内存，易于访问

8.2 文件存储空间的管理

- 8.2.1 空闲表法和空闲链表法
- 8.2.2 位示图法
- 8.2.3 成组链接法



8.2.3 成组链接法

成组链接法（Unix使用）是为了克服大型文件系统中空闲表法和空闲链表法的空闲表、空闲链表太长这一缺陷，将这两种方法结合而形成的一种空闲盘块管理办法。

❖ 1、空闲盘块的组织

* ① 将磁盘文件区中的所有空闲盘块分成若干个组；
* ② 设置一个空闲盘块号栈（临界资源），用来存放当前可用的一组空闲盘块号以及栈中尚有的空闲盘块总数N，N兼作栈顶指针用；
* ③ 将其余各组含有的空闲盘块总数和该组所有空闲盘块号记入前一组的最后一个盘块中；
* ④ 最后一组盘块的空闲盘块总数和空闲盘块号在记入时，需在最末一个盘块号后添加一个0，用作空闲盘块链结束标志。

8.2.3 成组链接法

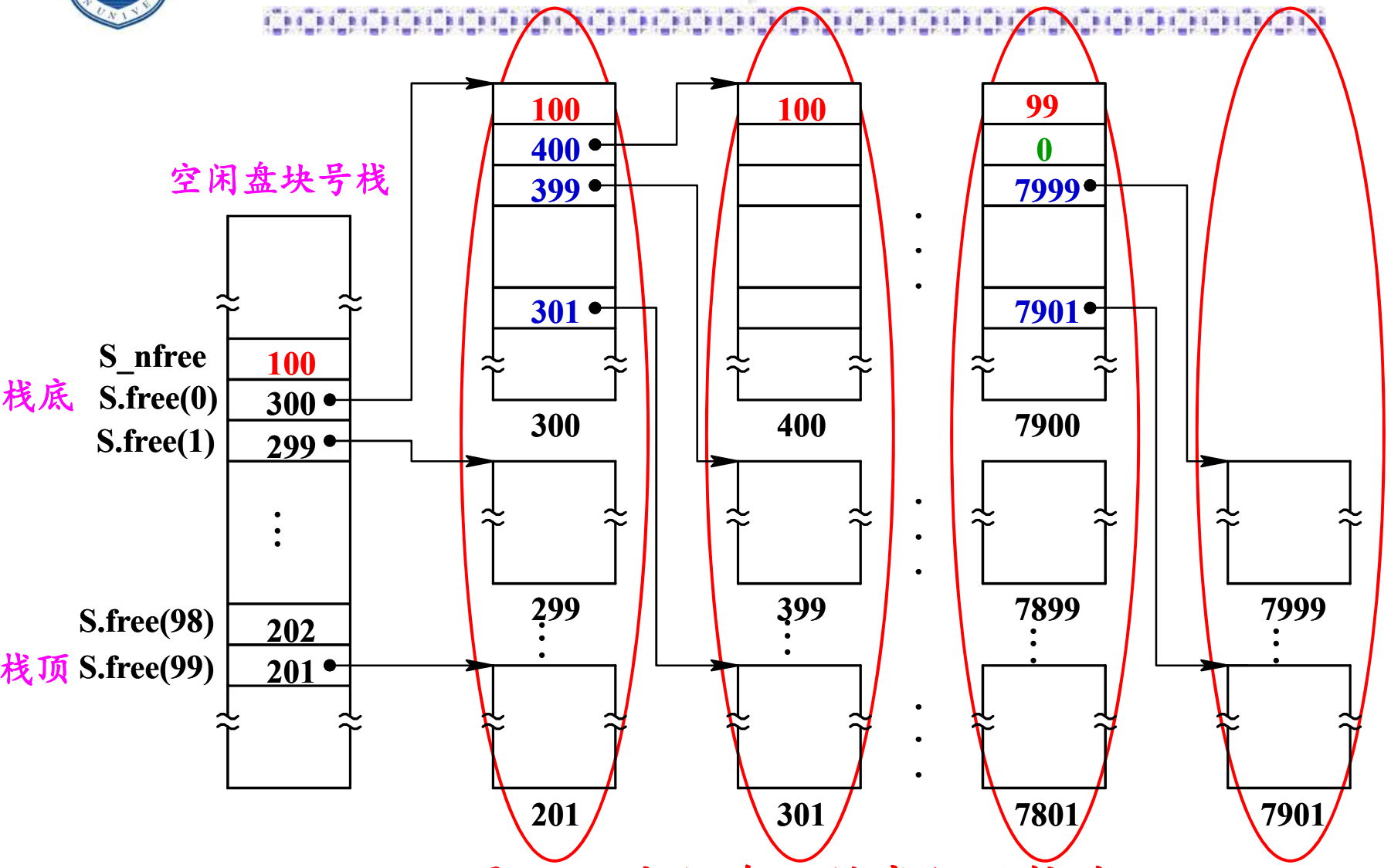


图8-11 空闲盘块的成组链接法

8.2.3 成组链接法

❖ 2、空闲盘块的分配

* 1> 检查空闲盘块号栈是否上锁，如未上锁，便从栈顶取出一空闲盘块号，将与之对应的盘块分配给用户，令N=N-1，将栈顶指针下移一格；
* 2> 若该盘块号不为0且位于栈底，调用磁盘读过程，将栈底盘块号所对应盘块的内容读入栈中，作为新的盘块号栈的内容，并把原栈底对应的盘块分配出去；如用户要求继续分配，返回第1>步；
* 3> 若该盘块号为0且位于栈底，则表示系统已无空闲盘块可分配，将该进程阻塞等待其它进程释放盘块。

8.2.3 成组链接法

❖ 3、空闲盘块的回收

* 1> 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。
* 2> 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。

8.2.3 成组链接法

- 【补充示例2】某个系统采用成组链接法来管理磁盘空闲空间，目前磁盘的状态如下图示：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，系统要删除另一个文件，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。

