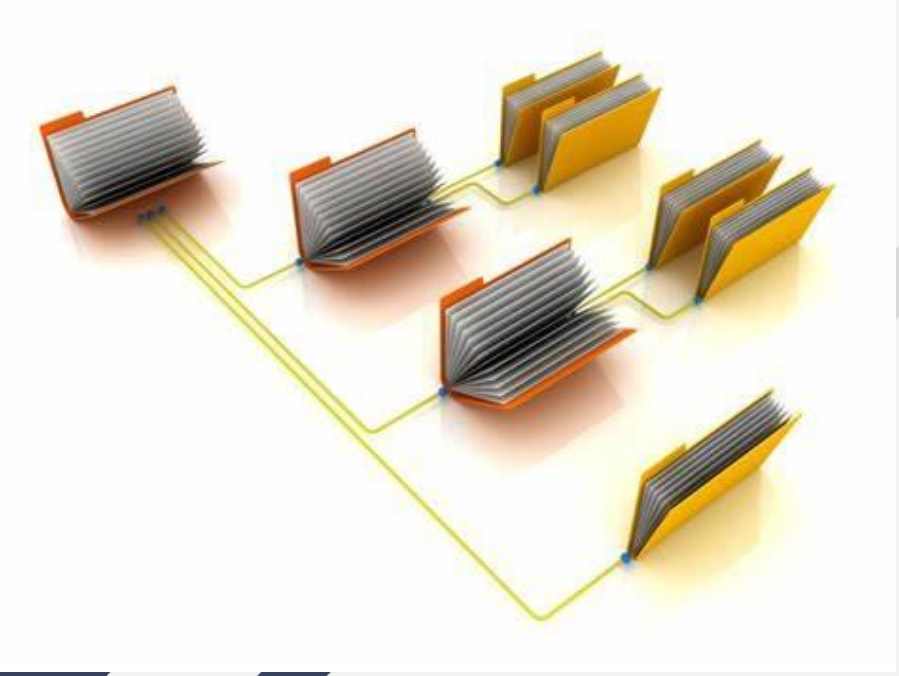


2022

操作系统A

第七章
文件管理

赵谷雨 \ 信息科学与工程学院

Email: gaiazhao@ysu.edu.cn

目录

CONTENTS

第七章 文件管理

7.1 文件和文件系统

7.1 文件和文件系统

文件系统的管理功能是将其管理的程序和数据通过组织为一系列文件的方式实现的。而**文件**则是指具有文件名的若干相关元素的集合。元素通常是记录，而记录又是一组有意义的数据项的集合。可见，基于文件系统的概念，可以把数据组成分为数据项、记录和文件三级。

