




第八章 磁盘存储器的管理

8.1 外存的组织方式

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



8.1 外存的组织方式

如前所述，文件的物理结构直接与外存的组织方式有关。对于不同的外存组织方式，将形成不同的文件物理结构。目前常用的外存组织方式有：

- (1) 连续组织方式。
- (2) 链接组织方式。
- (3) 索引组织方式。



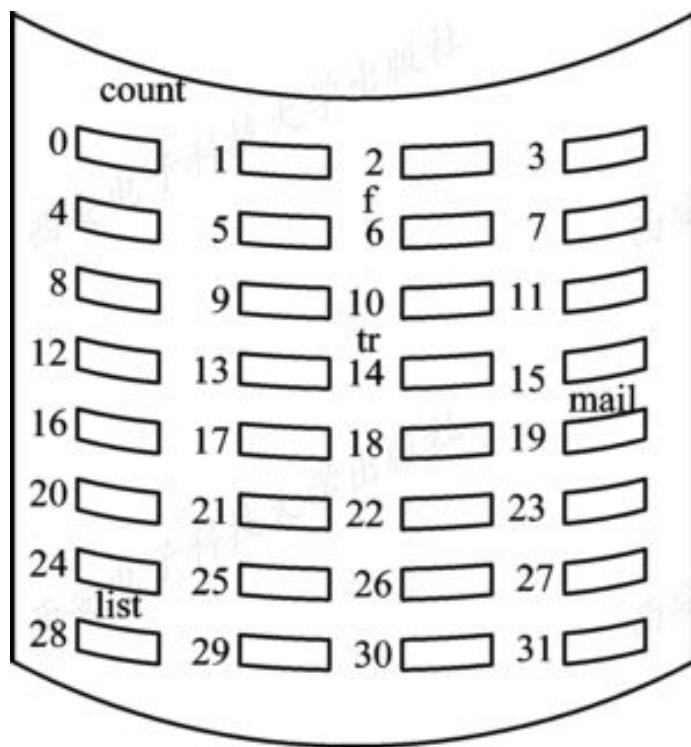
8.1.1 连续组织方式

连续组织方式又称连续分配方式，要求为每一个文件分配一组相邻接的盘块。

例如，第一个盘块的地址为 b ，则第二个盘块的地址为 $b+1$ ，第三个盘块的地址为 $b+2$ ，...。通常，它们都位于一条磁道上，在进行读/写时，不必移动磁头。

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，这样所形成的文件结构称为**顺序文件结构**，此时的物理文件称为**顺序文件**。

为使系统能找到文件存放的地址，应在目录项的“文件物理地址”字段中，记录该文件第一个记录所在的盘块号和文件长度(以盘块数进行计量)。



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

假定了记录与盘块的大小相同

图8-1 磁盘空间的连续组织方式



外存的碎片：随着文件建立时空间的分配和文件删除时空间的回收，将使磁盘空间被分割成许多小块，这些较小的连续区已难于用来存储文件，即**外存的碎片**。

紧凑：利用**紧凑**的方法，将盘上所有的文件紧靠在一起，把所有的碎片拼接成一大片连续的存储空间。

但为了将外存上的空闲空间进行一次紧凑，**所花费的时间**远比将内存紧凑一次所花费的时间多得多。







连续组织方式的主要优点有：

(1) 顺序访问容易

系统可从目录中找到该顺序文件所在的**第一个盘块号**，**从此开始顺序地、逐个盘块地往下读/写**。连续分配也**支持直接存取**。例如，要访问一个从 b 块开始存放的文件中的第 i 个盘块的内容，就可直接访问 $b+i$ 号盘块。

(2) 顺序访问速度快

因为由连续分配所装入的文件，其所占用的盘块可能是**位于一条或几条相邻的磁道上**，这时，磁头的移动距离最少，因此，这种对文件访问的速度是几种存储空间分配方式中最高的一种。



连续组织方式的主要缺点如下：

- (1) 要求为一个文件分配连续的存储空间
- (2) 必须事先知道文件的长度
- (3) 不能灵活地删除和插入记录
- (4) 对于那些动态增长的文件

8.1.2 链接组织方式

如果可以将文件装到多个离散的盘块中，就可消除连续组织方式的上述缺点。在采用链接组织方式时，可为文件分配多个不连续的盘块，再通过每个盘块上的链接指针，将同属于一个文件的多个离散的盘块链接成一个链表，由此所形成的物理文件称为链接文件。

链接组织方式的主要优点是：

- (1) 消除了磁盘的外部碎片，提高了外存的利用率。
- (2) 对插入、删除和修改记录都非常容易。
- (3) 能适应文件的动态增长，无需事先知道文件的大小。



