SAMS

北京科海培训中心

PETER NORTON'S COMPLETE GUIDE TO

DOS 6.22

More than 1,000,000 copies sold!

SIXTH EDITION

实用大全

[美] Peter Norton 著
夏小禹 赵 军 戴 笠 译
赵 军 审校

清华大学出版社

TP316
N91/1

SAMS 北京科海培训中心

DOS 6.22 实用大全

（第六版）

[美] Peter Norton 著

夏小禹 赵 军 戴 笠 译

赵 军 审校

清华大学出版社

(京)新登字 158 号

Peter Norton's Complete Guide to DOS 6.22
Sixth Edition

Copyright © 1994 by Peter Norton.

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher.

本书英文版由 Prentice Hall 出版社下属的 SAMS 公司于 1994 年出版。版权为 Peter Norton 所有。本书的中文版专有出版权由 SAMS 公司授予北京科海培训中心和清华大学出版社合作共同出版并发行。未经出版者书面允许不得以任何方式复制或抄袭本书内容。

版权所有，盗版必究。
本书封面贴有 PRENTICE HALL 激光防伪标志，无标志者不得销售。

书　名：DOS 6.22 实用大全
原著者：Peter Norton
出版者：清华大学出版社(北京清华大学校内，邮编 100084)
印刷者：北京门头沟胶印厂
发　行：新华书店总店北京科技发行所
开　本：16　印张：47.375　字数：1152 千字
版　次：1995 年 8 月第 1 版　1995 年 8 月第 1 次印刷
印　数：00001～10000
书　号：ISBN 7-302-01962-2 /TP・906
定　价：70.00 元

导 引

许多 PC 新用户试图忽视 DOS。他们把精力仅集中在每天使用的一两个应用程序上。但值得注意的是，一个字处理或数据库管理程序不可能查出并校正磁盘目录结构中的错误，这些程序不能修复错误删除的文件，而且，它们也不能检测和清除感染到磁盘上的病毒。用户对 DOS 了解得越透彻，其数据也就会越安全。同时这样还能明显地改善带有 DOS 的系统的总体性能。例如，用户可以使应用程序运行更快，也可将磁盘上的数据压缩到只有原来的一半。

本书的内容概要

在本书中，读者将学会如何做下列事情：

- 把系统升级到 DOS 6.22
- 使用 DOS 的屏幕 Help 系统
- 保护系统免受病毒侵害
- 保护数据以防意外删除
- 使用 DOS Shell 或 DOS 命令管理磁盘、目录和文件
- 设置 DOS Shell 使日常工作更为容易(在没有 Windows 或第三方销售商提供的 Shell 的情况下)
- 在 DOS Shell 下同时使用多个程序
- 使用 DOS 的全屏幕编辑器创建和修改文本文件
- 使用 DOS 的批处理编程命令编写自己的简单微型程序
- 理解和利用 CONFIG.SYS 文件
- 使用 DOS 的新备份系统保护和恢复文件
- 优化系统使用内存的方法
- 优化硬盘；读者既可尽量减少读写数据所花的时间，也可使可用空间加倍
- 使用带 Windows 的 DOS
- 在 Internet 上使用 DOS
- 在一个多媒体 PC 机上使用 DOS
- 升级到 QBASIC，如果已用 BASIC 的早期版本编程的话
- 通过 DOS 的 INTERLNK 工具把两个计算机连结起来
- 充分利用硬件，包括打印机、显示器、键盘和磁盘驱动器
- 建立系统使之适用于另一个国家的惯例和字符集

在本书的后面，提供了所有 DOS 命令的参考和在 DOS 6 中 Microsoft 决定砍掉的一定数量的印刷资料。大部分命令和驱动程序只在屏幕帮助系统中描述，但是这些信息常常是不完整的。如果你打算跳过基础部分，直接进入 DOS 命令和驱动程序，将发现两个最好的参考

材料：本书最后的“命令用法参考”和附录 A“设备驱动程序”。

本书介绍了 DOS 的基础知识和增强技术，还有 DOS 6 发布的主要的新工具，如 MEMMAKER (DOS 6.0)和 SCANDISK (DOS 6.2)。无论对于 DOS 新用户还是刚从早期 DOS 版本升级的 DOS 用户，都会发现在掌握新操作系统过程中所需的所有信息。

本版中的新内容

我已将《DOS 6.22》加到《DOS Guide》中，并且作了一些彻底的改变。最后我将两本书综合成一本书，把《Peter Norton's Advanced DOS》的从前版包含在此书中，这样读者就可找到关于 DOS 基础知识和增强信息的一个完整指南。第一部分到第四部分覆盖了 DOS 的基础知识，第五部分为 DOS 的熟练用户提供了丰富的增强技术。

另外，我还在本书中增加了以下几个新特点：

- 通过查询本书可找到直接包含在上下文中的操作步骤。这些步骤是一些实用练习，可指导你尝试大部分 DOS 工具。如果仔细完成这些步骤，你将明白 DOS 如何工作，有时，将明白 DOS 为什么不工作，这同样重要。

  每章中的这种练习是累积的。我的意思是说，前面的练习常常是为后面的练习打基础的。例如，你可能在前面一章创建了一个文件，在后几页可能会对它进行更名，移到另一个目录，在该章的结尾也许又会将它删除掉。因此，当你阅读一章时，要么做完所有的练习，要么不做任何练习。如果你跳过一些练习，就可能在完成余下的练习时发现文件或目录并不存在。

  但是，你也不必在本书中依次完成每章的练习。每章的练习与其他章节的练习是独立的，这样你就可选择阅读本书各章的顺序。而且，在任何一章的结尾，你应消除对系统所作的改变，除了那些有选择地保留下来的之外。(这里有例外：在第 23 章“MSBACKUP 和 MWBACKUP”中的练习在第 24 章“Restore 和 Compare”中应仍然存在，这样就可以练习备份和恢复文件了)。

  为了创建这些练习，需对你的 PC 机作一些基本的假定。这些练习建立在以下假定的基础上：系统已升级到 DOS 6.2 或 DOS 6.22；驱动器 C 是启动驱动器；已在此驱动器的 DOS 目录下安装了 DOS；没有删除 DOS 带来的任何文件(特别是.TXT 文件)；而且在此驱动器上至少有 1MB 的空闲空间，这些练习还假定驱动器 A 是一个软盘驱动器。如果不满足这些基本的假定，就应该调整系统。例如，如果把 DOS 安装在驱动器 D 上，则应把本书中对应的驱动器名换为 D。

- 本书还为常见任务逐条列出了详细的“过程”(Procedures)，这些常用的 DOS 任务包括删除一个文件，拷贝一个文件和创建一个新目录等。这些过程在伴随的文本中作了归纳，它们有助于对学习新任务提供一个总览。但更重要的是，当你在阅读一章几个月后需要完成一个任务，它们可用作快速复习和操作指导，这些过程列在特殊的方框内，很容易发现。
- 本书为最普遍的 DOS 问题，例如不能启动，或者在使用一个 DOS 实用程序时得到了一个错误信息等增加了“答疑”部分。每个“答疑”解释为什么会发生这种问题以及如何解决它。“答疑”用楷体字表示；下面就是第 6 章“目录管理”中的一个例子：

答疑： 当试图转换到软驱时，获得了一个错误消息，即使这时有一个软盘在驱动器中。

可能是软盘没被 DOS 格式化(它可能被其他的操作系统格式化)，或者它已被严重破坏以致 DOS 不能再读它。

如果试图读一张别人提供的盘，让他再提供一张，要保证用 DOS 格式化过。

如果试图使用一张新的空盘，必须首先格式化。第 17 章“格式化磁盘和取消格式化”将介绍如何格式化磁盘。

- 增加了几种新的提示。“作者忠告”推荐了一些好的 DOS 实践策略，例如使文件名具有意义。“网络注意”建议如何在 PC 联网上完成当前任务。“预览”注释让你知道以后几章中关于一个主题的更多信息。
- 增加了关于 Internet 和 Windows，以及 DOS 和多媒体 PC 机的章节。

本书约定

本书使用了以下约定：

- ALLCAPS(全部大写) 通常在文本情况下，命令、文件名、目录名、驱动器名和程序名都用大写使它们从上下文中突出出来。例如表达式“the copy program”指任何可进行拷贝的程序，而“the COPY program”特别指DOS的名为COPY的程序。虽然DOS通常忽略大小写，但是在本书中DOS命令全用大写表示。当然，在实际操作中你可输入大小写混合的命令，而许多用户乐于全部用小写字符工作。极少的地方DOS要区分大小写，这将在具体章节中指出。
- 用加号(+)连接键 两个或多个键的名字用+连接表示按下第二个键的同时要按住第一个键。例如，“Ctrl+Break”表示按Break键的同时要按住Ctrl键。
- 特殊字体类型(Special typeface) 屏幕显示和屏幕信息用特殊字体显现。
- 文件名(Filespecs) 表达式filespecs用来表示一个元素集，这个元素集用来确定一个文件或文件集。它可以包括一个驱动器名，一个路径名，一个文件名及一个扩展名。在许多情况下，单独一个驱动器名已足够，此时可选择这个驱动器当前目录下的所有文件。在其他情况下，你会发现具体化一个目录、一个文件名是必要的，有时可能还要使用扩展名。

目 录

第一部分 DOS运行环境概述

第1章 PC系统概述 …… (1)
1.1 硬件 …… (1)
1.1.1 系统主机 …… (1)
1.1.2 键盘 …… (4)
1.1.3 鼠标 …… (7)
1.2 软 件 …… (7)
1.2.1 系统软件 …… (8)
1.2.2 应用程序 …… (8)
1.3 磁盘 …… (8)
1.4 数据 …… (9)
1.4.1 数据如何存储 …… (9)
1.4.2 Kb,Mb,Gb和其他 …… (9)
1.5 用户 …… (10)
1.6 自测练习 …… (10)
第2章 DOS概述 …… (11)
2.1 什么是DOS …… (11)
2.2 DOS的工作方式 …… (11)
2.2.1 命令提示符 …… (11)
2.2.2 DOS Shell …… (12)
2.3 DOS的功能 …… (14)
2.3.1 命令提示符功能 …… (14)
2.3.2 批处理程序功能 …… (15)
2.3.3 磁盘和数据管理功能 …… (15)
2.3.4 保护和恢复数据 …… (15)
2.3.5 内存工具 …… (16)
2.3.6 优化系统性能 …… (16)
2.3.7 硬件功能 …… (17)
2.3.8 其他服务功能 …… (17)
2.4 DOS 6的新内容 …… (17)
2.5 自测练习 …… (19)
第3章 DOS的数据组织方式 …… (20)
3.1 DOS控制数据的方式 …… (20)
3.2 文件 …… (21)
3.2.1 文件名 …… (21)
3.2.2 全程文件名 …… (22)
3.2.3 属性 …… (23)
3.3 目录(Directories) …… (23)
3.3.1 根目录 …… (24)
3.3.2 目录树 …… (24)
3.3.3 当前目录 …… (25)
3.3.4 路径名 …… (26)
3.3.5 文件名中的路径 …… (26)
3.4 分配单元(Allocation Units)和文件分配表(File Allocation Table) …… (27)
3.5 DOS如何删除一个文件 …… (28)
3.6 自测练习 …… (28)
第4章 系统设置 …… (29)
4.1 基本设定 …… (29)
4.2 准备安装 …… (29)
4.3 作出关于备份的决定 …… (31)
4.4 运行SETUP …… (32)
4.5 系统设置中问题的处理 …… (32)
4.6 SETUP对系统的影响 …… (33)
4.7 查看附加资料 …… (33)
4.8 搜寻病毒 …… (34)
4.9 启动Delete Sentry …… (34)
4.10 制作一张紧急启动盘 …… (37)
4.11 取消安装DOS 6 …… (37)
4.12 DELOLDOS …… (38)
4.13 使用Expand …… (38)

第二部分 DOS Shell

第 5 章 DOS Shell 概述 …… (39)
5.1 决定什么时候使用 Shell …… (39)
5.2 启动 Shell …… (39)
5.3 探索 Shell 屏幕 …… (41)
5.4 使用 DOS Shell 的屏幕帮助系统 …… (43)
5.4.1 对话框 …… (44)
5.4.2 命令按钮 …… (44)
5.4.3 滚动 …… (44)
5.4.4 使用 Help 菜单 …… (46)
5.4.5 退出 DOS Shell …… (49)
5.5 自测练习 …… (49)

第 6 章 目录管理 …… (50)
6.1 目录管理 …… (50)
6.2 选择一个驱动器 …… (50)
6.2.1 选择一个空驱动器 …… (52)
6.2.2 更新驱动器 …… (52)
6.2.3 重画屏幕 …… (53)
6.3 创建目录 …… (53)
6.4 选择目录 …… (55)
6.5 裁剪和扩展目录树 …… (56)
6.6 目录更名 …… (58)
6.7 删除目录 …… (59)
6.8 自测练习 …… (60)

第 7 章 Shell 中的文件选择 …… (61)
7.1 在 Shell 中选择文件 …… (61)
7.2 选择单个文件 …… (61)
7.3 选择多个文件 …… (62)
7.3.1 用鼠标选择多个文件 …… (63)
7.3.2 将选择和取消选择结合起来 …… (63)
7.3.3 用键盘选择多个文件 …… (64)
7.3.4 选择和取消选择所有的文件 …… (65)
7.4 当没有文件被选择时 …… (65)
7.5 从多个目录选择文件 …… (66)
7.6 自测练习 …… (67)

第 8 章 在 DOS Shell 中拷贝和移动文件 … (68)
8.1 在 DOS Shell 中拷贝和移动文件 …… (68)
8.2 使用 Replacement Confirmation 功能 ……
…… (68)
8.3 拷贝单个文件 …… (69)
8.4 移动单个文件 …… (72)
8.5 拷贝和移动多个文件 …… (73)
8.6 拖动文件 …… (74)
8.6.1 改变视图 …… (76)
8.7 理解拷贝 …… (77)
8.8 从一个软盘移动和拷贝文件到另一个
软盘 …… (77)
8.9 移动和拷贝一大组文件到多个软盘 ……
…… (78)
8.10 各种问题 …… (78)
8.11 自测练习 …… (79)

第 9 章 Shell 中的其他文件管理 …… (80)
9.1 Shell 中的文件管理 …… (80)
9.2 创建实例文件 …… (80)
9.3 管理属性 …… (80)
9.3.1 浏览属性 …… (81)
9.3.2 改变属性 …… (82)
9.3.3 改变多个属性 …… (83)
9.4 文件更名 …… (84)
9.5 浏览文件 …… (85)
9.6 打印文件 …… (86)
9.6.1 通过 Shell 打印 …… (87)
9.6.2 使用打印队列 …… (88)
9.6.3 选择一个打印程序 …… (88)
9.7 删除一个文件 …… (89)
9.8 删除多个文件 …… (89)
9.9 自测练习 …… (91)

第 10 章 从 DOS Shell 运行程序 …… (92)
10.1 从 Shell 中运行文件 …… (92)
10.2 打开对象 …… (92)
10.3 从文件清单运行程序 …… (93)
10.4 放一个数据文件在一个程序文件上 ……
…… (94)
10.5 打开一个关联文件 …… (95)
10.6 从程序清单运行程序 …… (96)
10.7 在命令提示符下工作 …… (97)
10.8 任务切换 …… (98)
10.8.1 启动任务 …… (98)
10.8.2 任务间切换 …… (99)

10.8.3 结束任务 …… (101)
10.8.4 结束任务切换点 …… (102)
10.9 自测练习 …… (102)

第 11 章 自定义 Shell …… (103)
11.1 总览 …… (103)
11.2 管理程序清单 …… (103)
11.2.1 在组下工作 …… (104)
11.2.2 定义一个组 …… (104)
11.2.3 定义一个程序项 …… (106)
11.2.4 拷贝程序项 …… (108)
11.3 选择显示选项 …… (110)
11.4 定制屏幕颜色 …… (111)
11.5 定制文件清单格式 …… (112)
11.5.1 筛选文件清单 …… (112)
11.5.2 显示驱动器上所有的文件 …… (112)
11.5.3 找到不知道位置的文件 …… (113)
11.5.4 显示隐含和系统文件 …… (113)
11.5.5 在一组软盘中定位文件 …… (114)
11.5.6 给文件清单排序 …… (114)
11.6 自测练习 …… (115)

第三部分 DOS 命令

第 12 章 在命令提示符下工作 …… (116)
12.1 理解命令 …… (116)
12.2 到达命令提示符 …… (116)
12.2.1 解释命令提示符 …… (117)
12.2.2 辅助命令提示符 …… (117)
12.2.3 File,Run 命令 …… (118)
12.2.4 主要命令提示符和辅助命令提示符的优点和不足 …… (118)
12.3 理解命令格式 …… (119)
12.3.1 程序的名字 …… (120)
12.3.2 参数和选项 …… (120)
12.3.3 在命令中使用路径 …… (121)
12.4 重定向命令输出 …… (122)
12.5 命令参考介绍 …… (124)
12.6 获取命令的屏幕帮助 …… (125)
12.6.1 获取命令语法的屏幕帮助 …… (125)
12.6.2 得到带有完整命令信息的屏幕帮助 …… (126)
12.7 PATH 程序 …… (131)
12.8 PROMPT 命令 …… (133)
12.9 自测练习 …… (134)

第 13 章 DOSKEY …… (135)
13.1 理解 DOSKEY 做什么 …… (135)
13.2 加载 DOSKEY TSR …… (135)
13.3 使用已输入过的命令 …… (136)
13.4 编辑命令 …… (138)
13.4.1 插入和覆盖编辑模式 …… (138)
13.5 组合命令 …… (139)
13.6 使用宏 …… (140)
13.6.1 使用可替代的参数 …… (141)
13.6.2 控制 DOSKEY 缓冲区大小 … (142)
13.6.3 在 AUTOEXEC.BAT 中加载 DOSKEY …… (143)
13.7 浏览命令和宏 …… (143)
13.7.1 修订和删除宏 …… (144)
13.8 自测练习 …… (144)

第 14 章 目录管理 …… (145)
14.1 总览 …… (145)
14.2 改变缺省驱动器和目录 …… (145)
14.2.1 显示在另一个驱动器上的缺省目录的名字 …… (146)
14.2.2 改变另一个驱动器上的缺省目录 … …… (146)
14.3 显示目录树 …… (147)
14.4 列出一个目录 …… (148)
14.4.1 依照属性选择文件 …… (150)
14.4.2 清单排序 …… (151)
14.4.3 筛选名字 …… (151)
14.4.4 列出目录分枝的清单 …… (152)
14.4.5 找到一个丢失的文件 …… (152)
14.5 创建一个目录 …… (153)
14.6 为目录更名 …… (154)
14.7 删除一个目录 …… (155)
14.8 删除一个目录分枝 …… (155)
14.9 自测练习 …… (157)

第 15 章 拷贝和移动文件 …… (158)
15.1 总览 …… (158)
15.2 COPY 命令的使用 …… (158)

15.2.1 多个文件的拷贝 …… (160)
15.2.2 替换已存在的拷贝 …… (160)
15.2.3 用 COPY 快速打印文本文件 ……
…… (162)
15.3 XCOPY 命令的使用 …… (162)
15.3.1 指定一个目录或文件 …… (163)
15.3.2 使用 XCOPY 拷贝多个文件 ……
…… (164)
15.3.3 拷贝整个目录分枝 …… (165)
15.4 移动文件 …… (166)
15.4.1 移动多个文件 …… (167)
15.4.2 使用 MOVE 进行文件更名 … (168)
15.5 清除目录 …… (168)
15.6 自测练习 …… (169)
第 16 章 增加的文件管理 …… (170)
16.1 总览 …… (170)
16.2 创建一些练习文件 …… (170)
16.3 使用 REN 进行文件更名 …… (170)
16.4 用 ATTRIB 命令管理文件属性 … (172)
16.4.1 改变隐含文件和系统文件的属性
…… (173)
16.4.2 设置整个目录分枝范围内的文件属性
…… (174)
16.4.3 对目录的属性进行操作 …… (174)
16.5 文件删除 …… (174)
16.5.1 删除多个文件 …… (176)
16.5.2 删除受保护的文件 …… (176)
16.5.3 删除目录中的所有文件 …… (176)
16.6 用 TYPE 显示文件 …… (177)
16.7 用 PRINT 打印文件 …… (178)
16.8 用 FC 比较文件 …… (179)
16.8.1 二进制比较 …… (179)
16.8.2 ASCII 比较 …… (181)
16.8.3 重新同步 …… (182)
16.8.4 比较选项 …… (183)
16.9 自测练习 …… (183)
第 17 章 格式化磁盘和取消格式化 …… (184)
17.1 总览 …… (184)
17.2 格式化的工作方式 …… (184)
17.2.1 格式化新软盘 …… (184)
17.2.2 格式化新硬盘 …… (186)
17.2.3 重新格式化软盘 …… (186)
17.2.4 使用 FORMAT 准备新盘 …… (187)
17.2.5 使用 FORMAT 重新格式化一张盘
…… (190)
17.3 在 DOS Shell 中 FORMAT …… (193)
17.4 使用 UNFORMAT 恢复一个重新格
式化的磁盘 …… (193)
17.5 自测练习 …… (195)
第 18 章 增加的磁盘管理功能 …… (196)
18.1 总览 …… (196)
18.2 用 DISKCOPY 拷贝磁盘 …… (196)
18.2.1 在 DOS 6.0 或更早的 DOS 中使
用一个驱动器拷贝磁盘 …… (198)
18.2.2 在 DOS 6.2 或更新的 DOS 中使
用一个驱动器拷贝磁盘 …… (199)
18.2.3 用两个驱动器拷贝磁盘 …… (200)
18.2.4 验证拷贝 …… (201)
18.3 在 Shell 中拷贝磁盘 …… (201)
18.4 用 DISKCOMP 比较两个磁盘 …… (202)
18.5 用 VOL 和 LABEL 管理内部标号 ……
…… (203)
18.6 自测练习 …… (205)
第 19 章 DOS Editor …… (206)
19.1 DOS Editor 简介 …… (206)
19.2 启动 DOS Editor …… (207)
19.3 行操作 …… (208)
19.4 编辑文本 …… (208)
19.4.1 光标移动 …… (209)
19.4.2 插入文本 …… (209)
19.4.3 选择文本 …… (210)
19.4.4 删除文本 …… (210)
19.4.5 剪切和粘贴 …… (210)
19.5 搜寻和取代 …… (212)
19.6 无替换搜寻 …… (213)
19.7 文件管理 …… (214)
19.7.1 存储一个新文件 …… (214)
19.7.2 存储一个已经有名字的文件 ……
…… (215)
19.7.3 打开文件 …… (215)
19.8 打印文件 …… (216)
19.9 联机帮助 …… (216)
19.10 修改 Editor 屏幕 …… (218)
19.11 EDIT 命令选项 …… (218)

19.12 自测练习 …… (218)
第 20 章 基本批处理程序 …… (220)
20.1 创建自己的程序 …… (220)
20.2 创建一个批处理文件 …… (220)
20.3 运行批处理程序 …… (221)
20.4 定位批处理文件 …… (221)
20.5 批处理文件的运行 …… (222)
20.6 中止批处理文件 …… (222)
20.7 使用特殊的批处理命令 …… (222)
20.7.1 消除屏幕 …… (223)
20.7.2 程序注释 …… (223)
20.7.3 控制信息显示 …… (224)
20.7.4 暂停程序 …… (225)
20.7.5 调用其他批处理程序 …… (226)
20.8 使用可替换参数 …… (227)
20.9 AUTOEXEC.BAT 文件 …… (229)
20.10 在 DOS Shell 中使用批处理文件 ……
…… (230)
20.11 自测练习 …… (230)
第 21 章 系统配置 …… (231)
21.1 DOS 的启动 …… (231)
21.2 启动文件的保护 …… (231)
21.3 跳过启动文件 …… (233)
21.4 CONFIG.SYS 命令的含义 …… (234)
21.5 加载设备驱动程序 …… (234)
21.6 DEVICE 命令的含义 …… (235)
21.6.1 SETVER 的使用 …… (235)
21.6.2 HIMEM.SYS 的使用 …… (235)
21.6.3 ANSI.SYS 的使用 …… (236)
21.6.4 其他设备驱动程序 …… (236)
21.7 将 DOS 加载到高端内存 …… (236)
21.8 缓冲器的使用 …… (236)
21.9 文件柄的分配 …… (237)
21.10 命令处理器的定义 …… (237)
21.11 堆栈的分配 …… (238)
21.12 自测练习 …… (238)

第四部分 系统管理

第 22 章 数据保护概述 …… (239)
22.1 总览 …… (239)
22.2 病毒的处理 …… (239)
22.3 软件运行失常的处理 …… (240)
22.4 硬件故障的处理 …… (241)
22.5 删除的处理 …… (241)
22.6 重新格式化的处理 …… (241)
22.7 受损磁盘的处理 …… (241)
22.8 自测练习 …… (242)
第 23 章 MSBACKUP 和 MWBACKUP … (243)
23.1 文件备份 …… (243)
23.2 练习准备 …… (244)
23.3 MSBACKUP 或 MWBACKUP 的选择 …
…… (244)
23.4 开始备份 …… (244)
23.5 备份配置 …… (245)
23.5.1 开始检测 …… (246)
23.5.2 确认视频和鼠标配置 …… (246)
23.5.3 软驱变化行检测 …… (247)
23.5.4 软盘兼容性检测 …… (247)
23.5.5 配置问题的处理 …… (249)
23.5.6 重新检测系统 …… (250)
23.6 备份功能的选择 …… (250)
23.7 Setup 文件操作 …… (251)
23.7.1 为 Setup 选择文件 …… (252)
23.7.2 其他选择功能 …… (257)
23.7.3 选择备份类型 …… (258)
23.7.4 标识备份驱动器 …… (259)
23.7.5 备份选项的选择 …… (259)
23.7.6 管理 Setup 文件 …… (262)
23.8 运行备份 …… (263)
23.9 处理备份目录 …… (264)
23.10 保存备份磁盘 …… (265)
23.11 备份文件的管理 …… (266)
23.11.1 Setup 和目录保存在哪里 …… (267)
23.11.2 Setup 文件的打印 …… (268)
23.11.3 删除 Setup …… (268)
23.11.4 删除目录 …… (269)
23.12 通过软盘拷贝大型文件 …… (269)
23.13 自测练习 …… (270)

第 24 章 Restore 和 Compare …… (271)
24.1 恢复和比较 …… (271)
24.2 从 DOS 早期版本制作的备份恢复文件 …… (271)
24.3 恢复文件 …… (272)
24.3.1 打开目录(Catalog) …… (272)
24.3.2 选择文件 …… (273)
24.3.3 Restore 选项 …… (277)
24.3.4 运行 Restore …… (279)
24.3.5 Restore 的一般使用方法 …… (281)
24.4 Compare …… (285)
24.5 从 DOS 6 以前做的备份恢复文件 …… (285)
24.6 自测练习 …… (286)

第 25 章 DOS 的反病毒程序 …… (287)
25.1 为什么要进行病毒防护? …… (287)
25.2 Microsoft 反病毒功能 …… (287)
25.2.1 病毒扫描程序 …… (288)
25.2.2 监视病毒 …… (289)
25.3 何时使用反病毒程序 …… (289)
25.4 MSAV 的使用 …… (290)
25.4.1 选择驱动器 …… (291)
25.4.2 选项设置 …… (291)
25.4.3 运行扫描程序 …… (293)
25.5 病毒处理 …… (294)
25.6 其他 MSAV 命令 …… (294)
25.7 在批处理模式运行 MSAV …… (294)
25.8 MWAV 的使用 …… (296)
25.9 VSAFE 的使用 …… (296)
25.9.1 VSAFE 选项 …… (297)
25.9.2 卸载 VSAFE …… (299)
25.9.3 VSAFE 的加载选项 …… (300)
25.10 更新病毒清单 …… (301)
25.11 自测练习 …… (301)

第 26 章 UNDELETE 的使用 …… (302)
26.1 文件的删除与恢复 …… (302)
26.1.1 Delete Sentry 的使用 …… (303)
26.1.2 检查驱动器清单 …… (304)
26.1.3 列出可恢复文件 …… (305)
26.1.4 用 Delete Sentry 进行文件恢复 …… (306)
26.1.5 删空 SENTRY 目录 …… (307)
26.1.6 SENTRY 中的内容 …… (308)
26.2 UNDELETE 的目标目录 …… (308)
26.3 当 UNDELETE 无法恢复文件时 …… (309)
26.4 删除追踪的使用 …… (309)
26.4.1 列出受 Deletion-Tracking 保护的文件 …… (310)
26.4.2 部分恢复文件 …… (311)
26.4.3 恢复在 DOS 5 下删除的文件 …… (312)
26.5 DOS 目录方法的使用 …… (312)
26.5.1 被删目录 …… (313)
26.5.2 目录方法的缺陷 …… (314)
26.6 无提示恢复文件 …… (314)
26.7 在 DOS Shell 中恢复文件 …… (314)
26.8 Windows 的 Undelete 操作 …… (314)
26.8.1 Windows Undelete 的特点 …… (315)
26.8.2 目录的恢复 …… (315)
26.9 UNDELETE.INI 文件的设定 …… (316)
26.10 自测练习 …… (318)

第 27 章 CHKDSK 和 SCANDISK …… (319)
27.1 CHKDSK 和 SCANDISK 的用途 …… (319)
27.2 使用 CHKDSK 或 SCANDISK 的时机 …… (319)
27.3 检查驱动器问题 …… (320)
27.3.1 磁盘问题的处理 …… (321)
27.3.2 使用 CHKDSK 处理簇丢失 …… (322)
27.3.3 使用 SCANDISK 处理簇丢失 …… (323)
27.3.4 交缠文件 …… (323)
27.4 过时的 CHKDSK 功能 …… (324)
27.5 SCANDISK 的表面扫描 …… (325)
27.6 自测练习 …… (325)

第 28 章 系统优化 …… (326)
28.1 DOS 的功能 …… (326)
28.1.1 更有效的内存管理 …… (326)
28.1.2 去除文件碎片 …… (327)
28.1.3 数据压缩 …… (327)

第 29 章 内存管理 …… (329)
29.1 DOS 的内存之争 …… (329)
29.2 内存的概念 …… (329)

29.2.1 内存种类 …… (330)
29.2.2 扩充内存 …… (331)
29.2.3 扩展内存 …… (331)
29.2.4 高端内存区 …… (332)
29.3 浏览内存布局 …… (332)
29.3.1 加载 HIMEM …… (333)
29.3.2 在 HMA 中加载 DOS …… (334)
29.4 自测练习 …… (335)

第 30 章 为 386、486 和奔腾机增加的内存工具 …… (336)
30.1 DOS 和 386、486、奔腾机 …… (336)
30.2 提供扩充内存 …… (336)
30.2.1 使用高端内存块 …… (337)
30.3 MEMMAKER 的使用 …… (338)
30.4 MEMMAKER 所做的改动 …… (340)
30.4.1 加载 HIMEM 和 EMM386 … (340)
30.4.2 DOS 命令 …… (341)
30.4.3 DEVICEHIGH 命令 …… (341)
30.5 AUTOEXEC.BAT 文件 …… (341)
30.5.1 取消 MEMMAKER …… (342)
30.6 自测练习 …… (342)

第 31 章 驱动器缓存 …… (343)
31.1 有关缓存 …… (343)
31.2 SmartDrive 的用途 …… (343)
31.3 缓存和缓冲区 …… (344)
31.4 启动 SmartDrive …… (345)
31.4.1 SmartDrive 的双缓冲(double-buffering)功能 …… (345)
31.4.2 选择缓存方式 …… (346)
31.4.3 安装 SmartDrive 后改变驱动器缓存 …… (346)
31.4.4 SmartDrive 结果显示 …… (347)
31.4.5 迫使 SmartDrive 写入缓存数据 …… (347)
31.4.6 Windows 和 SmartDrive …… (348)
31.5 控制缓存大小 …… (348)
31.5.1 调整 SMARTDRV …… (348)
31.6 用 SMARTMON 衡量 SmartDrive 的性能 …… (349)
31.6.1 工作记录 …… (350)
31.6.2 SmartMon 选项 …… (351)
31.7 磁盘缓冲区 …… (351)
31.8 自测练习 …… (352)

第 32 章 RAM 驱动器 …… (353)
32.1 有关 RAM 驱动器 …… (353)
32.2 何处放置 RAM 驱动器 …… (353)
32.3 RAM 驱动器的设置 …… (354)
32.4 RAM 驱动器的使用 …… (355)
32.5 将临时文件引向 RAM 驱动器 …… (355)
32.6 覆盖缺省参数 …… (356)
32.7 自测练习 …… (356)

第 33 章 去除文件碎片 …… (357)
33.1 有关文件碎片化 …… (357)
33.2 运行 DEFRAG 前的准备工作 …… (358)
33.3 启动 DEFRAG …… (360)
33.4 Drive Map 诠释 …… (361)
33.5 优化进程 …… (362)
33.6 Optimize 菜单 …… (363)
33.7 目录项的分类 …… (364)
33.8 其他 DEFRAG 命令选项 …… (365)
33.9 以批处理模式运行 DEFRAG …… (366)
33.10 自测练习 …… (366)

第 34 章 DriveSpace 和 DoubleSpace …… (367)
34.1 DOS 6 和数据压缩 …… (367)
34.2 数据压缩的工作方式 …… (367)
34.3 压缩驱动器 …… (368)
34.4 已存在驱动器的压缩 …… (368)
34.5 设置 DriveSpace 或 DoubleSpace …… (369)
34.6 DOS 数据压缩初步 …… (369)
34.6.1 压缩过程 …… (371)
34.6.2 压缩其他驱动器 …… (371)
34.6.3 选择一驱动器 …… (372)
34.7 与其他 DOS 应用程序的关系 …… (373)
34.8 从 DoubleSpace 转换到 DriveSpace …… (374)
34.9 压缩驱动器的管理 …… (374)
34.10 DIR 和压缩驱动器 …… (375)
34.11 DriveSpace 和 DoubleSpace 的附加功能 …… (375)
34.12 自测练习 …… (376)

第 35 章 系统优化综述 …… (377)
35.1 检查 CONFIG.SYS 和 AUTOEXEC.BAT …… (377)

35.2 286 系统的设置 …… (377)
35.3 386 及其以上系统的设置 …… (378)
35.4 每日关机程序 …… (380)
35.5 每周关机程序 …… (380)
35.6 后续章节简介 …… (381)
35.7 自测练习 …… (381)

第五部分 DOS 高级应用

第 36 章 命令行状态下的 DOS 应用 …… (382)
36.1 最大限度地发挥 DOS 命令行及 DOS Shell 的作用 …… (382)
36.2 输入输出重定向 …… (382)
36.2.1 用 MORE 作分页显示 …… (383)
36.2.2 临时管道文件 …… (384)
36.2.3 向文本文件中附加一些内容 …… (384)
36.2.4 输入输出重定向的机制 …… (385)
36.2.5 标准错误输出 …… (385)
36.2.6 将输出重定向到 PostScript 打印机 …… (386)
36.2.7 用 ECHO 简化命令的运行 … (386)
36.3 DOS 的文件过滤功能 …… (387)
36.3.1 行的概念 …… (387)
36.3.2 用 SORT 命令对行排序 …… (387)
36.3.3 用 FIND 命令定位文本行 …… (390)
36.4 关于 DOS Shell …… (392)
36.4.1 定义程序项 …… (392)
36.4.2 关于 DOSSHELL.INI …… (395)
36.5 自测练习 …… (398)
第 37 章 批处理程序中的编程逻辑 …… (399)
37.1 逻辑编程引论 …… (399)
37.2 批处理文件的链接 …… (399)
37.2.1 链接和调用 …… (400)
37.2.2 链接 …… (400)
37.2.3 调用 …… (401)
37.2.4 传递可替换的命令行参数 …… (401)
37.2.5 多重链接 …… (402)
37.2.6 递归调用 …… (402)
37.2.7 中断有链接的批处理程序 …… (402)
37.3 BAT 程序中的分枝结构 …… (403)
37.3.1 分枝的种类 …… (403)
37.4 IF 命令 …… (404)
37.4.1 程序返回码 …… (404)
37.4.2 ERRORLEVEL 条件 …… (405)
37.4.3 测试文件是否存在 …… (408)
37.4.4 字符串比较 …… (409)
37.4.5 空字符串的比较 …… (409)
37.4.6 嵌套 IF 语句 …… (410)
37.5 循环语句 …… (411)
37.5.1 循环语句的示例 …… (411)
37.5.2 调用 SHIFT 命令的循环 …… (412)
37.5.3 FOR 循环 …… (413)
37.6 CHOICE 命令 …… (414)
37.6.1 CHOICE 命令的参数 …… (416)
37.7 用 DOSKEY 宏指令替换 DOS 命令 …… (416)
37.8 软盘上的批处理程序 …… (417)
37.9 批处理程序可能被破坏 …… (417)
37.10 自测练习 …… (418)
第 38 章 启动 DOS …… (419)
38.1 控制启动流程 …… (419)
38.2 跳过启动文件 …… (419)
38.2.1 带确认执行 CONFIG.SYS 命令 …… (421)
38.3 使用系统设置语句块 …… (422)
38.3.1 系统设置语句块 …… (423)
38.3.2 定义菜单 …… (423)
38.3.3 子菜单 …… (424)
38.3.4 在菜单中加入颜色 …… (425)
38.4 自测练习 …… (426)
第 39 章 目录管理 …… (427)
39.1 高级目录管理技术 …… (427)
39.2 目录结构的建议 …… (427)
39.3 用 DELTREE 删除文件和子目录 …… (428)
39.3.1 清除一个目录中的所有文件而保留这个目录 …… (428)
39.3.2 删除文件但保留目录 …… (428)

39.4 重新组织一棵目录树 …… (429)
39.4.1 为什么使用 APPEND 是危险的 …… (429)
39.4.2 使一个目录看起来像一个驱动器 …… (430)
39.4.3 用 SUBST 创建一个虚拟驱动器 …… (431)
39.4.4 为 SUBST 选择驱动器名 …… (431)
39.4.5 TRUENAME 命令 …… (432)
39.5 自测练习 …… (433)
第 40 章 高级拷贝技术 …… (434)
40.1 处理一些特殊拷贝任务 …… (434)
40.2 XCOPY 命令 …… (434)
40.2.1 XCOPY 和 DISKCOPY …… (435)
40.2.2 避免磁盘空间不足时命令中断 …… (435)
40.2.3 剪裁和移动 …… (437)
40.3 合并文件 …… (438)
40.3.1 附加文件 …… (439)
40.3.2 各种不同类型文件的合并 …… (440)
40.3.3 更新日期/时间 …… (441)
40.3.4 匹配合并 …… (441)
40.3.5 /A 和/B 选项的基本操作 …… (442)
40.4 REPLACE 命令 …… (444)
40.4.1 追加文件 …… (445)
40.4.2 替换已经存在的文件 …… (445)
40.5 自测练习 …… (445)
第 41 章 INTERLNK 命令 …… (447)
41.1 使用 INTERLNK 命令连接两台计算机 …… (447)
41.2 INTERLNK 建立过程 …… (448)
41.3 用缆线连接两台计算机 …… (448)
41.4 建立服务器 …… (448)
41.5 建立客户机 …… (450)
41.5.1 标识连接的端口 …… (452)
41.5.2 重定向驱动器 …… (453)
41.6 用 INTERLNK 建立两台计算机之间的连接 …… (453)
41.7 中止连接 …… (455)
41.8 拷贝 INTERLNK 软件 …… (455)
41.9 自测练习 …… (455)
第 42 章 在 DOS 上的 Windows …… (456)
42.1 DOS 和 Windows 相互支持 …… (456)
42.2 优化 Windows …… (456)
42.2.1 释放内存 …… (456)
42.2.2 释放磁盘空间 …… (457)
42.2.3 使用 SmartDrive …… (458)
42.3 在 Windows 下使用 DOS 应用程序 …… (458)
42.3.1 启动 DOS 应用程序 …… (459)
42.3.2 在 Windows 下执行 DOS 任务 …… (461)
42.3.3 程序信息文件(PIF) …… (462)
42.3.4 PIF 编辑器 …… (463)
42.3.5 实模式和标准模式下的 PIF 选项 …… (470)
42.4 自测练习 …… (473)
第 43 章 DOS 及其运行环境 …… (474)
43.1 DOS 的运行环境 …… (474)
43.2 环境变量 …… (474)
43.2.1 显示环境变量 …… (474)
43.2.2 DOS 的环境变量 …… (475)
43.2.3 建立环境变量 …… (476)
43.2.4 改变和删除环境变量 …… (477)
43.2.5 访问环境变量 …… (477)
43.3 控制命令解释程序和环境 …… (479)
43.3.1 环境大小 …… (480)
43.3.2 存放在其他地方的命令解释程序 …… (481)
43.3.3 使用第三方的命令解释程序 …… (481)
43.4 装入第二级命令解释程序 …… (481)
43.4.1 指定 COMSPEC …… (482)
43.4.2 忽略 COMSPEC …… (483)
43.4.3 动态环境分配 …… (483)
43.4.4 在其他命令解释程序为第一级命令解释程序时启用 COMMAND.COM …… (483)
43.5 装入新的第一级命令解释程序 …… (484)
43.6 用 COMMAND.COM 执行一个命令 …… (484)
43.7 用 COMMAND.COM 检验一个批处理程序 …… (485)
43.8 自测练习 …… (485)

第 44 章 腾出更多的内存空间 …… (487)
44.1 定制高端内存 …… (487)
44.2 使用 HIMEM.SYS 出现的问题 … (487)
44.2.1 装入正确的 HMA 驱动程序 ……
…… (488)
44.2.2 控制时钟速度 …… (489)
44.2.3 在 EISA 总线的计算机中，利用 16MB 以外的内存 …… (489)
44.2.4 控制直接访问扩展内存地址的老程序 …… (489)
44.2.5 确定 DOS 装入了 HMA …… (489)
44.2.6 影象 ROM-BIOS …… (489)
44.3 在高端内存中找到额外的空间…… (490)
44.3.1 强制性扫描(Aggressive Scan) ……
…… (490)
44.3.2 使用单显内存区(Use Monochrome Region) …… (491)
44.3.3 排除高端内存区的访问…… (491)
44.3.4 打开及关掉扩充内存…… (492)
44.4 使用 UMB 的问题 …… (493)
44.5 操作 TSR(驻留内存的程序) …… (493)

第 45 章 再谈 DrvSpace 和 DblSpace …… (495)
45.1 DrvSpace 和 DblSpace 基本功能之外的特色 …… (495)
45.2 DrvSpace 和 DblSpace 的核心程序 ……
…… (495)
45.3 DRVSPACE 或 DBLSAPCE 命令 ……
…… (496)
45.4 压缩其他的硬盘 …… (496)
45.5 压缩软盘 …… (498)
45.5.1 在宿主驱动器上的文件…… (500)
45.6 维护压缩驱动器及它们的宿主驱动器 …
…… (501)
45.6.1 为压缩驱动器安装高速缓存系统 …
…… (501)
45.6.2 查看压缩驱动器的信息…… (501)
45.6.3 重新设置驱动器的大小…… (502)
45.6.4 重整压缩驱动器…… (503)
45.6.5 改变压缩比 …… (503)
45.6.6 格式化压缩驱动器…… (505)
45.6.7 扫描驱动器上的错误…… (505)
45.7 释放内存 …… (505)
45.8 删除压缩驱动器…… (506)
45.9 解除驱动器的压缩方式…… (506)
45.10 自测练习 …… (507)

第 46 章 数据操作的其他技巧 …… (508)
46.1 基本技能以外的知识…… (508)
46.2 格式化非缺省容量的磁盘…… (508)
46.3 为旧的 DOS 版本格式化软盘 …… (510)
46.4 扩展 FC 的重新同步缓冲区 …… (510)
46.5 比较两个文件的功能…… (513)
46.6 了解 DOS 的验证功能 …… (513)
46.6.1 CRC(循环冗余校验)…… (514)
46.6.2 DOS 的验证功能 …… (514)
46.7 自测练习 …… (514)

第 47 章 高级补救技术 …… (515)
47.1 DOS 的恢复工具 …… (515)
47.2 用 SCANDISK 和 CHKDISK 处理结构错误 …… (515)
47.2.1 无效分配单元…… (515)
47.2.2 SCANDISK 和 CHKDSK 怎样修复无效分配单元…… (517)
47.2.3 分配错误…… (517)
47.2.4 无效子目录项…… (518)
47.2.5 磁盘控制区中的坏扇区…… (520)
47.3 在没有映象信息的情况下重建磁盘 ……
…… (521)
47.3.1 UNFORMAT 命令怎样重建磁盘 …
…… (522)
47.3.2 启动重建过程…… (522)
47.4 使用 MSD …… (523)
47.5 自测练习 …… (526)

第 48 章 控制台的控制 …… (527)
48.1 包装 DOS …… (527)
48.2 DOS 和视频监视器 …… (527)
48.2.1 CON 驱动程序 …… (527)
48.2.2 视频缓冲区…… (528)
48.2.3 ANSI.SYS 所提供的控制 …… (528)
48.3 ANSI.SYS 设备驱动程序 …… (529)
48.4 给 ANSI.SYS 发送命令 …… (529)
48.4.1 键入转义代码序列…… (530)
48.4.2 使用 PROMPT 命令 …… (530)
48.4.3 使用文本文件…… (531)
48.4.4 使用批处理文件…… (532)

48.5 用于屏幕颜色控制的转义 ANSI.SYS 代码 …… (532)
48.6 屏幕和光标控制 …… (533)
48.6.1 创建一个标题 …… (534)
48.6.2 显示一个信息框 …… (535)
48.6.3 字符回绕 …… (537)
48.6.4 屏幕的显示模式(图形) …… (537)
48.7 ANSI 对于键盘的控制 …… (538)
48.7.1 重新映象一些的键 …… (539)
48.7.2 为键指定字符串 …… (540)
48.8 控制 CON(控制台)模式 …… (541)
48.8.1 指定文本显示的行数和列数 …… (541)
48.8.2 控制键盘的响应 …… (541)
48.9 在标准控制台设备间切换 …… (542)
48.10 节省膝上型计算机电池的电能 … (543)
48.10.1 POWER 命令 …… (543)
48.11 自测练习 …… (544)

第 49 章 处理另一个国家的信息 …… (545)
49.1 用其他国家的语言通信 …… (545)
49.2 总览 …… (545)
49.3 国际格式 …… (546)
49.4 打印国际字符 …… (548)
49.5 使用国际键盘布局 …… (549)
49.5.1 KEYB 命令 …… (549)
49.6 使用代码页 …… (551)
49.6.1 怎样装入代码页 …… (552)
49.6.2 监视器代码页 …… (552)
49.6.3 装入软件代码页 …… (553)
49.6.4 切换代码页 …… (554)
49.6.5 一个完整的代码页装载实例 …… (554)
49.6.6 本国语言支持(National Language Support) …… (555)
49.6.7 使用 NLSFUNC 和 CHCP 的另一个代码页装载实例 …… (556)
49.6.8 熟练使用代码页 …… (556)
49.6.9 显示代码页信息 …… (557)
49.7 自测练习 …… (557)

第 50 章 DOS 的打印功能 …… (558)
50.1 DOS 最基本的打印服务 …… (558)
50.2 打印机类型 …… (558)
50.2.1 回显打印功能 …… (559)
50.2.2 打印文本屏幕 …… (559)
50.3 配置打印机 …… (560)
50.4 使用串行打印机 …… (561)
50.4.1 配置串行口 …… (561)
50.4.2 重定向串行打印机 …… (562)
50.4.3 查看重定向状态 …… (563)
50.5 PRINT 命令 …… (563)
50.5.1 配置 PRINT …… (564)
50.5.2 控制打印队列 …… (565)
50.6 打印图形屏幕 …… (566)
50.6.1 GRAPHICS 是如何工作的 … (566)
50.6.2 GRAPHICS 命令 …… (567)
50.6.3 GRAPHICS 选项 …… (567)
50.6.4 长宽比 …… (568)
50.7 自测练习 …… (569)

第 51 章 配置磁盘和网络 …… (570)
51.1 为 DOS 定义驱动器 …… (570)
51.2 配置物理磁盘 …… (570)
51.2.1 change-line(换线)支持 …… (572)
51.3 配置逻辑软盘 …… (572)
51.3.1 建立一个替代驱动器名字 …… (573)
51.3.2 建立两个替代驱动器名 …… (574)
51.3.3 增加一个外部软盘驱动器 …… (574)
51.3.4 选择使用 DRIVPARM 或 DRIVER.SYS …… (575)
51.4 在网络上运行 DOS …… (575)
51.4.1 把网络操作系统升级为 DOS 6.22 …… (576)
51.4.2 DOS 命令的局限 …… (576)
51.4.3 共享和锁定文件 …… (577)
51.5 自测练习 …… (577)

第 52 章 多媒体 PC 机 …… (578)
52.1 多媒体的新概念 …… (578)
52.2 多媒体硬件 …… (578)
52.3 软件 …… (579)
52.3.1 MSCDEX 程序 …… (580)
52.3.2 选择 IRQ(中断) …… (581)
52.3.3 优化 CONFIG.SYS 和 AUTOEXEC.BAT …… (582)

第 53 章 使用 QBASIC …… (583)
53.1 QBASIC 的特色和功能 …… (583)

53.2 QBASIC 界面 …… (584)
53.2.1 QBASIC 编辑器 …… (584)
53.2.2 改进编辑器效果 …… (585)
53.3 将 GW-BASIC 或 BASICA 程序转换成 QBASIC 格式 …… (587)
53.3.1 运行程序 …… (587)
53.3.2 行号 …… (588)
53.4 新的常数和变量类型 …… (588)
53.4.1 常数 …… (588)
53.4.2 数字变量 …… (588)
53.4.3 字符串变量 …… (589)
53.4.4 其他字符串函数 …… (589)
53.4.5 数组变量 …… (589)
53.4.6 记录数据类型 …… (590)
53.5 新的程序流程控制功能 …… (591)
53.5.1 扩展的 IF 语句 …… (591)
53.5.2 使用 Case 结构 …… (592)
53.5.3 循环结构 …… (593)
53.5.4 循环的内容 …… (594)
53.6 模块化程序设计 …… (594)
53.6.1 检查模块程序 …… (594)
53.6.2 建立过程 …… (594)
53.6.3 全局变量 …… (595)
53.6.4 子程序 …… (596)
53.6.5 变量传值 …… (597)
53.6.6 函数 …… (598)
53.6.7 建立函数 …… (598)
53.7 调试工具 …… (599)
53.7.1 Immediate 窗口 …… (599)
53.7.2 跟踪程序 …… (599)
53.7.3 断点 …… (599)
53.8 在 DOS 提示符下直接运行 QBASIC 程序 …… (600)
53.9 自测练习 …… (600)

第 54 章 FDISK …… (601)
54.1 用 FDISK 给硬盘分区 …… (601)
54.2 启动 FDISK …… (601)
54.3 建立 DOS 分析 …… (602)
54.3.1 建立主 DOS 分区 …… (603)
54.3.2 建立扩展 DOS 分区 …… (603)
54.4 设置活动分区 …… (603)
54.5 删除一个分区 …… (604)
54.6 查看分区信息 …… (604)
54.7 退出 FDISK …… (606)
54.8 自测练习 …… (606)

第 55 章 由 DOS 跨上 Internet 网 …… (607)
55.1 Internet 是什么? …… (607)
55.2 Internet 资源 …… (607)
55.2.1 数据库 …… (608)
55.2.2 新闻论坛(Newsgroups) …… (608)
55.2.3 电子邮件(E-mail) …… (608)
55.2.4 邮件论坛(Mailing Lists) …… (608)
55.2.5 网络闲谈(Chatting) …… (609)
55.2.6 其他服务 …… (609)
55.3 访问 Internet 网 …… (609)
55.3.1 有关调制解调器 …… (609)
55.3.2 远程通信软件 …… (610)
55.3.3 连通 Internet 的网关 …… (611)
55.4 Internet 的一些警告 …… (612)
55.5 怎样从 DOS 访问 Internet? …… (612)

第六部分 命令用法参考

命令用法参考 …… (613)
附录 A 设备驱动程序 …… (725)
附录 B 返回码 …… (738)
附录 C IBM 扩展字符集 …… (740)

第一部分　DOS 运行环境概述

第 1 章　PC 系统概述

简介： PC 系统不仅指硬件；它还包括使系统正常工作的软件、运行时的数据、存储信息的介质，还有最重要的就是："你"。

1.1　硬　件

硬件指系统的物理组成，至少包括：系统主机(单元)、显示器和键盘，可能还有一个鼠标和打印机，以及一些其他的部件。

1.1.1　系统主机

系统主机包含着计算机的"心脏"。在主机中有微处理器芯片，磁盘驱动器和电源。PC 被制造商制成可被打开的，这样可自由地除去机箱。但是，注意不要触摸和取下不了解的部件。

警告： 在打开系统主机之前要先关闭电源。

为了打开系统主机，要先把主机后板周围的螺丝卸掉，然后去掉机箱。图 1.1 显示了一个典型系统主机的内部。

提示： 主机打开后，吸掉或吹掉灰尘。

打开机箱后，可看到系统板(或母板)、电源、一些扩充板、磁盘驱动器和电缆组。系统板是一个插了许多元件的大型电路卡。它拥有微处理器模块、内存，可能还有其他一些芯片以及连接它们的电路。

1.1.1.1　微处理器

微处理器是计算机的大脑，它是由电子电路微型集成的一个单一的硅芯片。微处理器可运行字处理程序中的命令、CAD/CAM 系统、纸牌游戏等其他运行程序。微处理器在系统板的中间位置很容易看到。图 1.2 显示了一芯片模块。

286、386 等指系统包含的微处理器类型。最初的 IBM PC、XT 机都使用 Intel 公司的 8088 微处理器。AT 使用 80286 芯片，就是人们熟知的 286 机。更新的机器包含 386 或 486 芯片，还有一种是为 PC 设计的 Intel 芯片家族中的最新成员——Pentium 芯片。每个新的微处理器的容量和速度大约都是它的前一代芯片的两倍，Pentium 是个例外，一些 486 芯片可与 Pentium 的速度匹敌。后缀如 386SX 中的 SX 表示芯片基本结构的不同。

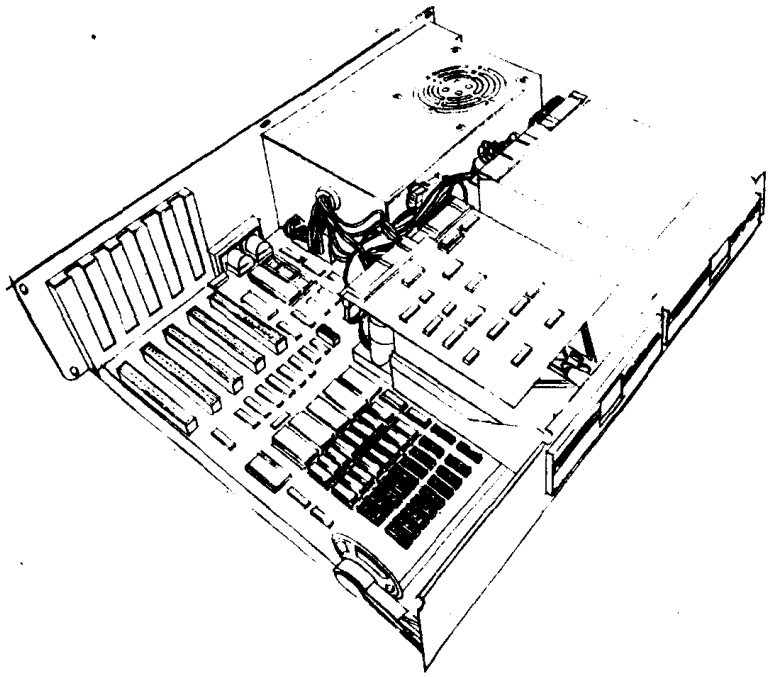

图 1.1 系统主机的内部

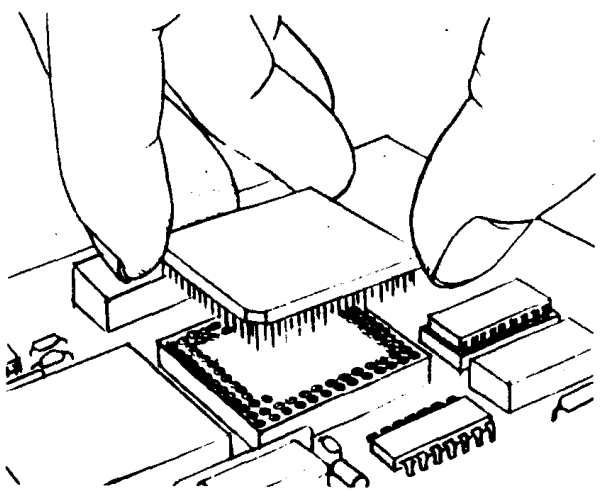

图 1.2 微处理器模块

1.1.1.2 协处理器

系统可能包括一个数字协处理器。这个协处理器比主处理器更能处理数字运算，可大大提高一些主要处理数学运算程序的运行速度。令人惊奇的是，电子表格软件和计算器并不是唯一的受益者。控制页边空白、管理字体、在图象或数据中定位光标、图形的放大和缩小等等都需要大量数学运算。当你安装了一个协处理器后，像桌面印刷和图形设计这样的程序可明显提高速度。新芯片的一些版本如 486，在处理器中都包含了一个协处理器(但 486SX 不包括数学协处理器)。

1.1.1.3　内存

内存用来保存当前使用的程序和数据。它提供了与微处理器紧密联系的短期、高速存储空间。但是，大部分内存需要持续的电力支持以保持数据，当计算机电源断开时，内存会丢失其中的数据。

许多人发现内存是计算机系统中最易混淆的组成部分——从理解和控制两方面来说都是这样。第 30 章“为 386、486 和奔腾机增加的内存工具”和第 31 章“驱动器缓存”，详细解释了 RAM、ROM、扩展内存(extended memory)、扩充内存(expanded memory)等概念。这两章还讲述了如何获得 DOS 系统的最大内存。

1.1.1.4　总线

在计算机词汇中，总线是指将数据从一个部件传送到另一个部件的电路。例如，总线在内存和微处理器之间传送数据。即使最简单的运算如两数相加，也要涉及几次数据传送，因此计算机总线的大小和速度与计算机的总体速度有很大关系。

例如，一个 386 有着 32 位的总线结构，这意味着它的基本数据存储单元——称为一个字(word)，可容纳 32 个数位。一个完全的 386 机器(386DX)包含 32 位总线，这样一次可传送一个整字。一个 386SX 内部使用 16 位总线来节约费用，因此要花两次才能传送一个完整的字。因此，386SX 比 386DX 便宜，但速度较慢。

1.1.1.5　CMOS 和它的电池

CMOS(Complementary Metal-Oxide Semiconductor，互补金属氧化物半导体)最先出现在 286 机器中，用来保存关于计算机硬件配置的信息，这些信息在系统启动过程中是至关重要的。如果系统不知道它有何种类型的硬件，它将不能读磁盘，从而不能加载 DOS。

因为 CMOS 必须在计算机电源关闭时保持它的信息，一个小电池就用作它的持续电源。大部分新一代计算机有一个锂电池，它的寿命一般为几年。当电池耗尽后，启动机器时就会看到 CMOS 错误消息。这时可以通过运行硬件的 SETUP 程序和手动设置驱动器类型、日期和时间来启动系统。硬件手册将告诉你如何使用 SETUP 程序；如果还不行，可以打电话找修理员或厂商。更换电池时，技术人员可能会用常规电池代替锂电池。常规电池使用的时间不会像锂电池那么长，但自己可以更换。CMOS 电池还支持时钟/日历芯片，以便机器关闭时日期和时间能继续变化。

