第九章 文件管理

- 9.1 文件及文件系统
- 9.2 文件目录管理
- 9.3 文件存储空间的分配与管理
- 9.4 文件的共享与保护

9.1 文件及文件系统

- 9.1.1 文件
- 9.1.2 文件系统及其功能
- 9.1.3 文件的逻辑结构和访问方式
- 9.1.4 文件的物理结构

9.1.1 文件

1. 文件

- 文件是由创建者定义的，具有文件名的且在逻辑上具有完整意义的一组相关信息的集合。它可以是一组相关的字符流的集合，也可以是一组相关的记录的集合。
- 文件不仅具有文件名，还包含文件类型、文件主、访问权限、文件被创建的时间、文件长度等，统称为文件属性。

9.1.1 文件

2. 文件分类

(1)按文件的性质和用途，可以把文件分成以下三类：

①系统文件：由系统软件构成，只允许用户调用。

②用户文件：由用户的源代码、可执行文件或数据构成。

③库文件：由标准子程序及常用例程构成，允许调用。

（2）按文件中的数据形式，可以把文件分成以下三类：

①源文件：由源程序和数据构成。

②目标文件：把源程序经过相应语言的编译程序编译。

③可执行文件：把目标文件由链接程序链接。

（3）按存取控制属性，可以把文件分成以下三类：

①只执行文件。

②只读文件。

③读写文件。

9.1.2 文件系统及其功能

1. 文件系统

文件系统是指含有大量的文件及其属性的说明，能对文件进行操纵和管理的软件，同时它还向用户提供了使用文件的接口，是操作系统的重要组成部分。

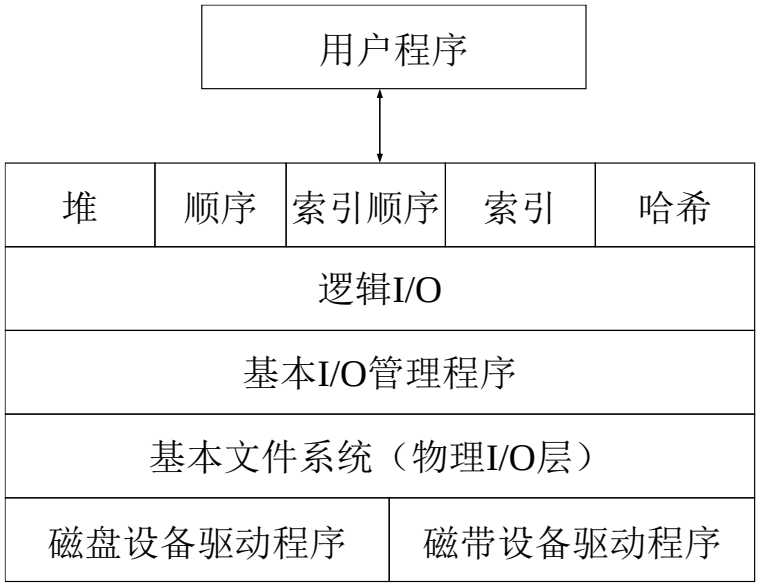


图9-1 文件系统软件体系结构

9.1.2 文件系统及其功能

2．文件系统的主要功能

（1）实现“按名存取”。

（2）文件存储空间的管理。

（3）对文件及文件目录的管理。

（4）文件组织。

（5）提供文件共享和保护等机制。

9.1.3 文件的逻辑结构和访问方式

文件的逻辑结构，是从用户观点出发，所观察到的文件的组织形式，是用户可以直接处理的数据及其结构，它独立于物理特性，又称为文件组织。它由用户访问记录的方式确定。

1．按文件是否有结构分类

（1）有结构文件(Structured File)，是指由一个以上的记录构成的文件，又称为记录式文件；而记录的长度又可分为定长和不定长两类。

① 定长记录 ② 变长记录

（2）无结构文件(Unstructured File)，是指由字符流构成的文件，又称为流式文件。流式文件也可以看成是每个记录中只含有一个字符的记录式文件的特例。

9.1.3 文件的逻辑结构和访问方式

2.按文件的组织方式分类

① 堆文件：是最简单的文件组织形式。数据按它们到达的顺序被采集，每个记录由一串数据组成。堆的目的仅仅是积累大量的数据并保存数据。

② 顺序文件：在这类文件中，所有记录都具有固定的格式，并且由相同数目、长度固定的域按照特定的顺序组成。

③ 索引文件：对于变长记录文件，查找一个记录必须从第一个记录顺序查起，这样很耗时。为了定位记录的方便，为文件建立一张索引表，并在表中为每一条变长记录设置一个索引项，用来记录指向该记录的指针以及记录的长度。以这样的方式组织的文件称为索引文件。对索引表可以按关键字排序，因此其本身也可以看成是一个定长记录的顺序文件。