8.2.3 成组链接法

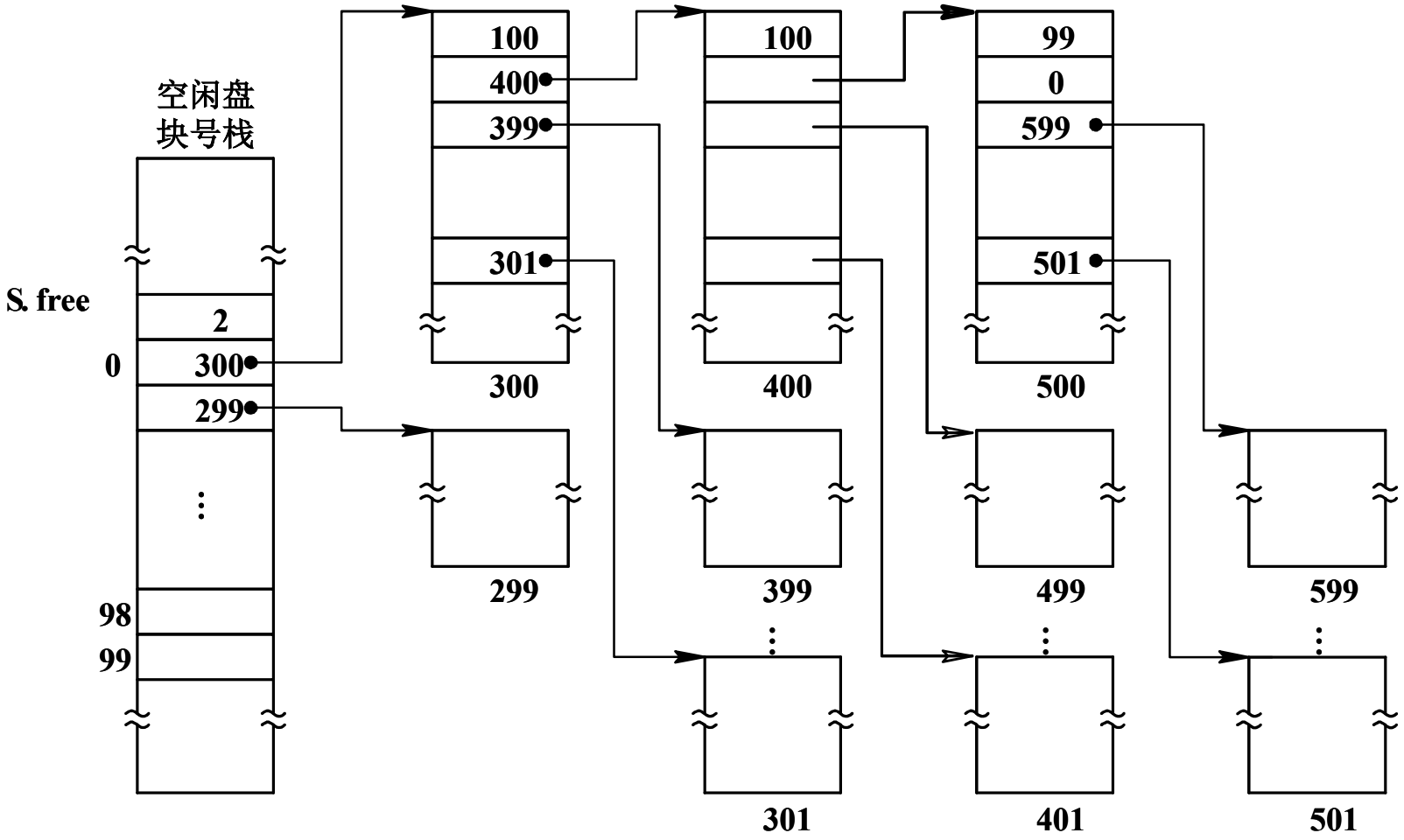
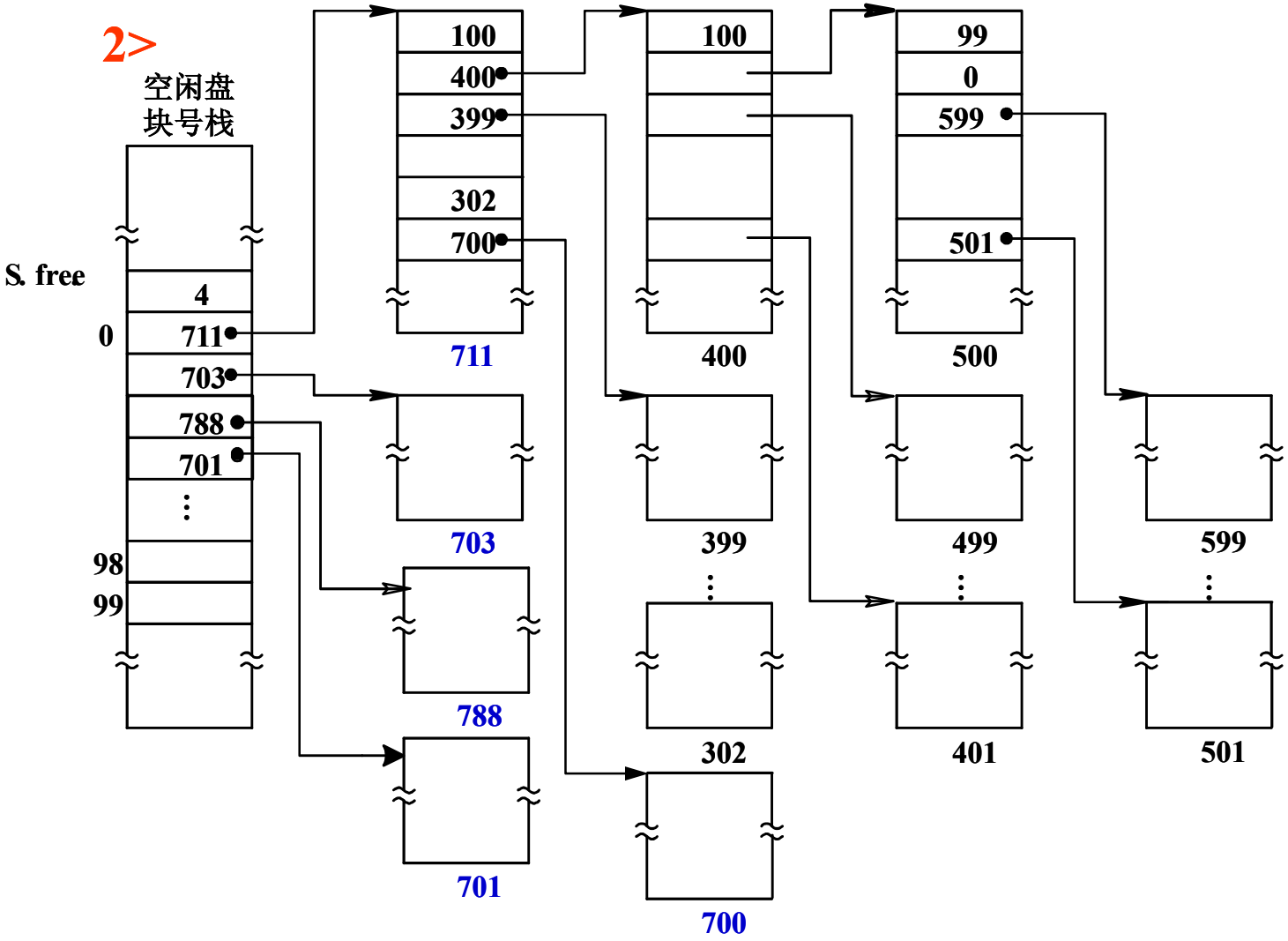


图 补充示例2

8.2.3 成组链接法

❖ 【解】 1> 空闲盘块数=2+200+99=301块



本章主要内容

- 8.1 外存组织（分配）方式
- 8.2 文件存储空间的管理
- 8.3 提高磁盘I/O速度的途径
- 8.4 提高磁盘可靠性的技术
- 8.5 数据一致性控制

8.3.1 磁盘高速缓存

❖ 1、磁盘高速缓存的形式

* 概念

- 磁盘高速缓存是利用内存中的存储空间，来暂存从磁盘中读出的一系列盘块中的信息。
- 磁盘高速缓存是一组在逻辑上属于磁盘，而物理上是驻留在内存中的盘块。

* 磁盘高速缓存的两种形式

- 1> 固定大小形式：在内存中单独开辟一个大小是固定的存储空间来作为磁盘高速缓存。
- 2> 可变大小形式：把所有未利用的内存空间变为一个缓冲池，供请求分页系统和磁盘I/O时(作为磁盘高速缓存)共享。

8.3.1 磁盘高速缓存

* 磁盘高速缓存时需考虑的问题
  - 1> 如何将缓存中的数据传送给请求进程?
  - 2> 缓存满时采用什么样的置换策略?
  - 3> 已修改的盘块数据在何时被写回磁盘?