7.1.1 数据项、记录和文件

1. 数据项

在文件系统中，数据项是最低级的数据组织形式，可把它分成以下两种类型：

(1) 基本数据项。

(2) 组合数据项。

2. 记录

记录是一组相关数据项的集合，用于描述一个对象在某方面的属性。一个记录应包含哪些数据项，取决于需要描述对象的哪个方面。由于对象所处的环境不同可把他作为不同的对象。

7.1.1 数据项、记录和文件

3. 文件

文件是指由创建者所定义的、具有文件名的一组相关元素的集合，可分为有结构文件和无结构文件两种。

文件属性：

文件类型

文件长度

文件物理位置

文件建立时间

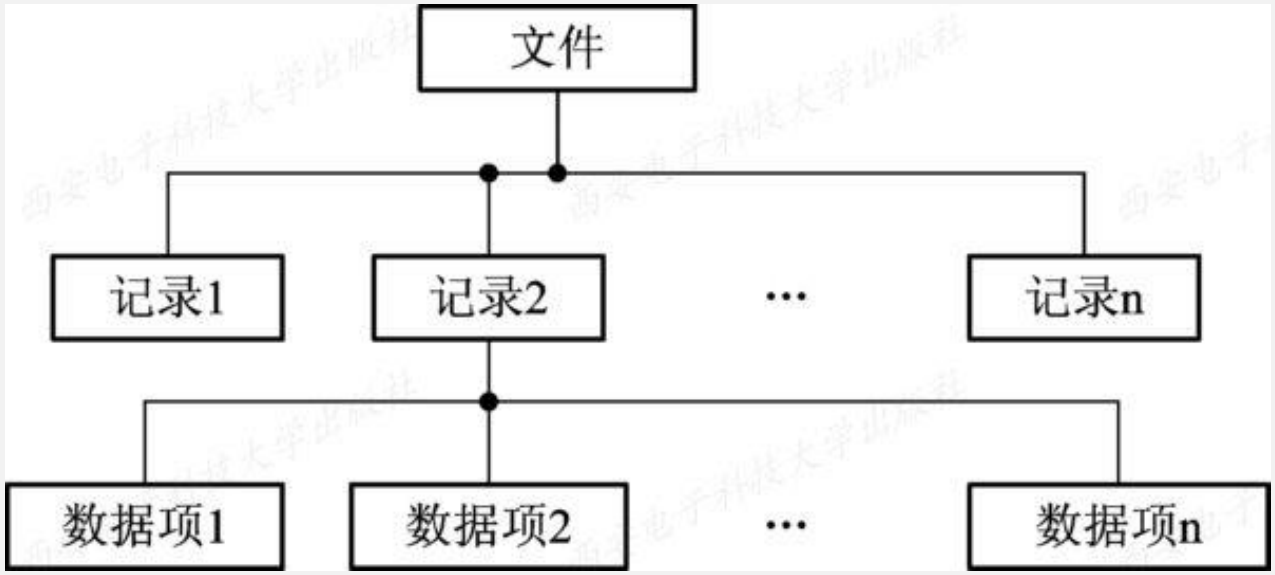


图7-1 文件、记录和数据项之间的层次关系

7.1.2 文件名和类型

1. 文件名和扩展名

(1) 文件名。

长度可能受限；特殊字符可能不能使用；

区分大小写和不区分大小写；

(2) 扩展名。

扩展名是添加在文件名后面的若干附加字符，又称为后缀名，用于指示文件的类型。在大多数系统中，用圆点“.”将文件名和扩展名分开。

7.1.2 文件名和类型

2. 文件类型

1) 按用途分类

根据文件的性质和用途的不同，可将文件分为三类：

(1) 系统文件，这是指由系统软件构成的文件。大多数的系统文件只允许用户调用，但不允许用户去读，更不允许修改；有的系统文件不直接对用户开放。

(2) 用户文件，指由用户的源代码、目标文件、可执行文件或数据等所构成的文件。用户将这些文件委托给系统保管。

(3) 库文件，这是由标准子例程及常用的例程等所构成的文件。这类文件允许用户调用，但不允许修改。

7.1.2 文件名和类型

2. 文件类型

2) 按**文件中数据的形式**分类

按这种方式分类，也可把文件分为三类：

(1) 源文件，这是指由源程序和数据构成的文件。通常，由终端或输入设备输入的源程序和数据所形成的文件都属于源文件。它通常是由ASCII码或汉字所组成的。

(2) 目标文件，这是指把源程序经过编译程序编译过，但尚未经过链接程序链接的目标代码所构成的文件。目标文件所使用的后缀名是“.obj”。

(3) 可执行文件，这是指把编译后所产生的目标代码经过链接程序链接后所形成的文件。其后缀名是 .exe。

7.1.2 文件名和类型

2. 文件类型

3) 按**存取控制属性**分类

根据系统管理员或用户所规定的存取控制属性，可将文件分为三类：

(1) 只执行文件，该类文件只允许被核准的用户调用执行，不允许读和写。

(2) 只读文件，该类文件只允许文件主及被核准的用户去读，不允许写。

(3) 读写文件，这是指允许文件主和被核准的用户去读或写的文件。

7.1.2 文件名和类型

2. 文件类型

4) 按**组织形式和处理方式**分类

根据文件的组织形式和系统对其处理方式的不同，可将文件分为三类：

(1) 普通文件。

(2) 目录文件。

(3) 特殊文件。

7.1.4 文件操作

1. 最基本的文件操作

最基本的文件操作包含下述内容：

(1) 创建文件。

(2) 删除文件。

(3) 读文件。

(4) 写文件。

(5) 设置文件的读/写位置。

7.1.4 文件操作

2. 文件的“打开”和“关闭”操作

所谓“打开”，是指将指名文件的属性（包括该文件在外存上的物理位置），从外存拷贝到内存打开文件表的一个表目中，并将该表目的编号（或称为索引号）返回给用户。当用户再次向系统发出文件操作请求时，系统根据用户提供的索引号可以直接在打开文件表中查到文件信息。

如果用户不再需要对该文件试试相应的操作，可利用“关闭”系统调用关闭文件，OS将该文件从打开文件表中的表目上删掉。

7.1.4 文件操作

3. 其它文件操作

OS为用户都提供了一系列文件操作的系统调用，其中**最常用的一类是有关对文件属性的操作**，即允许用户直接设置和获得文件的属性，如改变已存文件的文件名、改变文件的拥有者(文件主)、改变对文件的访问权，以及查询文件的状态(包括文件类型、大小和拥有者以及对文件的访问权等)。**另一类是有关目录的操作**，如创建一个目录，删除一个目录，改变当前目录和工作目录等。此外，还有用于实现文件共享的系统调用，以及用于对文件系统进行操作的系统调用等。

第七章 文件管理

7.2 文件的逻辑结构

7.2 文件的逻辑结构

(1) 文件的逻辑结构(File Logical Structure)。

(2) 文件的物理结构，又称为文件的存储结构。

7.2.1 文件逻辑结构的类型

对文件逻辑结构所提出的基本要求，首先是有助于提高对文件的检索速度，即在将大批记录组成文件时，应采用一种有利于提高检索记录速度和效率的逻辑结构形式。其次是该结构应方便对文件进行修改，即便于在文件中增加、删除和修改一个或多个记录。第三是降低文件存放在外存上的存储费用，即尽量减少文件占用的存储空间，不要求大片的连续存储空间。

7.2.1 文件逻辑结构的类型

1. 按文件是否有结构分类

1) 有结构文件

(1) 定长记录。　(2) 变长记录。

2) 无结构文件

信息管理系统和数据库系统中，广泛采用有结构的文件形式；在系统中运行的大量的源程序、可执行文件、库函数等，采用无结构的文件形式，即流式文件。其文件的长度是以字节为单位的。对流式文件的访问，则是利用读、写指针来指出下一个要访问的字符。可以把流式文件看做是记录式文件的一个特例：一个记录仅有一个字节。

7.2.1 文件逻辑结构的类型

2. 按文件的组织方式分类

根据文件的组织方式，可把有结构文件分为三类：

(1) 顺序文件。

(2) 索引文件。

(3) 索引顺序文件。

7.2.2 顺序文件(Sequential File)

1. 顺序文件的排列方式

在顺序文件中的记录，可以按照各种不同的顺序进行排列。一般地，可分为两种情况：

(1) **串结构。**存入时间

(2) **顺序结构。**关键字

7.2.2 顺序文件(Sequential File)

2. 顺序文件的优缺点

顺序文件的最佳应用场合是在对文件中的记录进行批量存取时(即每次要读或写一大批记录)。所有逻辑文件中顺序文件的存取效率是最高的。此外，对于顺序存储设备(如磁带)，也只有顺序文件才能被存储并能有效地工作。

7.2.4 索引文件(Index File)