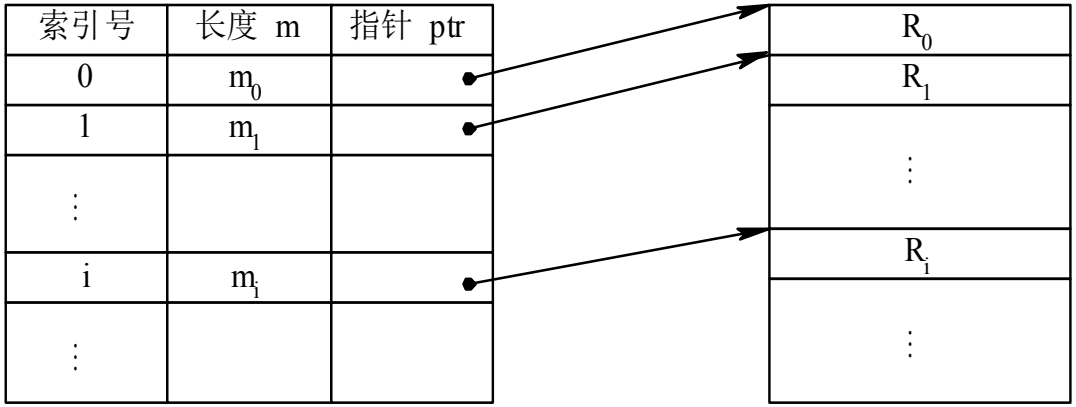


图 索引文件的组织

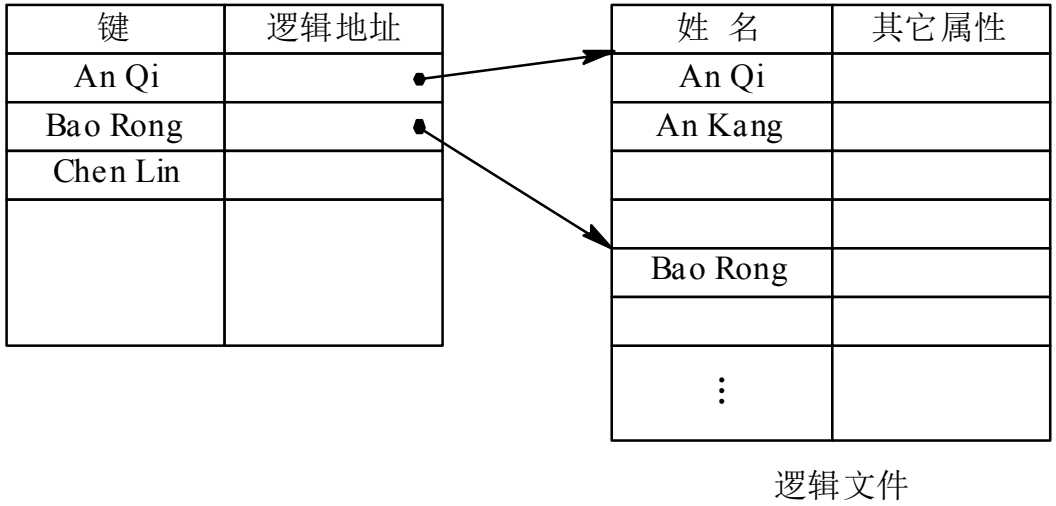


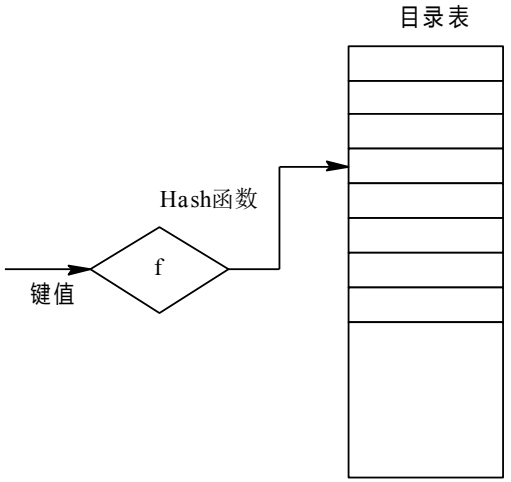
图 索引顺序文件

9.1.3 文件的逻辑结构和访问方式

2.按文件的组织方式分类（续）

④ 索引顺序文件：将顺序文件和索引文件相结合，将变长记录顺序文件中的所有记录分为若干个组，同时为文件建立一张索引表，并为每一组记录中的第一个记录设置一个索引项，其中包含指向该记录的指针和该组记录的关键字。

⑤直接文件或散列文件：直接文件可以根据给定的关键字直接获得记录的物理地址。目前应用最广泛的直接文件是哈希文件，它利用哈希函数将关键字转换为相应记录的地址。



9.1.3 文件的逻辑结构和访问方式

用户根据其对文件内数据的处理方法不同，有不同的访问数据的方法。一般的，用户对文件的访问有下列两种基本方法。

（1）顺序访问，指用户从文件初始数据开始依次访问文件中的信息。经常被顺序访问的文件的逻辑记录应该连续地存储在文件存储器上。这种起源于文件磁带模型的访问方式称为顺序访问。

（2）直接访问，指用户随机地访问文件中的某段信息。如果要支持用户以直接访问方式访问文件，文件必须存放于可以支持快速定位的随机访问存储设备中。

9.1.4 文件的物理结构

文件的物理结构，又称为文件的存储结构。是指文件在外存上的存储组织形式，与存储介质的性能有关。此外，文件在辅存中的物理组织还取决于分块策略和文件分配策略。

9.2 文件目录管理

- 9.2.1 文件控制块
- 9.2.2 索引结点的引入
- 9.2.3 单级目录结构
- 9.2.4 两级目录结构
- 9.2.5 树形目录结构
- 9.2.6 目录查询技术

9.2.1 文件控制块

为了能对文件进行正确的存取，就必须为文件设置用于描述和控制文件的数据结构，称为文件控制块FCB(File Control Block)。文件与FCB一一对应，而把FCB的有序集合称为文件目录。换言之，一个FCB就是一个文件目录项。通常，一个文件目录也可以看作是一个文件，称为目录文件。