> **提示：** 现在是为 CMOS 电池错误作准备的良机。硬驱由号码来确认。当电池失效时，如果你不知道这个号码，就不能完成 SETUP 程序和机器的启动。现在，写下你的号码，把它贴在系统主机上。如果硬件手册没有告诉你这个号码，DOS 6 中包括的 MSD 诊断程序可告诉你号码是什么。启动 MSD 并查询硬盘驱动器一节以获得消息“CMOS type nnn”。(MSD 在第 47 章“高级补救技术”中解释。)

1.1.1.6　跳线和 Dip 开关(JDS，Jumpers and Dip Switch)

JDS 并不是用来迷惑 PC 用户的(如图 1.3)。它们使硬件有更多的灵活性，当改变系统的基本设置时可重新配置它。例如，如果要增加更多的内存，可能必须重新设置一个 dip(双列直插式)开关来告诉系统现有多少内存。如果从一个锂电池改变到一个电池组，可能必须进行跳线以将电路导向新的电源。在了解相关内容之前，不要改变 JDS 设置。硬件手册可能

会指导你该怎么做。如果没有，可向有关技术人员求助。

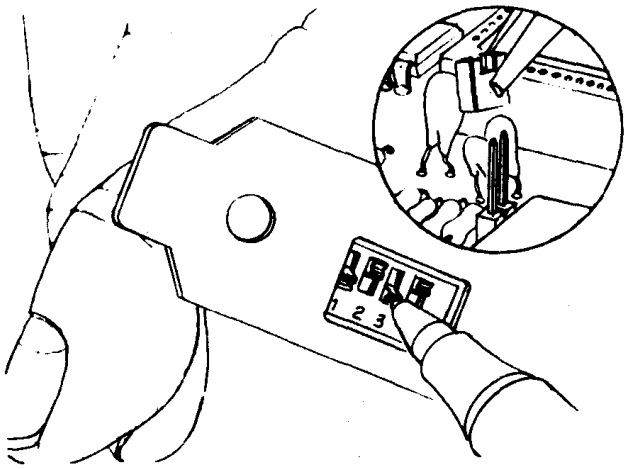

图 1.3 跳线和 dip 开关

注释： 有“插入即用”(Plug and Play)特点的系统不需要 JDS，一旦安装上一个新设备，就可开始使用它。

提示： 打印机常常有几个 dip 开关，它用来选择缺省字体、垂直空格和模拟标准打印机。

1.1.2 键盘

键盘工作时像一台打字机，但有许多重要的区别。另外除了正常的打字区域外，它还有功能键，一个 Keypad，和一些特殊用途键。

1.1.2.1 功能键

功能键标着 F1，F2 等等，位于键盘的打字区域的左侧或上方。可以有 10 个，12 个或更多的功能键，每个程序使用它需要的任何功能键。在本书中，你将学会在 DOS 中它们的功能，值得注意的是它们在 Windows、Word Perfect、Lotus 1-2-3 和其他程序中有着不同的功能。

提示： 在许多程序中，F1 用来在屏幕上获得帮助信息。

1.1.2.2 Keypad

Keypad 有两个功能。当 NumLock 处于打开状态时，它工作时像一个 10 键加法器。数字出现在键面的上部。当 NumLock 关闭时，Keypad 变成了一个光标移动键盘(光标是显示屏上的指示器，显示下一个字符打出的位置)。光标移动记号出现在键面的下面；只有这些数字键有移动光标功能，但“5”键没有这个功能。这些键具体做些什么，依赖于当前运行的程序，通常，上移键将光标移向屏幕的顶部，左移键把光标移向左边，等等。在 Keypad 上的灰色键只有一个功能，无论 NumLock 是打开还是关闭，它都能正常工作。

试着使用 NumLock 打开和关闭时的 Keypad：

1. 在键盘上有一个 NumLock。如果灯灭，按一下 NumLock 可使它变亮。当 NumLock 打

开时灯亮。

2. 按键1,2,4,/和*。

3. 你敲击的字符应在显示器上出现。

4. 按一下NumLock使灯灭,NumLock关闭时灯灭。

5. 按一下4键。注意它不再打印4,而是向左移动光标,重新定位在字符下面。
当NumLock关闭时,数字键具有移动光标的功能。

6. 按一下*键。注意它依旧打印星号。
这个键在NumLock打开或关闭时具有相同的功能。

1.1.2.3 额外的光标键

大部分键盘有101或102个键(最初键盘只有84个键)。新型键盘,即增强型键盘,有一组额外的光标移动键,这样可使Keypad处于数字状态。至于这些键的意义以及是否与Keypad键的功能重复取决于具体程序。

1.1.2.4 锁键

键盘有三个锁键:NumLock,ScrollLock和CapsLock。正如已看到的,NumLock将Keypad锁成数字或光标移动功能。大部分软件中,ScrollLock没有什么作用,许多电子表格程序会用到它。CapsLock将字母键锁成大写。与常规打字机的ShiftLock不同,CapsLock不影响数字和其他符号键。

在键盘上试一试CapsLock键:

1. 在键盘上找到CapsLock显示灯。如果它关闭,按下CapsLock键将它打开。CapsLock打开时灯亮。

2. 击A,B和C键。
注意这些键产生大写字母。

3. 击键1,2,3。再键入一个逗号,句号和一个斜杠(/)。
注意这些键即使在CapsLock打开时也出现同样的字符。

4. 按住Shift再键入!,@和#,还有<,>和?。
可见,需要用Shift键键入这些键上面的字符。

5. 按一下CapsLock使灯灭。
CapsLock关闭时灯灭。

6. 键入a,b,c,1,2,3,和一个逗号,句号和斜杠。
CapsLock关闭时,字母以小写出现,数字及符号不受CapsLock的影响。

一个锁键是一个开关键,按一下打开,再按一下关闭。

1.1.2.5 转档键

键盘有三个转档键——Shift,Ctrl和Alt,它们不产生自己的字符,但它们能改变其他键的功能。例如,在一个字处理程序中,没改变的G键可以打印字符g。如果按Shift和G,写作Shift+G,将产生一个大写字母G,Ctrl+G可能会产生其他的功能,还有像Alt+G,Shift+Ctrl+G,Shift+Alt+G,Ctrl+Alt+G和Shift+Ctrl+Alt+G等等组合。在DOS中,Ctrl键用得很多,如Ctrl+G或Ctrl+Alt+G。使用转档键的不同组合,程序可将八种不同的功能

分配给一个键。

Shift 键通常改变锁键功能。对一个字处理程序如果 CapsLock 打开，没改变的 G 键可能会打印大写 G，而 Shift+G 打印小写 g。对 Keypad 也是如此，如果处在数字状态，Shift 会产生光标移动功能。

试一试 Shift 键与 CapsLock 和 NumLock 的组合：

1. 按一下 CapsLock 使之打开。
2. 打印 G 键，则出现大写字母 G。
3. 现在按 Shift+G。
   则出现小写字母 g。
4. 按一下 NumLock 使之打开。
5. 在数字 Keypad 上打印 4。
   则出现数字 4。
6. 现在在数字 Keypad 上按 Shift+4。
   光标向左移动一个字符。

提示： 在多种程序中，Shift+Tab 与 Tab 相反，即 Shift+Tab 使光标后退。

1.1.2.6 特殊用途键

键盘为系统提供了 9 个特殊用途键。Pause 键通常中断当前程序，直到再按任何其他键，这时程序从中断处继续运行。此键和 Ctrl 一起使用，将产生一个中断信号，使当前程序退出运行（但在有些程序中不能如此）。PrintScreen 键（在你的键盘中可能是 Shift+PrintScreen）通常打印屏幕上的内容，但屏幕处于图形模式时会出现麻烦。（第 50 章“DOS 的打印功能”，描述了如何才能打印图形屏幕）。Esc 键通常取消发生的任何事情，但并不杀死程序。

注释： Ctrl+S 的功能与 Pause 相同，Ctrl+C 与 Ctrl+Break 功能相同。对于一个有经验的打字员来说，会更喜欢这些键，因为无需把手移开就可使用它们。

试试 Pause 和 Esc 键：

1. 已在屏幕上键入了几个字符。按下 Esc 告诉 DOS 忽略它们，DOS 显示一个反斜杠（\）并且光标移到下一行。
2. 在进行下一步操作前，找到 Pause 键并准备按下它。
3. 在屏幕上显示一些数据然后使之暂停。键入下列命令显示一个文件的内容。只要数据一开始出现在屏幕上就按 Pause 键。

   ```
   TYPE C:\DOS\README.TXT
   ```

   当你按 Pause 时，数据就会停止在屏幕上滚动。
4. 按任何键释放暂停，按 Pause 键再次暂停。
5. 找到 Ctrl 键，按下它，然后击 Break 键两次（Break 键也是 Pause 键）。
   第一次按 Ctrl+Break 时，只是释放了暂停。第二次则杀死了程序。DOS 重新显示命令提示符。

1.1.2.7 Typematic特性

当按住字母和光标移动键时它们将自动重复。例如，可按住*键得到一行星号。某些程序，如Windows和DOS，可以调节一个键按住多长时间后开始重复，以及重复的速度。第48章"控制台的控制"描述了如何调节DOS的typematic速度。

在你的键盘上试一试typematic特性：

1. 按住T键得到一串T。
2. 按住Backspace键删除一串T。

1.1.3 鼠标

键盘可对所有的程序起作用，但诸如Windows之类的图形程序还可使用鼠标。鼠标可直观地完成各种功能，而不是使用词句命令。只需指向你想做的事后再单击一下按钮。有时可抓住一个对象在屏幕上拖动它。使用鼠标常常能更快更容易地完成任务。

许多旧的DOS程序不支持鼠标。例如，不能在DOS提示符下使用鼠标。但大部分新应用程序和所有Windows程序都能使用鼠标。

鼠标带来了它自己的词汇：

- **Pointer(点标)**，它是屏幕上的图形符号，用来指示鼠标的位置。当你移动鼠标时，点标也随之移动，没有标准的点标符号。每个程序都有自己的点标。通常，当你在屏幕上移动点标时它的形状会改变，不同的形状显示不同的可利用功能。
- **Point(指向)**，指向一个对象，表示把点标移近或移到对象上。
- **Click(单击)**，单击是指不移动鼠标按住和释放一个鼠标按钮。单击一个对象，要先指向这个对象然后单击。通常，应单击鼠标的左按钮，但有些功能需要使用右按钮或两个按钮一起使用，鼠标这样设计是为了便于使用右手食指单击按钮。如果喜欢用左手，可交换鼠标左右按钮的功能。鼠标手册会告诉你如何做这件事。
- **Double-Click(双击)**，意指在不移动鼠标的情况下，快速连续单击鼠标按钮两次。
- **Drag(拖动)**，拖动一个对象，是指指向一个对象后，按住按钮然后移动鼠标，这样对象就会随之移动了。当对象到达指定地点后，释放按钮，就可把对象放在那儿。(这个技术称为drag-and-drop拖放技术)

鼠标不能在DOS命令提示符屏幕下工作，因为它没有设计成支持鼠标。但是，鼠标可用在DOS Shell，DOS实用程序如MSBACKUP和DEFRAG以及许多基于DOS的应用程序如Microsoft Word for DOS中。为了使鼠标在DOS环境下工作，必须加载一个鼠标驱动程序——它告诉DOS如何控制硬件设备。如果鼠标不工作，检查一下是否装了驱动程序。关于加载驱动程序可参看鼠标手册。(如果没有手册，销售商可能会帮你装上鼠标驱动程序)。

注释： 如果只在Windows下使用鼠标，则无须加载一个驱动程序，Windows在启动时会加载它自己的驱动程序。

1.2 软 件

软件即指程序，没有软件，计算机将无法工作。软件分为两个主要类别：系统软件和应用

程序。区别在于它们是解决系统内部的问题，还是解决系统外部的问题——用户日常工作中的问题。

1.2.1 系统软件

系统软件控制计算机系统和它的资源：内存、磁盘驱动器和微处理器。系统软件包括操作系统：DOS，Windows，设备驱动程序和各种应用程序——帮助管理系统的单独的程序。DOS 软件包包含有多种应用程序，可用来格式化磁盘，管理磁盘目录，拷贝和打印文件。

1.2.2 应用程序

应用程序使计算机成为一得力的工具。应用程序包括字处理程序、电子表格和数据库管理程序等等。一个应用程序帮助用户使用计算机解决计算机系统外部的问题。

1.3 磁 盘

磁盘常用作程序和数据的永久存储。可将常用的程序和数据保存在硬盘中，因为它速度较快，且比软盘容纳的数据要多得多。软盘通常用来在计算机间传送数据——例如，装运程序和数据给客户，还可用作长期的脱机保存。

DOS 为磁盘驱动器分配了一个单字母名字，第一个软驱称为 A，如果有第二个软驱则称为 B；否则，驱动器 A 也指驱动器 B。第一个硬盘驱动器称为 C，增加的硬驱称为 D、E 等等。其他类型的驱动器如 CD-ROM 驱动器，RAM 驱动器，和网络驱动器使用更后的字母。在命令中，常在驱动器名称后放一冒号，使之与其他类型的名字相区别。例如，为了格式化软驱 A 中的磁盘，必须键入以下命令（现在不要做它）：

```
FORMAT A:
```

无论是硬盘、软盘，还是其他，一个计算机磁盘包括一个旋转的圆盘，很像唱盘。它用磁技术在称为磁道的同心圆环上记录数据。

注释： CD 光盘使用光技术读写数据而不是磁技术。

磁盘的每个记录表面都有它自己的读写头。读写臂使磁头在磁盘表面移动以找到正确的磁道。然后，磁盘驱动器等待所需区域转到磁头下方，这样就可正确读写数据。因为磁盘旋转很快，驱动器不能只读写一个字母或一个数字，这样它把磁盘分成扇区，每一次至少读写一个扇区。磁盘一般以 512 字节作为一个扇区。（一个字节表示一个字符的存储空间）。

不是磁盘的所有部分都存储数据，其中一些用来保留系统信息。每个扇区有一个地址，用来标识它自身。每个扇区地址出现在扇区的头部，驱动器硬件通过读取扇区地址来寻找它需要的扇区。

与扇区地址一起还保存着一个特殊值，称为一个 CRC（它代表 cyclic redundancy check，即循环冗余校验），它有助于保证扇区中的数据的有效性。当在一个扇区中写数据时，驱动器控制器执行关于数据的一个特殊运算以得到 CRC 值，它与扇区地址存在一起。当驱动器控制器读一个扇区时，它重新计算 CRC 值，把它与存储的 CRC 值相比较。如果不匹配，驱动器控制器就知道数据出了问题，因而拒绝使用它；用户会发现屏幕上显示出错信息。第 27 章

“CHKDSK 和 SCANDISK”讲述了如何处理这类问题。

磁盘数据出错有几个原因。一个磁盘可能在制造、检验时太马虎，这类错误在DOS把数据写到磁盘上时会出现。另外，代表数据的磁圈一段时间后会移动；如果移动过大，数据就不能再读了，但是，最普遍的数据丢失是由用户操作不当引起的。可用采取以下措施来保护磁盘：

1. 不要使它们靠近磁性物体。要避免放在明显的磁性物体上如电冰箱上的磁铁，也要防止放在麦克风，无线电话接收机，磁纸夹甚至电线上。
2. 把计算机和磁盘放在一个干净的环境里。灰尘、烟雾和其他小颗粒都可使驱动器无法读盘。
3. 保持磁盘干燥、温度适当。
4. 不要触摸记录面。
5. 对于一个5¼英寸的软盘，不要把标签贴到磁盘上后再写标签，也不要折盘。

1.4 数 据

无论数据是医学报告、客户数据库、五年计划，或是不朽的文学作品，它都是系统的关键部分。硬件、软件和磁盘可被代替，但代替数据几乎是不可能的，除非你保存了一份很好的备份，这将在第23章“MSBACKUP 和 MWBACKUP”中介绍。

1.4.1 数据如何存储

可能你听说过bit(位)和byte(字节)这两个名词。Bit代表二进制数字，它是计算机内部表示数据的方法。一个二进制数位只能取两个值0和1，因此在计算机中，它可通过对磁盘某点的磁化和去磁来表示，在电路中则可用有电子脉冲和没有脉冲来表示，等等。

为了建立数字0与1以外的数据，八位二进制数位组成一个byte(字节)。当一个字节用来保存数字数据时，它可处理0(二进制00000000)到255(二进制11111111)之间的数，字节组合起来可以代表更多的数字。当一个字节用来保存字符数据时，它可表示一个字符。每个字节代表的字符取决于使用的字符集。许多不同的字符集都有着共同之处，它们中的大部分基于ASCII字符集。

ASCII码基于七位而不是八位。左端高位通常总是0，因此在ASCII码中只有128个字符。美国英语字母表中所有大小写字母，10个数字，和一个符号集(如句号，逗号等)包含在128个字符之中。当今的字处理程序和其他基于文本的软件需要更多的字符，它们的值在128个保留字节(10000000到11111111)中分配。问题是，那些扩展字符没有被ASCII标准化，这样，一个程序可把10001010看作ë，而另一个程序可把它看作Ç。实际上，相同的程序也可能作出不同的解释，这取决于所使用的字库。

1.4.2 Kb,Mb,Gb和其他

计算机磁盘或内存的容量用Kb、Mb和Gb来表达。在其他领域，Kilo表示1 000；而在计算机中，任何数字都得基于数2，Kilo指1 024(2^{10})。因此一张360KB的磁盘可容纳368 640个字节。1Mb是1 024Kb或1 048 576个字节，因此一个80Mb的磁盘可容纳83 886

080 个字节。1Gb 是 1 024Mb 或 1 073 741 824 个字节，因此 4Gb 内存可容纳 4 294 967 296 个字节。为了方便起见，可以认为 1 000 个字节是 1K，1 000 000(一百万)个字节是 1Mb，十亿字节是 1Gb。在不久的将来，将用 Tb 来衡量内存，它表示 1 万亿字节。

1.5 用 户

通常情况下，操作 PC 机的人被称为用户。系统的剩下部分取决于用户告诉它做什么和什么时候做。本书的真正目的是为了帮助用户在管理硬件、软件、数据和存储介质时作出明智的选择。这样做的关键是要明白系统如何工作，什么会带来错误，如何使用 DOS 提供的工具解决这些问题。有时，也可向外界求救。

1.6 自 测 练 习

- 关于硬件组成和外围设备更详细的描述——如底层技术、标准和具体内容，可查询 Winn L. Rosch 的 Hardware Bible，第三版，SAMS 出版(1994)。
- 为了更好理解计算机各组成部分如何协同工作，以及高层 DOS 命令在底层如何运行，可查询《386/486 技术精粹》清华大学出版社出版(Peter Norton′s Inside the PC，第 5 版，Brady 出版)。

第 2 章　DOS 概述

简介： 对 DOS 理解得越深入，越能更好地控制计算机。

2.1　什么是 DOS

操作系统是一个系统程序集，这些系统程序使得使用计算机成为可能。它提供了用户和硬件及其他软件之间的纽带。DOS 是 IBM 和 Microsoft 为最初的 IBM PC 机开发的。随着个人计算机发展到今天的 486 和 Pentium 机器，DOS 与它们一块成长。DOS 的最新版本，DOS 6.22，比最初的 DOS 精致和强大多了，也更容易使用。

注释： 在本书中，DOS（不带版本号）指任何版本的 DOS。DOS 6 指最新版本的 DOS，它最初作为 DOS 6.0 发行，大约一年以后升级到 DOS 6.2，现在已升级到 DOS 6.22。DOS 6.2 和 6.22 只在讨论 6.2 和 6.22 版特有的功能时才使用。

DOS 有三个主要功能：管理系统资源（如内存和磁盘），访问程序和数据，管理系统。

本章主要介绍操作系统。将进一步学习前面提到的三个主要功能，及一些具体特色的概貌，其操作细节将在后续章节中讲述。在本章中，主要讲述：

- 如何在 DOS 命令提示符或 DOS Shell 下进行 DOS 命令操作。
- DOS 的主要功能，如磁盘和数据管理工具以及数据保护工具。
- DOS 6.0，DOS 6.2 和 DOS 6.22 中新增添的内容。

2.2　DOS 的工作方式

一旦启动系统，DOS 将控制除用户之外系统的每一部分，而用户将控制 DOS。如何使用和控制 DOS？有两种方法供选择：传统的 DOS 命令提示符和 DOS Shell。

2.2.1　命令提示符

自从 DOS 1 以来就使用命令提示符，它有着大大小小的缺点，但许多人还是喜欢在 DOS 命令提示符下工作。图 2.1 是在命令提示符下命令操作的一个例子。当 DOS 显示命令提示符时，它在说，“我在准备接受你的下一个命令”。在图中，用户输入了一个命令显示驱动器 A 中的文件清单，DOS 的 DIR 程序显示这个清单。然后 DOS 显示下一个命令提示符。

对许多人来说，使用命令提示符最主要的问题是不得不记住 DOS 命令。软件的智能常由它理解自由格式语言的能力来衡量，按这个标准，DOS 是一个十足的傻瓜。用户必须用固定的方式对它说话：用有限的词汇和严谨的格式。例如，在图 2.1 中，如果输入 LIST、SHOW 或者 DIRECTORY 来代替 DIR，将无法显示 A 盘上的文件清单。如果用户省略 A 后的冒号，它将产生不同的结果。

```
C:\>DIR A:

 Volume in drive A has no label
 Volume Serial Number is 252B-0FEA
 Directory of A:\

SACK-EGA SEC     37178 12-24-91   8:54P
SACK-TDY SEC     50784 12-24-91   8:54P
DEMO       <DIR>       02-23-93   10:37p
        3 file(s)        87962 bytes
                        908800 bytes free

C:\>
```

图 2.1 命令提示符操作示例

输错命令是经常发生的事，即使是有经验的用户也不得不查询他们不常用的命令。DOS 提供了一个屏幕帮助工具，利用它无需离开机器便可查询命令。第 12 章“在命令提示符下工作”将介绍它的使用方法。

注释： 如果你不能从 DOS 的命令屏幕帮助工具中获得需要的信息，可参考本书末尾的命令用法参考。

使用命令提示符的另一个主要问题是 DOS 并不总是提供用户需要的反馈信息。下面是另外一个命令操作的例子，用户要求 DOS 删除以. BAK 为扩展名的所有文件，整个操作如下：

```
C:\>DEL *.BAK
C:\>
```

DOS 删除了文件，但没有返回关于文件删除的信息，如多少文件被删除，被删除文件的名字，还有哪些. BAK 文件不能被删除等，甚至诸如 Done 之类简单的信息也没有。从其他用户界面，如 DOS Shell 可获得非常丰富的反馈信息。

2.2.2 DOS Shell

最近几年，图形用户界面(GUI)普及得很快，与使用键盘相比，它更倾向于使用鼠标。需要键入一个命令时，可从一个菜单中选择它。可检查一个目录，单击想使用的文件。当删除文件时，可看见这些文件从清单中删除。当移动或拷贝文件时，可在屏幕上看到结果。许多操作甚至不需要菜单便可完成，可把对象在屏幕上拖动从而完成相应工作，这种类型的操作与按严格词汇和语法输入命令相比，对大部分人来说，显得更加自然。

DOS 最早在 DOS 4 中引入了它的 GUI，称为 DOS Shell。DOS 5 对 Shell 作了明显改进 DOS 6.0 没有再对它作大的改变。DOS 6.2 中，一些主要的 DOS 程序取消了这一特性，因为越来越多的人使用 Windows，它是一个更好的 GUI。但是，如果无法使用 Windows，DOS Shell 还是很有用的，毕竟比没有 GUI 好。如果你没有从早期 DOS 版本留下来的 Shell 程序，可从 Microsoft 买一个。参阅 DOS 用户手册可获得更多的细节。

图 2.2 显示了一个 DOS Shell 屏幕的例子。菜单在顶部——被命名为 File、Options 等。在菜单下面，可看到属于本系统的所有驱动器的一个清单。每个驱动器由一个小图形表示，

称为图标(icon),它可区分驱动器的类型:不同的图形分别用于显示软驱、硬驱和 RAM 驱动器。

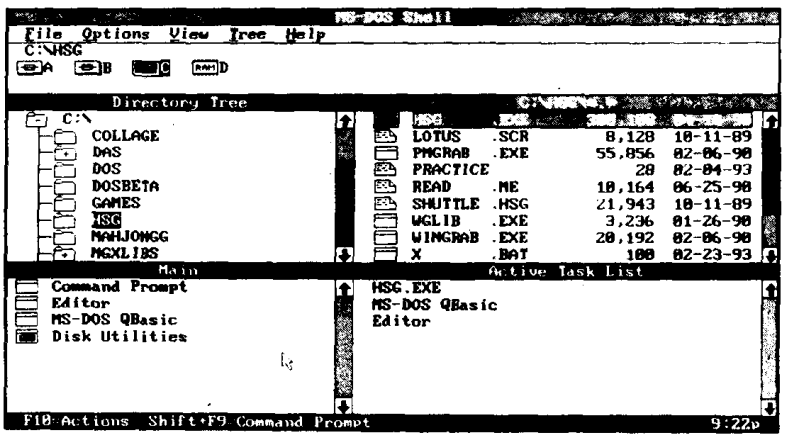

图 2.2　一个 DOS Shell 屏幕

在驱动器清单下面是一个目录树和文件清单,它显示了被选择的驱动器的内容。在底部,标志着 Main 和 Active Task List 区域是选择即可启动相应程序的命令清单。

在命令提示符下,一次只能运行一个命令。直到前一个命令运行完毕后才出现下一个命令提示符(除了内存驻留程序外,它们自成一类,在第 30 章"为 386、486 和奔腾机增加的内存工具"中讨论)。在 DOS Shell 中,你能同时运行几个程序,并可在它们之间来回切换,这个技术称为任务切换(task swapping)。注意图 2.2 的右下角标着 Active Task List 的区域,它显示同时加载的三个程序:HSG.EXE,MS-DOS QBasic 和 Editor。

举一个例子有助于解释在什么情况下需要切换任务。假设,当正在用一个字处理软件编辑文档时突然意识到需要一些电子表格信息。如果正在不能提供任务切换功能的命令提示符下工作,则必须做以下事情:

1. 退出字处理程序(要保证保存了文档)。
2. 打开电子表格程序。
3. 打开想要的工作表。
4. 找到需要的数据后手工抄录。
5. 关掉电子表格软件。
6. 打开字处理程序。
7. 打开编辑文档。
8. 找到文档中以前的位置。
9. 键入从电子表格获得的数据。

现在假设在带有任务切换功能的 DOS Shell 下工作。此时这个工作会变得简单而且迅捷:

1. 从字处理程序切换到 DOS Shell 屏幕。(使字处理程序处于运行状态,处于打开状态)。

2. 启动电子表格程序。
3. 打开想要的工作表。
4. 找到需要的数据后手工抄录。
5. 切换回字处理程序;此时将返回到离开字处理程序时所处的文档位置。
6. 输入电子表格数据。

现在假设电子表格软件已经打开,下次再想查询它,会更容易:

1. 直接从字处理程序切换到打开的电子表格(不需要通过 Shell 屏幕)。
2. 找到想要的数据后手工抄录。
3. 切换回到字处理程序的文档处。
4. 输入从电子表格获得的数据。

前面讲过,与 DOS 相比,Windows 是一个更好的 GUI,上面的过程在 Windows 下会获得更好的服务。在 Windows 下,只要在第二个窗口中打开电子表格软件,使用 Windows 剪切板就可把想要的数据拷贝到字处理程序中。不幸的是,DOS Shell 没有提供直接拷贝的工具,只能手工操作。

许多一直使用 DOS 的用户更乐于使用命令行,因为他们已经习惯在提示符下工作。然而,新用户将会发现 Shell 更容易学会和使用,至少开始是如此。许多有经验的用户也喜欢用 Shell 来移动和拷贝文件。本书重点为那些没有 Windows(或者没有由第三方销售商提供的更精致的界面如 Norton Desktop for DOS)的新用户解释如何使用 Shell。

随着对 DOS 熟悉程度的不断增加,将会发现已学会了越来越多的命令。当能使用更多的 DOS 命令时,也就能更好地控制 DOS。在以后的章节,将学习如何在命令提示符下工作。

2.3 DOS 的功能

DOS 提供了一些工具来帮助完成每天要做的工作,如文件管理等。它也为当系统发生故障时那些必须解决但又不常遇到的问题提供帮助。

2.3.1 命令提示符功能

当在 DOS 提示符下工作时,DOS 用以下多种方式提供帮助:

- 重新调用和修改前面的命令,而无需每次输入一个新命令。
- 创建宏来简化常用的命令。
- 在屏幕上显示信息资料来帮你理解命令的使用和格式。
- 设计自己的命令提示符。
- 如果一个命令显示内容超过一屏幕,可进行分页。
- 可以从屏幕上重定向命令产生的信息到一个文件、打印机或其他地方。甚至可把它作为另一个命令的输入。
- 为一个文件集自动重复运行一个命令。

2.3.2　批处理程序功能

DOS 使用户能创建自己的微型程序，称为批处理程序(batch program)，这是通过命令组合实现的。你不必是一个程序员就可编写批处理程序，许多不知道应用程序或系统编程的人都可编写批处理程序。

- 可测试某个条件是否成立(如 AUTOEXEC. BAT 文件是否存在)，然后决定哪个命令在下一行执行。
- 可显示信息给用户。
- 可从键盘读取一个字符，然后决定下一步做什么，这取决于用户输入的内容。

2.3.3　磁盘和数据管理功能

DOS 的一个主要功能是管理磁盘，从而使所有的程序在所有磁盘上使用相同的存储技术，以形成一个协调性很好的系统。运行 DOS 下的软件并不直接在磁盘上读写数据，而是让 DOS 来做这个工作。大多数情况下，用户甚至没意识到 DOS 在做这件事——因为它在后台工作。

通常，用户使用自己喜爱的应用程序来访问磁盘文件，如字处理程序和数据库管理程序。然而，DOS 提供了一些在大部分应用软件中找不到的工具。

- 它提供了几个命令来管理磁盘。例如，使用 FORMAT 程序可以格式化一张新的磁盘以备使用，使用 DISKCOPY 程序可为整张软盘作一备份。
- DOS 使用一个目录系统来跟踪文件在盘中存储的位置。可使用命令 MD(创建一个目录)，DIR(显示一个目录)，和 RD(删掉一个目录)等来创建和管理目录。
- 也许 DOS 命令中数量最大的一类被用来管理文件。利用它们可以进行移动、拷贝、浏览、打印、删除以及排序等工作。甚至有一个称为 EDIT 的编辑程序，利用它可创建和修改简单的文件。
- 为了防止找不到一个文件——特别在一个大容量硬盘中，这种情况常会发生，用户可按内容或名称来搜寻文件。

2.3.4　保护和恢复数据

在过去 10 年中，大约几百万人使用 PC 机，他们已发现有几百种丢失数据的可能。其中一些很容易发生，例如错误地删除文件。其他一些在现实生活中也常可发现：一个朋友的儿子决定将一些硬币放到他父亲的新计算机中一些看起来很有趣的槽中；另一个朋友试图用剃刀将一张 5¼英寸软盘的标签去掉；或者把一张有用的软盘放在带有磁铁的电冰箱上。

这些失误并不是丢失数据仅有的原因。硬件和介质功能失效是最普通的问题，病毒带来的数据丢失也发生得越来越频繁。

每个 DOS 新版本都增加了一些新工具来帮助用户保护数据避免丢失，在必要时还能进行恢复。DOS 6 现在包括以下功能：

- 可以在一段时间内保留关于删除文件的信息，以便如果改变主意时，还可恢复。
- Microsoft Anti-Virus 检查清除计算机病毒，这最好在病毒可能产生危害之前进行。

- 每一张磁盘在使用前必须格式化，但有时偶尔会错误格式化一张已存有数据的磁盘，用户可以恢复格式化以前盘上的内容。
- 可为一些文件写保护，以防它们被修改或删除。
- 可隐藏文件，这样其他人在目录清单下就看不见这些文件。
- 如果坏数据侵入一个磁盘的目录和文件表，DOS就会找不到某些文件，虽然这些文件是完好的。CHKDSK程序可鉴定和处理这类问题。
- DOS 6.2增加了SCANDISK，它不仅能检测和纠正目录和文件表的问题，还可以扫描磁盘物理表面的瑕疵并挽救坏扇区的数据。
- 可能最重要的是(因为不可能保证数据不丢失)，DOS 6很容易对一些重要文件备份，如果经常做备份工作，就可从任何数据丢失中恢复，包括整个计算机被偷盗或在火中熔化的情形。

注释： DOS给了用户这些工具来保护数据，但是否使用它们却取决于用户自己。本书将讲述如何使用这些工具。

2.3.5 内存工具

有一天，从朋友那儿接到一个电话，他刚开始使用带有16MB内存的Pentium机。当他试图运行一些在旧的286上运行良好的旧软件时，显示出内存不够的信息。有16MB内存怎么还会不够呢？他感到很沮丧。他的问题是他不理解DOS怎样使用内存以及如何重新安排才能使他的应用软件自由地使用这么多内存。花了一点时间，我们就能让旧应用程序顺利地运行起来。

内存是DOS在内部工作的另一个区域，DOS控制着对被称为常规内存(Conventional memory)的一部分内存的访问，大部分程序在这里运行。在286及以上档次的机器中，DOS还控制着内存的其他区域。DOS 6包括完成以下任务的工具和命令：

- 管理常规内存以外的内存区域。
- 把软件加载到专为系统保留的内存区域，这一操作使得在常规内存中留出更多的空间以供应用程序运行。
- 显示报告告诉用户什么程序位于内存的什么地方。
- 配置系统以充分利用内存空间。

2.3.6 优化系统性能

即使拥有最新的硬件，也要仔细使用DOS的优化工具来提高系统的性能。DOS包括以下几个功能可使系统运行更快或给出更多的磁盘空间：

- DOS的DEFRAG程序重新组织磁盘数据以加快访问速度，应定期运行DEFRAG使硬盘处于最佳状态。
- 可创建RAM驱动器，它比实际的驱动器工作更快。一个RAM驱动器使用一定的内存区域来模拟一个硬盘，而不需任何可移动部分。
- DOS的SmartDrive程序在内存中复制磁盘数据，这样DOS可在内存中读写而不需从磁盘上读写。这种复制使程序运行更快，因为它们不因硬盘的相对低速而降低速

度。

- 另外，DOS 6.0 引入了一个新工具，称为 DoubleSpace，它可压缩硬盘上的数据，这样几乎可以保存两倍于从前的数据——它几乎使硬盘存储量加倍，DOS 6.2 改进了 DoubleSpace 工具，用 DriveSpace 来代替它。

2.3.7　硬件功能

DOS 还负责大部分与硬件的交互作用。许多应用程序不直接访问键盘、显示器或打印机，而是要 DOS 为它们做这些事情。

DOS 通过设备驱动程序(device driver)作用于硬件，这是因为驱动程序是真正控制设备的程序。例如，当一个程序想在打印机上打印一行字符时，打印机驱动程序提供代码来定位打印头，选择字体，打印数据等。

驱动程序使 DOS 在它能访问的硬件类型方面更为灵活。如果硬件管理建立在 DOS 内部，操作系统只能处理那些已经建立在 DOS 内部的打印机、显示器和键盘。如果使用驱动程序，DOS 就可以访问任何硬件，只要有相应的驱动程序。

DOS 可以完成以下工作：

- 指定启动时欲加载的设备驱动程序。
- 控制串行口通信参数(如波特率等)设置。
- 设置硬件以显示和打印非英文字母，如 Ñ。
- 使用一个非键盘和显示器组合的设备在用户与 DOS 间进行沟通。
- 为获得最佳系统性能而设置硬件参数。

2.3.8　其他服务功能

DOS 还包含一些其他工具：

- 它包括一些使 DOS 6 更易安装的功能。
- 它包括一个可检查系统并提供诊断信息的程序。
- 可能会有一些为早期 DOS 版本设计的软件，它们并不认识 DOS 6，可设置 DOS 6 以忽略这些程序的版本号。
- 可通过 DOS 设置系统日期和时间。
- POWER 程序有助于维护笔记本的电池。

2.4　DOS 6 的新内容

对于熟悉 DOS 5 的用户来说，可能希望知道 DOS 6 中有哪些新内容。DOS 的基础部分是相同的，DOS 6 增加了一些新工具使你的工作更为容易：

- **数据压缩(Data compression)**。压缩数据可使你在一个磁盘上获取两倍于压缩前的空间。
- **INTERLNK**。使用户能用一根电缆连接两台机器，直接从一台机器拷贝文件到另一台机器。

- **Anti-Virus**。检查和消除系统中的病毒。(在 DOS 6 中包含一个 DOS 版和一个 Windows 版的程序)。
- **MEMMAKER**。使加载驱动程序和 TSR 到高端内存区更为容易。基本上,它可替用户作出决定。
- **DEFRAG**。重组磁盘以消除文件碎片。
- **POWER**。在笔记本电脑中节约电源。

除了这些新工具外,DOS 还改进了一些旧工具:

- **BACKUP**。旧的 DOS BACKUP 程序完全被一个新的 GUI 工具取代,新工具具有更多的功能,并且更易使用。DOS 6 中包含这个备份程序的一个 DOS 版和一个 Windows 版。(旧的 RESTORE 仍然可用,这样就可恢复用早期版本 DOS 所作的备份)。
- **UNDELETE**。DOS 6 的 UNDELETE 包括 Delete Sentry,它可将删掉的文件保留几天。这个安全的方法使正确恢复文件更为容易。在 DOS 6 中,UNDELECT 包括 DOS 版和 Windows 版。
- **DEVICEHIGH 和 LOADHIGH**。这两个命令现在包含各种选项,使用户能控制把 TSR 加载到高端内存的什么地方。新的 MEMMAKER 程序可为每个 TSR 决定最好的位置,并为用户创建 DEVICEHIGH 和 LOADHIGH 命令。
- **On-screen command help**。这是一个全新的命令帮助工具,它为 DOS 命令提供完整的联机资料(不仅仅是 DOS 5 提供的格式总结)。

推出 DOS 6.2 的主要目的是改进 DoubleSpace 的可靠性,因为它在一些系统中遇到了问题。DOS 6.22 用 DriveSpace 代替了 DoubleSpace,不是因为技术上的原因而是因为法律上的原因。

DOS 6.2 和 DOS 6.22 同样包括对 DOS 作的一些进一步的改进:

- **SCANDISK**。SCANDISK 检查驱动器,寻找物理表面和目标结构中的问题;它可纠正它发现的大部分问题。
- **Overwrite Protection**。COPY,XCOPY 和 MOVE 现在在冲掉一个文件前会发出警告。
- **SmartDrive**。SmartDrive 现在可缓冲 CD-ROM 驱动器,并且在其他几个方面作了改进。
- **Interactive Startup**。在启动时按 F8 键,DOS 现在会像运行 CONFIG.SYS 一样,逐行运行 AUTOEXEC.BAT 文件。
- **Interactive Batch Programs**。现在可逐行运行批处理文件。
- **DISKCOPY**。不再需要切换磁盘就可在一个驱动器中拷贝一个软盘。
- **Enhanced Readability**。现在 DOS 可对程序输出消息中大于 999 的数字插入逗号,这类程序包括 DIR,FORMAT 等。
- **DEFRAG**。可消除更大数目的文件碎片。

2.5　自 测 练 习

本书的简介中曾说过，不能只学习几个应用程序而忽视 DOS。本章指出了一些方法来证明这句话。看看在以下各个领域你是否能记住至少两个 DOS 功能：

- 保护数据免遭意外或故意的破坏。
- 如果发生意外，如何恢复数据。
- 避免出现 Out of memory 这样的消息。
- 管理磁盘及数据文件。
- 使系统运行更快、更顺利。
- 在磁盘上存放尽可能多的数据。

第 3 章 DOS 的数据组织方式

简介： 一旦懂得 DOS 如何在磁盘上组织数据，就能更好地利用 DOS，而不至于陷入困境。

3.1 DOS 控制数据的方式

DOS 的一个主要职责是控制所有保存在磁盘上的数据。如果某个应用程序想读写或删除一个文件，它将要求 DOS 进行实际的磁盘访问。用这种方法，DOS 能确保不会有一个应用程序覆盖或破坏另一个应用程序的文件。

DOS 采用一种结构化方法保存数据。如果知道 DOS 如何组织磁盘，就能更好地利用 DOS，而不至于陷入困境。本章主要讲述：

- 如何给文件命名
- 如何写一个文件名说明(filespec)来同时访问一组相关的文件
- DOS 如何用属性保护文件
- DOS 如何跟踪磁盘上的文件
- 如何访问不在当前目录下的文件
- DOS 如何删除文件

同时将讲述以下术语：

.

..

archive(文档)

attribute(属性)

branch(分枝)

chain(链)

child directory(子目录)

cluster(簇)

current directory(当前目录)

default directory(缺省目录)

default drive(缺省驱动器)

directory tree(目录树)

end-of-file(EOF)(文件结尾符)

File Allocation Table (FAT)(文件分配表)

filename(文件名)

filespec(文件名说明)

global filespec(全程文件名说明)

parent directory（父目录）
path name（路径名）
root directory（根目录）
slack（屑）
subdirectory（子目录）
wildcard character（通配符）

3.2 文 件

要在磁盘上保存一些东西，必须把它放在一个文件中。文件与一个文件夹中的一张纸相同。它可以是一封信，一个数据库，一个电子表格，一个图形，一个章节，一个程序，一种字体或其他内容。

3.2.1 文件名

在DOS中，每个文件必须有最多八个字符的文件名，可以使用字母、数字和以下符号：

_ ^ $ ~ ! # % & - { } ( ) @ ' '

DOS不区分大小写字母，如TechData、TECHDATA和techdate表示相同的名字，无论怎样键入这个名字，DOS都按名TECHDATA存储。

注释： 以下特殊文件名保留给DOS专用：CLOCK $、CON、AUX、COM1到COM4、LPT1到LPT3、NUL和PRN。

八个字符的文件名只是文件名的一部分。文件名还可有一个最长三个字符的扩展名，它通过一个点号与文件名连在一起，如TECHDATA.JUN或DESIGN.1。可使用扩展名区分文件类型，但这不是必需的。例如，Microsoft Word为它创建的每个文档文件加扩展名.DOC；DOS使用.TXT表示文本文件，HotShot Graphics使用.HSG表示它的图形文件。

如果避免在其他文件上使用应用程序中的扩展名，就会省去许多麻烦。如果使用Microsoft Word，就不要使用.DOC作为用DOS的EDIT创建的文件扩展名，因为Microsoft Word将不能正确打开这些文件。

DOS赋予以下扩展名以特殊含义，不要为其他类型的文件使用这些扩展名：

.COM 一类可执行程序文件
.EXE 另一类可执行程序文件
.BIN 另一类程序文件
.SYS 一类设备驱动程序或配置文件
.BAT 批处理程序文件
.HLP 帮助文件

作者忠告： 给文件以有意义的名称。

如果你命名一个文件为C23F4-XT.D#2，当频繁使用这个文件时，你可能会记住它，但是一年以后，想从一个有上千个文件的硬盘中找到它，将会浪费很多时间，因

为你不能记起这个文件。如果将文件命名为BUDGET94.DAT，就能在以后很容易地找到它。

像C23F4-XT.D#2这样的名字还增加了键入文件名的难度，也增加了误键的可能性，从而导致一个DOS命令失败。诸如BUDGET94.DAT这种名字的文件就不易出错。

3.2.2 全程文件名

当你想访问一个特定文件时，可使用它的文件名。例如，文件名TECHDATA.JUN只指一个文件。但是，有时处理全程范围内的相关文件更为方便，通常可使用一个全程文件名(global filespec)来做这件事。一个全程文件名(global filespec)包含通配符(wildcard characters)，它们用来代表任何其他字符。DOS通配符为"?"和"*"。

3.2.2.1 ? 通配符

? 可匹配除空格之外的任何一个字符。例如，全程文件名TR? CK可匹配TRACK，TRICK和TRUCK，但不能匹配TRCK或TRUELUCK，注意它也不能匹配TRACK.100，TRICK.DAT或任何其他有扩展名的文件名，因为TR? CK指定没有扩展名。

当? 出现在文件名的尾部或扩展名中时，它也可以匹配一个空格。例如，全程文件名GRAD?，匹配GRAD，还有GRADE和GRADY等。

可在一个样本名中使用多个问号。文件名? ART??.?? 匹配任何包含ART在其第二、三、四个字符位，且其后最多二个字符、扩展名最多有两个字符的文件。

警告： 使用? ART??.?? 可能会访问到比你想要的更多的文件。首先运行DIR ? ART??.??总是一个好主意，这样DOS将列出匹配它的文件名清单，第14章"目录管理"中将解释DIR命令。

3.2.2.2 * 通配符

* 通配符匹配任何数目的字符，包括空格。全程文件名TR*可匹配TR，TRY，TREK，TRIAD，TRADER，TRUDEAU和TRIPTALK等，但不匹配TRUST.ME，因为它没指定扩展名。

在一个全程文件名中最多可使用两个星号，一个放在文件名中，另一个放在扩展名中。DOS在文件名的两部分都忽略*后面的任何字符。因此，RE*.T*是合法的，而*RE*.TMP不合法。实际上(同时又是不幸的)，DOS中没有直接的方法来指定一个全程文件名包含RE在文件的任何地方。

提示： 当创建一组相关文件名时，应事先规划名字以便它们可被一个全程文件名访问，这样它们就会按字母顺序同时出现。例如，名字DAY1.DAT到DAY7.DAT就比SUN.DAT到SAT.DAT好。

如果在一个文件集中使用大于9的顺序数字，就使用01到09来保证开始的9个文件按字母顺序排列。例如，TRIAL1到TRIAL12将按以下顺序排列：TRIAL1，TRIAL10，TRIAL11，TRIAL12，TRIAL2，TRIAL3等，而TRIAL01到TRIAL12将以正确的数字顺序列出。

3.2.3　属性

每个文件有五种属性：只读、隐含、系统、归档和目录。这些属性控制着 DOS 和其他程序如何访问文件。每个属性可开或关。当一个属性开(on)时，程序就会注意它(注意下面的警告)。否则程序就忽略它。

警告： 不要认为只要设置了属性就可保护文件不受破坏。许多程序是忽视或凌驾于属性之上的。还有，任何懂得 DOS 基础知识的人都知道如何改变属性。

3.2.3.1　只读(Read-Only)

当只读属性打开时，文件不能被修改或删除。如果试图修改或删除一个只读文件，大部分 DOS 实用程序会显示一个 Access denied 消息。(在本书的后面章节中会看到例外的情况)，大部分程序是遵循只读属性的，但有的程序就不这样。

虽然当有人想故意破坏一个文件时，这个属性无能为力，但它却可防止意外的破坏。如，DOS 的 DEL 命令就不会删掉一个只读文件，DOS 的 COPY 命令也不会冲掉一个只读文件。

3.2.3.2　隐含(Hidden)

置隐含属性意味着一个程序寻找文件时忽略带有这种属性的文件。例如，DIR 命令在正常情况下不显示隐含文件(但用户可以改变)。DEL 命令不会删除一个隐含文件，因为它看不到隐含文件；但你会得到消息：File not found。但是，并不是所有的人都不会发现隐含文件。这个属性只能防止意外事故而无法防止蓄意破坏。如果想避免意外移动、拷贝、删除或处理一个文件，就应使之隐含。

3.2.3.3　系统(System)

系统属性将 read-only 和 hidden 组合在一起；一个系统文件在正常情况下不能修改或删掉，在搜寻文件时，它通常不会被程序看到。

很少有人使用系统属性。实际上，直到 DOS 5，用户不能自己控制这个属性，因为 Microsoft 认为对非程序员来说没必要做这件事。DOS 的核心程序文件，IO. SYS、MSDOS. SYS 和 DBLSPACE. BIN 或 DRVSPACE. BIN，都具有系统属性，其他系统程序也可能有一些系统文件。但很少有应用程序需要这样保护它们的文件。

3.2.3.4　归档(Archive)

归档属性表示一个文件从最后一次备份到现在已被修改。这个属性表示文件需要归档(Archived)或备份。DOS 的备份程序和大部分其他备份程序可按这个属性选择要备份的文件。备份完毕后，它们置归档属性为无效，因为此时，这个文件无需备份。

DOS 在创建或修改一个文件时自动设置文件的归档属性。这一操作意味着这个文件需要备份。本书后面章节，将学习如何为文件设置归档属性以及如何关闭此属性以控制下一次备份是否选择该文件。

3.3　目录(Directories)

DOS 使用目录来跟踪文件。一个目录就像一个电话号码本，但是用以下项代替了姓名、

地址和电话号码：

文件名
扩展名
文件大小
最后一次修改的日期
最后一次修改的时间
文件在磁盘上的起始位置
属性

DOS 使用这些信息来定位一个文件，也可显示一个目录看看它包含的文件。一个典型清单如下所示：

```
BUDNEWA    BAK    124,788  04-29-93    12:23a
SCRIPT     STY        512  07-24-90     9:23p
BUDNEWA    DOC    120,653  08-23-93    10:42a
FLOOD1     BAK     41,210  08-23-93    10:10a
FLOOD1     DOC     42,008  08-23-93    10:29a
```

每行描述一个文件。从中可看到文件名、扩展名、大小、最后一次修改的时间和日期。在这种类型的清单中，DOS 不显示属性或文件在磁盘上的开始位置。

3.3.1 根目录

每个磁盘至少有一个目录，称为根目录，它必须位于第一磁道。DOS 在它准备磁盘时安装根目录。根目录含有有限数目的目录项，多少取决于磁盘的大小。在一个硬盘上，你可能需要比根目录允许范围更大的目录空间。下节解释如何获得这个空间。

3.3.2 目录树

用户可创建自己的目录，它是根目录下的子目录，子目录的作用有时像个文件夹，把一组相关文件放在子目录中使它们与其他文件分开，也可使它们更易发现。例如，在硬盘上为主要程序创建几个分开的子目录：DOS、Windows、电子字处理程序、电子表格软件等。你想创建多少子目录就可创建多少子目录。

每一个子目录还可以有几个子目录，这些子目录还可依次有自己的子目录。例如，可以在字处理程序中为往来信件、文艺作品、工作表格、所写文章各建一个子目录。

一个磁盘的目录结构像一棵倒立的树：根目录在顶部，然后分成第一级子目录，还可依次分成第二级子目录等等。图 3.1 显示了一个磁盘目录结构的例子。可看到顶端是根目录；它的名字总是\。DOS 指定这个名字，用户不能改变它。子目录的命名方式与文件相同，如果需要可包含扩展名(许多人在子目录中不使用扩展名)。

注释： 子目录像文件一样也有属性，只是大部分人并不使用它们。

当一个目录有子目录时，这个目录称为父目录，子目录称为它的孩子，在图 3.1 中，ANIMATE 是 CRITTERS，SOUNDS 和 TOONS 的父亲，它们都是 ANIMATE 的孩子。

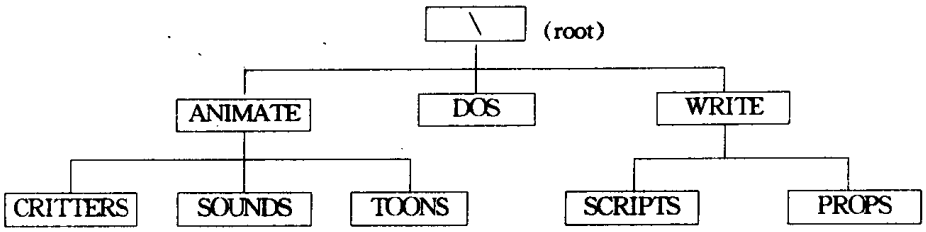

图 3.1　目录树示例

提示： 通常，应把文件保存在子目录中，而不应保存在根目录中。这不仅是因为根目录中空间有限，而且在有些情况下，当 UNFORMAT 程序不能恢复根目录中的文件时，这就变得更有必要了。

当列出一个目录的内容时，其子目录的 Size 栏显示<DIR>。日期和时间表示目录创建时的日期和时间，而不是最后修改的时间。下面是一个目录内容清单的样本。

```
.          <DIR>          10-05-93    10:32p
..         <DIR>          10-05-93    10:32p
FLOOD3  DOC        42,496 10-06-93     7:52p
JULY       <DIR>          10-07-93     7:38p
FLOOD3  STY          640  10-06-93     7:52p
TREAT   DOC      116,608  10-06-93     7:56p
FLOOD1  DOC      140,416  10-06-93     8:01p
BUDDY1  DOC       94,080  10-06-93     8:15p
JUNE       <DIR>          10-15-93    10:38p
```

这个目录包含五个文件和四个目录项。JULY 和 JUNE 是子目录。出现在每个子目录顶部的两个目录项帮助 DOS 跟踪目录树。第一个项用一个句点表示，代表这个目录本身。第二个目录项用两个句点表示，代表这个目录的父目录。用户不能看见父目录的名字，但可看出它是在 October 5，1993，10：32 p. m. 创建的。用户不能看见而 DOS 能看见的是这两个目录的起始位置，它帮助 DOS 把所有目录联结在一起。

提示： 可以用名字.和..表示当前目录和它的父目录，这比输入完整的目录名更为方便。

3.3.3　当前目录

每个驱动器都有一个缺省目录。当指定驱动器而不指定目录时，DOS 就访问缺省目录。例如，拷贝一个文件到驱动器 A，但不具体指定哪个目录，A 盘的根目录将成为这个驱动器的缺省目录。本书的后续章节将介绍如何使另一个目录成为缺省目录。

另外，总有一个驱动器是当前驱动器。除非指定另一个驱动器，DOS 总是访问当前缺省驱动器。如果拷贝文件到一个名为 BACKUPS 的目录，但不指定驱动器，DOS 就假定 BACKUPS 在当前驱动器上。当前驱动器上的缺省目录称为当前目录。

启动时，启动驱动器成为当前驱动器，以后章节将介绍如何转换到另一个驱动器。

当在一个 DOS 命令中既不指定一个驱动器，也不指定一个目录时，DOS 即认为是指当前目录。例如，如果输入一个命令删除 PCLIST. TXT 时没有指定驱动器或目录，DOS 将在当前目录中寻找此文件。

DOS 命令提示符通常显示当前驱动器和当前目录的名字，这样用户就能知道当前所处的驱动器和目录。例如，以下命令提示符表示驱动器 C 上的 DOS 目录是当前目录：

```
C:\DOS>
```

在 DOS Shell 中，用户总能知道当前目录，因为它显示在屏幕上。

3.3.4 路径名

当要访问一个非缺省目录时，必须使用它的路径名。路径名显示 DOS 如何到达想要的目录。路径名可以包含一个驱动器(如果不在当前驱动器)和一系列用反斜杠分开的目录名。例如，下面的表达式是图 3.1 中到达 TOONS 目录的正确路径名：

```
C:\ANIMATE\TOONS
```

这个名字告诉 DOS 到驱动器 C，从根目录开始(第一个反斜杠)，找到它的子目录 ANIMATE 后，再找 ANIMATE 的子目录 TOONS。如果 C 是当前驱动器，可省略路径名开头的 C:。

用户也可以从当前或缺省目录开始输入一个路径，而不是从根目录开始。假设当前目录是 C:\，想使用 TOONS 目录时，可用下面的路径名：

```
ANIMATE\TOONS
```

如果路径名不从代表根目录的反斜杠开始，DOS 便知道是从当前目录开始。如果当前驱动器是 A，要想访问驱动器 C 上的 TOONS 目录，必须把驱动器名加在路径中：

```
C:ANIMATE\TOONS
```

而且，因为根目录并未被指定为启始点，DOS 从指定驱动器的缺省目录开始。

现在，假设 TOONS 是当前目录。如果想到达 CRITTERS，路径可以从根目录开始：

```
\ANIMATE\CRITTERS
```

但 DOS 提供了一个捷径，可以用名字..告诉 DOS 从当前目录的父目录开始，形式如下：

```
..\CRITTERS
```

子目录完整的路径名应从代表根目录的反斜杠开始，不能超过 63 个字符。当设计目录结构时，总希望名字越短越好，一个 63 字符的路径名使用命令来说是一个负担。经验证明，最好不要创建多于四级的目录结构。

3.3.5 文件名中的路径

一个文件名可在文件名和扩展名前加入路径名以精确定位所需文件。如果当前目录是 C:\ANIMATE，想访问的文件 NEWBUMPS.WAV 在 A:\SOUNDS 中，可使用如下文件名：

```
A:\SOUNDS\NEWBUMPS.WAV
```

3.4　分配单元(Allocation Units)和文件分配表(File Allocation Table)

磁盘驱动器将磁道分成扇区,每次只能读写一个扇区。但如果 DOS 不得不跟踪一个磁盘上每个文件的每个扇区,它的系统表将占用比文件本身更多的空间。为了把系统表的大小减小到合理的尺寸,DOS 把扇区分组成分配单元(也叫簇 cluster),然后用跟踪簇来代替扇区。

分配单元的大小取决于磁盘的总体大小。当 DOS 准备磁盘以备使用时,它决定簇的最合理尺寸,软盘每簇大约包括一或两个扇区,大容量硬盘每簇可能包括四个或更多个扇区。

因为 DOS 跟踪簇而不是扇区的位置,它不能将簇的一部分给一个文件。它总是整簇整簇地分配文件空间;这也是为什么称它为分配单元的原因。甚至即使文件只有一个字节,DOS 也分给它一个整簇,剩下的空间(称为屑,slack)就浪费了。这样,当文件扩大时,有许多空间供使用。如果文件超过了一簇,DOS 将再分配给它一个簇。

很明显,这种分配单元法浪费空间,但 DOS 必须在浪费屑空间和占用更多时间跟踪文件间作出协调。

DOS 分配单元系统产生了许多易使 DOS 用户困惑的问题。例如,当从一个 1KB 簇的磁盘拷贝一个小文件(如 200 字节)到一个 4KB 簇的硬盘时,文件的大小还是 200 字节,但它要在硬盘上占用 4KB 空间。如果拷贝一组 20 个小文件,尽管所有小文件实占空间之和不足 1KB,但在一个 40KB 的硬盘上却可能没有足够的空间容纳它们。

本节将讨论 DOS 如何跟踪一个文件的分配单元。图 3.2 显示了一个文件分配表(FAT)的例子,这是 DOS 的系统表。它包含磁盘上每个簇的项。

驱动器的每个簇有一个数字地址,从 2 开始一直到磁盘结尾。文件的目录项显示了这个文件中第一个簇的地址。在 FAT 中,簇项显示了下一个簇的地址。这种模式一直持续到文件的最后一个簇,它包含一个特殊的代码表示文件结束(EOF)。因为 FAT 的各项间以链状连接,所以文件的项集也称为文件链。

假设 GROWTH92.DB 的目录项表示它从簇 5 开始。在图 3.2 的 FAT 中找到簇 5(记住这里的簇号从 2 开始,不是从 1 开始)。发现它包含一个 6,因此 6 是下一个簇。簇 6 连到簇 10,簇 10 连到簇 11。簇 11 显示文件的结束记号。因此,这个文件一共占用 4 个簇。这由存放在目录项的文件的实际大小确定。

		EOF	0	0	6	10	EOF
EOF	0	11	EOF	13	14	15	EOF
0	0	BAD	20	21	EOF	23	24
37	0	0	0	0	0	31	32
33	34	35	36	54	38	39	EOF

图 3.2　文件分配表示例(前边 38 项)

从图中还可看出 GROWTH92.DB 是一个分段存储的文件。因为它的簇不是毗邻的,而是被分割成两个碎片。DOS 常常分开一个文件以充分使用磁盘上的有用空间,尤其增大一个文件时,这种文件碎片更容易发生。例如,如果增大 GROWTH92.PB 直到它需要另一个簇,增大部分不能占用簇 12,因为簇 12 被另外一个文件占用了。它不得不使用下一个可用

