

当多道程序的概念还未产生时,文件系统是计算机管理的主要部分。计算机的重要作用之一是能快速处理大量信息。由于计算机的内存容量有限,且不能长期保存信息,需要文件作为数据的载体输入到应用程序,或者作为输出数据的载体长期保存。不管是程序、图像、电子邮件等,平时都只能以文件的形式存放在外存中,需要时再将它们调入内存。但用户并不想关心文件是怎么存放在外存上的问题,而是希望直接通过文件名就能使用它。因此,对文件的各种具体管理工作就交给了文件系统。

本章将介绍文件及文件系统的基本概念,并要求学生掌握文件系统如何对文件进行组织、存取和保护。文件系统包括文件的逻辑结构和物理结构、文件的目录管理、文件存储空间管理,以及对文件的各种操作、文件的共享、安全与控制等问题。

### 5.1 Windows 中的文件

Windows 支持长文件名,文件名最多可使用 256 个字符。通过扩展名可识别该文件的类型。例如,文件 Arj. exe 代表一个可执行文件,Leaves. bmp 代表一个图像文件,xz. dbf 代表一个数据库文件。除“? \ \* " < > |”外,文件名可以包含空格和其他所有字符。

文件的存放路径由文件所在驱动器和文件夹来确定。文件夹代表对文件及目录形式存放的信息的分组,并且还可以在一个文件夹里包含其他文件夹。这样,多层文件夹就构成了一个“文件夹树”,最底层的树叶才是一个文件。在 DOS 和 Linux 中,把文件夹称为“目录”,从而形成一棵“目录树”。

Windows 10 提供了两个十分有效的文件管理工具:“我的电脑”和“资源管理器”。下面来看看资源管理器是如何对文件进行管理的。

#### 5.1.1 资源管理器

Windows 资源管理器显示计算机上的文件、文件夹和驱动器的分层结构。使用 Windows 资源管理器,可以复制、移动、重新命名以及搜索文件和文件夹。通过“Windows 徽标”→“Windows 附件”→“文件资源管理器”命令,即可启动资源管理器,启动后显示如图 5.1 所示的窗口。

资源管理器的左边窗格称为树格,以层次结构显示所有的文件夹,“>”表示其内还包含有其他子文件夹,“v”表示可将文件夹收缩。单击“>”,将显示其下的所有文件夹;单击



图 5.1 Windows 10 的资源管理器

“∨”，将重新折叠文件夹。

单击左边窗格中代表某文件夹的相应图标，可在右边窗格看到相关内容，这时图标将显示为打开状态。如果用户要进行一些如复制、更名、移动等操作，则只需在选中对应图标后单击相应的命令按钮即可。资源管理器在幕后执行相应的文件系统调用命令来完成用户的要求。有了文件系统提供的专门管理文件的工具，用户的操作就大大简化了。用户还可以使用资源管理器来创建文件夹，只需选择“文件”→“新建”→“文件夹”命令就能完成。

要创建文件，可通过使用 Windows 文件系统提供的记事本、写字板等。创建文件后，可对它进行各种编辑。下面看看 Windows 的记事本。

### 5.1.2 记事本

选择“Windows 徽标”→“Windows 附件”→“记事本”命令，即可启动记事本，如图 5.2 所示。

记事本是一个文本文件编辑器，文本文件不需任何特殊格式代码和控制字符。通过“文件”→“新建”命令来建立一个新文件，或通过“文件”→“打开”命令来修改一个已有的文件。在打开的窗口工作区内可以直接输入文本，输入时，可使用编辑菜单提供的剪切、复制、粘贴等命令，还可通过“编辑”→“选择字体”命令来设置字体，在文档中插入当前日期和时间等。当文件编辑好后，选择“文件”→“保存”命令，可将文件信息长期保存。

Linux 同样提供了几个编辑器，如 vi 的标准文本编辑器和 emacs 的全能编辑器等。其中，emacs 远远超出普通编辑器的功能，可以保存日历，作为计算器使用，建立大纲，甚至浏览万维网。



图 5.2 Windows 的记事本

当用户打开文件进行编辑时,看到的文件是由一个个字符组成的有序的集合。而文件系统为了完成编辑任务,需要直接对辅存上的文件进行操作,因此,计算机处理的文件和用户眼中的文件是不一样的,就是说文件有不同的表现形式。

### 5.1.3 文件的不同形态

用户看到的文件是逻辑文件。有些文件由有序的字符流组成,如一封信、一个程序,称为流式文件。有些文件则由若干记录组成,如数据库文件,称为记录式文件。

计算机处理的文件称为物理文件。针对不同的存储介质,文件的存放形式是不一样的。因此,物理文件也有不同的形态,如连续存放、串联存放等。Windows 10 支持 3 个不同的文件系统(FAT、FAT32 和 NTFS),对物理文件的存储空间进行有效管理。具体内容以后再详细说明。

## 5.2 文件和文件系统的基本概念

文件的含义很广,一篇文章,一张照片,一首歌曲,一个程序,甚至是黑客编写的病毒等都可以构成文件。文件到底是什么呢?

### 5.2.1 文件

#### 1. 文件的定义

操作系统把文件视为字符流,可以简化管理。此时文件的基本单位是单个字符,字符之间只有顺序关系而没有结构上的联系。但在数据库管理方面,遇到的一些基本处理单位不能只用单个字符的。例如,对学校进行管理时,学生是基本单位,描述学生时应该包括学号、

姓名、年龄、所在系别班级等数据项。通常把一组相关数据项的集合称为记录。一个学生就是一个记录。

正是由于构成文件的基本单位可以是字符,也可以是记录,因此,文件有了下面的两个定义。

① 文件是一组具有符号名的相关联字符的集合。

② 文件是一组具有符号名的相关联记录的集合。

和文件相关的术语如下。

① 域(字段): 数据的基本单位,由字符、数字构成。

② 记录: 相关域的集合。

③ 数据库: 关联记录的集合,数据元素之间存在关系。

## 2. 文件名

文件名是一个用来标识文件的有限长度的字符串。

有了文件名就能区分不同的文件,还可以通过文件名来对文件进行管理。应用中的操作系统对文件的命名是有规定的。

DOS 和 Windows 10 中的文件名都采用“文件名. 扩展名”的形式,但 DOS 的文件全名最多 11 个字符,即“8. 3”格式,Windows 10 支持长文件名,即文件名最多可使用 256 个字符。

通过扩展名可用来识别该文件的类型。例如,. bat 表示批处理文件,. obj 表示目标文件,. zip 表示压缩文件,. bmp 表示图像文件等。

Linux 系统中规定文件名是一个以字母或下画线开头的不大于 255 个字符的字符串,且区分英文字母的大小写,Linux 系统中没有文件名和文件扩展名之分,文件的类型由用户命名时确定。例如,文件 test. c, Linux 认为这个文件名的长度是 6 个字符。

## 3. 文件的分类

从不同的角度,文件的分类是不一样的。

(1) 按性质和用途分类。

① 系统文件。系统文件主要由操作系统及其他系统程序和数据所组成。对这些文件,用户只能通过操作系统才能调用,不允许用户读写和修改它们。这些是最重要的文件,主要用来管理和维护计算机。

② 库文件。库文件由标准子程序和实用程序包组成。用户在进行软件开发过程中经常要使用到库文件,但只能调用,不允许修改。

③ 用户文件。用户文件由用户委托给系统保存的文件,如用户的源程序、一封信件或一张图片等。只要在允许的权限内,用户可对它们进行各种操作。

(2) 按文件的保护级别分类。

① 执行文件。执行文件只允许有权限的用户调用执行,是一种程序式文件,不允许用户查看和修改。

② 只读文件。只读文件允许有权限的用户查看,但不能修改和执行。

③ 读写文件。读写文件允许有权限的用户查看和修改。

④ 不保护文件。不保护文件没有任何保护级别,所有用户都可以查看和修改。

(3) 按文件的保存期限分类。

① 临时文件。文件存放的信息是临时性的。例如,有些程序在运行时,会产生一些临时文件,当程序正常结束时,临时文件会被清除。非正常关机时,会残留下一些临时文件,一般应定期删除以释放存储空间。

② 永久文件。永久文件是指要长期保存的文件,一般可存放在辅存上。

③ 档案文件。档案文件是用于备查或恢复时使用的存档文件。

(4) 按文件的逻辑结构分类。

① 流式无结构文件。流式无结构文件即由字符流构成的文件。例如,UNIX、Linux 系统在逻辑上都把文件视为该类型。

② 记录式结构文件。记录式结构文件是由若干记录构成的文件,在数据库管理方面,这类文件用得较多。流式文件可视为记录式文件的特例,即每个文件只由一个记录构成。

按文件的物理结构,还可把文件分为顺序文件、链接文件和索引文件。按文件的存取方法则可把文件分为顺序存取文件和随机存取文件。这些内容以后会进行详细介绍。

#### 4. 实用系统中文件的分类

(1) Linux 的文件类型。

Linux 系统中有 3 种基本的文件类型:普通文件、目录文件和设备文件。

① 普通文件是用户最经常面对的文件。它又分为文本文件和二进制文件。

文本文件:这类文件以文本的 ASCII 码形式存储在计算机中。它是以“行”为基本结构的一种信息组织和存储方式。

二进制文件:这类文件以文本的二进制形式存储在计算机中,用户一般不能直接读懂它们,只有通过相应的软件才能将其显示出来。二进制文件一般是可执行程序、图形、图像、声音等。

② 设计目录文件的主要目的是用于管理和组织系统中的大量文件。它存储一组相关文件的位置、大小等与文件有关的信息。目录文件往往简称为目录。

③ 设备文件是 Linux 系统很重要的一个特色。Linux 系统把每一个 I/O 设备都看成一个文件,与普通文件一样处理,这样可以使文件与设备的操作尽可能统一。从用户的角度来看,对 I/O 设备的使用和一般文件的使用一样,不必了解 I/O 设备的细节。设备文件可以细分为块设备文件和字符设备文件。前者的存取是以一个个字符块为单位,后者则是以单个字符为单位。

另外,还有连接输入输出的管道文件等。

Linux 用“-”代表普通文件,“d”代表目录文件,“c”代表块设备特殊文件,“t”代表字符设备特殊文件,“p”代表管道文件。

(2) Windows 的文件类型。

Windows 文件系统支持任意扩展名所指定的类型,只要求进行文件类型注册,同时还注册用什么程序打开这类文件之类的信息。下面介绍几种 Windows 中的常见文件类型。

① 程序文件。程序文件是计算机可以识别的二进制编码,其文件扩展名常为 .com 或 .exe,Windows 10 还有扩展名为 .pif 和 .lnk 的文件,代表了程序的快捷方式。

② 文本文件。文本文件是由 ASCII 码字符组成的文件,.txt 表示纯文本文件,.doc 表示由 Word 及写字板建立的文档文件。

③ 图像文件。图像文件中以不同的格式存储着图片的信息,常见的图像文件扩展名有 .bmp、.gif、.jpg 等。

④ 声音文件。由各种声音采集及处理软件产生的文件,如 .wav、.mp3 等。

⑤ 其他文件类型。例如,.ttf 是字体文件,.reg 是注册信息文件。

选择“控制面板”→“默认程序”命令,可看到在 Windows 中注册的所有文件类型,如图 5.3 所示。

图 5.3 中显示了不同类型的文件与特定应用之间的关联。



图 5.3 文件类型与特定应用

如果用户需要更改文件类型与特定应用的关联,可以右击目标文件的图标,在弹出的菜单中选择“打开方式”→“选择其他应用”命令,进行重新选择,如图 5.4 所示。此外,还可以选择“控制面板”→“默认程序”命令,来更改文件类型与特定应用的关联,如图 5.5 所示。