文件控制块FCB的基本内容如下：

（1）文件名。

（2）文件物理位置。

（3）文件逻辑结构。

（4）文件的物理结构。

（5）存储控制信息。

（6）管理信息。

9.2.2 索引结点的引入

文件目录通常是存放在磁盘上的。当文件很多时，文件目录就要占用大量的盘块。而查找目录时，就要逐个查找各个盘块，并多次启动磁盘。

实质上，在检索目录文件的过程中，只用到了文件名。为此，可以把文件名与文件描述信息分开，把文件描述信息单独形成一个称为索引结点（Index Node）的数据结构；而每个目录项中，仅包含文件名及指向该文件所对应的索引结点指针，这样可以大大节省系统开销。

9.2.2 索引结点的引入

根据索引结点存放的位置不同，分为两类：磁盘索引结点和内存索引结点。

1. 磁盘索引结点（Disk Index Node）

是指存放在磁盘上的索引结点。每个文件有唯一的磁盘索引结点。它主要包括以下内容：

（1）文件主标识。

（2）文件类型。一般文件/目录文件/特殊文件。

（3）文件存取权限。

（4）文件物理地址。（盘块号）

（5）文件长度。（字节）

（6）文件连接计数。共享该文件的进程（用户）计数。

（7）文件存取时间。

9.2.2 索引结点的引入

2．内存索引结点（Memory Index Node）

是指存放在内存的索引结点。当文件被打开时，要将磁盘索引结点拷贝到内存的索引结点中，以便于使用。内存的索引结点中，包括以下内容：

（1）索引结点编号；

（2）状态。指示该索引结点是否已上锁或已被修改；

（3）访问计数。指示访问该索引结点的进程个数；

（4）文件所在设备的逻辑设备号；

（5）链接指针：它包括分别指向空闲链表和散列队列的指针。

9.2.3 单级目录结构

目录结构的组织，关系到文件系统的存取速度、文件的共享性及安全性。目前常用的目录结构形式有：单级目录结构、两级目录结构和树形目录结构。

单级目录结构（如表9-1所示）是最简单的目录结构。整个系统中只有一张目录表，为每个文件分配一个目录项 (其中，状态位表示该目录项是否空闲，如“1”表示已分配，“0”表示未分配) 。

文件名	状态位	物理地址	其他属性
F1	1	...	...
F2	1	...	...
F3	1	...	...
...	...	...	...

表9-1 单级目录结构

2. 单级目录结构的优缺点

单级目录结构的优点是目录结构简单，能实现“按名存取”。但缺点也很明显，表现在：

（1）查找速度慢。用户所有的文件都在一个目录中，如果要查找某个文件，需要检索整个目录，比较费时。

（2）不允许重名，限制了用户对文件的命名。单级目录结构要求目录表中的所有文件名都是唯一的，不允许重名，这在多道程序环境下显然是难以做到的。因此，它只适用于单用户环境。

（3）不便于实现文件共享。单级目录结构要求所有用户只能用同一个名字来访问同一个文件，这对每个用户为文件命名来说，是不方便的。

9.2.4 两级目录结构

单级目录结构不适用于多道程序环境，为了解决这个问题，系统可以为每一个用户建立一个单独的用户文件目录UFD(User File Directory)，由每个用户所有文件的文件控制块组成。此外，在系统中还需要建立一个主文件目录MFD(Master File Directory)；在MFD中，每个UFD都占有一个目录项，其中记录了用户名和指向该用户目录的指针，这样的目录结构称为两级目录结构。

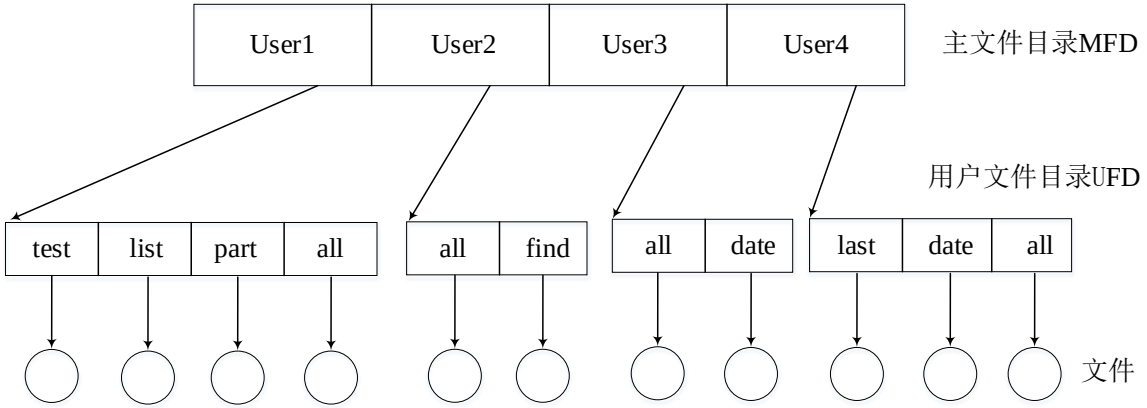


图9-2 两级目录结构

9.2.4 两级目录结构

两级目录结构的主要优点有：

（1）提高了目录的检索速度。

（2）在不同的用户目录中，可以命名相同文件名的文件，不会产生混淆，只要在用户自己的UFD中其文件名是唯一的即可，因此，解决了命名冲突问题。

（3）不同用户可以使用不同的文件名，来访问系统中的同一个共享文件。

主要缺点为：

（1）该结构虽然能有效地将多个用户隔离开来，但当要实现多个用户之间的文件共享时，这种隔离反而会为共享带来不便。

（2）当某个用户目录下文件较多时，查找文件时速度还是会比较慢。

9.2.5 树形目录结构

- 三级以及三级以上的文件目录结构称为树形目录结构。主目录在树形结构的目录中，作为树的根结点，称为根目录(Root Directory)。数据文件作为树叶结点，其它所有目录文件均作为树的中间结点。
- 树形目录结构具有检索效率高、允许重名、便于实现文件共享等一系列优点，因此被广泛使用，是目前广为流行的一种目录结构。