1. 隐式链接

在采用隐式链接组织方式时，在文件目录的每个目录项中，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。

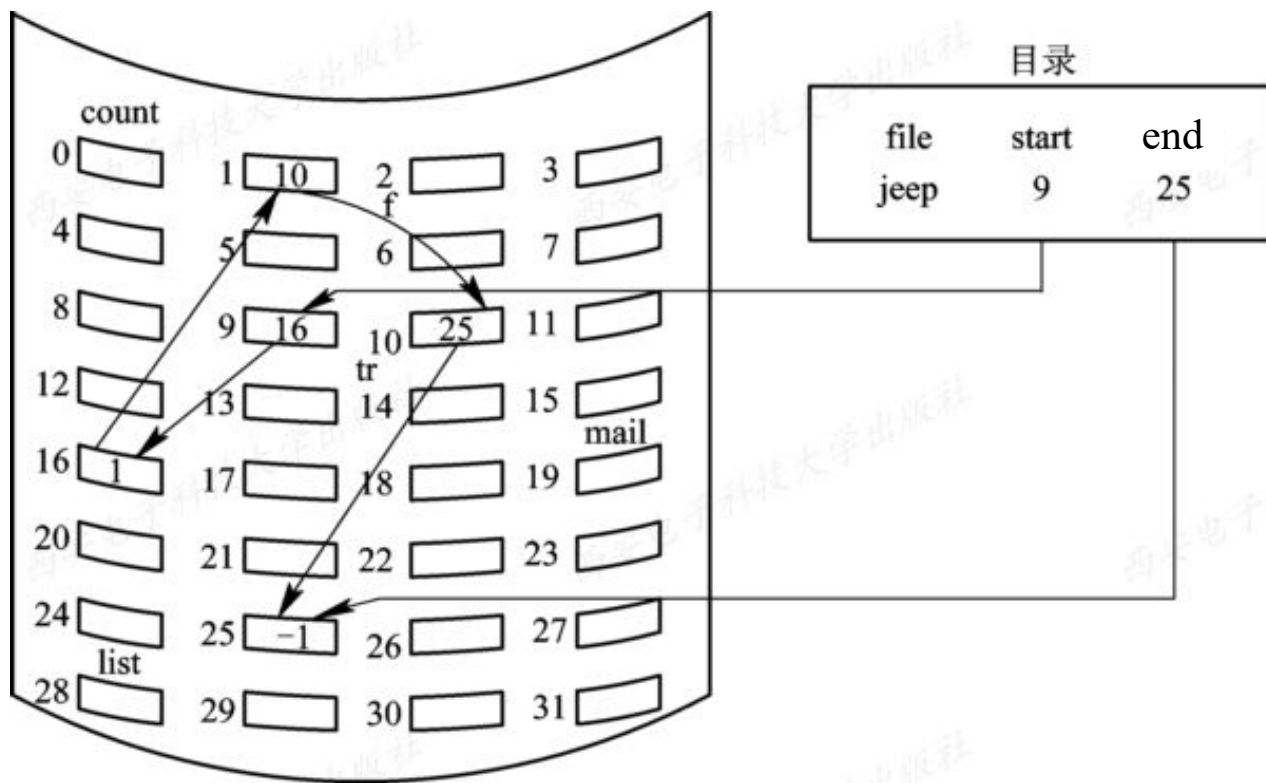


图8-2 磁盘空间的链接式分配



主要问题：

只适合于顺序访问，对随机访问是极其低效的。其可靠性较差，因为只要其中的任何一个指针出现问题，都会导致整个链的断开。

为了提高检索速度和减小指针所占用的存储空间，可以将几个盘块组成一个簇(cluster)。比如，一个簇可以包含4个盘块。在进行盘块分配时，以簇为单位。

2. 显式链接

这是指把用于链接文件各物理块的指针显式地存放在内存的一张链接表中。该表在整个磁盘中仅设置一张。

该表中，凡是属于某一文件的第一个盘块号，或者说是每一条链的链首指针所对应的盘块号，均作为文件地址被填入相应文件的FCB的“物理地址”字段中。

由于查找记录的过程是在内存中进行的，因而不仅显著地提高了检索速度，而且大大减少了访问磁盘的次数。

该表也称为文件分配表FAT（File Allocation Table）。

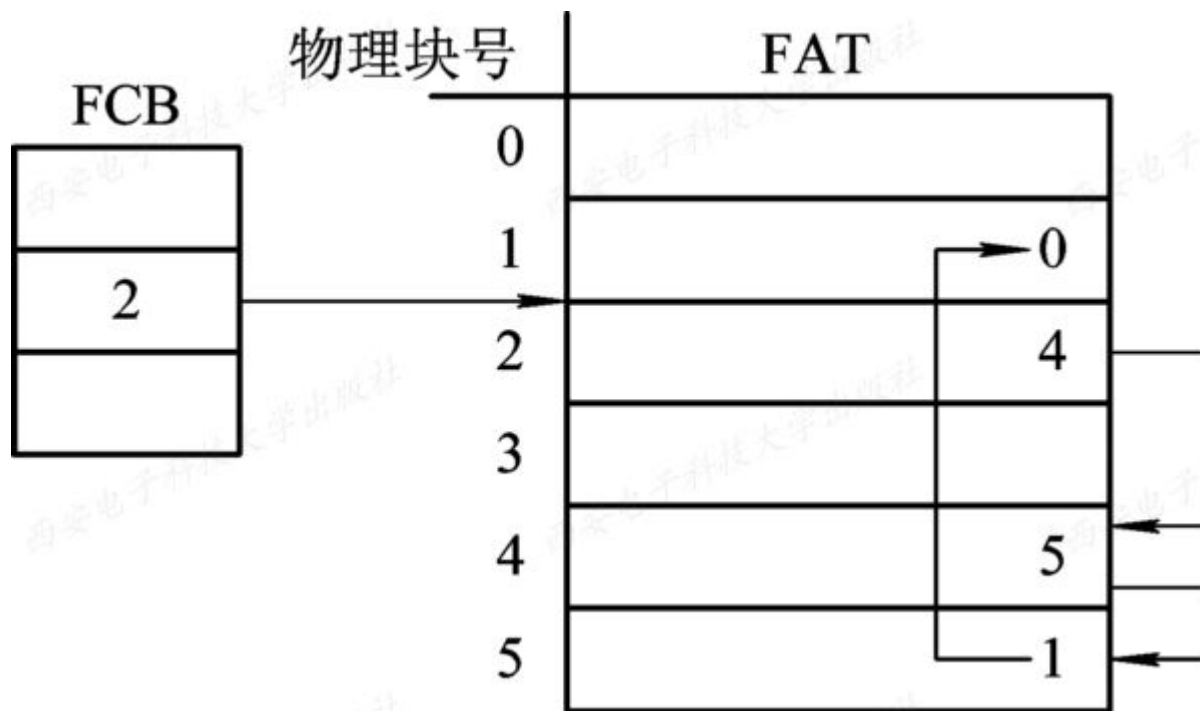


图8-3 显式链接结构



8.1.5 索引组织方式

1. 单级索引组织方式

链接组织方式虽然解决了连续组织方式所存在的问题(即不便于随机访问),但又出现了另外两个问题,即:

① **不能支持高效的直接存取**, 要**对一个较大的文件进行存取**, 须在FAT中顺序地查找许多盘块号;

② FAT需占用较大的内存空间, 由于一个文件所占用盘块的盘块号是随机地分布在FAT中的, 因而只有将整个FAT调入内存, 才能保证在FAT中找到一个文件的所有盘块号。