8.3.1 磁盘高速缓存

❖ 2、数据交付（传送）方式

* 概念：数据交付(Data Delivery)是指将磁盘高速缓存中的数据传送给请求者进程。
* 步骤：先查数据是否在缓存中，如不在再查磁盘并更新缓存。
* 数据交付的两种方式
  - 1> 数据交付方式：直接将高速缓存中的数据，传送到请求者进程的内存工作区中。
  - 2> 指针交付方式：将数据所在高速缓存区域的地址指针交付给请求者进程。
  - 指针交付方式所传数据量较数据交付方式少很多，因此更具效率。

8.3.1 磁盘高速缓存

❖ 3、缓存置换算法

* 同请求分页存储管理方式类似，磁盘数据在写入缓存时同样面临缓存置换问题，可借鉴页面置换算法思想，同时考虑缓存访问的自身特性来设计缓存置换算法，设计时需考虑：
  - 1> 最近最久未使用
  - 2> 访问频率
  - 3> 可预见性
  - 4> 数据的一致性
    - 将需要考虑一致性的数据块放在替换队列的头部，优先回写。

8.3.1 磁盘高速缓存

❖ 4、周期性写回磁盘

* 在LRU算法中，经常被访问的盘块数据可能一直保留在高速缓存中，长期不被写回磁盘(有可能丢数据)。
* 在UNIX系统中专门增设了一个修改(update)程序，使之在后台运行，该程序周期性地（每隔30S）调用一个系统调用SYNC，强制性地将所有在高速缓存中已修改的盘块数据写回磁盘。
* 在MS-DOS中所采用的方法是：只要高速缓存中的某盘块数据被修改，便立即将它写回磁盘，并将这种高速缓存称为“写穿透高速缓存”(write-through cache)。

8.3.2 提高磁盘I/O速度的其它办法

❖ 1、提高磁盘I/O速度的其它办法

* **1> 提前读(Read-Ahead)**
  - 在读当前块的同时，将下一盘块读入缓冲区
* **2> 延迟写**
  - 缓冲区中的数据不立即写回磁盘，而挂在空闲缓冲区队列的末尾
* **3> 优化物理块分布**
  - 使文件的物理块集中，以减小磁头移动距离
* **4> 虚拟盘**
  - 利用内存空间仿真磁盘，又称为RAM盘

8.3.2 提高磁盘I/O速度的其它办法

❖ 2、虚拟盘与磁盘缓存的区别

* 1> 虚拟盘存储的内容完全由用户控制(用户拿它当磁盘使用)
* 2> 高速磁盘缓存存储的内容由操作系统OS控制(用户不知道其存在)