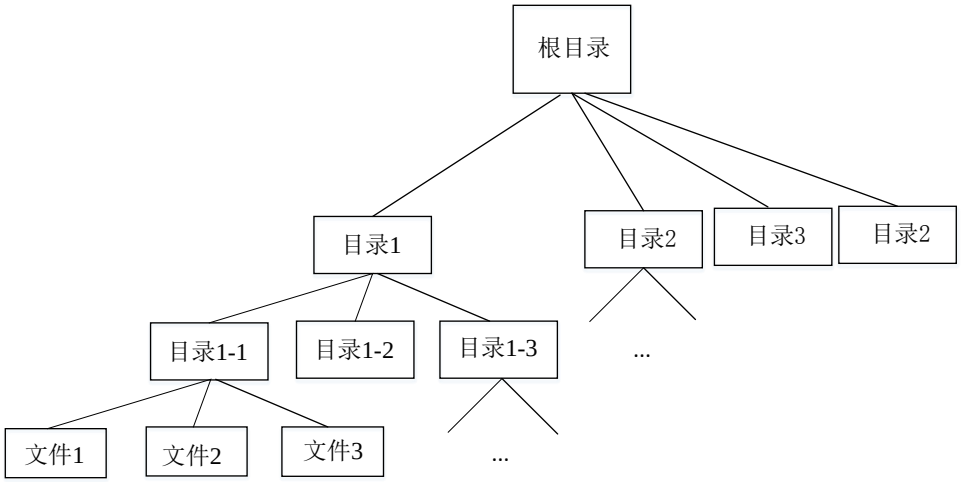


图9-3 树形目录结构

9.2.6 目录查询技术

有了文件目录，OS根据文件名查找文件会变得非常方便。具体查找过程为：

（1）系统利用用户提供的文件名，对文件目录进行查询，找出该文件的文件控制块或索引结点。

（2）根据找到的文件控制块或索引结点中所记录的文件盘块号，换算出文件在磁盘上的物理位置。

（3）启动磁盘，将所需文件读到内存。

对目录进行查询的方法有两种：线性检索法和Hash法。

9.2.6 目录查询技术

1.线性检索法（又称**顺序检索法**）

假设用户给定的文件路径名为d1/d2/d3…/dn/datafile,那么树形目录结构中采用线性检索法检索该文件的基本过程为：

（1）读入第一个文件分量名d1，用它与根目录文件中各个目录项的文件名顺序地进行比较，从中找出匹配者，并得到匹配项的索引结点编号，从对应的索引结点中获知d1目录文件所在的盘块号，启动磁盘，将相应盘块读入内存。

（2）对于d2~dn，以此类推。

（3）读入最后一个文件分量名datafile，用它与第n级目录文件中各个目录项的文件名顺序地进行比较，从中找到匹配项，从而得到该文件对应的索引结点编号，再从对应的索引结点中读出该文件的物理地址，目录查询成功结束。如果在上述查找过程中，发现任何一个文件分量名未能找到，则停止查找并返回“文件未找到”的出错信息。

9.2.6 目录查询技术

2. Hash方法

系统利用哈希函数，将用户提供的文件名转换为文件目录的索引值，再利用该索引值到目录中去查找，从而显著提高检索速度。

对进行文件名转换中可能出现的哈希“冲突”问题，可以采用以下解决方案：

（1）在利用Hash法索引查找目录时，如果目录表中相应的目录项是空的，则查找失败，表示系统中不存在指定的文件。

（2）如果目录项中的文件名与指定文件名相匹配，则查找成功，表示该目录项正是所要寻找的文件目录项，从该目录项中读取出该文件存放的物理地址即可。

（3）如果在目录表相应目录项中的文件名与指定文件名并不匹配，则查找失败，表示发生了“冲突”，这时必须将该Hash值再加上一个常数，形成新的索引值，再返回到第一步重新开始查找。

9.3 文件存储空间的分配与管理

- 9.3.1 文件存储空间的分配
- 9.3.2 磁盘空间管理

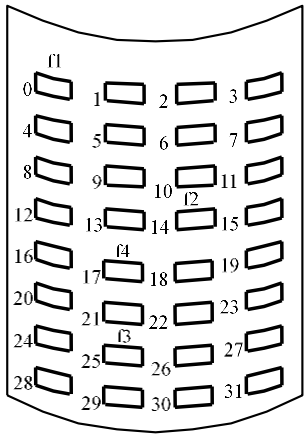
9.3.1 文件存储空间的分配

- 文件存储空间的分配方法主要有：连续分配、链接分配和索引分配三种。

1. 连续分配

连续分配（Continuous Allocation）要求为每一个文件分配一组相邻接的盘块。这种分配方式保证了逻辑文件中的记录顺序与存储器中文件占用盘块的顺序是一致的。

在采用连续分配方式时，可以把逻辑文件中的记录顺序地存储到相邻接的各个物理盘块中，这样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。为了使系统能找到文件在外存存放的具体位置，须在文件对应目录项的“物理地址”字段中，记录该文件所在的第一个盘块的盘块号和文件长度。



目录

file	start	length
f1	0	4
f2	14	3
f3	25	5
f4	17	2

1. 连续分配

连续分配方式的主要优点有：

（1）顺序访问容易。

（2）支持直接存取。

（3）访问速度快。

主要缺点如下：

（1）容易产生外部碎片。

（2）必须事先知道文件的长度。

（3）不便于文件的扩展。

2. 链接分配

把文件信息按照盘块大小的整数倍进行分段，各段分别存放到一些非连续的磁盘块中，每个盘块的最后设有链接指针，然后用指针将这些盘块按逻辑记录的顺序链接起来，就形成了文件的链式结构，由此所形成的物理文件称为链接文件。