索引分配方式为每个文件分配一个索引块，把分配到该文件的所有盘块号都记录在该索引块中。在建立一个文件时，只在其目录项中填上指向该索引块的指针。

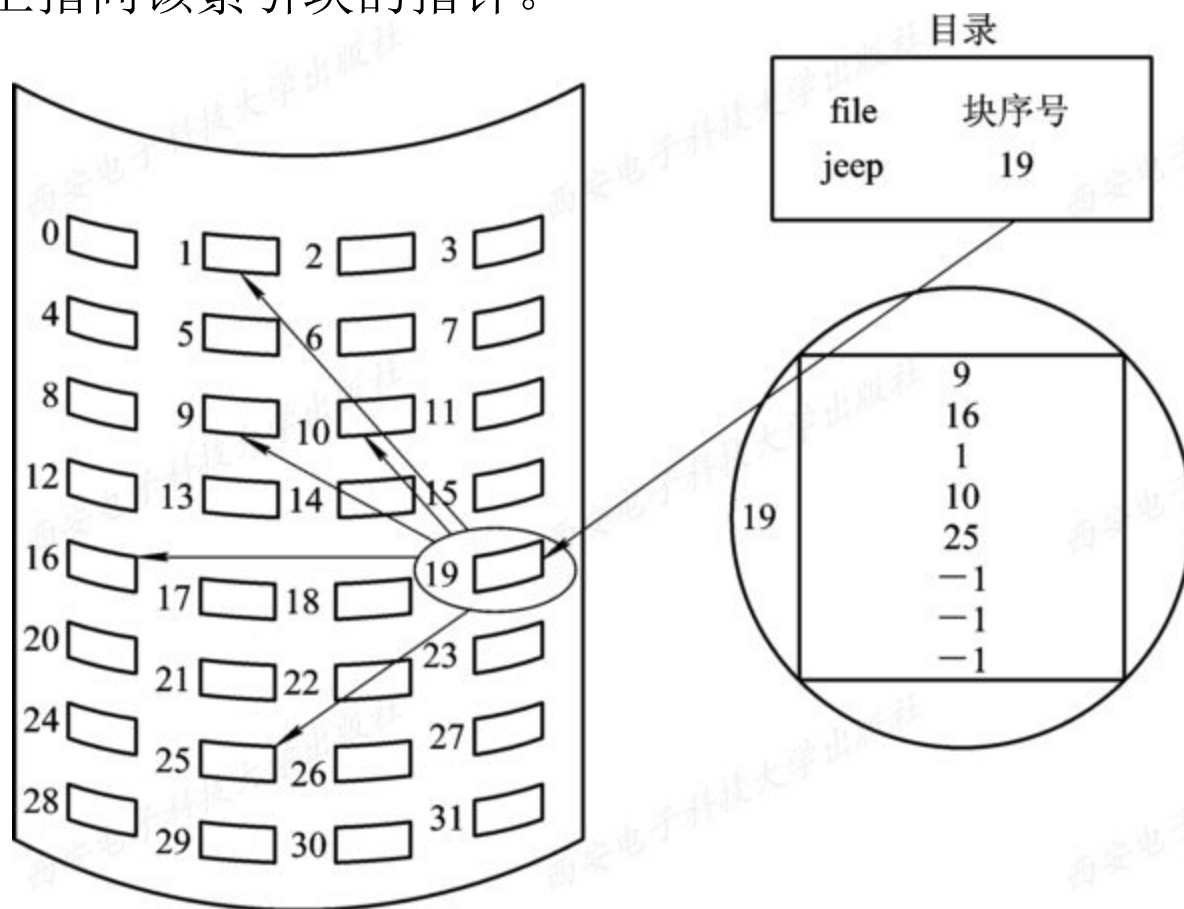


图8-6 索引分配方式



索引组织方式的主要优点是：**支持直接访问；没有文件碎片问题。**

索引组织方式的主要问题是：每当建立一个索引文件时，都要为该文件分配一个索引块，将分配给该文件的所有盘块号记录于其中，因而**增加了系统存储空间的开销。**





2. 多级索引组织方式

在为一个文件分配磁盘空间时，如果所分配出去的盘块的盘块号已经装满一个索引块时，OS须再为该文件分配另一个索引块，用于将以后继续为之分配的盘块号记录于其中。依此类推，**再通过链指针将各索引块按序链接起来。**

当文件太大，其索引块太多时，这种方法是低效的。此时，应为这些索引块再建立一级索引，称为第一级索引，即系统再分配一个索引块，作为第一级索引的索引块，将第一块、第二块、……等索引块的盘块号填入到此索引表中，这样便形成了**两级索引分配方式**。如果文件非常大时，还可用三级、四级索引分配方式。

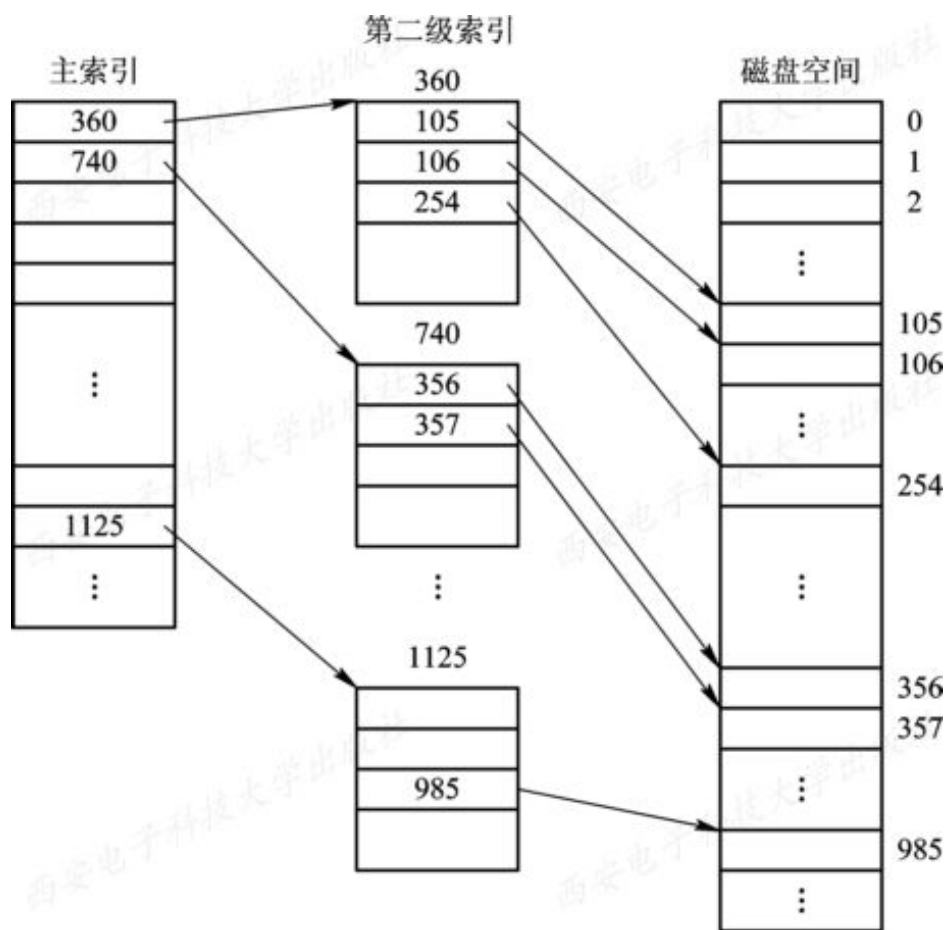
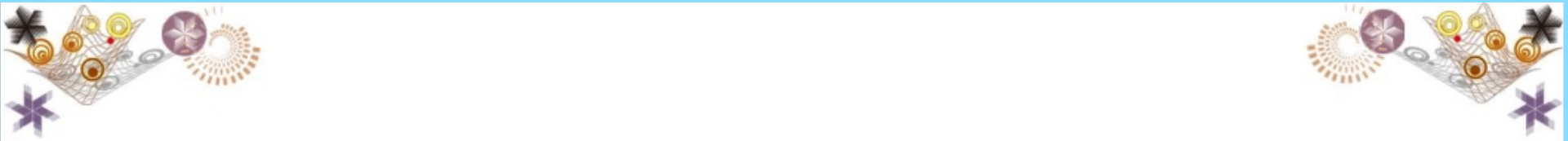


图8-7 两级索引分配



3. 增量式索引组织方式

1) 增量式索引组织方式的基本思想

为了能较全面地照顾到小、中、大及特大型作业，可以采取多种组织方式来构成文件的物理结构。

如果盘块的大小为1 KB或4 KB，对于小文件(如1 KB～10 KB或4 KB～40 KB)而言，最多只会占用10个盘块，为了提高对数量众多的小型作业的访问速度，最好能**将它们****的每一个盘块地址都直接放入文件控制块FCB(或索引结点)中**，这样就可以直接从FCB中获得该文件的盘块地址。



对于中等文件，可以采用**单级索引组织**方式。此时为获得该文件的盘块地址，只需先从FCB中找到该文件的索引表，从中便可获得，可将它称为一次间址。

对于大型和特大型文件，可以采用**两级和三级索引组织**方式，或称为二次间址和三次间址。所谓增量式索引组织方式，就是基于上述的基本思想来组织的，它既采用了直接寻址方式，又采用了单级和多级索引组织方式(间接寻址)。