8.3.3 廉价磁盘冗余阵列（RAID）

❖ 1、并行交叉存取

* 将一个盘块中的数据分成若干个子盘块数据，分别存储在不同磁盘的相同位置上，数据传送时采用并行传输方式，磁盘I/O速度提高N-1倍。

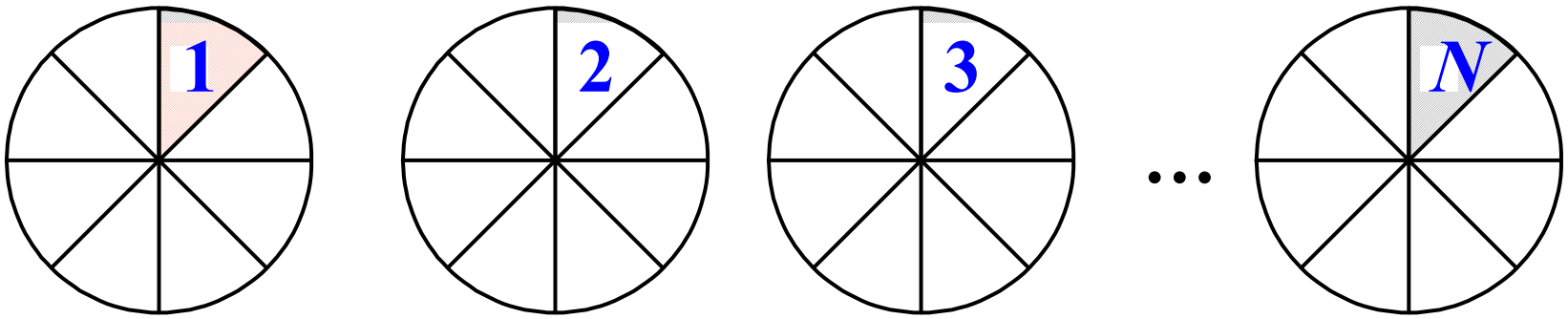


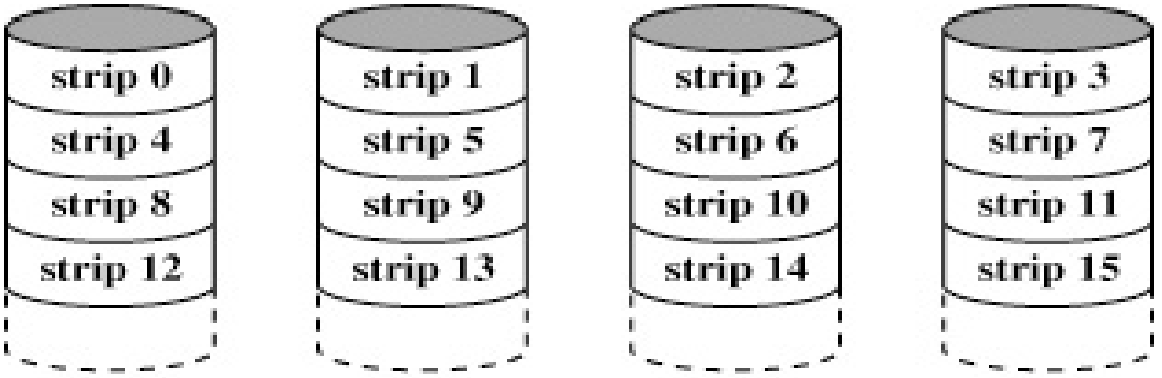
图8-12 磁盘并行交叉存取方式

8.3.3 廉价磁盘冗余阵列

❖ 2、Raid的分级

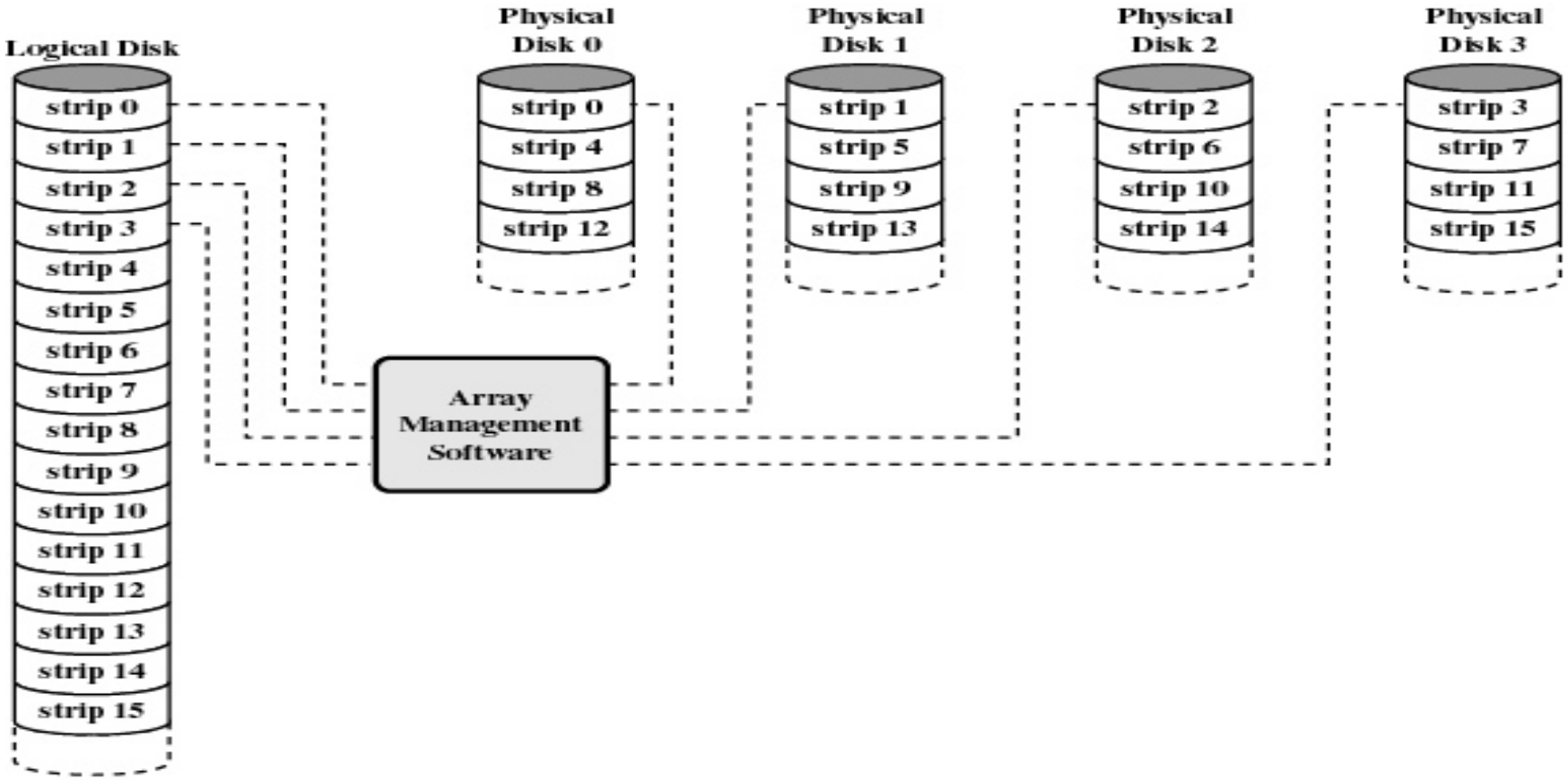
* 1> Raid 0

- 仅提供并行交叉存取；不冗余；不校验；低可靠性；低价格；并行 I/O 访问，速度快。



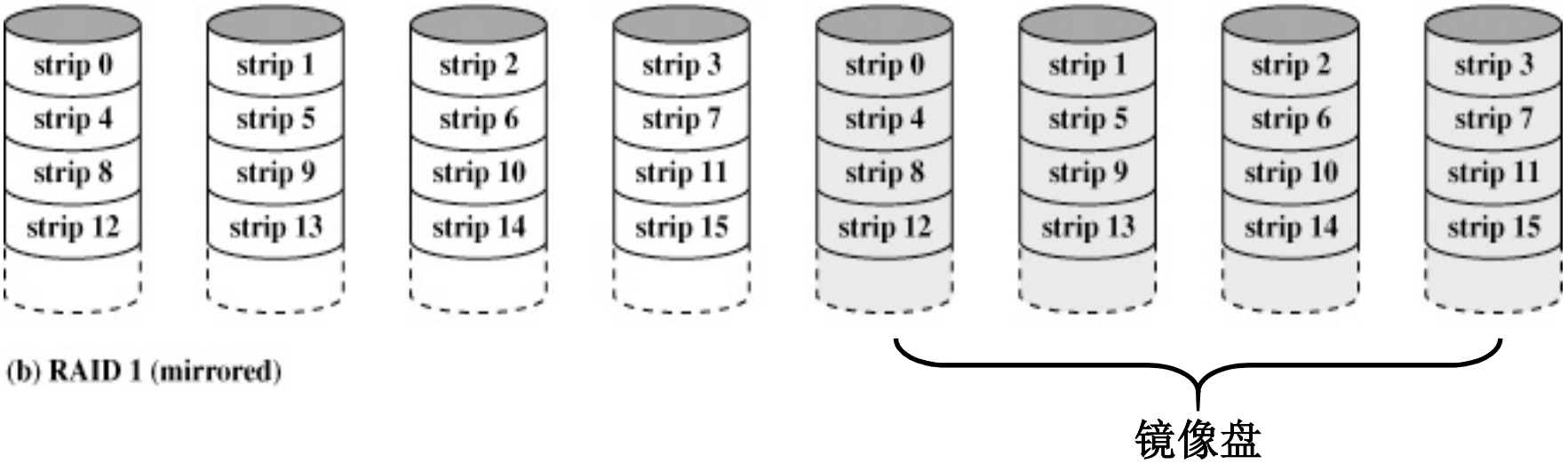
(a) RAID 0 (non-redundant)

8.3.3 廉价磁盘冗余阵列



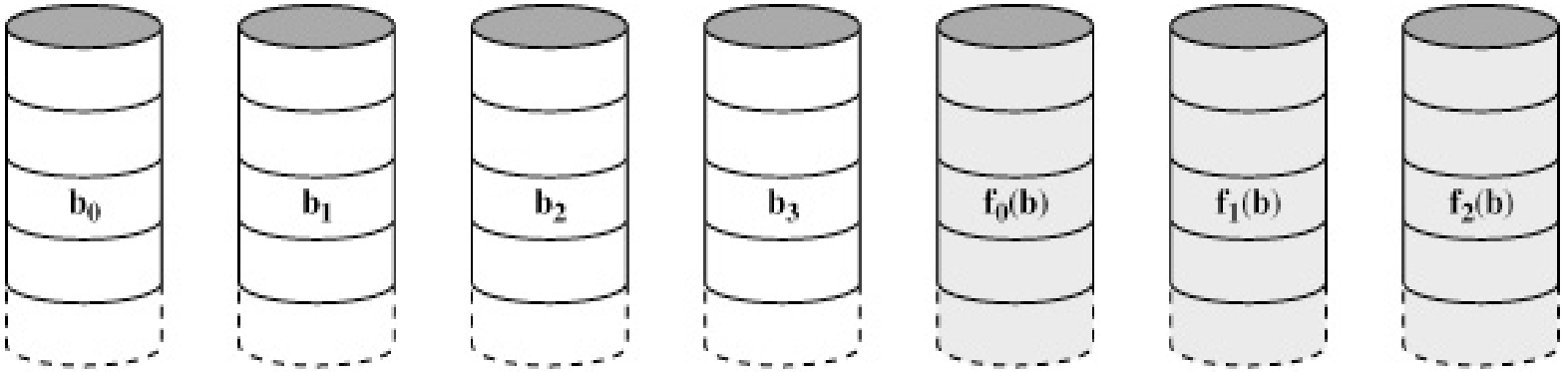
8.3.3 廉价磁盘冗余阵列

* **2> Raid 1**
  - 分布存放；镜像冗余；不校验；读性能比RAID 0好 (选择寻道时间小的磁盘访问)、写性能比 RAID 0差；存储开销大；可靠性高