单元，在 FAT 中标志为 0，本例中是簇 16，因此，又产生了一个碎片。

许多文件只需要一个簇。假设文件 ALLISON. MEM 从簇 2 开始。在 FAT 中，簇 2 就包含一个 EOF 记号；这是这个文件唯一占用的簇。

因为 FAT 对控制文件和磁盘上的空余空间都十分重要，所以 DOS 保存了两个相同的 FAT——一个是另一个的备份。每当创建一个新文件，或者增大、减小一个已有的文件时，DOS 需要修改两个 FAT。如果在修改完成之前中断它，FAT 就会出错，以后可能就不能再正确地访问文件了。为了避免这个问题，当驱动器灯亮时不要重新启动或关闭电源(除非需要中断一个病毒)。驱动器灯亮表示 DOS 正在访问驱动器。

3.5 DOS 如何删除一个文件

当需要 DOS 删除一个文件时，它不会立即从磁盘上擦去文件的数据，只是在文件目录项的第一个字符处存入一个特殊的代码，把这个文件的 FAT 项改为 0，这样，所删文件的目录项及簇就可为其他文件所用。当增大另一个文件时，它可能占用一个或多个释放的簇，这就破坏了被删除的文件的数据。当创建一个新文件时，也可能使用被删文件的目录项并占用它的一些簇。如此下去，所删文件的所有痕迹就消失了，但它可能会花一段时间。因此，恢复全部或部分被删去的文件数据是可能的。第 26 章“UNDELETE 的使用”将讨论 DOS 的恢复工具 UNDELETE。

3.6 自 测 练 习

在本书以后的章节中，假定读者已理解了数据存储概念，如文件名、路径名和文件分配表(FAT)等。试回答以下问题看是否能够继续：

- 哪一个是更好的文件名？JAN. BAT... DEC. BAT，MONTH01. BAT... MONTH12. BAT，还是 MONTH1. BAT... MONTH12. BAT？
- 想象在没有指定目录名的情况下，拷贝一个文件到驱动器 A。A 驱上的哪个目录将接收这个文件？
- 写出图 3.1 中 SCRIPTS 目录的路径名，假设 DOS 目录是当前目录。
- DOS 跟踪驱动器上的所有簇的表的名字是什么？
- UNDELETE 工具可以在文件删除后恢复它们。DOS 删除文件后，在什么条件下恢复才能成功？

第4章 系统设置

简介： 本章将指导从早期DOS版本升级的用户完成安装过程。

4.1 基本假定

本章以从DOS早期版本升级到DOS 6.22为例。如果从OS/2或其他操作系统转到DOS 6、22，所需操作过程可参考用户指南。本章还假定启动驱动器是C，当前版本的DOS安装在目录C:\DOS下；如果不是这样，无需调整命令，使用当前启动驱动器和DOS目录名即可。

注释： 如果以前没接触过DOS，可能不知道如何做下面的工作，不妨向有经验的DOS用户请教。如果没进行所推荐的准备工作就试图运行SETUP，很有可能出错而丢失硬盘数据。

4.2 准备安装

本节介绍在PC机上运行SETUP前的准备工作。需要做的事情包括：备份所有数据，准备一张紧急启动盘(万一SETUP失败，用于恢复硬盘)，想好在安装过程中必须作出的决定，准备需要的软盘。

1. 确保启动驱动器上有足够的自由空间。根据不同需要，升级到DOS 6.22需要5MB到7MB的空间，如果没有这么多的空间，可删除不再需要的文件。如果没有足够的文件可删，可在安装DOS时把一些文件移到软盘上，以后再恢复到硬盘驱动器。(如果启动驱动器被压缩了，那么在压缩驱动器上而不是在宿主驱动器上有可用的空间)。
2. 使用当前备份程序备份硬盘。如果可能，就备份所有的文件，包括程序文件。一旦SETUP失败后就需要恢复硬盘，如果在其他地方保存了所有的文件，恢复将会更容易。由于某些原因如果不能备份整个驱动器，丢下可从它们源盘重新安装的所有程序文件。
3. 为了给当前DOS版准备一张紧急启动盘，在驱动器A中插入一张新盘或没有重要数据的盘，键入如下命令：

   ```
   FORMAT A: /S
   ```

   警告： 这个过程将破坏软盘上的任何数据。

4. 使用以下命令拷贝FDISK、FORMAT和SYS程序到软盘上。因为一旦SETUP失败而不得不恢复硬盘时，就需要使用这些程序。

   ```
   COPY C:\DOS\FDISK.EXE A:
   ```

```
COPY C:\DOS\FORMAT.COM A:
COPY C:\DOS\SYS.COM A:
```

5. 如果使用 DOS 的 BACKUP 程序(在 DOS 5 或更早版本中)按步骤 2 备份,还需要拷贝 RESTORE 程序到紧急启动盘,这样在 SETUP 失败时可恢复备份文件。在命令提示符后键入以下命令再按回车键:

```
COPY C:\DOS\RESTORE.EXE A:
```

6. 如果从 DOS 6.0 或 DOS 6.2 进行升级,而且使用 MSBACKUP 或 MWBACKUP 按步骤 2 备份,还需将所有的 MSBACKUP 程序拷贝到紧急启动盘上。键入以下命令:

```
COPY C:\DOS\MSB*.* A:
```

注释: *一张 360KB 的盘容不下所有的 MSBACKUP 程序文件。应使用一个能处理的更大容量盘的驱动器。如果驱动器 A 是一个 360KB 的驱动器,你将不得不把一些文件放在另一张软盘中。*

7. 把紧急启动盘放在一个安全的地方,当然你会希望用不着使用它。当新版 DOS 安全安装,正常运行后,这张软盘将不再有效,因为它是为 DOS 的早期版本建立的。只有做完对 DOS 6.22 的一张完整备份后才能丢掉前面那张软盘的内容。直到这时,才不再需要那些备份程序。
8. 在运行 SETUP 时,要使任何会在屏幕上显示消息的程序失效。例如,如果使用一个弹出式提示信息来提醒有约会,应将它关闭。如果有一个后台运行的通信程序来接收传真或邮件,也要将它关闭。
9. 取消所有的磁盘缓冲、删除保护和反病毒程序。这类程序将阻止 SETUP 正常运行。(没必要取消 DOS 的 SmartDrive)。如果 CONFIG.SYS 和 AUTOEXEC.BAT 文件加载了这类程序,用以下步骤卸载:

   a. 使用一个 ASCII 文本编辑器(如 DOS 的 EDIT)打开 C:\CONFIG.SYS 文件。(如果使用字处理程序,要保证文件工作在 **ASCII** 模式下,不要把它作为一个文档文件)。
   b. 在加载磁盘缓冲、删除保护或者反病毒程序的命令行前插入 REM。
   c. 保存文件。
   d. 对 C:\AUTOEXEC.BAT 文件做相同的事。
   e. 退出编辑器。
   f. 按 Ctrl+Alt+Del 重新启动计算机。

   重新启动计算机可从内存中移去加了 REM 标记的程序。

注释: *成功升级后,要删掉插入的 REM,这样重新启动计算机时就会重新加载那些程序。*

10. 退出 DOS Shell 或 Windows,必须在 DOS 命令提示符下运行 SETUP。
11. 如果硬盘没有被第三方压缩软件如 Stacker 压缩的话,SETUP 将需要一或两张软盘以便它能创建 Uninstall 盘。利用 Uninstall 盘,当新版本无法在系统中正常工作时

可以恢复从前的 DOS 版本。确保有一张空软盘插在 A 驱动器中，如果 A 驱是 360KB 的驱动器(双密度 5¼英寸)，则需要两张空盘。

12. 如果使用 Windows，要决定是安装 DOS 还是 Windows 版本的 UNDELETE，BACKUP 和 Anti-Virus。可以选择其中任意一种版本，也可以两种全选，或者一种都不选。

事实上，用户只需要 Windows 或 DOS 版本的 BACKUP 和 Anti-Virus，但最好安装两个版本的 UNDELETE，原因如下：

- 无论在什么程序下工作，都可随时作出 UNDELETE 的决定；如果正在使用的不是 Delete Sentry，当需要删除一个文件时不启动 Windows 是非常重要的。
- Windows 版本的 UNDELETE 有许多 DOS 版本没有的特点。

注释： 如果你以后在选用哪个版本上改变主意了，可重新运行带/E 的 SETUP 来增加 DOS 或 Windows 版本的 BACKUP、UNDELETE 及 Anti-Virus。(参见本书末尾的命令用法参考里的 SETUP，其中有关于/E 的解释)。

4.3 作出关于备份的决定

在升级过程中，必须决定是否将 Backup 程序升级到 DOS 6.22。建议最好这样做，但在某些情况下，也应该将旧的备份程序保留一个备份。在有些情形下，DOS 6.22 不能从已作的备份中恢复，必须使用以前的备份程序恢复。

DOS 在 DOS 6.22 中改变了它的压缩方法，新的 MSBACKUP 和 MWBACKUP 不能读 DOS 6.0 或 6.2 所作的压缩备份，除非下面条件之一为真：

- 硬盘之一当前被 DoubleSpace 压缩。(如果升级到 DOS 6.22 的 DriveSpace，将无法读取旧备份)。
- 在 DOS 目录下，必须有 DBLSPACE.BIN 和 DRVSPACE.MR1 这两个文件，它们使 DOS 6.22 能够读取 DOS 6.0 或 6.2 所作的压缩备份。如果正好从 6.0 或 6.2 升级，在 DOS 目录已有了 DBLSPACE.BIN(只要未被删除)，SETUP 就不会破坏它。只有在使用 DOS 6.22 的 Stepup 版本而不是 Upgrade 版本或者使用一个非英语版的升级程序进行升级时，SETUP 才在 DOS 目录下安装.MR1 文件。英语升级软件包里不包含.MR1 文件。(如果需要这个文件，可从 Microsoft 购买)。

如果认为 DOS 6.22 不能读取从前的备份或者不能确定，推荐使用以下两个命令对当前备份程序作一个备份：

```
MD \OLDBACKS
XCOPY\DOS\MSB*.* \OLDBACKS
```

现在可安全地升级到 DOS 6.22 的备份程序。如果需要从旧的压缩备份恢复，转到 OLDBACKS 目录运行 MSBACKUP 即可。(如果需要恢复 DOS 6.22 中所作的备份，不要转到 OLDBACKS 目录，运行 DOS 目录下的 MSBACKUP 或 MWBACKUP 即可)。当不再需要 DOS 6.0 或 6.2 作的备份时，可删掉 OLDBACKS 目录。

4.4 运行 SETUP

注释： 在驱动器 A 或 B 中均可使用安装盘。如果用双密盘(360KB 或 720KB)代替高密盘，必须使用用户指南后面的订单订购。

把标有 Disk 1-Setup 的盘插入驱动器 A 或 B。键入下面命令：

```
A:SETUP
```

或

```
B:SETUP
```

SETUP 在屏幕上显示信息和指令。在选择了想装入哪个版本的 UNDELETE、BACKUP 和 Anti-Virus 后，可能就不用作更进一步的选择了，但仍需要按要求插入磁盘。如果启动驱动器没被第三方压缩程序压缩，SETUP 将提示什么时候插入作为 Uninstall 盘的空盘。

4.5 系统设置中问题的处理

如果 SETUP 在完成 Uninstall 盘后遇到了问题，它可能会要求使用 Uninstall 盘以便恢复从前的 DOS 版本。《用户指南》中题为“Diagnosing and Solving Problems(问题诊断与解决)”一章介绍了处理普通 SETUP 问题的过程。如果不能正常运行 SETUP，可与 Microsoft 的 Product Support Services(产品支持服务机构)联系以获得帮助，《用户指南》中有与他们联系的方法。

如果 SETUP 使启动驱动器处于不能使用的状态，可能需要恢复这个驱动器。如果一直遵循本章推荐的准备步骤，就应有恢复工作所需的数据。不过，最好的选择是打电话找 Microsoft's Product Team 寻求帮助。通常，可能需要重新分区和重新格式化驱动器，然后重新启动 DOS，从备份中恢复所有文件。

注释： 先前制作的紧急启动盘包含以下操作所需要的信息：

- 使用旧系统进行启动。
- 使用 FDISK.EXE 对硬盘重新分区，FDISK.EXE 将在第 54 章“FDISK”中解释。重新分区可能是不必要的，如果 FDISK 显示分区已存在，就不需要改变它们。
- 使用带/S 选项的 FORMAT 格式化硬盘，FORMAT 将在第 17 章“格式化磁盘和取消格式化”中解释，这样就可以在硬盘上安装以前的 DOS 版本。(可在开始恢复文件之前立即运行 SETUP 以升级到 DOS 6.22)。
- 使用旧的 backup/restore 程序恢复所有其他文件，这将在第 24 章“Restore 和 Compare”中解释。(如果驱动器以前压缩过，只有在装入压缩软件并压缩这个驱动器后才能把所有旧文件装入)。

4.6 SETUP 对系统的影响

SETUP 程序用新版本的文件代替了 C:\DOS 目录下的旧文件，如果可能，它将把所有的旧文件移到一个名为 OLD_DOS.n 的目录中；保留这些文件以便在必要时恢复安装这些文件。在 C:\DOS 下没有升级的文件未作任何改动，以保证碰巧装入 C:\DOS 中的任何文件不受损害。

另外，DOS 6 已抛弃的一些旧的 DOS 程序，如 RECOVER 和 ASSIGN 等在 DOS 目录中将继续存在，因为升级操作并没有覆盖掉它们。DOS 6 有很好的理由取消这些程序。大部分有缺陷的程序将被废除，而被更好的代替。在使用新版 DOS 一段时间后，如果确定还将继续使用它，可从 C:\DOS 目录删除下列文件，因为新版的 DOS 不需要它们。

4201.CPI
4208.CPI
5202.CPI
ASSIGN.COM
COMP.EXE
DOSSHELL.*（如果使用 Windows 或其他图形界面的程序）
EDLIN.EXE
EXE2BIN.EXE
GRAFTABL.COM
JOIN.EXE
LCD.CPI
MIRROR.COM
MSHERC.COM
PRINTER.SYS
RECOVER.EXE
SMARTDRV.EXE
DBLSPACE.*（如果不使用 DoubleSpoce）的话

4.7 查看附加资料

在 DOS 目录下，有一些.TXT 文件提供了用户指南。应该查看一下每个.TXT 文件，看看是否包含有关你的系统的东西。

浏览.TXT 文件：

1. 在 DOS 命令提示符下输入下面命令：

   DIR C:\DOS*.TXT /0D

   这个命令将按时间顺序列出 DOS 目录下的所有.TXT 文件。如果有日期为 1994 年 5 月以前的文件，它们对 DOS 6.22 来说业已过期，不再适用，可删除或忽略它们。

2. 如果想读某个文件，按以下步骤操作：

a. 输入下面的命令，这里 filename 是想读的文件的名字，如 README. TXT 或 NETWORKS. TXT：

```
EDIT C:\DOS\filename. TXT
```

这个命令将在 DOS 的编辑器中打开想读的文件。

b. 按 PgDn 和 PgUp 键浏览文件。

c. 如果要打印文件，应按住 Alt+F，再击键 P。（参考第 19 章“DOS Editor”中关于文件打印的那一部分）。

d. 按住 Alt+F，再击键 X 退出编辑器。

4.8 搜寻病毒

升级到 DOS 6.22 后，为什么不使用一下它的最有力的新工具 Microsoft Anti-Virus 呢？如果从未在系统中搜寻过病毒，现在可以立刻做这样事，即使你的系统是新买的。（如果经常定期地使用反病毒程序，现在可以放心地跳过本节）。在第 25 章“DOS 的反病毒程序”中可以更多地了解关于如何使用 Microsoft Anti-Virus。

使用 Microsoft Anti-Virus 的命令行版本搜寻病毒：

1. 在 DOS 命令提示符下，键入下列命令，并回车。

```
MSAV /L /C /P /F
```

这个命令可搜寻除软驱外任何本地驱动器上的病毒。如果发现病毒，它们将被清除。最后显示一个报告。

2. 如果报告显示探测到病毒，关闭系统，马上阅读第 25 章。

使用 Windows 版的 Anti-Virus 搜寻病毒：

1. 启动 Windows。
2. 打开 Microsoft Tools 组，将出现 Anti-Virus (Microsoft Windows Anti-Virus)图标。

注释： 如果找不到 Microsoft Tools 组或 Anti-Virus 图标，说明没有安装 Windows 版的 Microsoft Anti-Virus。

3. 打开 Anti-Virus 程序。（Anti-Virus 窗口将打开）。
4. 选择所有非软驱的本地驱动器。
5. 按 Detect 和 Clean 按钮。
6. 如果检测到病毒，按 Stop 按钮，退出 Anti-Virus，关闭正在运行的其他程序，退出 Windows，关闭系统，马上阅读第 25 章。

4.9 启动 Delete Sentry

DOS 6.22 还包含另外一个新功能，如果不曾使用过它，应马上安装。Delete Sentry 可使删除的文件保存几天时间，这样能很容易地取消删除。Delete Sentry 将在第 26 章“UN-

DELETE 的使用"中解释，但现在应该先将它建立起来(如果从没使用过的话)。

从 DOS 命令提示符下建立 Delete Sentry：

1. 键入下面的命令，并回车：

   EDIT C:\AUTOEXEC.BAT

   DOS 的编辑器打开这个文件以备编辑。

2. 按向下箭头键将光标移到文件的第一个空行处。

3. 在空行键入下面的命令：

   UNDELETE /SC

   这个命令将启动 Delete Sentry 用来保护驱动器 C，如果有多个硬盘，为每个驱动器加 /Sdrive 选项。例如，如果有驱动器 C、D 和 E，命令如下：

   UNDELETE /SC /SD /SE

4. 按 Alt+F，按 X，然后按 Y。此按键序列将退出编辑器并保存对 AUTOEXEC.BAT 所作的修改。

5. 移去驱动器 A 中的软盘，按 Ctrl+Alt+Del 重新启动系统，启动时将安装 Delete Sentry。

从 Windows 中建立 Delete Sentry。

1. 启动 Windows。

2. 打开 Microsoft Tools 组窗口(如图 4.1 所示)。Undelete 图标出现在该窗口中。

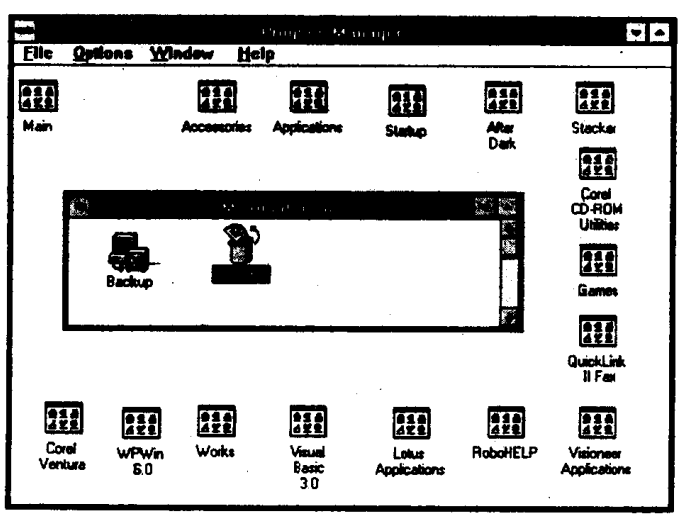

图 4.1 Microsoft Tools 组窗口

注释： 如果不能发现 Microsoft Tools 组窗口或 Undelete 图标，说明没有安装 UNDELETE 的 Windows 版。

3. 打开 Undelete。Microsoft Undelete 窗口将打开(如图 4.2 所示)。

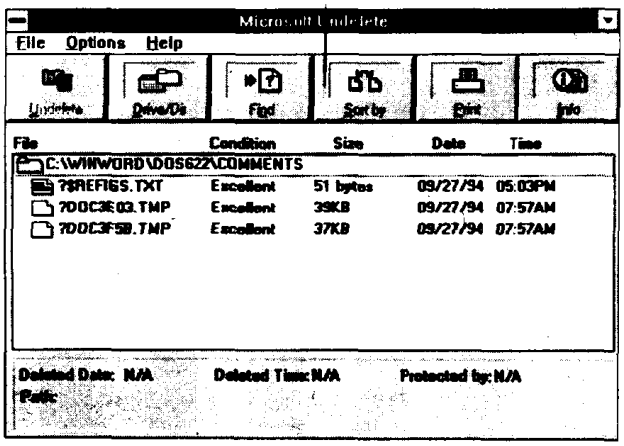

图 4.2 Undelete 窗口

4. 拉下 Options 菜单，选择 Configure Delete Protection 项。Configure Delete Protection 对话框打开(如图 4.3 所示)。

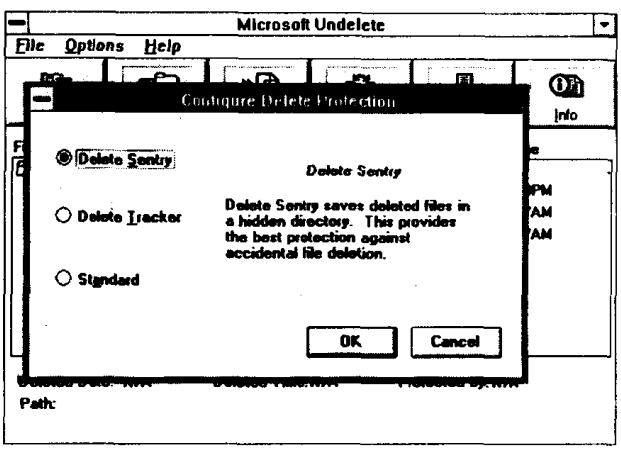

图 4.3 Configure Delete Protection 对话框

5. 单击 Delete Sentry 单选按钮，然后单击 OK。这一操作将打开 Configure Delete Sentry 对话框(如图 4.4 所示)。
6. 单击 Drives 按钮，打开 Choose Drives for Delete Sentry 对话框(如图 4.5 所示)。
7. 选择所有硬盘并按几次 OK 回到主 Microsoft Undelete 窗口。
8. 退出 Undelete，关闭其他程序，退出 Windows。
9. 移去驱动器 A 中的盘，按 Ctrl+Alt+Del 重新启动 DOS。启动时将安装 Delete Sentry。

安装了 Delete Sentry 之后，取消删除工作将变得很容易，第 26 章将介绍怎样取消删除。

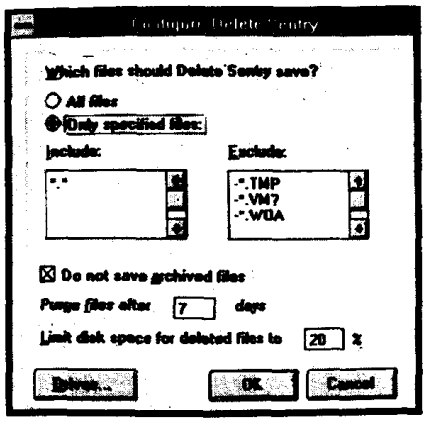

图 4.4 Configure Delete Sentry 对话框

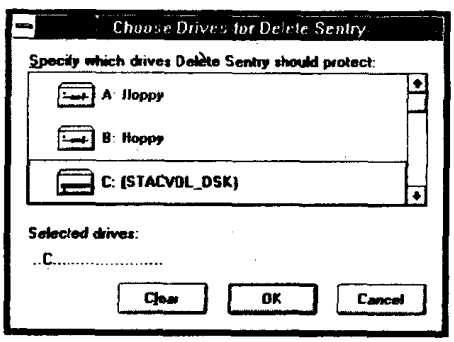

图 4.5 Choose Drives for Delete Sentry 对话框

4.10 制作一张紧急启动盘

新版 DOS 成功地运行后，还需要为它制作一张紧急启动盘。依照本章前面“准备安装”一节的第 3 和第 4 步可制作此盘，可重新使用以前制作的紧急启动盘，因为现在已不再需要它了。

4.11 取消安装 DOS 6

除非硬盘被一个第三方压缩程序压缩过，否则 SETUP 会记录取消安装信息，这样仍可使用原来的 DOS 版本，如果某个应用程序在新版本下不能运行，并且 SETVER（在第 21 章“系统配置”中解释）也不能解决这个问题，就必须回到从前的版本来解决这个问题。（也可能不得不把这个应用程序升级到与 DOS 6.22 兼容）。

注释： 如果自从安装新版 DOS 到现在已重新分区或重新格式化过启动驱动器，将不能取消安装 DOS。如果已删除或移动过 DOS 的两个核心程序文件（MSDOS.SYS 和 IO.SYS），删除了 OLD_DOS.n 目录，运行过 DELOLDOS，安装了 DriveSpace 或 DoubleSpace，或者安装了一个第三方压缩程序，也不能取消安装 DOS。

取消安装 DOS 6.22：

1. 插入取消安装盘到驱动器 A 中。
2. 重新启动。
3. 按照屏幕列出的指示进行操作。

4.12 DELOLDOS

当你确定想保存新版 DOS 时，名为 DELOLDOS 的程序可从硬盘中删去从前版本的 DOS，这将释放几 MB 的空间。输入以下命令即可完成删除工作：

```
DELOLDOS
```

警告： 运行 DELOLDOS 之后，将不能再取消安装 DOS 6.22。

DELOLDOS 只是从硬盘中删除了所有的 OLD_DOS.n 目录。（若没有 DELOLDOS，也可做这件事。只需使用 DELTREE OLD_DOS.1 命令，如果必要，再使用 DELTREE OLD_DOS.2 等等）。

4.13 使用 Expand

如果没有全部安装 DOS 6.22，或者安装后删除了一些文件，可能需要从安装盘拷贝一些文件到硬驱。盘上的大部分文件压缩过，在使用之前必须先将它们进行扩展（解压缩）。（这些文件不是使用 DoubleSpace 或 DriveSpace 压缩的，它们使用一种不同的压缩技术）。压缩文件扩展名的最后一个字符后有一下划线。DOS 6.22 中的 EXPAND 程序用来对这些文件解压缩（如图 4.6 所示）。例如，从驱动器 A 上扩展 MSAV.EX_ 到 C:\DOS 中，可用以下命令：

```
EXPAND A:\MSAV.EX_ C:\DOS\MSAV.EXE
```

用户可指定对多个的文件解压缩，但这时的文件名不能是全程文件名。如果指定多个文件名，目标（destination）一定不能包含文件名，EXPAND 将使用压缩文件名（包括下划线）来命名解压缩后的文件，用户必须在硬盘上为各个文件重新命名。

EXPAND

扩展压缩文件：

句法：

```
EXPAND [filespec ...[destination]]
```

参数：

none	提示输入 filespec 和 destination。
filespec	指定一个要扩展的压缩文件。
destination	指定扩展后的文件的位置和/或名字。

图 4.6 EXPAND 命令总结

第二部分　DOS Shell

第 5 章　DOS Shell 概述

简介： DOS Shell 是 Windows 的一个不太好的替代品，但如果没有 Windows，使用 Shell 比用 DOS 命令提示符更直观更容易。

5.1　决定什么时候使用 Shell

DOS Shell 是 DOS 的图形用户界面(GUI)。通过它，用户能看到正在工作的驱动器和目录，能更容易地管理磁盘、目录和文件，还可启动如 DOS Editor 之类的程序或游戏等。通过使用鼠标或键盘，选择清单中的项，从菜单中选择命令，从对话框中选择选项来与 Shell 进行交互，甚至可将文件从一个地方拖到另一个地方来移动或拷贝它们。

也许 Shell 最重要的特点是任务切换(task swapping)，通过它，用户能同时运行几个程序并在它们中转移。当你在编辑一个文档时要查询电子表格数据，或者在保存花了两个小时创建的图形之前想在硬盘上腾出一些空间时，这个特点可节约许多时间和麻烦。

当然，如果有一个更精致的 Shell 如 Norton Desktop for DOS 或一个真正的多任务程序如 Windows 时，就不再真正需要 DOS Shell。如果已有一个这样的程序，可直接跳到本书的第三部分。

如果没有一个更精致的程序，就应该使用 Shell 来完成大部分 DOS 任务。本书的后续章节，将介绍如何在命令提示符下完成在 Shell 下不能完成的工作。

在本章中，主要讲述：

- 启动和退出 Shell
- 用 Shell 屏幕的不同部分工作
- 当用 Shell 工作时获得屏幕联机帮助
- 在一个清单中滚动
- 操作菜单和对话框

5.2　启动 Shell

在启动 Shell 之前，应该处于 DOS 命令提示符下。如果还没有启动，现在启动。如果系统已启动，退出正在运行的任何程序，如字处理程序或电子表格软件。

在启动或退出所有的程序后，应该看到如下所示的命令提示符(可能是不同的字母)：

```
C:\>
```

如果不是这样，说明有人已修改了系统。若要达到命令提示状态，要与在该机器上工作的人联系，找出原因以决定如何做。

启动 Shell：

1. 在命令提示符下键入下面的命令：

DOSSHELL

2. 按回车(Enter)键。

一个如图 5.1 所示的屏幕将会出现(内容可能有所不同，但通常外观是相同的)，还可以看到一个 DOS 正在读目录的消息。当 DOS Shell 准备就绪时消息消失。

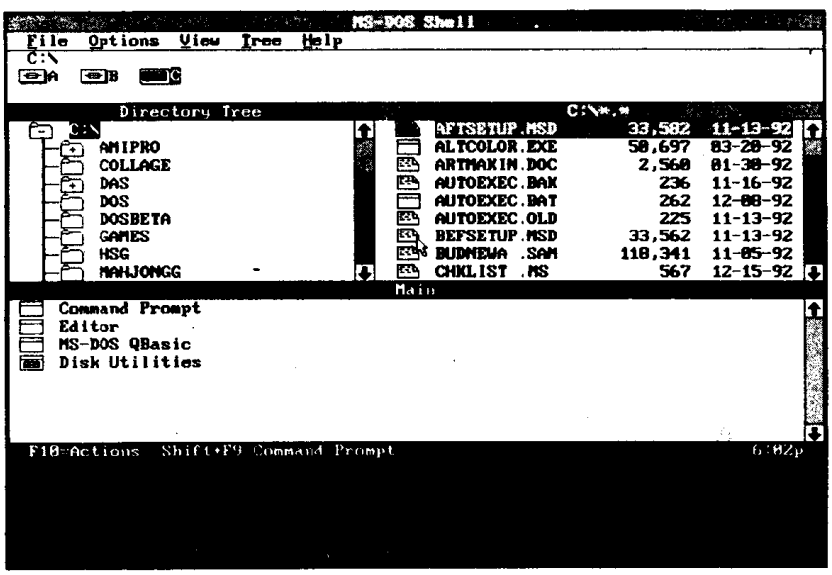

图 5.1 DOS Shell 屏幕的一个例子

注释： DOSSHELL 命令包含几个参数，大部分人从不使用。DOSSHELL 完整的语法格式在本书后面的命令用法参考中给出。

答疑： 输入 DOSSHELL 命令时，得到消息 Bad command or file name。

因为 Windows 变得很流行，Shell 就变得不像以前那么必要了，从 DOS 6 开始，DOS 不再继续提供 Shell。如果私自安装 DOS 6.2 或 6.22 而不是升级，就没有 Shell 软件。《DOS 用户指南》将提供关于订购或卸载 Shell 程序文件的指导性建议。

如果从 DOS 4.0 到 6.0 的任何版本升级，应该有 Shell。试着再次输入 DOSSHELL 命令并保证拼写正确(注意有两个 S)。

如果保证拼写正确而 DOS 还是不能发现 Shell，输入 MEM 命令再按回车键。不要担心显示的消息；在第 29 章“内存管理”将介绍 MEM。现在只是想看看它是否能正常工作。

如果 MEM 仍产生消息 Bad command or file name，说明 DOS 不能发现 DOS 目录。在第 12 章“在命令提示符下工作”中将介绍如何处理这样的问题。这时，使

用以下命令启动 DOS Shell。(如果没有安装 DOS 到驱动器 C 的一个名为 DOS 的目录下,你应该用正确的名字代替它,使 DOS 处于 C:\DOS 中)。

```
C:\DOS\DOSSHELL
```

如果 MEM 运行正常,但 DOSSHELL 不能运行或者以上命令也失败,这可能是因为有人从你的系统中删除了 Shell,这时可从 Microsoft 获得免费拷贝。

答疑: 当 DOS Shell 启动后,不能读屏幕。

正在使用的显示器可能不能显示缺省的 Shell 屏幕,它需要一个彩色图形监视器,按 F3 退出 Shell 回到命令提示符。

用黑白取代彩色启动 Shell 用以下命令:

```
DOSSHELL /B
```

用文本模式代替正常图形模式,用以下命令:

```
DOSSHELL /T
```

必要时可同时使用/B 和/T:

```
DOSSHELL /B /T
```

如果不得不使用文本模式,屏幕内容就与本书给出的图示有所差别。例如,它不会有图标(表示驱动器、目录和文件的小图形)。但是,它与图形模式很接近,通过本书中的例子学会它不会有任何麻烦。

第 11 章"自定义 Shell"将介绍如何调整 DOS Shell,这样就不必使用/B 和/T 了。

提示: 如果打算始终使用 DOS Shell,可以把 DOSSHELL 命令作为 AUTOEXEC.BAT 文件中的最后一个命令,以便每次启动时自动运行它。第 20 章"基本批处理程序"解释了 AUTOEXEC.BAT 更多的信息以及如何在其中插入命令。

5.3　探索 Shell 屏幕

当使用 DOS Shell 时,Shell 屏幕就是主要的工作区域。顶部的标题栏(如图 5.2 所示)提醒你正在使用 DOS Shell。其下是菜单栏,本章稍后将介绍如何使用。底部的状态栏显示消息帮助执行当前任务。在本章后面学习使用 Shell 功能时会学到更多这类消息。

屏幕体包含四个主要清单(参见图 5.3):

- 驱动器清单显示了所有可用的磁盘驱动器,包含 RAM 驱动器和登录的网络驱动器。如果正在使用一个数据压缩系统,如 DriveSpace,它显示压缩驱动器和宿主驱动器。高亮度标志的是当前选择的驱动器(图 5.3 中为 C)。在图形模式下,每个图标表示出了驱动器的类型。在文本模式下,驱动器名出现在方括号中,无法表示驱动器类型。
- 目录树显示所选驱动器的目录结构。在这个例子中,根目录为 C:\,它的子目录是 AMIPRO,COLLAGE,DAS,DOS 等等。高亮度表示当前目录,在本例中是 C:\。

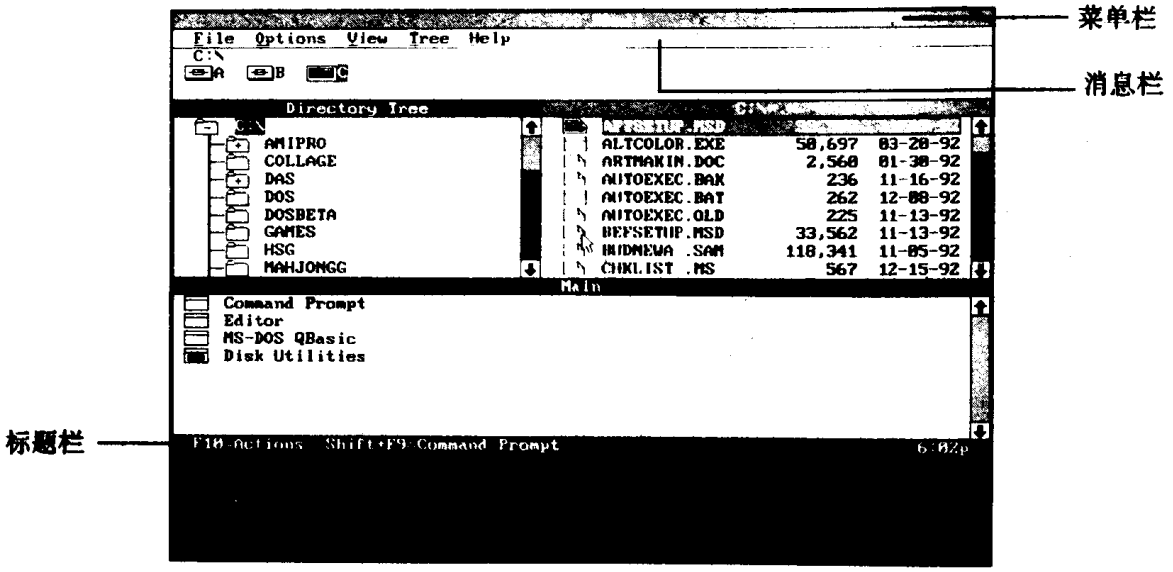

图 5.2 Shell 屏幕上的三个栏

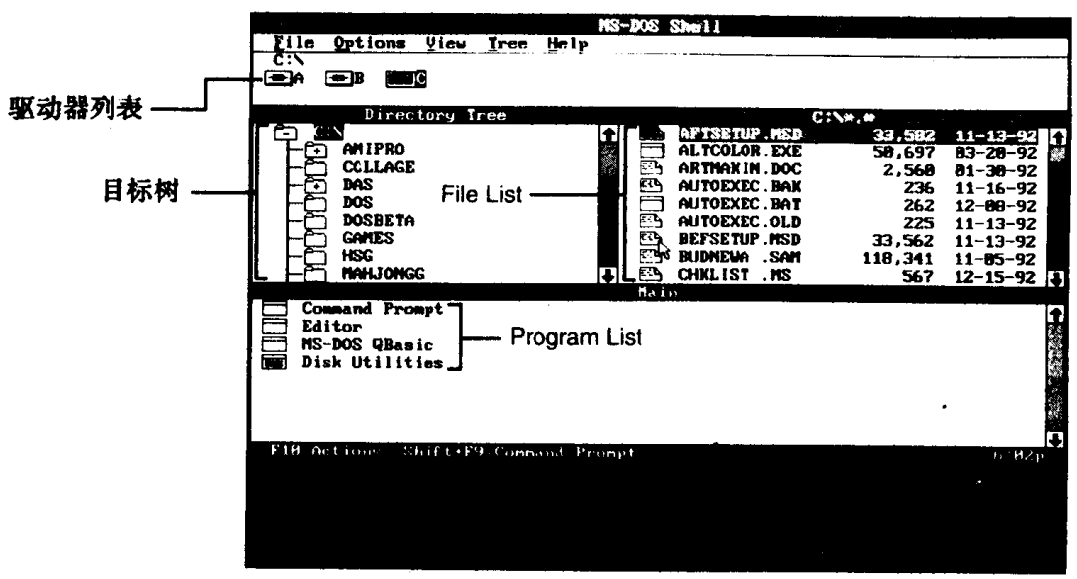

图 5.3 DOS Shell 的清单

注释： 在缺省状态下，目录树只显示第一级子目录。标有加号(+)的子目录还有没显示出来的子目录，第 6 章"目录管理"中将解释如何显示它们。

文本模式下的目录树与图形模式下的树很相象，只是文件夹图标处出现的是方括号。

- 文件清单显示当前选择目录中的文件。图形模式下的文件图标表示文件(程序或数据)的类型，还有文件名，它的大小及最后一次修改的日期，高亮度表示当前选择的文件，在图 5.3 的例子中，如果立即选择了 File Delete 命令，将删掉文件 AFTSETUP.MSD 文件。文本模式下不显示图标，用箭头指示选择的文件。

- 程序清单显示了只要从这个清单中选择即可运行的程序。例如，从图 5.3 的清单中选择 Editor，即可运行 DOS 的全屏幕编辑器。这在 DOS Shell 下通常是一个最容易的运行程序方法。

这些区域中总有一个处于激活状态，意指它必须包含光标或焦点。我们使用焦点(focus)这个词汇是因为激活的清单是所运行的任何命令的集中点。例如，假设按下 Delete 键。如果焦点在文件清单中，将删除一个文件；但如果焦点在目录树中，将删除整个目录，用户可用键盘或鼠标移动焦点。

如何在 DOS Shell 中移动焦点
用键盘移动焦点：按 Tab 键或 Backtab(Shift＋Tab)键直到到达想要的清单。 用鼠标移动焦点：单击想要的清单。

从键盘上使用 Tab 键移动焦点：

1. 看着屏幕按几次 Tab 键。
   焦点从一个清单移到另一个清单。当一个清单包含焦点时，其标题栏高亮显示，驱动器清单例外，它没有标题栏。
2. 按 Shift＋Tab(也叫 Backtab)几次。
   焦点又移动了，只是方向相反。

使用鼠标时，就不用考虑那么多了。当想选择一个项目时，单击它，焦点就自动移到那里。

试试用鼠标移动焦点：

1. 在驱动器清单中单击 C：
   焦点移到驱动器清单，驱动器 C 呈高亮度，如果在单击之前驱动器 C 不是当前驱动器，目录树和文件清单会随驱动器的更新而更新。
2. 在目录树中单击 DOS。
   焦点移到目录树，DOS 变成当前选择的目录。如果在单击之前 DOS 不是当前目录，文件清单将改变来反映新的目录。
3. 单击文件清单中的 ANSI.SYS。
   焦点移到文件清单，ANSI.SYS 成为选择的文件。
4. 在程序清单中单击 Editor。
   焦点移到程序清单，Editor 变成选择的程序。(这不会启动所选程序。在第 10 章“从 DOS Shell 运行程序”中介绍如何启动)。

试着作一些其他的焦点移动工作，但不要改变驱动器。下一章将告诉你怎样做。

5.4　使用 DOS Shell 的屏幕帮助系统

假设你正在做一项 DOS Shell 任务，但不知道下一步做什么。可以查询本书，但这里还

有一个更快的方法。

1. 按 F1 键。

   DOS Shell 打开一个对话框显示当前包含焦点的清单的帮助信息。

2. 选择 Esc 关闭 Help 对话框。

按 F1 键请求上下文帮助，意指 DOS Shell 在选择一个帮助题目时考虑焦点的位置或上下文。它能反应任何主要清单、菜单上的一个命令或一个对话框。

5.4.1 对话框

DOS Shell 使用对话框显示不是清单部分的信息。当一个对话框打开时，必须对它作出反应。如果不结束或取消对话框，将不能拉下菜单，也不能从主 Shell 屏幕选择项目，或完成其他活动。要对对话框作出反应，必须使用下一节介绍的命令按钮。

5.4.2 命令按钮

每个对话框至少有一个命令按钮，它可使一些活动发生。在你学习如何使用 Help 时将学习五种帮助命令按钮(如图 5.4 所示)。

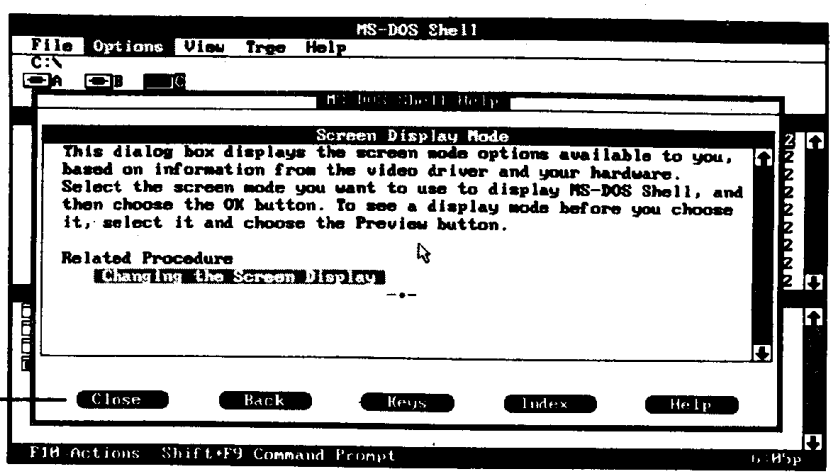

图 5.4 帮助对话框中的命令按钮

用键盘选择命令按钮，按 Tab 键或 Shift+Tab 键直到光标出现在按钮上(看起来像下划线)，然后按回车键。

提示： 根本不需要在 Help 对话框中选择 Close 按钮，因为只按一下 Esc 就可做相同的事。

用鼠标选择命令按钮，只需单击。

5.4.3 滚动

许多 Help 题目内容太长以致对话框不能全部显示，必须滚动才能看到全部内容。在框中滚动会移动文本。向上滚动会移向题目的首部，向下滚动则会移到题目的尾部。

如何用键盘滚屏

键	功能
up-arrow	光标上移一行
down-arrow	光标下移一行
Home	光标移到第一行
End	光标移到最后一行
PgUp	光标上移一页
PgDn	光标下移一页
字符	将光标移到这个字符开始的下一项的位置

用鼠标滚动需要滚动条(看下面过程中的图形,滚动条上滑动方块滚动框)总会显示当前页在整个清单中的位置(即使用键盘滚动时也是这样)。用鼠标单击滚动条的端点一次上滚动一页(一页由在可视区域看到的若干行组成)。如果想滚动更长的距离,可拖动滚动框直到想看的页。在箭头图标或滚动条上按住鼠标键,滚动条会重复迅速向一定方向滚动,与按住光标键的情形很相像。

注释: 水平滚动条与垂直滚动条的工作方式一样,只是滚动的方向是左右而不是上下。

用键盘和鼠标滚动之间有明显区别。当用键盘滚动时,光标在清单中上下移动,改变着选择的项,但当使用鼠标时,光标停留在它所在的那个项目上;当项目移动时,光标可能卷出可视域。用鼠标滚动从来不会改变被选择的项目;必须单击另一项来改变选择项,而有时不想这样做,当习惯于这个不同后,选择鼠标还是键盘将取决于哪个对你更方便了。

如何用鼠标滚动

当有太多的信息出现在使用的清单中时,会出现一个滚动条。必须用鼠标使滚动条滚动(图 5.5 所示)。

让我们看一看标题为 File List Overview 的完整题目:

1. 激活文件清单。
2. 按 F1 键打开 Help 对话框,名为 File List Overview 的题目就出现在对话框中。
3. 按向下箭头键几次,向下滚动几行。
4. 按 PgDn 几次,向下一次滚动一页。
5. 按 End 滚动到题目的结尾。
6. 现在试一试 up-arrow,PgUp 和 Home 键。

注意滚动框在滚动条内上下移动来显示你正在看的题目的哪一部分。

如果有一个鼠标,用它来滚动题目:

1. 在滚动条的底端单击 down-arrow 键会向下滚动一行。
2. 单击滚动条中滚动框下面的黑色部分会向下滚动一页。
3. 拖动滚动框到滚动条的底部。
4. 现在试试几种向上滚动的方法。

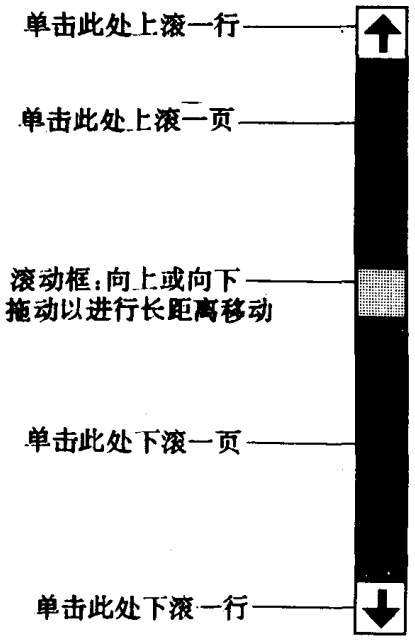

图 5.5 滚动条的解析

5. 按 Esc 键结束 Help。

提示： 如果按住向下滚动键，或者在单击滚动条时按住鼠标按钮，动作会迅速重复，使文本在窗口中滚动，当看到想要阅读的区域后，释放键或按钮。

5.4.4 使用 Help 菜单

有些时候用户想自己选择帮助题目，而不是让 DOS Shell 做这件事，Help 菜单可帮助你访问整个帮助库(参见图 5.6)。

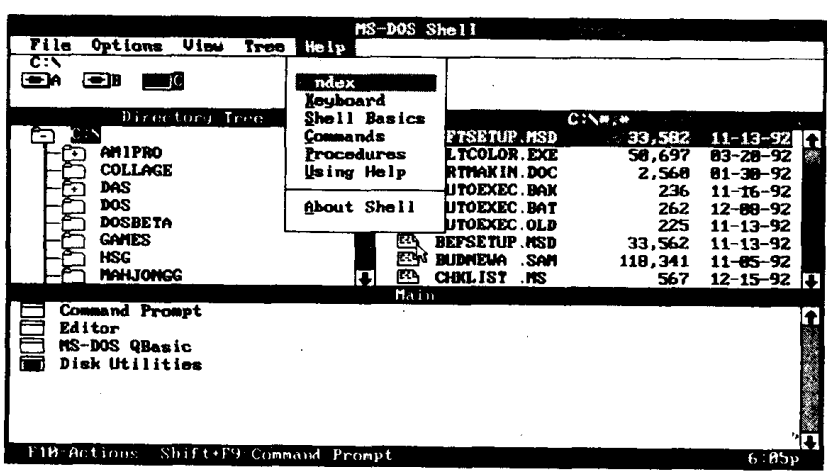

图 5.6 Help 菜单

如何使用菜单

从键盘上：

- 通过按 Alt＋letter 拉下菜单，letter 是菜单名的第一个字母。
- 当菜单拉下后，用左右箭头键可拉下邻接菜单。
- 按一个命令名的下划线字母键可选择这个命令。
- 不选择命令可按 Esc 键关闭这个菜单。
- 如果命令旁有一个快捷键，选择该命令时无需拉下菜单，按快捷键即可。

从鼠标上：

- 在菜单栏上单击名字就可拉下菜单。
- 单击命令名选择一个命令。
- 再次在菜单栏上单击菜单名关闭该菜单。

菜单上一些命令可能很模糊(呈灰色)。这种命令表示它们不能使用，因为使用它们的一些条件还不具备。

必须拉下 Help 菜单才能使用它。用键盘拉下一个菜单，按 Alt＋letter，其中 letter 表示菜单名的第一个字母。通过用鼠标单击菜单名也可拉下菜单。

试着用键盘拉下 Help 菜单：

1. 按 Alt＋H 键，Help 菜单出现。
2. 按 Esc，Help 菜单消失。

也可用鼠标拉下这个菜单：

1. 在菜单栏上单击单词 Help。Help 菜单出现。
2. 在菜单栏上再单击 Help 一次。Help 菜单消失。

提示： 单击菜单外的任何地方将可关闭一个菜单。但是，因为单击时可能会选择其他的项目；要养成第二次单击这个菜单的安全习惯。

5.4.4.1 Help Index

应该从 Help 索引开始探讨 Help 系统(如图 5.7 所示)，它为所有题目提供一个通用索引。它把题目分组成几个类，如键盘帮助、命令帮助、过程帮助等，甚至还有一个如何使用 help 系统自身的类别，这个索引中的许多题目还有更详细的索引。

用键盘打开 Help Index：

1. 按 Alt＋H 拉下 Help 菜单。
2. 按 I 选择 Index。

   你通过击菜单中项目名中带下划线字母来选择一个命令项，I 是 Index 中的下划线字母。

滚动索引可看到它包含的所有分类和题目。题目用不同的颜色显示。要选择一个题目，可用鼠标双击或按 Tab 键使光标在其上，再按回车键。

1. 滚动索引直到看到题目 General MS-DOS Shell Keys。
2. 按 Tab，直到 General MS-DOS Shell Keys 高亮显示。

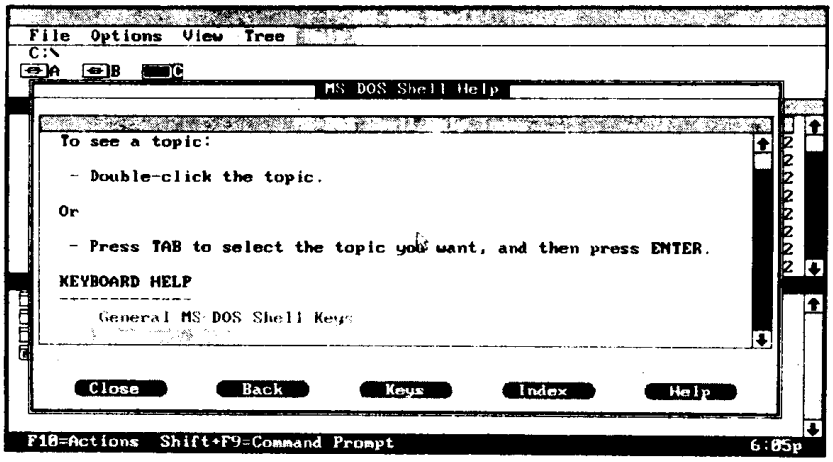

图 5.7 Help 索引

注意光标从帮助题目移动到命令按钮，然后再返回。

3. 按回车键查看你选择的题目。

题目列出在 DOS Shell 中使用的各种不同的键。滚动文档看看它包含些什么。看完后，可选择 Index 命令按钮回到 Help Index。这个命令按钮在每个 Help 对话框中都有，总可通过它回到索引下。

1. 按 Tab 直到光标处于 Index 命令按钮。
2. 按回车键回到 Help Index。

现在试着用鼠标完成前面两个过程：

1. 双击 General MS-DOS Shell Keys 打开这个题目。
2. 单击 Index 按钮回到 Help Index。

帮助系统跟踪查看的窗口，可选择 Back 按钮查看前面看过的内容，Back 按钮出现在每个窗口中。

1. 选择 Back 回到前面的题目。(凭你的喜好使用键盘或鼠标)。
2. 再选择 Back。
3. 按 Esc 或选择 Close 关闭 Help 对话框。

注释： 不幸的是，在 Shell 屏幕上工作时不能在屏幕上保持帮助窗口。

5.4.4.2 其他帮助选项

在 Help 菜单顶部的其他命令也可访问部分 Help Index。例如，如果想查看 Keys Index 部分，无需经过 Help Index 就可直接到达目标。具体索引如下：

Keyboard	列出了关于在 DOS Shell 中如何使用键盘的题目，如 Movement keys 和 File List Keys。这个 Keys Index 也可通过在 Help 对话框中按 Keys 命令按钮访问。
Shell Basic	列出了关于使用 Shell 的通常题目，如 Dialog Boxes 和程序清单。

Commands	列出了关于 Shell 菜单的命令和题目，如 File 菜单和 Tree 菜单。
Procedures	提供了在 Shell 中如何做事情的一步一步的步骤。如可以看到如何使用文件清单和如何运行程序。
Using Help	列出了关于帮助系统本身的题目，如 Using Help Buttons 和 Using Help Menu Commands。可从 Help 对话框按 Help 按钮访问相同的索引。

5.4.4.3 About Shell(关于 Shell)

About Shell 命令出现在 Help 菜单的区分线的下面，这是因为它和其他 Help 选项完全不同。选择这个命令会打开一个提示框显示当前的 DOS 版本号。如果打电话到 Microsoft 要求关于 DOS 的技术支持，可能会需要这个版本号。

1. 选择 Help 的 About Shell 看看 DOS 版本号。
2. 按 Esc 关闭 Help 对话框。

5.4.5 退出 DOS Shell

退出 DOS Shell 分为临时退出和永久退出两种。临时退出可回到 DOS 命令提示符状态，这样可以在不关闭 Shell 的情况下运行一些命令；当回到 Shell 时，屏幕与离开时完全一样。永久退出也会回到 DOS 命令提示符状态，但它关闭了 Shell，如果重新启动 Shell，将回到它的最初状态。

按 Shift＋F9键作一个临时退出回到命令提示符状态(注意状态栏的提示)，必须使用 EXIT 命令再次返回 Shell。

1. 按 Shift＋F9键。
   命令提示符在屏幕上出现。
2. 键入单词 EXIT，回车。
   重新回到 Shell 屏幕。

注释： *如果在命令提示符下工作时对驱动器作了一些变动，从 DOS 提示符回到 Shell 中时可能需要更新驱动器。*

按 F3键或使用 File Exit 命令将永久退出 Shell，按 F3键让 Shell 退出系统。

5.5 自 测 练 习

在以后章节的学习中和每天的工作中将获得许多启动和退出 Shell 的实践。除了在实际工作中需要使用 Help 系统外，一般不会有太多的机会实践一下使用 Help 系统。这就得靠你自己去继续探索 Help 系统，以便在需要时知道在哪里找到想要的题目，建议作以下练习：

- 在 File 菜单中找到并阅读关于 Exit 命令的内容。
- 找到并阅读关于如何在 DOS Shell 中使用鼠标的内容。

第6章 目录管理

简介： Shell 的目录树使创建，删除及其他目录管理变得很简单。

6.1 目录管理

Shell 通过将目录显示在 Shell 屏幕上使管理驱动器和目录更容易。用户能很容易地选择要处理的项，然后从 Shell 的菜单中选择命令对之进行管理。更好的是，甚至一些功能（如删除一个目录）可用快捷键来完成，完全避免了使用菜单的麻烦。

在本章中，主要讲述：

- 选择一个驱动器使之成为当前驱动器
- 使 DOS Shell 重读当前驱动器
- 创建一个新目录
- 选择一个目录使之成为当前目录
- 重新命名一个目录
- 管理目录树
- 删除一个目录

6.2 选择一个驱动器

驱动器清单显示了所有驱动器，包括网络驱动器和虚拟驱动器，如 RAM 驱动器（图标显示了相应驱动器的类型）。只须在驱动器清单中选择便可转换到另一个驱动器。

如何在 DOS Shell 中改变驱动器

用键盘选择一个驱动器：

1. 激活驱动器清单。
2. 按左右箭头键使欲选驱动器呈高亮显示。
3. 按回车键选择并更换新的驱动器或者按空格键选择但不更新。

从屏幕的任何区域选择一个驱动器但不更新它：

- 按 Ctrl+letter，如按 Ctrl+A 选择驱动器 A。

用鼠标选择一个驱动器：

- 在驱动器清单中单击驱动器就选择了它，但不更新。
- 否则，在驱动器清单中双击驱动器，这样既选择了它，又更新了它。

更新当前驱动器(不选另一个新驱动器):

· 从 Shell 屏幕的任何地方按 F5键。

无需选择一个新驱动器而重画屏幕:

· 按 Shift+F5。

答疑: 如果只有一个软驱,但 DOS Shell 在驱动器清单中显示一个驱动器 B。
当只有一个软驱时,DOS 把驱动器名 B 作为这个驱动器的一个可选名。(Drive B 称为幻影驱动器(Phantom))。这样就有可能将一个盘里的文件在相同驱动器里拷贝到另一个盘上。第8章"在 DOS Shell 中拷贝和移动文件"将介绍如何使用幻影驱动器拷贝文件。

注释: 如果某一个驱动器不出现在驱动器清单中,这说明 DOS 不能识别它。参考第51章"配置磁盘和网络"中的"DOS 定义驱动器",其中有如何使驱动器能用的建议。

用键盘选择驱动器:

1. 插入一张盘到驱动器 A 中。任何一张已格式化的盘均可。
2. 激活驱动器清单。
3. 按左箭头键直到 A 高亮显示。
4. 按空格键选择驱动器 A。
   目录树和文件清单相应改变以显示新的当前驱动器的内容。
5. 按右箭头键直到驱动器 C 高亮显示。
6. 按空格键选择驱动器 C。

答疑: 当试图转换到软驱时,获得了一个错误消息,即使这时有一个软盘在驱动器中。
可能是软盘没被 DOS 格式化(它可能被其他的操作系统格式化),或者它已被严重破坏以致 DOS 不能再读它。
如果试图读一张别人提供的盘,让他再提供一张,要保证用 DOS 格式化过。
如果试图使用一张新的空盘,必须首先格式化。第17章"格式化磁盘和取消格式化"将介绍如何格式化磁盘。

这里有一个从键盘选择一个驱动器的更容易的方法,所有要做的事就是按 Ctrl 键加上想要用的驱动器名,如 Ctrl+A。试着使些这种捷径选择驱动器:

1. 按 Ctrl+A 选择驱动器 A。
2. 按 Ctrl+C 选择驱动器 C。

提示: 使用 Ctrl+letter 快捷键时不需要激活驱动器清单,可以在 Shell 屏幕的任何地方做这件事,这样便可在文件清单或目录树下改变驱动器。这是在 Shell 下工作时改变驱动器最快的方法。

不要忘记也可用你的鼠标选择驱动器：

1. 单击驱动器 A 选择它。
   注意，当这样作时焦点移到该驱动器上。
2. 单击驱动器 C 选择它。

6.2.1 选择一个空驱动器

如不小心选到一个空驱动器，将看到如图6.1所示的一个错误消息。可以在这个驱动器中放一张盘选择选项1，Try to read this disk again。否则，选择选项2，Do not try to read disk again，然后再选择一个不同的驱动器。