1. 按关键字建立索引

定长记录的文件可以通过简单的计算，很容易地实现随机查找。但变长记录文件查找一个记录必须从第一个记录查起，一直顺序查找到目标记录为止，耗时很长。

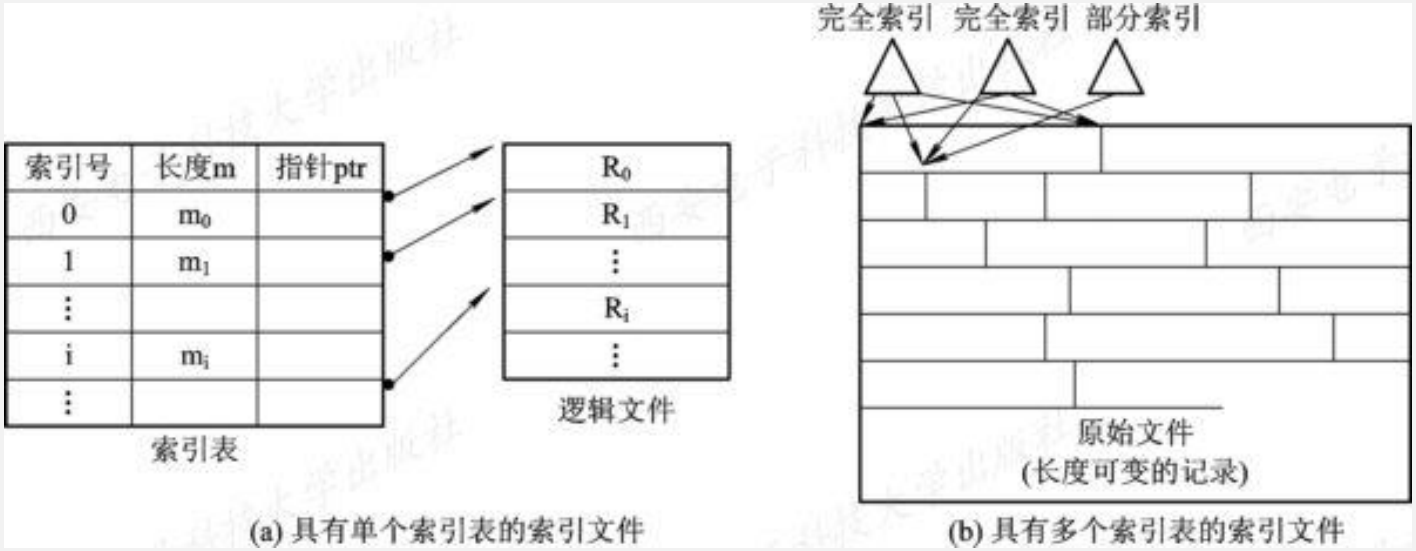


图7-4 具有单个和多个索引表的索引文件

7.2.6 直接文件和哈希文件

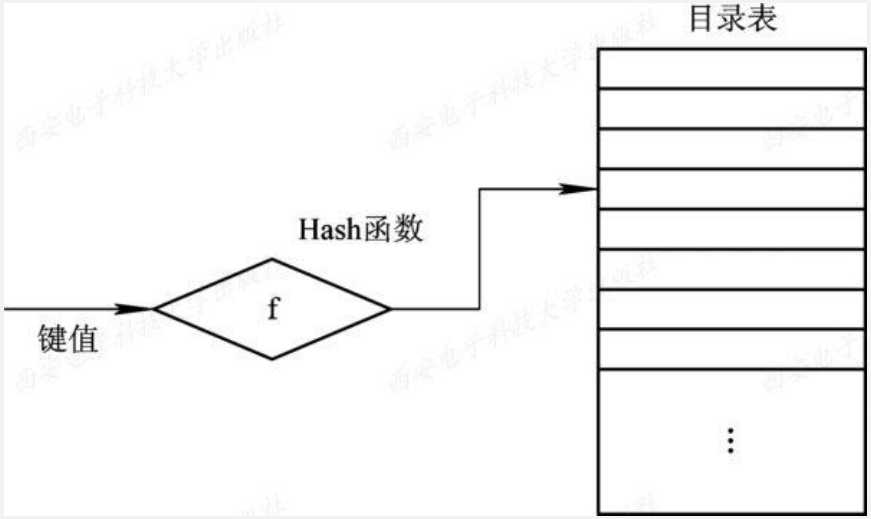
1. 直接文件

采用前述几种文件结构对记录进行存取时，都须利用给定的记录键值，先对线性表或链表进行检索，以找到指定记录的物理地址。然而对于直接文件，则可根据给定的关键字直接获得指定记录的物理地址。换而言之，**关键字本身就决定了记录的物理地址**。

7.2.6 直接文件和哈希文件

2. 哈希(Hash)文件

这是目前应用最为广泛的一种直接文件。它利用Hash函数(或称散列函数)可将关键字转换为相应记录的地址。但为了能实现文件存储空间的动态分配，通常由Hash函数所求得的并非是相应记录的地址，而是指向某一目录表相应表目的指针，该表目的内容指向相应记录所在的物理块，如图7-6所示。



第七章 文件管理

7.3 文件目录

7.3 文件目录

文件目录是一种数据结构，用于标识系统中的文件及其物理地址，供检索时使用。目录管理的要求：

(1) 实现“按名存取”。

(2) 提高对目录的检索速度。

(3) 文件共享。

(4) 允许文件重名。

7.3.1 文件控制块和索引结点

1. 文件控制块FCB(File Control Block)

为了能对系统中的大量文件施以有效的管理，在文件控制块中，通常应含有三类信息，即基本信息、存取控制信息及使用信息。

1） 基本信息类

(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。

2） 存取控制信息类

存取控制信息类包括文件主的存取权限、核准用户的存取权限以及一般用户的存取权限。

7.3.1 文件控制块和索引结点

1. 文件控制块FCB(File Control Block)

为了能对系统中的大量文件施以有效的管理，在文件控制块中，通常应含有三类信息，即基本信息、存取控制信息及使用信息。

3）使用信息类

使用信息类包括文件的建立日期和时间、文件上一次修改的日期和时间，以及当前使用信息。这些信息包括当前已打开该文件的进程数，是否被其它进程锁住，文件在内存中是否已被修改但尚未拷贝到盘上等。

7.3.1 文件控制块和索引结点

2. 索引结点

1) 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，文件目录可能要占用大量的盘块。在查找目录的过程中，必须先将存放目录文件的第一个盘块中的目录调入内存，然后将用户所给定的文件名，与目录项中的文件名逐一比较。若未找到指定文件，还需要将下一盘块的目录项调入内存。

7.3.1 文件控制块和索引结点

2. 索引结点

2) 磁盘索引结点

这是存放在磁盘上的索引结点。每个文件有唯一的一个磁盘索引结点，它主要包括以下内容：

(1) 文件主标识符，即拥有该文件的个人或小组标识符；

(2) 文件类型，包括正规文件、目录文件或特别文件；

(3) 文件存取权限，指各类用户对该文件的存取权限；

(4) 文件物理地址，每一个索引结点中含有13个地址项，即iaddr(0)～iaddr(12)，它们以直接或间接方式给出数据文件所在盘块的编号；

7.3.1 文件控制块和索引结点

2. 索引结点

2) 磁盘索引结点

这是存放在磁盘上的索引结点。每个文件有唯一的一个磁盘索引结点，它主要包括以下内容：

(5) 文件长度，指以字节为单位的文件长度；

(6) 文件连接计数，表明在本文件系统中所有指向该(文件的)文件名的指针计数；

(7) 文件存取时间，指出本文件最近被进程存取的时间、最近被修改的时间及索引结点最近被修改的时间。

7.3.2 简单的文件目录

1. 单级文件目录

这是最简单的文件目录。在整个文件系统中只建立一张目录表，每个文件占一个目录项，目录项中含文件名、文件扩展名、文件长度、文件类型、文件物理地址以及其它文件属性。此外，为表明每个目录项是否空闲，又设置了一个状态位。单级文件目录如图7-9所示。