8.3.3 廉价磁盘冗余阵列

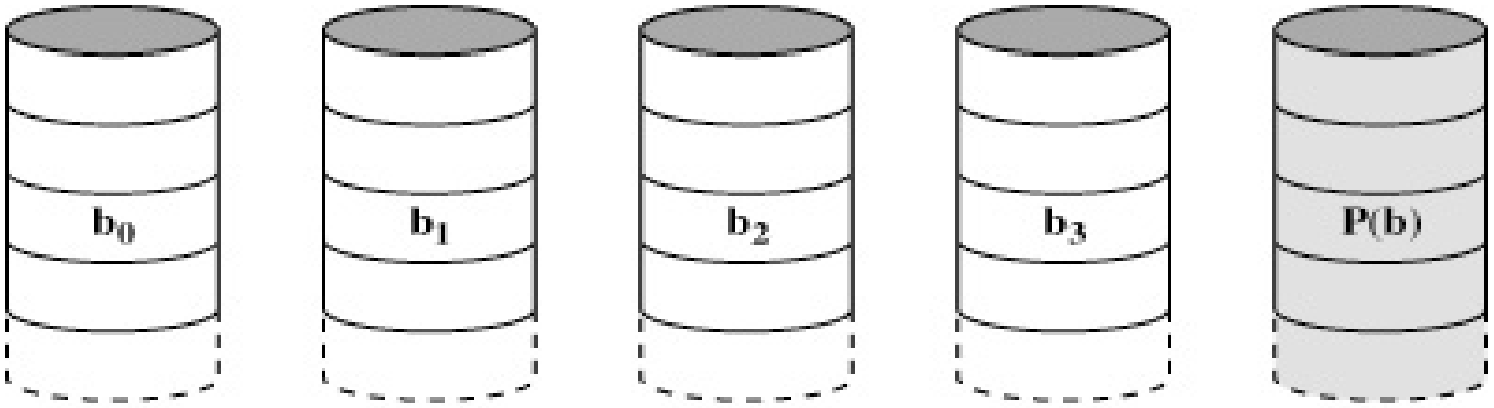
* **3> Raid 2**
  - 汉明码校验冗余



(c) RAID 2 (redundancy through Hamming code)

8.3.3 廉价磁盘冗余阵列

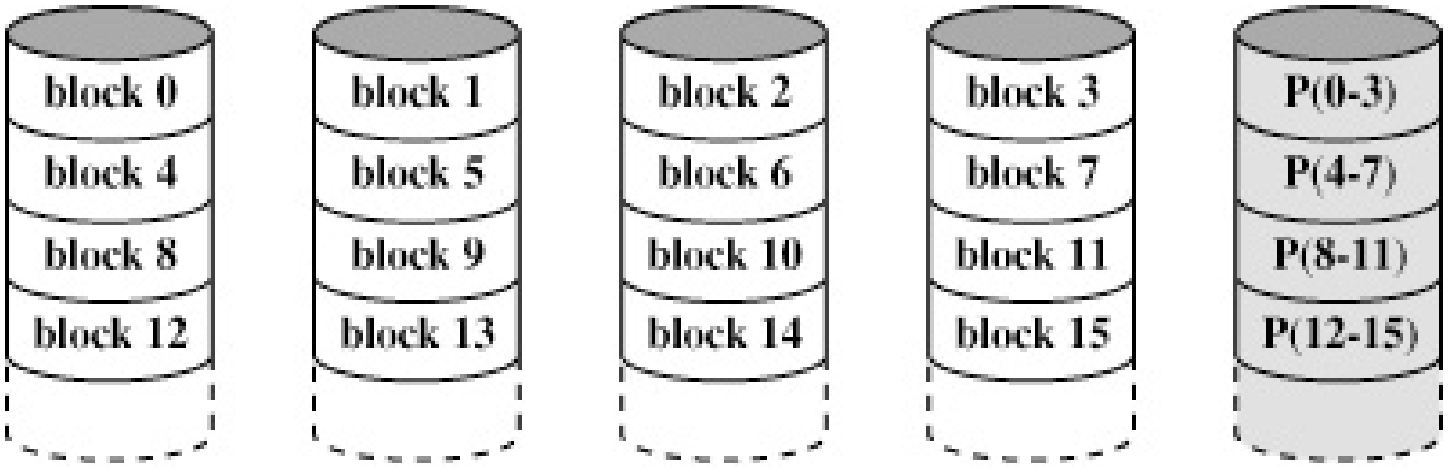
* **4> Raid 3**
  - 采用奇偶校验，用一个磁盘作为校验盘，其余的作为数据盘。校验盘易成“I/O瓶颈”。



(d) RAID 3 (bit-interleaved parity)

8.3.3 廉价磁盘冗余阵列

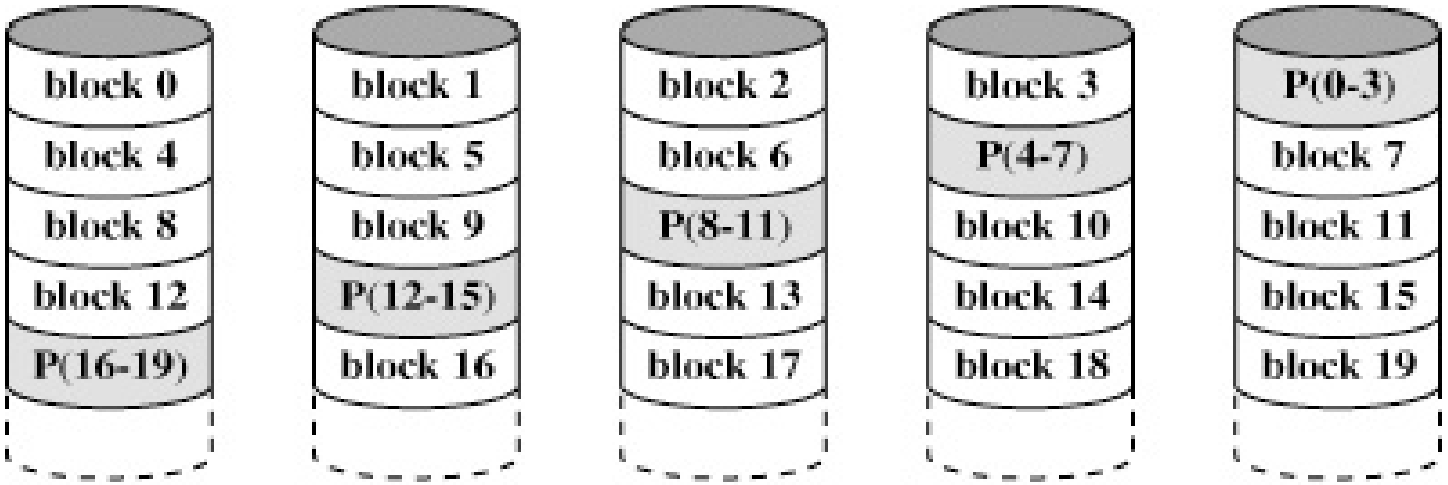
* **5> Raid 4**
  - 和RADI3相比较，RAID4基于大的块校验。