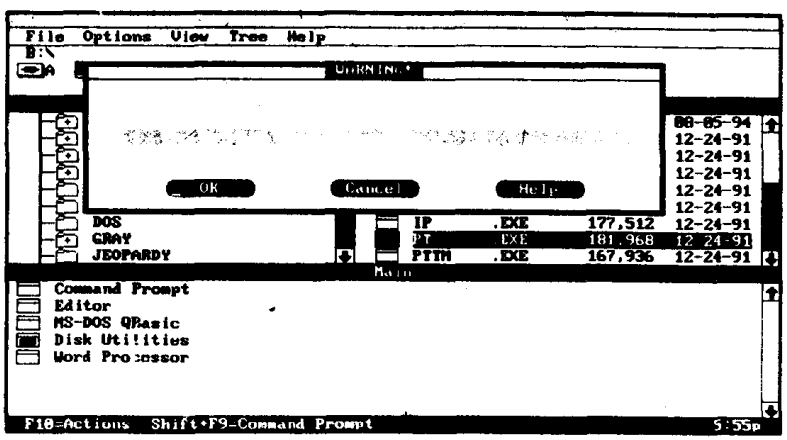

图6.1 Drive Not Ready 警告消息

试一试每个空驱动器选项：

1. 从驱动器 A 中移去软盘。
2. 选择驱动器 A(无论使用哪种技术)。Drive Not Ready 警告消息将出现。
3. 双击选项2或者按下箭头键使选项2高亮显示，然后回车。
   警告消息消失，驱动器 A 成为当前驱动器，尽管它是空的。
4. 再选择驱动器 A。
   警告消息再次出现。
5. 放一张磁盘到驱动器 A 中。
6. 选项1高亮显示。按回车键选择它。或者如果喜欢用鼠标的话，双击它即可。
   驱动器 A 变成当前驱动器。

注释： 当 Drive Not Ready 警告消息出现时，可按 Esc 键(或选择对话框中的 Cancel)以取消驱动器改变，仍停留在当前驱动器。

6.2.2 更新驱动器

启动后第一次选择一个驱动器，DOS Shell 将读这个驱动器的目录结构。这时可能会看到一个消息(如果是一个较快的系统和一张小的盘，这个消息闪现一下，由于很快，所以可能

看不到它)。从这时起,DOS Shell 记住这个目录结构。如果从这个驱动器离开,然后又回到这个驱动器时,目录树和文件清单会立刻出现,因为 DOS 不需要重新读它。

但是如果改变了软驱中的磁盘,需要看到新盘的目录结构时,DOS Shell 继续显示旧的结构。这样就需要强迫 Shell 重读驱动器。这个过程称为更新驱动器。

迄今为止,所学的选择驱动器技术不能更新驱动器。必须按回车键或双击所选的驱动器才能更新。

1. 在驱动器 A 中插入另一张盘。可使用任何格式化过的盘。
2. 用空格键选择驱动器 A(不是回车键)。

   注意目录树和文件清单将显示旧盘的目录结构。
3. 现在用回车键选择驱动器 A。

   DOS Shell 读并显示新盘的目录结构。

现在试一试用鼠标更新驱动器:

1. 再改变驱动器 A 中的磁盘。
2. 在驱动器清单上双击驱动器 A。

   DOS Shell 读并显示新盘的目录结构。

如果想更新当前驱动器,不需要重新选择它。在屏幕的任何地方按 F5键即可。试一试更新当前驱动器:

1. 再改变驱动器 A 的磁盘。
2. 按 F5键

   DOS Shell 读并显示新盘的目录结构。

6.2.3 重画屏幕

有时 Shell 屏幕上会出现一些没有完全闭合的对话框。按 Shift+F5或选择 View Repaint screen 在不更新当前驱动器的情况下,清除屏幕。现在试着按 Shift+F5;将看到屏幕闪灭然后闪亮,好象它被重画一般。

6.3 创建目录

使用 File 的 Create Directory 命令可以在目录树上创建新的目录。无论何时,在选择 DOS Shell 菜单中后跟省略号(...)的命令,如 Create Directory 时,将打开一个对话框。如图 6.2所示即为选择 File 的,Create Directory 命令时打开对话框。

注释: 不能在一个只读驱动器,如 CD-ROM 驱动器上创建新目录。

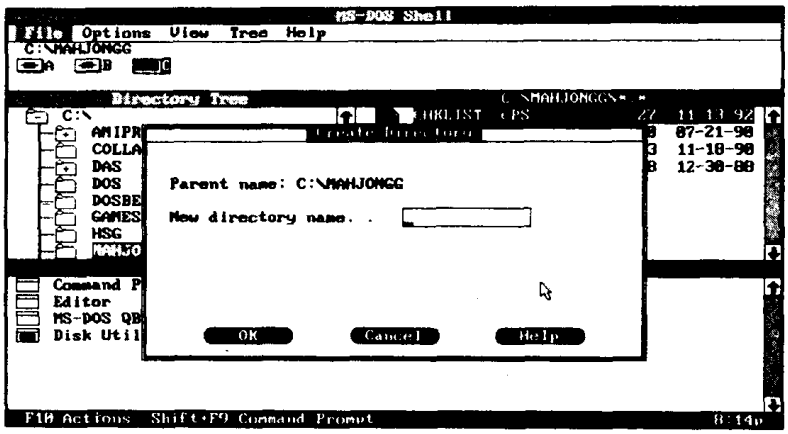

图6.2 Create Directory 对话框

如何在 DOS Shell 中创建一个新目录

1. 激活目录树。
2. 选择想建子目录的父目录。
3. 选择 File,Create Directory。
4. 保证 Parent Name 域正确(如果不正确,选择 Cancel,重新再试)。
5. 在 New Directory Name 文本框中输入想用的名字。
6. 选择 OK。

答疑: 想创建一个新的目录,但 File,Create Directory 命令呈灰色。
这个命令只有在目录树包含焦点时才可用。激活目录树,再试一次。

在 Create Directory 对话框中,标着 Parent Name 的域显示当前选择的目录,它将要成为新的父目录。如果它是一个错误目录,选择 Cancel,选择正确的目录,再试一下。

在对话框中间较小的标着 New Dircetory Name 的框是一个文本框。可在此框中输入内容。当这个对话框首次出现时,光标指向文本框的开始位置,用户要做的事只是将新创建的目录名键入框中。

键入的目录名大小写没有关系。DOS 会自动将它转化为大写出现在目录树中。需要时可使用空格、Home、End、左箭头、右箭头和 Del 键来更正拼写错误。

作者忠告: 目录名要短且有意义
当工作在命令提示符下时,常常需要在文件名前指定一系列目录名,如在 C:\WORK\JUNE\BACKUP\OLDORDER.DAT 中,C:\,WORK,JUNE 和 BACKUP 是目录名,而 OLDORDER.DAT 是文件名。如果你使用带有扩展名的长目录名或用难懂的目录名如 CZ15T25%,将给工作带来困难。而且也可能会创建 DOS 不能处理的太长的表达式。

DOS Shell 对话框中三个命令按钮是最普通的按钮。在文本框中输入名字准备创建新目录时选择 OK;否则,如果改变主意或不想创建目录时,需关闭对话框,选择 Cancel。Help 按

钮的作用与F1键类似，提供关于对话框的帮助信息。

在本章的练习部分，试着创建一个完整的目录分枝如下：

```
PRACTICE
        OTHER
        PLAY
        WORK
                LATER
                NOW
```

现在创建PRACTICE目录：

1. 选择驱动器C。
2. 激活目录树。
3. 按Home键选择根目录。
4. 选择File的Create Directory命令。
   Create Directory对话框出现。
5. 保证Parent Name域是C:\。如果不是，选择Cancel回到步骤3。
6. 在文本框中键入PRACTICE。
7. 选择OK创建新目录。
   PRACTICE目录将出现在目录树中，且按字母顺序排列，如果此时根目录上有许多目录，就可能不得不翻卷后才能看到它。

答疑： 试图创建一个新目录时，得到Access Denied消息。

每个目录名在它的父目录中应是独一无二的。在同一个父目录下不能有相同的目录名或文件名。如果试图分配一个不是独一无二的名字，Access Denied消息就会出现在Create Directory对话框中。

第一个选项(Skip this file or directory and Continue)与Cancel功能相同。它取消Create Directory命令回到Shell屏幕。用户可能想要选择这个选项，在再试之前看看目录树和文件清单。

第二个选项(Try this file or directory again)回到Create Directory对话框，这样可以试试另一个名字。

要选择一个选项，双击该选项或使之高亮显示后按OK。

6.4 选择目录

选择一个目录后，它便成为当前目录且其文件出现在文件清单中。用键盘选择一个目录，只需将光标移在其上。

试着用键盘选择一些目录：

1. 确定已选择了驱动器C。
2. 按Home选择根目录。
3. 按D键选择DOS目录。如果选择了另一个以D开始的目录，再持续按几下D键，直至到达DOS目录。

从一个目录移到另一个目录时，观察文件清单的变化。

如果使用鼠标，只需单击一个目录就可选择它。在单击之前你可能需要滚动屏幕才能发现这个目录。

1. 单击 C:\选择根目录。
2. 单击 DOS 选择 DOS 目录。

现在创建在本章需要的 PRACTICE 子目录的余下部分：

1. 选择 PRACTICE 目录。
2. 选择 File 的 Create Directory 命令。

   Create Directory 对话框出现。
3. 键入 WORK 并按回车。

   WORK 将作为 PRACTICE 的一个子目录出现(可能需要滚动才能看到它)。
4. 重复2和3创建 PLAY 和 OTHER 目录。
5. 选择 WORK 目录在其下创建 LATER 和 NOW 目录。

注意到出现在 PRACITCE 和 WORK 目录旁的文件夹图标的减号了吗?它们将在下节解释。

6.5 剪裁和扩展目录树

用四、五级目录创建了许多文件后，可能会发现在目录树中找东西很麻烦。大部分时间里，如果目录树的较低层子目录不出现，可很容易地找到目录分枝。Tree 菜单可以用来控制出现的目录分枝(见图6.3)。

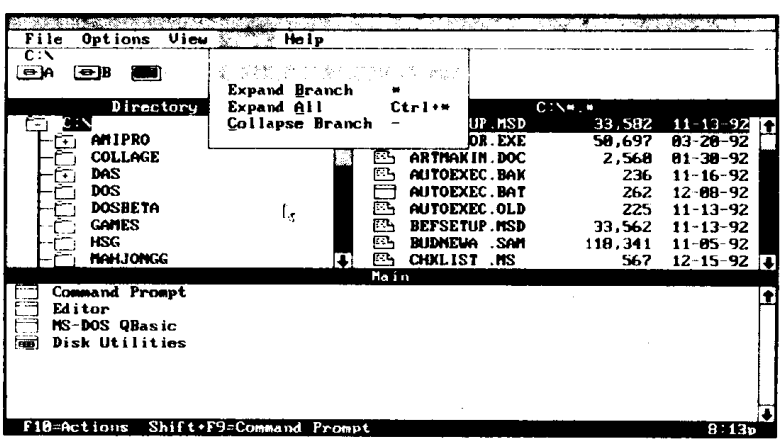

图6.3 Tree 菜单

在目录树中，每个目录用一个形状象文件夹的图标表示。文件夹中的加号表示它是一个带有子目录的目录，可扩展这个目录见到其子目录。文件夹中的减号表示一至少有一个子目录显示的目录，可以通过剪裁隐藏它的子目录。一个空白文件夹表示它没有子目录。

如何扩展和剪裁 DOS Shell 的目录树

当焦点在驱动器清单、目录树或文件清单时，这个过程使用的 Tree 菜单和快捷键工作在当前目录上。

扩展树：

扩展当前目录的第一级子目录：按＋键或单击文件夹图标上的＋号。

扩展当前的所有级子目录：按＊键。

扩展整个目录树：按 Ctrl＋＊。

剪裁树枝：

剪裁当前目录：按－键或单击文件夹图标中的－号。

如果想剪裁 PRACTICE 目录以便不显示它的任何子目录。用键盘按如下步骤进行：

1. 选择 PRACTICE 目录。
2. 按－(减号)键。可使用常规键盘上或数字小键盘上的－键。
   PRACTICE 的子目录消失，文件夹图标中出现加号。

现在假设想看它的第一级子目录：

1. 确认选择了 PRACTICE 目录。
2. 按＋(加号)键。可使用常规键盘或数字小键盘上的＋键。
   OTHER，PLAY 和 WORK 目录出现在 PRACTICE 下面，也许通过滚动屏幕才能看见。

如果想看整个 PRACTICE 目录分枝，使用＊键代替－键：

1. 确认选择了 PRACTICE 目录。
2. 按常规键盘或数字小键盘上的＊键。
   LATER 和 NOW 目录将出现在 WORK 下面，可能必须滚动屏幕才能看到它们。

提示： 扩展或剪裁目录树时无需激活它。当目录树、驱动器清单或文件清单激活时，Tree 菜单和它的快捷键将对当前目录起作用。但当程序清单激活时，Tree 菜单从菜单栏中消失而且它的命令都不能使用。

要扩展整个目录树以便看到驱动器上所有的目录，按 Ctrl＋＊或选择 Tree Expand All。

1. 在常规键盘或数字小键盘上按 Ctrl＋＊。(在常规键盘上，必需按 Ctrl＋Shift＋＊，因为＊是上位键)。
   整个目录树已被扩展。
2. 滚动整个目录树可看到所有分枝。

当使用鼠标而不是键盘时，可通过单击文件夹图标中的＋和－来扩展和剪裁分枝。

1. 单击根目录旁的文件夹中的－。
   整个目录被剪裁，只显示出根目录。＋出现在图标里。
2. 单击＋号。
   在根目录下出现第一级子目录。现在已恢复到目录树的缺省状态。

6.6 目录更名

有时需要为目录更名。可使用 File 的 Rename 命令来完成更名工作。图6.4显示了光标在目录树中时选择 File 的 Rename 命令打开的对话框。当前目录出现在 Current Name 域。

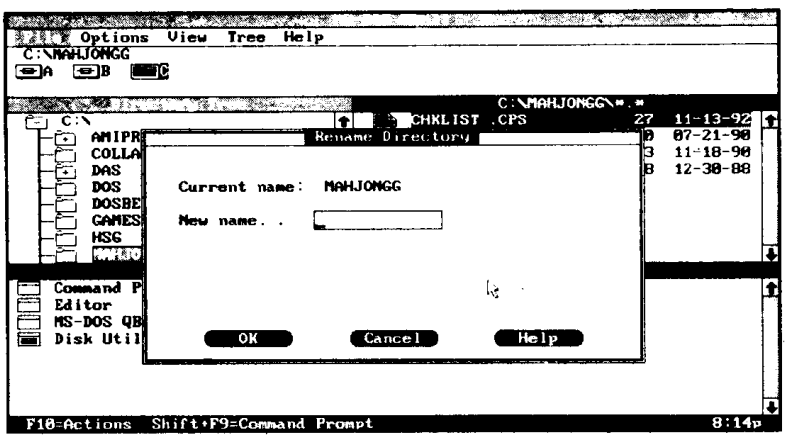

图6.4 Rename Directory 对话框

注释： 不能在只读驱动器如 CD-ROM 驱动器上更名或删除目录。如果这样做，会得到 Access Denied 消息。

如何在 DOS Shell 中为一个目录更名

1. 激活目录树。
2. 选择要更名的目录。
3. 选择 File,Rename。
   Rename Dircetory 对话框打开。
4. 确认 Current Name 域显示正确的目录名(否则按 Cancel 重新开始)。
5. 在 New Name 文本框中键入新名。
6. 按 OK。

实践一下将 WORK 更名为 DETAIL：

1. 选择 WORK 目录。
2. 选择 File,Rename。
   Rename Directory 对话框打开。
3. 确认 Current Name 域是 WORK。否则，选择 Cancel 重试。
4. 在 New Name 文本框中键入 DETAIL。
5. 选择 OK。
   DOS Shell 将 WORK 更名为 DETAIL，并按字母顺序将它放在 PRACTICE 下正确的位置。

当一个更名的目录移动到新的字母位置时，你可能需要滚动屏幕才能看到它。

注释： 因为根目录不允许更名，所以选择根目录时，File，Rename 不能使用。

答疑： 试图在 DOS Shell 中为一个目录时更名，得到一个 Access Denied 消息。每个目录名在它的父目录中必须是独一无二的。在相同的父目录下不能有重复的目录名或文件名。如果试图分配一个不是独一无二的名字，Access Denied 消息将出现在 Rename Directory 对话框中。如果目录是只读的，也会得到这个消息。

第一个选项(Skip this file or directory and continue)与 Cancel 的功能相同。它取消 Rename 命令并返回到 Shell 屏幕。选择此项可以在重试之前再看一看目录树和文件清单。

第二个选项(Try this file or directory again)返回到 Rename Directory 对话框，这样可立刻再试另一个名称。

选择一个选项，可双击这个选项或使之高亮显示后按 OK。

6.7 删除目录

在用 DOS Shell 删除一个目录之前，该目录必须为空，这意味着这个目录中既不包含文件也不包含子目录。如果想删除整个分枝，必须从最低层开始向上工作。

提示： DELETE 命令是一个较快删除目录的方法。它在第14章“目录管理”中解释。

选择空目录后，按 Del 键可将它删除，如果 DOS Shell 有 Delete Confirmation(删除确定)功能，将出现一个对话框，允许用户确认或取消删除。如果 Delete Confirmation 功能失效，系统不再征询用户意见就删除这个目录。在第9章“Shell 中的其他文件管理”将介绍如何设置或取消 Delete Confirmation 功能。缺省时它是能用的，除非有人修改了系统，否则按 Del 键时会出现 Delete Directory Confirmation 对话框(见图6.5)。

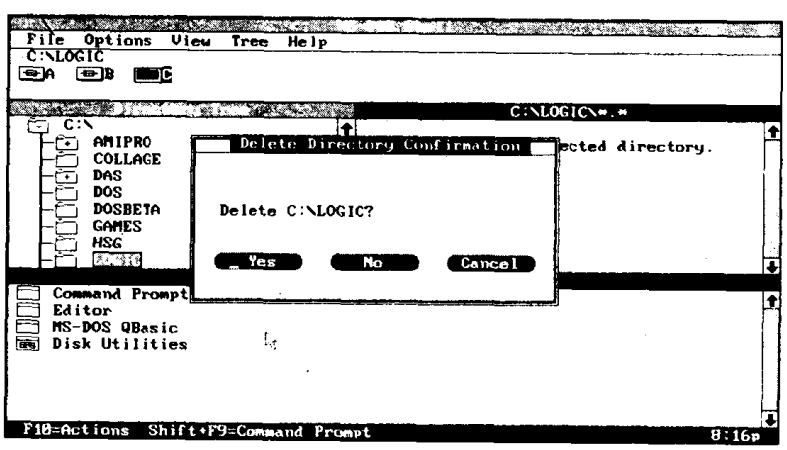

图6.5 Delete Directory Confirmation 对话框

如何在 DOS Shell 中删除一个目录

注释： 使用这个技术删除目录之前目录应为空，它必须既不包含子目录也不包含文件。

1. 激活目录树。
2. 选择打算删除的目录。
3. 按 Del 键。
4. 如果 Delete Directory Confirmation 对话框打开，确认它显示的目录名是否正确。如果正确，选 Yes(否则，选 No 或 Cancel)。

被选择的目录从目录树中消失；下一个高亮度目录被选择。

删除本章前面创建的目录：

1. 选择 LATER 目录。
2. 按 Del 键。

   出现 Delete Directory Confirmation 对话框。
3. 保证目录名是 LATER。否则，选择 Cancel 重试。
4. 选择 Yes。
5. 删除 NOW，DETAIL，PLAY，OTHER，最后是 PRACTICE。

警告： 在按 Del 键前要确保选择的是想删的目录，如果 DOS 的删除确认功能失效，那么在删去选择的项之前没有机会改变主意。

答疑： 试图在 DOS Shell 中删除一个空目录时，却看到消息 You cannot delete a non-empty directory。

这个消息可能令人奇怪，因为它在删除认为已空的目录时出现，实际上，目录中可能包含在当前目录树或文件清单中看不到的文件和子目录。

如果想删的目录旁有一个加号，则它有看不见的子目录。按 * 键显示整个目录分枝。从目录分枝顶部往上删除。

一个目录可能包含看不到的文件。缺省时，DOS Shell 不显示隐含文件和系统文件。如果文件清单中的消息是 No files match the file specifier 而不是 No files in selected directory，说明目录还包含文件。在删除目录之前必须显示并删除其中的文件。第9章介绍了如何完成这项工作。第11章“自定义 Shell”介绍了删除隐含文件。

6.8 自测练习

本章介绍如何改变驱动器和目录，以及创建、更名和删除目录。靠自己试一下这些任务。推荐作以下练习：

- 在软盘上创建一个完整的目录分枝
- 更名目录分枝中的每个目录
- 删除这个目录分枝

第7章 Shell 中的文件选择

简介： 在 Shell 中，在文件清单中选择文件后，通过 File 菜单中诸如 Print，Copy，Delete 之类的命令进行处理。

7.1 在 Shell 中选择文件

选择文件的方法与选择目录的方法很相似，一个重要的区别是：一次可选择多个文件，甚至可来自多个目录。这样一来，如果需要删除硬盘上的所有.BAK 文件，就可以只用一个命令选择并删除它们。

在本章中，主要讲述：

- 选择单个文件
- 选择几个相邻文件
- 选择几个不相邻文件
- 当多个文件被选择后，取消选择个别文件
- 在多个目录中选择文件

7.2 选择单个文件

一次只处理一个文件的情况经常遇到，例如想浏览或打印单个文件。选择单个文件和选择单个目录一样容易——只需单击它或将光标移在其上。高亮显示的文件就是选择的那个文件；这个文件变成了文件清单激活时你要求的任何 File 菜单命令的目标。例如，如果按 Del 键，选择的文件将被删除。

试一试用键盘选择一个单个文件：

1. 选择 DOS 目录。
   DOS 目录中的文件出现在文件清单中。注意它们按字母表示顺序排列(除非有人修改过 DOS Shell)。
2. 激活文件清单。
   清单中的第一个文件被选择。
3. 按 M 键将光标跳到 MEM.EXE。如果另一个以 M 开头的文件被选择，可继续按 M 直到光标到达 MEM.EXE。
   不论何时，高亮显示的任何一个文件总是当前选择的文件。注意，当移动光标时，前一个文件就取消选择了。

取消选择意味着取消一个文件的被选状态，这样它就不被选择了。无论何时，选择了一个文件，同时也就取消选择了前一个文件。如果以前选择了多个文件，它们全部被取消选择。

记住这一点是很重要的，即每次在文件清单中移动光标或单击一个入口，选择一个新文件时，取消选择了前面的所有文件。改变到另一个目录也会取消选择前面目录中选择的文件。(缺省时，新目录的第一个文件被选择)。

注释： 用鼠标卷屏不影响文件选择。但用键盘卷屏会影响，因为它会移动光标。

7.3 选择多个文件

有时选择一组文件进行处理是很方便的，但用鼠标和键盘选择多个文件在方法上有很大区别。

如何在 DOS Shell 中选择文件

用鼠标选择相邻文件：

1. 单击第一个文件。
2. 按住 Shift，单击最后一个文件。

用鼠标选择不相邻文件：

1. 单击第一个文件。
2. 按住 Ctrl，单击每个不相邻文件。

用鼠标选择相邻和不相邻混在一起的文件：

1. 单击相邻范围内的第一个文件。
2. 按住 Shift，单击相邻范围的最后一个文件。
3. 按住 Ctrl，单击想选择的每一个不相邻文件。

用键盘选择相邻文件：

1. 将光标移到此范围的第一个文件。
2. 按住 Shift，把光标移到此范围的最后一个文件。

用键盘选择不相邻的文件：

1. 选择第一个文件。
2. 按 Shift+F8进入 Add 模式(词 ADD 出现在状态行)。
3. 移动光标到要选择的下一个文件。
4. 按空格键选择它。
5. 重复步骤3和4以选择所有想要的文件。
6. 按 Shift+F8退出 Add 模式。

在 Add 模式下取消选择个别文件：

1. 移动光标到选择了的文件名。
2. 按空格键。

选择所有的文件，按 Ctrl+/。

取消选择所有的文件，按 Ctrl+\。

7.3.1 用鼠标选择多个文件

用鼠标选择多个文件的技术取决于想选择的文件在文件清单中是否相邻。

使用 Shift 键选择多个文件。试着选择在 DOS 目录中从 APPEND. EXE 到 CHKDSK. EXE 的所有文件：

1. 必要时翻卷文件清单，单击 APPEND. EXE。
2. 有必要再翻卷文件清单，按住 Shift，单击 CHKDSK. EXE。在 APPEND. EXE 到 CHKDSK. EXE 之间的所有文件呈高亮度。

要从上往下或从下往上创建一定范围的被选择文件时，可从欲选择范围的第一个文件或最后一个文件开始，然后按住 Shift，单击范围的另一端。

按住 Shift 时单击并不移动光标，光标仍保留在范围的第一个文件上，可再使用 Shift+单击调节本范围的另一端。

选择非相邻文件时你必须使用 Ctrl，试试选择文件 APPEND. EXE，CHKDSK. EXE 和 MEM. EXE：

1. 单击 APPEND. EXE。
   注意当这样做时取消选择了对前面的文件范围的选择。
2. 在必要时翻卷，按住 Ctrl，然后单击 CHKDSK. EXE(在这处情况下必须用鼠标翻卷，如果试图用键盘翻卷，将取消选择 APPEND. EXE)。
   CHKDSK. EXE 在 APPEND. EXE 保持高亮度时也被高亮显示。
3. 如果有必要再翻卷，按下 Ctrl 后再单击 MEM. EXE。
   MEM. EXE 在其他两个文件保持高亮度时被高亮显示(有可能因为在屏幕外看不见)。
   每次使用 Ctrl+单击选择一个文件时，它被置高亮度而且不取消选择前面的文件。

如果想选择一组混合文件，有的相邻，有的不相邻，可将两种方法结合起来，但使用 Shift+单击选择第一个相邻文件，然后使用 Ctrl+单击选择单个文件，使用 Ctrl+Shift+单击选择增加的一定范围的文件。

试着选择从 APPEND. EXE 到 CHKDSK. EXE 和 MEM. EXE 到 MODE. EXE 的文件：

1. 必要时翻卷文件清单，单击 APPEND. EXE。
2. 必要时再翻卷文件清单。按住 Shift 并单击 CHKDSK. EXE(必须使用鼠标翻卷)。
   在 APPEND. EXE 和 CHKDSK. EXE 之间的所有文件高亮显示。
3. 必要时翻卷文件清单。按住 Ctrl 并单击 MEM. EXE。
4. 必要时再翻卷文件清单，按住 Ctrl+Shift 且单击 MODE. COM(必须使用鼠标翻卷)。
   在 MEM. EXE 和 MODE. COM 之间的所有文件呈高亮度。你现在选择了两个范围的文件。

7.3.2 将选择和取消选择结合起来

如果仅想取消选择个别文件而不是全部，按住 Ctrl 并单击这个文件。Ctrl+单击可以选

择和取消选择个别文件但不影响其他已选择的文件。

假设想选择 DOS 目录中除 APPEND. EXE 外的所有文件：

1. 必要时翻卷文件清单，单击这个目录中的第一个文件。
2. 翻卷到文件清单的结尾，按住 Shift，然后单击目录的最后一个文件。(翻卷到结尾最快的方法是拖住滚动框直到滚动条的底端。)
   目录中的所有文件都呈高亮度。
3. 向上翻卷到文件 APPEND. EXE，按住 Ctrl，然后单击 APPEND. EXE。
   APPEND. EXE 被取消选择，高亮度消失，其他文件保持高亮度。

7.3.3 用键盘选择多个文件

与用鼠标相似，当用键盘选择文件时，如何选择一定范围的相邻文件和如何选择非相邻文件也有一定区别。

通过在从范围一端翻卷到另一端时通过按住 Shift 键来选择一定范围的文件。试着用键盘选择从 APPEND. EXE 到 CHKDSK. EXE 之间的文件：

1. 按字母 A 选择 APPEND. EXE。持续按 A 直到 APPEND. EXE 是被选文件。
2. 按住 Shift 并按 C 直到高亮显示延伸到 CHKDSK. EXE。这两文件之间的所有文件呈高亮度。

当按住 Shift 键时，可使用下面任意方法移动光标：上、下箭头键，PgUp 或 PgDn，Home 或 End，或文件名的第一个字母，也可在按住 Shift 键时移动光标来调节选择范围的结束端。假设想选择清单中除首尾两个文件外的所有文件，可选择第二个文件，按 Shift＋End 选择清单中剩下的文件，然后按 Shift＋上箭头键取消选择最后一个文件。

记住，选择个别文件将取消选择所有其他文件。如果不打算取消选择一个范围，最好不要在没按住 Shift 时移动光标。

为了使用键盘选择不相邻文件，必须使用一个特殊的文件选择模式——Add 模式。Add 模式使你能在移动光标时不选择或取消文件。Shift＋F8组合打开和关闭 Add 模式；当打开时，单词 ADD 出现在状态栏上。

1. 按 Shift＋F8。
   ADD 出现在状态栏且 Add 模式打开。
2. 按几次上箭头键。
   注意，从前选择的文件没被取消选择，移动光标时也没有任何文件被选择。
3. 按 M 翻卷到 MEM. EXE，按空格键选择它。
   MEM. EXE 在没有取消选择其他文件的情况下呈高亮度。

当处于 Add 模式时，也可通过按住 Shift 键后移动光标来选择一定范围内的文件。同样地，也可以通过按空格选择本范围的一端，然后移动光标到范围的另一端，然后按 Shift＋空格选择。在 Add 模式下也可以选择多个范围的文件，因为移动光标不会取消选择文件。

在处理选择文件之前不必离开 Add 模式。例如，可按 Del 删除所有选择的文件，然后在选择更多的文件后选择 File Print。在 Add 模式下，空格键成了一个文件选择开关，就像使用

鼠标时 Ctrl＋单击的功能一样，可通过移动光标到文件上，然后按空格键对它取消选择，当然只有这个文件被取消选择，所有其他文件仍保持被选状态。

1. 按 A 键直到光标在 APPEND. EXE 上。
2. 按空格键。

   APPEND. EXE 被取消选择，它的高亮度被消除。因为光标还在这个文件上，判断是否清除高亮度的唯一办法是看文件行开始的图标，看它是否仍高亮显示。

7.3.4　选择和取消选择所有的文件

通过使用 File 菜单的 Select all 和 Deselect All 命令可以选择或取消选择所有的文件。然而，使用它们的热键操作起来更方便；Ctrl＋/选择所有文件；Ctrl＋\取消选择所有的文件。

1. 按 Shift＋F8退出 ADD 模式。

   ADD 从状态栏消失。
2. 按 Ctrl＋/选择所有的文件。
3. 按 Ctrl＋\取消选择所有的文件。

   注意仍有一个文件保持选择——光标在其上的文件。

7.4　当没有文件被选择时

在 Shell 中取消选择所有的文件也是可能的。如果不小心（或故意）使用 Ctrl＋单击或空格取消选择最后保留的选择文件，File 菜单上的大部分命令将变暗（如图7.1所示）。为了使这些命令再次可用，选择一些文件。

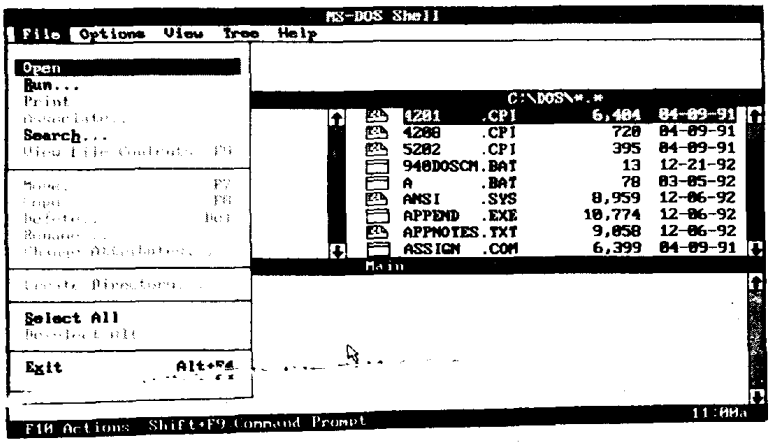

图7.1　当没有文件被选择时的 File 菜单

7.5 从多个目录选择文件

至此为止，只学习了如何在一个目录中选择文件。但如果同时在几个目录中都有要处理的文件，可使用 Shell 一次将它们全部处理(在命令提示符下不能这样做)。

在 Shell 中有两种方法从多个目录中选择文件。一种方法，称为 Select Across Directories，它很危险，许多人使用它常意外丢失文件；因此，本书将不讨论如何使用它。另外一种方法，称为 View All Files，相对比较安全，虽然也冒点险。

View All Files 在文件清单中显示一个驱动器上所有的文件(如图7.2所示)。在这个视图中，目录树和程序清单不被显示。文件清单包括这个驱动器上所有的文件，按字母表顺序排列。可不考虑目录选择任意组合的文件。按以下步骤试一下 View All Files：

1. 选择 View All Files。

   屏幕改变成在一个文件清单中显示所有文件。

2. 翻卷文件清单看看硬盘上的所有文件。

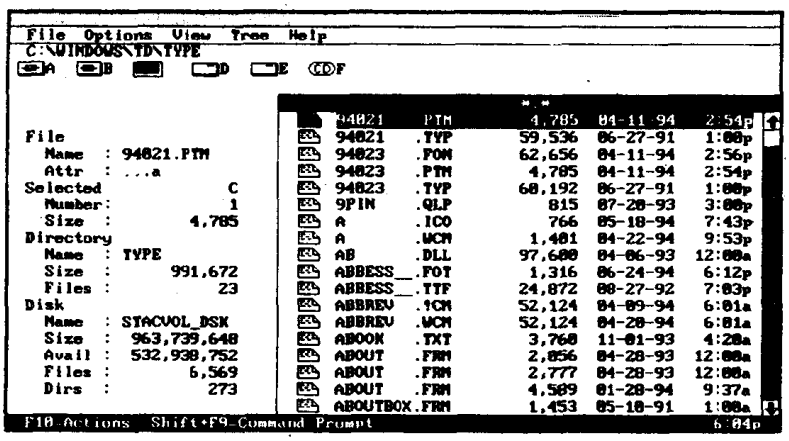

图7.2 View All File 清单

当在不同的目录下有两个或多个相同名字的文件名时，使用 View All Files 会有一点危险。例如，如果 PROFILE. DOC 出现在三个不同的目录上，它们可能是三个完全不同的文件，可能是一个相同文件的三个相同拷贝，也可能是一个文件的三个不同版本，或这些可能情况的组合。在 View All Files 清单中，这三个名字一个接一个地排列在清单中，带有它们的目录入口细节。如果想选择这些文件中的一个，很容易出错。

在 View All Files 中，左边的框可帮你决定该选择的文件。它显示关于驱动器和被选择文件的信息。Directory Name 域显示最近选择的文件所在的目录。如果清单显示具有相同名字的几个文件，且用户不能确定要选择哪一个，可逐个选择以查看它放在哪个目录中。这将有助于找出要选择的文件。

从 View All Files 中选择文件与在正常文件清单中选择完全相同。可选择单个文件、相邻的文件、不相邻文件，或者相邻和不相邻的文件的组合。

回到常用的 DOS Shell 屏幕，选择 View Program/File Lists：

1. 选择 View Program/File List。
屏幕回到正常状态。

7.6 自 测 练 习

在后面的章节学习移动、拷贝和其他文件管理时会有许多机会使用这些技术。为了确保能顺利地选择文件，试作以下练习：

1. 选择单个文件。
2. 选择几个不相邻文件。
3. 选择一定范围的相邻文件。
4. 选择两个或多个范围的相邻文件。
5. 在选择范围中取消选择一些个别文件。
6. 在不取消选择其他范围的条件下取消选择其中的一个范围。
7. 选择两个范围和一些单独文件。
8. 选择一个目录中的所有文件。
9. 取消选择所有选择的文件，光标所在的文件除外。
10. 在 DOS 目录和根目录下选择几个文件。
11. 在继续下一章之前取消选择所有文件，只保留一个文件以恢复到正常视图。

第8章 在DOS Shell中拷贝和移动文件

简介： 在工作中拷贝或移动文件的频繁发生到了叫人吃惊的地步。DOS Shell使这种操作更容易、更安全。

8.1 在DOS Shell中拷贝和移动文件

DOS最普通的任务之一是拷贝或移动一个文件。在这方面，使用DOS Shell要比使用命令提示符更容易。但即使在Shell中也会出现一些问题，因而应该有心理准备。本章主要讲述：

- 打开和关闭replacement confirmation(替换确认)功能
- 拷贝单个文件到另一个目录或在同一目录中拷贝
- 移动单个文件到另一个目录
- 拷贝和移动多个文件到另一个目录
- 拖动文件到另一个目录或驱动器
- 使用Shell屏幕的可选视图使拷贝和移动更容易

8.2 使用Replacement Confirmation功能

这看起来像是最安全的操作，但拷贝或移动文件实际上可能是危险的——如果不给出任何警告则可能丢失数据。当目标目录(接收文件的目录)已包含一个相同名字的文件时，DOS可能用新文件将它代替。DOS不能判断这两个文件是否相同或是不是相同文件的两个版本；如果名字相同，DOS将取代它。通常称为覆盖(clobbering)文件。

警告： DOS的UNDELETE程序不能恢复被COPY，XCOPY或Move命令覆盖的文件。(但Norton的实用程序Unerase可以)。

作者忠告： 请尽量使用DOS Shell拷贝和移动文件——它更安全(对6.2以前的DOS版来说)。

在DOS 6.2以前，COPY，XCOPY和MOVE命令会没有警告就覆盖文件，但DOS Shell不会。DOS 6.2改进了这些命令。如果还没有从早期的版本升级且没有Windows或一个第三方Shell，尽可能使用DOS Shell移动和拷贝文件。

图8.1显示了选择Options Confirmation命令时打开的对话框。这个对话框包含复选框。一个复选框就相当于一个开关：如果打开时选择它，将使之关闭；再选择时将重新打开它。框中有X时表示它已被打开。

第二项选择(Confirm on Replace)控制着在替换一个文件时Shell是否警告。检查一下，

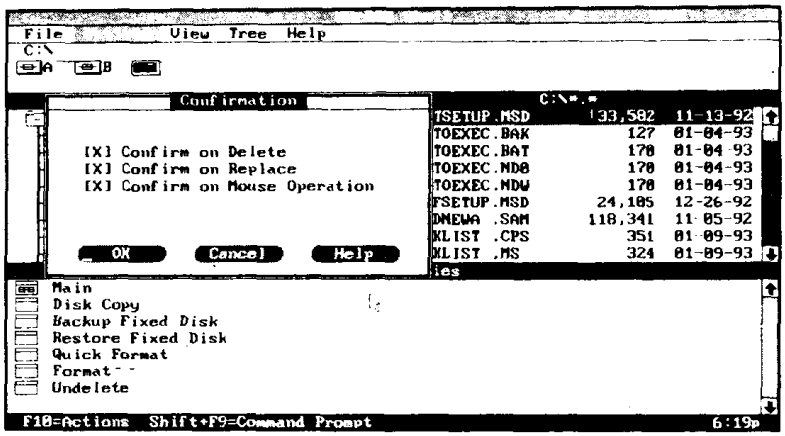

图8.1　Confirmation 对话框

以保证 Confirm on Replace 选项已被打开：

1. 选择 Options Confirmation。
   打开 Confirmation 对话框。
2. 如果 Confirm on Replace 有一个 X，说明此选项打开。选择 OK 关闭对话框。
3. 如果 Confirm on Replace 里没有一个 X，单击它一次使之打开。如果使用键盘，按向上箭头键两次将光标移到该选项上；然后按空格将它打开。
   当看见 X 出现在框中时，选项打开。
4. 选择 OK 关闭对话框。

在任何时候保持替换确认是一个好主意。只有在确实想替换几个文件，又不想在每次替换文件时作回答 Confirmation 对话框 OK 的厌烦工作时，才能关闭替换确认选项，只要可能应让该项处于打开状态。

8.3　拷贝单个文件

当拷贝单个文件到其他目录时，可以指定一个新名字，也可使用原来的名字，但在同目录中拷贝文件时，必须给它一个新名，因为一个目录下不能有相同名字的两个文件。

注释： 可以从只读驱动器如 CD-ROM 上拷贝文件，但不能拷贝文件到其上。

为了拷贝一个文件，先选择它，然后选择 File，Copy 命令（或按快捷键 F8）。图8.2显示了选择这个命令时打开的对话框，当前目录的路径名出现在 To 框中。如果想在相同目录下作一个备份，必须在路径名中增加一个文件名。当然也改变 To 框中的路径名，以将文件拷到一个新目录下，在这时可以用原来名字也可以更名。如果省略了文件名，拷贝将使用与最初相同的名字。

如何在 DOS Shell 中移动或拷贝单个文件：

在同一目录中拷贝文件：

1. 拷贝一个文件。
2. 选择 File,Copy 或按 F8(文件清单必须处于激活状态)。
3. 检查 Form 框中的文件名(如果错误,按 Cancel 再试)。
4. 按 End 移动光标到 To 框的文本结尾。
5. 如果 To 框中的路径名不以反斜杠结尾,键入一个反斜杠。
6. 在路径名中加一个文件名。
7. 按 OK。

拷贝一个文件到不同的一个目录：

1. 选择这个文件。
2. 选择 File,Copy 或按 F8(文件清单必须处于激活状态)。
3. 检查 From 框中的文件名(如果错误,按 Cancel 再试)。
4. 编辑或取代 To 框中的路径名
5. 如果想给拷贝到一个新名,增加一个文件名和扩展名(可选)。
6. 按 OK。

移动一个文件：

1. 选择一个文件。
2. 选择 File,Move 或按 F7(文件清单必须处于激活状态)。
3. 检查 From 框中的文件名(如果错误,按 Cancel 再试)。
4. 在 To 框中编辑或取代缺省的路径名。
5. 如果想给文件一个新名,增加一个文件名和扩展名(可选)。
6. 选择 OK。

答疑： File 菜单的 Copy 和 Move 命令都呈灰色,无法拷贝或移动任何文件。或者按 F7或 F8键移动或拷贝一些文件时,计算机只是发出嘟嘟声。

可能是事先没有首先选择文件或没有激活文件清单。激活文件清单并至少选择一个文件后,这两个命令才可用。

答疑： 在 File 菜单中选择了 copy 命令,但不能执行拷贝任务。

系统可能是处于程序清单,而不是文件清单。先按 Esc,然后激活文件清单,在至少选择一个文件后再试。

本章中,你需要创建两个专用于接收文件的目录,这样就可以不增加任何不想要的文件到工作目录中。现在创建目录,然后拷贝 C:\DOS\Help.COM 到第一个目录。(这是一个小文件,占用一个簇。它不会花你许多时间拷贝,且不会在硬盘上占用多少空间)。

1. 创建目录 PRAC1和 PRAC2作为根目录的两个子目录。
2. 选择 DOS 目录。
3. 选择 HELP.COM。
4. 按 F8键打开 Copy File 对话框。

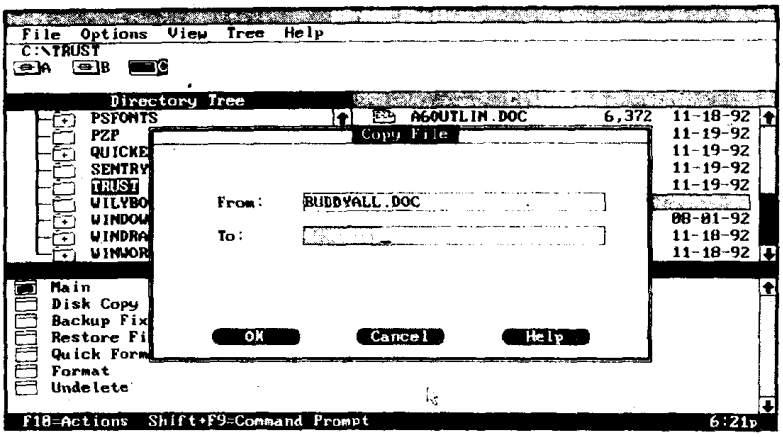

图8.2　Copy File对话框

Copy File对话框使用一个文本框接收从用户来的信息。当对话框第一次出现时，DOS高亮显示To框中的所有文本，表示这个文本被选择了。如果想取代它们，只须键入新内容。键入第一个字母后，框中原有的内容将消失。

5. 输入\PRAC1。

   路径名C:\DOS被\PRAC1代替。

6. 按回车完成拷贝。(按回车选择OK按钮)。

   DOS拷贝HELP.COM文件到\PRAC1目录。

7. 选择PRAC1目录查看新文件。

注释： *From框看起来象文本框，可以把光标移到框中，但试图键入内容时，DOS会嘟嘟响。*

假设你想拷贝文件到C:\DOS\OLDPROGS，而不是PRAC1。To框中已包含C:\DOS，这时不必键入整个新路径。可编辑已存在的路径而不是代替它。如果首先按一个水平箭头键(向左箭头，向右箭头，Home或End)，高亮度显示随即消失。这时End和向右箭头键对光标位置(处于尾部)无影响；Home和向左箭头键将把光标移到开头。无论使用哪种方法，现在可开始编辑了。

继续按左或右箭头键将光标移到想要编辑的地方。(在高亮度消失后，按左和右箭头键每次移动一个字符)。DOS可在光标处插入键入的内容。这样，为切换C:\DOS到C:\DOS\OLDPROGS，可先按End，然后键入\OLDPROGS。

提示： *开始编辑的另一种方法是单击想放光标的地方。*

如果与拷贝文件同名的文件已存在且正在使用Confirm on Replace，Replace File Confirmation对话框将出现(见图8.3)。它显示名字、日期和两个文件的大小，以帮你决定做什么。在这时若选择Yes则系统将执行替换；而选择No或Cancel则意味取消当前动作。

再试着拷贝HELP.COM到\PRAC1。这次，Replace File Confirmation框将出现，因为\PRAC1中已包含一个文件名字为HELP.COM的文件。

1. 选择DOS目录，再次选择HELP.COM文件。

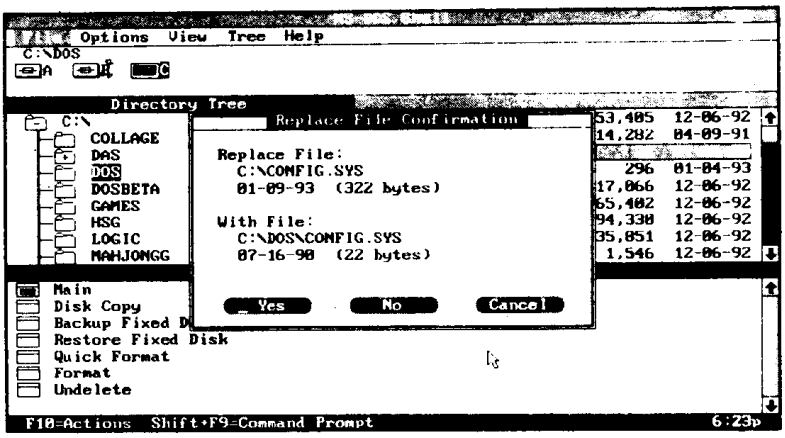

图8.3 Replace File Confirmation 框

2. 按 F8键以再次选择 COPY 命令。

   COPY File 对话框出现。

3. 键入\PRAC1并按回车。

   Replace File Confirmation 框将出现，因为 PRACTICE 目录中已包含一个名为 HELP.COM 的文件。

4. 选择 Yes。

   DOS 拷贝 HELP.COM 到\PRAC1，并覆盖这个目录中已有的文件。

答疑： 试图移动或拷贝一个文件到软盘时，得到一个 Access Denied 消息。

软盘是写保护的。可通过更换软盘或取消写保护来解决这个问题。之后选择选项2，Try this file again。

答疑： 试图拷贝一个文件时，得到一个 File Can't be copied onto itself 消息。

忘记了在 To 框中更换文件名，如果想拷贝到相同的目录，必须提供一个新的文件名。如果想移动或拷贝文件到一个不同的目录，则必须提供路径名。不能移动一个文件到相同的目录，因为它已存在。

现在能做的是取消操作，重新开始，更仔细地键入文件名或路径名。

8.4 移动单个文件

移动一个文件与拷贝一个文件相同，只是完成移动后 DOS 从源目录中将该文件删除。文件必须移到一个不同的目录，而不能移到相同的目录。移动一个文件的步骤如下：

1. 选择 PRAC1目录。
2. 选择 HELP.COM。
3. 按 F7选择 Move 命令。

   Move File 对话框出现。(如果 DOS Shell 嘟嘟响，说明文件清单没有激活，将它激活后再试)。

4. 这次，将文件移到 PRAC2，并把它的名字改成 MYFILE. XXX。把 C:\PRAC1编辑成 C:\PRAC2\MYFILE. XXX 再按回车。

   DOS 将文件从 PRAC1移到 PRAC2。可以看到它从 PRAC1目录中消失。

5. 选择 PRAC2目录。

   可看到 MYFILE. XXXX 在这个目录中。

注释： 不能从一个只读驱动器如 CD-ROM 驱动器移动一个文件，移向它也不行。

8.5　拷贝和移动多个文件

当选择多个文件进行移动或拷贝时，不能将它们更名。它们必须保持原来的名字，这意味着必须将它们拷贝到另一个目录。

如何在 DOS Shell 中移动或拷贝多个文件

1. 选择多个文件(保证没有其他文件被选)。
2. 选择 File，Copy (F8)或 File，Move(F7)。
3. 在 From 框中翻卷(如果不正确，按 Cancel 再试)。
4. 编辑或取代 To 框中的路径名以确认目标目录。
5. 选择 OK。

图8. 4显示了选择多个文件时 Copy File 对话框的样子。从中你可看见 From 框中的最后几个文件名，但却必须翻卷才可看到其他的文件。为安全起见，应在继续之前检查所有的名字。在 Shell 中极易多选文件。

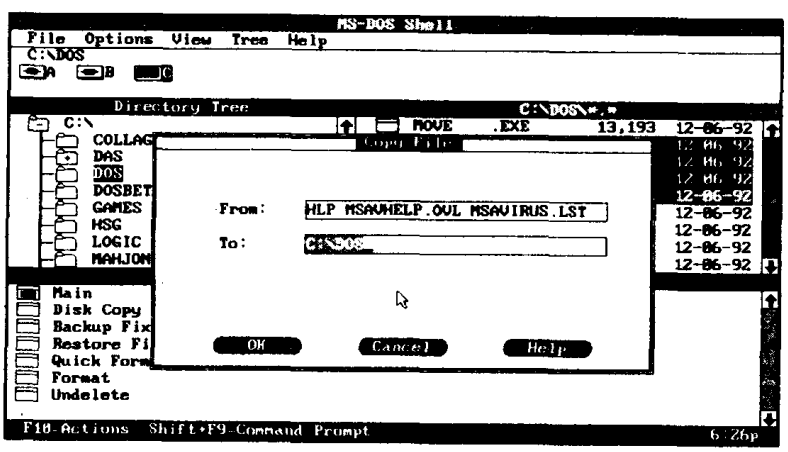

图8. 4　拷贝多个文件

使用光标键翻卷 From 框。试着拷一个文件集到 PRAC1目录中：

1. 在 DOS 目录中，选择 HELP. COM，LOADFIX. COM 和 MSD. COM。
2. 按 F8键。

   这时屏幕上将出现 Copy File 对话框。

3. 按 Home 和右箭头键翻卷 From 框中的清单(如果不正确,按 Cancel 再试)。

4. 按下箭头键移动光标到 To 框。

5. 键入\PRAC1并按回车。

如果在移动或拷贝多个文件时,屏幕上将出现 Replace File Confirmation 对话框(如图8.3所示),其中的 No 和 Cancel 分别对应不同的功能。通常,Cancel 取消整个命令,没有任何文件被拷贝;而 No 只取消当前文件,即如果还有下一个文件,则继续拷贝。

答疑: 试图移动或拷贝多个文件时,得到消息 You have more than one file selected。

如果指定一个并不存在的路径名作为目标目录,则 DOS 会认为这个名字是文件名而非路径名,同时认为用户可能想更名一个拷贝或移动的文件。但当移动或拷贝多于一个文件时,不能指定一个新的文件名。因此 DOS 在对话框中显示 You have more than one file selected。这个消息的真实意思是,"输入的路径名不存在,所以我认为它是一个文件名",如果真把它作为路径名,修改这个名字后再试。

8.6 拖动文件

既然已知道移动和拷贝文件的困难所在,应该学习一个容易的方法。如果有一个鼠标,则可用它选择想要的文件并拖到新的位置。但必须将它们拖到一个不同的目录,而且没机会更名。

如何在 DOS Shell 中通过拖动来移动或拷贝文件

移动文件到相同驱动器下的另一个不同目录:

1. 翻卷目录树和扩展子目录直到看到想要的子目录,但不要选择目标目录(源目录按以下步骤选择)。
2. 选择文件。
3. 拖动选择的文件到想要的目标目录中(在目录树中)。
4. 释放鼠标按钮放下文件。
5. 如必要,确认移动。

移动文件到不同的驱动器:

1. 选择目标驱动器(在驱动器清单中)。
2. 选择目标目录(在目录树中),把它作为这个驱动器的缺省目录。
3. 选择源驱动器(在驱动器清单中)。
4. 选择移动的文件。
5. 按住 Alt 键拖动选择的文件到目标驱动器(在驱动器清单中)。
6. 释放鼠标按钮放下文件。
7. 释放 Alt 键。
8. 如果必要,确认移动。

拷贝文件到相同驱动器上的另一个目录：

1. 在必要时翻卷目录树和扩展子目录以使在目录树中显示目标目录，但不选择该目标目录。(源目录按下列步骤选择)。
2. 选择文件。
3. 按住 Ctrl 键，把所选择的文件拖到目标目录上(在目录树中)。
4. 释放按钮放下文件。
5. 释放 Ctrl 键。
6. 必要时确认拷贝。

拷贝文件到一个不同驱动器：

1. 选择目标驱动器(在驱动器清单中)。
2. 选择目标目录(在目录树中)使之成为这个驱动器的缺省目录。
3. 选择源驱动器(在驱动器清单中)。
4. 选择要拷贝的文件。
5. 拖动选择的文件到目标驱动器(在驱动器清单中)。
6. 释放按钮放下文件。
7. 必要时确认拷贝。

为了拖动文件，先要选择它们；然后，用鼠标指向任何一个被选择的文件，按住鼠标按钮，把鼠标光标移到目标目录(在目录树中)或驱动器(在驱动器清单中)。如果目标目录不在相同的驱动器，DOS 自动拷贝文件到那个驱动器。但是，如果目录在相同的驱动器，则 DOS 将移动而不是拷贝它们。按 Ctrl 或 Alt 键可使 DOS Shell 不按缺省设置操作。其中 Ctrl 强迫 DOS Shell 拷贝文件，而 Alt 强迫它移动文件。

试着把 MSD.COM 从 PRAC1移到 PRAC2：

1. 选择 PRAC1目录。
2. 移动鼠标光标到 MSD.COM，按住按钮。
3. 在目录树中移动鼠标光标到 PRAC2目录，在这时请按住鼠标按钮。
   注意在屏幕上移动时鼠标光标会改变形状。关于鼠标光标形状的知识将在下节学习。
4. 当鼠标光标(此时像一个文件图标)在 PRAC2目录上时，释放鼠标按钮。
   这时屏幕上将出现 Confirm Mouse Operation 对话框它将问你是否确定要移动选择的文件到 C:\PRAC2。如果指定了错误的操作或目标目录，选择 No 再试。

提示： *要仔细阅读 Confirm Mouse Operation 对话框，以保证指定了正确的动作(移动或拷贝)和目标目录。*

5. 如果在确认框中指定了正确的动作和正确的目录，选择 Yes，这时 DOS 将把文件移到 PRAC2目录。同时该文件从 PRAC1目录中消失。
6. 选择 PRAC2目录并浏览移动的文件。

Shell 提供了许多可视线索来确定你在做什么。鼠标光标变成单个文件图标还是图标集取决于拖动多少文件。当移动鼠标光标经过不能放文件的区域(如程序清单时)时，鼠标光标将变成一个 No 符号——一个中间有一横线的圆。当它再变成一个文件图标时，已到达了一

个合法目标。另外，当不能放下文件时，状态栏显示 Move files to directory，在能放下时显示 Copy files to directory。

注释： 拖动文件时，可以翻卷目录树。这时只要将鼠标光标移到滚动条的箭头上就可看到清单翻卷。

如果设置了 Confirm on Mouse Operation（如图8.1，前面已显示），则屏幕上将出现一个对话框，以询问你是否确定了想移动或拷贝文件。这个确认功能几乎和 Confirm on Replace 一样重要，使之打开才能有改变主意的机会。如果经常误用移动和拷贝，或者经常把文件放到错误的地方，则可能需要确认步骤。但如果这些步骤已成为多余，可提前关闭。

提示： 注意看消息行。如果想拷贝时出现 Move 或者其他消息，则可按 Ctrl 或 Alt 以在拖动中改变命令，而无需重新开始。

注释： 在拖动文件时，你可注意到可把文件放到文件清单中的程序文件上。第10章“从 DOS Shell 运行程序”将解释这样做会发生什么事情。

8.6.1 改变视图

当拖动文件时，用户可以把 Shell 屏幕设置为自己最感方便的形式。图8.5显示了没有程序清单的屏幕外观。在这里出现了一个更大的目标树，这样更易到达一个目标目录。

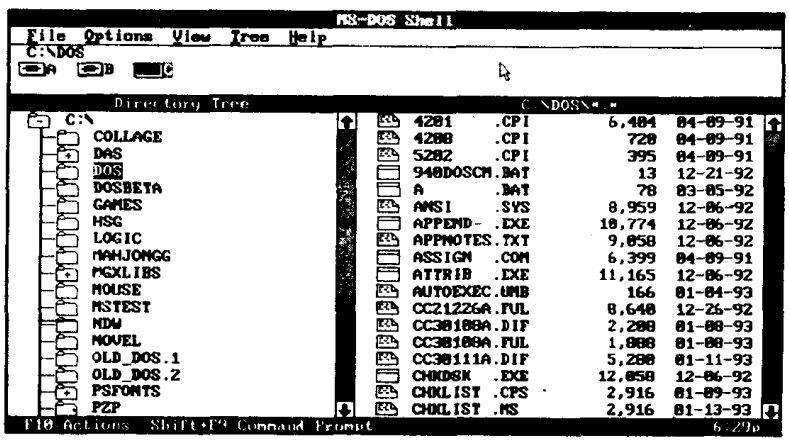

图8.5 清除程序清单

· 选择 View Single List 以观察屏幕上文件清单的样子。

图8.6显示了 Shell 屏幕的另一个视图，屏幕上有两个目录。用户可在屏幕上半部建立源目录，在下半部建立目标目录。当涉及两个驱动器时这个视图更方便，这将在例子中看到。该视图甚至可有助于相同驱动器上的两个目录之间的操作。此外，这个视图还可以带来额外的好处，即可在目标目录中看到已存在的文件。