文件名	扩展名	文件长度	物理地址	文件类型	文件说明	状态位	
文件名 1							
文件名 2							
文件名 3							

图7-9 单级文件目录

7.3.2 简单的文件目录

2. 两级文件目录

为了克服单级文件目录所存在的缺点，可以为**每一个用户再建立一个单独的用户文件目录UFD(User File Directory)**。这些文件目录具有相似的结构，它由用户所有文件的文件控制块组成。此外，**在系统中再建立一个主文件目录MFD(Master File Directory)**；在主文件目录中，每个用户目录文件都占有一个目录项，其目录项中包括用户名和指向该用户目录文件的指针。

7.3.2 简单的文件目录

2. 两级文件目录

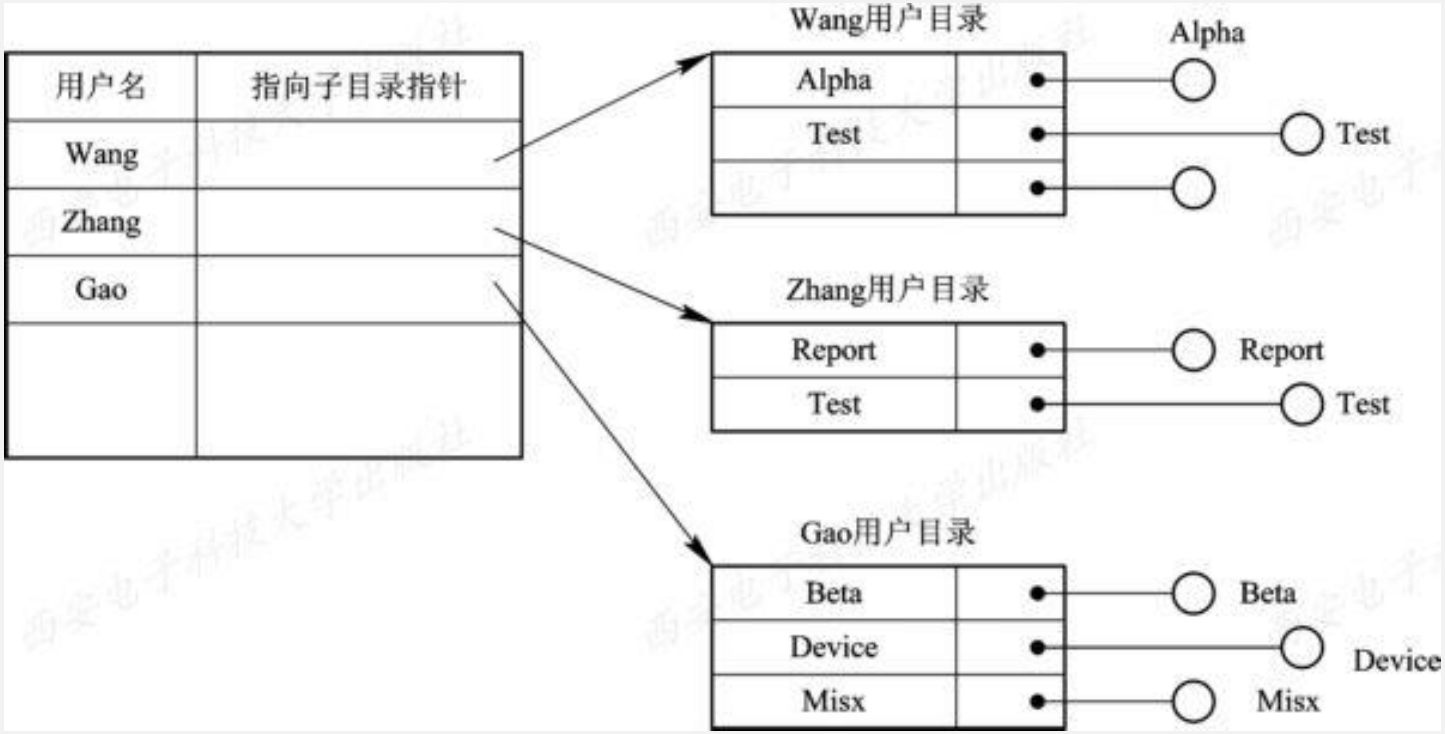


图7-10 两级文件目录

7.3.3 树形结构目录(Tree-Structured Directory)

1. 树形目录

在现代OS中，最通用且实用的文件目录无疑是树形结构目录。它可以明显地提高对目录的检索速度和文件系统的性能。主目录在这里被称为根目录，在每个文件目录中，只能有一个根目录，每个文件和每个目录都只能有一个父目录。把数据文件称为树叶，其它的目录均作为树的结点，或称为子目录。图7-11示出了树形结构目录。

7.3.3 树形结构目录(Tree-Structured Directory)