## 5.2.2 文件系统

### 1. 文件系统的定义和功能

文件系统是指文件命名、存储和组织的总体结构。文件系统是与管理文件有关的软件和数据集合。一个好的文件系统应该具有以下功能。

(1) 对用户友好的接口让用户实现按名存取。用户要使用某个文件时,只要给出文件名即可,由文件系统根据文件名到文件存放的存储器中去存取,用户无须关心文件的物理存放位置,以及文件如何传输这些物理细节。这是文件系统最重要的任务。

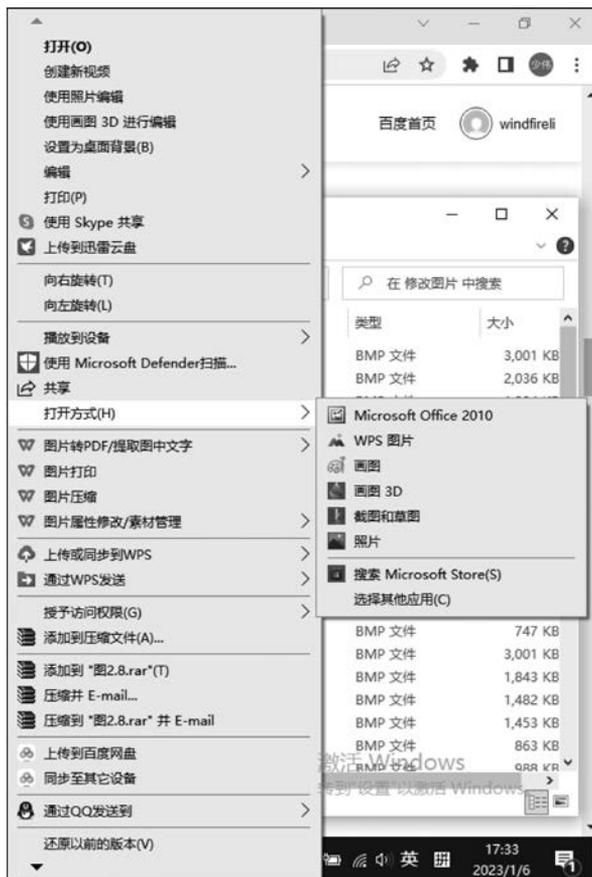


图 5.4 更改文件类型与特定应用



图 5.5 选择应用

(2) 能提供对文件的各种操作。例如,创建文件、读文件、写文件、删除文件、设置文件的访问权限等。

(3) 可以实现文件共享与保护。多个用户如果要使用同一个文件,没必要为每个用户都备份该文件,这就需要提供共享文件机制。为防止非授权的访问,还要为文件提供保护措施。

(4) 对外存存储空间的管理。由于内存容量有限,且只能临时存放数据与文件,一般都把文件存放在大容量的外存上长期保存,需要时再调入内存。那么多的文件挤在一个存储器中,文件系统必须能有效、合理地管理及分配辅助存储器空间。

(5) 文件系统应提供各种安全措施。多个文件放在一起,要防止它们互相破坏。对于非法用户,需要采取有效手段拒之门外。随着网络的发展,黑客无孔不入,安全成为越来越重要的问题。因此,文件系统必须提供层层安全措施,有效保障文件的安全。

当然,不同的文件系统具有的功能不一定完全一样,但以上 5 点是最基本的要求。

## 2. 几种常用文件系统

Linux 最初使用的文件系统是 Minux,但 Minux 有相当的局限性,性能比较差。它的文件名不能长于 14 个字符,最大的文集大小是 64MB。第一个专为 Linux 设计的文件系统——扩展文件系统(Extended File System,EXT),在 1992 年 4 月引入,解决了许多问题,但是仍然感到性能低。所以在 1993 年,增加了扩展文件系统第二版 EXT2。目前 Linux 支持多种不同的文件系统,这让它非常灵活,可以和许多其他操作系统共存。

当 EXT 文件系统增加到 Linux 的时候,进行了一个重要的开发。真实的文件系统通过一个接口层从操作系统和系统服务中分离出来,这个接口称为虚拟文件系统 VFS。VFS 允许 Linux 支持许多(通常是不同的)文件系统,每一个都向 VFS 表现一个通用的软件接口。Linux 文件系统的所有细节都通过软件进行转换,所以所有的文件系统在 Linux 核心以外和系统中运行的程序有同样的表现形式。Linux 的虚拟文件系统层允许同时透明地安装许多不同的文件系统。

EXT2 在 Linux 社区中是最成功的文件系统,它支持的磁盘分区容量可达到 4TB,且文件名可长达 255 个字符。EXT2 已成为所有 Linux 发行版本的基本文件系统。文件存储的最小单位是数据块,这些数据块长度相同。块设备可以看作是一系列能够读写的块。文件系统无须关心自身要放在物理介质的哪一个块上,这是设备驱动程序的工作。当一个文件系统需要从包括它的块设备上读取信息或数据的时候,它请求对它支撑的设备驱动程序读取整数数目的块。

DOS 的文件系统是 FAT。最典型的是 FAT 16,它规定的文件名是 8.3 格式,能支持最大磁盘分区为 256MB,采用 16 位实模式驱动程序,用户界面不够友好。

Windows 操作系统支持 16 位文件分配表(FAT 16)、32 位文件分配表(FAT 32)、光盘文件系统(CDFS)、通用磁盘格式(UDF)及 Windows NT 文件系统(NTFS)。FAT 32 支持 255 个字符的长文件名,能支持最大磁盘分区可达 2TB,采用 32 位保护模式驱动程序。NTFS 提供了 FAT 文件系统的所有功能,同时又提供了对高级文件系统特征(安全模式、压缩和加密)的支持。它是为在大磁盘上完成文件操作而设计的。其最小簇尺寸可为 4KB。NTFS 是一个用于网络的文件系统,支持包括卷装配点、远程存储、文件系统加密、稀疏文件及磁盘限额在内的众多存储增强功能。

表 5.1 显示了 Windows 支持的 3 个磁盘分区文件系统(NTFS、FAT 和 FAT 32)的特性。

表 5.1 Windows 文件系统的特性比较

性能	文件 系 统		
	NTFS	FAT	FAT32
兼容性	运行 Windows 10 或 Windows 2000 的计算机可以访问 NTFS 分区上的文件。其他操作系统则无法访问	可以通过 MS-DOS、所有版本的 Windows、Windows NT、Windows 2000、Windows 10 和 OS/2 进行访问	只能通过 Windows 95 以上的操作系统进行访问
文件大小	文件大小只受卷的容量限制	最大文件大小为 2GB	最大文件大小为 4GB
安全性	能使用诸如活动目录和基于域的安全性等功能	没有文件加密等安全机制	没有文件加密等安全机制

Windows 10 可以在文件不发生变化的情况下实现磁盘格式从 FAT 和 FAT32 到 NTFS 的转换。

### 5.2.3 文件的逻辑结构和存取方法

逻辑结构代表用户对文件的看法,文件系统通过一定的存取方法来实现对文件的操作。

#### 1. 文件的逻辑结构

用户可见的文件结构称为文件的逻辑结构。图 5.6 显示两种不同的文件逻辑结构。

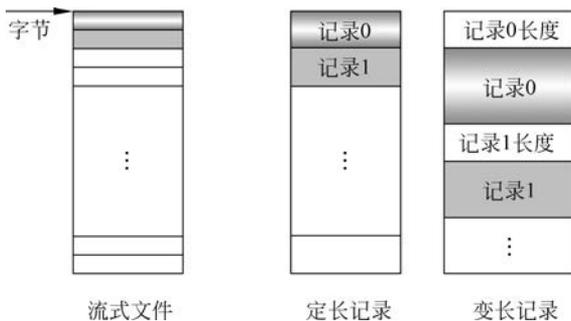


图 5.6 文件的逻辑结构

##### (1) 流式无结构文件。

流式无结构文件是由相关联的字符流组成的文件,文件的长度为所含字符数,字符为基本管理单位。由于不用对格式进行额外说明,空间利用上就较节省。大量的源程序、可执行程序、库函数等都采用流式无结构文件形式。DOS、UNIX、Windows、Linux 系统中的普通文件都是流式文件。

##### (2) 记录式结构文件。

记录式结构文件是有结构的文件,由相关联的若干记录构成的。这些记录分别以  $0 \sim n$  按顺序编号,称为记录 0、记录 1、……、记录  $n$ ,记录的编号就是记录在文件中的逻辑地址,这样的记录称为逻辑记录。

记录是一个具有特定意义的信息单位,由一组相关联的字段组成。例如,学生登记表文件 xsdjb.dbf 中的每一行就是一个记录(见表 5.2)。

表 5.2 学生登记表文件 xsdjb.dbf 中的记录

姓名	学号	性别	出生年月	家庭地址
张三	93101	男	1975.10	汉口江大路 132 号
李四	93102	女	1974.2	武昌街道口 65 号
...	...	...	...	...

根据文件中记录长度是否相等,记录式文件可分为如下两种。

① 定长记录文件。所有记录的长度都相等,文件的长度可以直接用记录数目表示。它处理方便,开销小,应用较为广泛。

② 变长记录文件。记录的长度可以不相等,在每个记录前面都要记载该记录长度,变长记录文件的长度为各记录长度之和。

记录式文件主要用于数据库管理系统中,可以把文件中的记录按各种不同的方式排列,如按学号进行排列,或按出生年月进行排列,这样就构成不同的逻辑结构,便于用户对文件中的记录进行修改、追加、查找等操作。

## 2. 文件的存取方法

用户通过对文件的存取,完成对文件的修改、搜索等操作。根据文件的性质和用户使用情况,决定不同的存取方法。

### (1) 顺序存取。

顺序存取是指按照记录的逻辑排列次序依次存取每个记录。若上次读取的是记录  $N$ ,则本次要读取的记录自动确定为  $N+1$ ,故每次存取不必给出具体的存取位置。

### (2) 随机存取。

随机存取又称为直接存取,即允许随意存取任一记录,而不管上次访问了哪个记录。每次存取操作都要指定存取操作的开始位置。

流式文件只适合顺序存取,记录式文件既可以顺序存取,也可以随机存取。

## 5.2.4 文件的物理结构和存储设备

文件在辅存上的存放形式称为文件的物理结构。如何组织文件的物理结构,才能既提高存储空间利用率,又减少存取文件信息的时间,这是文件系统要研究的一个重要问题。

要知道文件是如何存放在辅存上,首先应该了解辅存的特性。

### 1. 常见辅存设备介绍

#### (1) 磁带。

磁带机对磁带进行存取访问时,是将磁带转到所需位置,再由磁头读取信息。由于磁带的启动需要一段时间,存取单位之间要留出一定的间隙。磁带的存取单位为块,在块与块之间设置间隙,每次存取都以块作为单位,如图 5.7 所示。一个物理块中可能存放多个逻辑记录,也可以一个逻辑记录占用多个物理块。由于每次存取都以一个物理块为单位,将一个物理块中的信息称为物理记录。

磁带只能顺序存取,无须寻找磁道,但要考虑磁头寻找记录区的等待时间。磁头总是固定的而磁带是移动的。磁带的寻址时间是磁带转动到磁头将访问的记录区所在位置的时间。

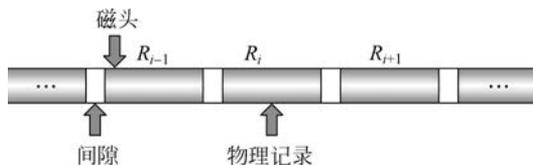


图 5.7 磁带结构示意图

## (2) 磁盘。

磁盘是现在用得较多的一种辅存设备,单位面积上它的容量比磁带大,价格便宜。磁盘的结构如图 5.8 所示。

软盘由一张双面各覆盖一层磁敏感材料的盘片构成,盘片上方通过一个读写磁头来从盘片上读出或写入数据。操作系统将每个盘面划分成若干个称为磁道(track)的同心圆,再将每个磁道划分成扇区(sector)。扇区是磁盘的最小存储单位,通常为 512B。磁道由外向内从 0 开始编号,扇区则从 1 开始编号,于是就可通过盘面号、磁道号和扇区号来决定一个扇区在磁盘上的绝对位置。如位于 0 面、第 1 磁道、第 2 扇区的扇区地址是“0,0,2”。要访问数据,只要将磁头对应到磁道和扇区即可。