通常又可将这种组织方式称为**混合组织方式**。

在UNIX系统中所采用的就是这种组织方式。



2) UNIX System V的组织方式

在UNIX System V的索引结点中设有13个地址项，即 $i.addr(0) \sim i.addr(12)$ ，如图8-8所示。

- (1) 直接地址。
- (2) 一次间接地址。
- (3) 多次间接地址。

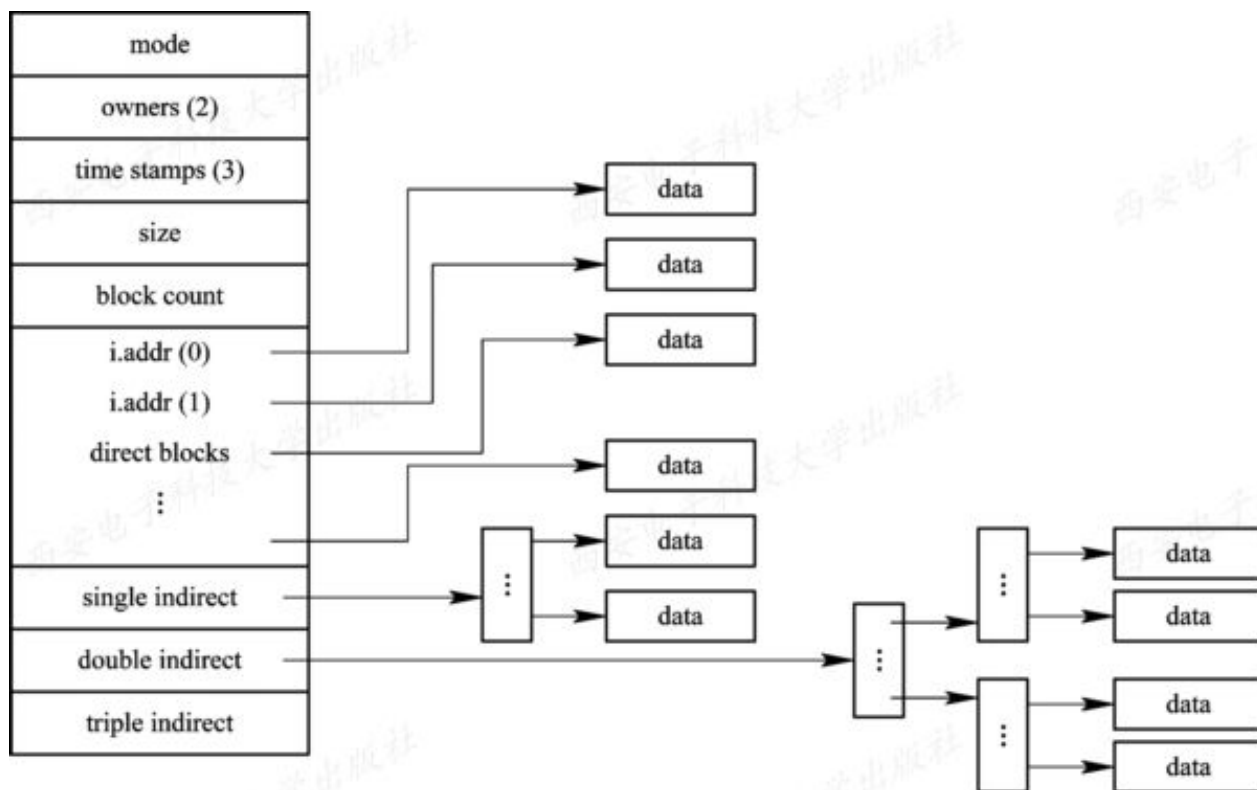


图8-8 混合索引方式

8.2 文件存储空间的管理

8.2.1 空闲表法和空闲链表法

1. 空闲表法

1) 空闲表

空闲表法属于**连续分配方式**，它为每个文件分配一块连续的存储空间。

即系统也为外存上的**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，其中包括表项序号、该空闲区的第一个盘块号、该区的空闲盘块数等信息。再将所有空闲区按其起始盘块号递增的次序排列，形成空闲盘块表。