1. 树形目录

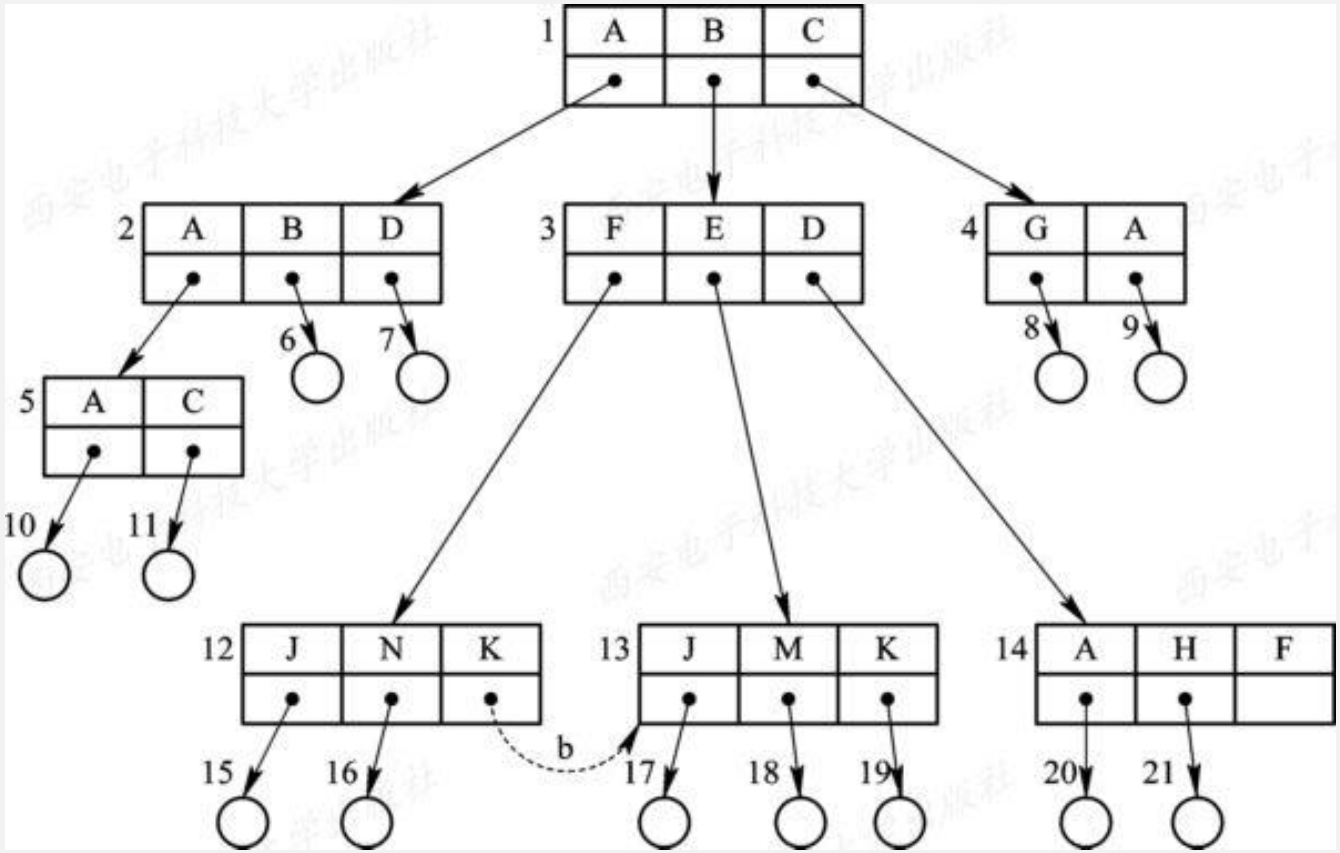


图7-11 多级目录结构

7.3.3 树形结构目录(Tree-Structured Directory)

2. 路径名和当前目录

1) 路径名(path name)

在树形结构目录中，从根目录到任何数据文件都只有一条唯一的通路。在该路径上，从树的根(即主目录)开始，把全部目录文件名与数据文件名依次地用“/”连接起来，即构成该数据文件唯一的路径名。

2) 当前目录(Current Directory)

当一个文件系统含有许多级时，每访问一个文件，都要使用从树根开始，直到树叶(数据文件)为止的、包括各中间节点(目录)名的全路径名。

7.3.3 树形结构目录(Tree-Structured Directory)

3. 目录操作

(1) 创建目录。

(2) 删除目录。

① 不删除非空目录。

② 可删除非空目录。

(3) 改变目录。

(4) 移动目录。

(5) 链接(Link)操作。

(6) 查找。

第七章 文件管理

7.4 文件共享

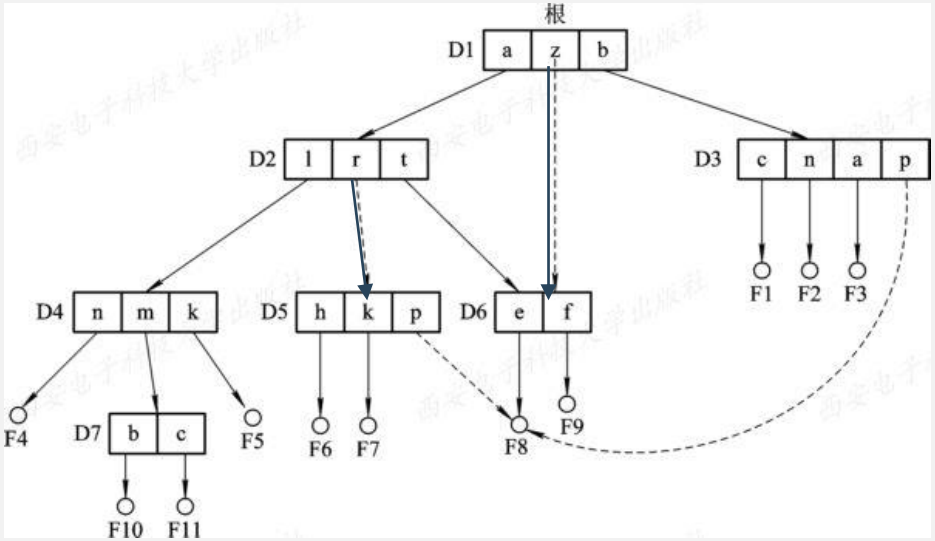
7.4 文件共享

在现代计算机系统中，必须提供文件共享手段，即指系统应允许多个用户(进程)共享同一份文件。这样，在系统中只需保留该共享文件的一份副本。如果系统不能提供文件共享功能，就意味着凡是需要该文件的用户，都须各自备有此文件的副本，显然这会造成对存储空间的极大浪费。

7.4.1 基于有向无循环图实现文件共享

2. 利用索引结点

引用索引结点，即诸如文件的物理地址及其它的文件属性等信息，不再是放在目录项中，而是放在索引结点中。在文件目录中只设置文件名及指向相应索引结点的指针，如图7-14所示。