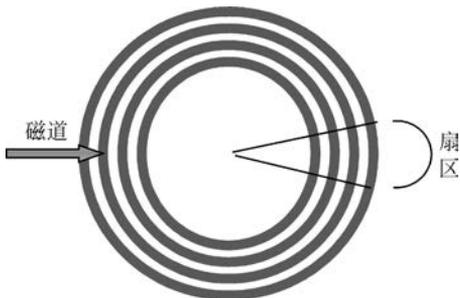


图 5.8 磁盘结构示意图

硬盘类似于多张软盘的层叠。每个盘片都有两个面,每面能存储数据,并对应一个磁头。由于多层磁道形成一个个圆柱形,硬盘的磁道称为柱面。由磁头号、柱面号、扇区来决定硬盘物理单位的绝对地址。

许多操作系统还将扇区进行组合来形成不同的磁盘分区,以方便用户对文件进行逻辑组织。

磁盘存储器采取直接存取方式,寻址时间包括两部分:一是磁头寻找目标磁道所需的寻道时间  $T_s$ ;二是找到磁道以后,磁头等待所需要读写的扇区旋转到它的下方所需要的等待时间  $T_w$ 。由于寻找相邻磁道和从最外面磁道找到最里面磁道所需的时间不同,磁头等待不同扇区所花的时间也不同,因此,取它们的平均值,称作平均寻址时间  $T_a$ ,它由平均寻道时间  $T_{sa}$  和平均等待时间  $T_{wa}$  组成:

$$T_a = T_{sa} + T_{wa} = (T_{smax} + T_{smin})/2 + (T_{wmax} + T_{wmin})/2$$

平均寻址时间是磁盘存储器的一个重要指标。硬磁盘存储器比软磁盘存储器的平均寻址时间短。

目前应用广泛的辅助存储设备还有光盘(如 CD-ROM、CD-RW、DVD-ROMH 和 DVD-RW)、闪存、移动硬盘等。

## 2. 文件物理结构

文件的物理结构代表了数据的存储方式,常见有以下 3 种。

### (1) 连续文件。

连续文件是指把逻辑上连续的文件信息依次存放到连续的物理块中,如图 5.9 所示。

磁带和磁盘都可以采用连续文件的存储方式,只要有大小合适的连续的存储空间,就能存放文件。对于磁带上的连续文件,只适用顺序存取的方法。而对于磁盘上的连续文件,既可用顺序存取,也可用随机存取的方法。假定采用随机存取,如文件逻辑记录和物理块大小相等,要访问文件 A 的第二个逻辑记录,只需将磁头定位到物理块 6 处即可。

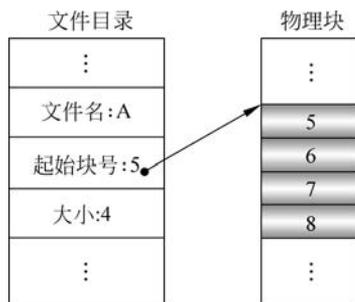


图 5.9 连续文件

连续文件结构简单,实现容易。若采用顺序访问方式,因文件是分配在连续的存储空间,只要找到文件第一块位置,就可很快访问完所有信息。但连续存储空间的要求导致大量较小的区域无法分配和利用。对于需要动态增长的文件,连续文件往往无能为力,因为该文件后面的连续块可能已分配给其他文件使用。Linux 系统中保留了连续文件结构。

### (2) 串联文件。

串联文件又称为链接文件,它把逻辑上连续的文件信息分散存放于不连续的块中,每个物理块最末一个字作为链接字指向与它链接的下一物理块,文件的结尾块则存放结束标记“^”。如图 5.10 所示,文件 A 存放在 4 个不连续的物理块 5、7、10、12 中。

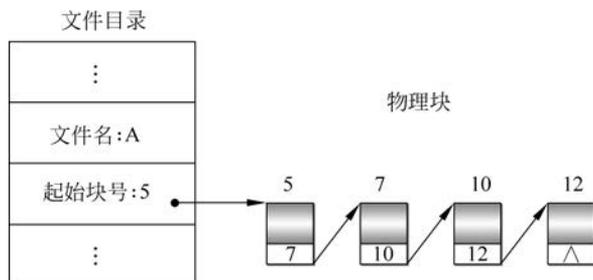


图 5.10 串联文件

串联文件只适用于磁盘,不适用于磁带,且对串联文件只能顺序存取。若采用随机存取,如要访问文件最后一块的内容,实际上要从文件头开始,通过指针依次向后访问,直到访问到文件的最后一块,这其实就变成了顺序存取。

串联文件实现了文件的非连续存储,提高存储空间利用率,消除了外部碎片。如果文件大小要变化,则只需再链接空闲块,或删除链中某块即可,这样便于动态修改和扩充。但串联文件搜索效率低,只适宜顺序存取,不适宜随机存取。

### (3) 文件映照。

在系统中建立一张文件映照表,把所有盘块的指针都存放在该表中,每个指针占一个表项。用户目录中存放文件的第一个块号,利用这一块号到文件映照表中找到下一块号,文件的结尾块则存放结束标记“^”,通过文件映照表可获得该文件占用的所有块号,如图 5.11 所示。文件 A 通过文件映照表对应于 5、7、10、12 物理块。

大容量磁盘的文件映照表很大,一般被作为文件保存在磁盘中,需要时,调入内存即可。文件映照方式只适用于磁盘,既可进行顺序存取,又能进行随机存取。例如,要读取 A

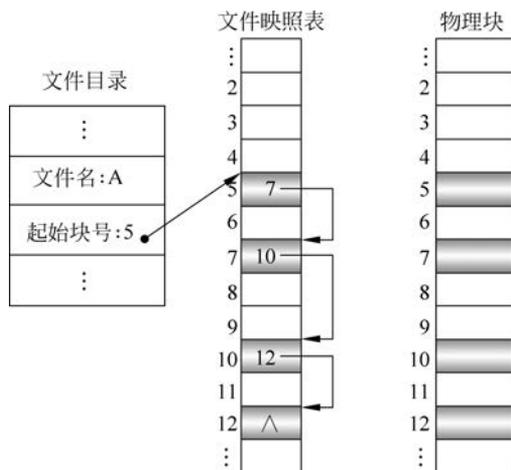


图 5.11 文件映照表

文件尾的信息,通过用户文件目录和文件映照表,可知道文件尾信息存放在物理块 12 中,就可直接读取磁盘中第 12 块的内容,没必要把文件从头读到尾。

文件映照表既保持了链接文件的优点,又克服了其缺点,但是增加了文件映照表的存储开销,访问速度的提高是用存储空间的增长来换取的。在 DOS 中,使用称为 FAT 的文件映照表来完成文件的映照,而在 Windows 中使用 FAT32 来完成文件的映照。

#### (4) 索引文件。

索引文件的思想类似于存储管理中的分页管理,把文件划分为大小相同的若干连续的逻辑块,每个逻辑块可存放到存储空间中的任一物理块中。系统为每个文件建立一张索引表,给出逻辑块号和分配给它的物理块号的对应信息。图 5.12 中可通过用户文件目录表中的文件 A 的索引表指针找到对应的索引表,得到 A 的存储情况。

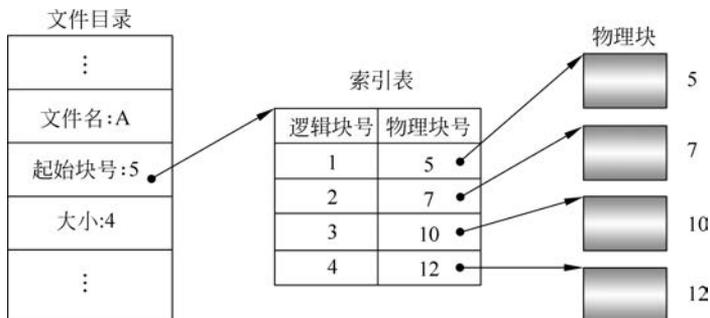


图 5.12 索引文件

索引文件只适用于磁盘,对索引文件除了能进行顺序存取外,也可较方便实现随机存取。若要对文件进行增加或删除,只需修改索引表。但因为每个文件都有一张索引表,如果把索引表全部放入内存,必然占据过多内存空间,一般把索引表以文件的形式存放到外存,需要时调入内存即可。

对于中、小型文件,存放索引表文件可能只需一个物理块,但对于大型文件,由于索引表比较大,需要用多个物理块来存放,物理块之间再通过链接指针相互链接,索引表的访问效

率必然降低。这时可采用两级索引的方法,即为存放索引表的物理块(简称为索引块)再建立索引,如图 5.13 所示。

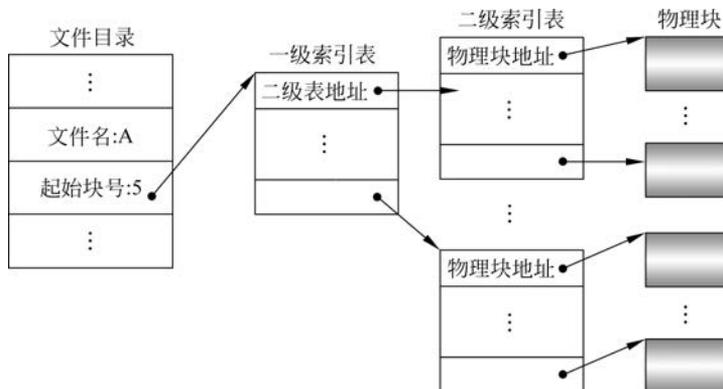


图 5.13 二级索引目录

索引结构是计算机操作系统中普遍采用的结构,如在 Linux 系统中,小型文件采用一级索引结构,大型文件采用二级索引结构,巨型文件则采用三级索引结构。

(5) 文件物理结构的比较。

从查找时间来看,连续文件最快,索引文件和文件映照次之,串联文件最慢。

从空间开销来看,连续文件不需要额外的空间开销;串联文件的每个物理块中需要存放链接字;文件映照需存放文件映照表;索引文件为每个文件建立一张索引表。

从适宜设备和存取方法来看,连续文件可用于磁带和磁盘;串联文件、索引文件和文件映照只适用于磁盘;串联文件只适合顺序存取;而文件映照、索引文件和磁盘上的连续文件,除了能进行顺序存取外,也能实现随机存取。

从文件增删来看,连续文件不能动态增长,其他 3 种都可较容易实现文件的动态改变。

(6) 存储设备、文件物理结构和存取方法的关系。

文件物理结构和存取方法与存储设备密切相关,图 5.14 列出了三者之间的关系。

存储设备	磁带	磁盘		
文件物理结构	连续	连续	串联	索引
存取方法	顺序	顺序、随机	顺序	顺序、随机

图 5.14 存储设备、文件物理结构、存取方法的关系

### 5.2.5 Linux 的文件物理结构

EXT2 是 Linux 最为成功的文件系统。每个文件对应一个索引节点( $i$  节点),每个  $i$  节点有一个唯一的整数标识符,所有文件的  $i$  节点都保存在  $i$  节点表中。 $i$  节点内的索引结构如图 5.15 所示。