试试一次看两个目录：

1. 选择 View，Dual File Lists。（这时第二个驱动器清单、目录树和文件清单出现在 Shell 屏幕的下半部分）。

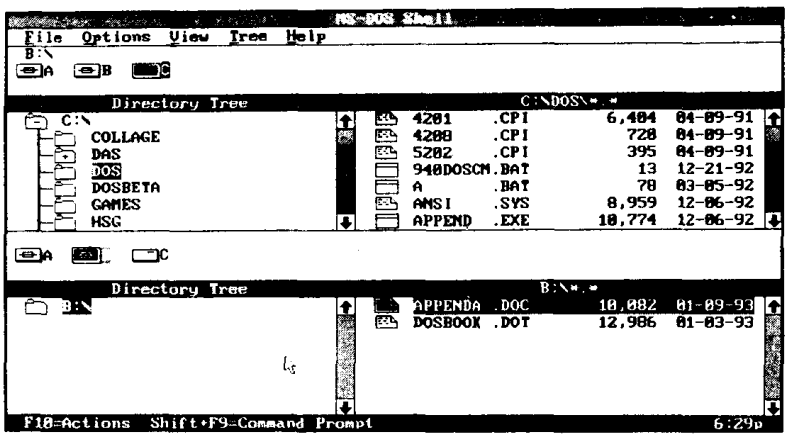

图8.6　两个文件清单

2. 在屏幕的上半部分选择 PRAC1目录，下半部分选择 PRAC2目录。
3. 把文件 LOADFIX.COM 从 PRAC1拖到 PRAC2，也可将它拖到下半部分的文件清单中。

   现在可看到文件从 PRAC1目录中消失，并出现在 PRAC2目录中。
4. 选择 View Program/File Lists 回到正常视图。

   这时 DOS Shell 将消除下半部分的清单集。

8.7　理解拷贝

当 DOS 拷贝一个文件时，它也拷贝了源文件的日期和时间，而不是分配当前日期和时间到新的拷贝。原因是日期和时间表明文件的版本——如创建于 March 13，3:10 p.m，的版本。只要不为拷贝更换版本，它们就具有相同的时间和日期，并且由此很容易知道它们是相同的。当其中之一更版时，日期和时间不再匹配，这样不仅可以知道它们不同，还可知道哪一个更新。

在拷贝和移动带有隐含、系统和只读属性的文件时，DOS Shell 就可能有点“太宽容”了，因为 Shell 忽视源文件的属性，即使有可能从源目录删去一个隐含或只读文件。也许更糟的是，Shell 这时会在不向你警告属性的情况下就在目标目录中取代了一个隐含或系统文件。

当 Shell 取代一个只读文件时，一个确认框会指出只读属性(在双括号间)。(见图8.7)，不管 Confirm on Place 是否置打开，这个框都会出现，但它不能阻止你覆盖那个文件。

8.8　从一个软盘移动和拷贝文件到另一个软盘

当你只有一个软驱时，它可以叫驱动器 A 或驱动器 B；驱动器 B 称“幻影”(phantom)驱动器。当启动时，DOS 认为它是驱动器 A。如果在驱动器清单选择驱动器 B(或在命令中用到驱动器 B)，DOS 将嘟嘟地响，并要求插入一张代表驱动器 B 的软盘，这时你可自己更换这个驱动器中的软盘。如果想继续使用正在驱动器 A 中使用的相同的软盘，可按 Enter 键而不

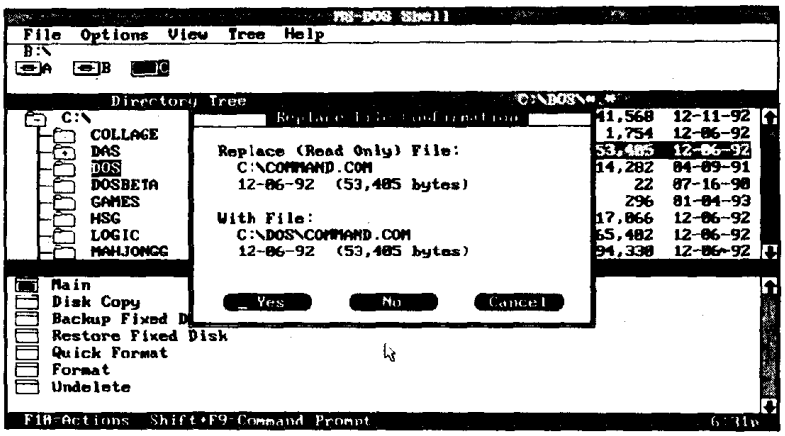

图8.7 Read Only 警告

更换软盘。从这时起，驱动器被认为是 B。当再次使用驱动器 A 时，DOS 再嘟嘟响一次，请求驱动器 A 软盘。在 DOS Shell 中，请求软盘的消息出现在状态栏。

如果只有一个软驱，可使用幻影驱动器从一个软盘拷贝和移动文件到另一个软盘。当在 A 与 B 之间移动或拷贝文件时，DOS Shell 嘟嘟响并在状态栏显示一个消息，以告诉你什么时候插入“驱动器 A”软盘和什么时候插入“驱动器 B”软盘。你必须记清这两个盘各是什么盘。

当做这个操作时，如果让驱动器处于“驱动器 B”状态，DOS Shell 认为当前是驱动器 B。如果后来选择 A，DOS Shell 嘟嘟响，并且叫你插入驱动器 A 软盘。

8.9 移动和拷贝一大组文件到多个软盘

如果目标盘没有足够的空间容纳新文件，DOS 将显示消息 The disk is full。如果目标是一个硬驱，则必须取消这个操作，同时找出如何解决的办法。但如果正拷贝或移动文件到一个软盘，可以改变软盘并选择选项1(Skip this file and continue)，从而继续拷贝操作。(它不会跳过任何文件)。

假设想移动或拷贝一大组文件到几个软盘。一次选择所有的文件并开始移动或拷贝。当每个软盘满了并出现 disk full 的消息时，插入一张空盘，同时选择选项1继续。

8.10 各种问题

在移动或拷贝文件时，可能会遇到一些问题。如可能键入了一个不存在的驱动器或目录名，或者把文件拖到了一个没有软盘的驱动器。由于这些错误动作，一个软盘可能被破坏了——当然硬盘也有可能。在每种情况下，Shell 显示几种消息并使你有机会修改这些问题。如果已拷贝文件到错误的目标，也不要担心，因为下一章将告诉你如何将它们删去。

8.11　自 测 练 习

试试拷贝多个文件到 PRAC1目录，然后把它们移到 PRAC2。另外也可试试从 PRAC1和 PRAC2移动和拷贝文件到一个软盘。(注意不要从有用目录移动文件)。

当你准备继续时，删去 PRAC1和 PRAC2中的文件，并删除这些目录，同时也删除拷贝到软盘的文件。

第9章 Shell中的其他文件管理

简介： 学习DOS的一个原因是它使你能做应用程序不能做的事情，如指定文件属性和重命名文件。

9.1 Shell中的文件管理

除文件移动和拷贝外，Shell还提供了几种文件管理功能。它们中的一些与你在许多应用程序中的功能相同，如显示或打印文件。但是当在Shell屏幕上工作时，有时使用Shell比启动应用程序更方便。其他功能则可能是应用程序所不具有的，如显示并改变某个文件的属性。本章将描述如何使用前面各章节尚未提到的Shell文件管理功能。

本章主要讲述：

- 浏览和改变属性
- 重命名文件
- 浏览文本文件
- 打印文本文件
- 删除文件

9.2 创建实例文件

在本章中你将学会许多不同的文件管理任务，且需要许多文件来作为练习的实例。现在我们花点时间来建立一个带有一个实例文件的目录。

1. 创建一个PRACTICE目录，作为根目录的子目录。
2. 拷贝HELP.COM，LOADFIX.COM和MSD.COM到PRACTICE下。

9.3 管理属性

你已经明白一个文件的属性如何来控制程序处理文件的方式（虽然甚至DOS也不总遵守这种约定）。表9.1归纳了四个属性。通过DOS Shell你可以浏览和改变一个文件的属性。

表9.1 文件属性

属性	功能
Archive(归档)	表示文件需要被备份
Read-Only(只读)	防止文件被修改或擦除

（续表）

属性	功能
Hidden(隐含)	防止在目录中看到文件名
System(系统)	组合隐含和只读

9.3.1 浏览属性

用户有时可能要看一看文件的属性，如为了备份一个文件，或为了用一个组删除命令将一个文件删除。图9.1显示了 Show Information 对话框，当在 Options 菜单下选择 Show Information 时该对话框将被打开。这个对话框提供一些关于当前文件、已选定文件的完整集合、当前目录和当前驱动器的统计数据。当前选择的文件的属性也包含其中。

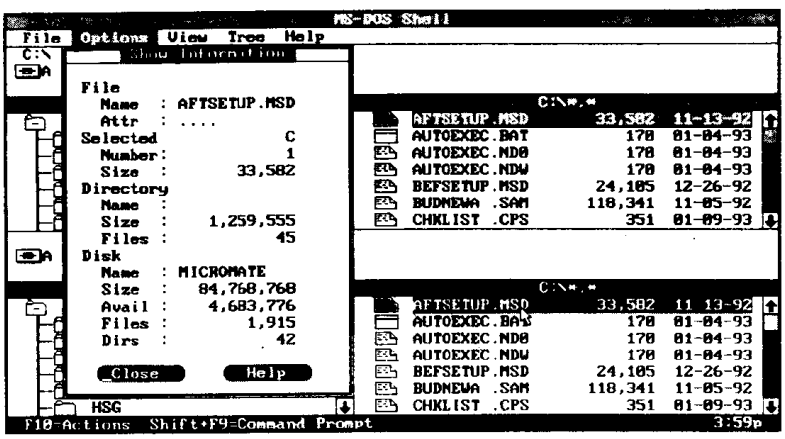

图9.1 Show Information 对话框

如何在 DOS Shell 中浏览一个文件的属性

1. 选择文件。
2. 选择 Options Show Information，如果有属性，可在 Attr 域看到。
3. 选择 Close 回到 Shell 屏幕。

警告： 要保证知道看到的是哪个文件。在第一个信息段显示的文件是最近选择或取消的选择的文件。例如，如果选择了文件 FILEA 和 FILEB，然后取消选择 FILEA，File 段将提供关于 FILEA 的信息(因为它是取消选择的最后一个文件)，而 Selected 段会告诉你关于 FILEB 的信息(因为它是选择的文件)。

Attr 域显示四个属性的状态。当字母 r(read-only)，h(hidden)，s(system)，或 a(archive)出现时，它表示它们代表的属性打开，...意味着属性关闭。在图9.1中，所有的属性关闭。

试着浏览 C:\PRACTICE\MSD.COM 的文件信息：

1. 选择 PRACTICE 目录。
2. 选择 MSD.COM。
3. 选择 Options，Show Information。

这时屏幕上将出现 Show Information 对话框。从中你可以看到 MSD.COM 的属性。

4. 选择 Close 关闭对话框。

9.3.2 改变属性

有时你可能发现有必要改变一个文件的属性。如为了拷贝某个文件而关闭该文件的隐含属性，或者为了备份一个文件而打开该文件的归档属性。使用 File，Change Attributes 命令可实现这些功能。

如何在 DOS Shell 中改变文件属性

改变单个文件的属性：

1. 在文件清单中选择文件。
2. 选择 File，Change Attributes。(文件清单必须被激活)。
3. 单击一个属性使之打开或关闭。或者，用 Tab 把光标移到一个属性，用箭头键选择一个，然后按空格键打开或关闭。
4. 在设置完所有的属性之后，选择 OK。

为一组文件设置相同的属性：

1. 选择所有的文件(同时取消选择所有其他文件)。
2. 选择 File，Change Attributes(文件清单必须激活)。
3. 选择选项2，Change all Selected files at the same time。
4. 选择所要的属性并进行相应设置。
5. 选择 OK。

图9.2显示了 Change Attributes 对话框。当只有一个文件被选择时，这个对话框显示该文件的当前属性(在图中为 Read Only)。现在选择属性并将其打开和关闭。如果使用鼠标，那么你可以单击一个属性使之打开或关闭。若使用键盘，则应该先使用 Tab 键移到一个属性，用箭头键使该属性处于高亮显示，然后按空格键使之打开或关闭。

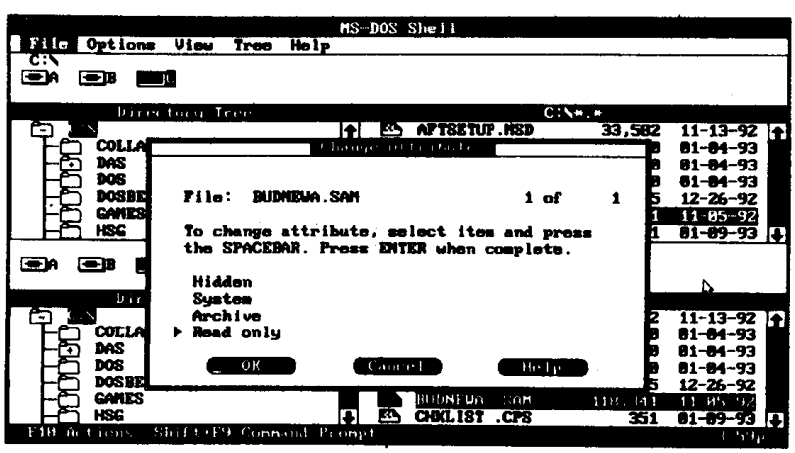

图9.2 Change Attributes 对话框

下面我们试着改变文件 C:\PRACTICE\MSD.COM 的属性：

1. 选择 PRACTICE 目录的 MSD.COM 文件。
2. 选择 File,Change Attributes(文件清单必须被激活)。
   Change Attributes 对话框打开。
3. 单击 Read only。若使用键盘,则可按 Tab 键将光标移到 Read only,然后按空格键。
   这时一个箭头出现在 Read only 旁边,表示它被打开。
4. 选择 OK 以关闭对话框。
5. MSD.COM 仍被选择时,选择 Options,Show Information 观察新的属性。
   现在你将看到 r,这意味着 Read only 属性被打开。
6. 选择 Close 关闭对话框。

注释： 你不能改变一个只读驱动器如一个 CD-ROM 驱动器上的文件的属性。

9.3.3 改变多个属性

当选择 File,Change Attributes 时,如已选择了多个文件,则 Change Attributes 对话框将如图9.3所示。

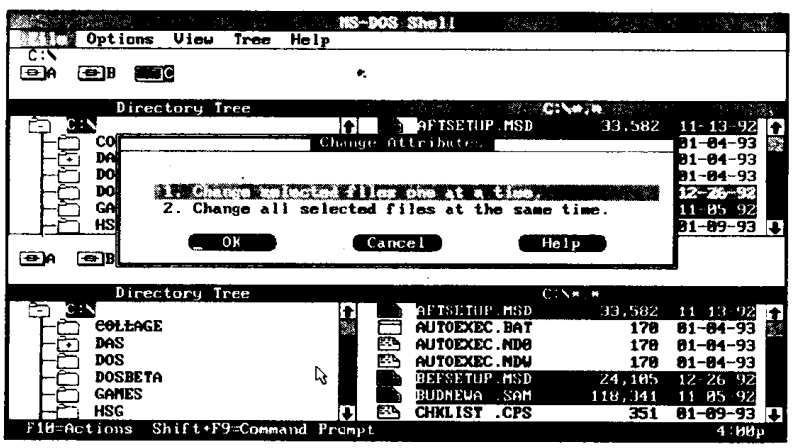

图9.3 改变多个文件的属性

必须在两个选项中选择。如果决定一次一个地改变选择的文件,那么图9.2中的对话框将为每个选择的文件出现一次。在对话框的右上角,你可以看到选择了多少个文件,并且知道哪个文件是你正在工作的文件,(如1 of 1,或4 of 7等)。

一旦决定改变所有选择的文件的属性,DOS 就立刻清除它们的属性。然后,它显示如图9.2所示的对话框,这样可为整个组设置属性。

假设你想让 PRACTICE 目录中的所有文件都是只读而没有其他的属性：

1. 选择 PRACTICE 中的所有文件。(记住快捷键 Ctrl+/可选择一个目录中的所有的文件)。
2. 选择 File,Change Attributes。
   Change Attributes 对话框出现。

3. 选择选项2,Change all Selected files at the same time。
   对话框改变到显示属性清单,所有选项均处于关闭状态。
4. 选择 Read only。
   这时一个箭头出现在 Read only 的旁边。
5. 选择 OK。
   DOS 为所有选择的文件设置 Read only 属性,同时清除其他属性。

注释: 如果想为一组文件设置一个指定属性而不改变它们的其他属性,必须一次一个地处理文件。如果选择一次全部完成,DOS 自动清除它们的当前属性。

9.4 文件更名

使用 File,Rename 命令可以很轻松地为文件更名。当选择这个命令时,屏幕上将出现一个对话框,在其中可输入这个文件的新名。当选择多个文件时,相同的对话框将反复出现,直到所有的文件被重新命名。在 Shell 中无法一次将它们全部更名。

如何在 DOS Shell 中为文件更名

为一个或多个文件更名:

1. 选择一个或多个文件。
2. 选择 File,Rename 命令(文件清单必须激活)。
3. 在对话框中,保证文件的当前名正确。(否则,按 Cancel 重新再试)。
4. 键入新名后选择 OK。
5. 重复第3步和第4步直到所有的文件都被更名。

下面试着为 PRACTICE 目录下的文件更名:

1. 选择 PRACTICE 下的所有文件。
2. 选择 File,Rename 命令(文件清单必须激活)。
   Rename File 对话框为第一个选择的文件打开。
3. 在对话框中,保证文件的当前名是正确的(否则,按 Cancel 重新再试)。
4. 键入 MYFILE1. XXX 后按回车。
   第二个被选择的文件出现在对话框中。
5. 重复3,4步直到所有文件被更名。另外两个文件更名为 MYFILE2. XXX 和 MYFILE3. XXX。

当最后一个文件被更名后,对话框消失,新的文件名出现在文件清单中。

在一个较长的清单中,新的文件名按字母顺序排列,可能需要滚屏才能见到它们。

注释: 不能为只读驱动器如 CD-ROM 驱动器上的文件更名。

答疑: 当试图在 DOS Shell 下为一个文件更名时,得到一个 Access Denied 消息。你需要的名字已经在目录中存在,该名字或者作为一个文件名,或者作为一个目录

名。它可能是当前没有显示在文件清单中的一个文件的名字。

在这时，如果选择了几个文件想重命名，并在为这个文件更名之前完成其他文件的更名工作，那么可以选择选项1(Skip this file or directory and continue)。当选择这个选项时，当前文件不会被更名，但 DOS 将显示下一个选择的文件(如果有的话)。

如果想在继续之前分配另一个名字给当前文件，选择选项2(Try this file or directory again)。

如果想取消这个工作以找出问题在哪里，选择 Cancel。

预览： 不能在 Shell 中为一组文件更名，但可在命令模式下实现，如将所有的 *.SAV 文件更名为 *.OLD。第16章"增加的文件管理"将介绍其具体操作方法。

9.5 浏览文件

在 Shell 下工作时观察一个文件的内容有时是很方便的。虽然总是可以通过启动创建数据文件的应用程序来浏览该数据文件，但在 Shell 下浏览会更快更容易。当然，当通过 Shell 来浏览一个文件时，不能像启动文件的应用程序那样对该文件进行编辑。

为了浏览一个文件，可以先选择它，然后选择 File，View File Contents 或按 F9键。这时 Shell 的文件浏览器将占满整个屏幕(如图9.4所示)。可用光标键在文件中滚动。(Enter 的功能如 PgDn，可通过状态栏的提示看到)。这里没有任何滚动条，但可以用鼠标单击屏幕底部消息栏中的 PgUp，PgDn，↑和↓来翻卷文件。

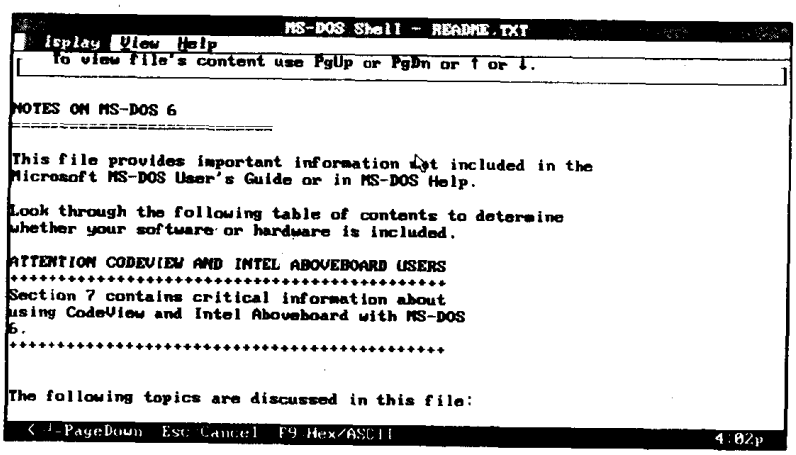

图9.4 在 Shell 中浏览一个文件

试浏览 DOS 目录下的 README. TXE：

1. 选择 DOS 目录。
2. 选择 README. TXT。
3. 按 F9键。

   DOS Shell 的文件浏览器占满屏幕并显示 README. TXT 文件。

4. 按 PgDn 几次。

   每按一次，文件向下翻一页。

5. 按向下箭头键。

每按一次，文件下卷一行。

6. 如果使用鼠标，单击屏幕底部 PgUp。

向上翻卷一页。

7. 如果使用鼠标，单击↑。

向上翻卷一行。

除 Help 菜单外，文件浏览器还提供了两个菜单。其中 Display 菜单包含两个命令，ASCII 和 Hex，它们控制着 Shell 如何显示数据。ASCII 按文本方式显示文件，如例子中显示的内容。Hex 按十六进制数显示文件中的字节，它对编程人员有用而对其他人无用。实际上不需要用 Display 菜单在 ASCII 和 Hex 两个模式间转换；正如你在状态栏中看到的，F9起着它们之间的转换作用。

注释： Shell 自动在 ASCII 模式下显示文本文件，而在 Hex 模式下显示所有其他类型文件。

View 菜单包含两个命令：Repaint Screen 和 Restore View。Repaint Screen(Shift+F5)做它在 Shell 菜单中相同的事，即清理一个杂乱的屏幕。Restore，View 命令(Esc)用于退出浏览器回到 Shell 屏幕(注意在消息行上的 Esc=Cancel 信息)。

8. 按 Esc 键关闭浏览器并回到 Shell 屏幕。

9.6 打印文件

在上一节已经看到了 DOS Shell 怎样显示一个文本文件，同样也能通过 DOS Shell 打印文本文件，但这时必须在内存中首先加载一个小的 TSR，而且应该于 DOS Shell 启动之前在 DOS 命令提示符下做这件事。

如何初始化 PRINT 程序
如何初始化 PRINT 程序： 1. 必要时退出 Shell。 2. 在命令提示符下键入 PRINT 后，回车。 3. 若使用的是连接于 LPT1的标准打印机，则直接回车，否则，键入打印设备名，再按回车。

作者忠告： 尽可能从命令提示符下加载 TSR。

TSR(terminate-and-stay-resident，即驻留内存程序)是一个一旦加载就保留在内存中的程序，它准备着在你需要时提供服务。程序如 DOS 的 DOSKEY，跟踪键入的所有命令的就是一个 TSR。一个弹出式警告程序就是另一个 TSR 的例子。

如果在启动了一个程序如 DOS Shell 或 Windows 后再加载 TSR，它将不能有效利用内存，以后有可能导致用尽内存。这样，最好在启动 DOS Shell 或 Windows 之前加载 TSR。

现在加载 PRINT TSR，以便在 Shell 下练习打印文件。

1. 按 F3键退出 Shell。(保证用 F3是永久退出;不要做临时退出)。
   系统返回到命令提示符下。
2. 键入 PRINT,回车。
   如果得到消息 PRINT queue is empty,说明 PRINT TSR 已经加载了。如果得到消息 Name of list device [PRN],说明 TSR 没有加载。继续到步骤3。
3. PRINT 需要知道打印机在哪里。如果它连在第一个(或唯一)并行口,只要按 Enter。如果它连在一个不同的端口,键入 LPTn(并行口)或 COMn(串行口),这里 n 是端口号,然后按 Enter。如果不能确定,也按 Enter。
   这时你将看到如下消息,它说明 TSR 已装上,当前没有打印任何东西。

```
Resident part of PRINT installed.
PRINT queue is empty
```

提示: 如果需要经常使用 PRINT,可把这个命令放在 AUTOEXEC. BAT 文件中。这样,每次启动时,PRINT 均被加载。第20章"基本批命令程序"将告诉你如何修改 AUTOEXEC. BAT。

9.6.1 通过 Shell 打印

初始化 PRINT 程序后,可通过 Shell 的 File,Print 命令打印文件。这时 PRINT 工作在后台,从而允许你继续其他工作,在这里它只是"忙里偷闲"地将字符送到打印机。(打印机与计算机的其他部分相比非常慢,因此,PRINT 可使打印机一直占用,而同时用户可将大部分处理时间放在想继续做的任何其他工作上)。

注释: 在打印机能使用之前,它必须通过电缆连在机器上,插在端口中,打开开关,并且选择了 online。同时打印机还必须有纸、色带、色粉或墨水盒,可能的话还应有字体盒或打印轮。

最适合于用 PRINT 打印的文件是 ASCII 文本文件。ASCII 文本文件是只包含一般字符的文件,这些字符可在一个标准美国键盘上输入,同时不使用 Ctrl 或 Alt 键,不包含为粗体、斜体、tab 键和图形字符等设定的任何特殊符号。它也不包含大部分应用程序插在文件中的额外的管理信息。一个 ASCII 文本文件不是很出色,但它与任何编辑器,如字处理程序、桌面印刷系统兼容。DOS 必须能读的文件,如 CONFIG. SYS,AUTOEXEC. BAT 和所有的批处理程序文件,必须是 ASCII 文本文件。DOS 的 EDIT 程序可用于创建和编辑 ASCII 文本文件。

你能打印任何类型的文件,但非 ASCII 文件会打印出一些奇怪的结果。许多非 ASCII 字节会作为图形或外文文字打印,而其他则会触发一些非正常的打印机功能如警告时的嗽叭响、吐出不完整的纸或打印"怪"字体。这不仅对于非文本文件如图形和程序文件来说如此,而且对于字处理程序准备的文档文件也是如此,因为一个字处理程序的格式码如完成字回卷和粗体类型等功能的代码常常是非 ASCII 代码。

大部分带有扩展名. TXT 的文件是 ASCII 文本文件,用 PRINT 程序就可以打印得很好。

试着打印目录 DOS 中的文件 README. TXT:

1. 必要时准备一下打印机。

2. 在 DOS 目录中选择 README. TXT。

3. 选择 File,Print。

这时一个确认消息出现在屏幕上,但系统较快时只能看见闪了一下,文件立刻开始打印。否则,打印机可能没有准备好。检查电缆是否接好以及打印机是否打开。

9.6.2 使用打印队列

PRINT 带有一个打印队列,它是等待打印的一系列文件。最多可在队列中排10个文件。每次选择 File,Print 时,PRINT 增加一个选择的文件到队列的尾部。

图9.5显示了 Print File 对话框,该对话框的出现意味着打印队列已没有更多空间。随后 PRINT 将继续打印,在这时你可以决定下一步的动作,你可以在一些文件结束打印的同时离开屏幕消息,然后选择选项1并增加更多的文件到打印队列;这样就可提高计算机效率,达到了使用 PRINT 后台打印的目的。或者,选择选项3(或按 Cancel),让 PRINT 在你做其他工作时完成当前的队列打印,在那之后你再选择那些尚未打印的文件重新再试。(选项2在打印队列满时不会有什么作用;它也不会有空间给下个文件)。

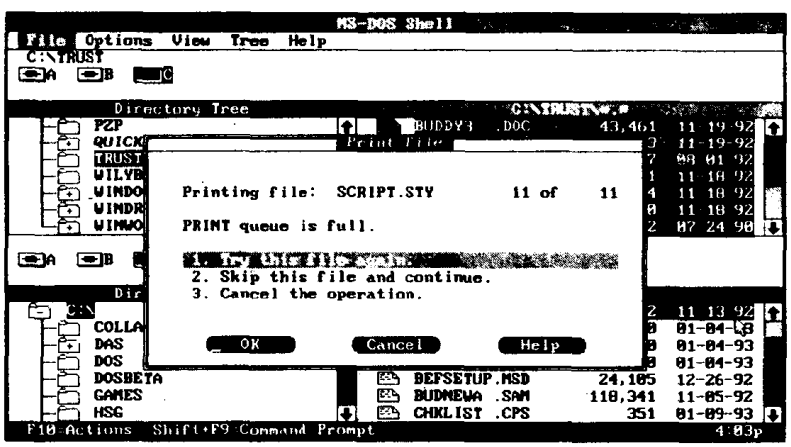

图9.5 Print queue is full 消息

9.6.3 选择一个打印程序

作为打印功能程序,PRINT 是最原始的一种。它不能做文本编辑器和字处理程序的打印功能能做的许多事情,如为页编号或打印多个拷贝。虽然它可以将文本分页,但这只是一些最基本的分页功能。PRINT 的主要优点是,在初始化后,用户不必在每次使用之前加载程序,而且它后台打印,从而使用户可以在打印期间继续进行其他工作。

如果打印几个纯 ASCII 文本文件且不考虑页号、页眉和尾注等,那么加载和使用 PRINT 即可。对于其他类型的打印工作,最好使用打印工具程序如 DOS 的编辑器、字处理程序或其他应用程序。

预览: 如果发觉要常常使用 PRINT,你可能想多学一点关于它的知识。其实 PRINT 命令还有许多其他功能特点。例如,扩大打印队列或者多分点时间打印(少

花点时间干别的事)。在命令提示符下工作时,可以浏览打印队列,把文件从中删掉或增加一些其他的文件。第16章将解释关于 PRINT 命令更多的信息。

9.7 删除一个文件

删除一个文件很简单,只需选择这个文件,然后按 Del 键(或者选择 File,Delete)。如果设置了 Confirm on Delete,那么会有一个对话框问你是否想删除这个文件。如未设置 Confirm on Delete,则 DOS 在按 Del 键时将立刻删掉文件。

提示: 因为很容易选错文件或无意中按 Del 键,因而最好将 Confirm on Delete 保持在打开状态。如果在删除一组文件时,不愿受它干扰,可以关闭,但完成后务必马上再将它置成打开。

按以下步骤保证 Confirm on Delete 是打开的。

1. 选择 Options Confirmation。
   Confirmation 对话框打开了。
2. 如果 Confirm on Delete 复选框中没有 X,单击它使之有 X。如果没有鼠标,用 Tab 键将光标 移在其上,然后按空格键将它置打开。
3. 选择 OK 关闭对话框。

现在试着从 PRACTICE 目录删除 MYFILE1.XXX:

1. 选择 MYFILE1.XXX。
2. 按 Del 键。
   Delete File Confirmation 对话框打开。注意文件是 Read only 的警告,因为已在本章前面将所有置为 Read only。DOS Shell 可以删除任何类型的文件。
   确信对话框指的是文件 C:\PRACTICE\MYFILE1.XXX。如果不是,选 Cancel 再试。
3. 选择 Yes 删除文件。
   对话框关闭且该文件从 PRACTICE 目录中消失。

预览: 几乎每个 DOS 用户有时都可能需要在删除文件之后恢复它们。DOS 6包含的 UNDELETE 工具可做这个工作。第26章"UNDELETE 的使用"将描述如何进行文件恢复。

9.8 删除多个文件

为了删除多个文件,应将需要删除的文件全部选择。保证没有其他文件被选择——在多个目录中选择时尤其要注意。当按 Del 键时,屏幕上将出现一个对话框,如图9.6所示。这时可翻卷清单来保证所选文件是正确的文件。

当选择 OK 时,如果置了 Confirm on Delete,删除每个选择的文件时都会出现一个确认对话框,从而必须为每个文件选择 Yes 或 No。选择 No 跳过当前文件,继续到清单的下一个文件。选择 Cancel 停止整个过程。

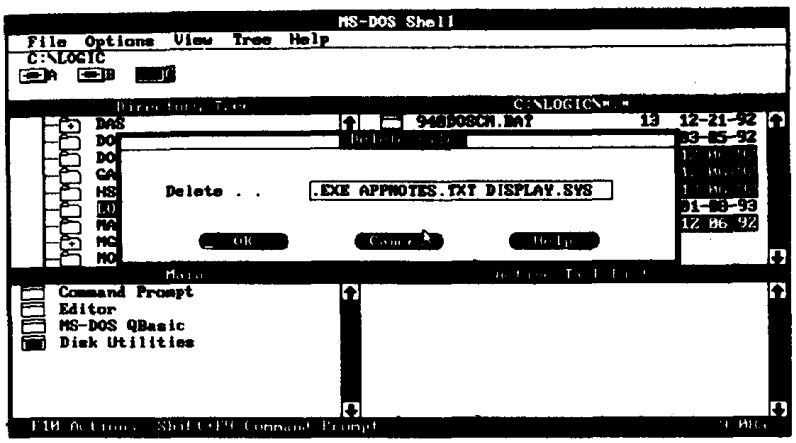

图9.6 Delete File 对话框

试着删除 PRACTICE 目录中的余下文件：

1. 选择文件 MYFILE2. XXX 和 MYFILE3. XXX。
2. 按 Del 键。

   Delete File 对话框打开。

   保证 Delete 框中的文件是你想删除的文件。否则，按 Cancel 重试。
3. 选择 OK。

   针对 MYFILE2. XXX 的 Delete File Confirmation 对话框出现。
4. 选择 Yes。

   针对 MYFILE3. XXX 的 Delete File Confirmation 对话框出现。
5. 选择 Yes。

   对话框关闭，两个文件从目录中删除。

警告： 删除确认功能给了你第二次机会去注意是否选择了不想删除的文件。许多人会变得不耐心，开始迅速选择 Yes 而不阅读确认框中的文件名。如果养成这个习惯，会很容易地不小心删除不想删除的文件。最好养成在确认框中检查文件名的习惯。

如果试图从一个写保护磁盘或一个只读驱动器如 CD-ROM 驱动器删除文件，将看到 Write Protection Warning 对话框(如图9. 7所示)。只要磁盘是写保护的，就不能删除其中的文件，同时也不能删除一个只读驱动器上的文件。但可去掉一个正常盘的写保护后再试。

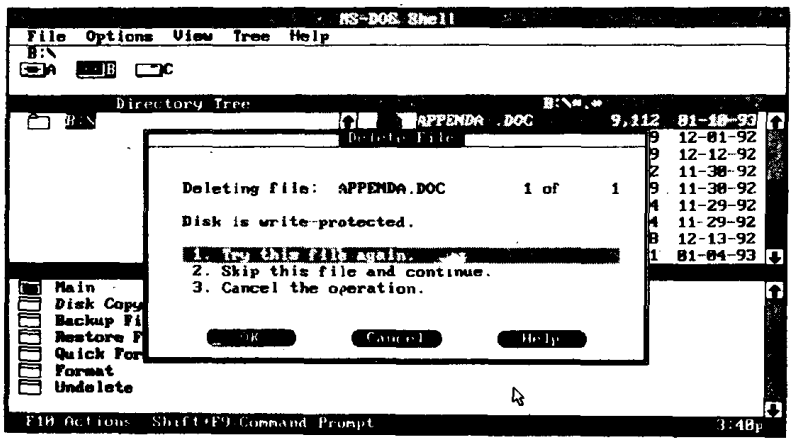

图9.7　Write-Protected 消息

9.9　自 测 练 习

本章描述了几种不同的文件管理技术。如果你愿意用更多时间来实践它们，试一下以下几项：

1. 从 C:\DOS 拷贝所有的.TXT 文件到 C:\PRACTICE。(最容易的方法是限制文件清单只显示.TXT 文件)。在 C:\DOS 中找到所有的.TXT 文件。

以下几个练习仅对 C:\PRACTICE 目录中的.TXT 文件操作：

2. 使它们全部隐含。
3. 清除隐含属性(在选择它们之前，需要包括隐含和系统文件在文件清单中)。简单检查一下.TXT 文件。看看它们是否都包含关于系统的信息。
4. 如果发现有些文件与系统有关，将它们打印出来。
5. 从 PRACTICE 目录中删除文件(不要从 DOS 目录中删除)。
6. 删除 PRACTICE 目录。

第10章 从 DOS Shell 运行程序

简介： 不必退出 DOS Shell 就可运行一个文件。实际上，从 Shell 运行文件常常更好。

10.1 从 Shell 中运行文件

当使用 Shell 作为 DOS 主要界面时，可从 Shell 屏幕启动应用程序。当退出一个应用程序时，回到 Shell 屏幕完成下一个工作。如果需要运行一个 DOS 实用程序如 FORMAT（格式化一张软盘）或 MS Anti-Virus（扫描病毒），也可直接从 Shell 屏幕运行。

警告： 在 DOS 的目录结构以外工作的实用程序如 SCANDISK（它分析和修理目录结构）和 DEFRAG（它重组一个驱动器上的簇）不能从 Shell 或 Windows 运行。运行这样的实用程序前要退出所有的程序。

有几种方法可从 Shell 屏幕启动程序。每种方法各有优点和缺点，在本章中主要讲述：

- 从文件清单运行程序
- 通过把一个数据文件程序文件上来运行程序
- 使文件与程序相互关联
- 从一个相关文件运行程序
- 从程序清单运行文件
- 启动及终止任务切换功能
- 启动和终止任务
- 在任务间切换

10.2 打 开 对 象

本章将告诉你如何打开一个对象而不是仅仅选择它。可通过双击或选择后按 Enter 键（即回车键）来打开一个对象。

你已经明白在驱动器清单上选择一个对象和将它打开之间的区别。如果你选择了一个驱动器，不需要更新它就可切换到其上；但是如果打开它，就更新了它。

在本章中，你将学会如何通过打开一个程序的程序文件，或者打开一个属于该程序的数据文件，或打开程序清单中的某一项来运行一个程序。

10.3 从文件清单运行程序

启动程序的一种方法是从文件清单打开它的可运行文件。除非你知道这个程序文件的名字，否则不能这样做，但是这个问题并不难解决。它常常是带有.EXE 扩展名的程序名，如在 WORD.EXE(Microsoft Word)或123.EXE(Lotus 123)中。一些程序使用.COM 或.BAT 而不使用.EXE。

当通过打开一个程序的可运行文件启动该程序时，Shell 屏幕被程序的屏幕(或者如果程序没有自己的屏幕，用 DOS 命令提示屏幕代替)所取代。当程序结束时，将有一个消息提示你按任意键回到 MS-DOS Shell。

如何从 DOS Shell 的文件清单运行程序

通过打开它的程序文件运行一个程序：

1. 打开包含这个程序的目录。
2. 双击程序的.EXE,.COM 或.BAT 文件。或者选择.EXE,.COM 或.BAT 后按 Enter。
3. 当结束一个程序时，按任一键回到 Shell 屏幕。

通过把一个数据文件放在程序文件上来运行程序：

1. 选择一个数据文件。
2. 把它拖到程序文件的入口，必要时滚动屏幕。
3. 把数据文件放在程序文件上。
4. 在确认对话框中选择 Yes。
5. 当程序终结时，按任一键回到 Shell 屏幕。

使用以下一种方法从一个相关文件运行一个文件。

- 双击该关联文件。
- 选择关联文件后按 Enter(或选择 File,Open)。

你将实践在文件清单里运行一个程序，在这里运行的是 DOS Editor，并且使用它来创建和编辑文本文件。在这种情况下，你只要启动编辑器然后退出，不必编辑任何文件。

1. 选择 DOS 目录。
2. 翻卷文件清单到 EDIT.COM。
3. 双击 EDIT.COM 或选择 EDIT.COM 后按 Enter。
   Edit 的屏幕代替了 Shell 屏幕。现在不要担心这个屏幕意味着什么。你将在第19章“DOS Editor”学习如何使用编辑器。
4. 为了退出编辑器，按 Esc，然后按 Alt+F，再按 X。
   编辑器屏幕关闭，一个消息将提示你按任意键回到 MS-DOS Shell。
5. 按一键回到 Shell 屏幕。

不能用这种方法运行任何一个文件。例如 Windows 应用程序在 Windows 之外就不能运行，否则会收到一个错误消息。许多 DOS 命令(如 DIR 和 COPY)建立在 DOS 的命令处理程

序内部，没有它们自己的程序文件。此外，需要一个启动参数的程序（如文件名）不能从文件清单启动，因为没有机会输入启动参数。例如，FORMAT 程序需要一个驱动器名。如果试图从文件清单中运行它，将收到错误消息 Required parameter missing。

10.4 放一个数据文件在一个程序文件上

从文件清单运行文件的另一种方法是使用鼠标将一个数据文件放在一个程序文件上。这个过程使用放下的文件作为初始数据文件来启动程序。

例如，如果拖动文件名 APPNOTES. TXT 并放到 PRINT. EXE 上，将出现 Confirm Mouse Operation 对话框（如图10. 1所示）。当选择 Yes 时，PRINT 程序增加 APPNOTES. TXT 文件到它的打印队列。再举一个例子，把名为 MEMO. DOC 的文件放到 WORD. EXE 上，会启动 Microsoft Word，其中 MEMO. DOC 文件已打开。

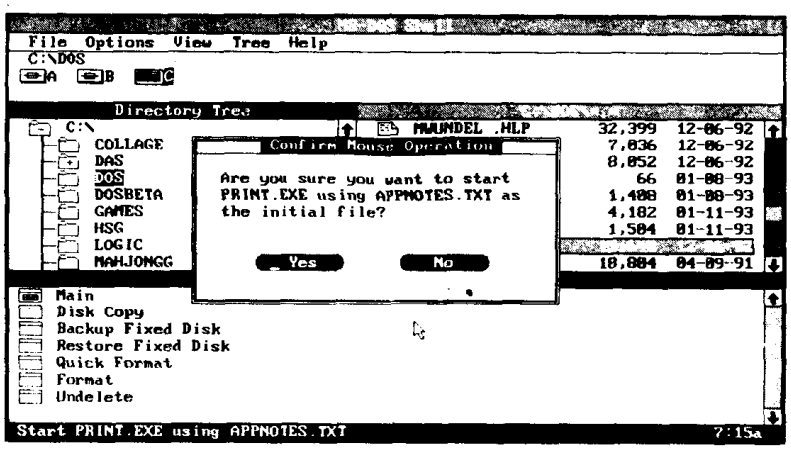

图10. 1 Confirm Mouse Operation 对话框

Microsoft 在文本文件 README. TXT 中包含一些 DOS 版本的最新的资料。试着用 README. TXT 文件启动 DOS 的编辑器。

1. 确定 DOS 目录被选择。
2. 翻卷文件清单直到看见 README. TXT。
3. 开始向上拖动 README. TXT（因为文件按字母顺序排列，EDIT. COM 文件在 README. TXT 的上面）。
   当鼠标光标在可执行文件上时变成一个文件图标，在其他类型文件上时变成一个“no”图标。
4. 因为看不到 EDIT. COM 文件，你必须拖动鼠标翻卷文件清单，将鼠标光标移到滚动条的向上指的箭头上即可。（不要让 README. TXT 文件掉了）。
   文件清单开始向上翻卷。
5. 当看到 EDIT. COM 文件时，箭头移开。（如果你走过了，可使用向下箭头调整位置，但不要放开鼠标按钮）。
   当光标移出滚动条时，文件清单停止翻卷。

6. 把鼠标光标移到 EDIT. COM 文件并释放按钮。

一个 Confirm Mouse Operation 对话框问你是否确定要使用 README. TXT 作为初始文件启动 EDIT. COM。

7. 选中 Yes。

这时 DOS Shell 屏幕消失，同时出现一个 Edit 屏幕，其中显示的是 README. TXT 的文本内容。

通常这样可阅读一个文件，甚至可以打印它的部分或全部。但现在只要关闭当前屏幕然后回到 Shell 屏幕。

1. 要退出程序，按 Alt＋F，然后按 X。

Edit 屏幕消失，出现消息 Press any key to return to MS-DOS Shell。

2. 按任意键。

系统返回到 Shell 屏幕。

当你在文件清单上拖动一个数据文件时，如果鼠标光标在运行文件上时将从“no”符号改变成一个文件图标。不要把这个信号错误地认为是这个程序适合于这个数据文件。DOS 允许把一个数据文件放到任何程序文件上，而不管是否适合。

因为 Edit 编辑文件，Edit 被设计成接收一个初始数据文件。但上述过程在程序并不工作在数据文件上时不发生作用。例如，DOS 的 FORMAT 程序为 DOS 准备一张使用的磁盘；它不作用于数据文件。如果你把一个数据文件放到 FORMAT. COM 上，将得到一个错误消息。

假设数据文件和程序文件在不同的目录下，甚至可能在不同的驱动器下，那么在拖动数据文件时由于不能改变驱动器或目录，如何将数据文件放在程序文件上呢？这里有一个方法，即使用 View，Dual File Lists 在屏幕的下半部分打开第二个清单集。把数据文件的目录安排在上半部分屏幕，把程序文件的目录建立在下半部分屏幕，然后把数据文件从上面拖到下面的程序文件上。

10.5　打开一个关联文件

你发现将 README. TXT 拖到 EDIT. COM 上的不足之处了吗？这里有一个更容易的方法来启动一个带有一个初始文件的程序，但它只对几个程序有效。一些文件扩展名与使用它们的应用程序有联系。这意味着你可通过在文件清单中打开数据文件来运行带有初始文件的程序。缺省状态下，DOS Shell 将所有的. TXT 文件与 EDIT 程序相关联，将所有的. BAS 文件与 QBASIC 文件连接起来。例如，如果打开 README. TXT，DOS 将以 README. TXT 为初始文件启动 EDIT。

试着通过打开 README. TXT 文件来启动 EDIT 程序：

1. 双击 README. TXT 或选择它后按 Enter。

DOS Shell 屏幕消失，EDIT 屏幕出现，并显示 README. TXT 的文本。

2. 按 Alt＋F，然后按 X，退出程序。

Edit 屏幕消失，出现消息 Press any key to return to MS-DOS Shell。

3. 按任意键。

返回 Shell 屏幕。

这是从 Shell 运行程序最简单的方法之一，但 DOS 只提供上面这两种关联，你可能想建立更多这种关联。例如，你可能想将文档文件与字处理程序相关联，将工作表格文件与电子表格程序相关联，将数据库文件与数据库管理程序相关联，等等。

可使用 File Associate 命令创建一个关联。图10.2显示了选择这个命令时打开的 Associate File 对话框。

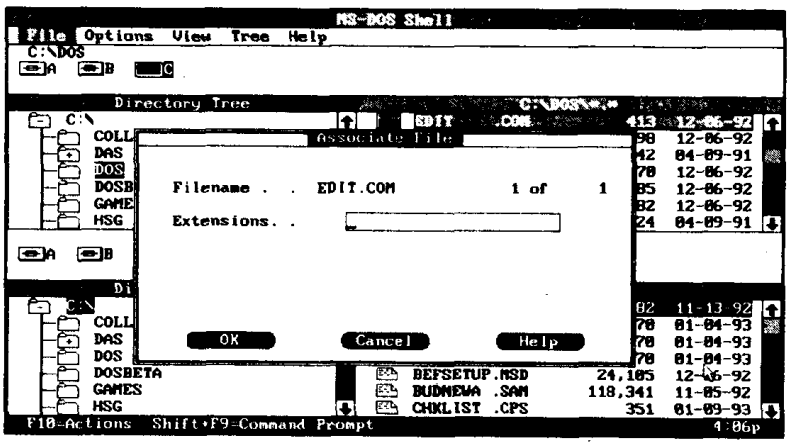

图10.2 Associate File 对话框

如何将文件扩展名与一个程序相关联

1. 选择程序文件。
2. 选中 File，Associate。
3. 将扩展名加入到对话框中(如果想使这个程序与多种类型的文件相关联，可加入多个扩展名)。
4. 选中 OK。

一个程序可以与任意多个扩展名相关联，反之则不然。一个扩展名只可与一个程序连接，否则，DOS Shell 在你打开连接文件时不知道启动哪个程序。

提示： 如果你使用 Windows，就不需要花时间关联 Windows 应用程序和它们的文件。因为 Windows 应用程序不能在 Shell 下运行。

提示： 在使空扩展名和一个程序相关联时，只需在 Associate File 对话框中键入一个句点(.)即可。例如，若使空扩展名与 EDIT.COM 相关联，则打开如 OCTOBER，MYDAY 和 FORM10之类文件时(都没有扩展名)将启动 DOS 的编辑器。

10.6 从程序清单运行程序

另外一个启动程序的方法是打开程序清单中的项。例如，打开 Editor 项会启动 DOS 编辑器。

程序清单中的一项可能被定义为显示一个对话框，这样可输入启动参数，如一个文件名或驱动器名。图10.3显示了 File to Edit 对话框，它在打开 Editor 项时出现。在这个对话框中，可以为 EDIT 程序输入一个初始文件的名字。

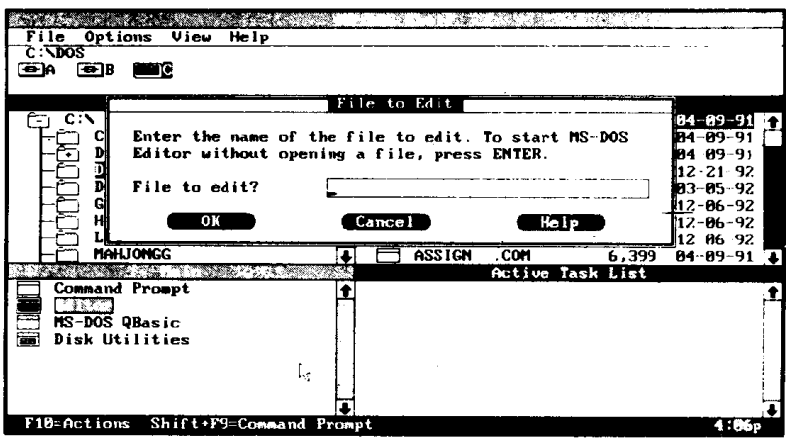

图10.3　从程序清单启动 EDIT

试着从程序清单启动 Editor：

1. 双击 Editor，或选择它再按 Enter 键。
   这时一个对话框将提示你输入一个文件的名字。
2. 键入\DOS\README. TXT 到对话框中。(记住 DOS 忽略大小写，如果你愿意，可以小写形式键入)。
   Editor 打开并显示 README. TXT。
3. 按 Alt＋F，然后键入 X 关闭编辑器。
   可立刻回到 Shell 屏幕，而不必按一键返回。

此外用户还可请求许多程序项的相关帮助，在这时只需选择相应的项(但不打开它)并按 F1键即可。帮助信息将解释关于程序及其参数的一些信息：

1. 在程序清单中选择 Editor(但不打开它)。
2. 按 F1键。
3. Help 对话框将显示关于 Editor 程序项目的信息。
4. 按 Esc 关闭对话框。

当这个程序终结时，有时可能不需要按键就可回到 DOS Shell。有些程序直接回到 Shell，但是，在屏幕上留下一个最后消息的程序常常让你决定什么时候返回。

注释： *第11章“自定义 Shell”解释了如何在程序清单中增加自己的项。*

10.7　在命令提示符下工作

由 Shell 运行程序的另一个方法是回到命令提示符，输入一个命令，然后运行该命令，当它结束后再回到 Shell。在掌握了更多关于 DOS 命令提示符后，你将在第12章“在命令提示

符下工作"中学习这个技术。

10.8 任务切换

用 DOS Shell 一次可启动多个程序。例如，你可能启动一个字处理程序，一个电子表格处理程序，一个数据库和一个图形程序。这样，就能按工作需要在它们之间进行切换。

使用任务切换，每次只将一个任务激活。当激活一个任务时，其他的任务处于睡眠状态。你不会失去对睡眠任务的控制，但当它们没有激活时不能做任何处理，这是与一个诸如 Windows 之类的多任务系统最本质的区别。在多任务系统中，几个程序可同时激活，共享处理器时间，这样在一个程序中输入数据时另一个程序可进行打印或计算。

Shell 的任务切换器与 Windows 的另一个区别是，不能用任务切换器从一个任务拷贝数据到另一个任务。如果你在写一封信时想从一个数据库拷贝一个名字和地址到信中，则必须记住，打印或抄在纸上；然后转到字处理程序键入数据。比较而言，Windows 可以为你拷贝名字和地址。

Windows 与任务切换器间还有另一个主要区别：在 Windows 中可在屏幕上同时看到几个任务，而用任务切换器，则只能看到激活的任务。

虽然任务切换不提供一个真正的多任务环境，但它毕竟为没有多任务功能的系统提供了一些优点。

为启动任务切换器，可以选择 Options 菜单中的 Enable Task Swapper 命令。这时 DOS Shell 屏幕改变到显示任务清单，开始它是空的(如图10.4所示)。随后你启动的每个程序都将被加到任务清单中。在图10.4中，系统已启动了三个程序。

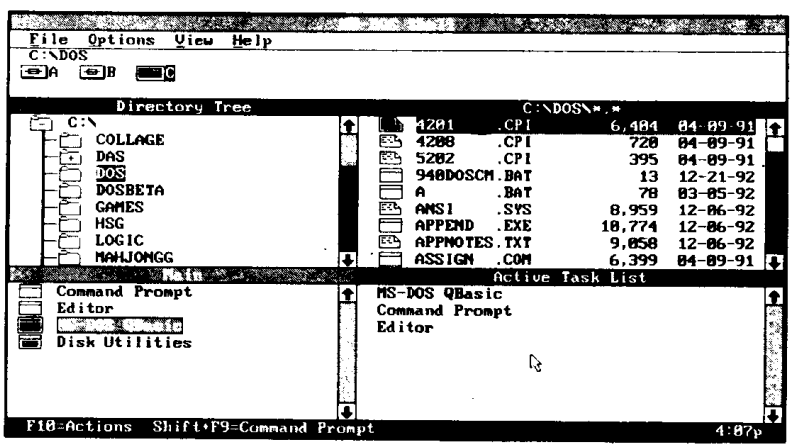

图10.4 带有任务切换器的 Shell 屏幕

现在可选择 Options，Enable Task Swapper，以使任务交换器在系统上工作。这时你可看到任务清单加到了屏幕上。

10.8.1 启动任务

为了启动一个任务，只需启动一个程序即可。这个程序自动变成一个任务且被增加到任

务清单中。当然，在一开始你并不能看到任务清单，因为你正看着程序的屏幕。在本章后面你将学会如何从一个程序转回到 Shell 屏幕。

此外也可在启动程序时按住 Shift 键，从而无需离开 Shell 屏幕即可启动一个任务。

下面试着不离开 Shell 屏幕而启动两个程序，这两个程序为 Editor 和 HELP. COM（它显示关于 DOS 命令的信息）。

1. 在双击 Editor 的同时按住 Shift，或者选择 Editor，随后按 Shift＋Enter。
   这时将出现一个对话框，它提示你输入一个文件的名字。（这时可放开 Shift）。
2. 键入\DOS\README. TXT 并按 Enter。
   对话框关闭。Editor 任务在任务清单中出现。
3. 在 DOS 目录中，当在文件清单中打开 HELP. COM 时按住 Shift。
   HELP. COM 出现在任务清单顶部。

你可多次启动相同的程序，这时每个拷贝是一个独立的任务。如可以让一个 Editor 任务工作在 AUTOEXEC. BAT 上，另一个 Editor 任务工作在 CONFIG. SYS 上，第三个 Editor 任务工作在 REPORT. TXT 上。第一个任务在清单中显示 Editor，第二个显示 Editor.（带一个点），第三个显示 Editor..（带两点）。

10.8.2　任务间切换

为从 Shell 屏幕切换到一个特定任务，只需打开任务清单中的这个任务即可。或者，如果任务旁有一个快捷键，则只需按快捷键。当你是在某个任务中工作时，可按 Ctrl＋Esc 回到 Shell 屏幕。

下面我们试着切换到 DOS 的编辑器然后返回：

1. 在任务清单中双击 Editor，或选择它后按 Enter。
   Editor 屏幕出现，同时 README. TXT 文件被打开。
2. 按 Ctrl＋Esc。
   返回 Shell 屏幕。

注释： 一些程序（如 DOS 6的 INTERSVR）禁止任务切换。但是，这些都是一些特殊程序，如果不终结这些程序而进行任务切换，将有破坏性效果或死机。

不必总通过 Shell 屏幕从一个任务切换到另一个任务。表10.1显示了直接在任务间切换的组合键。（不幸的是，没有与它们相同的鼠标动作）。

表10.1　任务切换热键

组合键	功　能
Ctrl＋Esc	切换到 DOS Shell
Alt＋Tab	切换到前一个打开的任务
Alt＋Tab，Tab，...	向前循环显示任务清单
Shift＋Alt＋Tab，Tab，...	向后循环显示任务清单
Alt＋Esc	切换到清单的下一个任务
Shift＋Alt＋Esc	切换到清单的上一个任务

提示： 要养成切换一个任务之前保存文件的习惯，以防止出现问题后无法回到这个任务。

假设你已启动五个任务，你现在工作在字处理程序下，想转到图形程序一段时间。字处理程序是任务1，图形程序是任务4。首先保存当前文件，然后按住 Alt，按几次 Tab 键，直到在屏幕上看到图形程序的名字。当看到想要的任务的名字时，释放 Alt，这个任务被打开，而且移到了任务清单的顶部。

完成了对图形程序的使用后，回到字处理程序的最快方法是按 Alt＋Tab，这种方法总会将你返回到上次打开的任务，并把那个任务移到清单的顶部。按 Alt＋Tab 就可在字处理程序和图形程序间来回切换。

注释： DOS Shell 即使不出现在任务清单中也被作为一个任务计入。它总处于清单的底部。

试着在任务 Editor 和 HELP.COM 之间进行切换：

1. 按 Alt＋Tab 回到 Editor 任务。
   Editor 出现在屏幕上。
2. 按住 Alt 后按 Tab 一次。不要释放 Alt 键。
   消息 MS-DOS Shell 出现在屏幕上。
3. 按住 Alt，再按 Tab。
   消息 HELP.COM 出现在屏幕上。
4. 释放 Tab 键以切换到 HELP.COM 任务。
   这时 HELP 屏幕占满了显示器（现在不要考虑这个屏幕的内容。它将在第12章解释）。
5. 按 Alt＋Tab 切换到 Editor 任务。
6. 再按 Alt＋Tab 切换到 HELP 任务。
7. 按 Ctrl＋Esc 切换到 Shell 屏幕。

提示： 当任务清单很长时，可以通过按 Shift＋Alt＋Tab，Tab，...Tab 在任务清单中向后循环浏览，以到达你想到的任务。

答疑： 我试着切换任务，但计算机挂起，现在我不能做任何事情。现在必须重新启动机器。

你可能需要为显示卡安装正确的文件。拿出 DOS 程序盘，寻找文件 MONO.IN_，EGA.IN_，VGA.IN_ 等等。它们可能没有赶上升级步伐；你可能需要用到那些你用来安装或升级到 DOS 6.0的磁盘。

在试图安装一个抓钩（grabber）文件之前，将当前 DOSSHELL.GRB 文件更名为 BACK.GRB。如果安装了一个新的抓钩文件不能工作，再更名到 DOSSHELL.GRB 回到系统的原来状态。

当为显示卡找到这些抓钩文件后，试着找出正确的一个（如果不确定，一个一个地试不会有什么危害）。为了安装一个文件，键入下列命令，该命令中的 filespec 指你想安装的文件名，如 EGA.IN_：

EXPAND A:\filespec C:\DOS\DOSSHELL.GRB

然后再试着切换任务。

如果还不能切换任务，需要从 Microsoft 获得帮助。参看《DOS 用户指南》，其中有如何获得技术支持的指导意见。

在表10.1中，你可看到 Alt+Esc 和 Shift+Alt+Esc 也可在任务清单中循环。但是，它们与 Alt+Tab 和 Shift+Alt+Tab 相比有两个重要区别：

- 若使用 Esc 键，则当循环时每个当前任务都打开，从而可以看到它的屏幕。这会花更多的时间，但如果你记不住任务的名字时它是好的方式。
- Esc 键不会改变任务的顺序，虽然整个任务清单是被循环显示，从而当前任务出现在清单的顶部。

如果激活的任务有一个快捷键，可通过按这个键直接从其他任务转到这个任务。如果两个或多个激活的任务有相同的快捷键，按这个快捷键只能在这些任务中循环。但是，如果当前在屏幕上的程序正用这个组合键完成它自己的功能，则快捷键不会工作。