7.4.1 基于有向无循环图实现文件共享

2. 利用索引结点

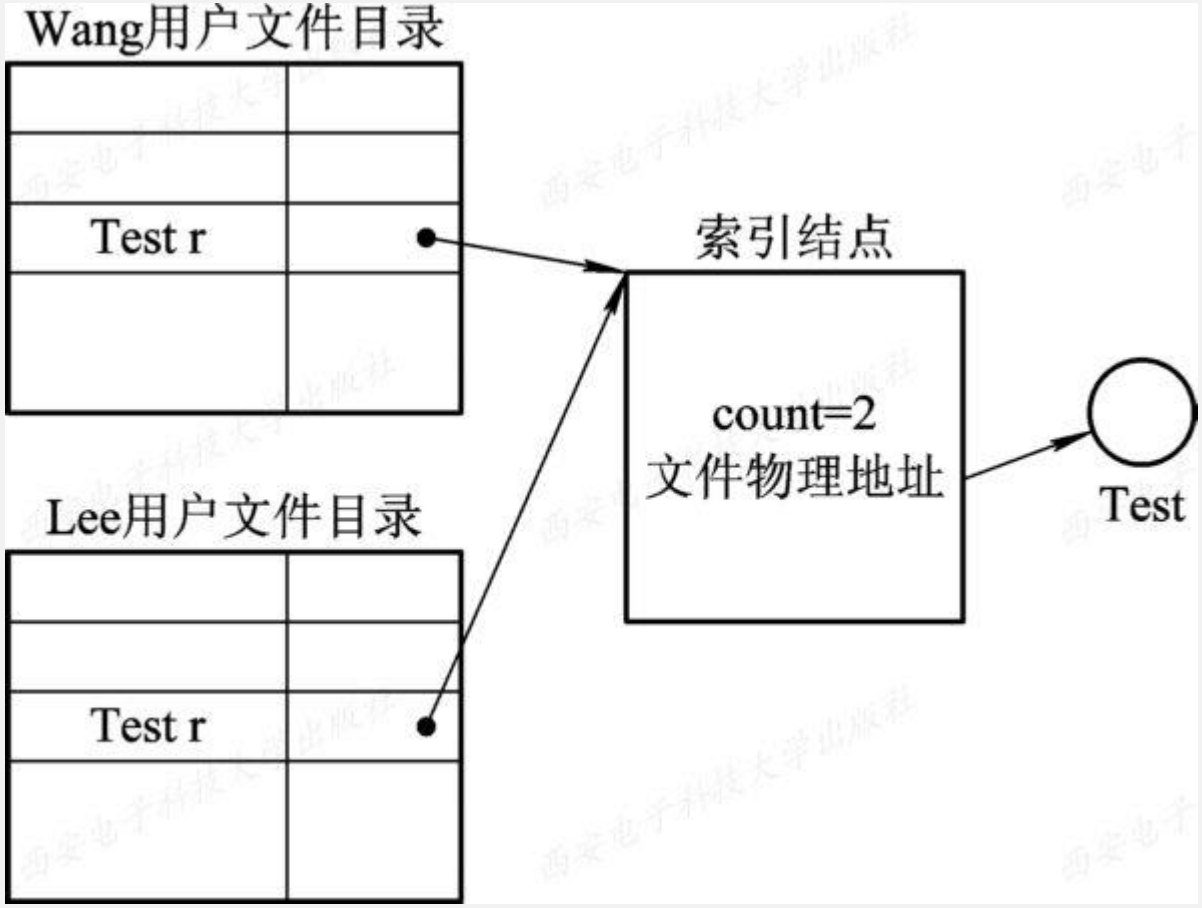


图7-14 基于索引结点的共享方式

7.4.1 基于有向无循环图实现文件共享

2. 利用索引结点

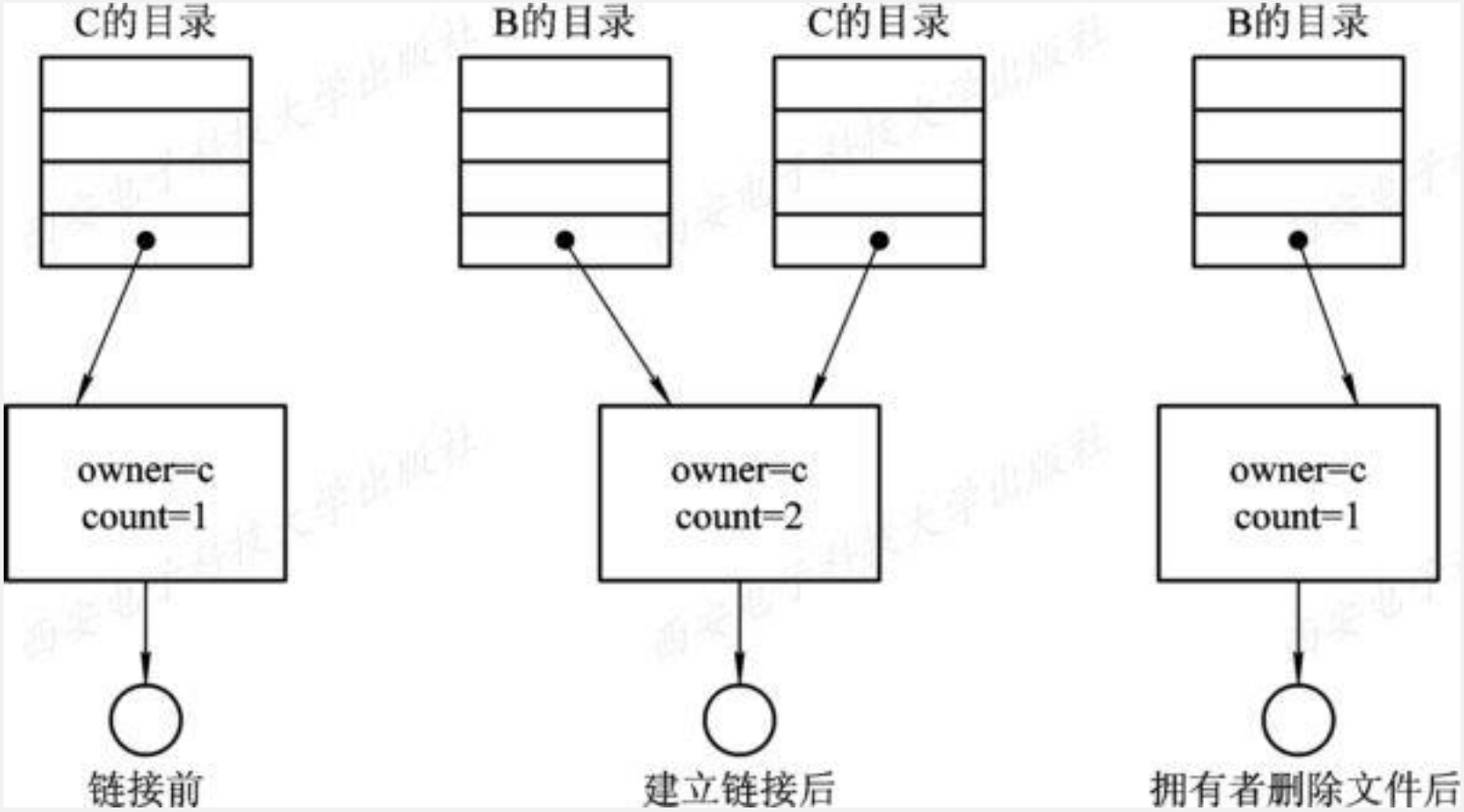


图7-15 进程B链接前后的情况

7.4.2 利用符号链接实现文件共享

1. 利用符号链接(Symbolic Linking)的基本思想

利用符号链接实现文件共享的基本思想，是允许一个文件或子目录有多个父目录，但其中仅有一个作**为主(属主)父目录**，其它的几个父目录都是通过符号链接方式与之相链接的(**简称链接父目录**)。