**注意:** 图 5.15 只画出  $i$  节点内的文件地址索引表项, $i$  节点内的其他内容,如文件名、文件类型、文件长度等未列出。

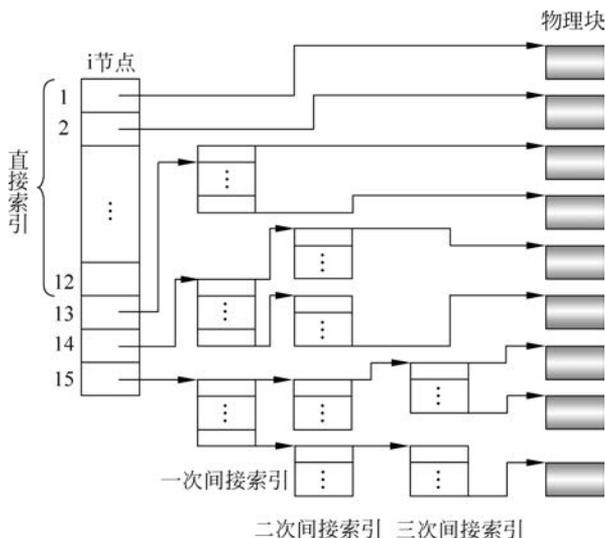


图 5.15 EXT2 的 i 节点内的索引结构

## 5.3 文件目录管理

文件系统是如何通过文件名知道文件实际存放位置的呢？这是文件目录管理的功劳，它的基本功能就是实现“按名存取”。文件目录还要能合理组织目录结构，使得各个文件的查找速度较快，还要能提供对文件的共享，即让多个用户共用一个文件。文件目录是一张记录所有文件的基本信息目录表，如文件名、文件存放的物理位置以及文件说明和控制方面的信息。

### 5.3.1 文件控制块

文件系统要实现对文件的按名存取，关键是要使文件与文件的物理地址建立联系。因此，文件系统为每个文件建立一个文件控制块 FCB，里面存放了有关文件名、文件地址等多方面的描述信息。文件系统借助文件控制块中的信息，实现对文件的管理。

文件控制块的有序集合就构成了文件目录，即文件目录的每个目录项就是一个文件控制块。文件控制块的基本内容如下。

(1) 文件名：用于标识一个文件的符号名。不同的操作系统，文件名命令规定是不一样的。

(2) 文件的物理位置：用于指明文件在外存的具体存储位置，通过该项内容，系统就能找到这个文件。

(3) 文件的逻辑结构：用于指明是流式文件还是记录式文件。

(4) 文件的物理结构：用于指明文件是连续文件、串联文件还是索引文件，这项内容确定了系统对文件可以采用的存取方式。

(5) 文件的存取控制权限：用于规定各类用户对文件的存取权限。

(6) 文件的使用信息：用于指明文件的使用信息，如文件建立日期和时间、文件上一次

修改的日期和时间、当前已打开该文件的进程数、文件是否被其他进程锁住等。

由于功能的不同,对于不同的操作系统,FCB的内容不会完全一样。

### 5.3.2 Linux 的索引节点

在Linux中,采用了把文件名和文件描述信息分开的方法,将文件目录项中除文件名之外的信息都放到一个数据结构中,该数据结构称为索引节点(index node),简称为i节点。这样,在文件目录项中,就只需存放文件名和该文件名对应的i节点号,大大减少了文件目录的规模,节省了系统开销。可以看出,在这里文件控制块已变成索引节点。

Linux索引节点的内容如下。

- (1) 设备号:指包含该文件的设备的标识符。
- (2) 索引节点号。
- (3) 文件的访问权限位:表示对该文件能进行何种操作。
- (4) 连接计数:表示连接到这个文件的目录项个数。当该数为零时,表示该节点可被丢弃或重新使用。
- (5) 文件的用户识别号(UID)和组织识别号(GID)。
- (6) 设备特殊文件的主设备号和辅设备号。
- (7) 其他:如文件大小、文件最后一次访问时间、文件最后一次修改时间以及文件最后一次状态改变时间等。

### 5.3.3 一级目录结构

当系统确定了文件控制块的内容后,就可以建立一张目录表来存放所有文件的文件控制块,即每个目录项存放一个文件控制块,这样就构成了最简单的目录结构:一级目录结构。图5.16简化了一级目录结构目录项的内容,目录项中的状态位表明该目录项是否空闲。



图 5.16 一级目录结构

每个目录项都指向一个普通文件的物理地址,文件名和文件是一一对应的,有了这张表,就可以通过文件名来对文件进行各种操作,即实现“按名存取”。

如要读某个文件,系统根据文件名去查找目录表,获得该文件的物理地址,就可对文件进行读操作。如要创建一个新文件,系统首先到目录表中查看是否有和新文件同名的文件。若无同名文件,则通过查看状态位找到一个空闲目录项,就可将新文件名及其有关信息填入其中。如要删除文件,则只需要将文件在目录表中的目录项清除掉。

可看出一级文件目录有如下特点。

- (1) 结构简单、清晰,便于维护和查找。
- (2) 可实现按名存取。
- (3) 搜索速度慢。为查找一个文件的目录,平均需查找目录表的一半,若是大型目录表,则搜索效率非常低。

(4) 不允许文件重名。由于所有文件目录都存放在一个目录表中,故不能有重名的文件,以免造成混乱,用户取文件名时必须要知道所有文件的名字,给用户造成极大的不便。

(5) 不允许文件别名。由于文件名和文件是一一对应的,一个文件不能取不同的名字,即不允许用户以不同的名字来访问同一个文件,造成文件共享的不便。

为了解决一级文件目录的缺点,引入了二级文件目录。

### 5.3.4 二级文件目录

二级文件目录把目录表分成一个主目录 MFD(Master File Directory)和下一级的用户文件目录 UFD(User File Directory)。

系统为每个用户建立一个用户文件目录 UFD。该目录由该用户的所有文件的文件控制块组成,整个系统再建立一个主目录 MFD,每个用户占其中一个目录项,存放用户名及指向用户文件目录 UFD 的指针,如图 5.17 所示。

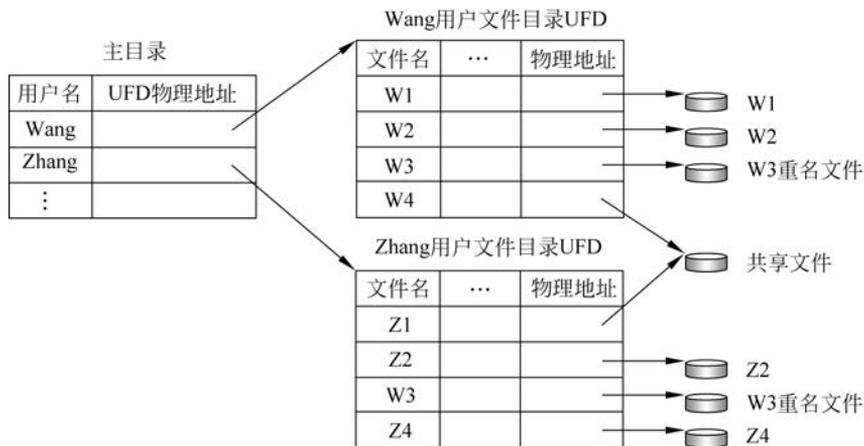


图 5.17 二级目录结构

要创建一个新文件,系统根据用户名查找 MFD 中对应的目录项,得到该用户的 UFD 地址,然后在 UFD 中取一个空闲目录项,填入新文件的 FCB 信息。要访问一个文件,需通过查找 MFD,得到对应用户的 UFD 地址,然后在 UFD 中找到对应的文件名,获得文件物理地址后才能对文件进行访问。要删除一个文件,只需回收其存储空间,然后将文件在 UFD 中的表目清空。

二级文件目录具有以下优点。

(1) 搜索速度得到提高。根据用户名先搜索 MFD,然后才根据文件名到 UFD 中去搜索该文件,不必将所有的文件目录都搜索一遍,显然大大提高了搜索速度。

(2) 允许文件重名。例如,用户 Wang 和用户 Zhang 都有名为 W3 的文件,由于系统存取文件时是先找用户名再找文件名,因此,完整的文件名是由用户名和文件名组成,即分别为 Wang/W3 和 Zhang/W3,它们被视为两个不同的文件。当然,同一用户的 UFD 中不允许有同名文件。

(3) 允许文件别名,即不同用户对相同文件可取不同名字。例如,用户 Wang 的 W4 文件和用户 Zhang 的 Z1 文件,虽然文件名不同,但它们在 UFD 中指向同一个文件。这样就

可以让多个用户以不同的文件名共享一个文件。

虽然二级文件目录有了很大改进,但随着外存容量的增大,可容纳的文件数越来越多,单纯分为二级结构已不能很方便地对种类繁多的大量文件进行管理。于是把二级文件目录的层次关系加以推广,在 UFD 下再创建一级子目录,将二级文件目录变为三级文件目录,以此类推,进一步形成四级、五级等多级目录。

### 5.3.5 树形目录结构

所有目录和文件组合在一起,构成了一个树状层次结构,称为目录树,如图 5.18 所示。树的顶部是一个单独的目录,称为根目录,用“/”表示。所有的子目录和文件都存放在根目录下,其中子目录中又可包含其他子目录,这样层层嵌套,将最底层的文件可看作树叶,而子目录则是树枝节点。

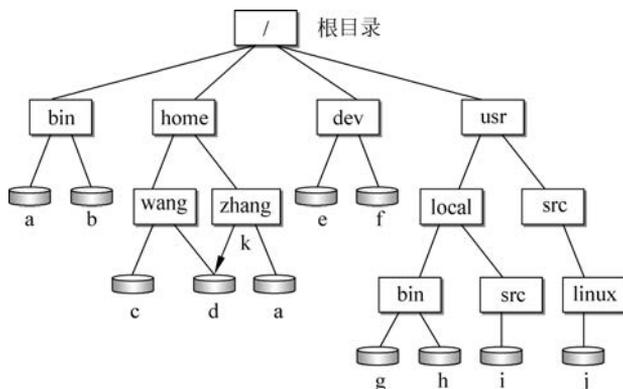


图 5.18 树状目录结构

有了目录树后,可以通过路径名来引用一个文件,路径名由文件名和包含该文件的目录名组成,路径名的类型有以下两种。

(1) 绝对路径名:指由根目录开始的路径名,如/home/wang 和/usr/local/bin/g。通过绝对路径名,可以区分重名文件。使用绝对路径名,一个文件必须从根目录处开始往下检索,因此查找路径较长。

(2) 相对路径名:指从当前工作目录开始的路径名,如用户正在/usr/local 下工作,则称 local 为当前目录。若要访问文件 g,则只需使用/bin/g,即从当前工作目录向下查找,提高了文件的搜索效率。在 Linux 系统中,对于一个刚登录的用户,他的当前工作目录是系统管理员为之建立账户时确立的,称为用户主目录。

采用树状目录结构,可以用文件别名来实现文件的共享。如/home/zhang 想以文件名 k 来访问/home/wang 目录下的 d 文件,只需在/home/zhang/k 和/home/wang/d 之间建立一个链接,让/home/zhang/k 直接指向/home/wang/d。

树状目录结构具有很多优点,如层次清楚,便于组织和管理;搜索速度比单级、二级目录快很多;解决了文件重名问题,每个文件在文件系统中由其绝对路径名唯一确定;解决了文件别名问题。Windows 中的多级目录称为文件夹树,图 5.19 是一个文件夹树的片段。