链接分配采取离散分配方式，其主要优点如下：

（1）消除了磁盘的外部碎片，可以显著地提高外存空间的利用率。

（2）无需事先知道文件长度，方便文件的扩展。

（3）方便对文件进行增加、删除、修改操作。

2. 链接分配

链接分配方式根据链接指针的存放方式，又可分为隐式链接和显式链接两种。

（1）隐式链接结构

在该结构中，文件目录的每个目录项中，包含指向链接文件第一个盘块和最后一个盘块的指针，其他盘块号则由链接指针记录。

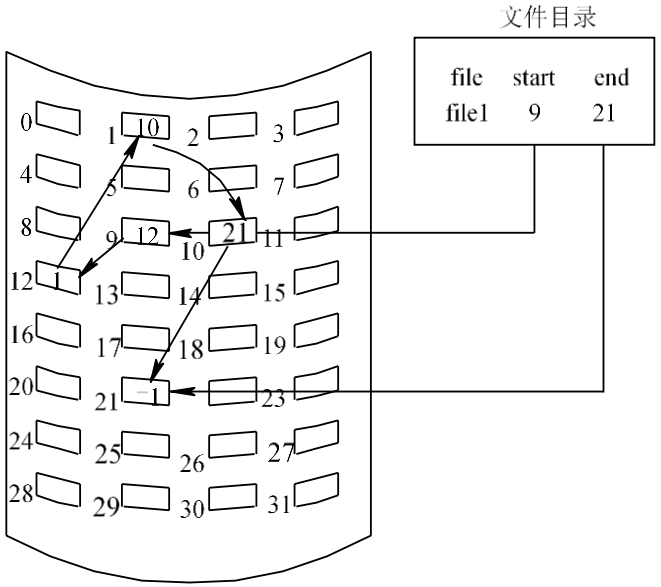


图9-5 文件的隐式链接结构示意图

该方式主要问题：它只适合于顺序访问，随机访问是极其低效的。为了提高检索速度和减少指针所占用的存储空间，可以将几个盘块组成一个簇，但却增大了内部碎片，改进有限。

2. 链接分配

（2）显式链接结构

显式链接是把用于链接文件各个盘块的指针，显式地存放在一张称为文件分配表FAT（File Allocation Table）的链接表中。表的序号是物理盘块号，FAT表项中存放的链接指针就是当前盘块链接的下一个盘块号，而每个文件的第一个盘块号作为文件地址被填入FCB的“物理地址”字段中。FAT一般较大，因此不宜保存在内存中，而是作为一个文件保存在磁盘上。

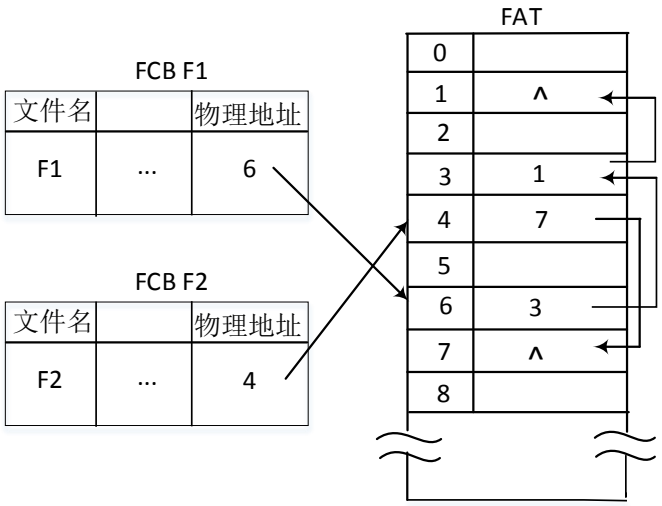


图9-6 文件的显示链接结构图

该方式主要问题：① 每次读取文件时，FAT需占用大量的内存空间。② 不支持高效的直接存取。对一个较大的文件进行读取时，要在FAT中查找许多盘块号，当这些盘块号恰好存放在FAT的同一个块中时，才能降低开销。