7.4.2 利用符号链接实现文件共享

1. 利用符号链接(Symbolic Linking)的基本思想

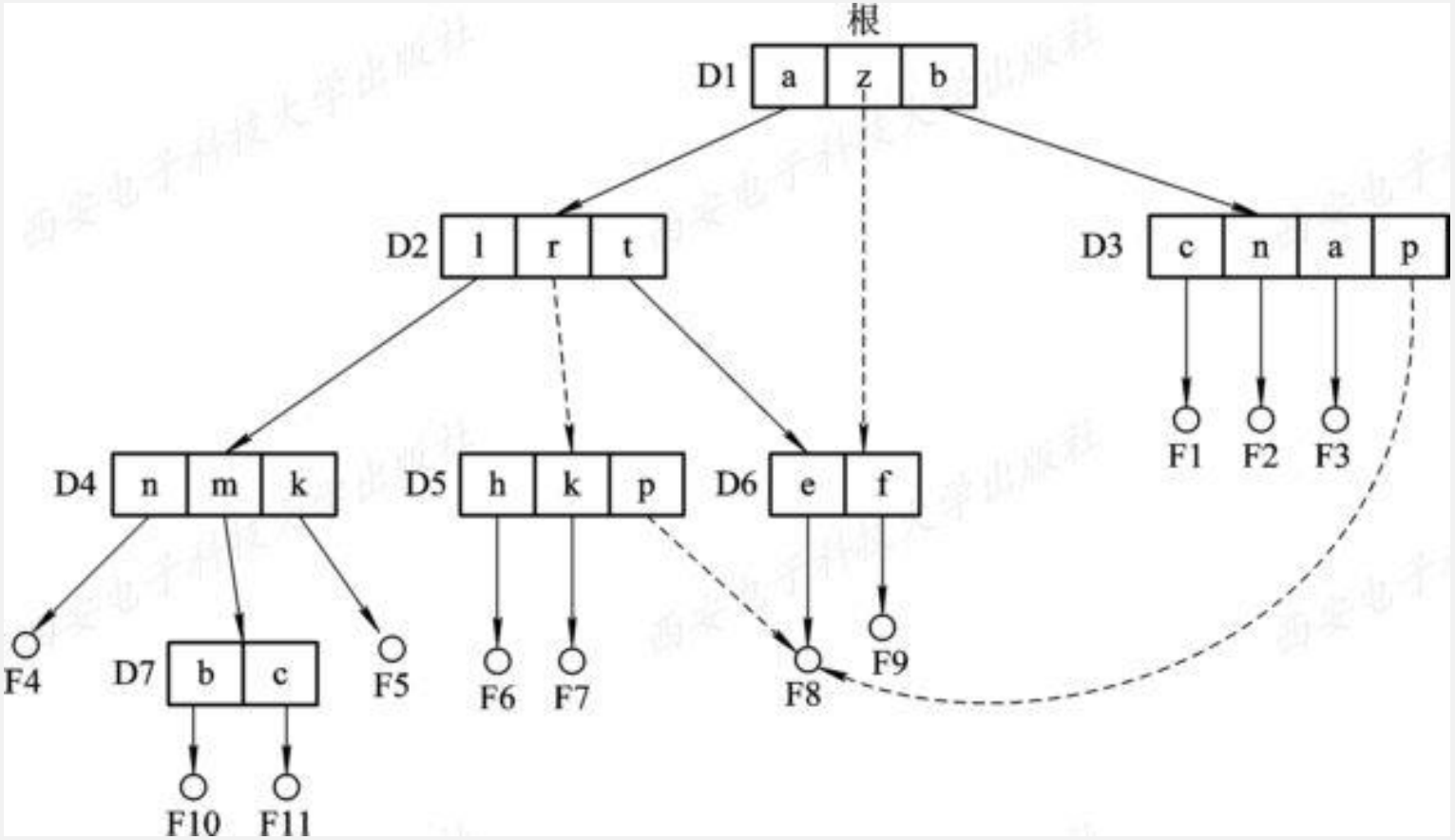


图7-16 使用符号链接的目录层次

7.4.2 利用符号链接实现文件共享

2. 如何利用符号链实现共享

为使链接父目录D5能共享文件F，可以由系统创建一个LINK类型的新文件，也取名为F，并将F写入链接父目录D5中，以实现D5与文件F8的链接。在新文件F中只包含被链接文件F8的路径名。这样的链接方法被称为符号链接。新文件F中的路径名则只被看做是符号链。当用户通过D5访问被链接的文件F8，且正要读LINK类新文件时，此要求将被OS截获，OS根据新文件中的路径名去找到文件F8，然后对它进行读(写)，这样就实现了用户B对文件F的共享。

7.4.2 利用符号链接实现文件共享

3. 利用符号链实现共享的优点

在利用符号链方式实现文件共享时，只是文件主才拥有指向其索引结点的指针；而共享该文件的其他用户则只有该文件的路径名，并不拥有指向其索引结点的指针。这样，也就不会发生在文件主删除一共享文件后留下一悬空指针的情况。当文件的拥有者把一个共享文件删除后，如果其他用户又试图通过符号链去访问一个已被删除的共享文件，则会因系统找不到该文件而使访问失败，于是再将符号链删除，此时不会产生任何影响。

7.4.2 利用符号链接实现文件共享

4. 利用符号链的共享方式存在的问题

利用符号链的共享方式也存在着一些问题：当其他用户去读共享文件时，系统是根据给定的文件路径名逐个分量(名)地去查找目录，直至找到该文件的索引结点。因此，在每次访问共享文件时，都可能要多次地读盘。这使每次访问文件的开销甚大，且增加了启动磁盘的频率。此外，要为每个共享用户建立一条符号链，而由于链本身实际上是一个文件，尽管该文件非常简单，却仍要为它配置一个索引结点，这也要耗费一定的磁盘空间。

第七章 文件管理

7.5 文件保护

7.5 文件保护

在现代计算机系统中，存放了越来越多的宝贵信息供用户使用，给人们带来了极大的好处和方便，但同时也有着潜在的不安全性。影响文件安全性的主要因素有：

(1) 人为因素。

(2) 系统因素。

(3) 自然因素。

为了确保文件系统的安全性，可针对上述原因而采取三方面的措施：

(1) **通过存取控制机制**，防止由人为因素所造成的文件不安全性。

(2) 采取系统容错技术，防止系统部分的故障所造成的文件的不安全性。

(3) 建立后备系统，防止由自然因素所造成的不安全性。

7.5.1 保护域(Protection Domain)