Linux 采用了树状目录结构(见图 5.20)。使用命令 tree 可以查看。

在树状目录结构目录项中存放了文件 FCB 的所有信息,这将造成目录内容太多,使文

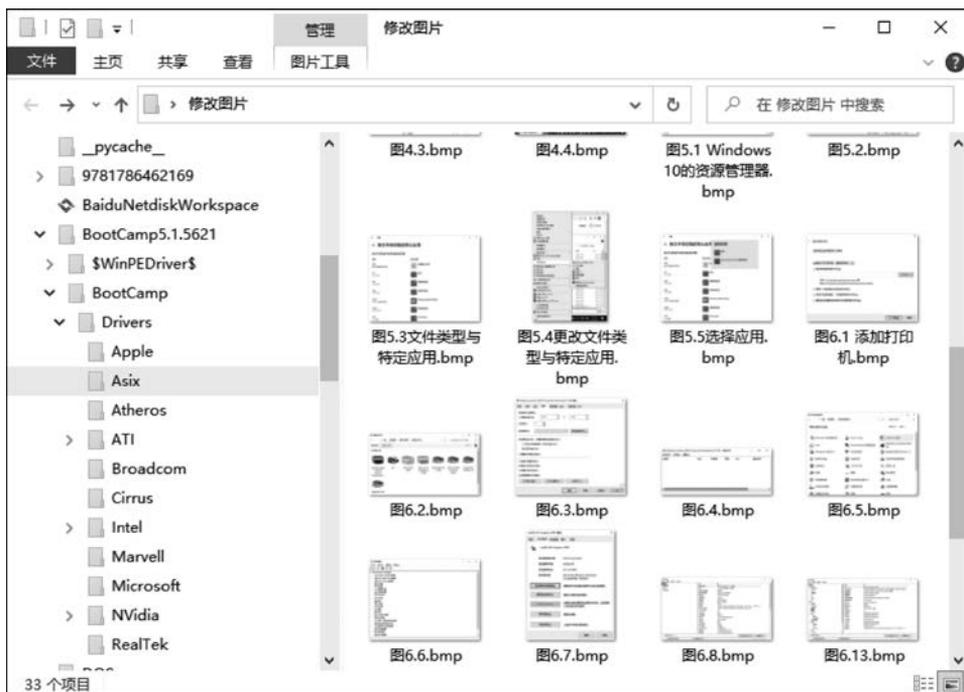


图 5.19 Windows 文件夹树的片段

```
[stu45@lxsvr home]$ tree -a
├── administrator
│   ├── state
│   └── ups [error opening dir]
├── tmp
├── www
│   ├── cgi-bin
│   ├── html
│   ├── addon-modules
│   ├── cgi-bin -> /var/www/cgi-bin/
│   ├── manual -> ../../usr/share/doc/apache-manual-1.3.26
│   ├── icons
│   └── small
└── yp
```

图 5.20 Linux 树状目录结构片段

件的操作变得复杂。为此可引入基本文件目录和符号文件目录来加以改进。

### 5.3.6 基本文件目录和符号文件目录

系统给所有文件赋予唯一的标识符,将文件目录的内容分为两部分:用符号文件目录来记录文件的相互关系,用基本文件目录来记录文件的说明信息。整个系统设置一个基本文件目录,每个用户对应一个符号文件目录。目录结构如图 5.21 所示。

基本文件目录中的 0、1、2 项固定赋予基本文件目录、空白文件目录和主目录。如要查找文件“/Wang/W1”,通过基本文件目录找到主目录,在主目录中查找到 Wang 的符号文件目录的标识符为 3,于是到基本文件目录中读出标识符为 3 的 Wang 的符号文件目录,得到文件 W1 的标识符为 10,基本文件目录为 10 的表项中记录了文件 W1 的所有信息。

利用基本文件目录和符号文件目录,既减少了目录内容,也解决了文件重名和别名问题,且便于文件共享。Linux 文件系统中的 i 节点就是一种基本文件目录。

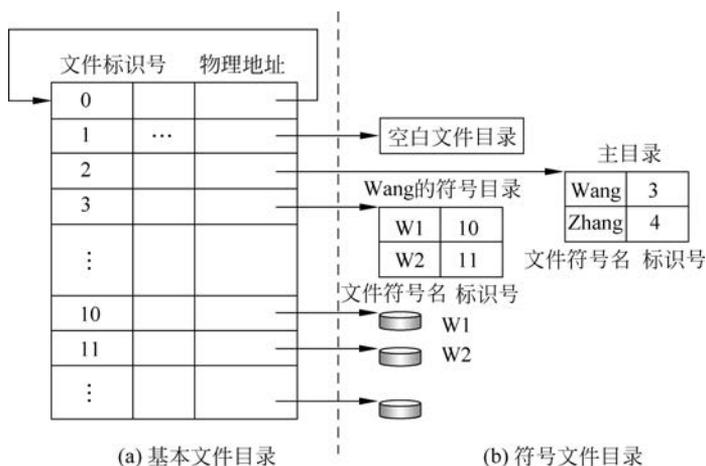


图 5.21 采用基本文件目录和符号文件目录的多级目录结构

### 5.3.7 Linux 目录结构的特点

(1) Linux 采用树状目录结构,目录树分支可以是一个磁盘、硬盘上的不同分区、光盘或者不同的文件系统。若目录树分支是 Linux 以外的文件系统,则需通过装载和拆卸来实现对目录树分支的挂接与撤销。mount 命令用来挂装各种文件系统,unmount 命令则完成对文件系统的拆卸。

(2) Linux 通过 i 节点的配合,采用硬链接解决目录树中同一文件系统的文件别名问题。每当建立一个硬链接,索引节点的引用计数值加 1,这样只要使用的计数位不为 0,就不能将这个文件删除,也可以防止用户删除其他用户正在使用的共享文件。图 5.22 为一个 i 节点中的信息及其分布。

① Mode: 包括两组信息,即这个 i 节点描述了什么,以及用户对于它的权限。

② Owner Information: 这个文件或目录的用户和组标识符。使文件系统能正确地进行文件访问权限控制。

③ Size: 文件的大小(字节)。

④ Timestamps: 这个 i 节点创建的时间和它上次被修改的时间。

⑤ Direct Blocks: 直接索引区,指向这个 i 节点描述的物理块的指针。直接索引区有 12 个索引项,可以索引 12 个物理块。

⑥ Indirect Blocks、Double Indirect 和 Triple Indirect 是 3 个间接索引区,分别指向各级间接索引表。这意味着文件越大,访问距离越长,小于或等于 12 个数据块大小的文件比更大的文件的访问更快。

Mode
Owner Information
Size
Timestamps
Direct Blocks
Indirect Blocks
Double Indirect
Triple Indirect

图 5.22 一个 i 节点中的信息及其分布

### 5.3.8 Windows 10 文件系统的结构

Windows 10 的 NTFS 文件系统支持卷装配点。卷装配点允许将一个卷装配到一个已

经存在的文件夹上,而不仅限于一个盘符的根部。通过为空的 NTFS 目录创建一个新的卷装配点,管理员就可以在无须额外盘符的情况下将新的卷嫁接到名称空间上。在系统发生改变的情况下,如从一台机器添加或删除设备等,卷装配点功能就显得非常强大和灵活。存储应用程序必须为处理由卷装配点功能所引起的名称空间的动态变化做好准备。

## 5.4 文件存储空间管理

为便于长期保存,文件通常都被存储在大容量的辅存上。因此,文件系统的重要任务之一就是要随时掌握存储空间的使用情况,以便有效而合理地分配空闲存储空间,并及时回收不用的存储空间。

### 5.4.1 文件系统常用的存储空间管理方法

#### 1. 位示图

文件系统在内存中为每一个辅存设备建立一张称为位示图的表,通过该表来反映辅存设备中所有物理块的使用情况。这张表由若干字节组成,每个二进制位对应辅存设备中的一个物理块。若该二进制位的值为 0,则表示它对应的物理块为空闲;若为 1,则表示已分配使用。例如,现有一个文件 file1 需要分配不连续的 3 个物理块,于是系统到位示图中依次查找空闲块,把 2、3、5 块分配给了 file1,且把这些位的值设为 1,如图 5.23 所示。

位示图初始状态	7	6	5	4	3	2	1	0
	0	0	0	1	0	0	1	1

分配给文件file1后位示图的状态	7	6	5	4	3	2	1	0
	0	0	1	1	1	1	1	1

图 5.23 位示图的变化

采用位示图的方法管理辅存空间较为简单,并且由于位示图占有空间很小,可放在内存中,访问速度较快,因此,如 CP/M、PDP-11 操作系统都采用了该方法。

#### 2. 空白文件目录

辅存上的一片连续的空闲区,可视为一个空白文件,系统设置一张空白文件目录来记录辅存上所有连续空闲块的信息。每个目录项存放一个空白文件的信息,包括该空白文件第一个空闲块号、空闲块个数、该文件所有空闲块号等信息,如图 5.24 所示。

序号	第一个空闲块号	连续空闲块个数	空闲块号
1	2	2	2,3
2	5	3	5,6,7
3	16	5	16,17,18,19,20
⋮			

图 5.24 空白文件目录

为一个新文件分配辅存空间时,与内存的动态分区类似,根据系统的要求采用最先适应算法、最佳适应算法或最坏适应算法,在空白文件目录中找到一个最合适的空白文件,把它

分配出去,然后在空白文件目录中调整该表项。

当要撤销一个文件时,就将文件占用的连续空间释放掉,然后将被释放空间的信息登记到空白文件目录中。这时涉及空闲块的合并问题,即如果释放的块和某个空白文件是相连的,就把它们合并为一个大的空白文件。显然这种方法适合于连续文件结构,但此方法有两个明显的缺点。

(1) 如果文件太大,在空白文件目录中将没有合适的空白文件能分配给它,尽管这些空白文件的总和能满足需求。

(2) 经过多次分配和回收,空白文件目录中的小空白文件越来越多,很难分配出去,形成了碎片。

### 3. 空闲链表法

该方法把所有的空闲块链接在一起,形成一个空闲块链表。空闲块链表法如图 5.25 所示。

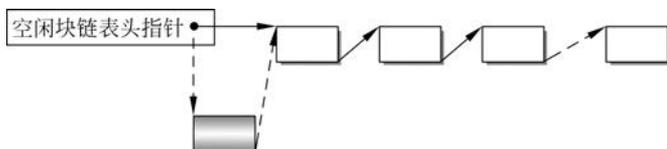


图 5.25 空闲链表法

当需要为一个文件分配存储空间时,系统从链头开始摘取所需要的空闲块,然后调整空闲块链表头指针。这样,只要文件大小不超过空闲块链总长度,系统总能为之分配足够的存储空间,且可让文件存放在不连续的存储空间,提高存储利用率。当要撤销文件时,只需把归还的块依次从链表头部链入即可。

空闲链表法的优缺点如下。

(1) 可实现不连续分配。

(2) 由于每个空闲块的指针信息都是存放在上一空闲块中的,这样就不用占用额外的存储空间,与空白文件目录管理方法相比节省了存储开销。

(3) 因为链接信息是存放在每个空闲块中的,每当在链上增加或删除空白块时需要很多输入输出操作,系统开销大。

(4) 对于大型文件系统,空闲链将会太长。

针对空闲链表法的缺点,还可以采用其他方法,如成组链接等,在此不再赘述。下面谈谈具体应用中系统对磁盘存储空间的管理。

## 5.4.2 FAT 磁盘格式

### 1. 基本概念

(1) 簇。

DOS 将若干连续扇区作为存储分配的单位,称为簇。不同的磁盘,簇的大小不一样,它随磁盘容量的增大而增大,如 512MB~1GB 的硬盘分区,每簇含 32 个扇区,1GB 以上分区每簇含 64 个扇区。



假设要访问根目录下的文件 FILE1。DOS 首先在根目录区找到 FILE1 文件的 FDT, 得到 FILE1 的起始簇号为 3, 然后到 FAT 表中找到 3 号表项, 其中的值说明下一个簇是 5, 5 号表项的数据说明下一个簇为 8, 8 号表项的值为 FFFF, 表示已到文件的结尾。于是可以知道 FILE1 在磁盘上的物理块为 3、5、7、8。这样很容易实现对整个文件的寻址。