按其起始盘块号
递增的次序排列

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

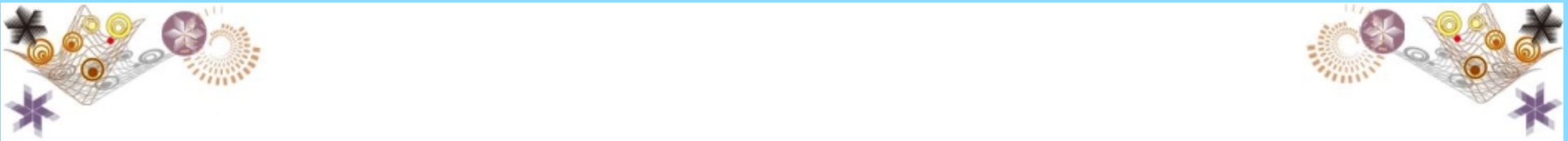


2) 存储空间的分配与回收

空闲盘区的分配与内存的动态分配类似，同样是采用**首次适应算法、循环首次适应算法等**。

在前面所介绍的对换方式中，对**对换空间**一般都采用**连续分配方式**。

对于文件系统，当文件较小(1~4个盘块)时，仍采用连续分配方式，为文件分配相邻接的几个盘块；当文件较大时，便采用离散分配方式。



2. 空闲链表法

1) 空闲盘块链



这是将磁盘上的所有空闲空间以**盘块为单位**拉成一条链，其中的每一个盘块都有指向后继盘块的指针。

2) 空闲盘区链

这是将磁盘上的所有**空闲盘区**(每个盘区可包含若干个**盘块**)拉成一条链。

在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(盘块数)的信息。

分配盘区的方法与内存的动态分区分配类似，通常采用首次适应算法。





8.2.2 位示图法

1. 位示图

位示图是利用二进制的一位来表示磁盘中一个盘块的使用情况。

当其值为“0”时，表示对应的盘块空闲；为“1”时，表示已分配。有的系统把“0”作为盘块已分配的标志，把“1”作为空闲标志。

盘上的所有盘块都有一个二进制位与之对应，这样，由所有盘块所对应的位构成一个集合，称为位示图。

	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

2. 盘块的分配

根据位示图进行盘块分配时，可分三步进行：

(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：

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

式中，*n*代表每行的位数。

(3) 修改位示图，令 $\text{map}[i, j] = 1$ 。

3. 盘块的回收

盘块的回收分两步：

(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：

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

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

(2) 修改位示图。令 $\text{map}[i, j] = 0$ 。





这种方法的主要优点是，从位示图中很容易找到一个或一组相邻接的空闲盘块。

例如，需要找到6个相邻接的空闲盘块，这只需在位示图中找出6个其值连续为“0”的位即可。

由于位示图很小，占用空间少，**因而可将它保存在内存中**，进而使在每次进行盘区分配时，无需首先把盘区分配表读入内存，从而节省了许多磁盘的启动操作。





8.2.3 成组链接法

1. 空闲盘块的组织

(1) 空闲盘块号栈，用来存放当前可用的一组空闲盘块的盘块号(最多含100个号)，以及栈中尚有的空闲盘块(号)数N。顺便指出，N还兼作栈顶指针用。

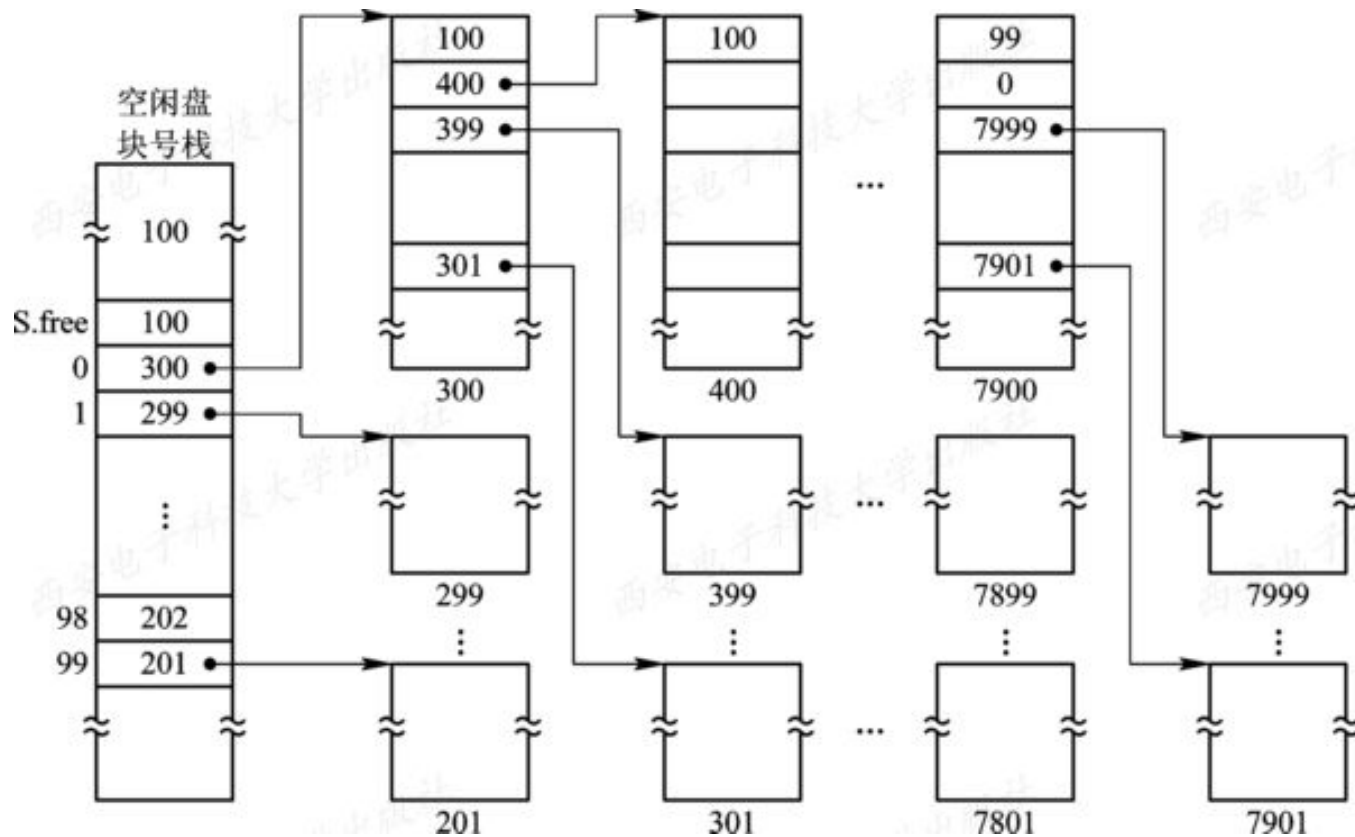




图8-11 空闲盘块的成组链接法



(2) 文件区中的所有空闲盘块被分成若干个组，比如，将每100个盘块作为一组。假定盘上共有10000个盘块，每块大小为1 KB，其中第201~7999号盘块用于存放文件，即作为文件区，这样，该区的最末一组盘块号应为7901~7999；次末组为7801~7900，...，倒数第二组的盘块号为301~400；第一组为201~300，如图8-11所示。

(3) 将每一组含有的盘块总数N和该组所有的盘块号记入其前一组的第一个盘块的S.free(0)~S.free(99)中。这样，由各组的第一个盘块可链成一条链。



(4) 将第一组的盘块总数和所有的盘块号记入空闲盘块号栈中，作为当前可供分配的空闲盘块号。

(5) 最末一组只有99个盘块，其盘块号分别记入其前一组的S.free(1)~S.free(99)中，而在S.free(0)中则存放“0”，作为空闲盘块链的结束标志。(注：最后一组的盘块数应为99，不应是100，因为这是指可供使用的空闲盘块。其编号应为(1~99)，0号中放空闲盘块链的结尾标志。)



2. 空闲盘块的分配与回收

当系统要为用户分配文件所需的盘块时，须调用盘块分配过程来完成。该过程首先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，然后将栈顶指针下移一格。若该盘块号已是栈底，即S.free(0)，这是当前栈中最后一个可分配的盘块号。

因此，须调用磁盘读过程，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(其中的有用数据已读入栈中)。然后，再分配一相应的缓冲区(作为该盘块的缓冲区)。最后，把栈中的空闲盘块数减1并返回。



在系统回收空闲盘块时，须调用盘块回收过程进行回收。它是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。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。





8.3 提高磁盘I/O速度的途径

- (1) 改进文件的目录结构以及检索目录的方法来减少对目录的查找时间；
- (2) 选取好的文件存储结构，以提高对文件的访问速度；
- (3) 提高磁盘的I/O速度，能将文件中的数据快速地从磁盘传送到内存中，或者相反。其中的第1和第2点已在上一章或本章作了较详细的阐述，本节主要对如何提高磁盘的I/O速度作一简单介绍。

8.3.1 磁盘高速缓存(Disk Cache)

指在**内存中**为磁盘盘块设置的一个缓冲区，在缓冲区中保存了某些盘块的副本。

在设计磁盘高速缓存时需要考虑的问题有：

- (1) 如何将磁盘高速缓存中的数据传送给请求进程；
- (2) 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。



1. 数据交付(Data Delivery)方式

如果I/O请求所需要的数据能从磁盘高速缓存中获取，此时就需要将磁盘高速缓存中的数据传送给请求进程。

所谓的数据交付就是指将磁盘高速缓存中的数据传送给请求者进程。系统可以采取两种方式将数据交付给请求进程：

(1) 数据交付：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：只将指向高速缓存中某区域的指针交付给请求者进程。