3. 索引分配

为每个文件分配一个索引块，把分配给该文件的所有盘块号，都登记在该索引块中。并在创建一个文件时，在文件FCB中登记该索引块的地址。

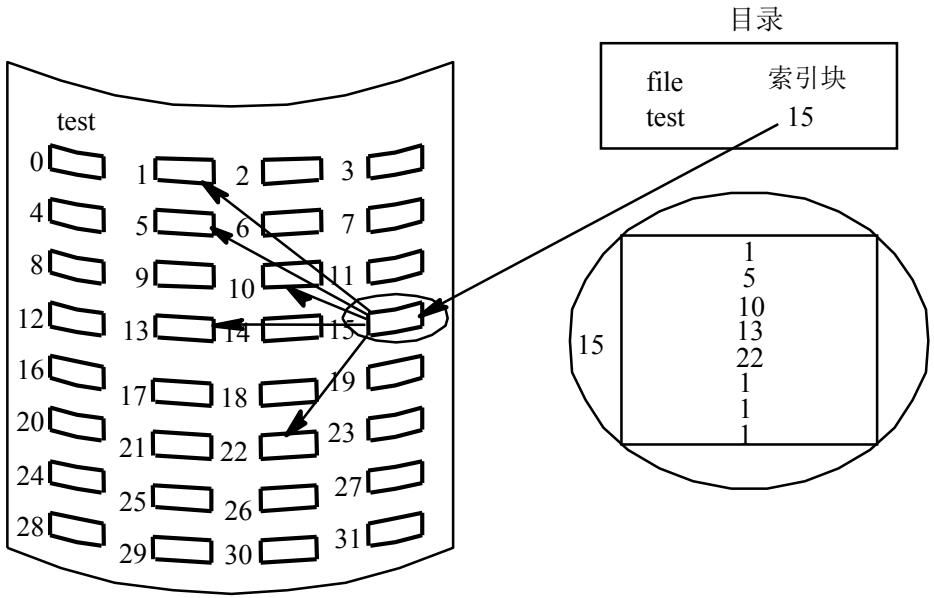


图9-7 文件的索引分配示意图

3. 索引分配

索引分配方式的优点主要有：

（1）既适合顺序存取，也支持随机存取。

（2）容易实现记录的插入、删除和修改。

（3）该分配方式不会产生外部碎片。

主要缺点是：

（1）可能占用较多的外存空间。每当建立一个文件时，便为之分配一个索引块，且将文件的所有盘块号登记在其中，这样无疑增加了存储空间的开销。

（2）采用此分配方式时，对于小文件来说，其索引块的利用率是非常低的。

（3）降低了文件存取的速度。在存取文件时，需要两次访问外存：第一次访问索引块，找到所访问的物理盘块的位置；第二次访问的才是要存取的物理块。

4. 混合索引结构

Unix中，采用了多级混合索引分配。它将直接寻址、一级索引、二级索引和三级索引融为一体，每个文件的索引结点中，使用13个地址项。其中，前10个地址项直接指出存放文件的盘块号，属于直接地址；第11个地址项指向一级索引块，属于一次间接地址；第12个和第13个地址项分别属于二次间接地址和三次间接地址。

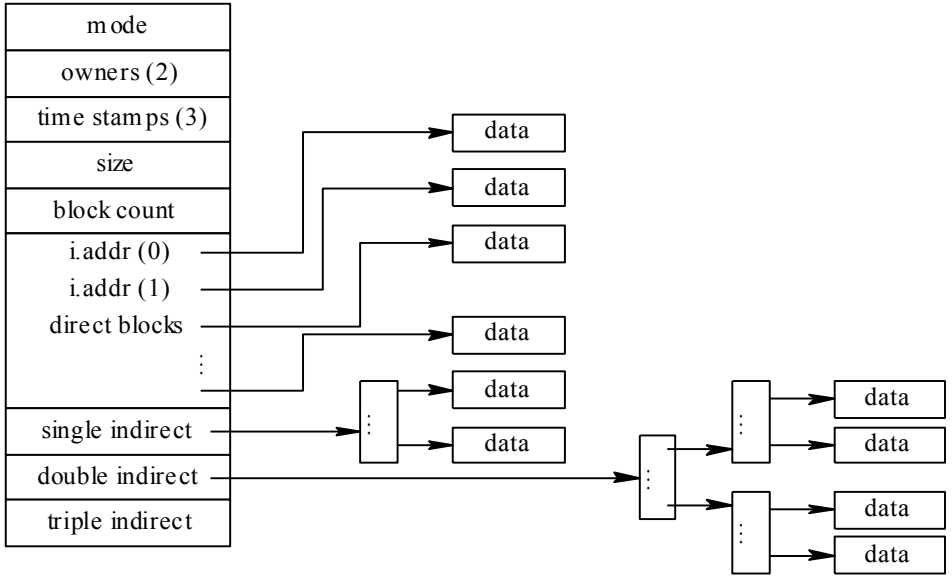


图9-8 Unix的混合索引结构

9.3.2 磁盘空间管理

1.空闲表法

空闲表法属于连续分配方式。它为每个文件分配一个连续的存储空间。系统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。空闲表中包括：序号、该空闲区的第一个盘块号、该区的空闲盘块数等信息。

序号	第一空闲盘块号	空闲盘块数
1	3	5
2	12	3
3	17	4
4	—	—

表9-2　空闲表

9.3.2 磁盘空间管理

2.空闲块链接法

是将所有空闲盘区链接成一条空闲链。根据构成链所用基本元素的不同，可以把链表分成两种形式：

（1）空闲盘块链(Free Disk Block Link)

优点：分配和回收过程简单；

缺点：空闲盘块链可能很长，要重复操作多次。

（2）空闲盘区链(Free Disk Area Link)

9.3.2 磁盘空间管理

3.位示图法

位示图是利用二进制的一位(bit)来表示磁盘中一个盘块的使用情况。磁盘上的所有盘块都可以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	1	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
.																
16									. . .							

图9-9 位示图

9.3.2 磁盘空间管理

盘块的分配过程：

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

(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。盘块号的计算公式为：b=n(i-1)+j

(3) 修改位示图：找到图中相应的位，将“0”改为“1”即可。

盘块的回收分为两步：

(1) 将回收盘块的盘块号转换成位于图中的位置号;

i=(b-1) div n +1

j=(b-1) mod n +1

(2) 修改位示图：找到图中相应的位，将“1”改为“0”即可。

9.3.2 磁盘空间管理

这种方法的主要优点：

(1) 从位示图中很容易找到一个或一组相邻接的空闲盘块。

(2) 由于位示图很小，占用空间少，因而可以将它保存在内存中，从而在每次进行盘区分配时，无需把磁盘分配表先读入内存，省掉许多磁盘的启动操作。因此，该方法常用于微型机和小型机中。

9.3.2 磁盘空间管理

4.成组链接法

在Unix系统中，是将空闲表和空闲块链两种方法相结合而形成的一种管理方法，称为成组链接法。它将空闲盘块分成若干组，把指向一组各空闲块的指针集中在一起，这样既方便查找，又可以减少为了修改指针而启动磁盘的次数。