如果用 Delete 命令删除文件 FILE1, DOS 只将 FILE1 的 FDT 的第一字节改为“E5”, 并不修改 FAT, 因此, 在对应 FAT 未发生变化前, 可以用 Undelete 命令恢复文件。

DOS 的 FAT 表中每个表项大小最初为 12 位, 称为 FAT12, 后来扩充为 16 位, 称为 FAT16。FAT16 对磁盘空间的管理很有效, 但它最多只支持 2GB 的硬盘分区, 这显然不适合目前的硬盘容量。又由于 DOS 分配是以簇为单位, 如果有一个只包含 1 字节的文件, 也需分配一个簇给它, 极易造成磁盘空间的浪费。加上 FDT 容量有限(最多只能有 512 项)等缺点, FAT16 逐渐被 FAT32 替代。在 Windows 中可采用 FAT32, 顾名思义, 它是 32 位的 FAT 表。

### 5.4.3 FAT32 磁盘格式特点

(1) 支持大硬盘及分区。FAT32 表的表项大小为 32 位, 则最多可有  $2^{32} = 4\text{GB}$  个簇。对高于 8GB 的硬盘, 分区大小为 1~2GB, 簇的大小为 4KB。FAT32 能支持的磁盘分区可达 2TB, 这对再大的硬盘也足够了。簇大小为 4KB, 也减少了磁盘空间的浪费。所以当你将原来 FAT16 的硬盘转换为 FAT32 后, 会发现硬盘的剩余空间增多了。

(2) 根目录下可容纳无数多个文件或目录。由于 FAT32 系统可将根目录表存于硬盘的任何位置, 且大小不受限制, 这样就没有了根目录只能有 512 项的限制。

(3) FAT32 采取对关键磁盘提供冗余备份, 使分区不易损坏或造成数据崩溃。

但 FAT32 也有缺点, 如不能与 FAT16 兼容而导致有些专为 FAT16 设计的文件没法在 Windows 下运行; 不能格式化已压缩的驱动器; 运行速度较慢等。

## 5.5 文件的操作

一个好的文件系统应该能提供种类丰富, 功能强大的文件操作命令, 以满足用户对文件的多种操作要求。而且, 用户在使用这些命令时, 希望简单方便, 即能有一个良好的用户界面。这也是一个文件系统受欢迎的重要方面。

### 5.5.1 有关文件操作的系统调用命令

不同的操作系统, 提供的有关文件操作的系统调用命令无论在数量上还是功能上都不可能完全一致。其中, 有六条系统调用命令是所有文件系统都应具备的, 即建立文件、打开文件、读文件、写文件、关闭文件、删除文件。

#### 1. 打开文件和关闭文件的必要性

文件系统通过文件目录表和文件控制块(FCB)来对文件进行管理。当要使用某个文件时, 文件系统通过文件名去查文件目录表, 以便在那里找到对应文件的 FCB, 获得文件的相关信息。因此, 文件目录表的存放位置就会影响系统的工作效率, 文件目录表应该放在内存还是外存呢?

通常把文件目录表作为文件存放在辅存, 在对文件进行任何操作之前, 可预先将该文件

的 FCB 复制到内存,以后对文件的操作就可通过内存中的 FCB 获得所需文件信息,避免了频繁访问磁盘,提高了存取速度。复制到内存的 FCB,称为活动文件目录,所有活动文件目录构成了一个活动文件目录表。

把文件的 FCB 预先复制到内存的操作就是打开文件。所以,对一个已建立好的文件进行读写等任何操作之前,必须要执行打开文件操作。

内存中活动文件目录表的大小是有限的,当对文件执行完读、写等操作后,暂时不使用该文件的话,就应将它在活动文件目录表中占用的表目撤销,归还系统空间,以供打开别的文件使用。关闭文件主要就是完成撤销表目的工作。因此,对一个打开的文件,使用后别忘了将它关闭。

## 2. 文件系统基本调用命令

文件系统基本调用命令有如下执行顺序:建立文件→打开文件→读/写文件→关闭文件→撤销文件。

### (1) 建立文件。

当用户希望在文件系统中建立一个新文件时,就需要使用建立文件命令。该命令将为新文件在相应的文件目录表中找一个空目录项,存放该文件的文件控制块(FCB),然后把文件名、文件属性、系统分配给文件的辅存地址等信息填入此目录项中,如图 5.29 所示,以后就可通过访问 FCB 获得文件信息。

当一个文件被创建后,它就一直存在于系统中,直到被撤销为止。

### (2) 打开文件。

打开文件命令的工作,就是将该文件的目录项内容(即 FCB)复制到内存,形成活动文件目录,如图 5.30 所示。对文件进行读写操作时,通过活动文件目录即可获得文件信息,而不必访问外存的文件目录,提高了速度。在多用户系统中,如果要打开的共享文件已被其他用户打开,则只需将活动文件目录的当前用户计数加 1。

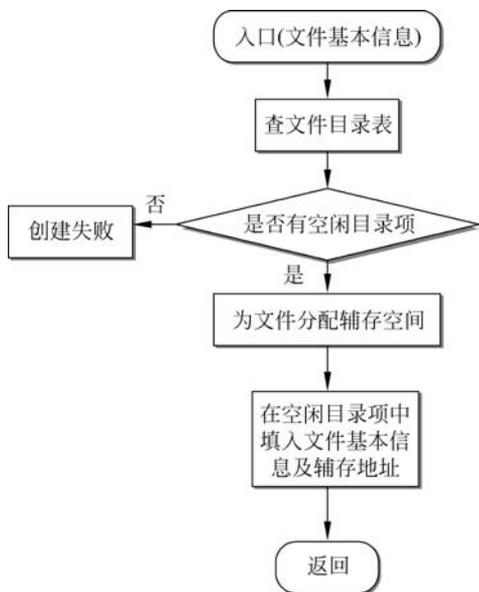


图 5.29 建立文件

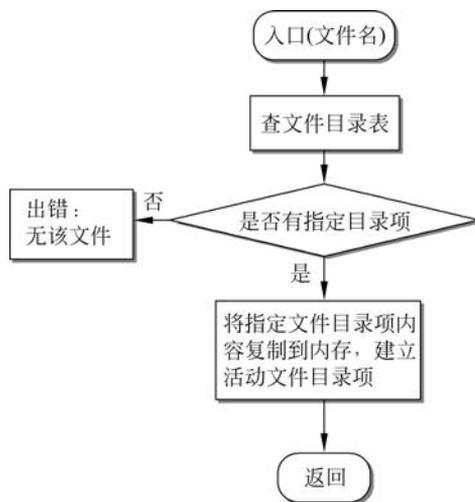


图 5.30 打开文件

## (3) 读/写文件。

文件打开后,就可根据用户的读、写命令对文件进行相应的读、写操作,如图 5.31 所示。

有些系统,将打开文件命令隐含在读/写命令中,即发读/写文件命令时,若在活动文件目录表中没找到对应的目录项,即文件还未打开,则先执行打开操作,然后才去读/写。

## (4) 关闭文件。

一个文件使用完毕后,用户应关闭此文件。关闭文件命令的主要功能就是根据文件名,将文件在内存活动文件目录表中的目录项撤销,如果要撤销的目录项已被修改过,则还要写回辅存的文件目录中,这样,就能保证 FCB 中的信息是最新的。关闭文件的工作过程如图 5.32 所示。如果是共享文件,则先将当前用户计数减 1,如果计数等于 0,表示已无人使用该文件,这时才真正做撤销活动文件表目的操作,否则该目录项保留。

## (5) 撤销文件。

当一个文件不再被需要时,用户执行撤销文件命令,收回文件占用的存储空间,撤销文件在辅存文件目录表中的目录项(即 FCB),文件于是真正消亡了。撤销文件的工作如图 5.33 所示。在多用户系统中,对于共享文件,只要将 FCB 中的共享用户计数减 1 即可,当计数值为 0 时才撤销该文件。

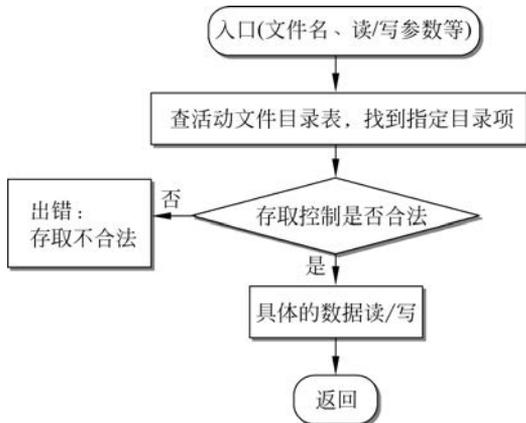


图 5.31 读/写文件

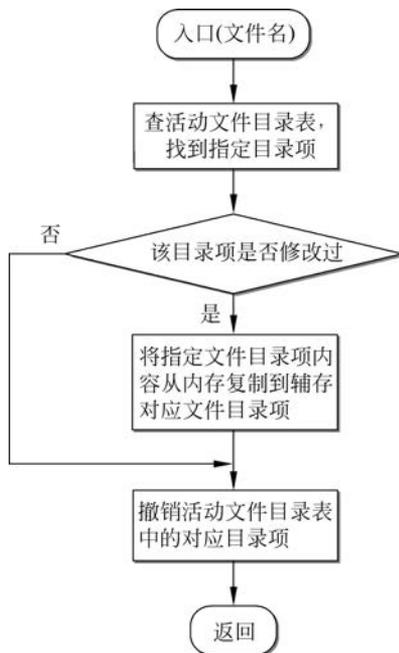


图 5.32 关闭文件

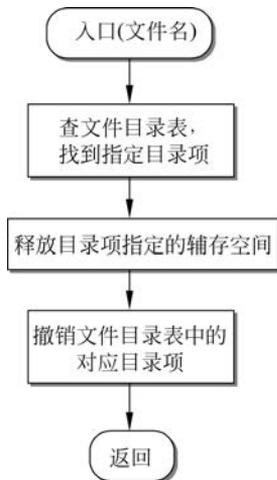


图 5.33 撤销文件

## 5.5.2 Linux 中的文件系统调用命令及工作过程

Linux 提供大量和文件操作有关的系统调用命令,如创建文件(Create)、打开文件(Open)、从文件中读取数据(Read)、向文件写入数据(Write)、关闭文件(Close)、挂装一个文件系统(Mount)、卸装文件系统(Umount)、设置文件系统的组标识符(Setfsgid)等。

本节主要介绍打开文件(Open)及其工作过程,其他系统调用命令的详细信息可查阅相关资料。

### 1. 目录文件、外存索引节点和文件存储块的关系

前面已经知道,对文件进行操作之前,都必须先调用 Open 来将指定文件打开,即将文件在外存中的有关目录信息、外存索引节点复制到内存。复制到内存的 i 节点称为内存索引节点或活动索引节点,由所有的内存索引节点构成内存索引节点表。Linux 系统利用文件描述符表和打开文件描述表这两个数据结构,配合内存索引节点表来完成 Open 系统的调用。三者关系如图 5.34 所示。

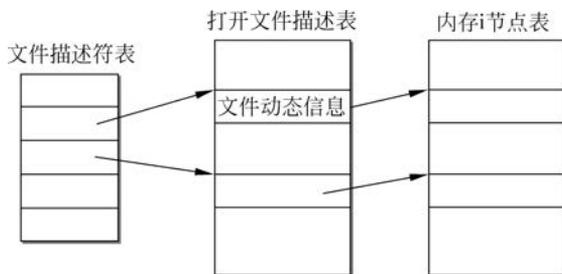


图 5.34 打开文件的三种数据结构的关系

### 2. 三种数据结构

三种数据结构如下。

#### (1) 内存索引节点。

内存索引节点的内容前面已介绍过。

#### (2) 打开文件描述表。

打开文件描述表的目录项称为打开的文件描述(Open File Description),打开的文件描述中包含指向内存索引结构的指针和文件的动态信息。

#### (3) 文件描述符表。

一个用户经常需要打开多个文件,这个情况记录在被称为文件描述符表的数据结构中。一个用户建立一个文件描述符表。该表中的每个目录项称为文件描述符(File Descriptors, FD),指向打开的文件描述表中的一个表项。