2. 置换算法

现在不少系统在设计其高速缓存的置换算法时，除了考虑到最近最久未使用这一原则外，还考虑了以下几点：



- (1) 访问频率。
- (2) 可预见性。
- (3) 数据的一致性。



3. 周期性地写回磁盘

还有一种情况值得注意，那就是根据LRU算法，那些经常要被访问的盘块数据可能会一直保留在高速缓存中，长期不会被写回磁盘。

需要周期性的写回磁盘，UNIX系统的SYNC系统调用可以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，时间间隔为30s。



8.3.2 提高磁盘I/O速度的其它方法

能有效地提高磁盘I/O速度的方法还有许多，如提前读、延迟写等：

1. 提前读：预知下一次要读的盘块，提前读入缓冲区
2. 延迟写：推迟缓冲区中的数据写回磁盘的时间。
3. 优化物理块的分布：将文件的数据块安排在同一条磁道的盘块上，提高访问速度。



4. 虚拟盘

由于访问内存的速度远高于访问磁盘的速度，于是有人试图利用内存空间去仿真磁盘，形成所谓虚拟盘，又称为RAM盘。

该盘的设备驱动程序也可以接受所有标准的磁盘操作，但这些操作的执行不是在磁盘上而是在内存中。这对用户都是透明的。



8.3.3 廉价磁盘冗余阵列(RAID)

1. 并行交叉存取

这是把在大、中型机中，用于提高访问内存速度的并行交叉存取技术应用到磁盘存储系统中，以提高对磁盘的I/O速度。

在该系统中，有多台磁盘驱动器，系统将每一盘块中的数据分为若干个子盘块数据，再把每一个子盘块的数据分别存储到各个不同磁盘中的相同位置上。以后当要将一个盘块的数据传送到内存时，采取并行传输方式，将各个盘块中的子盘块数据同时向内存中传输，从而使传输时间大大减少。

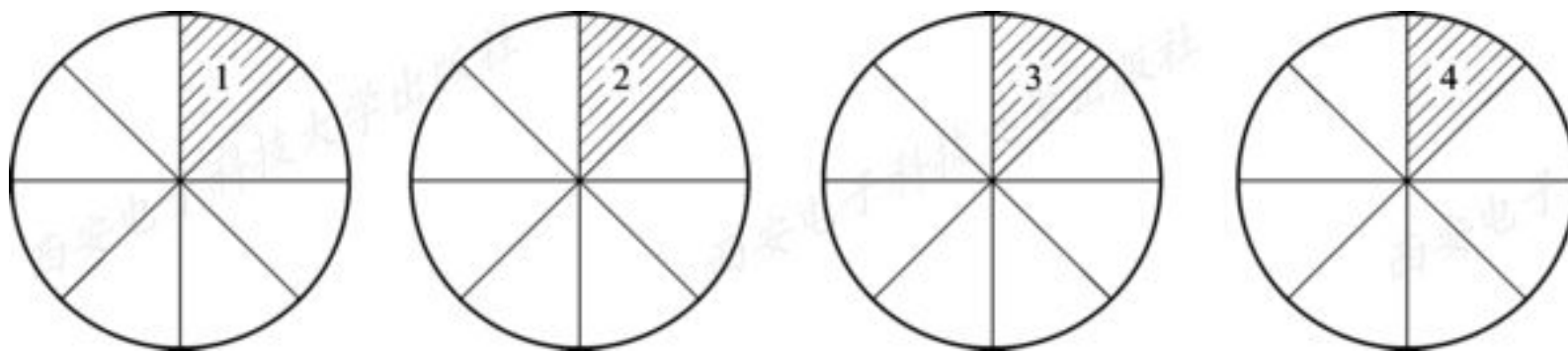


图8-12 磁盘并行交叉存取方式





2. RAID的分级

RAID在刚被推出时，是分成6级的，后来又增加了RAID 6级和RAID 7级。

- (1) RAID 0级：并行交叉存取。
- (2) RAID 1级：磁盘镜像。
- (3) RAID 3级：并行传输。
- (4) RAID 5级：独立传送功能。
- (5) RAID 6级和RAID 7级：强化了了的RAID。



3. RAID的优点

(1) 可靠性高，除了RAID 0级外，其余各级都采用了容错技术。当阵列中某一磁盘损坏时，并不会造成数据的丢失。此时可根据其它未损坏磁盘中的信息来恢复已损坏的盘中的信息。其可靠性比单台磁盘机高出一个数量级。

(2) 磁盘I/O速度高，由于采取了并行交叉存取方式，可使磁盘I/O速度提高 $N-1$ 倍。

(3) 性能/价格比高，RAID的体积与具有相同容量和速度的大型磁盘系统相比，只是后者的 $1/3$ ，价格也只是后者的 $1/3$ ，且可靠性高。换言之，它仅以牺牲 $1/N$ 的容量为代价，换取了高可靠性。



8.4 提高磁盘可靠性的技术

磁盘容错技术是通过增加冗余的磁盘驱动器，磁盘控制器等方法来提高磁盘系统可靠性的一种技术。

8.4.1 第一级容错技术SFT-I

第一级容错技术(SFT-I)是最基本的一种磁盘容错技术，主要用于防止因磁盘表面缺陷所造成的数据丢失。它包含双份目录、双份文件分配表及写后读校验等措施。



1. 双份目录和双份文件分配表

在磁盘上存放的文件目录和文件分配表FAT，是文件管理所用的重要数据结构。

为了防止这些表格被破坏，可在不同的磁盘上或在磁盘的不同区域中分别建立(双份)目录表和FAT。其中一份为主目录及主FAT，另一份为备份目录及备份FAT。一旦由于磁盘表面缺陷而造成主文件目录或主FAT的损坏时，系统便自动启用备份文件目录及备份FAT，从而可以保证磁盘上的数据仍是可访问的。



2. 热修复重定向和写后读校验

由于磁盘价格昂贵，在磁盘表面有少量缺陷的情况下，则可采取某种补救措施后继续使用。一般主要采取以下两个补救措施：

- (1) 热修复重定向。
- (2) 写后读校验方式。



8.4.2 第二级容错技术SFT-II

1. 磁盘镜像(Disk Mirroring)

为了避免磁盘驱动器发生故障而丢失数据，便增设了磁盘镜像功能。为实现该功能，须在同一磁盘控制器下，再增设一个完全相同的磁盘驱动器，如图8-13所示。

2. 磁盘双工(Disk Duplexing)

如果控制这两台磁盘驱动器的磁盘控制器发生故障，或主机到磁盘控制器之间的通道发生故障，磁盘镜像功能便起不到数据保护的作用。因此，在第二级容错技术中，又增加了磁盘双工功能，即将两台磁盘驱动器分别接到两个磁盘控制器上，同样使这两台磁盘机镜像成对，如图8-14所示。