(1) 空闲盘块号栈

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个盘块号)，以及栈中尚有的空闲盘块号数s.nfree=N。其中，S_free[0]是栈底，栈满时的栈顶为S_free[99]。由于栈是临界资源，每次只允许一个进程去访问，所以系统为栈设置了一把锁。

9.3.2 磁盘空间管理

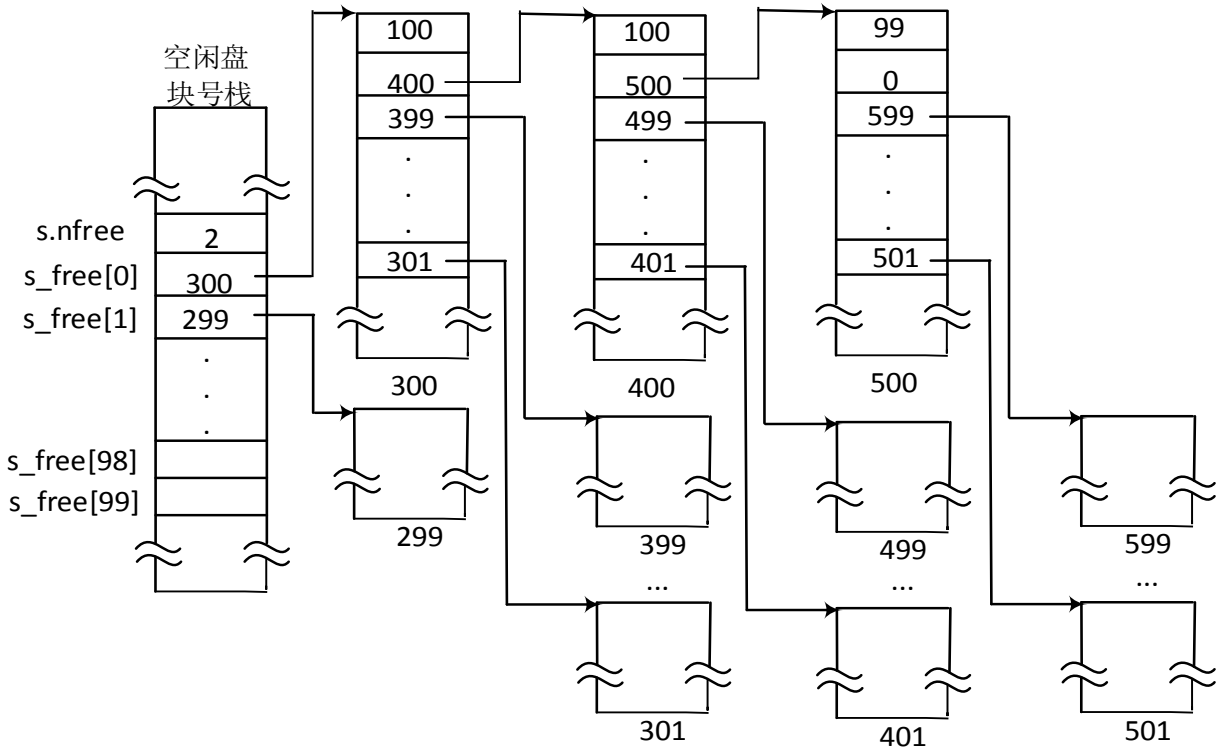


图9-12成组链接法中空闲盘块的组织

9.3.2 磁盘空间管理

(2) 空闲盘块的组织

① 磁盘文件区中的所有空闲盘块从后向前，每100个盘块分为一组（最后一组是99个盘块）。当然，最前面的一组可能不足100块，这一组的盘块号和盘块数就放在空闲盘块号栈中，作为当前可用的空闲盘块号。

② 将每一组含有的盘块总数和该组所有的盘块号记入其前一组的最后一个盘块的S_free[0]～S_free[99]中，其位置和格式与空闲盘块号栈里的相同。这样，由各组的最后一个盘块可以链成一条链。

③ 最末一组只有99个盘块，这些盘块号分别记入其前一组最后一个盘块的S_free[1]～S_free[99]中，而在S_free[0]中存放“0”，作为空闲盘块链的结束标志。

9.3.2 磁盘空间管理

(3) 空闲盘块的分配与回收

①检查空闲盘块号栈是否上锁，若已上锁，则进程等待；如未上锁，且栈顶指针>0，则系统直接弹出栈顶，把栈顶盘块号对应的盘块分配给用户。若栈顶指针=0，说明它是当前栈中最后一个盘块号。由于在该盘块号对应的盘块中记录有下一组可用的盘块号，因此，需启动磁盘，将栈底盘块号对应盘块的内容读入空闲盘块栈中，作为新的栈内容，这时可以把原栈底对应的盘块分配出去。

②分配一相应的缓冲区作为该盘块的缓冲区。

③把栈中的空闲盘块数即s.nfree减1并返回。

例如，若s.nfree原来的值为2，则被回收盘块的块号填入s_free[2]中，然后s.nfree加1变成3。但是，当栈中空闲盘块号数目已达100时，表示栈已满，这时要将现有栈中的100个盘块号，记入接下来新回收的盘块中，再将其盘块号作为新的栈底填入s_free[0]，s.nfree置为1。