通过文件描述符,找到打开的文件描述,再通过打开的文件描述,就可找到内存索引节点,从内存索引节点中的地址索引数组,即能得到文件的真正物理地址。

## 5.5.3 Windows 中的文件系统

Windows 10 中使用的 NTFS 文件系统提供了许多 FAT 文件系统中没有的高级特征,

它支持文件级别安全、Unicode 文件名、文件压缩和文件系统恢复；提供磁盘限额指定，限额允许就给定的 NTFS 卷以用户为基础执行硬限制或软存储限制，以此限定用户的磁盘空间，在用户空间超过限额时拒绝或发出警报；引入了卷标装配点，使应用程序和用户通过多重卷标定信息，而不需了解卷标的物理挂接。另有索引服务、分布式文件系统(DFs)、活动目录等。

Windows 文件系统提供了许多文件操作。创建或打开文件 `CreateFile()`，可以针对所有能用文件流动表示的对象，如文件、控制台、通信端口、目录、磁盘、邮件位或管道等；文件读取 `ReadFile()`，以同步或异步方式读取文件中的指定数目的字节；文件写 `WriteFile()`，以同步或异步方式向指定文件中写入指定数目的字节；获取文件大小 `GetFileSize()`，返回指定文件的大小；文件删除 `DeleteFile()`，删除由文件名指定的已有文件；另有其他一些文件操作，涉及对文件目录、临时文件的搜索及利用。

## 5.6 文件的共享与安全

### 5.6.1 文件的共享

文件共享指一个文件被若干用户共同使用，文件系统的一个重要任务就是为用户提供共享文件的手段，这样，避免了系统复制文件的开销，并节省文件占用的存储空间。

#### 1. 实现文件共享的常用方法

实现文件共享的方法有以下两种。

##### (1) 绕道法。

给每个用户一个当前目录，用户对所有文件的访问都是相对于当前目录进行的，用户文件的路径名就由当前目录到共享文件通路上所有各级目录的目录名加上该文件的符号名组成，并规定到达文件的通路可用往上走的方法，系统用“\*”表示一个给定目录文件的父目录。如在图 5.18 的目录树中，假定用户 wang 的当前目录为 `/home/wang`，用户 wang 若要访问 zhang 的文件 a，使用路径名 `* /zhang/a`。

由于绕道法要花很多时间访问多级目录，导致搜索效率不高。可采用另一种共享方法——链接法。

##### (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 5.3 节介绍的基本文件目录和符号文件目录。由于该方法将文件的符号名和文件说明信息分开，因此，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。

#### 2. 实用系统中的文件共享方法

Linux 中采用硬链接和符号链接两种方法来实现共享，硬链接是基于索引节点的共享方式，符号链接则是利用符号链来实现共享的。

Windows 中采用了动态数据交换方法，即对象的链接与嵌入 (Object Linking and

Embedding, OLE)方法实现共享。例如,用“复制”命令,把要共享的表格复制到剪贴板中,切换到 Word 工作报告中,选择“编辑”→“选择性粘贴”→“粘贴链接”命令,就成功地为表格和工作报告建立了链接。现在,电子表格仍然保存在 Excel 中,而且工作报告中只保存了指向该表格的链接。这种方法一方面节省了内存,更重要的优点是具有自动更新的能力,即当表格改变时,改动将自动反映到目标文档中,实现了共享。

但是 OLE 只能共享本机资源,为了共享网络资源,Windows 也使用了符号链的方法,只要给出文件所在计算机网络地址和文件路径名,我们就可以访问网上任何允许访问的文件。对文件共享的指定由文件创建者完成(双击“我的电脑”图标,右击磁盘符号或要共享的文件→“属性”→“共享”命令,见图 5.35)。

一旦完成文件共享的创建,在该文件或目录的所有图标显示中就会出现特殊的共享标记(见图 5.36),即磁盘左下角的双人图片。



图 5.35 磁盘共享的创建



图 5.36 共享标记

要查看本文件系统中的所有共享目录及文件,可选择“控制面板”→“管理工具”→“计算机管理”→“系统工具”→“共享文件夹”→“共享”命令就能完成(见图 5.37)。

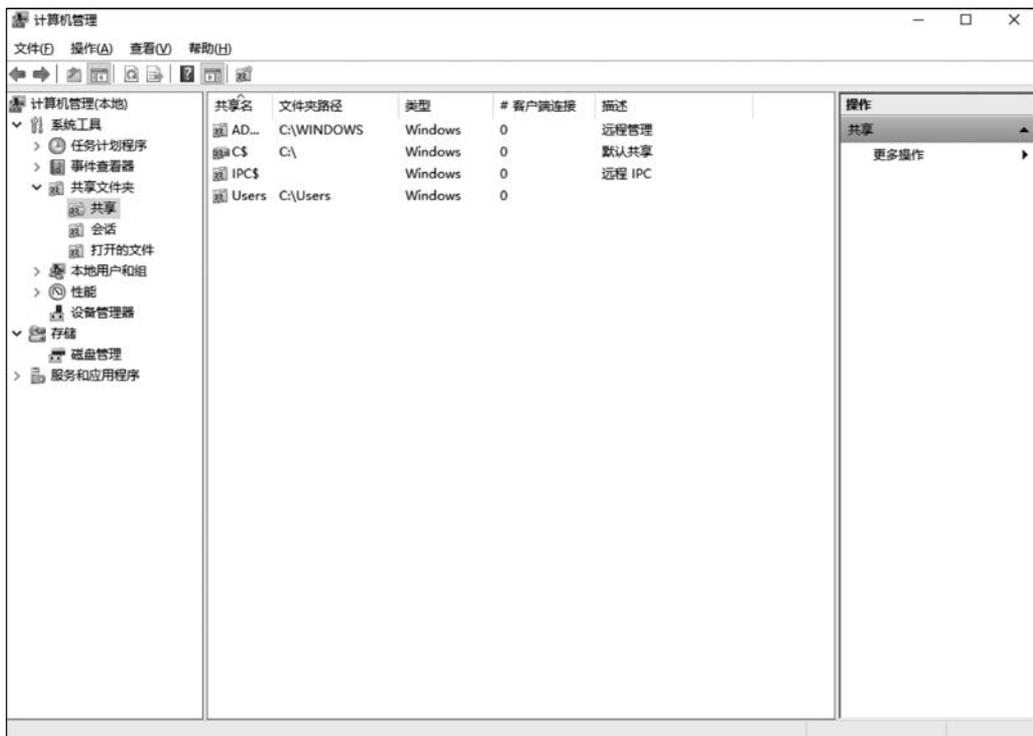


图 5.37 系统中的所有共享文件及目录

## 5.6.2 文件的安全

伴随着共享,文件的安全性成为一个重要的问题。面对错综复杂的用户,计算机要能识别出哪些用户是允许访问共享文件的,访问的权限有多大,而现在的黑客总是令人防不胜防。

因此,为了防止非法用户的访问,一个优秀的操作系统应该能提供层层保护功能,在多个级别中保证系统的安全性。文件的安全管理通常包含四个级别:系统级、用户级、目录级和文件级。

### 1. 系统级安全管理

这是第一层保护,即不允许非法用户进入系统。如果用户无法进入系统,就没法使用系统中的任何资源了。

在进入系统之前,所有用户都需要登录。因此,可设置一张用户注册表,里面存放允许登录的用户账号和密码。每次登录时,用户输入账号和密码,系统通过用户注册表进行验证之后,用户才能进入系统。

在 Linux 中,用户的各种信息是存放在口令文件/etc/passwd 中的,里面包含了用户名、口令密码、用户标识符、账号全名及其说明信息、用户主目录、用户的 shell。

用户账号的建立可以以 root 身份手工进行,也可以利用 Linux 系统提供的各种工具完

成,如 adduser 命令。而 root 账号的建立则是在安装 Linux 的过程中建立的。

Windows 中可以在一台本地机上设置多个用户,图 5.38 显示了本地机用户的账户类型修改界面(通过选择→“控制面板”→“用户账户”命令)。



图 5.38 用户账户类型修改

账户类型决定了用户在系统中的权限。

## 2. 用户级安全管理

当用户通过系统级验证,成功登录后,又要面临用户级安全的考验。

用户级安全首先将所有用户进行分类,然后为指定用户分配文件访问权,即决定用户对哪些文件能执行什么操作。

### (1) 用户分类。

不同的系统对用户的分类方法不完全一样,如有的系统把用户分为文件主、伙伴和一般用户。或按用户权限高低,将用户分为超级用户、系统操作员、用户和顾客。例如,Linux 系统中将用户分为四类。

① 超级用户 root: 指系统特权用户类,在系统中拥有至高无上的权力,不受任何限制,对所有的文件和目录具有完全权限,可以做任何事,所以一定要慎用 root 账号。

② 文件主: 指建立文件的用户。

③ 同组用户: 和文件主在同一组的用户,他们具有相同的权限。

④ 其他用户: 指不属于上面三类的任何其他用户。

Windows 中将用户分成若干组,每组的权利状态如图 5.39 所示。

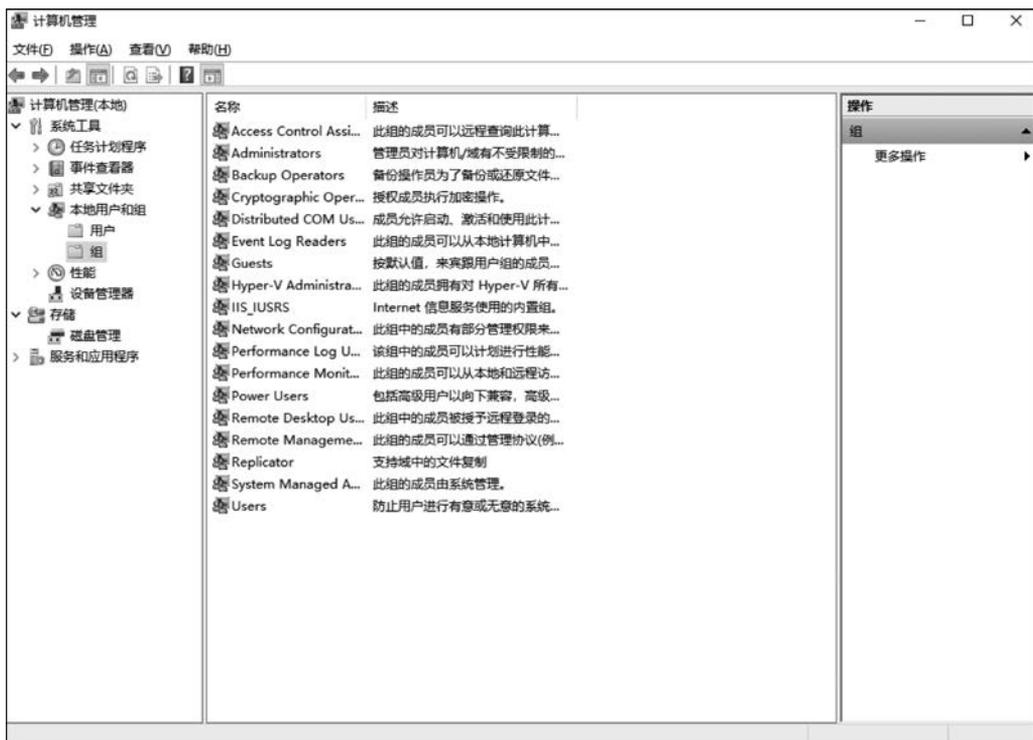


图 5.39 用户分类状态

## (2) 用户的资源访问权。

如果对某一个组名赋予某种权限,则该组中所有成员都具有这种权限。通常对文件可定义建立、打开、读、写、修改、删除权限,对目录可定义查询、建立、改名、删除权限。

### 3. 目录级安全管理

用户最终目的是为了访问文件,而文件都是存放在某个目录下,因此,用户必须先去看目录。系统会为每个目录也设置权限,只有当用户权限和目录权限一致时,用户才能获得对这个目录的有效访问权。