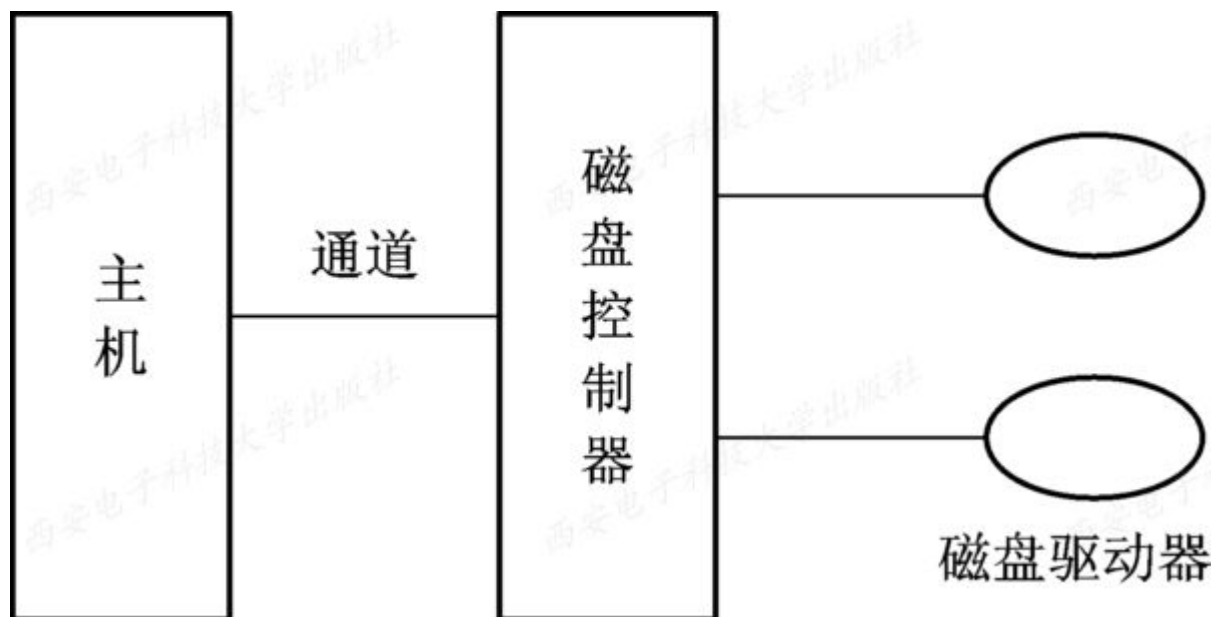


图8-13 磁盘镜像示意图

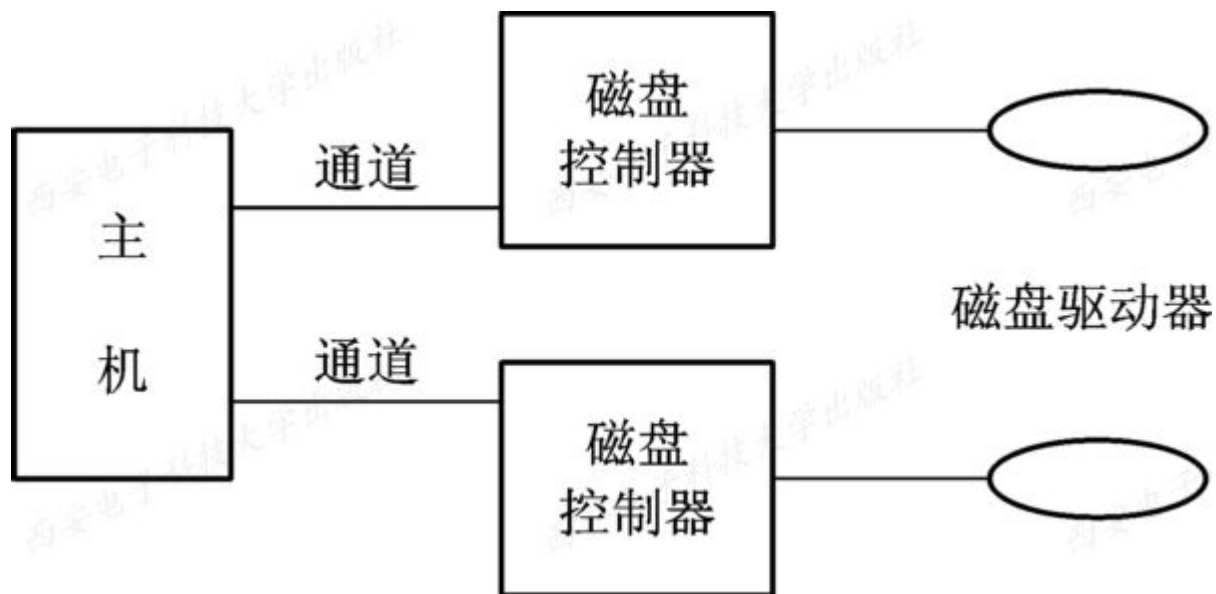




图8-14 磁盘双工示意图



8.4.3 基于集群技术的容错功能

1. 双机热备份模式

如图8-15所示，在这种模式的系统中，备有两台服务器，两者的处理能力通常是完全相同的，一台作为主服务器，另一台作为备份服务器。

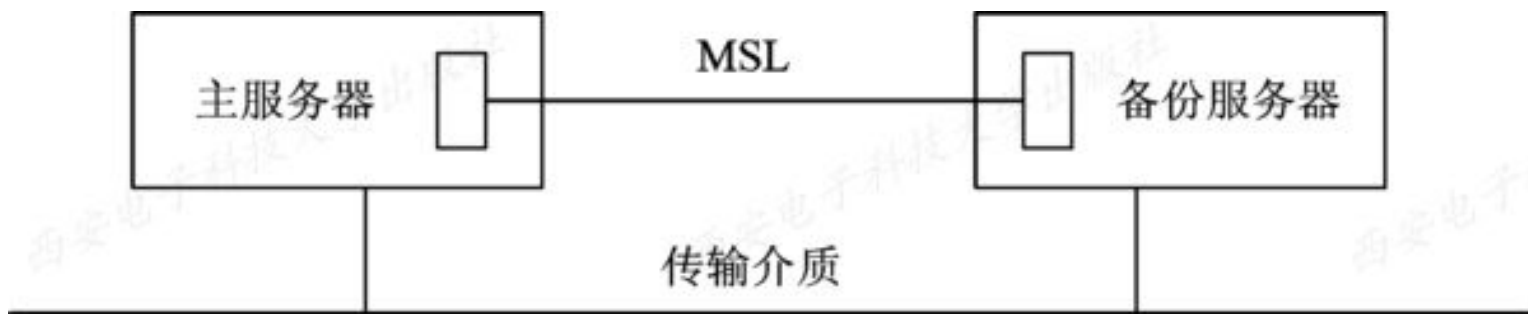


图8-15 双机热备份模式

2. 双机互为备份模式

在双机互为备份模式中，平时，两台服务器均为在线服务器，它们各自完成自己的任务，例如，一台作为数据库服务器，另一台作为电子邮件服务器。为了实现两者互为备份的功能，在两台服务器之间，应通过某种专线将其连接起来。如果希望两台服务器之间能相距较远，最好利用FDDI单模光纤来连接两台服务器。在此情况下，最好再通过路由器将两台服务器互连起来，作为备份通信线路。图8-16示出了双机互为备份系统的情况。