预览： 第11章将讲述如何定义快捷键。

10.8.3 结束任务

当结束一个程序时，DOS Shell 关闭该程序所对应的任务并将它从任务清单中清除。如果不使用任务切换，按你平时退出程序的方法退出。

按以下步骤终结 Editor 任务：

1. 切换到 Editor 任务。

   Editor 屏幕出现。

2. 按 Alt+F，然后按 X。

   Editor 关闭，系统回到 Shell 屏幕。注意现在 Editor 从任务清单中清除。

注释： 必须结束所有的任务后才能退出 DOS Shell。但是，不必关闭任务交换功能。如果常使用任务切换，只需始终让它处于打开即可。

有时可能有一个程序会挂起。这就是说，它不能正确地完成任务。在大部分情形下，它完全无法实现其正常功能，你甚至不能正常终止任务。这时最常用的解决方法是重新启动机器，从而杀掉这个功能不正常的任务。如果只运行一个程序，这不会有任何不良后果。但是，如果任务切换正常，其他任务已被启动，那么就不要通过重新启动杀掉这个功能不正常的任务，除非没有其他选择。重新启动系统会终结所有的当前任务，这样就不能正常保存和关闭它们的文件，你可能会丢失一些数据。

在 DOS Shell 中任务切换时如何终结一个功能不正常的程序

1. 如果可能的话，返回到 Shell 屏幕。
2. 选择功能不正常的任务。
3. 按 Del 删除这个任务。(出现一个警告框)。
4. 在警告框中选择 OK。
5. 正常关闭所有其他的任务，当然这是指在可能的情况下。
6. 退出 DOS Shell。
7. 重新启动。

如果真的需要取消一个任务，可从 Shell 屏幕终止它。随后，所有其他程序都被挂起。这时应该尽快将它们全部终止，然后退出 Shell 并尽快重新启动计算机。

10.8.4 结束任务切换器

在将其关闭之前，任务切换功能将一直保持，即使你退出 Shell 并重启动也是如此。因为任务切换器会占用内存空间，当不需要它时，你可能想关闭它。要在你的系统中关闭任务切换，首先应终止所有的任务。在关闭任务切换功能时请按以下步骤进行：

1. 打开 HELP.COM 任务。
   HELP 屏幕出现。
2. 按 Alt+F，然后按 X 以退出当前程序。
   消息 Press any key to return to MS-DOS Shell 将出现在屏幕上。
3. 按任意键。
   Shell 屏幕出现。
4. 在 Options 菜单中选择 Enable Task Swapper。
   注意在 Enable Task Swapper 旁的箭头，它表示本项功能当前起作用。当再选择一次时，菜单消失，活动任务清单也从屏幕消失。

10.9 自 测 练 习

如果你想进行更多在 DOS Shell 下的运行程序练习，可试试以下活动：

1. 设置任务切换。
2. 通过加载数据文件而启动字处理程序(或另一个应用程序)。
3. 切换到 Shell 屏幕(不要退出字处理程序)，然后用一个不同的数据文件启动这个程序的另一个拷贝。
4. 实践一下在这两个任务间切换。
5. 关闭所有的任务然后关闭任务切换。

第11章 自定义 Shell

简介： 你可依照自己的习惯设计 Shell 屏幕，即将它裁剪修改以符合自己的工作风格。

11.1 总 览

到目前为止，我们一直是在标准 Shell 屏幕下工作。但裁剪屏幕可使工作更舒适，例如可定义自己的程序项、修改显示模式和颜色。

本章主要讲述：

- 定义和使用用户自己的程序组
- 定义和使用用户自己的程序项
- 从一个组拷贝项到另一个组
- 删除程序组和程序项
- 改变视频显示模式
- 为 Shell 屏幕选择不同的颜色模式
- 在文件清单中筛选文件名
- 改变文件清单的排序方式
- 在硬盘上找到某个特定文件或一组文件的位置
- 在一组软盘中寻找文件的位置
- 在文件清单中显示隐含和系统文件

11.2 管理程序清单

程序清单是启动程序最容易和最灵活的方式。当你第一次安装 Shell 时，程序清单包含许多项，这些项被分成两组：Main 组（在屏幕上可见）和 Disk Utilities 组。如果 Shell 是主要 DOS 界面，则你可能会希望在这个清单里增加自己的组和项。你甚至还会重新定义 Microsoft 提供的项。

假设你管理三个数据库：CUSTOMER. DB，ORDERS. DB 和 SUPPLIER. DB，并且偶尔工作在几个其他数据库上，那么你可能想建立一个带有几项的独立的 Database 组，该组包含的项如下：

- Customers database（打开 CUSTOMER. DB）
- Orders database（打开 ORDERS. DB）
- Suppliers database（打开 SUPPLIER. DB）
- Other database（让你在对话框里键入想要的数据库文件）

可以为经常使用的主要字处理程序、表格处理程序和其他应用程序建立几个类似的组。

11.2.1 在组下工作

程序清单一次只能显示一个组。前面已看到过 Main 组，现在让我们看一看 Disk Utilities 组。注意 Disk Utilities 有一个与命令提示符、Editor 等不同的图标。这个图标表示现在是一个组代替了一个程序项。下面试着打开 Disk Utilities 组：

1. 打开名为 Disk Utilities 的组。
   程序清单切换到 Disk Utilities 组。这时标题 Disk Utilities 出现在程序清单的标题栏，其中的程序项包括 Disk Copy 和 MS-Anti-Virus 之类的程序。在以后章节我们会学到如何使用这些程序。
2. 保持此组打开，同时阅读以下章节的内容。

注意在 Main 清单中的第一项，它有组日志。现在打开 Main 以回到 Main 组。

组的组织方式与磁盘上的目录相似。Main 组是最高层的组，它不能被删除或更名。Main 组可以有子组（如 Disk Utilities 组），子组可依次有自己的子组。每个子组清单的第一项总是父组，这样你可回到父组。

11.2.2 定义一个组

当屏幕显示为程序清单时，菜单选项 File，New 用来创建一个新组或新程序项，新项目被增加到当前组中。

答疑： 找不到 DOS Shell 的 File，New 选项，它们不在 File 菜单中。

DOS Shell 用它的菜单来装配激活的清单。New 用来为程序清单定义项目，只在屏幕显示为程序清单时它才出现在 File 菜单中。激活程序清单就会发现这个命令。

如何创建和管理 DOS Shell 程序组

创建新组：

1. 打开将成为新组的父组的组。
2. 选择 File，New（程序清单必须处于活动状态）。
3. 在第一个对话框中，选择 Program Group。
4. 在下一个对话框中，键入一个标题，最多23个字符。
   如果愿意，可以键入一些帮助文本（最多255个字符）。也可以定义一个口令（最多20个字符）。
5. 选择 OK。

修改一个组定义：

1. 选择程序组（在程序清单中）。
2. 选择 File，Properties。可能需要键入一个口令。（Program Group Properties 对话框显示了组的定义。它看起来与 Add Group 对话框相同）。
3. 改变定义。
4. 选择 OK 关闭对话框。

删除一个程序组：

1. 打开组。
2. 删除其中所有程序项和子组。(不要试图删除父组名)。
3. 回到父组。
4. 选择子组，不要打开它。
5. 按 Del 键。
6. 在确认框中选择 Delete This Item，然后选择 OK。

下面我们试着创建一个新组。我们将创建一个 Practice 组，以作为 Main 组的子组。为创建一个新组，按以下步骤操作：

1. 保证屏幕显示为程序清单且 Main 组打开。
2. 选择 File，New。

   New Program Object 对话框打开，其中有两个选项：Program Group 或 Program Item。

注释： New Program Object 对话框中的这种选项叫做选项按钮，因为必须有一个而且只有一个选项被选中。在圆中的黑点表示已选中的项。若希望用鼠标选择一项，可单击它。用键盘选择时，先按 Tab 到标有 New 的框中，然后按上下箭头键移动黑点。

3. 选择 Program Group，按 Enter。

   Add Group 对话框出现(如图11.1所示)。

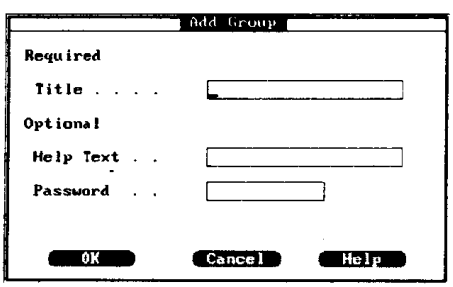

图11.1　Add Group 对话框

你必须为这个组提供一个名字。所有其他项是可选的。名字最多为23个字符，其中包括空格。在这里输入的任何东西将出现在程序清单中。为提供名字给这个组，请按以下步骤操作：

1. 在 Name 框中键入 Practice。
2. 按 Enter。

   对话框关闭，Practice 组出现在程序清单中。

如果其他人想使用新组的话，可能还需要提供帮助题目。帮助文本最多255个字符长，包括空格，DOS Shell 在有人选择这个组名并按 F1时显示帮助题目。

提示： 在帮助文本中开始新的段时不要按 Enter。按 Enter 会关闭对话框。可键入(^)后按一次 M。两个^M 代码会在段间插入一空行，它使段间分隔更明显。

在输入帮助文本过程中，如果不小心关闭了对话框，那么可在程序清单中选择这个新项，然后按 File Properties 重新将它打开。

此外也可以分配一个口令（最多20个字符）给新组。当一个组被口令保护时，DOS Shell 在任何时候有人试图打开或修改组时会显示 Password 对话框（如图11.2所示），这时必须输入正确的口令才能继续。

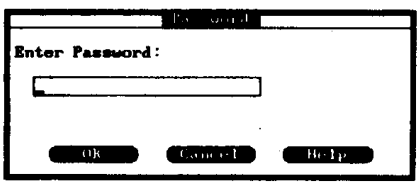

图11.2 Password 对话框

警告： DOS Shell 的口令保护并不十分有力。一个有经验的用户可通过检查 DOSSHELL.INI 文件找到口令。（如果忘记了口令也能从中找到）。

口令是区分大小写的。例如，如果口令是 Jane's Coffee Shop，那么键入 Jane's coffee shop 或 JANE'S COFFEE SHOP 就不能工作。这是 DOS 区分大小写有限的几处之一。

你可能不会有太多理由去修改一个组定义，但如果需要的话，可以打开相应组的父组，并选择这个组（但不要打开它），然后选择 File，Properties。Program Group Properties 对话框（很像 Add Group 对话框）显示当前定义，从而可在这里修改它。

11.2.3 定义一个程序项

现在已有了一个新组，你可能想增加一些项到其中。图11.3显示了 Add Program 对话框，在这里可定义一个程序项，其中 Program Title 和 Commands 是必需定义的，而其他域则是可选的。

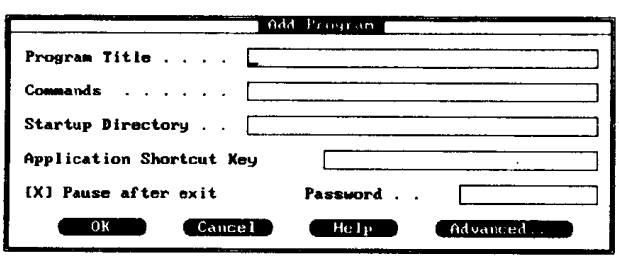

图11.3 Add Program 对话框

如何创建和管理一个 DOS Shell 程序项

定义一个新程序项：

1. 打开将包含这个新项的组。
2. 选择 File，New。（程序清单必须被激活）。
3. 在第一个对话框中选择 Program Item，然后选择 OK。
4. 在下一个对话框中键入一个程序标题（最多23个字符）和启动命令（最多25个字符）。

5. 键入一个启动目录并定义一个快捷键，然后关闭 Pause After Exit，或如果想要的话输入一个口令。
6. 如果想输入帮助文本，则可选择 Advanced，并在 Advanced 对话框输入帮助文本，然后选择 OK 回到 Add Program 对话框。
7. 选择 OK 关闭 Add Program 对话框。

修改程序项的定义：

1. 选择项。
2. 选择 File，Properties。这时可能不得不输入一个口令。(Program Item Properties 对话框打开并显示该项的定义)。
3. 按需改变定义。
4. 选择 OK。

删除一个项：

1. 选择这个项，但不要将其打开。
2. 按 Del 键。
3. 在确认对话框中选择 Delete This Item，然后选择 OK。

拷贝一个项：

1. 选择这个项。
2. 选择 File，Copy(程序清单必须被激活)。
   这时在状态栏有一个提示会告诉你按 F2完成拷贝。
3. 打开目标组。
4. 按 F2。

程序标题最多有23个字符，其中包括空格。在这里键入的任何东西都出现在程序清单中。Commands 框用于定义一个命令，以便在有人打开这个项时可将此命令提交给 DOS 运行。例如，为了启动 Microsoft Word，可以在 Commands 框中键入 WORD。在 Commands 框中你最多可键入25个字符，其中包括空格。

为增加一个项到 Practice 组中以显示系统内存使用情况，可按以下步骤操作：

1. 打开 Practice 组。
   程序清单只包含一项(Main)，该项可以使你回到 Main 组。
2. 选择 File，New。
   现在 New Program Object 对话框被打开。缺省时 Program Item 选项被选择。
3. 按 Enter。
   这时屏幕上出现 Add Program 对话框。
4. 在 Program Title 框中键入 Display Memory。
5. 单击 Command 对话框或用 Tab 将光标移到其上。
6. 在 Commands 框中键入 MEM。
   这是运行 DOS 的 MEM 程序的命令。
7. 选择 OK。

对话框关闭，Display Memory 出现在程序清单中。

提示： 在 Commands 框中的文件描述中请务必包含一个完整的路径，包括驱动器名字，因为当程序项被打开时你不知道什么驱动器或目录将被激活，但如果键入一个完整的路径，则可保证 DOS 能发现这个文件。
但是，如果为程序清单指定一个启动目录，那么可以忽略这个规则。若想要的文件在启动目录中，则不必为它提供路径名。(启动目录将在下节讨论)。

现在试运行一下新的程序项：

1. 打开 Display Memory 项。
   这时 Shell 屏幕消失。MEM 在屏幕上显示系统的内存分配情况，同时在屏幕底部有一个提示为 Press any key to return to MS-DOS Shell。
2. 按任意键。
   系统返回 Shell 屏幕。

如果在 File，New 对话框中指定了一个启动目录，那么 DOS Shell 在启动程序前将转到该目录。如果启动一个用于维护诸如 EDIT，或者字处理程序，抑或数据库之类文件的程序，那么最好把应用程序用以保存数据文件的目录设置为启动目录。这使你在使用这个应用程序时打开和保存文件更容易。启动目录选项在 Shell 中改变了当前目录，有些人觉得这是一件令人烦恼的事情。

Application Shortcut Key 域是为任务切换使用的，这在第10章“从 DOS Shell 运行程序”中已有过介绍。为分配一个快捷键给这个项，可以把光标放在文本框中，然后按你想使用的组合键。快捷键必须使用 Ctrl，Shift 和/或 Alt 加一个字符键、功能键或光标键(但不能是箭头键)。Shell 将拼出你按键的组合。例如，如果按 Ctrl+E，表达式 Ctrl+E 出现在文本框中。如果没有东西出现，说明键组合不是合法的快捷键，这时你需要试另一个。缺省条件下，Pause After Exit 框是打开的，这使 DOS Shell 在程序终结时会显示信息 Press any key to return to MS-DOS Shell。如果关闭这个框，Shell 屏幕在程序终结就自动返回。关闭这个框去掉了一个额外的步骤，但要保证程序在结束之前不会显示一些需要你阅读的消息。如果发现不能阅读你想要的所有信息，可以再打开此框。

在 Password 框中可以键入最多20个字符，以便用口令保护程序项(有时它很脆弱)。

此外也可以提供帮助文本。选择 Advanced 按钮将打开另一个对话框，该框包含着与 Help 文本框一起的增强选项。在 Help 文本框中最多可输入255个字符，其中包括空格(使用 ^ M 开始一个新的段落)。

预览： 第36章“命令行状态下的 DOS 应用”，显示如何在 Advanced 对话框中使用其他项目，以及如何创建一个对话框(如为 Editor 程序项创建的那个对话框)。

如果以后发现需要修改项的定义，那么请先打开这个项的父组，然后选择这个项(但不要打开它)，并选择 File，Properties。Program Item Properties 对话框显示的是一个项的当前定义，它与 Add Program 对话框相似，可以在这里修改项定义。

11.2.4 拷贝程序项

假设已在 Shelly 组中创建了一个名叫 Word Processor 的项，现在想在 Donna 和 Charles

组中创建相同的项，那么可通过拷贝来代替两次或更多次定义过程，这样可节约时间和减少麻烦。即使项目在三个组中的定义略有不同(例如不同的口令和启动目录)，拷贝和修改这个项目也比从头开始定义快得多。

注释： File,Copy 菜单项的快捷键 F8在程序清单中不能使用。

为拷贝 Display Memory 项到 Disk Utilities 组，可按以下步骤操作：

1. 打开 Practice 组，选择 Display Memory 项(但不打开该项)。
2. 选择 File,Copy。
   在状态栏出现提示 Display the group to copy to ,then press F2. Press Esc to cancel。
3. 返回到 Main 组并打开 Disk Utilities 组。
   提示仍然保留在状态栏中。
4. 按 F2。
   Display Memory 项目出现在 Disk Utilities 程序清单中。提示从状态栏中消失。

程序清单没有相应的 File,Move 命令。如果你希望移动一个项而不进行拷贝，只须回到原来的组并将该项删除即可。

既然已学会创建和管理程序组和程序项，在学习新题目前我们先来删除已建立的项目。要删除已创建的项目，可按以下步骤操作：

1. 在 Disk Utilities 组中，选择 Display Memory 项目，按 Del 键。
   Delete Item 对话框打开。
2. 选择选项1，Delete this item。
   Display Memory 项目将从清单中删除。
3. 回到 Main 组。

在删除组之前它必须为空，这样必须在 Practice 组中删除它的项。

1. 打开 Practice 组。
2. 删除 Display Memory 项。

Practice 组还包含 Main，但不能删除这个父组。在删除当前组时 Main 将自动删除。

1. 返回 Main 组。
2. 删除 Practice 组。

提示： DOS Shell 将显示模式和屏幕颜色的选择保存在一个叫 DOSSHELL. INI 的文件中。DOSSHELL. INI 用于在每次启动 Shell 时初始化屏幕。当找到你喜欢的显示模式和颜色后，Shell 将始终用这些选项进行启动。
如果已经在 DOSSHELL 启动命令使用/T 开关(文本模式)或/B 开关(黑白颜色模式)，那么可以在 Screen Display Mode 对话框和 Color Scheme 对话框选择相应的选项。DOS Shell 将记住已作出的选择，以后将不再需要使用这些开关。

答疑： 我已经自定义了 DOS Shell 程序清单，但现在它回到缺省清单。如何才能回到我自己的定制清单？

对 DOS Shell 屏幕作的任何修改，如新的程序组和项目，都被保存在一个名叫 DOSSHELL. INI 的文件中。如果当 DOS Shell 启动时不能找到这个文件，它将回到它的缺省设置。

如果你(或其他人)移动了 DOS SHELL. INI，那么需要把它移回 DOS 目录。若已删除了 DOSSHELL. INI，则要么取消删除，要么从它的最新备份中恢复。如果不曾删除或改变 DOSSHELL. INI，它可能以某种方式被破坏了，应该从最新备份中将它恢复。

如果不能从一个最近的备份中恢复或者取消删除，那么将永远不能回到修改后的清单。

答疑： 当退出 DOS Shell 时得到消息 Unable to update DOSSHELL. INI。

DOS Shell 在你每次退出时在 DOSSHELL. INI 中保存它的当前设置。如果它不能发现 DOSSHELL. INI，或者如果文件已被标志为隐含、系统或只读，抑或如果磁盘已满，那么 Shell 将显示这个消息。在这种情况下，当你退出时，对程序清单、屏幕显示模式和颜色作的任务变化将被忘却。

请确保 DOSSHELL. INI 在 DOS 目录中，而且它的只读、隐含和系统属性被关闭。同时还要保证在驱动器上有一些自由空间。这将解决上面这个问题。

如果 DOSSHELL. INI 在正确的位置，而且没有写保护，但还得到错误消息，那么可能是由于 DOS 目录不在程序的搜寻路径中。在 AUTOEXEC. BAT 文件中检查 PATH 命令，它应该包括 DOS 目录，可能有一些其他目录。如果未包括 DOS 目录，则应增加一个分号，并在命令的结尾加入正确的路径名，如下所示：

```
PATH=other-path-names;C:\DOS
```

如果 DOS 目录出现在 PATH 命令中，则可能是 PATH 命令太长(它不能超过127个字符)。如果这样，你应该从中消去一些目录。第12章“在命令提示符下工作”解释了 PATH 命令。

11.3 选择显示选项

现今的显示器每屏幕可显示的行数远多于旧标准的25行，但许多人并没有充分利用这个增强能力。接下来将介绍更多关于屏幕的信息，这样在寻找所需项目时，翻滚动作会更少，同时一个更长的清单还能使你在相同的目录下拖一个数据文件到一个程序文件更容易。如果两个文件一起出现在屏幕上，则不必翻卷就可找到程序文件。

图11.4显示了当选择 Options，Display 时打开的 Screen Display Mode 对话框。这个对话框显示了显示器的选项。你可以尝试这些选项，以找到最好的一种。

在显示器上试几种选项：

1. 选择 Options，Display。

   Screen Display Mode 对话框将打开。

2. 选择想试的选项。

3. 选择 Preview。

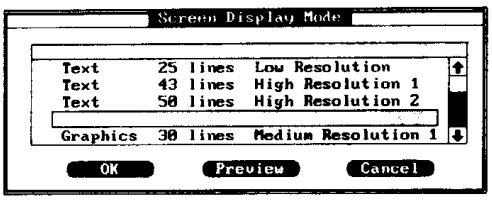

图11.4 Screen Display Mode 对话框

这时屏幕改变到显示已选择的模式，但对话框保持打开。

4. 重复2和3，直到你已决定使用哪个选项。
5. 选择 OK。

对话框关闭，屏幕改变到新的显示模式。

11.4 定制屏幕颜色

图11.5显示了 Color Scheme 对话框，该对话框在选择 Options，Colors 时打开。Shell 提供了多种颜色模式，包括黑和白。在你的显示器上有些颜色组合的显示效果可能要比其他组合更好。你可使用 Preview 按钮来反复尝试以找到你最喜欢的一种。

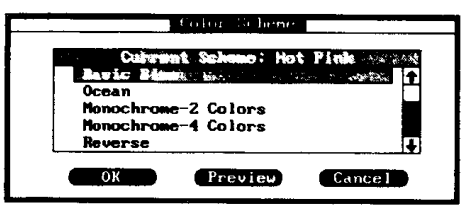

图11.5 Color Scheme 对话框

在显示器上试一些选项：

1. 选择 Options，Colors。

ColorScheme 对话框打开。

2. 选择你想试的一个选项。
3. 选择 Preview。

屏幕改变到显示选择的颜色，但对话框保持打开。

4. 重复2和3直到决定了你想使用的选项。
5. 选择 OK。

对话框关闭，屏幕改变到新的选择。

注释： 改变 DOS Shell 的显示模式和颜色既不影响从 Shell 启动的任务程序，也不影响退出 Shell 后的命令提示符下的屏幕。

11.5 定制文件清单格式

到目前为止，我们一直在缺省的文件清单下工作：除了隐含和系统文件外的所有文件均被显示，它们按字母表顺序排列。然而，你可裁剪文件清单显示文件的一个不同的子集，并且使它们按不同顺序排列。图11.6显示了在选择 Options，File Display Options 时打开的 File Display Options 对话框。

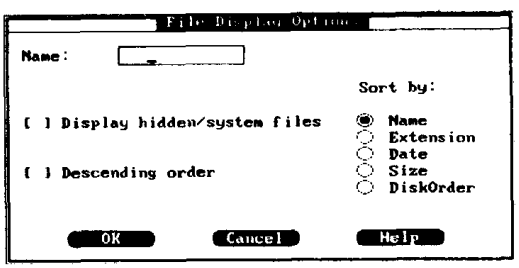

图11.6 File Display Options 对话框

11.5.1 筛选文件清单

有时你未必需要显示一个目录中的所有文件。为此，可以通过在 Name 框中输入一个名字来筛掉与这个名字不匹配的任何文件。若要在当前目录中只显示.TXT 文件，可按以下步骤执行：

1. 选择 Options，File Display Options。
   File Display Options 对话框打开。
2. 键入 *.TXT 并按 Enter。
   对话框关闭。文件清单的标题栏上显示 C:*.TXT(以前它总是显示 C:*.*)。取决于哪个目录是当前目录，文件清单可能显示一些.TXT 文件，或者可能显示 No files match file specifier。

11.5.2 显示驱动器上的所有文件

可将一个文件筛选器与 View All Files 结合起来，以显示一个驱动器上某种特定类型的所有文件。例如，假设想删除驱动器上所有.BAK 文件，可以输入 *.BAK 文件到 Name 框中，然后选择 View All Files，以显示驱动器上所有的.BAK 文件。按 Ctrl+/可以选择所有这些文件，而不选择驱动器上的其他文件，因为你只能选择列出了的文件。这样，按 Del 就可将它们全部删掉。

提示： 为了这个操作，可能需要将 Confirm on Delete 选项关闭。如果这样做了，请在操作完成后马上重新将它打开。

为显示驱动器 C 上所有的.TXT 文件清单，可按以下步骤操作：

1. 确保只显示所有的.TXT 文件。

2. 选择 View,All Files。

屏幕改变到 All Files 显示。你可能会得到一个在整个硬驱上的多个. TXT 文件的一个清单,同时也可能会有多个名字均为 README. TXT 的文件,因为许多软件制造商都会把这样一个文件包含在它们的应用程序中。

注释: 改变驱动器和视图不会影响筛选器,但终止 DOS Shell 时,系统将重又回到缺省的 *. * 筛选器。

11.5.3 找到不知道位置的文件

将 View All Files 和一个文件筛选器结合使用可以找到不知道位置的文件。假设有一个. TXT 文件,但你不记得它的确切名字或位置,那么可以在屏幕清单上将它找出来。

如果你想找到一个以 JUNE 开头而不知文件名的剩下部分的文件,可以输入 JUNE *. * 作为文件筛选器,以显示所有名字以 JUNE 开头的文件。

甚至可使用一个具体的文件名作为筛选器。下面我们试着显示在硬驱上的所有 README. TXT 文件。

1. 选择 Options,File Display Options。

File Display Options 对话框打开。

2. 键入 README. TXT 按 Enter。

文件清单显示硬驱上的所有的 README. TXT 文件。你可选择某个文件并考察左边的信息栏,以确定该文件属于哪个目录。

有时容易忘掉有一个文件筛选器在起作用。如果你不能找到你知道在当前目录中的一个文件,或者如果消息说目录非空,而没有文件清单显示,那么请检查一下标题栏,看看是否正筛选这个清单。如果它不是 *. * 而是其他东西,那么目录可能有看不见的文件。

提示: 目录有看不见的文件的一个真正线索是消息 No files match file specifier in the file list,这个消息说明目录包含文件但它们没被筛出来。当一个目录真的为空时,相应消息将是 No files in selected directory。

11.5.4 显示隐含和系统文件

对于带有隐含和系统属性的文件通常不能进行诸如删除、移动和拷贝这类的操作。因为这个原因,DOS Shell 缺省状态下在文件清单中并不显示它们。但是,有时你也许希望可以显示它们,特别是试图删空一个目录时必须将这些文件删掉。

在 File Display Options 对话框中,你可选择 Display Hidden/System Files,以便将上述文件包含在文件清单中。再取消检验此框会隐藏它们。记住,如果一个文件没有显示,就不能选择它。但要注意,有些程序可以访问和处理那些在文件清单中看不见的文件。

在根目录中有几个隐含和系统文件。为观察这些文件,可执行以下步骤:

1. 选择 View,Program/File Lists。

屏幕回到正常的显示模式。

2. 选择根目录。

3. 选择 Options,File Display Options。

File Display Options 对话框打开。

4. 在 Name 框中键入 *.SYS。

5. 选择 Display Hidden /System Files。

一个 X 出现在检验框中。

6. 按 Enter。

文件清单将显示 IO.SYS 和 MSDOS.SYS。它们是 DOS 核心程序文件。

警告： *千万不要在根目录下移动、删除、更名或破坏任何隐含和系统文件，因为这可能使 DOS 不能工作或使存取硬盘上的文件变得困难。*

如前所述，当不可见的文件存在于当前目录中时会出现消息 No files match file specifier。如果当文件筛选器是 *.* 时得到这个消息，不可见文件可能有隐含和/或系统属性。

11.5.5 在一组软盘中定位文件

假设存储了一个名叫 MISSING.TXT 的文件在一组软盘的一张中，现在忘记它在哪个盘上。如何才能找到这个文件呢?答案是首先设置筛选器为 MISSING.TXT，然后把第一张盘放在驱动器 A 中，选择驱动器。必要时更新这个驱动器，以强迫 DOS 读它的目录。

继续插入软盘，并按 F5以更新驱动器，直到 MISSING.TXT 出现在文件清单中。

11.5.6 给文件清单排序

缺省状态下，DOS Shell 按字母顺序升序显示文件清单，但存在多种组织文件清单的其他方式。在 File Display Options 对话框中，可选择按扩展名(而非文件名)字母顺序，或者按日期和时间，或者按文件大小顺序排列文件清单。甚至 DiskOrder 选项有时也可用，这个选项在文件清单中按文件存在磁盘上时实际出现在目录上的顺序排列。

试着按日期顺序排列当前显示文件。为做这件事，可在恢复到正常的筛选器和从当前显示中移去隐含和系统文件后，按以下步骤操作：

1. 选择 Options,File Display Options。

File Display Options 对话框打开。

2. 在 Name 框中输入 *.* 。

3. 选择 Display Hidden/System Files，以便从选择框中删去 X。

4. 选择 Date 选项按钮，按 Enter。

对话框关闭，文件清单按日期递增的顺序显示。

Descending Order 选项将使清单的顺序相反。例如，如果选择 Size 和选择 Descending Order 框，文件清单将从最大的文件显示到最小的文件。如果颠倒 Date 顺序，文件将从新到旧显示文件。

如果用 View Dual File Lists 显示两个文件清单，两个清单将受显示选项影响。当退出 DOS Shell 时，缺省文件筛选器恢复，但其它显示选项将被记住，并在你下次启动 DOS Shell 时发生作用。

在本章结束之前，回到文件清单的正常视图。要返回正常视图，按以下步骤操作：

1. 选择 Options，File Display Options。
   File Display Options 对话框打开。
2. 选择 Name 选项按钮，按 Enter。
   对话框关闭。文件清单根据文件名按字母表中字母顺序显示所有文件(除了隐含和系统文件)。

11.6 自 测 练 习

以下各项将帮助你实践按自己的方式裁剪 DOS Shell：

1. 创建名为 MYGROUP 的程序组。
2. 从其他组拷贝两个项到 MYGROUP 中。
3. 创建一个启动字处理程序的项。
4. 删除 MYGROUP。
5. 在驱动器上确定. CPI 文件的位置。
6. 恢复屏幕到正常视图。

第三部分 DOS 命令

第 12 章 在命令提示符下工作

简介： 在命令提示符下工作与使用 Shell 屏幕有很大区别。本章将介绍一些基础知识。

12.1 理解命令

在命令提示符下可以做许多在 Shell 中不能做的事情。例如，DOS 6 包含一个 DELTREE 命令可以删除整个目录分枝，而不需要所有的目录首先为空。DOS 实用程序如 FORMAT，UNFORMAT，UNDELETE，DEFRAG 和 SCANDISK 都以命令形式启动。实际上，最后两个实用程序只能在命令提示符下运行，从来不能从 Windows 或 Shell 中运行，因为它们会破坏打开的文件。

当知道如何使用命令后，就可以为 Shell 的程序清单或那些可从 Shell 的文件清单中启动的批程序创建你自己的程序项，从而可以在命令提示符下运行这些程序或者把它们拷给其他人。

另外，你可理解从别人那儿得到的批处理程序。你可以理解你自己的 AUTOEXEC.BAT 程序，它在每次启动时运行，并帮助设置系统——这是一个优点，因为它对系统很重要。无论何时在 PC 下工作，理解系统发生的事越多，情况会越好。

本章主要讲述：

- 访问命令提示符
- 读一个命令格式描述
- 重定向程序输出
- 用命令获得屏幕帮助
- 定义一个程序搜寻路径
- 设计自己的命令提示符

在以后章节，你将学会如何使用特定命令如 DELTREE，以及如何创建和使用批处理程序文件如 AUTOEXEC.BAT。

12.2 到达命令提示符

你可用几种方法到达命令提示符，每种有不同的效果。最原始的方法当然是启动或重启动计算机。如果 AUTOEXEC.BAT 文件不启动 Windows 或 DOS Shell(或其他程序)，命令

提示符出现。

如果 AUTOEXEC.BAT 文件启动了另一个程序，则必须终结这个程序以到达命令提示符。如果一个程序现正在计算机中运行，退出这个程序就可到达命令提示符。要退出这个程序，按以下步骤：

1. 如果是在 DOS Shell，那么你可以关闭任意正运行的程序，并且按 F3 以退到命令提示符。
2. 如果在 Windows 中，则需关闭所有正运行的程序，并且选择 File，Exit from Program Manager 的菜单栏。
3. 如果正使用其他程序，则需关闭打开的所有文件，正常退出程序。

通过这种方法到达的命令提示符是主要命令提示符。因为在 DOS 上操作，所以这样讲。正如本书即将讲述的，你可以在这个命令提示符下做你不能在其他命令提示符情形下完成的工作。

12.2.1　解释命令提示符

命令提示符通常包含当前目录的路径名，后面跟一个大于号。例如，下面的命令提示符告诉你驱动器 C 的根目录是当前目录：

```
C:\>
```

在本章中，你将处于驱动器 C 的根目录。如果命令提示符显示另一个目录是当前目录，按以下步骤改变到 C:\：

1. 键入 C:按 Enter。(C 后面的冒号很重要，一定要键入它)。
   DOS 转到驱动器 C 并显示一个新的命令提示符。在新的命令提示符中你应该看见驱动器字母 C。

注释： 本书假设驱动器 C 是 DOS 启动驱动器。如果不是，要保证使所有的例子适应于新的 DOS 启动驱动器。

2. 转换到根目录，可输入 CD\，按 Enter。(保证在 CD 后键入空格后再键入反斜杠，而不是斜杠)。
   DOS 转到根目录并显示一个新的命令提示符。命令提示符现在为 C:\>。

答疑： 命令提示符不显示当前驱动器和目录，它显示其他的信息，或者根本什么都不显示。

有人已修改过命令提示符。要学会适应新的命令提示符，或键入 PROMPT PG 到命令提示符下，随即按 Enter 以将它改成缺省提示符。(在命令参考中查找 PROMPT 命令，看看关于该命令的解释)。

12.2.2　辅助命令提示符

许多程序允许临时地在命令提示符下运行一个或两个命令，随后返回到程序原来的位置。例如，Windows 在 Applications 组中提供了一个 MS-DOS Prompt 程序项，用户可在 Main

组中打开 Command Prompt 项或按 Shift+F9 快捷键。当按这种方式到达一个命令提示符时——在另一个程序仍旧打开的情况下——这个命令提示符是一个辅助命令提示符。

无论从什么程序切换到命令提示符，总可以从命令提示符下按相同的方式返回这个程序。只需不带参数地键入 EXIT，命令提示符就随即消失，系统返回到前面的程序。

在辅助命令提示符下输入命令，通常和主要命令提示符一样，但有一些重要的例外。

- 不要从辅助命令提示符加载任何 TSR 程序。这样做会导致以后用完内存或使一个程序挂起。这也是为什么第 9 章告诉你要先退出 DOS Shell 后再初始化 PRINT 程序，然后再启动 Shell 的原因所在。
- 不要运行直接处理 FAT 和磁盘目录结构及需要所有文件被关闭的程序。当前处于暂停的程序如 DOS Shell 可能引起几个文件处于打开状态。由于这个原因，几个 DOS 实用程序不能在辅助命令提示符下运行。例如，CHKDSK 和 SCANDISK 程序（在第 27 章“CHKDSK 和 SCANDISK”中讨论）分析了 FAT 和目录结构的错误和不一致性以及如何修复它们，这些程序可能破坏打开的文件。还有另一个例子，DEFRAG（在第 33 章“去除文件碎片”）重组簇中的数据以更有效地利用硬盘空间，这个程序也会破坏打开的文件。

答疑： 虽然对磁盘作了一些改变，但以上修改在 DOS Shell Directory Tree 或 File List 中显示不出来。

你可能从一个辅助命令提示符或应用程序中作了以上修改，DOS Shell 不知道它们。按 F5 强迫 DOS 重读当前驱动器，所作修改将显示在 Shell 屏幕上。

12.2.3 File，Run 命令

许多应用程序提供了一种访问辅助命令提示符的方法，这个命令提示符足够长，可运行一个命令。图 12.1 显示了 DOS Shell 的 Run 对话框，当你选择 File，Run 时它被打开。你将在许多应用程序中发现相同的特征。当在框（最多 127 个字符）中键入一个命令时，DOS Shell 打开一个辅助命令提示符，运行这个命令，然后显示消息：Press any key to return to MS-DOS Shell。

如何从 DOS Shell 中的辅助命令提示符下运行一个命令

1. 选择 File，Run。
2. 键入命令到 Run 对话框中。
3. 按 OK。
4. 当程序终止后，按一个键回到 Shell 屏幕。

12.2.4 主要命令提示符和辅助命令提示符的优点和不足

访问一个命令提示符的方法总有它的优点与不足。当加载一个 TSR 如 PRINT，或运行一个操作磁盘结构的程序如 SCANDISK 时使用主要命令提示符。不足之处是不得不完全退出 Shell，待做完后再启动。

当只想运行一个简单的命令时从 DOS Shell 中使用 File，Run 命令即可。优点是不必键

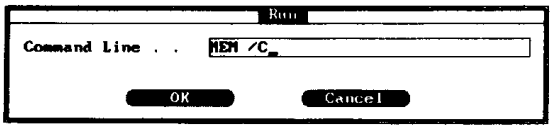

图 12.1　Run 对话框

入 EXIT 就返回 Shell，不足之处是只能运行一个命令。

想运行多个命令时，退到辅助命令提示符。优点是在返回 Shell 之前可键入想运行的许多命令。不足之处在于必须键入 EXIT 才能回到 Shell，而且不能运行特定类型的程序。

作者忠告： 只能从主要命令提示符重新启动(或关闭计算机)。

当正在程序如 Word，Windows 或 DOS Shell 下工作时，它可能有文件被打开(即使你没有处理任何数据文件)。在重新启动或关机之前，程序需要正常关闭所有的文件。若正常退出程序，则系统可以有机会关闭它的文件。你应该一直等到磁盘驱动器的灯停止闪烁，然后才能安全地重新启动或关机。

当在辅助命令提示符下工作时，如果有一个暂停程序等着恢复，就可能会有需要关闭的打开文件。因此，请确保在重新启动或关机之前从辅助命令提示符退到暂停程序并正常结束它。通用的规则是只从主要命令提示符重新启动或关闭。

这个规则有一个重要的例外。如果一个程序已经挂起——也就是说，它已经停止运行，已完全束缚住了计算机，此时不能再做其他任何事——只有重新启动。在有些情况下，甚至重新启动也不能工作，你将不得不按重设按钮或关闭电源再打开。当不得不重新启动或重新开始时，应该有心理准备，即它将丢失一些程序正在处理的一些数据。

12.3　理解命令格式

一个 DOS 命令有以下常用格式：

命令名 [参数，参数…] [选项]

注释： 在一个格式描述中，小写表示当你实际键入命令时要在这儿用一个词或其他数据代替；方括号包含着可选项；在方括号以外的任何东西都是必选项。

DOS 在命令中忽略大小写。本书只以大写字母显示命令，这是为了使它与文本区别。你可能会发现用小写键入更方便。因此当一些指导步骤说，“键入 VER，按 Enter”，你可以键入 VER，Ver，ver 或其他任意的大小写组合。对 DOS 来说它们都是相同的。

一个命令限制在 127 个字符以内(在典型的 80 列显示器上，大约占一行半)。大部分命令从不会接近这个限制，尽管在以后章节可能会看到有一些接近 127 个字符的命令。

在一个命令中用空格来分隔不同的词和表达式。例如，在命令 CD DOS (用于切换到 DOS 目录)中，在 CD 后有至少一个空格是很重要的。否则，DOS 不能解释 CDDOS(没有一个空格)。

注释： 在一些系统中，如果命令名后的下一个表达式是一个以\开始的路径名，则

不需要空格。例如,DOS 可在 PC 上将 CD\DOS 正确地作为 CD..\DOS 解释。

12.3.1 程序的名字

程序名是你想运行的程序的名字。例如,如果程序文件命名为 FORMAT.COM,命令名将是 FORMAT。如果程序文件是 RUNOUT.BAT,则命令名为 RUNOUT。

试着键入 VER 命令以找出正在使用的 DOS 的版本:

1. 在命令提示符下键入 VER。
2. 按 Enter。

   VER 显示 DOS 的版本,并终止自身运行,再显示一个命令提示符,准备接收下一个命令。

有它们自己的程序文件的程序,如 DOSSHELL.EXE 和 EDIT.COM,称为外部程序。DOS 软件包包含一些外部程序。另外,用户应用程序软件也提供外部程序。

DOS 也在它的 COMMAND.COM 程序中包含一些功能,如刚才运行的 VER 程序。因为这些功能在 COMMAND.COM 的内部,它们没有自己的程序文件,因此被称为内部命令或内部程序。在命令提示符下通过键入命令既可运行外部程序也可运行内部程序。

12.3.2 参数和选项

许多程序在它们的命令中包含参数和选项。一个参数是一种可变信息,它告诉程序你想做什么。例如当启动 DOS 的文本编辑器时,可包括一个文件名以打开编辑一个资料文件。

试着启动 DOS 的文本编辑器,用 README.TXT 文件作为初始化文件:

1. 在命令提示符下键入下列命令,按 Enter:

   EDIT README.TXT

   Editor 屏幕出现在显示器上,README.TXT 被打开。(如果当前目录不包含一个 README.TXT 文件,EDIT 将创建一个新的空文件)。如果得到一个消息 Bad command or file name,那么你也不要担心,这个消息将在下一节解释。你只需跳过下面的练习并继续。
2. 按 Alt+F,再按 X,退出 EDIT。

   EDIT 终止运行,DOS 显示另一个命令提示符。

选项是一个特殊类型的参数,常常用一个斜杠作为前缀,后跟一个字母或单词,它为程序选择一个特殊选项。例如 DIR 程序一行一个地列出一个目录中的文件。如果增加选项/W,DIR 按多列显示文件名在屏幕上。

试着在当前目录下使用/W 选项和不使用它显示文件:

1. 键入 DIR,按 Enter。

   DIR 显示文件清单后终止。(如果是一个长的清单,屏幕翻卷,你将不能看到上面的文件。当在第 14 章"目录管理"中学习更多关于 DIR 的信息时将学会如何处理这个问题。)然后 DOS 显示一个新的命令提示符。
2. 现在键入 DIR /W,按 Enter。

你将很快看到/W 选项的效果。文件名将横向列出。信息如文件尺寸和日期将不再显示。

12.3.3　在命令中使用路径

当输入一个命令时，DOS 将努力找到命令需要的程序，它首先检查自己的 COMMAND.COM 文件，看这个程序是否是内部的，然后在当前目录下查找正常文件名的可运行文件(带扩展名.EXE,.COM 或.BAT)

如果 DOS 在当前目录中没找到这个可运行文件，将发生什么事呢？DOS 保存一系列目录，称为程序搜寻路径，用它来搜寻程序。DOS 在每个搜寻路径目录中查找，直到找到需要的运行文件。在本章后面将介绍如何创建搜寻路径。

答疑： 输入一个命令时，得到消息：Bad command or file name。这出了什么错？如何才能运行程序呢？

DOS 不能找到作为输入命令的程序文件。首先要保证正确拼写命令的名字。如果不对，试着重新拼写。

如果确信命令名字是正确的，看看你自己是否能找到程序文件。输入下面的命令看一下驱动器 C 上所有带有这个命令名的文件。

```
DIR \command-name /S /B
```

例如，如果命令名是 WORD，你可输入 DIR \WORD /S /B(看 DIR 关于这个命令的解释)。

在结果清单中，寻找带扩展名.COM,.EXE 或.BAT 的文件。如果找不到，说明程序没有安装在硬盘上。安装这个程序，再试。

如果你能找到一个.COM,.EXE 或.BAT 文件，注意它在目录的路径名。使用 CD 命令到达这个目录。然后再试这个命令，它将成功运行。例如假设目录清单是：

```
C:\MS-WORD\WORD.EXE
```

则应键入 CD C:\MS-WORD 转到正确的目录，随后 WORD 将正常运行。(在第 14 章“CD”一节有关于这个命令的解释)。

如果经常使用这个命令，应该将它加入到程序搜寻路径中。这样，每次当你启动程序时无须转到这个目录。看本章的“PATH 程序”一节，它将解释如何增加一个程序到搜寻路径。

你可以不管 DOS 的常用规程，并且告诉它程序的确切位置，这时只须把路径连到命令名上。用这种方法可到达任意目录的文件。即使一个目录在程序搜寻路径中，包含路径名也可使 DOS 不必先搜寻其他目录。

试着加 C:\DOS 路径名到 EDIT 命令：

1. 输入下列命令：

```
C:\DOS\EDIT README.TXT
```

DOS 这时可以更快地找到 EDIT，虽然你可能不会注意到有什么不同。

2. 按 Alt+F，然后按 X 键退出编辑器。

把路径附加到一个命令名之前仅仅影响 DOS 在什么地方搜寻程序文件，而不会影响包含在命令中作为参数的文件描述。如果想打开一个不在当前目录的文件，也可在文件描述中加入路径名。

试着在 DOS 目录中打开 README. TXE 文件：

1. 输入下列命令：

```
C:\DOS\EDIT C:\DOS\README. TXT
```

这次，EDIT 打开随 DOS 一起提供的 README. TXT 文件。

2. 按 Alt+F，按 X 退出。

你可以看到，用这种方法寻找文件并不很有趣，而且常常会犯错误。因此，还有其他更好的办法帮助 DOS 找到一个程序文件：

- 增加想要的程序的目录到程序搜寻路径中。这使寻找程序更容易，但它不影响数据文件的寻找。
- 转到想要的目录，这样在输入命令前它是当前目录。(将在第 14 章学习用 CD 命令改变目录)。这会影响程序文件和数据文件的寻找。

试着转到 DOS 目录，打开 Editor：

1. 键入下面的命令转到 DOS 目录：

```
CD DOS
```

你将看到新的当前目录名出现在命令提示符中。

2. 输入下列命令启动 DOS Editor，并打开 README. TXT 文件：

```
EDIT README. TXT
```

编辑器从 DOS 目录启动并编辑 README. TXT 文件。

3. 按 Alt+F，按 X 退出 Editor。

12.4 重定向命令输出

许多 DOS 命令都会显示消息。例如 DIR 命令将列出一个目录的内容；PATH 命令在不带参数时显示当前搜寻路径。在常规环境下，这样的消息显示在显示器上，但你可重定向它们到一个文件或一个外部设备如打印机。大于号(>)用来重定向命令消息。下面的命令将保存当前目录清单到一个名叫 DIRC. TXT 的文件：

```
DIR > DIRC. TXT
```

警告： *这个命令将覆盖已存在的 DIRC. TXT 文件。*

如果有一台并行打印机连在第一个(或唯一)并行口上，试着打印一个目录清单：

1. 确保打印机已准备好(参看第 9 章“通过 Shell 打印”中的关于细节清单的提示)。

2. 输入下列命令：

```
DIR >LPT1
```

这时 DOS 目录将在打印机上打印，从而用户将得到至少两页数据。如果没有，看下面对问题的解决。

答疑： 重定向一个命令到打印机，但它不打印，或打印除最后一页外的内容。

如果没打印什么，说明计算机现在挂起。DOS 正花所有的时间在寻找打印机。可能打印机没有打开，没有正确连到计算机或没有联机。如果能发现并解决这些问题，计算机可接着工作。否则，不得不重新启动计算机。

如果计算机没有挂起，DOS 已传输入了数据，但打印机不打印。检查打印机的控制板，保证它没有缺墨或缺纸。

如果打印机已准备好，还是不能打印，那么它可能正在等待一个页排斥信号。大部分激光和喷墨打印机在打印前存储一整页数据，除非它接收到一个页排斥信号。在重定向数据时 DOS 不会发出一个页排斥信号，这样如果最后一页不满页，打印机只会保存它等待更多的数据。如果你的打印机有一个页排斥按钮，试着按个按钮。如果没有，在 DOS 提示符下键入下列命令。其中 ^ L 表示按 Ctrl+L(而不是键入 ^ L)。

```
ECHO ^ L>PRN
```

表 12.1 显示了能重定向输出的设备名。

表 12.1　输出设备名字

名字	设备
LPTn	并行打印机端口 n；LPT1 是第一个并行打印机端口
PRN	与 LPT1 相同
COMn	串行通信端口 n；COM1 是第一个串行端口
AUX	辅组端口
NUL	没有；输出被遏止
CON	显示器屏幕

DOS 命令产生 ASCII 文本消息。当重定向消息到一个文件时，将创建一个标准的文本文件。当重定向它们到一个输出设备时，这个设备必须设置成处理 ASCII 文本。大部分设备都可处理 ASCII 文本。但不能重定向命令消息到一个 PostScript 打印机，除非你能设置这个打印机使之能处理 ASCII 文本。

不是所有的命令消息都能通过>重定向，只有标准消息才可这样，而错误消息仍旧送到显示器，这在如 DIR 之类的程序中会产生一个有趣的情形。当 DIR 不能发现一个文件时，它正常显示一组消息如：

```
Volume in drive C is MICROMATE
Volume Serial Number is 199A-59E3
Directory of C:\NEWBOOK
File not found
```

前三行是标准消息;最后一行是一个错误消息。如果重定向输出到一个文件,前三行将保存在这个文件中,而最后一行显示在屏幕上。

许多程序产生非标准消息输出,它不用>重定向。例如,COPY 程序在目标目录或设备中创建一个文件,如果重定向 COPY 的输出,则只有标准消息(如 1 Files Copied)重定向,而新文件不被重定向。

注释: 可重定向任何在显示器上产生标准消息的程序的输出。这不仅仅局限于 DOS 应用程序。

预览: >符号只是 DOS 命令重定向符号中的一个。第 36 章"命令状态下的 DOS 应用"将介绍其余三个,但它们用得很少。

12.5 命令参考介绍

本书后面的部分是一个完整的 DOS 命令参考,其中命令名按字母顺序排列。对每个命令,命令参考提供了命令格式、参数和选项,以及关于命令的一些提示等资料。此外还提供若干完整的例子。

命令参考包括所有 DOS 命令和选项,甚至还有那些在本书教程部分没有包括的中高级命令,命令参考不是用教程的方式解释命令,而只是把它们列出来。

命令参考使用了几种规则来描绘命令格式,其中以大写形式表示的词和词组在录入时须原样录入。例如,VER 命令格式如下:

```
VER
```

这个命令显示当前的 DOS 版本号。要使用它,键入 VER 后按 Enter 即可(它不必全用大写。记住 DOS 大部分时间忽略大小写)。

当你输入命令时,小写的词和词组必须用特定信息代替。如 DEL 命令用来删除文件,它的格式如下:

```
DEL filespec
```

你必须在 DEL 后跟一个文件名,如下所示:

```
DEL C:\CONFIG.OLD
```

包含在方括号中的词和词组是可选的,它们与命令的作用不同。例如,DATE 和 TIME 命令用以检查和改变系统日期和时间,它们的格式如下:

```
DATE [date]
TIME [time]
```

若键入 DATE 时未带日期,DOS 就显示当前系统日期并访问用户是否要改变它。若在此命令后加上了日期,DOS 不加询问就改变系统日期。与此类似,不带时间的 TIME 将显示当前时间并询问是否修改,但若指定 TIME 3:00,DOS 不加询问就将系统时间改为 3:00。

有时选项内部还有选项,它用嵌套的方括号括起来。如带/A 开关的 DIR 命令语法如下:

DIR [/A[attributes]]

当在 DIR 命令后带上/A 开关时，DIR 将列出具有某种属性的文件。在这里你既可以指定属性也可以只用/A。若只用/A，所有属性的文件都将列出。但若加上指定属性，则只有该属性的文件才被列出。例如，下述命令只列出可读属性的文件：

DIR /AR

有些情况，参数只能与另一个参数结合使用，这也引起方括号的嵌套。如 DEVICEHIGH 命令（将设备驱动程序装入内存高端）：

DEVICEHIGH=filespec[/L:region[,min][/s]]

其中的参数/L:region 是可选的。若使用它，还可加上一个逗号和 min 参数，并在此之后可加上/S 开关。注意，/S 放在包含/L:region 的方括号内，因此不带/L:region 就不能使用/S。

省略号(…)用来表明参数可重复。例如 PATH 命令如下：

PATH [pathname[;pathname]…]

若指定一个路径参数，就可加入第二个路径，中间用分号分开。上面的三点表明可以任意次重复；pathname 参数(DOS 命令有 127 字符长的限制)，从而向本命令加入更多的目录。

有时可以在两个或多个相互排斥的参数中选取，这些参数之间用垂直条(|)隔开。例如 VERIFY 命令：

VERIFY [ON|OFF]

该命令控制 DOS 是否检验写入磁盘的所有内容。VERIFY ON 打开检验，VERIFY OFF 则关闭，而不带参数的 VERIFY 则用于显示当前检验状态。

当组合这些要素后，命令语法变得相当复杂。在命令参考部分，注释和例子将帮助你如何在实际中使用该命令。

12.6　获取命令的屏幕帮助

假设想删除当前搜索路径，但记不清 PATH 命令的有关语法，则在这时，如果不想翻阅本书后的命令参考，那么可以不用离开键盘就查找到该命令。DOS 包括两种屏幕帮助：FASTHELP 和 HELP。其中 FASTHELP 显示命令的简要命令语法，而 HELP 显示有更完整的命令文档。

12.6.1　获取命令语法的屏幕帮助

若在 DOS 命令后加/?，DOS 就显示该命令的简要语法描述。假设想知道如何缩小程序搜索路径，可显示 PATH 命令的简要语法，这时应键入 PATH /? 然后按 Enter 键。DOS 将显示 PATH 命令的语法(见图 12.2 所示)。

用此方法并不能弄清该命令，但它能很好地提示你是否忘了某个开关名或某些细节。这里，它告诉你在 PATH 命令后加上分号就可删除搜索路径。对于刚才的问题，此方法足够

了。

DOS 的语法描述使用了与本书中类似的格式。所有大写的词(如 PATH)都必须原样键入,不过键入时既可以大写也可以小写。在本书中用小写形式表示可替换项,DOS 中只使用小写(如 drive:)。在方括号中的项为可选项,省略号表示可重复项。

```
c:\path /?
Displays or sets a search path for executable files.

PATH [[drive:] path[;...]]
PATH ;

Type PATH ; to clear all search-path settings and direct MS-DOS to search
only in the current directory.
Type PATH without parameters to display the current path.

c:\>
```

图 12.2 PATH 命令的屏幕帮助信息

本书的格式描述与在命令提示符下的稍有不同。在许多情况下,两种版本表示相同的事情,但本书中试图以一种更简单更清楚的方式表达。在某些方面,对于 DOS 版本有点不准确的地方,本书将提供一个正确的版本。大部分情况下,DOS 版本就足以告诉你命令的选项或参数的名字。

注释: 如果系统是从 DOS 5 升级,那么你可能已习惯于在 HELP 命令后跟一个命令名来显示简短的格式消息。HELP 在 DOS 6 中已经改进,这你会在下节中看到。这个更简捷的实用程序被更名为 FASTHELP。如果不想使用/? 选项,则可输入 FASTHELP command 看简短的格式。

FASTHELP 不带参数时显示 DOS 命令的完整清单,每个命令带有一个简短的解释。这个清单在屏幕上自动分页,但重定向到打印机上时不分页。

提示: 如果你负责帮助一组没经验的人使用 DOS,你可能会希望可以调整 Help 清单,以使之包含你自己的提示、注释和警告,或者为了将你创建的命令和宏归入资料,抑或(最为特殊地)为了修改用 DOSKEY 宏代替的命令解释。在这时你可以通过编辑文件 DOSHELP.HLP 来修改命令清单。该文件的开始解释了编辑清单的正确方法。

12.6.2 得到带有完整命令信息的屏幕帮助

为了获得更完整的屏幕资料,键入带或不带一个命令名的 HELP 命令。HELP 程序启动和显示一个窗口,从中你能访问关于内部命令、外部命令、批命令或 DOS 提供的设备驱动程序的信息。(FASTHELP 不包含设备驱动程序和批命令,但 HELP 有)。

注释： 获得如何使用关于 Help 系统本身的帮助，在 Help 窗口打开后按 F1。

12.6.2.1 内容清单

当你输入 HELP 并且不带命令名时，Help 窗口将打开并显示内容清单（如图 12.3 所示）。每对尖括号间的内容如＜What'S New in MS-DOS 6.22＞叫做一个 jump（跳越）。通过单击尖括号内的任何部位可以跳到一个新的题目，你也可先将光标移到一个 jump，然后按 Enter 以显示这个题目。

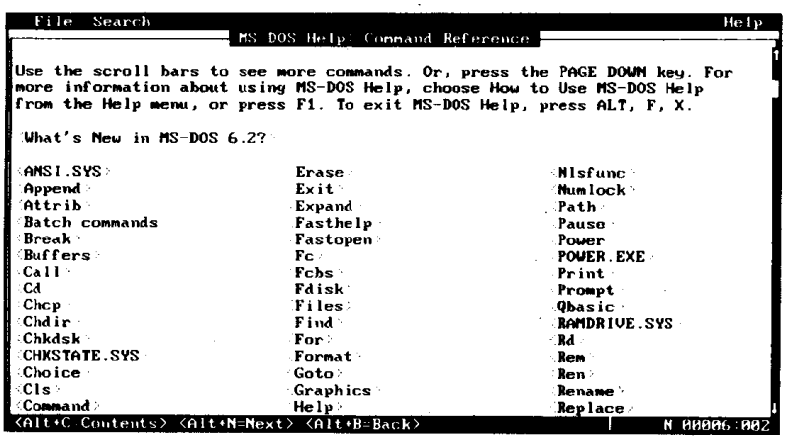

图 12.3　Help 内容

试着在屏幕上打开内容清单：

1. 键入 HELP，不带参数。

   Help Contents 清单在屏幕上打开。

2. 读下一节时保留屏幕上的清单。

12.6.2.2 在 Help 窗口移动

表 12.2 显示了用来在 jump 间移动光标的键。不幸的是，题目是按字母顺序向下列出的，但 Tab 键却只能水平移动光标。因此，如果想从＜Call＞到＜Chdir＞，则不得不按 Tab 九次。按字母 C 或按下箭头键三次可以加快速度。当然用鼠标最快，只须单击项目即可。

表 12.2　在 Help 窗口中移动光标

键	动作
Tab	横向移到下一个 jump
Shift＋Tab	横向移到前一个 jump
↓	向下移动一行
↑	向上移动一行
Letter	移到以 Letter 开头的下一个 jump
Shift＋letter	移到以 Letter 开头的上一个 jump
Page Down	移到下一个窗口
Page Up	移到上一个窗口

光标移动键如下箭头键和 PgDn 不会从一个 jump 移到另一个 jump；它们只是在窗口

中移动光标。如果光标碰巧停在一个 jump 上更好。否则，将不得不手工将光标移到 jump 上。(其他光标移到键如左箭头键和 Home 键也在屏幕上工作，但它们对从 jump 移到 jump 没有用)。