Linux 中,对目录可以执行的操作有读、写、搜索,分别用 r、w、x 代表,其中读表示可以读包含在目录中的文件名称,写表示可以到目录中增加或删除某个文件,搜索表示能访问该目录所包含的文件或子目录。

在 Windows 中,右击文件夹,选择“属性”命令,在弹出的文件夹属性窗口中,可对文件夹的属性进行设置。

### 4. 文件级安全管理

通过前三个级别的检查后,用户权限和文件的属性一致时用户才能访问文件。

文件权限是由系统管理员或文件主设置的,通常可设只读、执行、读写、共享、隐含、系统等。在 Linux 中,文件属性有读、写、执行,分别用 r、w、x 表示。图 5.40 显示了 Linux 中文件对应不同用户的访问权。

图 5.40 中文件的访问权用 10 个字符表示,第一个字符指定了文件类型,后面 9 个字符每三个分别代表文件主、组用户和其他用户的访问权限。横线代表空许可,r 代表只读,w

```

[stu05@lxsvu stu05] $ ls -l
total 44
-rw-r--r-- 1 stu05 stu 129 May 23 2005 \
-rw-r--r-- 1 stu05 stu 15 Jun 6 09:12 aa
-rw-r--r-- 1 stu05 stu 0 May 23 2005 aaa
-rw-r--r-- 1 stu05 stu 127 May 23 2005 abc~
drwxr-xr-x 2 stu05 stu 4096 May 16 2005 abciji.c/
-rw-r--r-- 1 stu05 stu 29 May 23 2005 asdf
-rw-r--r-- 1 stu05 stu 142 May 23 2005 asdf~
drwxr-xr-x 2 stu05 stu 4096 May 16 2005 b.d/
-rw-r--r-- 1 stu05 stu 25 May 23 2005 io
drwxr-xr-x 2 stu05 stu 4096 May 23 2005 subdir/
drwx----- 2 stu05 stu 4096 Jun 6 09:32 tmp/
-rw-r--r-- 1 stu05 stu 24 May 23 2005 uhm~
[stu05@lxsvu stu05] $

```

图 5.40 Linux 中文件对应不同用户的访问权

代表写, x 代表可执行。在通常意义上, 一个目录也是一个文件。如果第一个字符是横线, 表示是一个非目录的文件; 如果是 d, 表示是一个目录。

例如, 文件 asdf 的访问权限为 -rw-r--r--, 表示它是一个普通文件, 其文件主有读写权限, 与文件主同组的用户只有读权限, 其他用户也只有读权限。

在 Windows 中, 右击文件名, 选择“属性”命令, 在弹出的文件属性窗口中, 可对文件的属性进行设置(如图 5.41 所示), 该文件可以设置的属性有只读和隐藏。

只有通过了四级安全控制, 用户才能对文件进行访问, 这样, 文件的安全性大大增强了。

### 5.6.3 安全控制手段

为了实现四级安全管理, 一般可利用四种控制手段: 存取控制表、用户权限表、口令和密码。

#### 1. 存取控制表

为每个文件设置一张存取控制表, 存放在文件的 FCB 中。存取控制表中包含文件主、同组用户、其他用户对该文件的存取控制权限, 如图 5.42 所示。UNIX、Linux 系统都采用该方法。

如果文件数很多, 每个文件都设一张存取控制表, 会增加系统空间开销。

#### 2. 用户权限表

存取控制表是以文件来考虑用户的存取权限, 现在以用户为主考虑可对哪些文件有存取权限。

为每个用户在特定区域建立一张用户权限表, 把用户或用户组所要存取的所有文件名及对应的存取权限都放在该表中, 如图 5.43 所示。

当用户对一个文件提出存取要求时, 系统可到相应的文件权限表中进行查找, 以确定用户的存取权限是否合法。该方法的缺点之一是当用户对大量的文件都有存取权时, 用户权



图 5.41 文件属性设置

限表将很长。

文件名: F1	
文件主	RWE
同组成员	RW
其他用户	R

图 5.42 存取控制表

文件 用户	F1	F2	...
zhang	RW	R	...

图 5.43 用户权限表

### 3. 口令

存取控制表和用户权限表都会占用大量的存储空间,因此,可采用另一种较简单的方法——口令。

用户为自己的每个文件设置一个口令,存放在文件的 FCB 中。任何用户要存取该文件,都必须提供和 FCB 中一致的口令,才有权存取。

设置口令,管理简单,使用方便,且口令所占内存空间少。但由于口令存放在 FCB 中,很容易被窃取,保密性差,而且如果文件主更改了口令,需要重新通知其他允许存取文件的用户,这就不太方便了。

### 4. 密码

在文件建立保存时,加密程序根据用户提供的密钥对文件进行编码加密,在读取文件时,用户提供相同的密钥,解密程序根据该密钥对加密文件进行译码解密,恢复为源文件。

只有知道密钥的用户才能正确访问文件,由于密钥不存放在系统中,该方法保密性很强。但耗费大量编码、译码时间,系统开销大而且降低了访问速度。

一般可将几种安全控制手段综合应用。

Windows 的 NTFS 提供一种核心文件加密技术,该技术用于存储已加密的文件。对加密该文件的用户,加密是透明的。这表明不必在使用前手动解密已加密的文件。然而,获得未经许可的加密文件和文件夹物理访问权的入侵者将无法阅读文件和文件夹中的内容。如果入侵者试图打开或复制已加密文件或文件夹,入侵者将收到拒绝访问消息。

正如设置其他任何属性(如只读、压缩或隐藏)一样,通过为文件夹和文件设置加密属性,可以对文件夹或文件进行加密和解密。如果加密一个文件夹,则在加密文件夹中创建的所有文件和子文件夹都自动加密。

如果将加密的文件复制或移动到非 NTFS 格式的卷上,该文件将会被解密。如果将非加密文件移动到加密文件夹中,则这些文件将在新文件夹中自动加密。然而,反向操作不能自动解密文件。

Windows 还提供系统文件保护,通过对操作系统文件进行保护来阻止对重要系统文件的替换。一旦发生系统文件被删除或重写的情况,系统文件保护功能可用原始文件来替换已被删除或重写的系统文件,而这些原始文件则来自于系统文件保护功能所维护的缓冲内存或安装媒介。

## 5.6.4 备份

为避免系统遭受严重损坏后数据的流失,备份是一个不错的方法。备份就是创建逻辑驱

动器或者文件夹中数据的副本,然后将数据存储到其他存储设备。如果硬盘上的原始数据被意外删除或覆盖,或因为硬盘故障而不能访问该数据,则可以十分方便地从存档副本中还原该数据。图 5.44 为 Windows 的备份工具(选择“控制面板”→“系统和安全”→“备份和还原(Windows 7)”命令)。



图 5.44 Windows 的备份工具

由于没有做任何备份,因此图 5.44 中未显示任何备份信息。

## 5.7 科技前沿——银河麒麟

银河麒麟(KylinOS)是一个由国防科技大学研发的国产服务器操作系统,拥有层次式内核,其安全等级达到结构化保护级,能支持多种微处理器和多种计算机体系结构,可与 Linux 目标代码兼容。

经过对该系统的严格测评,其结果表明:银河麒麟不仅具备典型服务器操作系统的全部功能,还具有高安全性、高可用性、强实时性、可扩展性和软硬件适配性等。其系统整体性能与国际主流 UNIX 操作系统相当,部分性能指标以及实时性指标表现更好。该系统已通过自由标准组织的 Linux 标准基认证,并在国内率先通过公安部等有关部门的安全认证。

国防科技大学已与联想集团等签署银河麒麟的产业化合作协议,并将其成功应用于金融、政府、教育、国防等领域。

## 5.8 本章小结

文件系统的主要功能是对大量的文件进行管理,涉及对文件逻辑结构和物理结构的转换,对文件目录、文件存储空间的管理,并提供对文件的共享和安全机制,通过文件系统还可对文件进行多种操作。文件逻辑结构有无结构的流式文件和记录式有结构文件,受存储设备类型的影响,文件有连续文件、串联文件(包括文件映照)和索引文件三种物理结构。为便于管理、组织大量的文件,现代计算机系统大都采用树形目录结构。对文件存储空间的管理有位图、空白文件目录、空闲链表法。DOS 采用了 FAT 格式对文件存储空间进行管理,Windows 采用改进的 FAT32 文件系统和 NTFS 文件系统,Linux 则用 EXT2 进行管理。通过文件系统,对文件可进行创建、打开、读、写、关闭、撤销等操作。文件系统还提供对文件的共享手段和多级安全控制。

### 习题

- 5.1 什么是文件? 实用系统中文件命名有何规则?
- 5.2 文件有哪些逻辑结构? 各有什么意义?
- 5.3 文件有哪些类型? 为什么要将文件分类?
- 5.4 以 Windows 和 Linux 为例,你认为实用系统中文件分类的依据是什么。
- 5.5 将一些外部设备也看成文件,其根据是什么? 将给用户带来哪些好处?
- 5.6 什么是文件系统? 文件系统具有哪些功能?
- 5.7 从文件系统功能的角度谈一谈实用系统中的文件系统。
- 5.8 有哪些文件的存储方法? 文件逻辑结构与文件的存储方法有何联系?
- 5.9 文件的物理机构是什么? 文件的物理结构与文件的存储介质有何关系?
- 5.10 连续文件、串联文件、文件映照、索引文件有何异同点? 为什么?
- 5.11 在 Linux 操作系统中,是如何存放巨型文件的?
- 5.12 文件控制块的作用是什么? 包含哪些内容? 你认为 Linux 的索引节点是什么?
- 5.13 文件目录的作用是什么?
- 5.14 假设某系统现有两个用户 user1 和 user2 共用其文件系统。假定 user1 有四个文件,其文件名为 A、B、C、D; user2 也有四个文件,其文件名为 A、E、F、G。已知: user1 的文件 A 和 user2 的文件 A 实际上不是同一个文件, user1 的文件 C 和 user2 的文件 E 是同一文件。要求如下。
  - (1) 画出该系统的文件目录结构,使这两个用户能共享该文件系统而不致造成混乱。
  - (2) 此题中有无重名问题? 如有,是如何解决的?
  - (3) 此题中有无文件共享问题? 如有,是如何解决的?
- 5.15 多级目录是如何解决文件重名和查找速度慢的问题的?
- 5.16 什么是路径? 什么是绝对路径? 什么是相对路径? 一个文件能否同时拥有这几种路径?

- 5.17 为什么要引入基本文件目录和符号文件目录的概念?
- 5.18 Linux 系统的目录结构是如何构造的? 采用索引节点的方法有何意义?
- 5.19 从占用空间、访问速度和管理三方面比较位示图、空白文件目录和空闲链表法。
- 5.20 简要说明 FAT 文件系统。
- 5.21 为什么文件操作中设立了打开和关闭环节? 可否去掉这两个操作原语?
- 5.22 说明文件创建和撤销的步骤, 及其每一步的作用。
- 5.23 通过实验说明 Windows 或 Linux 中的文件系统调用命令的意义和使用过程。
- 5.24 有哪些文件共享的方法? 以 Linux 为例说明这些方法的应用。
- 5.25 可以将文件的安全管理分为几个级别? 用实用系统中的例子说明这些安全管理的内容。
- 5.26 比较四种控制手段: 存取控制表、用户权限表、口令和密码。
- 5.27 为什么需要文件备份? 如何完成文件备份?