(e) RAID 4 (block-level parity)

8.3.3 廉价磁盘冗余阵列

* **6> Raid 5**
  - 奇偶校验，校验信息写在螺旋地写在所有数据盘上，解决了RAID4校验盘不可靠性问题。



(f) RAID 5 (block-level distributed parity)

8.3.3 廉价磁盘冗余阵列

* 7> Raid 6
  - 设置有专用的异步校验盘，该盘具有独立的数据通路，具有比Raid3 和Raid5更好的性能，但性能改进有限，且价格昂贵。
* 8> Raid 7
  - 是对Raid6的改进，该阵列中所有磁盘都具有较高的传输速率和优异的性能，是目前最高档次的磁盘阵列，但价格也较高。

8.3.3 廉价磁盘冗余阵列

❖ 3、Raid的优点

* **1> 可靠性高**
  - 采用容错技术
  - 可实现镜像、双工等冗余方式
* **2> 磁盘I/O速度高**
  - 采用并行交叉存取，速度快
* **3> 性能/价格比高**
  - 与其它高性能磁盘系统相比，容量、速度、可靠性高，但价格低

本章主要内容

- 8.1 外存组织（分配）方式
- 8.2 文件存储空间的管理
- 8.3 提高磁盘I/O速度的途径
- 8.4 提高磁盘可靠性的技术
- 8.5 数据一致性控制

8.4 提高磁盘可靠性的技术

❖ 1、文件保护

* 1> 通过存取控制机制来防止由人为因素所造成的文件不安全性。（第7章已授）
* 2> 通过磁盘容错技术来防止由系统因素特别是磁盘故障所造成的文件不安全性。
* 3> 通过“后备系统”来防止由自然因素所造成的不安全性。

8.4 提高磁盘可靠性的技术

❖ 2、磁盘容错技术（系统容错技术）

* 1> 第一级容错技术SFT-I

- 最基本的磁盘容错技术，主要用于防止因磁盘表面缺陷所造成的数据丢失。
- ① 双份目录和双份FAT
  - 主目录、主FAT；备份目录、备份FAT
- ② 热修复重定向
  - 热修复重定向：系统将磁盘容量的一部分作为热修复重定向区，用于存放磁盘有缺陷时的待写数据。
- ③ 写后读校验
  - 将内存缓冲区中的数据写入磁盘后，立即又将该数据从磁盘读出，再与内存缓冲区中的数据比较。

8.4 提高磁盘可靠性的技术

* 2> 第二级容错技术SFT-II
  - 防止磁盘驱动器和磁盘控制器故障导致的系统不能正常工作。
  - ① 磁盘镜像
    - 在磁盘控制器下再增设一个磁盘驱动器，每次向主磁盘写入数据后，再将数据写到备份磁盘上。
    - 缺点：磁盘I/O速度没有提高，磁盘利用率降至50%
  - ② 磁盘双工
    - 将两台磁盘驱动器接到两台磁盘控制器上，使两台磁盘机镜像成对。
    - 优点：可向磁盘并行读、写数据。

8.4 提高磁盘可靠性的技术

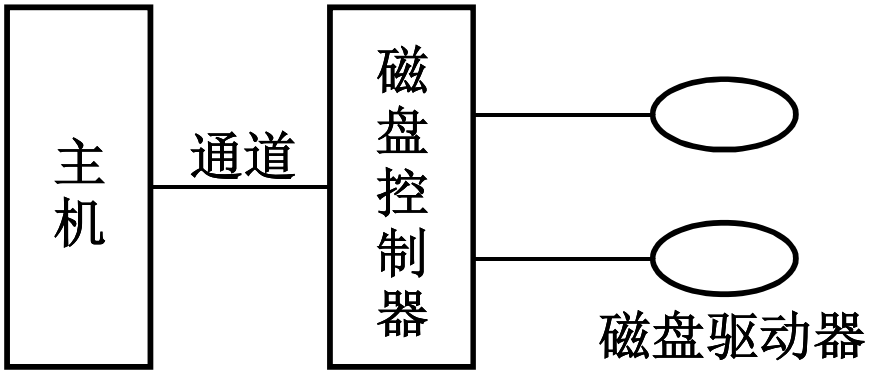


图8-13　磁盘镜像示意图

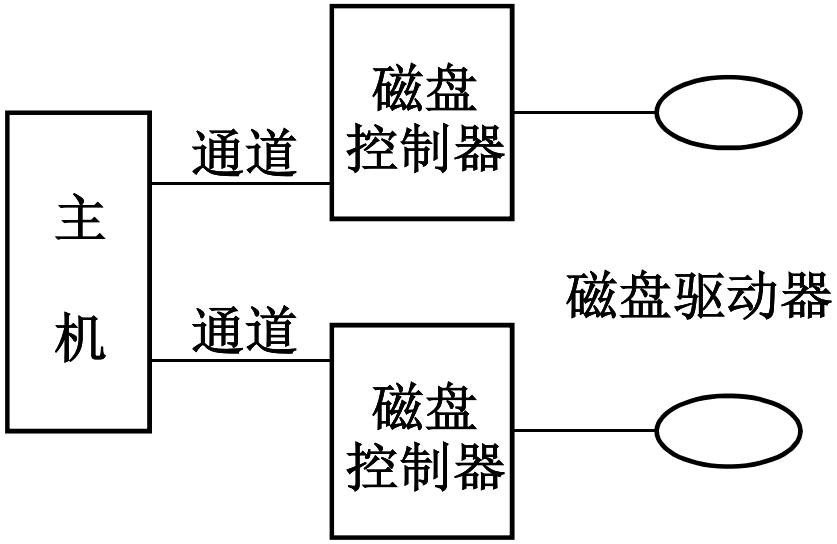


图8-14　磁盘双工示意图

8.4 提高磁盘可靠性的技术

* 3> 第三级容错技术——基于集群的容错技术

- 集群(Cluster)：是指由一组互连的自主计算机组成统一的计算机系统，使多台计算机能够像一台机器那样工作或者看起来好像一台机器。
- ① 双机热备份模式
  - 采用两台处理能力相同的服务器，一台作为主服务器，另一台作为备份服务器。一旦主服务器出现故障，备份服务器立即接替主服务器的工作而成为系统中的主服务器，修复后的服务器再作为备份服务器。
  - 优点：简单、易实现、易作，支持远程热备份。
  - 缺点：备份服务器处于被动等待状态，系统使用率只有50%。

8.4 提高磁盘可靠性的技术

图8-15 双机热备份模式

8.4 提高磁盘可靠性的技术

- ② 双机互为备份模式
  - 两台服务器均为在线服务器，各自完成自己的任务，同时互为对方进行数据备份；正常运行时，镜像盘对本地用户是锁死的；一旦一台服务器发生故障，则由正常服务器向故障服务器的客户机进行广播表明要切换；切换后，正常服务器除了完成自己的工作外，同时接替故障服务器的工作。
  - 优点：系统效率较高。
- ③ 公用磁盘模式
  - 多台计算机连接到一台公共的磁盘系统上，将公共磁盘划分成若干个卷，每台计算机使用一个卷。如果某台计算机发生了故障，选择另一台计算机接替并拥有故障机器的卷的控制权。