1. 访问权

为了对系统中的对象加以保护，应由系统来控制进程对对象的访问。对象可以是硬件对象，如磁盘驱动器、打印机；也可以是软件对象，如文件、程序。对对象所施加的操作也有所不同，如对文件可以是读，也可以是写或执行操作。我们**把一个进程能对某对象执行操作的权力，称为访问权(Access right)，用有序对（对象名，权集）表示**。

7.5.1 保护域(Protection Domain)

2. 保护域

为了对系统中的资源进行保护而引入了保护域的概念，**保护域简称为“域”。“域”是进程对一组对象访问权的集合，进程只能在指定域内执行操作。**这样，“域”也就规定了进程所能访问的对象和能执行的操作。

图7-17 三个保护域

7.5.1 保护域(Protection Domain)

3. 进程和域间的静态联系

在进程和域之间可以一一对应，即一个进程只联系着一个域。这意味着，在进程的整个生命期中，其可用资源是固定的，我们把这种域称为“静态域”。在这种情况下，进程运行的全过程都是受限于同一个域，这将会使赋予进程的访问权超过了实际需要。

7.5.1 保护域(Protection Domain)

4. 进程和域间的动态联系方式

在进程和域之间，也可以是一对多的关系，即一个进程可以联系着多个域。在此情况下，可将进程的运行分为若干个阶段，其每个阶段联系着一个域，这样便可根据运行的实际需要来规定在进程运行的每个阶段中所能访问的对象。

7.5.2 访问矩阵

1. 基本的访问矩阵

我们可以**利用一个矩阵来描述系统的访问控制，并把该矩阵称为访问矩阵(Access Matrix)。**访问矩阵中的行代表域，列代表对象，矩阵中的每一项是由一组访问权组成的。因为对象已由列显式地定义，故可以只写出访问权而不必写出是对哪个对象的访问权，每一项访问权access(i, j)定义了在域Di中执行的进程能对对象Qj所施加的操作集。

7.5.2 访问矩阵

1. 基本的访问矩阵

域 \ 对象	F_1	F_2	F_3	F_4	F_5	F_6	Printer 1	Plotter 2
D_1	R	R, W						
D_2			R	R, W, E	R, W		W	
D_3						R, W, E	W	W

图7-18 一个访问矩阵

7.5.2 访问矩阵

2. 具有域切换权的访问矩阵

为了实现在进程和域之间的动态联系，应能够将进程从一个保护域切换到另一个保护域。为了能对进程进行控制，同样应将切换作为一种权力，仅当进程有切换权时，才能进行这种切换。为此，在访问矩阵中又增加了几个对象，分别把它们作为访问矩阵中的几个域；当且仅当switch∈access(i, j)时，才允许进程从域i切换到域j。

7.5.2 访问矩阵

2. 具有域切换权的访问矩阵

对象 / 域	F_1	F_2	F_3	F_4	F_5	F_6	Printer 1	Plotter 2	域 D_1	域 D_2	域 D_3
域D_1	R	R, W								S	
域D_2			R	R, W, E	R, W		W				S
域D_3						R, W, E	W	W			

图7-19 具有切换权的访问控制矩阵

7.5.3 访问矩阵的修改

1. 拷贝权(Copy Right)

我们可利用拷贝权将在某个域中所拥有的访问权(access(i, j))扩展到同一列的其它域中，亦即，为进程在其它的域中也赋予对同一对象的访问权(access(k, j))，如图7-20所示。

域 \ 对象	F_1	F_2	F_3
D_1	E		W^*
D_2	E	R^*	E
D_3	E		

(a)

域 \ 对象	F_1	F_2	F_3
D_1	E		W^*
D_2	E	R^*	E
D_3	E	R	W

(b)

图7-20 具有拷贝权的访问控制矩阵

7.5.3 访问矩阵的修改

2. 所有权(Owner Right)

人们不仅要求能将已有的访问权进行有控制的扩散，而且同样需要能增加某种访问权，或者能删除某种访问权。此时，可利用所有权(O)来实现这些操作。

域 \ 对象	F_1	F_2	F_3
D_1	O, E		W
D_2		R^*, O	R^*, O, W
D_3	E		

(a)

域 \ 对象	F_1	F_2	F_3
D_1	O, E		
D_2		O, R^*, W^*	R^*, O, W
D_3		W	W

(b)

图7-21 带所有权的访问矩阵

7.5.3 访问矩阵的修改

3. 控制权(Control Right)

拷贝权和所有权都是用于改变矩阵内同一列的各项访问权的，或者说，是用于改变在不同域中运行的进程对同一对象的访问权的。控制权则可用于改变矩阵内同一行中(域中)的各项访问权，亦即，用于改变在某个域中运行的进程对不同对象的访问权的。如果在access(i，j)中包含了控制权，则在域Di中运行的进程可以删除在域Dj中运行的进程对各对象的任何访问权。

7.5.3 访问矩阵的修改

3. 控制权(Control Right)

域 \ 对象	F_1	F_2	F_3	F_4	F_5	F_6	Printer 1	Plotter 2	域 D_1	域 D_2	域 D_3
域D_1	R	R, W									
域D_2			R	R, W, E	R, W		W				Control
域D_3						R, E	W	W			

图7-22 具有控制权的访问矩阵

7.5.3 访问矩阵的实现

1. 访问控制表(Access Control List)

这是指对访问矩阵按列(对象)划分，为每一列建立一张访问控制表ACL。在该表中，已把矩阵中属于该列的所有空项删除，此时的访问控制表是由一有序对(域，权集)所组成的。由于在大多数情况下，矩阵中的空项远多于非空项，因而使用访问控制表可以显著地减少所占用的存储空间，并能提高查找速度。

7.5.3 访问矩阵的实现

2. 访问权限(Capabilities)表

如果把访问矩阵按行(即域)划分，便可由每一行构成一张访问权限表。换言之，这是由一个域对每一个对象可以执行的一组操作所构成的表。表中的每一项即为该域对某对象的访问权限。当域为用户(进程)、对象为文件时，访问权限表便可用来描述一个用户(进程)对每一个文件所能执行的一组操作。

7.5.3 访问矩阵的实现

2. 访问权限(Capabilities)表

	类　　型	权力	对　　象
0	文件	R−−	指向文件 3 的指针
1	文件	RWE	指向文件 4 的指针
2	文件	RW−	指向文件 5 的指针
3	打印机	−W−	指向打印机 1 的指针

图7-23 访问权限表

谢谢大家！ Q&A