1. 使用光标键移到<Exit>。
2. 现在移到<Choice>。

现在保留屏幕上的内容清单，继续阅读下一节。

12.6.2.3 翻卷 Help 窗口

如在图 12.3 中看到的，整个内容清单不能全部放在窗口中，因而必须翻卷才可看到全部内容。当使用表 12.2 的任何技术移动光标时，清单在必要时将自动翻卷，以显示光标的新位置。

试着用键盘翻卷清单：

1. 按 PgDn，直到到达了清单的底端。

   一个小方块表示到了清单的底端。
2. 使用 Shift+Tab 移到前一个 jump。
3. 按 C 移到以 C 开头的第一个命令。
4. 按 Shift+C 移到以 C 开头的最后命令。
5. 多按 Shift+C 几次，在以 C 开头的命令中反向从一个 jump 转到另一个 jump。

也可使用滚动条和鼠标在不移动光标的情况下翻卷。参考第 5 章“DOS Shell 概述”中的“如何用鼠标滚动”部分可得到关于使用滚动条的方法。

如果有一个鼠标，用它来试一下翻卷内容清单：

1. 单击滚动条上端的向上箭头几次，观察清单向上一行一行翻卷。
2. 将鼠标光标移到向上箭头上，按住鼠标按钮并观察清单快速向上翻卷。
3. 单击滚动盒下面的部位几次，观察清单一次向下翻卷一个窗口。
4. 拖住滚动盒到滚动条的底部，以观察清单的结尾。

12.6.2.4 题目屏幕

当从 HELP 窗口选择一个命令如<PATH>时，将看到它有三个题目：Syntax，Notes 和 Examples。当你首次选择一个命令时，Syntax 题目出现(如图 12.4 所示)，而另外两个题目作为这个题目的开始的两个 jump，选择其中一个就和在内容清单中选择一个 jump 一样。如果在这个题目中向下翻卷，jump 将离开屏幕的顶部。当然你可向上滚动回到 jump 所在位置，或者按 Tab 或 Shift+Tab 直到光标到达想要的 jump，或者按 N(Notes)，E(Example)或 S(Syntax)移动光标到想要的 jump。

下面我们对 PATH 命令进行一番考察：

1. 单击<path>，或按 P 把光标移到其上后按 Enter。

   Path-Syntax 题目打开。
2. 翻卷到题目的底端。

   不用担心没机会阅读这个题目；将在本章的后面学习 PATH。

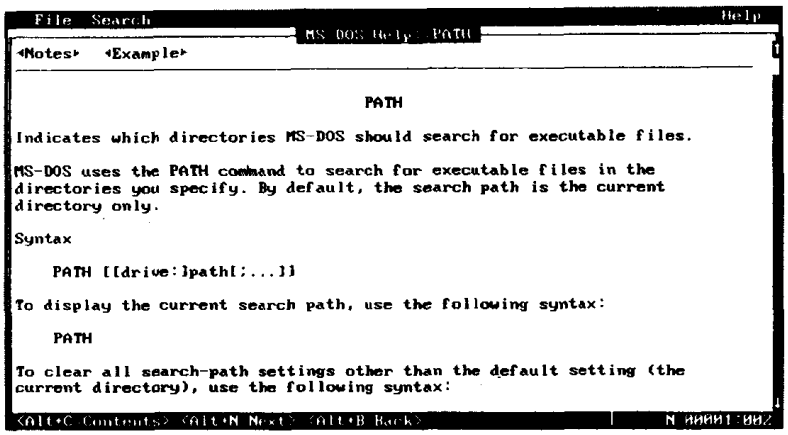

图 12.4　PATH—Syntax 题目

现在试着看一个 Path-Notes 题目：

1. 按 N 移到<Notes> jump。
   清单自动卷回顶端。
2. 单击<Notes>或按 Enter 打开 Path-Notes 题目。
3. 向下翻卷到题目内容的底端。
   注意这是一个有关 PATH 命令的 Notes 集。

现在试一下 Path-Example 题目：

1. 单击<Example>或输入 E 后按 Enter。
   Path-Example 题目出现。注意这是一个 PATH 命令的例子。

12.6.2.5　其他求助操作

为回到 Contents 清单，可在底端状态栏单击<Alt+C=Contents>或按 Alt+C；按 Alt+N 或单击<Alt+N=Next>可以浏览库中的下一个题目；而按 Alt+B 或单击<Alt+B=Back>则可回到你已看过的题目。

试一下这些操作：

1. 单击<Alt+N=Next>或按 Alt+N。
   Pause-Syntax 题目出现，这是在帮助库中 Path-Example 后面的下一个题目。
2. 单击<Alt+B=Back>或按 Alt+B。
   Path-Example 题目重新出现，这是已浏览过的前一个题目。你可继续选择 Back 回溯整个帮助部分。
3. 单击<Alt+C=Contents>或按 Alt+C。
   Contents 清单出现。

提示： 不从 Contents 清单开始也可看一个具体命令。在命令提示符下键入 HELP command 就可打开 Help 窗口，上面是需要的命令的 Syntax 题目。如为了观察 PATH 命令格式，你可输入 HELP PATH。

如果想直接到达一个命令的 Notes 或 Examples 题目，只需增加两个短横，后面接着 NOTES 或 EXAMPLES 即可。如 HELP PATH-NOTES 或 HELP PATH-EXAMPLE。

12.6.2.6 帮助菜单

Help 的三个菜单提供下列机制：

Print(在 File 菜单中)	打印当前帮助题目。可选择将其付诸打印机打印或保存在一个文件中。
Exit(在 File 菜单中)	退出 Help 程序
Find(在 Search 菜单中)	在所有的 Help 题目中搜寻你指定的任何词组。 例如，如果你不能确信哪个命令删除文件，可以搜寻词 delete。
Repeat Last Find (在 Search 菜单中)	重复最近的搜寻命令。例如，你可以重复搜寻 delete 几次，直到找到想要的命令。
How to Use MS-DOS Help(在 Help 菜单中)	显示关于如何使用 Help 程序的一系列话题。
About(在 Help 菜单中)	显示 Help 程序的拷贝权和版本信息。

提示： 不回到 Contents 清单从一个命令转移到另一个命令的快速方法是搜寻命令名。例如，如果搜寻 PATH，则 HELP 不会向你显示 Help 库中词 PATH 的下一个实例，而是跳到 Path-Syntax 题目。Repeat Last Find 在你这样做后不会工作，因为仅有一个 Path-Syntax 题目。

12.6.2.7 搜寻机制

如果不能确认想查询哪个命令，可以在整个帮助库中搜寻一个词或词组。例如，假设你想发现哪个命令是对 CD-ROM 驱动器的，试着在帮助库中找到 CD-ROM 的参考信息。

1. 在菜单栏单击词 Search 或按 Alt+S。
   这时将出现 Search 下拉式菜单。
2. 在 Search 菜单上单击 Find 命令或按 F。
   Find 对话框打开(如图 12.5)所示。

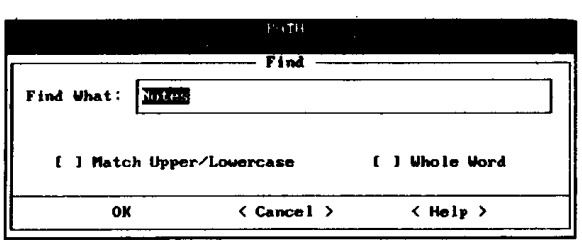

图 12.5 Help 的 Find 对话框

3. 键入 CD-ROM，然后按 Enter。

DOS 搜寻 Help 库，看有没有词 CD-ROM 的下个实例。它可能会花一点时间，因此需耐心等待。消息 Searching－Press Esc to Cancel 出现在屏幕底端的条上，它告诉你 DOS 正在搜寻 CD-ROM。

最后，一个新题目出现在屏幕上。光标在 CD-ROM 的第一个字母 C 下面。

注释： 在搜寻对话框中键入的文本可以是大小写的任意组合。我们在本书中为了清楚起见，全用大写。DOS 忽略大小写，除非 Find 对话框中 Match Upper/Lower-case 被选择。

当文本被找到时，题目名字出现在屏幕的顶部，它告诉你什么命令处理这个题目。你可翻卷阅读题目的其余内容，或跳到相关题目去寻找需要的信息。

如果选择的题目不是想要的那个，则可在库中继续搜寻。这时你只需在 Search 菜单中选择 Repeat Last Find 或按 F3。

试着寻找 CD-ROM 的下一个实例：

1. 按 F3。

   DOS 发现下个 CD-ROM 实例，光标处于字符 C 下。(如果题目没有改变，则在相同页上寻找光标的新位置)。

2. 继续按 F3 看更多的 CD-ROM 实例。

   有时题目会翻卷，有时出现一个新的题目，有时光标只是在相同的屏幕上移到下一个 CD-ROM 实例 。

现在，试着打印一个题目：

1. 单击 File 或按 Alt＋F，拉下 File 菜单。
2. 单击 Print 或按 P 选择 Print 命令。

   Print 对话框打开(如图 12.6 所示)。缺省选项是连接到 LPT1 上的打印机。

3. 单击＜OK＞或按 Enter。

   Help 题目开始打印。如果没有，参考本章“重定向命令输出”的“答疑”。

既然已经对 Help 系统作了一番探索，现在可以关闭 Help 窗口并返回命令提示符，以便学习一些 DOS 命令：

1. 单击 File 或按 Alt＋F 拉下 File 菜单。
2. 单击 Exit 或按 X 选择 Exit 命令。

   Help 窗口关闭，返回命令提示符。

12.7　PATH 程序

PATH 程序用于设置程序搜寻路径。图 12.7 显示了这个命令的格式。如果只键入 PATH，而不带参数，那么 PATH 将显示当前搜寻路径。PATH 命令的可能形式如下：

```
PATH C:\DOS;C:\;C:\WP;C:\MAILING
```

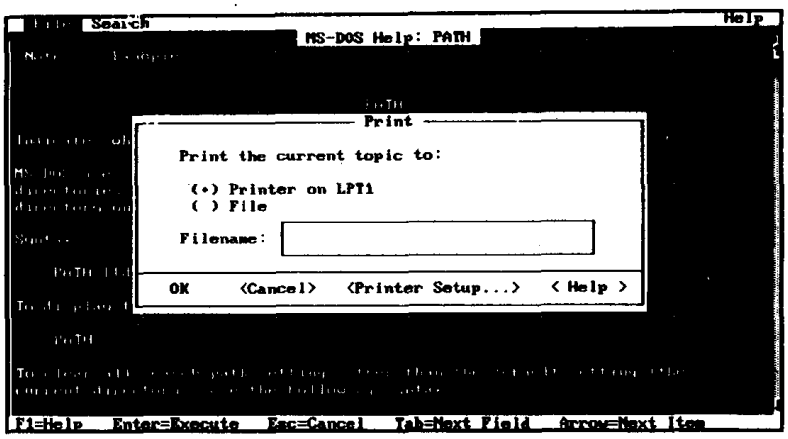

图 12.6 Help 的 Print 对话框

PATH

为外部可执行文件(包括批文件)定义一个搜寻路径。

句法:

PATH[pathname[;pathname...]]

PATH;

参数:

none	显示当前搜寻路径。
pathname	指定搜寻程序文件的一个目录。
;	只有这个参数时清除搜寻路径。

图 12.7 PATH 命令总结

注释: 在这些章节显示的命令格式常常比较简洁。从命令参考部分可找到完整的命令信息,其中包括增强参数、实例和考虑。

为了要用一个新路径代替当前路径,可在 PATH 后加一串由分号隔开的一系列路径名。不要在参数中加入空格,尽管这样很难看,可是整个路径名串必须作为同一个参数。列出路径名的顺序决定了 DOS 在寻找一个程序时搜寻它们的顺序。为了依次搜寻 DOS 目录、WP 目录和根目录(按这个顺序),你可以输入如下命令:

```
PATH C:\DOS;C:\WP;C:\
```

提示: 在 PATH 命令中安排路径名时,用得最多的排在前面,以使 DOS 可以尽可能快地找到程序。

当启动系统时,不存在搜寻路径。大部分用户在 AUTOEXEC.BAT 中建立它们自始至终使用的路径。AUTOEXEC.BAT 文件中可能已经包含由 DOS 的 SETUP 程序插入的一个 PATH 命令。此外,当安装一个新程序时,它的 INSTALL 或 SETUP 程序可以将它的目录加到 AUTOEXEC.BAT 的 PATH 命令中。

每隔一段时间要看一下 AUTOEXEC.BAT 文件，并且编辑 PATH 命令，以删去不再使用的程序的目录并改变目录的顺序以适应你使用系统的方式。

DOS 命令限制在 127 个字符，因而不能让 PATH 命令过长，否则命令结尾的目录将被忽略。

当键入 PATH 命令时，DOS 不会检查目录是否合法。在搜寻一个程序文件时，它只是简单地忽略不合法的目录。

注释： 记住可以在路径名中加入程序名来忽略搜寻路径，如在 D:\DAS\FLICK 中。

试着浏览当前搜寻路径：

1. 键入 PATH，不带参数。
   DOS 显示搜寻路径。
2. 检查搜寻路径以决定是否想改变它。有更好的路径名顺序吗？它包括不再使用的目录吗？它省略了常用使用的目录——特别是 DOS 目录吗？
3. 如果决定改变它，则键入 PATH 命令，并定义想使用的搜寻路径。保证完整准确地拼出每个路径名，包括驱动器名在内，同时保证用分号隔开路径。此外还要保证在参数中没有空格。
   如果键入一个新的 PATH 命令，则旧的搜寻路径完全被新的一个取代。

注释： 如果想长久使用一个不同的程序搜寻路径，应该用它取代 AUTOEXEC.BAT 中的 PATH 命令。第 20 章“基本批处理程序”解释了如何编辑 AUTOEXEC.BAT 文件。

12.8　PROMPT 命令

命令提示符通常采用当前驱动器和目录后跟一个大于号(>)的形式表示。但也可以设置成想要的任意提示符。只要以以下格式输入一个命令：

```
PROMPT [text]
```

如果不指定任何文本，那么当前提示符被删除，同时一个简易提示符被建立。你可能从未看见过简易提示符，它只是驱动器名后跟一个大于号。你自己试一试：

1. 键入不带参数的命令 PROMPT。
   DOS 显示的下一个命令提示符只是驱动器名后跟一个大于号。

因为大部分人不仅想知道当前驱动器，而且想知道当前目录，因而缺省时 DOS 显示的是一个大家都习惯的命令提示符。

可以将提示符改成想用的任何文本。试着创建一个提示符是 What's next?：

1. 在命令提示符下键入下面的命令：

```
PROMPT What's next?
```

DOS 显示新的命令提示符。

注释： 这是 DOS 注意大小写的另一个方面。PROMPT 使用你在 PROMPT 文本中设置的任何字体。

表 12.3 显示了可用在 PROMPT 命令中的一些特殊代码。其中的一些代码可将 DOS 信息(例如日期或当前目录)插入提示符中。其他代码将在提示文本中产生特定字符，这些字符可被解释成 PROMPT 命令的一部分。你已经知道了>号是重定向符号，所以不能在提示文本中使用它，但可使用 $G(G 代表大于)要求一个>号。缺省提示符表达为 PG，这里 $P 要求当前驱动器和当前目录，$G 显示大于号。

表 12.3 特殊 Prompt 命令代码

代码	意义
$G	大于号(>)
$$	美元标志($)
$P	当前驱动器和目录
$D	当前日期
$T	当前时间

现在，恢复缺省提示符：

1. 键入下列命令：

```
PROMPT $P$G
```

返回至标准提示符。

12.9 自 测 练 习

本章告诉你如何对命令提示符进行操作。如果愿意针对在本章学习的这些技术多作些练习，那么可试着做下面的练习：

1. 保证已理解主要和辅助命令提示符的区别，保证知道如何到达主要命令提示符，同时保证知道如何从 DOS Shell 或 Windows 到达辅助命令提示符。
2. 通过重定位 VER 命令输出到打印机来打印 DOS 版本号。
3. 在 FASTHELP 和 HELP 中查找 VER。
4. 显示当前程序搜寻路径。
5. 改变命令提示符，使之显示当前日期和时间以及当前目录。
6. 恢复正常的命令提示符。

第 13 章 DOSKEY

简介： 在 DOS 推出 DOSKEY 之前在命令提示符下工作令人可畏而且易混淆。现在简单了。

13.1 理解 DOSKEY 做什么

通常，调回并编辑一个曾使用过的命令比重新键入一个新命令要方便得多。早期的 DOS 版本对此有一定的限制，你不得不盲目工作(因为看不见前面的命令)。现在可加载一个 TSR 程序 DOSKEY，它使命令编辑容易多了。注意当从 Windows 或 OS/2 启动一个新的 DOS 命令会话时，DOSKEY 清单将从头开始。

DOSKEY 不仅可以进行命令编辑，而且能做下列事情：

- 调回前一个命令，并按原样运行
- 组合两个或多个命令成为一个命令
- 创建命令宏。宏是一个复杂的命令，它可以通过按一个简单的名字来运行

如果打算经常在命令提示符下工作，那么应该使用 DOSKEY。它将节约许多时间，并能减轻负担。

本章主要讲述：

- 加载 DOSKEY TSR
- 调回并编辑命令
- 组合两个或多个命令
- 定义和使用 DOSKEY 宏
- 管理宏清单
- 管理命令清单
- 用宏超越 DOS 命令

13.2 加载 DOSKEY TSR

图 13.1 显示了启动 DOSKEY 的命令格式。如果你的系统设置合适，那么你可将 DOSKEY 加载在高端内存。这将在第 30 章“为 386、486 和奔腾机增加的内存工具”中解释。

DOSKEY

保存并提供在 DOS 命令提示符下键入的命令和宏的访问：

句法：

DOSKEY [/BUFSIZE=size][/INSERT|/OVERSTRIKE][/REINSTALL]

参数和选项：

none	加载 DOSKEY TSR。BUFSIZE 和 INSERT|OVERSTRIKE 使用缺省值，同时 DOSKEY 缓冲区马上开始记录命令。
/BUFSIZE=size	指定 DOSKEY 缓冲区的大小。此参数只有当加载或重装 DOSKEY TSR 时才有效。缺省值为 512 个字节。
/INSERT	设置缺省的插入和键入模式。
/OVERSTRIKE	设置缺省的覆盖键入模式，通常用缺省值。
/REINSTALL	安装 DOSKEY 的一个新拷贝。

图 13.1 DOSKEY 命令总结(加载 DOSKEY)

试着使用所有缺省选项加载 DOSKEY：

1. 不带参数键入 DOSKEY。
   不幸的是，当 DOSKEY 顺利加载后你不会得到任何消息。所看到的只是一个新的命令提示符。(这就是所谓没有新闻才是最好的新闻)。
2. 这样就可以用一些命令在 DOSKEY 上工作，输入命令 VER(不带参数)。
   DOS 显示当前 DOS 版本号。
3. 键入 PATH 命令(不带参数)。
   DOS 显示当前程序搜寻路径。
4. 键入 PROMPT 命令(不带参数)。
   DOS 变到基本提示符，驱动器名后跟大于号(>)。

是否注意到看不见 DOSKEY 在工作？但它确实是在工作，这将在下节看到。

注释： 如果使用缺省设置，DOSKEY 占用 4KB 内存。如果增减缓冲区的大小(这将在本章后面的“控制 DOSKEY 缓冲区大小”中解释)，内存也跟着改变。DOSKEY 可被加载到高端内存(这将在第 30 章中解释)。

13.3 使用已输入过的命令

DOSKEY 已经记录了三个刚输入的命令。表 13.1 显示了从已输入过的命令中调回命令所使用的键。最常用的是用向上箭头键得到想用的命令。每按一次向上箭头键，DOSKEY 在缓冲区中往回多移一个命令，并在命令提示符下显示。如果回得太远，可按向下箭头键再往前移。当看到想要的命令后，编辑后按 Enter。

表 13.1　从 DOSKEY 历史中调回命令

键	动　作
↑	调回缓冲区中前一个命令
↓	调回缓冲区中下一个命令
PgUp	调回缓冲区中最早的命令
PgDn	调回缓冲区中最新的命令
F8	调回缓冲区中前面的以命令行字母开头的命令
F7	显示所用存储的命令清单
F9	键入命令号码以调回那个命令
Alt+F7	删除缓冲区中所有的命令

试着调回从前输入的命令：

1. 按向上箭头键。

   DOSKEY 显示 PROMPT 命令，注意此时这个命令不运行。它只是在命令行上显示。你能编辑它，键入它或调回另一个命令。

2. 再按一次向上箭头键。

   DOSKEY 显示 PATH 命令。

3. 再按一次向上箭头键。

   DOSKEY 显示 VER 命令。

4. 按 Enter 运行 VER 命令。

   DOS 显示当前版本号。

如果想回到从前的命令，键入命令的开头几个字符再按 F8 将是很方便的。试着调回 PATH 命令：

1. 键入 PA，按 F8。

   DOSKEY 显示 PATH 命令。

2. 键入 V 后按 F8 可调回 VER 命令。

回到更早的命令的另一种方法是按 F7，它将显示缓冲区中所有的命令，每个命令分配一个数字号，当决定使用清单中的哪一个命令时，按 F9。DOSKEY 要求一个命令号，然后它显示与这个号码对应的命令。试着使用 F7 和 F9 调回 PROMPT 命令：

1. 按 F7。

   DOSKEY 显示缓冲区中所有命令的一个编号清单。

2. 找到 PROMPT 命令的号码。

3. 按 F9。

   DOSKEY 要求一个命令号。

4. 键入 PROMPT 的号码后按 Enter。

   DOSKEY 显示 PROMPT 命令。

明显地，它比只按几次向上箭头键更容易。但是，如果需要调回 25 或 30 个命令，F7—F9 组合就更快了。（如果已知道命令号码，只需按 F9，而 F7 就不需要了）。

13.4 编辑命令

既然命令行有一些文本，那末怎样编辑它呢？只需用光标键前后移动，插入、删除和覆盖键入字符即可。当命令准备好后，按 Enter 就可执行。DOSKEY 又把它作为最近的命令保存在历史中。

采用早期的 PROMPT 命令恢复常用的命令提示符：

1. 如果 PROMPT 命令不在命令行上，使用 DOSKEY 调回它。
2. 把光标放在命令的结尾(按 End)，编辑如下面的样子：

```
PROMPT $P$G
```

3. 按 Enter 执行这个命令。

最熟悉的命令符恢复了。

13.4.1 插入和覆盖编辑模式

如果你已在如 DOS 的 EDIT 这样的文本编辑器中做过字处理或文本编辑，那么你会知道插入和覆盖模式的区别。编辑命令时，DOSKEY 也提供了两种模式，而且你可通过编辑键在这两种模式间转换。

覆盖模式是缺省模式，因为加载 DOSKEY 时没有指定/INSERT 选项。试试覆盖模式：

1. 按向上箭头键调回最新 PROMPT 命令。
2. 按向左箭头键移动光标到命令的第一个 $ 上。
3. 用 $D 覆盖 $P。

因为正处于覆盖模式，而不是插入模式，故 $D 代替了 $P。(如果它插入，说明缺省模式为插入模式。如果有人在 AUTOEXEC.BAT 中放入了命令 DOSKEY/INSERT，则每次启动时按插入模式加载。现在删除这个命令，并且键入命令 DOSKEY/OVERTYPE，然后再试一试本练习)。

4. 按 Enter 以运行这个命令。

下一个命令提示符在大于号前显示日期。

现在试一下插入模式：

1. 按 Insert 键置插入模式。

注意光标由矩形代替了正常的下划线。这表明系统处于非缺省的编辑模式。

2. 按向上箭头键调回前一个 PROMPT 命令。
3. 按向左箭头键把光标移到第一个 $。
4. 键入 $P，后跟一个空格。

$P 插入 $D 的前面。

5. 按 Enter 运行这个命令。

新的提示符显示当前驱动器和目录，后面跟着今天的日期，再后面是大于号。光标又变成了下划线，表示恢复到缺省编辑模式。

如果喜欢用插入模式作为缺省编辑模式，键入带选项/INSERT 的 DOSKEY 命令。加载时可将/INSERT 包含在 DOSKEY 命令中，或者在任何时候想改变时键入另一个 DOSKEY 命令。

1. 键入下列命令可使插入模式成为缺省模式：

```
DOSKEY/INSERT
```

2. 按向上箭头键调回前面的 PROMPT 命令。
3. 使用向左箭头键把光标放在命令的第一个 $ 下。
4. 键入 $T，后面跟一个空格。
   注意 $T 插入命令中，因为现在插入模式是缺省模式。
5. 按 Insert 键。
   光标变成闪烁的矩形，表示现在是非缺省的键入模式起作用。
6. 键入 $ $，后面跟一个空格。
   注意这些字符覆盖了命令中的 $D。因为插入现在是缺省编辑模式，覆盖键入是非缺省的模式。
7. 按 Enter 执行这个命令。
   下一个提示符显示当前时间、美元符和当前日期，后面跟一个大于号。光标变成闪烁的下划线，现在是缺省的插入编辑模式。

现在整理命令提示符：

1. 按向上箭头键调回前一个命令。
2. 编辑命令成 PROMPT PG。
3. 按 Enter 恢复正常命令提示符。
4. 键入 DOSKEY/OVERTYPE 恢复覆盖缺省编辑模式。

13.5 组合命令

假设想看一下系统日期和时间，那么一种方法是改变命令提示符以显示日期和时间，然后马上改回到正常的命令提示符。你可以输入两个单独的命令做到这一点，但用 DOSKEY 可将这两个命令组合成一个命令，同时用 ¶ 将它们分隔。

提示： 按 Ctrl+T 可键入 ¶ 符号。

试着把两个命令组合成一个：

1. 键入下列命令：

```
PROMPT Date:$D  Time:$T$_$_¶PROMPT $P$G
```

第一个 PROMPT 命令使 DOS 显示如下一行：

```
Date:Fri 08-19-94 Time: 10:27:15.88
```

它后面跟两个空行(因为 $_ $_)。然后第二个 PROMPT 命令被显示并执行，从而回

到正常的命令提示符。

组合命令有两个优点。第一，它比输入两个单独的命令快；第二，这个命令被作为一个命令记录在 DOSKEY 命令缓冲区中，因而当调回它们时，用调回一个命令的代价就获得了两个命令。

现在你可通过调回前面的命令在任何时候显示系统日期和时间。

现在试一下：

1. 按向上箭头键调回前面的命令。
   DOSKEY 在命令行显示两个命令。
2. 按 Enter 执行它。
   DOS 显示日期和时间，然后恢复正常的命令提示符。

在一个命令行还可使用多于两个的命令。事实上，你可以在 127 个字符内组合尽可能多的命令。（¶ 不作为一个字符计算）。

13.6 使 用 宏

假设在一天中运行前面的命令许多次，则在 DOSKEY 中你可分配给命令一个名字，如 DT（日期和时间），随后只需键入 DT 就可运行该命令，这称为宏。宏可以可以节约时间和精力，并可减少在键入命令中常出现的错误。你也可以在更长的时间里使用宏，而命令历史的命令常被新命令代替。

图 13.2 显示了创建宏的 DOSKEY 命令格式。

DOSKEY

定义一个宏。

结构：

DOSKEY macroname=text

参数：

macroname	命名一个宏。
text	定义一个与 macroname 相联系的命令。

提示：

宏定义可包含如下特殊字符：

$G	重定向输出。与输出重定向符号（>）相等。
$T	分隔命令。与 DOSKEY 命令的分隔符(Ctrl+T)等价。
$$	指定美元符号字符。
$n	宏运行时，代表指定的参数。n 可以是 1 到 9 的任意数，这与批处理参数 1%到 9%相似。
$*	代表所有参数。$* 被宏名后的所有东西取代。

图 13.2 DOSKEY 命令总结（创建宏）

下面我们从用一个简单的宏 V 代替 VER 命令开始：

1. 键入下列命令可定义 V 宏：

```
DOSKEY V=VER
```

DOSKEY 不会对宏定义进行确认，你得到的所有反应是另一个命令提示符。

2. 要运行这个新宏，键入 V。

DOS 显示版本号。

不能在宏定义中使用如 ¶ 或 > 之类的符号，因为一输入命令 DOSKEY 就对其进行解释，而不会将它们保存在宏中。不过你可使用图 13.2 中的特殊符号。

假设想改变 V 宏，使之打印版本号而不是显示它，同时还打印 DOS 搜寻路径。在命令提示符下，相应的命令如下：

```
VER > PRN ¶ PATH>PRN
```

要把这个命令变成一个宏定义，必须用 $G 代替 >，用 $T 代替 ¶。

1. 键入下列命令定义宏：

```
VER $g PRN $T PATH $G PRN
```

2. 保证打印机已准备好。

3. 键入 V 运行这个宏。

DOS 将打印版本号和搜寻宏。（如果不能打印，参考第 12 章的“在命令提示符工作”中的“重定向命令输出”一节）。

现在让我们试一个更有用的宏。在此之前，你曾用一个组合命令显示日期和时间：

```
PROMPT Date: $D Time: $T$L$L ¶ PROMPT $P$G
```

这个命令你可能会在一天使用几次。让我们将它转变成一个宏 DT：在这个定义中，每个美元符号必须用两次，这样 DOSKEY 就可将它存在宏中，而不是试图马上解释它。这意味着 $D 必须改成 $ $D，$T 改成 $ $T，等等。另外，¶ 必须被 $T 代替。注意这个命令将包含 $ $T 以作为 PROMPT 命令的一部分（显示时间），同时包含 $T 以作为两个 PROMPT 命令之间的命令分隔符号。

按以下步骤定义宏：

1. 键入以下命令定义 DT 宏。（提示：可调回前一个 PROMPT 命令，然后编辑它。这样将节约许多时间）。

```
DOSKEY DT=PROMPT Date: $ $D  Time: $ $T$ $_$ $_$TPROMPT $ $P$ $G
```

2. 键入 DT 试一下新的命令（DT 是刚刚创建的宏）。DOS 会显示日期和时间，然后恢复到正常的命令提示符。如果不正常工作，回到第 1 步再试。

13.6.1　使用可替代的参数

有时可能想在宏中放一些变量参数，以便在运行宏时提供它们。这就使宏更灵活并由此而更有用。假设想创建一个名叫 PS 的宏，以用它来打印任何 DOS 命令的命令格式（从 FASTHELP）。这个宏重定向 command/? 的输出到打印机。

在宏定义中使用符号 $1 到 $9 来表示在哪里插入变量信息。例如，当运行 PS 时提供命令名，可使用 $1 要求命令名：

1. 输入下面的命令定义 PS：

```
DOSKEY PS=$1 /? $G PRN
```

2. 键入 PS PATH 打印 PATH 命令的命令格式。

DOSKEY 用 PATH 代替宏中的 $1，传送下列命令到 DOS：

```
PATH /? >PRN
```

结果，DOS 打印 PATH 命令格式。（如果什么都没打印，则请参考第 12 章中“重定向命令输出”中的“答疑”部分）。

如果在一个宏中使用多个可替代的变量，DOSKEY 用命令行中的第一个参数代替 $1，第二个参数代替 $2，其余由此类推。必须保证在运行这个命令时按正确的顺序放置参数，否则它们将插入宏中错误的位置。不能省略参数，结尾除外。例如，如果一个宏要求两个参数，那么决不能省略第一个参数而保留第二个参数，因为 DOSKEY 在命令行中从左到右读参数。在合适的时候可省略第二个参数，或省略两个参数。

当省略一个参数时会发生什么情况呢？DOSKEY 将用一个空值(null)代替(表示什么都没有)。例如，如果只键入 PS，不带一个命令名，DOSKEY 将创建命令：

```
/? >PRN
```

结果是 Bad command or file name 消息。

提示： 如果看到一个长宏因为在运行它时忘记了包含一个变量参数而出错，可按 Ctrl+C 停止。如果这个宏包含多个命令，必须为其余每个命令按一次 Ctrl+C。

13.6.2 控制 DOSKEY 缓冲区大小

当加载 DOSKEY 时，它额外占用一定数量的内存空间作为一个缓冲区(Buffer)(它只是内存中的一个存储区域)。缓冲区的大小决定了 DOSKEY 可存储多少命令和宏。如果不定义宏的话，DOSKEY 的缺省配置可存储大约 50 个命令。但每个宏都将占去了多个命令的空间。DOSKEY 不会让你用宏占用多于一半的缓冲区，另一半将始终为命令保留。因此，如果计划定义宏，缺省配置有大约 25 个命令的空间，这对大部分人来说已足够了。

使用 DOSKEY 一会儿后，如果发现需要更多的命令，可在下次加载 DOSKEY 时包含 /BUFSIZE选项以申请更多的缓冲区空间。例如，命令 DOSKEY /BUFSIZE=1024 将占用两倍于原来的空间。但是，只有在不得已的情况下才增加缓冲区的大小，因为它会占用其他程序的宝贵内存。相反，如果发现只需要大约 10 个命令和一些宏，可以释放一些不需要的内存空间，这时可通过命令 DOSKEY /BUFSIZE=256 加载 DOSKEY 以将缓冲区减少到 256 个字节(DOSKEY 缓冲器的最小值)。

注释： 不幸的是，没有方法卸载 DOSKEY，除非重新启动或关机。因此改变缓冲区大小的唯一较好的办法是重新启动。使用/REINSTALL 可以在无须重新启动的情况下重新安装 DOSKEY，但这个技术摒弃了 DOSKEY 的第一份拷贝和它的缓冲区内存，从而浪费了大量宝贵的内存空间。如果想改变缓冲区大小，最好是重新

启动。

每次保存一个新的宏时，它都会冲掉缓冲区中的一些命令——当然也有不必要的最早的命令，从而会丢失了一些命令历史。但相反过程却不是这样——新命令不会冲掉宏。如果保存一个宏，除非你重新启动，否则它将始终停留在内存中。

13.6.3　在 AUTOEXEC.BAT 中加载 DOSKEY

每次启动时，DOS 在启动驱动器的根目录下寻找名叫 AUTOEXEC.BAT 的文件，并运行在这个文件中找到的所有命令。因此，你可以将任何想在启动时运行的命令保存在 AUTOEXEC.BAT 中。你也可在 AUTOEXEC.BAT 命令中包含命令加载 DOSKEY，从而使之每次总是可用的，而且你不必为之费神。如果不想使用缺省值，可包含/BUFSIZE 和/INSERT 选项到命令中。第 20 章“基本批处理程序”解释了更多关于 AUTOEXEC.BAT 的内容以及如何在其中加入命令。

宏存在内存中而不是在磁盘中，这既有好处又有坏处。好的方面是，这个宏可运行很快，因为它不必从磁盘加载。坏的方面是，当重新启动或关闭电源时，所有的宏将丢失。但为什么不把常用的宏定义加入到 AUTOEXEC.BAT 中，从而使得在每次启动时这些宏都被定义呢？

如果决定在 AUTOEXEC.BAT 中定义宏，要记住 DOS 遇到的第一个 DOSKEY 命令时加载这个 TSR。同时如果你想的话，它也可定义一个宏。但是，如果想指定缓冲区大小，那么必须在第一个命令中包含/BUFSIZE 参数。

预览： 如果定义的一个宏的名字与另一个程序名相同，宏将超越于程序之上。例如，如果命名一个宏 FORMAT，当键入 FORMAT 时，将访问这个宏，而不是 DOS 的 FORMAT 程序。在第 37 章“批处理程序中的编程逻辑”将告诉你如何使用这个特点取代或提供一定的 DOS 命令。

13.7　浏览命令和宏

图 13.3 显示了两个 DOSKEY 选项，它们用于浏览命令和宏。这两个功能的最为有用之处是可重定向宏和命令到打印机。下面我们试着浏览和打印历史清单：

DOSKEY

显示命令历史和宏清单。

句法：

DOSKEY [/HISTORY] [/MACROS]

参数和选项：

/HISTORY　列出 DOSKEY 缓冲区中所有命令，简写为/H。

/MACROS　列出所有当前宏，简写为/M。

图 13.3　DOSKEY 命令总结(浏览命令和宏)

1. 为观察历史清单，可输入命令 DOSKEY /H。
2. 为打印宏清单，可输入命令 DOSKEY /M >PRN。

预览： 第 20 章介绍了如何重定向命令清单或宏清单到一个文件，以及如何将它变成一个批处理程序。

13.7.1 修订和删除宏

假设你不喜欢一个宏，可通过键入另一个宏定义取代这个相同的宏名，这在本章中已做过多次了。或者，可通过不带命令名的宏定义将它一起删去。下面试着删除在本章中创建的宏：

1. 键入 DOSKEY DT=，删除 DT 宏。
2. 键入 DOSKEY PS=，删除 PS 宏。

13.8 自 测 练 习

在本章中学习了许多关于 DOSKEY 的知识。以下练习帮助你复习学过的内容：

1. 退出使用的所有程序，然后重启动以卸载 DOSKEY。
2. 以最小的缓冲区大小加载 DOSKEY，以插入方式作为缺省编辑模式。
3. 输入一个用以显示 DOS 版本号和程序搜寻路径的命令行。
4. 定义和测试一个用以打印程序搜寻路径的宏。
5. 显示命令历史和宏清单。
6. 删除宏。
7. 通过调回命令历史中的定义命令再次定义宏。
8. 如果喜欢的话，改变缺省编辑模式到覆盖键入。

在以后章节中，还有许多练习调回和编辑命令的机会。我会常常提醒你，这是创建一个新命令最好的办法。在后面我也将提醒你修改 AUTOEXEC.BAT 文件，以加载 DOSKEY 和你喜爱的宏。

第14章 目录管理

简介： 在DOS提示符下，可以实现在Shell下可实现的任何目录管理——甚至更多。

14.1 总 览

你不能在根目录下保存所有的文件。即使空间没有限制，一个目录包含几千个文件也将完全没法管理。硬驱越大，将文件组织成一个设计良好的子目录层次就越重要。

你必须能创建目录，管理目录树，在目录中随意移动，以及定位想要的文件。本章将告诉你怎么做。在这里你将学会：

- 改变缺省驱动器和目录
- 显示一个目录树
- 列出一个目录的内容
- 列出一个目录中指定的文件
- 找到在硬盘上丢失的一个文件
- 创建一个新的目录
- 为一个目录更名
- 删除一个目录

注释： 可以没有限制地在本地驱动器上创建和管理目录结构。但是否能在一个网络驱动器上创建，罗列，搜寻，更名，或删除目录，要取决于你的访问权限。

14.2 改变缺省驱动器和目录

每次启动时，当前驱动器的启动目录就变成了当前目录，除非AUTOEXEC.BAT切换到了另一个目录。你可能想改变到其他驱动器和目录，以便与应用程序和它们的数据文件在一起工作。

为改变驱动器，可输入想要的驱动器名，后跟一个冒号，如A:或D:。

注释： 在DOS命令中驱动器名后常要跟冒号，这样DOS才会认为它们是驱动器名而非文件名。

试着改变到驱动器A，然后再返回：

1. 插入一张盘到驱动器A。（只要磁盘格式化过就可以）。
2. 键入A:

DOS用驱动器A作为缺省驱动器，用驱动器A上的缺省目录（可能是根目录）作为

当前目录。你可在命令提示符中看到新的当前驱动器和目录。

3. 键入 C:

DOS 用驱动器 C 作为缺省驱动器,用驱动器 C 上的缺省目录作为当前目录。你将在命令提示符下看见新的当前驱动器和目录。

使用 CD 命令改变缺省目录(如图 14.1 所示),你也许可看到被用作 CD 的一个替代命令的 CHDIR。这两个命令是相同的命令。

CD

改变缺省目录或显示缺省目录名。

句法:

CD [drive] [pathname]

参数:

none	显示驱动器缺省目录。
drive	显示指定驱动器的缺省目录。
pathname	指定新的缺省目录的名字。pathname 必须是一个已存在的目录。

图 14.1 CD 命令总结

试着使用 CD 在硬驱上改变目录:

1. 键入 CD\转到根目录。
2. 键入 CD DOS 转到 DOS 目录。

注释: 不要忘记,一个路径名必须显示如何从当前目录到达想到的位置,上面这个路径名 DOS 只有在根目录为当前目录时才起作用。从其他目录,将不得不使用\DOS。

14.2.1 显示在另一个驱动器上的缺省目录的名字

正常情况下,命令提示符显示当前目录的名字(也就是说,当前驱动器上的缺省目录)。但如果想知道另一个驱动器上的缺省目录的名字,怎么办呢?下面的操作(其中包括命令提示符),显示了如何显示驱动器 D 的缺省目录名:

```
C:\DOS>CD D:
D:\ORDERS\BACKLOGS

C:\DOS>
```

在这种情况下,驱动器 D 上的缺省目录是\ORDERS\BACKLOGS。注意命令提示符继续显示当前目录,而在其他驱动器上的缺省目录只作为一个消息显示。如果另有一个驱动器,那么请输入 CD d:以找到这个驱动器上的当前目录。

14.2.2 改变另一个驱动器上的缺省目录

CD 命令不一定只在当前目录下使用,有时用它改变另一个驱动器上的缺省目录是很

方便的。下面试着改变另一个驱动器上的目录：

1. 保证在驱动器 A 有一个磁盘。
2. 转到驱动器 A。
   命令提示符显示 A 是缺省驱动器。
3. 输入 CD C:\DOS 使 DOS 目录是驱动器 C 上的当前目录。
   你将不会看到这个变化的任何证据。命令提示符不改变。
4. 转到驱动器 C。
   现在可看到命令提示符中 DOS 目录是当前目录。

14.3 显示目录树

当目录结构变得复杂时，容易忘记它是如何组织在一起的，这时可使用 TREE 命令以得到一张目录结构图(如图 14.2 所示)。TREE 描绘出结构中目录之间的关系，特别像在 Shell 中的目录树。

TREE

图形化显示一个目录结构。

句法：

TREE [drive|pathname]

参数和选项：

none	显示以当前目录开始的目录结构。
drive	显示指定驱动器的目录结构。
pathname	显示指定目录的目录结构。

图 14.2 TREE 命令总结

为显示硬盘的整个目录树，可键入 TREE\。

目录树显示在显示器上(如图 14.3 所示)。如果它是一个长树，则将自动翻卷，使之顶部离开屏幕的顶部。

提示： 如果在阅读之前目录树已翻卷出屏幕，那么你可以做如下几件事情：

- 在运行命令之前将手指按在 Pause 键上。当信息在屏幕上滚动时，按 Pause 键暂停，按任何键又重新滚动。成功与否取决于你和你的系统的速度。你可能会发现其他两个技术更令人喜欢。
- 重定向输出到打印机，如 TREE\>PRN。
- 把输出导向 MORE，如 TREE \| MORE。管道 MORE 将在第 36 章“命令行状态下的 DOS 应用”中解释。

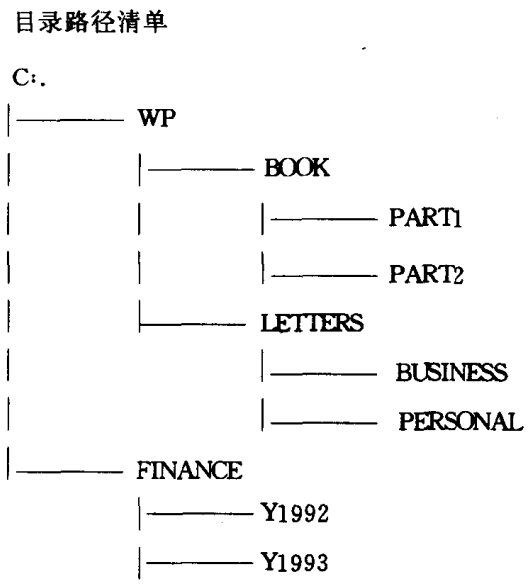

图 14.3 TREE 清单

如果想显示一个没有子目录的目录树，如 DOS 目录，那么你将看到消息 No Sub-directories exist。

预览： TREE 有两个不常使用的参数。其中/A 要求连接线用文本字符代替图形字符，这对于无法处理图形字符的打印机而言效果会更好；/F 列出包含在每个目录中的文件。参见命令参考中关于每个参数的完整细节描述。

14.4 列出一个目录

可以使用 DIR 命令观察一个目录的内容(如图 14.4 所示)。

试着列出一些目录的内容：

1. 键入 DIR \列出根目录内容。
   DOS 在显示器上显示根目录的内容。
2. 键入 DIR \DOS 列出 DOS 目录内容。(注意：从 DOSKEY 调回前面的命令并加以编辑便可得到这个命令)。
3. 为在 DOS 目录清单中分页，键入 DIR \DOS /P。(注意：可通过调回编辑前面的命令得到)。
   DOS 显示 DOS 目录清单的第一页。这时屏幕最后一行是消息 Press any key to contiune。
4. 按任意键。
   DOS 显示目录清单的下一页。
5. 继续按任意键，直到清单结束并出现命令提示符。

DIR

显示一个或多个目录的内容。

句法：

DIR [filespec] [/P] [/W] [/A] [/O] [/S] [/B]

参数和选项：

none	列出当前目录内容。
filespec	指定列出的目录和/或文件。
/P	一次显示一页。
/W	以宽格式显示清单，每行五个名字。
/A	列出所有的文件，而不考虑属性；若没有/A，则不列出隐含文件和系统文件。
/O	将清单排序。
/S	列出子目录的内容。
/B	抑制头部和总结信息的显示，并且列出每个条目的完整的文件描述(包括路径)。

图 14.4 DIR 命令总结

一个目录内容清单的显示格式大致如下：

```
Volume in drive C is MY HARDDISK
Volume serial number is 152C-B387
Directory of C:\WP\LETTERS\PERSONAL
.                <DIR>          01-17-90      6:21a
..               <DIR>          01-17-90      6:21a
JANJOHNS   LET       1,152      02-22-94      12:08p
FOLK-FEB   LET       3,840      02-28-94      10:49a
DOR-BDAY   FAX       1,792      03-15-94      12:07p
CONGRESS         <DIR>          01-19-94      7:30p
FOLK-MAR   LET       1,787      03-06-94      9:30a
CONNIE     FAX       1,659      03-04-94      9:31a
FOLK-APP   LET       1,280      04-10-94      11:48a
MOM-BDAY   LET       1,244      09-12-94      9:30a
        10 file(s)     1,2754   bytes
                    52,318,464  bytes free
```

注释： 前一个例子显示了一个较新的DIR清单。在DOS 6.2以前，字节值不用逗号隔开，因此很难读。许多用户在屏幕上用手指将数字如10057238分成三个一组，以此来发现一个驱动器是否还剩1MB或10MB空间，这容易把屏幕弄乱。

头部显示卷标、系列号和被列出清单的目录的名字。主体显示目录的内容。每个子目录在开始处包含两项：.表示目录自己，..表示目录的父目录。<DIR>显示此处有目录，而没有文件。注意这个目录还有一个子目录叫CONGRESS，它显示在清单的下面。

对于目录项，日期和时间显示的是目录创建时的日期和时间。对于文件，它显示的是文件创建或最后一次修改的日期和时间。文件大小给出包含在一个文件中的实际字节数，而不是它占用磁盘上的空间量。实际占用驱动器空间由驱动器上簇的大小和驱动器是否压缩决定。如果想拷贝或移动文件到另一个驱动器，那么将与这个数字无关。

在清单最后的总结行显示了列出的项数(包括文件和目录)和这些文件的字节数。(子目录中的文件不算在总数之内)。此外还显示了驱动器上自由字节的数目。不要希望字节数和自由空间的总和会等于磁盘的大小,因为在磁盘上有比列出的更多的文件。

如果你是在一屏上看整个长目录,那么你可能需要用/W 选项以按宽格式列出。这种格式不显示大小和时间,而是按每行分五列的格式列出以显示更多的文件。试着显示 DOS 目录内容:

1. 键入命令 DIR \DOS /W。
   DOS 目录的文件的宽清单出现在屏幕上,但它还是太长,屏幕容不下,因而不能看见清单的顶部。
2. 键入命令 DIR \DOS /W /P,在屏幕上将宽格式清单分页。
   现在可看到整个清单分成两页。

14.4.1 依照属性选择文件

缺省状态下,DIR 命令不显示系统或隐含文件。如果想在不考虑属性的条件下浏览所有的文件,那么可使用/A 开关。你不能在目录内容清单上看到属性,但它们还是用来控制清单的内容。第 16 章"增加的文件管理"解释了如何用 ATTRIB 命令在命令提示符下显示和管理属性。

下面试着用/A 开关显示根目录清单:

1. 在启动驱动器上转到根目录。
2. 键入 DIR /P,以列出所有非隐含和系统文件。
3. 键入 DIR /P /A,以列出这个目录中的所有文件,包括隐含和系统文件。
   你可以在这第二个清单中看到 DOS 核心程序文件,如 MSDOS. SYS 和 IO. SYS 文件。如果有隐含和系统文件,使用/A 时它们才显示。

目录也可以拥有文件能拥有的所有属性。一些应用程序(如 Delete Sentry 删除保护实用程序)会在驱动器上放置隐含目录。用 DIR 不能看到这些目录,除非使用/A 开关。(隐含目录也不显示在一个 TREE 清单中——但这里没有方法可以显示)。

你可增加属性代码到/A 选项以列出有特定属性的文件(见表 14.1)。例如,/AH 使 DIR 列出隐含文件。当包含一个或多个属性代码时,只有满足条件的文件才列出。试一下使用属性代码:

1. 保证处在根目录下。
2. 键入 DIR /AH 只看目录中的隐含文件。
   注意这个清单与前面练习的清单的区别。只有隐含文件包含在清单中。
3. 键入 DIR /AD 列出当前目录的所有子目录。
4. 使用多个代码可以限制显示具有组合属性的文件(或子目录)。键入 DIR /AHD 看有隐含和系统属性目录——另言之,就是隐含目录的项。
   如果根目录没有隐含子目录,将显示消息:File not found。
5. 可以用减号列出不具有指定属性的文件。键入 DIR /A-D /P 列出除子目录外的所有文件。

6. 键入 DIR /AH—D 看隐含文件而不是隐含子目录。

表 14.1 DIR 命令的属性代码

代码	含义
A	归档属性
D	目录属性
H	隐含属性
R	只读属性
S	系统属性

14.4.2 清单排序

缺省状态下,DOS 不会将目录内容清单排序。文件和子目录按它们实际出现在目录中的顺序排列。由于 DOS 管理目录的方法,这个顺序常常是完全随意的。大部分时间,你会发现清单按某种顺序是很有用的。/O 选项按文件名的字母顺序排列文件,其中所有的目录列在开头。为在系统上试一试/O 选项,可键入命令 DIR /O /P。

注意与前面练习的条目顺序之间的不同。

你可增加顺序代码到/O 选项以指定清单的排列顺序(表 14.2)。下面我们在系统上试一些顺序代码:

1. 键入 DIR /ON 按名字排序。
2. 键入 DIR /OS 按大小排序。
3. 你可在代码前增加一个减号以使顺序完全相反。键入 DIR /O—S 按大小递减的顺序排列入口。
4. 如果指定多个顺序代码,那么 DOS 将按第一个代码排序。当两个或多个项有相同的排序代码时,DOS 按第二个代码排序。键入 DIR /O—DNE 按日期递减的顺序排列目录项;对于有相同的日期的文件,再按名字递增的顺序;对于有相同的日期和名字的文件,再按扩展名递增的顺序。
5. G 顺序代码将所有的子目录清单放在清单的首部。键入 DIR /OGNE 按名字和扩展名排列,其中子目录在开始。(如果键入/O 不带任何顺序代码,这就是缺省顺序)。

表 14.2 DIR 命令的顺序码

代码	含义
C	按压缩率的数字顺序
D	按日期的数字顺序
E	按扩展名的字母顺序
N	按名字的字母顺序
S	按大小的数字顺序
G	把子目录列在清单的开始

14.4.3 筛选名字

有时可能你只想看目录清单中的一个文件或一个文件集。也许只想找出该文件是否在

这个目录中，或者找到它的大小，或日期和时间。当增加一个文件名到一个 DIR 命令后时，DIR 只显示与这个文件名匹配的文件，筛选掉其他所有文件。

HELP. HLP 文件包含关于 HELP 程序的信息，这在第 12 章“在命令提示符下工作”中学过。试着列出它的目录项：

1. 键入命令 DIR \DOS\HELP. HLP，列出 DOS 目录中名为 HELP. HLP 的文件。
   你将看到头部行，一个 HELP. HLP 文件清单以及总结行。
2. 如果你将为一个目录键入几个 DIR 命令，转到这个目录常常更容易，因为不必在文件名前加路径名。键入 CD \DOS 转到 DOS 目录。
   命令提示符显示 DOS 为当前目录。
3. 键入 DIR HELP. HLP 再次看一下 HELP. HLP 文件。（注意：调回上一个 DIR 命令，编辑后可产生这个命令，当然得先加载 DOSKEY）。

你可键入一个全范围的文件名以列出与之匹配的所有文件。

1. 键入 DIR *. HLP 列出 DOS 目录中所有以. HLP 为扩展名的文件。
   这样的文件并不多，这些文件包含关于 DOS 的几个实用程序的信息。
2. 在 DOS 目录中可能没有. XXX 文件。这时键入命令 DIR *. XXX 看会发生什么。
   你将得到一个 File not found 消息。在这种情况下，DIR 显示头部信息但没有总结，因此你不能找到驱动器上有多少可用空间。

14.4.4 列出目录分枝的清单

你可能想列出一个目录分枝的清单。从分枝顶部开始并使用/S 选项是一个快速的方法。/B 选项和/S 一起使用将使列出清单更易处理，因为它删去了头部和总结行。你只看到入口，包括完整的路径，这样你知道每个文件在哪里可找到。

试着列出硬驱上所有的. HLP 文件：

1. 键入 DIR *. HLP /S，从根目录开始，列出每个目录的完整清单。
   在有机会阅读之前，大部分信息卷离屏幕。但是清单的最后一行会显示一共有多少个. HLP 文件被列出，以及它们包含的字节数和驱动器上剩下的空间。如果只想知道驱动器上有多少. HLP 文件以及它们共占多少字节，这是一个好的方法。
2. 在前一个命令中使用/P 选项将清单分页。键入 DIR *. HLP /S /P 命令。
   DOS 一次一页地显示. HLP 文件清单。
3. 试着增加/B 选项到命令：DIR *. HLP /S /P /B

注释： 选项可以任意顺序键入。

一个完全不同的显示出现了。没有头部和总结行，每行显示完整的路径名。带有/B 或/W时不显示文件大小、日期和时间。

14.4.5 找到一个丢失的文件

如果忘记了一个文件在驱动器上的位置，那么处理起来也许比想象的要容易，/S 和/B 选项会帮助你找到它。命令 DIR \filespec /S /B 从根目录开始（因为文件名前有一个\），搜

寻驱动器上的所有目录，并列出满足描述名的所有文件。/B 选项在每个项上包含完整的路径名，因此很容易定位文件。

试着在硬驱上通过键入 DIR \COMMAND.COM /S /B 寻找 COMMAND.COM 文件。

你会看到不同目录下的两个或三个 COMMAND.COM 文件拷贝的清单。从这个清单中，找到想要的文件是容易的。

预览： DIR 命令有一些增强特点，这将在本书后面学习。例如，当学习第 34 章“DriveSpace 和 DoubleSpace”的 DriveSpace 时将学习如何压缩信息。在第 43 章“DOS 及其运行环境”中，还将学习如何在 DIR 中设置缺省选项。

14.5 创建一个目录

目录如何产生呢？当然，它们中的一些在安装程序时创建，但你将自己创建许多目录，图 14.5 显示 MD 命令的总结，它用来创建一个目录(MKDIR 与 MD 相同)。

MD

创建一个新目录。

句法：

MD pathname

参数：

pathname 指定新目录的名字和位置。

注意：

新的目录的完整路径名包括反斜杠，从根目录开始，包括这个新目录最多不得超过 63 个字符。

Pathname 对父目录来说要是独一无二的，不能使用两个相同的属于同一父目录的目录名或文件名。

图 14.5 MD 命令总结

你可以看到，创建一个新的目录是一件简单的事情。和其他目录命令一样，在这时要保证你的路径名能告诉 DOS 如何到达想到的位置。你可以从当前目录开始，但如果没有把握的话，指定目录的完整路径名(从根目录开始)。

注释： 可用 DIR 和 TREE 列出一个只读驱动器，如一个 CD-ROM 驱动器的目录。你可以转到一个只读驱动器，但不能以任何方式改变其目录结构，也不能在这些驱动器上使用 MD，MOVE 或 RD 命令。

试着创建一个名字为**PRACTICE**的目录作为 DOS 目录的子目录：

1. 键入命令 MD \DOS\PRACTICE。

   没有新闻才是好的新闻。如果没有得到错误消息，表明目录创建成功。

2. 试着转到新目录。键入 CD \DOS\PRACTICE。

14.6 为目录更名

DOS 6 的一个新命令使你能改变一个目录的名字。在 DOS 6 以前，不得不使用 Shell 或其他程序来更名一个目录。现在在命令提示符下使用 MOVE 命令就可以做这样事了(如图 14.6 所示)，第 15 章"拷贝和移动文件"解释了这个命令如何将文件从一个目录移到另一个目录，但是，当你使用一个目录名代替一个文件名时，MOVE 只是更名目录。

MOVE

更名一个目录。

句法：

MOVE pathname newname

参数：

pathname 指定想更名的目录。

newname 指定新名字。

注释：

如果 pathname 包含一个父目录，newname 也要包含相同的父目录。例如，如果 pathname 是 C:\DOS\PRACTICE，newname 必须是 C:\DOS\name。

图 14.6 MOVE 命令总结

试着改变 PRACTICE 目录的名字：

1. 你不能更名当前目录，因此如果 PRACTICE 是当前目录，用 CD.. 命令转到它的父目录。

   命令提示符将显示 C:\DOS 是当前目录。如果不是，键入 CD\DOS 到达 DOS 目录。

2. 键入命令 MOVE PRACTICE EXERCISE。

   你将得到消息 c:\dos\practice => c:\dos\exercise[ok]。

答疑： 试图用 MOVE 更名一个目录时，得到消息 permission denied，出什么问题了呢？

首先，保证你不是在更名当前目录。如果是，转到另外一个目录再试。

第二，保证你没有试图改变目录的位置。当前名和新名必须指定相同的驱动器和父目录名。如果不知如何指定位置，转到其父目录再指定不带路径的目录名，如 MOVE PRACTICE EXERCISE。

第三，保证没有指定一个全程的路径名。MOVE 一次只可改变一个目录的名字。

14.7 删除一个目录

如果有一个空目录想删除，可使用RD命令(如图14.7所示)。RMDIR与RD相同。如果想删除一个带有文件或子目录的目录，最好使用DELTREE，该命令将在下面讨论。

RD

删除一个目录。

句法：

RD pathname

参数：

pathname 指定想删除的目录，它不能是驱动器上的缺省目录。

图14.7 RD命令总结

练习删除刚创建的EXERCISE目录：

1. 在删除一个目录之前它必须为空。列出EXERCISE的内容，保证其为空。
   你应该得到头部行后的File not found消息。
2. 转到DOS目录，它是EXERCISE的父目录。(注意：如果EXERCISE是当前目录，可键入CD..转到它的父目录，而无需拼出其目录名)。
   命令提示符将显示C:\DOS为当前目录。
3. 现在通过RD EXERCISE删除EXERCISE目录。
   如果没有得到错误消息，目录被成功删除。
4. 键入DIR EXERCISE以保证已没有EXERCISE目录。你将得到消息File not found。

答疑： 试图删除一个目录时，得到消息Invalid path,not directory,or directory not empty。

首先，要保证正确地键入路径名。如果不对，再试。

其次，保证不是在删除当前目录。如果是，键入CD..转到父目录，再试RD命令。

最后，目录可能没有真正空。如果DIR显示为空，但还不能删除它，试着使用DIR /A命令看看是否有一些隐含文件、系统文件或子目录在目录中。

如果看了它包含的内容后仍想删除，使用DELTREE命令删除整个分枝、文件和其他所有东西。

14.8 删除一个目录分枝

在DOS 6以前，一次只能删除一个目录，而且它必须为空。但DOS 6有一个新命令DELTREE，它可删除整个目录分枝，包括隐含、系统和只读项。在图14.8中，你可以看到这个有力但危险的工具。如果使用这个命令，请务必谨慎！

在这个练习中，你将创建一个目录分枝，然后删除它：

1. 创建一个目录 EXERCISE，作为根目录的子目录。
2. 创建一个目录 PRAC1，作为 EXERCISE 的子目录。
3. 创建一个目录 PRAC2，作为 EXERCISE 的子目录。
4. 看 RD 和 DELTREE 有何不同，先使用 RD。试着使用 RD 命令删除 EXERCISE。
   你得到消息：Invalid path，not directory or directory not empty，因为 RD 命令不能删除有子目录的目录。
5. 键入 DELTREE EXERCISE 删除整个目录分枝。

   Delete directory "exercise" and all its subdirectories. [y/n]

6. 按 Y。
   你看到消息：

   Deleting exercise...