8.4 提高磁盘可靠性的技术

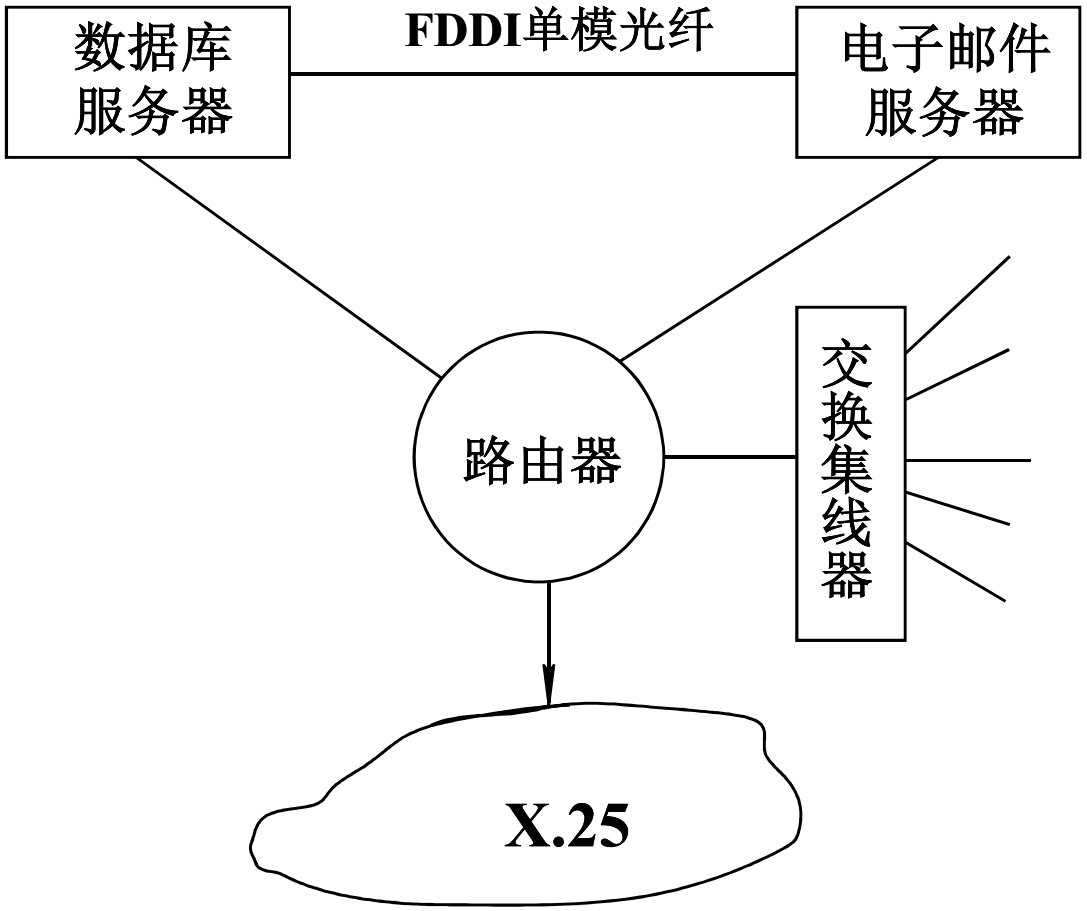


图8-16 双机互为备份系统的示意图

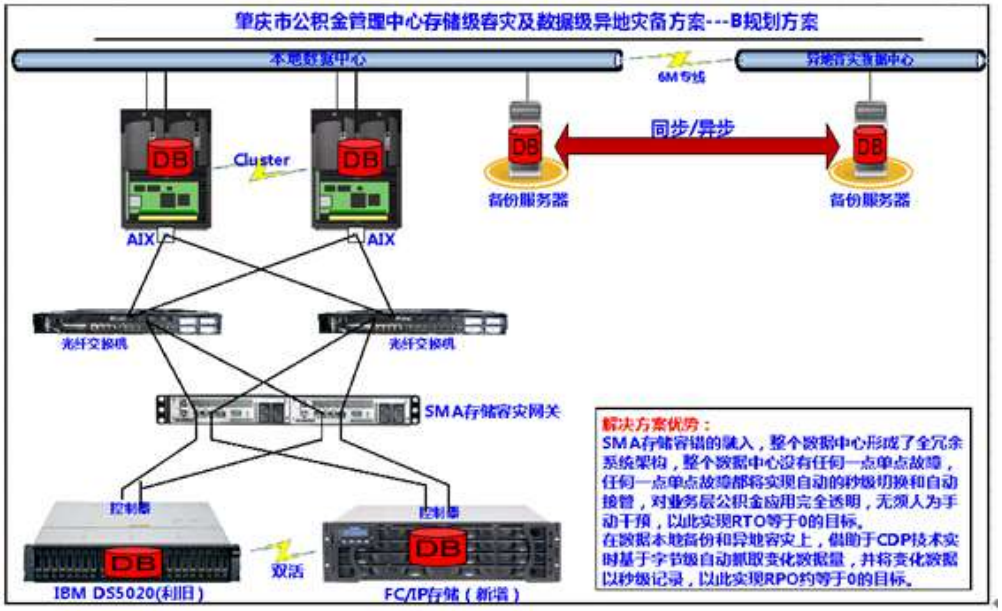
8.4 提高磁盘可靠性的技术

❖ 3、后备系统

* 1> 转存

- 将重要数据异步或同步备份到磁带机、硬盘、光盘驱动器。

* 2> 建立灾备中心

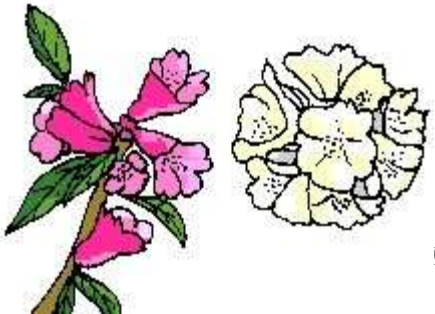


本章主要内容

- 8.1 外存组织（分配）方式
- 8.2 文件存储空间的管理
- 8.3 提高磁盘I/O速度的途径
- 8.4 提高磁盘可靠性的技术
- 8.5 数据一致性控制

8.5 数据一致性控制

- 8.5.1 事务
- 8.5.2 检查点
- 8.5.3 并发控制
- 8.5.4 重复数据的数据一致性问题



8.5.1 事务

❖ 事务

* 事务：用于访问和修改各种数据项的一个程序单位(通常包括若干个操作)，其操作具有原子性。
* 事务操作
  - 若当前事务所有操作都均执行成功，可执行commit operation来结束当前事务。
  - 若当前事务任一个操作失败，须执行abort operation来roll back当前事务。
* 事务记录：用来记录在事务运行时对数据项修改的全部信息，也称运行日志。
  - 事务记录的数据结构