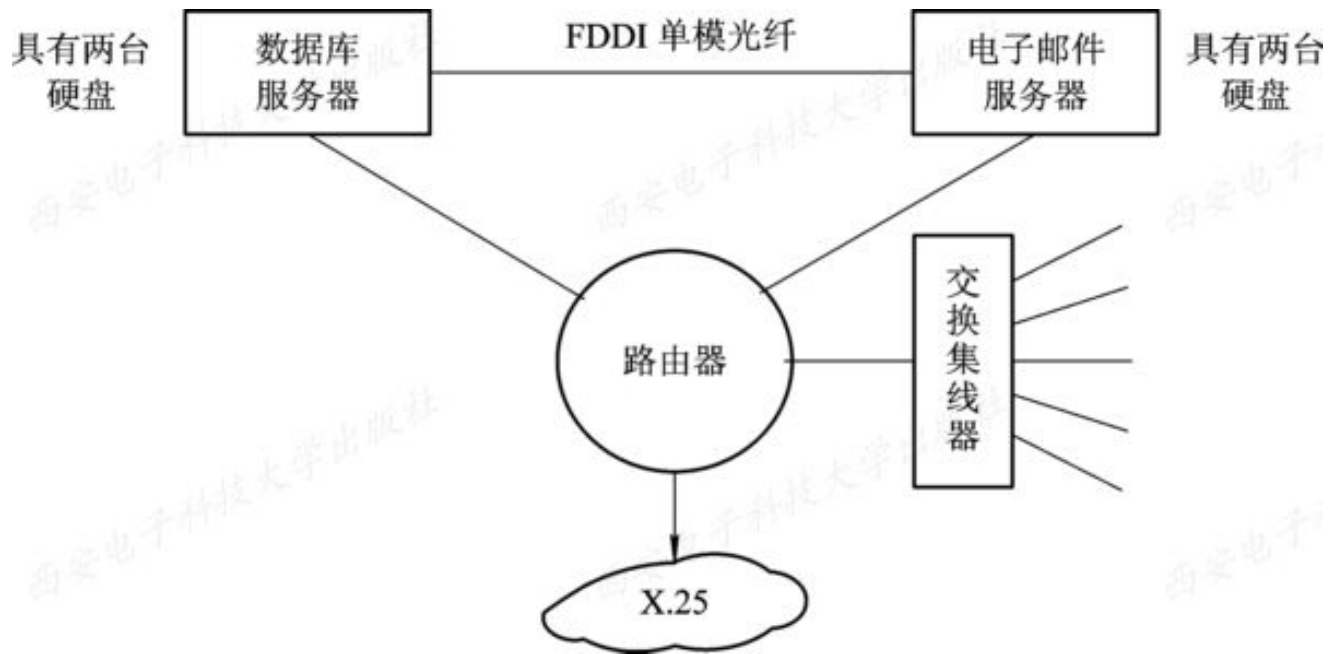


图8-16 双机互为备份系统的示意图



3. 公用磁盘模式

为了减少信息复制的开销，可以将多台计算机连接到一台公共的磁盘系统上去。该公共磁盘被划分为若干个卷。每台计算机使用一个卷。如果某台计算机发生故障，此时系统将重新进行配置，根据某种调度策略来选择另一台替代机器，后者对发生故障的机器的卷拥有所有权，从而可接替故障计算机所承担的任务。这种模式的优点是消除了信息的复制时间，因而减少了网络和服务器的开销。



8.4.4 后备系统

1. 磁带机

它是最早作为计算机系统的外存储器。但由于它只适合存储顺序文件，故现在主要把它作为后备设备。磁盘机的主要优点是容量大，一般可达数GB至数十GB，且价格便宜，故在许多大、中型系统中都配置了磁带机。其缺点是只能顺序存取且速度也较慢，为数百KB到数MB，为了将一个大容量磁盘上的数据拷贝到磁带上，需要花费很多时间。



2. 硬盘

(1) 移动磁盘。

(2) 固定硬盘驱动器。

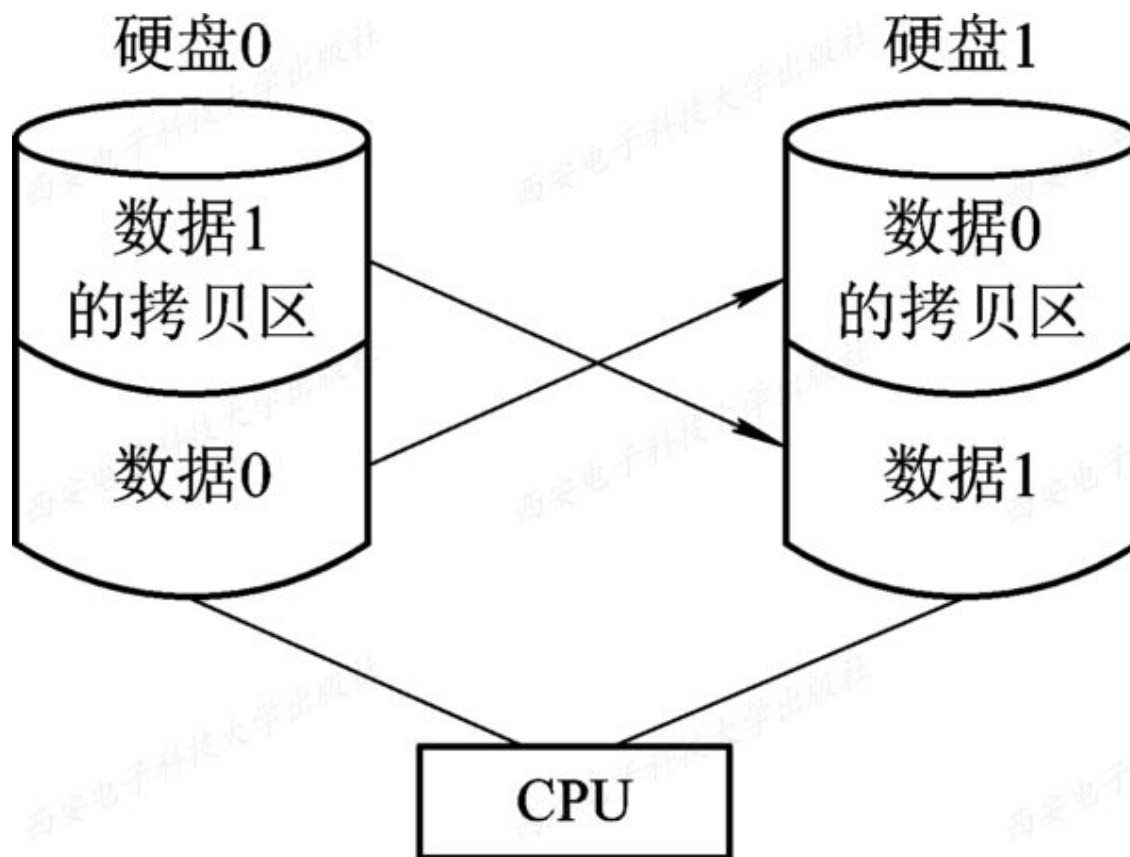


图8-17 利用大容量硬盘兼做后备系统



3. 光盘驱动器

光盘驱动器是现在最流行的多媒体设备，可将它们分为如下两类：

- (1) 只读光盘驱动器CD-ROM和DVD-ROM。
- (2) 可读写光盘驱动器。