9.4 文件的共享与保护

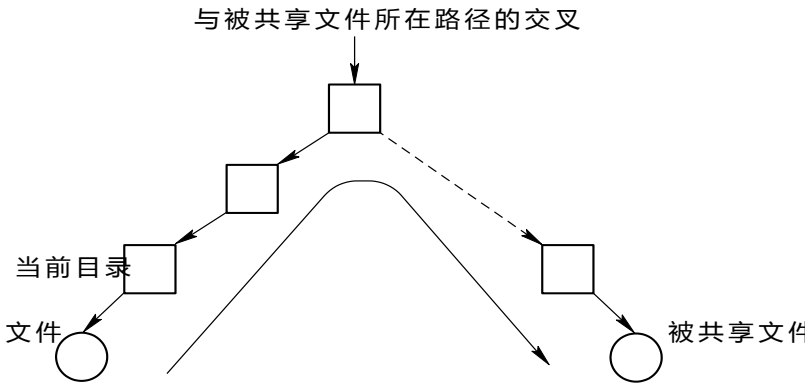
- 9.4.1 文件共享
- 9.4.2 文件保护

9.4.1 文件共享

1. 绕弯路法

绕弯路法是早期操作系统采用的一种文件共享方法。在该方法中，每个用户有一个“当前目录”，用户的所有操作都是相对于这个当前目录的。若用户希望共享的文件在当前目录下，则查当前目录表即可得到文件的外存地址，对文件进行访问；若不在当前目录下，则要通过一个“绕弯子”的路径，以查找文件的外存地址。

绕弯路法要花很多时间访问多级目录，因此搜索效率不高。



9.4.1 文件共享

2. 链接法

在相应目录表之间进行链接，即将一个目录中的表目直接指向被共享文件所在的目录，则被链接的目录以及子目录所包含的文件都为共享的对象。

链接法的另一种形式是采用基本文件目录和符号文件目录。如果一个用户要共享另一个文件，只需在他的符号文件目录中增加一个目录项，填上被共享文件的符号名和此文件的内部标识即可。

9.4.1 文件共享

3. 基于索引结点共享法

有了索引结点，一个文件信息在磁盘上就分为三部分：目录项、索引结点和文件本身。在文件目录中只设置文件名及指向相应索引结点的指针，索引结点中还有一个链接计数count，用于表示链接到本索引结点上的用户目录项的数目。

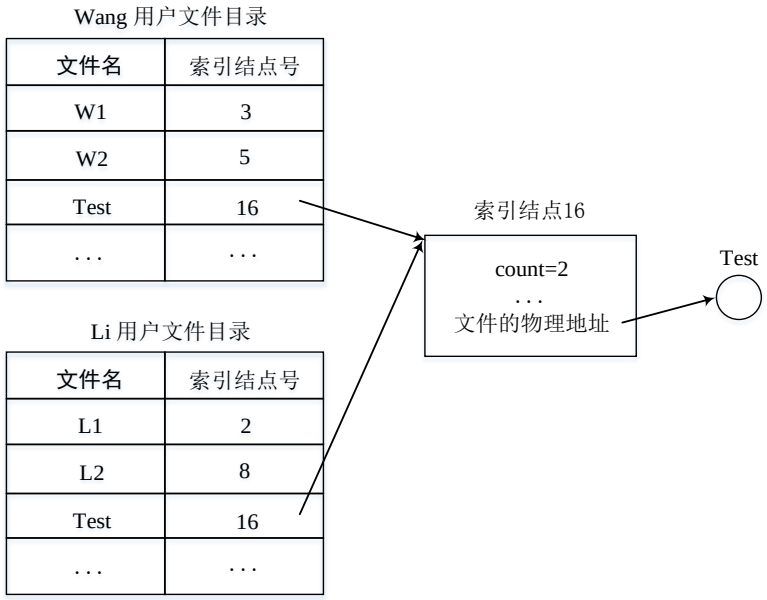


图9-13 基于索引结点的共享方式

问题：如果用户删除共享文件时，不一定物理删除该文件，有时仅仅是删除一个链接，可能形成“悬空指针”。

9.4.1 文件共享

4. 基于符号链共享法

是网络系统中常用的文件共享方法。用户Li要共享用户Wang的文件Test，系统不是直接让Li的目录项指向文件Test的索引结点，而是由系统创建一个LINK类型的新文件Test’，其内容是Test的路径名（包括网络地址和文件在本机上的路径）。当用户Li发出访问该共享文件的请求去读Test’时，OS将根据文件内容访问Test，实现文件的共享。

这种共享方法不会出现“悬空指针”。但是，众多的LINK文件虽然内容简单，但都各有自己的索引结点，加大了存储空间的开销。因此，这种共享方法在单机系统中不适用，但用于网络系统就很方便。

9.4.2 文件保护

1.防止系统故障造成破坏

系统偶尔发生软件、硬件故障是难以避免的，当故障发生时，文件管理系统应提供一定的措施使文件尽可能地不被破坏。

（1）定时转储。

（2）建立副本。

（3）后备系统。

（4）磁盘容错技术。磁盘容错（System Fault Tolerance，SFT）就是通过设置系统冗余部件的方法来提供文件保护。共分为三级容错措施：

① SFT-I是低级磁盘容错技术。

② SFT-Ⅱ是中级磁盘容错措施。

③ SFT-Ⅲ是高级磁盘容错技术，典型的是廉价磁盘冗余阵列（Redundant Arrays of Inexpensive Disk，RAID。

9.4.2 文件保护

2.防止人为因素造成的破坏

要防止人为因素给文件带来的破坏，可以设置不同用户对不同目录、不同文件的使用权限。

（1）基于目录的存取权限。

（2）基于文件的存取权限。

（3）存取控制矩阵。把各用户对文件和目录的使用权，用二维矩阵的形式来表示。当某个用户提出使用某个文件的请求时，系统按存取控制矩阵进行权限核对，只有在核对相符后才允许使用该文件。

小 结

关于文件和文件系统的概念；

了解文件逻辑结构；

了解目录管理，重点是关于索引节点的内容；

了解几种目录结构和查询技术；

了解文件共享和保护；