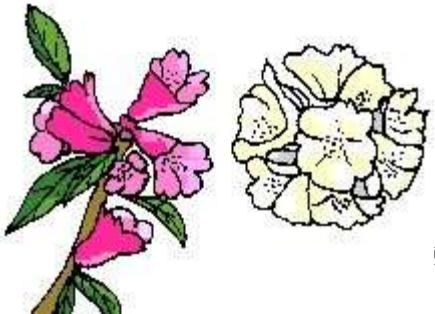
事务名	数据项名	旧值	新值

8.5.1 事务

* 恢复过程
  - undo<T_i>：将T_i修改过的数据恢复到修改前的值
  - redo<T_i>：将T_i修改过的数据设置为新值
* 恢复算法
  - 如果系统发生了错误，系统需对以前发生的事务进行处理：如果事务T_i在事务记录表中出现了〈T_i托付〉记录，表明该事务的各类操作已完成，则执行redo〈T_i〉操作；反之，如果在事务记录表中仅有〈T_i开始〉而没有出现〈T_i托付〉记录，表明该事务的各类操作尚未全部完成，则执行undo〈T_i〉操作。

8.5 数据一致性控制

- 8.5.1 事务
- 8.5.2 检查点
- 8.5.3 并发控制
- 8.5.4 重复数据的数据一致性问题



8.5.2 检查点

❖ 1、检查点的作用

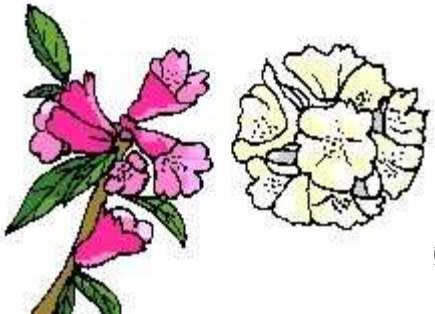
* 对事务记录表中事物记录的清理工作经常化，每隔一定时间便做一次清理工作。当发生故障后，只需对最后一个检查结点之后的事务记录进行处理，从而可以大大减少恢复处理的开销。

❖ 2、检查点恢复算法

* 系统如发生故障，恢复例程首先查找事务记录表，确定在最近检查点以前开始执行的最后的事务T_i，对其后的事务利用redo和undo过程对它们进行处理。

8.5 数据一致性控制

- 8.5.1 事务
- 8.5.2 检查点
- 8.5.3 并发控制
- 8.5.4 重复数据的数据一致性问题



8.5.3 并发控制

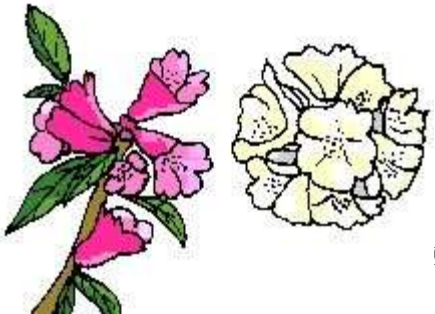
❖ 事务的并发控制

* 顺序性：由于事务操作的原子性，使得各事务对数据项的修改是互斥的，这种特性称为事务顺序性。
* 并发控制：实现事务顺序性的技术称为并发控制。
* 并发控制的实现
  - 1> 利用互斥锁实现事务顺序性
    - 对事务操作的对象上锁，简单易行，但效率不高
  - 2> 利用互斥锁和共享锁实现顺序性
    - 共享锁允许多个事务对相应对象执行读操作，互斥锁只允许一个事务进行写操作，效率提高了。

8.5 数据一致性控制

- 8.5.1 事务
- 8.5.2 检查点
- 8.5.3 并发控制
- 8.5.4 重复数据的数据一致性问题

是指主文件与备份文件的数据一致性问题、共享文件的链接数一致性问题



8.5.4 重复数据的数据一致性问题

❖ 1、重复文件的一致性

文件名	i 结点
文件 1	17
文件 2	22
文件 3	12
文件 4	84

(a) 无重复文件目录

文件名	i 结点		
文件 1	17	19	40
文件 2	22	72	91
文件 3	12	30	29
文件 4	84	15	66

(b) 有重复文件目录

* 对主文件与其各备份文件的数据一致性进行检查。
* 重复文件修改处理办法：如某重复文件的其中一个文件被修改了，则据文件目录查找到该重复文件的其它各备份文件索引结点号，进而找到这些备份文件完成修改或替换。

8.5.4 重复数据的数据一致性问题

❖ 2、链接数一致性检查

* 对共享文件的链接数一致性进行检查。
* 设置文件索引结点计数器表，从根目录开始统计各文件的索引结点数量，目录检查完后，将计数器各表项中的索引结点号计数值与该文件索引结点中的链接计数count值加以比较，如果两者一致，则表明正确，否则便产生了链接数据不一致性错误。

8.5.4 重复数据的数据[illegible]

正常情况下两组计数器中的数据应互补

❖ 3、盘块号一致性检查

* 对盘块分配使用情况进行一致检查。

计数器组 \ 盘块号	0 1	2 3	4 5	6 7	8 9	10 11	12 13	14 15
空闲盘块号计数器组	1 1	0 1	0 1	1 1	1 0	0 1	1 1	0 0
数据盘块号计数器组	0 0	1 0	1 0	0 0	0 1	1 0	0 0	1 1

(a) 正常情况

计数器组 \ 盘块号	0 1	2 3	4 5	6 7	8 9	10 11	12 13	14 15
空闲盘块号计数器组	1 1	0 1	0 1	1 1	1 0	0 1	1 1	0 0
数据盘块号计数器组	0 0	0 0	1 0	0 0	0 1	1 0	0 0	1 1

(b) 丢失了盘块

8.5.4 重复数据的数据一致性问题

盘块号 / 计数器组	0 1	2 3	4 5	6 7	8 9	10 11	12 13	14 15
空闲盘块号计数器组	1 1	0 1	2 1	1 1	1 0	0 1	1 1	0 0
数据盘块号计数器组	0 0	1 0	0 0	0 0	0 1	1 0	0 0	1 0

(c) 空闲盘块号重复出现

盘块号 / 计数器组	0 1	2 3	4 5	6 7	8 9	10 11	12 13	14 15
空闲盘块号计数器组	1 1	0 1	1 0	1 1	1 0	0 1	1 1	0 0
数据盘块号计数器组	0 0	1 0	0 2	0 0	0 1	1 0	0 0	1 1

(d) 数据盘块号重复出现

本章小结

**掌握
* 理解

- 文件的物理结构**
  - * 三种分配方式：连续、链接（隐式、显式）、索引
- 文件存储空间的管理**
  - * 空闲表法、空闲链表法、位示图法、成组链接法
- 提高磁盘I/O速度的途径
  - * 指针交付、置换算法、周期性回写、提前读、延迟写、优化物理块分布、虚拟盘
  - * 并行交叉存取：Raid 1-7
- 文件保护*
  - * 三种保护措施：访问控制、磁盘容错(共三级)、后备系统
- 数据一致性控制
  - * 事务、检查点、并发控制、重复数据的一致性

本章作业

- 要求：
  - 一定要做在作业本上
- 交作业日期：
  - 第15周周五课上交最后一次作业
- 作业内容：
  - 操作系统第8章网络在线测试 （12月8日前完成）
  - 教材P276 第7、11、14、15题 (第15周课上交作业)

本章课程结束！谢谢大家！