DELTREE

删除一个目录及它的所有文件和子目录。
句法：

DELTREE [/Y] pathname ...

参数和选项：

pathname	要删除的指定目录枝上一级的目录。
/Y	越过确认步骤。

图 14.8 DELTREE 命令总结

删除一个巨大的目录分枝及其所有文件可能会花一点时间。命令提示符在 DELTREE 完成后返回。

可在DELTREE命令中包含多个路径名，以删除多个目录分枝。无论包含多少路径名，均可用/Y 选项跳过确认步骤。

1. 创建三个目录：PRAC1，PRAC2 和 PRAC3。每个目录都是根目录的子目录，这样它们都在相同的层次上。
2. 键入 DELTREE /Y PRAC1 PRAC2 PRAC3 删除所有三个目录，无需确认。
   你会看到消息：

   Deleting prac1...
   Deleting prac2...
   Deleting prac3...

提示： 即使在一个目录没有子目录时使用 DELTREE 也很方便，因为它不需使目录首先为空后才可删除。

预览：DELTREE 也可用来删除一个目录分枝的指定文件，而不删除目录分枝。DELTREE 无需特殊努力便可删除一个隐含或只读文件。第 39 章“目录管理”显示了如何实现这些功能。

14.9 自测练习

本章描述了一些目录管理技术。下列各项将有助于你进行更多的实践：

1. 创建目录 ONMYOWN 和 NOHELP 作为根目录的子目录。
2. 创建目录 MYDIR，YOURDIR 和 OURDIR 作为 ONMYOWN 的子目录。
3. 显示 ONMYOWN 的目录分枝。
4. 显示 DOS 目录的内容，一次一页。
5. 在硬驱上寻找 CONFIG. SYS 文件。
6. 更名 NOHELP 为 NEWNAME。
7. 用一个命令删除 ONMYOWN 和 NOHELP 目录。

第 15 章 拷贝和移动文件

简介： 以前在命令提示符下拷贝和移动文件常会发生一些问题。DOS 6.2 对此作了改进，从而使得操作更为安全。

15.1 总 览

DOS 给出了几种拷贝和移动文件的方法。本章描述了三个最常用的命令，其中 COPY 和 XCOPY 用来拷贝文件，而 MOVE 用来移动文件。

- 拷单个的文件到相同的目录，给文件重新命名
- 拷贝文件到另一个目录，带或不带新名
- 拷贝文件到打印机或其他设备
- 从一个目录分枝拷贝文件到另一个目录分枝
- 移动文件到另一个目录

网络注意： 本章的过程和命令在本地驱动器上工作。它们可不可以在网络驱动器上工作，取决于你的访问权限。例如，你可以拷贝网络驱动器上的文件到本地驱动器，但不能移到本地驱动器，因为你没有权利从网络驱动器上删除文件。

15.2 COPY 命令的使用

DOS 最原始的拷贝命令是 COPY(图 15.1)，它可用来拷贝一个或多个文件到目的地。

COPY

句法：

COPY source [destination][/[-]Y]

参数和选项：

source	指定将被拷贝的文件。
destination	指定拷贝到的位置和/或名字。
/Y	不需确认就覆盖目标文件。
/-Y	覆盖目录文件时需要确认。

图 15.1 COPY 命令总结

COPY 最简单的用法可能是在拷贝文件时，把拷贝放在相同的目录中，但换上一个新的名字。试着对 HELP.COM 作一次拷贝。

1. 键入 CD \DOS 转到 DOS 目录。(你可能不在源目录中作拷贝，但这会更困难，因为不得不在源文件和目标文件名中包含路径名)。
2. 键入 COPY HELP.COM PRACTICE.XXX。
   屏幕显示 1 file(s) copied(拷贝了一个文件)。

拷贝文件到另一个目录几乎一样容易：

1. 键入命令 MD TRIAL 创建一个新目录 TRIAL 作为 DOS 目录的子目录。
2. 键入 COPY PRACTICE.XXX TRIAL，拷贝 PRACTICE 到一个新目录。

因为命令未指定目标文件名，新的拷贝也被命名为 PRACTICE.XXX。如果想要有一个不同名字的拷贝，在目标中应包含新名。

1. 键入 COPY PRACTICE.XXX TRIAL\FILE 1.XXX，以拷贝文件到 TRIAL 目录，并将它命名为 FILE1.XXX。屏幕显示 1 file(s) copied。
2. 键入 DIR TRIAL 列出 TRIAL 目录中的文件。
   屏幕显示 PRACTICE.XXX 和 FILE1.XXX。

在这些例子中，原始文件不被改变，拷贝的日期和时间与原始文件相同。拷贝的归档属性打开，隐含、系统和只读属性被关闭。

答疑： 我想访问一个文件，得到消息 File not found(未找到文件)。为什么 DOS 不能找到我的文件？

首先检查一下以确保键入的文件名正确。如果键入错误，纠正命令重新访问。

其次确保处于正确的目录。如果文件不在当前目录，必须在文件名前加上路径名以告诉 DOS 在哪儿找这个文件。例如，如果当前驱动器为 C:\DOS，要想在驱动器 A 的根目录下访问名为 RUNYMEDE 文件，必须按以下格式键入：

A:\RUNYMEDE

最后看一下文件是否是一个隐含或系统文件。许多 DOS 命令，包括 COPY、XCOPY、MOVE、REN 和 DEL，不能访问隐含或系统文件。为检查文件的属性，可键入 ATTRIB filespec。如果这个文件有隐含或系统属性，而你还是决定处理它，键入 ATTRIB filespec －H(如果它是隐含的)，或－S(如果它是系统文件)关闭这些属性。如果既是隐含又是系统，在同一个命令使用－H 和－S 即可。(欲了解更多内容，参见 ATTRIB)。关闭属性后，就可像正常文件那样处理它们了。

如果不指定目标目录，COPY 将把新文件放在当前目录中。试着从 TRIAL 拷贝 FILE1.XXX 到当前目录：

1. 键入命令 COPY\TRIAL\FILE1.XXX。
   屏幕显示 1 file(s) copied。
2. 在当前目录中键入 DIR FILE1.XXX 列出这个文件。

答疑： 我试图拷贝一个文件，屏幕显示 File cannot be copied onto itself（不能拷贝文件自身）。出什么问题了呢？

不能在一个文件自身的目录中不改变名字就拷贝它。必须指定一个新文件名，或者一个不同的目录名，抑或同时指定两者并放到 COPY 或 XCOPY 命令的目的参数中。

注释： 可以从只读驱动器如 CD-ROM 驱动器拷贝一文件，但不能拷贝文件到其中。而且不能从那儿移动文件或移文件到其上。

15.2.1 多个文件的拷贝

可使用一个全局文件名拷贝几个文件。下面试着从 DOS 目录拷贝一些. HLP 文件到 TRIAL 目录中。为了减轻系统工作量，我们只拷贝名字以 D 开头的文件：

1. 确保在 DOS 目录中。
2. 键入 COPY D*.HLP TRIAL 拷贝一些. HLP 文件到 TRIAL 目录。
   在 DOS 进行拷贝时你将看见一系列文件名。

注释： 每个拷贝的名字、日期和时间与原始文件相同，打开归档属性，关闭只读、隐含和系统属性。

答疑： 我在拷贝一组文件的过程中得到消息 Disk full，拷贝停止。如何才能完成文件拷贝工作？

首先，你需要在磁盘上释放一些空间。删除一些不必要的文件或压缩磁盘。（不一定要从拷贝任务的目标目录删除文件，从驱动器上任何目录上删除文件都能释放所需要的空间）。如果不能找到一些空间，那就不能用这个磁盘完成工作。

要特别注意：如果你经常使用 DOS 的 Delete Sentry，但有时碰巧将它关闭了，那么驱动器可能在它还没满的时候就显示已满了。这时你可以打开 Delete Sentry 以释放空间，以便完成拷贝任务。

没有可行的方法从中间开始继续工作。唯一的办法是从头开始拷贝已拷贝过的文件。可在 COPY 或 XCOPY 命令中使用/Y 选项，这样就不必在每个文件再拷贝时确认一次。但是，如果目标目录中包含一些其他不应被替换的文件的话，则应省略/Y 以便可确认每次取代。

当通过在目标中包含一个全局的文件名进行多份拷贝时，你可以改变文件名。试着拷贝所有的以 D 开头的. HLP 文件到 TRIAL，但将它们的扩展名改为 *.XXX。

1. 键入命令 COPY M*.HLP TRIAL * .XXX
   拷贝时将显示一系列文件名。这些都是源文件名，而非目标文件名。
2. 键入 DIR TRIAL 看看 practice 目录的所有文件。
   你将看到 D*.HLP 文件和 M*.XXX 文件。

15.2.2 替换已存在的拷贝

有时当你试图拷贝一个文件到另一个不同的目录时，那个目录已有一个相同名字的文

件。DOS 不允许在一个目录中有两个同名文件，因此必须采取一些措施。处理这个问题的方法在 DOS 6.0 和 DOS 6.2 中有很大不同。

在 DOS 6.0(和更早的)版本中，COPY 会自动覆盖已存在的文件，而不发出警告，除非该文件有只读属性。UNDELETE 不把取代视为删除，因此它不能恢复前面版本的任何信息。一般情况下，一旦 COPY 覆盖了一个文件，就不能通过任何 DOS 工具将它恢复。

提示： 如果不小心用 Move 或 COPY 操作覆盖了一个重要文件，Norton Utilities 的 Unerase 工具有时能恢复全部或部分被覆盖的文件的数据。但在恢复数据之前不要进行任何其他操作。事实上，甚至不应将机器关闭，保持运行状态，直到恢复需要的数据(如果需要等较长时间且无其他屏幕保护机制，直到准备运行 Unerase)。

对于 DOS 6.2 和 6.22，COPY 对已存在的文件添加了保护措施。当你正在命令提示符下工作时，它会问你是否想覆盖已存在的文件。如果键入 Y，文件将被取代；如果键入 N，拷贝被取消，可用另一个名字再试。也可回答 A，表示 All(全部)，它取消剩下工作的保护措施。也就是说，后面的所有文件都将被覆盖而不警告。如果已经知道将覆盖一个或多个文件，并想从开始就取消保护措施，可键入/Y 选项在命令中。

答疑： 我想访问一个文件，但出现消息 Access denied(无访问权限)。为什么 DOS 不处理我的文件？

文件的只读属性一般是打开的。许多 DOS 命令，包括 MOVE、REN 和 DEL，都不处理一个只读文件。COPY 和 XCOPY 同样不会覆盖一个只读文件。

如果确实需要处理这个文件，必须首先关闭只读属性。这时键入命令 ATTRIB filespec －R 关闭这个属性，然后，就可以作为正常文件对其进行处理(参阅 ATTRIB 命令以了解更多的内容)。

试着从 DOS 再拷贝 D*.HLP 文件到 TRIAL：

1. 确认在 DOS 目录中。
2. 键入 COPY D*.HLP TRIAL 开始拷贝。
   如果使用的是 DOS 6.0 或更早的版本，拷贝直接生成，以下的练习也就不起作用了。
   如果使用的是 DOS 6.2 或更新的版本，则会显示 Overwrite TRIAL\filename(Yes/No/All)。
3. 键入 Y 覆盖第一个文件。
   屏幕为下一个文件而显示 Overwrite TRIAL\filename(Yes/No/All)。
4. 按 N 跳过下一个文件。
   再次显示上面的屏幕信息。(这时显示很快，因为 DOS 不必花时间拷贝文件)。
5. 键入 A 覆盖所有剩下的文件。
   列出剩下的文件，紧接是拷贝的文件的数目。

现在再试一下同样的练习，但这次使用/Y 选项关闭确认功能：

1. 确认在 DOS 目录中。
2. 键入 COPY D*HLP TRIAL /Y 开始拷贝。
   拷贝直接进行而无进一步的提示。

注释： DOS 的覆盖保护对于批处理文件中的命令来说工作方式就不一样了，这里它自动关闭，必须使用/—Y 选项将它打开，这在第 20 章“基本批处理程序”将具体讨论。

预览： 如果不喜欢新版本 DOS 的缺省覆盖保护，可以改变它，第 43 章“DOS 及其运行环境”讨论如何改变。

15.2.3 用 COPY 快速打印文本文件

拷贝命令的源或目标可以是一个设备，如 COM1（第一个串行接口）或 PRN（缺省打印机）。参考表 13.1 中的完整设备名。

网络注意： 如果涉及的设备是一共享资源，网络软件将向正确的打印机输出结果，把它放在队列中等待打印。

在 COPY 中使用设备最普通的方法是快速打印一个 ASCII 文本文件。试着打印 README.TXT：

1. 确认打印机已准备好。（参看第 9 章“Shell 中的其他文件管理”中“通过 Shell 打印”的一系列检验项）。
2. 输入 COPY README.TXT PRN 打印 README.TXT。
   如果不打印，参见第 12 章的“命令提示符下工作”中标题为“重定向命令输出”一节以寻求问题的解决方法。

这种方法打印出来的结果完全没有格式。COPY 甚至不像 PRINT 那样分页，而只是打印到具有连续形式的打印纸上。用这种方法打印文件的唯一优点是不必加载和初始化 PRINT。

注释： 有时，一个软件包含的 README 文本有标题、页号和插入的空行，以便更好地用 Copy 打印出来。这时只需使用标准大小的纸张并正确放置即可。

预览： COPY 有几个可能有帮助的功能。例如，它可把两个或多个文件连接成一个文件。第 40 章“高级拷贝技术”将讨论 COPY 的增强功能。

15.3 XCOPY 命令的使用

DOS 包含一个更新、更快、更有力的用于文件拷贝的命令，称为 XCOPY（extended copy：扩展拷贝）。使用 XCOPY 比 COPY 快许多的一个原因是它处理多个文件的方式。使用 COPY，每个文件单独拷贝，而 XCOPY 一次能转移大量数据到内存，然后写到目标目录。有时它两步就能做完整个工作。图 15.2 总结了 XCOPY 命令。

除速度快以外，XCOPY 比 COPY 更好的一个主要优点是它能拷贝子目录。XCOPY 可拷贝整个分枝到目标驱动器上。

警告： 和 COPY 相似，在 DOS 6.2 版本以前，XCOPY 在覆盖一个目标目录中的同名文件时不加警告。DOS 6 的 UNDELETE 工具不能用来取消以这种方式对一

个文件的覆盖。从 DOS 6.2 开始，XCOPY 以与 COPY 相同的方式保护文件。

XCOPY

拷贝文件和子目录

句法：

XCOPY source [destination][/S][/E][/[－]Y]

参数和选项：

source	指定想拷贝的文件。
destination	指定拷贝写入的位置，缺省为当前目录。
/S	扩展拷贝到以当前目录开头的整个分枝。
/E	当拷贝到整个分枝时拷贝空子目录，如果使用/E，必须选/S。
/Y	取代目标文件时不需确认。
/－Y	取代目标文件时需要确认。

图 15.2　XCOPY 命令总结

试着用 XCOPY 拷贝 HELP.COM 到 TRIAL：

1. 确认 DOS 是当前目录。（不一定总是从当前目录拷贝，但第 2 步命令假设 DOS 是当前目录）。
2. 键入 XCOPY HELP.COM TRIAL。
   这个命令产生屏幕信息：

```
Reading sourse file(s)...
HELP.COM
        1 File(s) copied
```

第一行在 XCOPY 寻找和读 HELP.COM 时显示，第二行在写 HELP.COM 时显示，而最后一行表示工作结束。当处理多个文件时，在显示 Reading source file(s)时 XCOPY 将读取尽可能多的文件到内存，然后写这些文件，同时显示每个文件的名字。XCOPY 将重复这个过程，直到工作结束。总结行只在工作结束时显示。

提示： XCOPY 可用的内存越多，它工作得越快。

15.3.1　指定一个目录或文件

如果指定了一个不存在的目标，XCOPY 不能区分是想创建这个名字的新文件还是新目录。（COPY 没有这个问题，因为 COPY 不能创建目录）。XCOPY 将显示下列信息（其中 TEMPY 是指定的目标）：

```
Does TEMPY specify a file name
or directory name on the target(TEMPY 是目标文件名还是目录名)
(F=file,D=directory)?
```

注释： 即使目标指定的是一个已存在的文件，XCOPY 也会问这个问题。只有在目标指定一个已存在的目录时，才不问这个问题。

如果按 F 键，XCOPY 将创建一个新的 TEMPY 文件或覆盖当前目录中存在的 TEMPY 文件。如果按 D 键，XCOPY 创建一个新的名为 TEMPY 的当前目录的子目录，并把这个文件按原名拷在其上。

练习拷贝 HELP.COM 到一个名为 TRIAL2 的新目录：

1. 确认 DOS 是当前目录。
2. 键入 XCOPY HELP.COM TRIAL2。
   XCOPY 询问 TRIAL2 是一个文件还是一个目录。
3. 按 D 表示目录(不必再按 Enter)。
   XCOPY 创建新目录并把 HELP.COM 拷入其中。
4. 键入 DIR TRIAL 2 列出新目录的内容。
   你将看到 HELP.COM 是其中唯一一个文件。

提示： 可以在目标参数后加一个反斜杠以跳过"file or directory"的询问，如 TRIAL2\。这样 XCOPY 就会知道想要的是一个新目录，它创建这个目录并把需要的文件拷入其中。

如果想拷贝到一个新文件名，如 NEWFILE.DAT，可键入下面的命令：

```
XCOPY HELP.COM TRIAL2\NEWFILE.DAT
```

XCOPY 同样会询问 NEWFILE.DAT 是一个文件还是一个目录(即使\TRIAL2\NEWFILE.DAT 已经作为一个文件存在)，它根据你的回答进行处理。试着拷贝 HELP.COM 到 TRIAL2，但这次命名为 NEWFILE.DAT：

1. 键入 XCOPY HELP.COM TRAIL2\NEWFILE.DAT。
   XCOPY 询问 NEW FILE.DAT 是文件还是目录。
2. 按 F 表示是文件。
   XCOPY 拷贝这个文件到 TRAIL2，并命名为 NEWFILE.DAT。
3. 键入 DIR TRIAL2 列出目录内容。
   你将看到 HELP.COM 和 NEWFILE.DAT。

注释： 与 COPY 相同，当 XCOPY 拷贝一个文件时，原始文件不改变，拷贝接收到与原始文件相同的日期和时间，拷贝的归档属性打开，隐含、系统和只读属性关闭。XCOPY 不能拷贝隐含或系统文件。

答疑： 我想用 XCOPY 创建一个新目录，但得到消息 Unable to create directory(无法创建目录)。这是怎么回事？

这是因为要创建的新目录名已在父目录中存在，或者是一个文件或者是一个目录。在 DOS 中，一个目录不能包含有相同名字的文件和子目录。可通过删除、重命名一个文件或使用一个不同的目录名来解决这个问题。

15.3.2 使用 XCOPY 拷贝多个文件

对于多个文件应尽量用 XCOPY 代替 COPY，因为 XCOPY 快得多。这两个命令的命令

格式是相同的。假设你想拷贝 DOS 中所有的 D*.HLP 文件到一个名为 TRIAL3 的新目录，这个目录用 XCOPY 创建：

1. 确认 DOS 是当前目录。
2. 键入 XCOPY D*.HLP TRIAL3\。(最后的反斜杠表示避免"file or directory"询问)。
   XCOPY 创建新目录并将要拷的文件拷入其中。
3. 键入 DIR TRIAL3 列出新目录的内容。
   它将包含几个 D*.HLP 文件。

一个源和目标中都带有通配符的 XCOPY 命令与 COPY 功效相同。在下面的例子中，XCOPY 知道目标是一个文件名，因此它不问"file or directory"问题：

```
XCOPY *.BAT *.SAV
```

如果你试图拷贝多个文件到单个目标，你可能会认为 XCOPY 会把目标视作一个目录，但它不会。假设键入命令：

```
XCOPY *.BAT \NEW
```

XCOPY 将询问熟悉的"file or directory"问题。如果指定\NEW 是一个文件，XCOPY 依次拷贝每个.BAT 文件到\NEW，最后\NEW 的内容是所拷贝的源文件中的最后一个。这可能有点出乎意料。

15.3.3　拷贝整个目录分枝

与 COPY 相比，XCOPY 的一个主要优点是它可以在整个目录分枝的基础上工作。当在 XCOPY 命令中包括/S 选项时，XCOPY 在整个目录分枝上搜寻与源文件名匹配的文件，并将其拷贝到目标分枝，同时创建需要的子目录以存放拷贝。

例如，假设使用/S 选项拷贝 C:*.SAV 到驱动器 A 中的软盘，同时假设驱动器 C 在下列目录中包含.SAV 文件：

```
\(根目录)
\BUDGET
\BUDGET\BACKUP
\BANDS\LETTERS
```

XCOPY 从 C 的根目录拷贝.SAV 文件到 A 的根目录。它创建 A:\BUDGET(除非它已存在)并从 C:\BUDGET 中拷贝.SAV 文件到 A:\BUDGET。同样，它创建 A:\BUDGET\BACKUP 并从 C:\BUDGET\BACKUP 中拷贝需要的文件到其中。为了创建 A:\BANDS\LETTERS，XCOPY 必须首先创建 A:\BANDS，除非这些目录都已存在于 A 中。然后，XCOPY 从 C:\BANDS\LETTERS 拷贝.SAV 文件到 A:\BANDS\LETTERS。

注释： 如果命令用选项/E 代替/S，XCOPY 将复制源目录分枝的整个目录结构，即使一些目录没有接收到拷贝文件。在前一个例子中，根目录为源目录分枝的头部，驱动器 C 的整个目录结构将在 A 中复制，但只有.SAV 文件被拷贝。在 6.2 以前的 DOS 版本中，不能单独使用/E，除非首先设定了/S。

网络注意： 你是否可在一个网络驱动器上创建一个新的目录，这取决于你的访问权限。如果没有权限，将不能使用 XCOPY 复制这个驱动器上的分枝。尽管如此，如果目录已存在，就可使用 XCOPY 拷贝文件到其上。

假设你想将 1992 年的所有财政记录放在一个软盘上以便永久保存，然后从硬盘删掉它们，那么用下面两个命令就可完成。在这里假设整个 1992 目录分枝放在一个磁盘上，而且不涉及隐含或系统文件或子目录：

```
XCOPY C:\FINANCE\Y1992\*.* A:\ /S
DELTREE C:\FINANCE\Y1992
```

试着从以 DOS 目录开头的练习分枝中拷贝所有.XXX 文件到驱动器 A 中的磁盘：

1. 插一个已格式化但没有用过的软盘到驱动器 A 中(如果你没有这样一个软盘，可以跳过这些练习)。
2. 确认 C:\DOS 是当前目录。
3. 键入 XCOPY *.XXX A:\ /E。
   当 XCOPY 创建目录和拷贝文件时，XCOPY 列出带路径名的所有这些文件。
4. 键入 TREE A:\ /F，看 A 的结果。
   TREE 显示驱动器 A 的目录结构，包括每个目录中的文件(由/F 选项指定)。你可看到根目录和 TRIAL 目录包含一些.XXX 文件，但 TRIAL2 和 TRIAL3 目录为空。

预览： XCOPY 提供了某些有用的增强功能。如可基于归档属性、日期和时间来选择拷贝的文件。

如果想拷贝大量的文件到一个软盘，而且 COPY 或 XCOPY 不能将它们安置在这个磁盘上，那么可以用 MSBACKUP 把它们放到两个或多个磁盘上。第 23 章“MSBACKUP 和 MWBACKUP”将解释 MSBACKUP。

而 REPLACE 命令可以通过比较源和目标目录来选择拷贝的文件，这将在第 40 章中描述。

15.4 移动文件

在 DOS 6 之前，在命令提示符下进行文件移动的步骤是：拷贝文件，确认拷贝已成功完成，最后删除源文件。这个过程不像听起来这么简单，在删除源文件之前，判断所有需要的文件都已被成功地拷贝到目标上是不容易的。

DOS 6.0 介绍了一个新命令 MOVE，它可一步移动文件(它与第 14 章“目录管理”中讲述的 MOVE 命令相同)。DOS 6.2 改进了 MOVE 命令，使之具有与 COPY 和 XCOPY 相同的覆盖保护机制。图 15.3 对进行文件移动的 MOVE 命令进行了总结。

MOVE

移动一个或多个文件或为一个文件更名。

句法：

MOVE [/[－]Y] filespec [...]destination

参数和选项：

filespec	指定所需移动或更名的文件。
destination	指定所需移动文件到达的文件或目录或文件新名。
/Y	不经确认就覆盖目标文件。
/－Y	覆盖目标文件时需要确认。

图 15.3　MOVE 命令总结(见 move 文件)

试着逐一地移动一些文件。

1. 如果不在 DOS 目录中，请现在转到其中。
2. 键入 MOVE PRACTICE. XXX TRIAL2。
   DOS 移动文件 PRACTICE. XXX 到 TRIAL2 目录。
3. 键入 MOVE FILE1. XXX TRIAL。
   因为 TRIAL 中已有一个名为 FILE1. XXX 的文件，DOS 询问是否想覆盖它。
4. 键入 Y 覆盖这个文件。

答疑： 键入一个 MOVE 命令，却得到消息 Too many parameters(参数过多)。这是因为误把/Y 参数放在了命令行的最后。MOVE 把最后一个参数作为一个目标，因此必须把/Y 放在命令行的较前面。

答疑： 我试图移动一个文件，却得到消息 Unable to create destination(无法创建目标)。我如何才能移动文件？

在目标目录中存在与你想移动的文件名字相同的只读文件。关闭其只读属性以使目标文件可被覆盖，或者改变目标文件或源文件的名字。

15.4.1　移动多个文件

可在源文件中使用通配符以移动多个文件，但目标必须是一个目录。在这时你不能指定一个目标文件名，即使使用通配符也不可以。

试着移动一组文件：

1. 转到 TRIAL 目录。
2. 键入 MOVE *. XXX TRIAL2，将所有的. XXX 文件移动到 TRIAL2 目录。

答疑： 我试图移动一组文件到另一个目录，得到消息 Cannot move multiple files to a single file(无法将多个文件移动到单一文件中)，出什么问题了？我如何才能移动我的文件呢？

DOS 不认识你指定的目标目录名，如果这个目录已存在，你可能拼错了名字或路

径表达错误。改正命令再试。

如果目录不存在，你试图用这个命令创建它时，MOVE 会问 Make directory "name"？[yn]（创建目录"名称"？），要创建新目录就应键入 Y。

Move 命令还可使用一系列源文件名，后面跟一个目标。下面的例子是从当前目录移动文件 MY *. *，YOURFIL. TXT 和\OURFIL. DAT 到\NEW 目录。

```
MOVE MY *. * YOURFIL. TXT \ OURFILE. DAT \NEW
```

提示： 如果省略了 COPY 或 XCOPY 的目标，当前目录被设定为目标目录。MOVE 需要一个目标参数，但可使用一点(.)指定当前目录为目标目录。

MOVE 总是认定命令行的最后一个参数为目标参数，而目标参数之前每个文件名为源文件。当多于一个文件被移动时，目标必须是一个目录。如果目录不存在，MOVE 问 Make directory "name"？[yn]。如果回答 Y，MOVE 创建目录并把文件移入其中。

提示： 可在 MOVE 命令中包含/Y 选项来跳过"Make directory"询问。

试着移动一些文件到 MOVE 创建的目录：

1. 转到 TRIAL2 目录。
2. 键入 MOVE *. * \DOS\TRIAL4。
   MOVE 询问 Make directory "c:\dos\trial4"？[yn]。
3. 按 Y。
   MOVE 创建目录并移动文件。在移动时可看到文件的清单。

答疑： 我试图移动一个目录中的所有文件，却得到消息 Unable to open source（无法打开源文件）。

MOVE 在源目录中遇到一个子目录时显示这个消息。这个消息只是想告诉你子目录是不可移动的。你可放心地忽略这个消息。

15.4.2 使用 MOVE 进行文件更名

如果移动一个文件到它自身所在的目录，效果如同对它进行更名。在 DOS 6.2 以前的版本中，MOVE 不经警告就用想要的名字取代一个已存在的文件，而且文件不能使用 DOS 6 的 UNDELETE 工具（即使用 Delete Sentry）来取消删除。这使 DOS 6.2 以前的 MOVE 与 REN 相比更为危险，因为当新名已在目录中使用时，REN 会拒绝更名。下章将讲述 Ren 命令。

15.5 清除目录

在本章中创建了许多目录。键入下列命令将删除这些目录及其所包含的文件：

```
DELTREE \DOS\TRIAL *
```

15.6　自 测 练 习

至此已学习了如何用 COPY、XCOPY 和 MOVE 来拷贝和移动文件，下面的练习将帮助你熟悉学过的技术。

1. 拷贝硬盘上所有的 COMMAND.COM 文件到驱动器 A 中对应的目录中。
2. 列出驱动器 A 中的一个目录，看它的新结构和文件。
3. 从 A:\移动所有的文件到 A:\dos。
4. 删除驱动器 A 上的结构。

第 16 章 增加的文件管理

简介： DOS 包含了许多从命令提示符和批处理程序内部进行文件管理的命令和工具。

16.1 总 览

大部分应用程序只提供了有限的文件管理功能，它们通常不能进行以下操作，如设置属性，比较一个文件的两个版本，甚至删除文件。DOS 为所有这些任务提供了命令。而且，只为了浏览一个文件而花上一段时间等待应用程序载入该文件是一件麻烦的事，在这方面，DOS 提供了显示和打印一个文本文件的功能。

在本章中，你将学会如何：

- 为文件更名
- 浏览并改变文件及目录的属性
- 删除文件
- 在显示器上显示文本文件
- 打印文本文件
- 比较两个文件并显示其不同之处

网络注意： 在本地驱动器上对文件进行操作不会有什么麻烦，但是否可以改变或删除网络驱动器上的文件取决于访问权限。

注释： 因为不能在只读驱动器上进行写操作，所以在 CD-ROM 驱动器上不能改变文件的属性，也无法更名、删除文件，但可浏览文件属性，显示和打印文件，以及比较文件。

16.2 创建一些练习文件

在开始学习文件管理之前，先创建一些练习文件：

1. 转到根目录。
2. 创建一个名为\PRACTICE 的新目录，拷贝所有的\DOS*.HLP 文件到其中(注意：这可通过使用命令 XCOPY 来实现)。

16.3 使用 REN 进行文件更名

当你想在命令提示符下对一个或多个文件进行更名时，可使用 REN 命令(如图 16.1)。

REN

REN filespec newname

参数：

filespec　指明要更名的文件。

newname　指明更名文件的新名字。

图 16.1　REN 命令总结

REN 与 Shell 的更名功能相比有一点小小的优点。Shell 不能完成一组文件的更名，但 REN 却可以这样做。下面试着将 PRACTICE 中的所有.HLP 文件更名为.XXX 文件：

1. 键入 CD \PRACTICE，以转到练习目录。
   确保在 PRACTICE 中完成这个练习是很重要的，这可避免在真正有用的目录中更名所有的.HLP 文件。
2. 键入 DIR 命令以确认在 PRACTICE 目录中只有.HLP 文件。
   如果发现在其他目录，回到第 1 步。
3. 键入 REN *.HLP *.XXX，更名所有的.HLP 文件。
   如果没有得到错误消息，则文件已更名。
4. 键入 DIR 看看更名后的文件。
   你将看到所有文件都以.XXX 为扩展名。

答疑： 我试图更名一个文件，得到消息 Duplicate filename or file not found（文件名重复或未找到文件）。这是什么意思？

不幸的是，REN 将两个消息合并为一个，让你自己找出发生的问题属于哪一个：或者你键入的文件名没有指定一个存在的文件，或者你要的新名已存在于目录中。

首先检查原始文件名，看看是否有拼写错误或者忘掉了包含一个路径名。试用 DIR filespec 保证文件已存在。如果发现名字错误，更正命令并重试。

如果确信源文件名正确，则肯定是存在一个与你试图分配的名字相同的文件或子目录（可在 DIR 命令中使用/A 选项列出一个目录中所有的文件和子目录，包括隐含和系统文件或目录）。重新键入命令，但使用一个不同的新名。

如果确信源文件名正确，新名与一个已存在的文件名不重复，那么文件必定是一个隐含或系统文件。使用 ATTRIB filespec 找出这个文件的属性，然后使用 ATTRIB filespec 的－H 选项或－S 选项关闭隐含和系统属性。在完成这个文件的更名后，使用 ATTRIB filespec 的＋H 选项或＋S 选项恢复隐含或系统属性。

作者忠告： 请确保文件组更名的简单易行。

因为 REN 无确认选项，如果想用这个命令做一些不同寻常的事，可能会得到预料之外的结果。例如，假设你想将 CH01.DOC 到 CH15.DOC 文件更名为 OLD01.DOC 到 OLD15.DOC。如果键入命令 REN CH*.DOC OLD*.DOC，那么文件 CH01.DOC 到 CH10.DOC 将被更名为 OLD1.DOC 到 OLD10.DOC，而对 CH11.DOC 到 CH15.DOC 进行操作时会产生出错消息，因为 OLD1.DOC 到 OLD5.DOC

已经存在。

16.4 用 ATTRIB 命令管理文件属性

文件属性影响本章中讲述的大部分命令。例如，不能对一个有隐含或系统属性的文件进行删除和更名操作。ATTRIB 命令用于检查和修改属性(图 16.2)，它可对文件和目录进行操作。

当键入一个特定的或全范围的文件名而不带有任何选项，ATTRIB 将显示这些文件的属性。试着显示 PRACTICE 目录中文件的属性：

1. 如果不在\PRACTICE 目录，现在切换到它，以便不必在文件名中包含路径名。
2. 键入 ATTRIB HELP. XXX，以观察指定文件的属性。
   ATTRIB 显示信息如下：

   A C:\PRACTICE\HELP. XXX

   其中左边显示的是属性，A 表示这个文件具有归档属性，而 S、H 和 R 则分别表示文件具有系统、隐含和只读属性。少一个字母就表示没有这种属性，如同上面的例子那样。
3. 键入 ATTRIB M *. * 看看当前目录中所有以 M 开头的文件的属性(注意：你可调回上一个命令，然后编辑此命令)。
   ATTRIB 显示指定文件的属性。
4. 键入不带参数的 ATTRIB 命令，以浏览当前目录中所有文件的属性。
   ATTRIB 显示指定文件的属性。

ATTRIB

显示或改变文件和目录的属性。

句法：

ATTRIB [+R|-R][+A|-A][+H|-H][+S|-S][filespec [/S]]

参数和选项：

none	显示当前目录中所有文件的属性。
+R|-R	打开或关闭只读属性。
+A|-A	打开或关闭归档属性。
+H|-H	打开或关闭系统隐含属性。
filespec	指定处理的文件。
/S	处理当前目录或指定目录下的分枝中所有的文件。

图 16.2 ATTRIB 命令总结

为改变文件属性，可在命令中包含属性选项。一个选项前加一个+号表示打开这个属性；加一个-号表示关闭这个属性。假设你想关闭 HELP. XXX 归档属性并打开它的隐含属性：

1. 键入 ATTRIB HELP. XXX －A +H。
   不幸的是，ATTRIB 不对属性改变进行确认。
2. 键入 DIR HELP. XXX，看看文件是否真的被隐含。
   你将得到消息 File not found。
3. 键入 ATTRIB HELP. XXX 看看这个文件的属性(本章最后一次提醒您：如果已安装了 DOSKEY，可以调回前面的 ATTRIB 命令并进行编辑)。
   ATTRIB 将显示这个隐含文件的属性。显示的消息如下：

   H　　C:\PRACTICE \HELP. XXX

4. 键入 ATTRIB MW *. XXX +S 使目录中的一些文件变成系统文件。
5. 键入 ATTRIB DOS *. XXX +R 使目录中的一些文件变成只读文件。
6. 键入 ATTRIB(不带参数)列出目录中所有文件的属性。

16. 4. 1　改变隐含文件和系统文件的属性

改变隐含文件或系统文件的属性可能要多花一点工夫。只有改变了文件的隐含/系统属性，ATTRIB 才能处理文件的其他属性。这个过程较为复杂，首先要关闭隐含/系统属性，然后改变你想修改的属性，最后再打开隐含/系统属性。但有一个更容易的方法，即采用相同的命令在改变任意其他属性时确认隐含属性或系统属性。

试着使 HELP. XXX 成为一个只读文件：

1. 键入 ATTRIB HELP. XXX +R。
   因为 HELP. XXX 是一个隐含文件，你将得到出错消息：

   Not resetting hidden file(未重新设置隐含文件) C:\PRACTICE\HELP. XXX

2. 在命令中加入+H 选项，如 ATTRIB HELP. XXX +R +H。
   这次，选项不会出现错误消息。
3. 键入 ATTRIB HELP. XXX 看看文件的属性。
   你将看到它既是隐含的又是只读的。

如果想在设置另一个属性的同时关闭隐含/系统属性，也可用－H 或－S 代替+H 或+S，只要在命令中处理隐含/系统属性——确认或者关闭——就可以设置其他属性。

当一个文件的隐含属性和系统属性都打开时，ATTRIB 报告隐含属性但不报告系统属性。因此，如果得到消息：Not resetting hidden file filespec，文件也可能是一个系统文件(许多隐含文件也是系统文件)。如果两个属性都打开，你必须在命令中一并处理。

假设 SECRET. TXT 的四个属性都打开。为了只关闭其中的只读属性，你将不得不键入 ATTRIB SECRET. TXT +H +S －R。归档属性不需要确认。为关闭系统属性而不关闭隐含属性，可键入 ATTRIB SECRET. TXT +H －S(归档属性和只读属性都不需要确认)。

答疑： 我试图改变一个文件的属性，却得到消息 Write-protect error writing drive A，Abort，Retry，Fail?(在对驱动器 A 进行写操作时，出现写保护错误)我如何处理这个文件?

你不能在一个加了写保护的磁盘上改变或删除一个文件，因为从物理上不能写盘，

这时甚至不能改变文件的名字或属性。

如果真的决定改变或删除这个文件，则请从驱动器中取出磁盘，去掉写保护，再插入驱动器中，然后按 R 重试。

如果并不想改变一个文件或者想先了解该磁盘写保护的原因所在，按 A 取消命令，回到命令提示符。

16.4.2 设置整个目录分枝范围内的文件属性

当使用一个全局文件名时，可以用一个命令显示或改变一些文件的属性。/S 选项将命令的范围扩展到整个指定目录分枝。幸运的是，当试图改变每个隐含/系统文件时，会看到针对每个文件的消息 Not resetting...，你需要分别处理这些文件。

16.4.3 对目录的属性进行操作

目录具有同文件一样的属性。例如，你可以隐含一个目录，使它不在 DIR 或 TREE 清单中显示。只有当你指定目录的具体名字时，ATTRIB 才能操作一个目录，而全局文件名不能用于访问目录。

试着隐含 PRACTICE 目录：

1. 切换到父目录(不能改变所在目录的属性)。
2. 键入 TREE 以显示完整的目录分枝。
   可看到 PRACTICE 在目录树中。
3. 键入 ATTRIB PRACTICE +H 以隐含目录。
4. 用 TREE 显示目录分枝。
   这次 PRACTICE 将不被显示。

当显示父目录的内容时，DIR 不会列出隐含目录。但如果指定这个目录的名字，则可以列出该目录，或者切换到它等等。这样，如果你知道目录名，就可以访问该目录，否则就无法对其实施访问。下面一起来试一试：

1. 键入 DIR。
   PRACTICE 将不出现在目录清单中。
2. 键入 DIR PRACTICE。
   DOS 列出 DIR 目录的内容，而不管它是否隐含。
3. 键入 CD PRACTICE，转到 PRACTICE 目录。
   虽然它是隐含的，但进入它不会遇到任何麻烦。

警告： 不要认为隐含目录是一个安全方法，因为大部分程序(包括 DOS Shell)忽略目录的属性，而且有一定 DOS 知识的人都知道如何通过使用/A 选项来找到目录中包含的隐含文件或隐含目录。

16.5 文件删除

在命令提示符下进行文件删除不是最好的方法，因为一些应用程序如 Windows 和 DOS

Shell 可通过确认删除来保护文件，而且对于多个文件，还可以选择特定的文件进行删除，而不必使用全局文件名，因为后者可能删掉你不想删的文件。不过，如果真想在命令提示符下删除文件，DEL 命令有一个安全选项，称为/P，你应该习惯使用它(如图 16.3)。一旦加上/P，DEL 在删除文件之前将要求进行确认。

DEL

句法：

DEL filespec [/P]

参数和选项：

filespec　指定删除的文件。

/P　　每次删除前要求进行确认。

图 16.3　DEL 命令总结

无论何时使用 DEL，都应记住它的局限性：

- 如果不使用/P 选项，DEL 不加警告或确认就删掉文件，这时它甚至不显示删除的文件的名字。
- 用一个全局文件名很容易删除你想删的文件，而另一方面，它也可能不删除某些你想删的文件，这在下两条解释。
- DEL 不会删除隐含或系统文件。当使用一个指定文件名时，如果它是隐含或系统文件，那么你会看到消息 File not found。但如果用一个全局文件名，即使使用/P，DEL 也不会显示关于它所忽略的文件的消息。
- DEL 也不会删除一个只读文件。当使用一个具体的文件名时，可看到 Access denied 消息；如果使用一个带有/P 的全局文件名，也是如此。但如果使用全局文件名且不带/P，当只读文件被略过时你不会接收到任何警告。

请牢记这些限制，现在假设想从 PRACTICE 删除文件 EDIT.XXX：

1. 键入 DEL EDIT.XXX /P。
   DOS 显示消息：

   C:\PRACTICE\EDIT.XXX,　　Delete (Y/N)?

2. 保证路径名和文件名都正确，如果不正确，键入 N 后再试。
3. 当路径名和文件名都正确后，键入 Y，删除这个文件。
   如果没有得到出错消息，文件被删除。

作者忠告： 使用 DEL 命令时要使用/P。

即使删除一个文件，/P 也可防止因将文件名拼写错误而意外地删除其他文件。当使用全局文件名删除一组文件时，使用/P 更加重要，因为这样可逐一确认每个删除的文件。

但如果你习惯于按 Y 键而不检查显示的文件，那么/P 不带来任何好处。因此，请

养成在按 Y 之前仔细阅读每个文件名的习惯。

16.5.1 删除多个文件

可用一个全局文件名删除一组文件：

1. 键入 DEL MS*.XXX /P 删除 PRACTICE 目录中所有以 MS 开头的.XXX 文件。
   DOS 要求你确认与文件名匹配的第一个文件。
2. 仔细检查文件名。如发生错误，按 Ctrl+Break 取消整个命令，然后从头开始。

提示： Ctrl+Break 可在中途取消许多程序的运行。它通常适用于一些旧程序，如 DEL、REV 和 TYPE 等，因为这些程序只在命令提示符下显示消息而不会打开任何类型的 GUI，但对于 DOSSHELL 或 MSBACKUP 等不起作用。
如果一个程序（如 DEL 或 TYPE）正访问一个错误目录或文件，按 Ctrl+Break 将其中止，然后回到命令提示符。

3. 如果路径名和文件名正确，按 Y 删除这个文件。
   DOS 要求你确认下一个与文件名匹配的文件。
4. 重复 3，直到所有文件被删除。

16.5.2 删除受保护的文件

消息 Access denied 意味着文件是只读的，不能被删除，除非首先改变它的属性。此外你也可能得到消息 File not found，这个消息意味着当前文件可能是隐含或系统文件，或者文件在当前（或指定）目录中不存在。

试着删除一些被属性保护的文件：

1. 试着删除 DOSHELP.XXX。
   DOSHELP.XXX 是一个只读文件，因此你会得到消息 Access denied。
2. 试着删除 HELP.XXX。
   HELP.XXX 是一个隐藏文件，因此将显示 File not found。

注释： 如果你改变了主意，DOS 6 包含一个 UNDELETE 命令可以恢复被删除的文件。第 26 章“UNDELETE 的使用”将描述如何使用 UNDELETE。

16.5.3 删除目录中的所有文件

为删除一个目录中的所有文件，不必包含文件名。你只需输入路径名，随后 DELETE 就会认为你指的是这个目录中的所有文件。因此，若要删除驱动器 A 的根目录中的所有文件，则只需输入 DEL A:\；而若要删除当前目录中的所有文件，则可输入 DEL.或者 DEL *.*，这两个命令功能相同。

当你试图删空一个目录时，DOS 显示下面信息让你有机会改变主意：

```
All the files in the directory will be deleted(目录中的所有文件将被删除)!
Are you sure(你确定吗) (Y/N)?
```

必须键入 Y 以继续。

试删除当前目录中的所有文件：

1. 确认\PRACTICE 是当前目录。
2. 键入 DEL. 以删除所有文件。
   DOS 询问是否确实要删。
3. 键入 N 取消命令。

预览： DEL 一次只可以处理一个目录，但 DELTREE 可删除整个分枝中的文件，且不考虑属性。第 39 章“目录管理”将告诉你怎么做。

16.6　用 TYPE 显示文件

TYPE 命令可在命令提示符屏幕中显示一个文件(如图 16.4)。TYPE 在浏览较短的 ASCII 文本文件(如 CONFIG. SYS)时是很方便的，但在其他情况下，其显示结果就未必令人满意，因为 TYPE 只是一鼓脑儿将文本显示在屏幕上，没有格式，也不会在文件中翻卷。

TYPE
显示一个文件的内容。
句法：
TYPE filespec
参数：
filespec　　指定想浏览的文件。

图 16.4　TYPE 命令总结

如果一个文件太长，屏幕装不下，那么文件的开头部分在你想看之前就消失在屏幕上方了。你可以用 Pause 键控制，但系统越快，这样做也就越困难。因为 TYPE 不能往回显示，一旦文本消失在屏幕的上方，不再一次输入命令就不能再看到前面的内容。因此，最好用 Shell (File，View File Contents 命令)或 EDIT(在第 20 章“基本批处理程序”中讲述)看较长的 ASCII 文本文件，这两个程序都有翻卷功能。

答疑： 我试图在命令提示符屏幕上显示一个文件，但 TYPE 开始显示一些怪字符，伴随着不断的嘟嘟声。该怎么办？

如果还没结束，按 Ctrl+Break 停止 TYPE，回到命令提示符。

这表明文件不是一个 ASCII 文本文件——它包含二进制数据。计算机将二进制数据解释成扩展字(如表情符)或控制符号(如响铃)。

你不能用 TYPE 显示这样的文件，而必须使用创建这个文件的应用程序进行浏览。

包含非 ASCII 代码的文件用 TYPE 不能很好显示。非 ASCII 代码可以出现为图形字符、响铃警告，或者使命令在整个文件结束显示前终止。浏览一个非 ASCII 文件的最好方法是使用最初创建它的应用程序。

16.7 用 PRINT 打印文件

DOS 有几种打印 ASCII 文本文件的方法。可重新定向 TYPE 的输出列到打印机端口如下：

```
TYPE AUTOEXEC.BAT >PRN
```

这对短文件效果很好，但因为 TYPE 没有任何格式处理，一个较长的文件会超过页的底端。COPY 一个文件到打印机端口有相同的问题。对于较长的 ASCII 文本文件，PRINT 命令是最好的选择，因为它会将文本分页。

网络注意： 如果你的打印机在一个网络上与其他机器共享，PRINT 命令不一定有效，你的网络管理员会告诉你用于在 DOS 提示符下打印文件的命令。如果不能使用 PRINT 命令，那么你可以跳过本节或用 EDIT 打印。

PRINT 命令必须在第一次使用它之前进行初始化，这个过程称为“如何初始化 Print 程序”，第 9 章“Shell 中的其他文件管理”已经对其有过叙述。图 16.5 显示了初始化后如何使用 PRINT 打印一个或多个文件。

PRINT

句法：

PRINT [/T][filespec]

参数和选项：

none	显示打印队列的内容或加载这个 TSR。
/T	从打印队列中删除所有文件。
filespec	指定即将放到打印队列中的文本文件。

图 16.5 PRINT 命令总结

不能打印 PRACTICE 目录中的.XXX 文件，因为它们都不是 ASCII 文本文件，但可以打印 DOS 目录的 README.TXT 文件。

1. 确认打印机已准备好(第 9 章“通过 Shell 打印”一节解释了需要检查的所有项目)。
2. 键入 PRINT \DOS\README.TXT。
   如果得到消息 C:\DOS\README.TXT is currently being printed，转到第 4 步。
   如果显示如下信息，继续第 3 步：

   ```
   Name of list device [PRN]:
   ```

3. 必须告诉 PRINT 打印机在哪儿。如果打印机连在第一个并口，按 Enter 键。如果连在其他接口，键入接口的名字，如 LPT2 或 COM1。如果不能确定，则按 Enter 键。
   将得到消息：

   ```
   Resident part of PRINT installed
   ```

C:\DOS\README.TXT is currently being printed

4. 如果不在一个网络上，文件将开始打印。如果不打印，参看第 12 章“在命令提示符下工作”中的关于“重定向命令输出”问题的解决方法。

如果你的打印机对于计算机来说是本地的——也就是说，如果它不是网络上的一个共享资源，那么 PRINT 将保存一个最多为 10 个文件的准备打印队列。你既可添加文件到队列，也可从中删除文件。

网络注意： 如果打印机是网络上的一个共享资源，则由网络软件维护打印队列，它可能保存多于 10 个的文件。你不能自己删空队列。网络管理员能告诉你如何从队列中移去一个文件或移动一个高优先级的任务到队列中。

1. 键入 PRINT *.TXT 增加多个文件到队列中。
   你可看到一系列文件被加到队列中。
2. 键入 PRINT(不带参数)以观察队列的状态。
   状态报告哪个文件正在打印，哪些文件正在等待。
3. 如果你的打印机不是一个网络共享资源，键入 PRINT /T 从队列中移去所有文件。
   屏幕显示：Print queue is empty(打印队列为空)。

注释： 清除打印队列后，打印机可能继续从它的内存中调出内容进行打印，因而你应该关掉打印机以清除它的内存。

提示： 可用一个命令清除打印队列并增加新文件到队列中，但必须在文件描述名之前使用/T，因为参数从左到右处理。下面的命令会清除队列并增加新的文件到队列中：

```
PRINT /T TRY*.TXT SAVE*.TXT
```

提示： 通过使用 PRINT 可以配置 PRINT TSR 以控制一些因素，如打印队列大小及分配给打印工作的时间等。此外也可不清除整个队列而只从中移去个别文件。

16.8　用 FC 比较文件

假设你发现两个不同目录的两个文件有相同的名字和大小，但日期/时间不同。你想保存两个吗？如果不，你想删除哪一个文件？FC(file comparison：文件比较)命令可通过比较两个文件和显示不同之处来帮你决定。

FC 可做两种基本类型的比较：二进制和 ASCII。在二进制模式中，FC 一个字节一个字节地比较两个文件，显示一些不匹配的字节。在 ASCII 模式，FC 一行一行地比较两个文件，显示不匹配的行。二进制比较比 ASCII 比较简单多了，但当你想找出两个文本文件如何不同时，这两种比较模式同样无能为力。

16.8.1　二进制比较

图 16.6 显示了 FC 进行二进制比较的格式。FC 对于明显的程序文件自动使用二进制

模式比较文件。也就是说，对于.COM、.EXE、.BIN、.SYS、.OBJ 和.LIB 文件，FC 都当作程序文件。其他文件在缺省状态下都使用文本模式，如果想用二进制模式必须使用/B 选项。

FC

比较两个文件，显示不匹配的行或字节。

句法：

FC [/B] filespec1 filespec2

参数和选项：

filepec1 指定用于比较的第一个文件。

filepec2 指定用于比较的第二个文件。

/B 按二进制模式一个字节一个字节地比较文件，在一次不匹配后不试图重新同步。这是带有扩展名.EXE、.COM、.SYS、.OBJ、.LIB 或.BIN 的文件的缺省模式。

图 16.6 FC 命令总结(二进制比较)

试比较 PRACTICE 目录中的 HELP.XXX 和其他一些文件：

1. 转到 PRACTICE 目录。
2. 清除 HELP.XXX 的隐含属性，以使 FC 可找到它。
3. 键入 FC HELP.XXX \DOS\HELP.HLP /B 比较两个匹配的文件。
   你将看到消息：

   Comparing files HELP.XXX HELP.HLP(比较 HELP.XXX 和 HELP.HLP)
   FC:no difference encountered (未遇到差异)

4. 键入 FC HELP.XXX MEMMAKER.XXX /B 比较两个不匹配的文件。
   整个数据流在屏幕上翻卷。
5. 按 Ctrl+Break 中断比较。

因为两个文件完全不同，FC 发现和报告每个不同的字节，这就是为什么 FC 发送那么多消息到屏幕上的原因。FC 消息如下：

```
00000002:15  14
00000003:60  00
```

意思是指第一个文件的第二个字节是 15，而第二个文件是 14；第一个文件的第 3 个字节是 60，而第二个文件是 00。地址和字节值都以十六进制报告，它与我们习惯的十进制不同。清单如此继续可能有几千条。

很明显，这是只有程序员才明白的信息。对普通用户来说，仅能得知这两个文件不同，并且是大不相同。

假设刚拷贝了一组文件，并且想验证一下以确保拷贝正确，那么一个二进制模式的 FC 比较是最好的选择，你将很快找出文件是否相同。

答疑： 我试图比较两个文件，但 FC 开始显示一些古怪的字符到屏幕上，同时伴随着许多铃响声，我破坏了计算机吗？

你的计算机工作正常。这是因为至少你的一个文件中包含二进制数据，而你进行的是 ASCII 比较。当 FC 试图作为 ASCII 文本显示二进制数据时，计算机将它们翻译为扩展字符（如表情符）或控制信号（如响铃）等。试着增加/B 选项改成二进制比较，问题将得到解决。

16.8.2　ASCII 比较

一个 ASCII 比较告诉你两个文件中哪些行不同。图 16.7 显示了用作 ASCII 比较的 FC 命令的格式。FC 对于没有. EXE、. COM. SYS、. OBJ、. LIB 或. BIN 这样的程序扩展名的所有其他文件自动作 ASCII 比较。

FC

比较两个文件，显示不匹配的行。

句法：

FC [/L] [/C] [/N] filespec1 filespec2

参数和选项：

filespec1	指定用于比较的第一个文件。
filespec2	指定用于比较的第二个文件。
/L	按 ASCII 模式比较，一行一行比较，不匹配后不重新同步。
/C	忽略大小写。
/N	显示 ASCII 比较中不匹配行的行号。

图 16.7　FC 命令总结（ASCII 比较）

如果你想对一个带有程序扩展名的文件作 ASCII 比较，必须使用/L 选项。大部分程序文件是二进制文件，它们无所谓行，一个 ASCII 比较对这些文件没什么作用。但 CONFIG. SYS 是一个带有程序扩展名的 ASCII 文本文件，因此使用/L 选项对这个文件就有意义了。

当 FC 在 ASCII 比较中发现一个不匹配时，它为每个文件显示一个行集。每个行集从最后的匹配行开始，显示所有不匹配的行，然后显示再次匹配的第一行（除非这时到了文件结尾）。这里有一个比较两个 AUTOEXEC. BAT 文件的例子：

```
*****AUTOEXEC.BAT
PATH C:\DOS;C:\WINWORD;C:\NDW;C:\WINDOWS
mouse\mouse bon

*****AUTOEXEC.NDW
PATH C:\DOS;C:\WINWORD;C:\NDW;C:\WINDOWS
cls
mouse\mouse bon
*****
```

其中第一个行集（用 * * * * * AUTOEXEC. BAT 标志）显示 AUTOEXEC. BAT 中的内容；在 * * * * * AUTOEXEC. NDW 之后的行显示的是另一个文件的内容；以 PATH 开头的行是最后匹配的行，该行在两个文件中均可看到；每个集中以 Mouse 开头的行显示

错行后的第一个匹配行。AUTOEXEC. BAT 在这两行中没有其他行，而 AUTOEXEC. NDW 有一个 cls 命令，这个命令就是两个文件的不同之处。换言之，AUTOEXEC. NDW 比另一个文件多一行。

如果两个文件有多个不匹配行，你将看到多个像这样的不匹配报告。每个报告以 * * * * * 开头，后面跟着第一个文件的名字。

16.8.3 重新同步

当 FC 在一个 ASCII 比较中发现一个不匹配的地方时，它会寻找再次匹配的地方，以便使比较继续下去，这称为使文件重新同步。缺省状态下，FC 不认为单个匹配行能构成重新同步的证据，它需要一次有两个匹配行以便继续比较。如看下面这些“文件”：

GARDEN	PRODUCE
Apples	Apples
Bananas	Pears
Peaches	Plums
Tomatoes	Artichokes
Plums	Peppers
Cherries	Figs
Figs	Onions
Onions	Carrots
Carrots	

第一行匹配，然后开始不匹配。浏览 GARDEN 文件，可发现第 5 行与 PRODUCE 文件的第 3 行匹配，但其下一行又不匹配，所以 FC 在这点不再重新开始比较。再往下，GARDEN 第 7 行的 Figs 与 PRODUCE 的第 6 行匹配，下一行 Onions 也匹配。因为两行一致匹配，所以 FC 将从这点继续比较。FC 的消息如下：

```
***** GARDEN
Apples
Bananas
Peaches
Tomatoes
Plums
Cherries
Figs
***** PRODUCE
Apples
Pears
Plums
Artichokes
Peppers
Figs
*****
```

消息显示匹配的最后一行(Apples)，不匹配行和重新匹配的第一行(Figs)。

FC 产生的信息确切地指出两个文件的不同之处，这样就可决定对这些不同之处采取何种措施。在大部分场合，它能帮你决定保留哪个文件，删除哪个文件。但有时它还可确定从一个文件拷到另一个文件的数据。如何使用 FC 报告取决于具体情况。

在许多情况下，你可能需要报告的一个打印副本。这时可使用命令格式 FC parameter >PRN 将 FC 的消息重定向到打印机。

16.8.4 比较选项

FC 注重大小写，故通常 Figs 和 figs 或 FIGS 不匹配。但在比较两个文件，特别是程序文件时，往往不需要关心大小写。例如，如果 AUTOEXEC.BAT 和 AUTOEXEC.SAV 的唯一的区别是 CLS 一个是大写，一个是小写，则这两个文件会产生同样的效果。当不关心大小写区别时，可使用/C 选项让 FC 忽略大小写。

在长文件中，只看不匹配行可能还不够。这时可用/N 选项增加显示的行号，它可帮助你确定文件中所需编辑行。

预览： FC 包含几个参数，它们用于控制 ASCII 比较的方式以及两个文件如何重新同步。例如，你可以控制 FC 搜寻重新同步的距离以及多少行匹配后比较才能继续。第 46 章“数据操作的其他技巧”解释了 FC 命令的增强特点。

16.9 自 测 练 习

至此已学习了如何完成各种文件管理任务，接下来进行自测练习：

1. 在 PRACTICE 目录中将所有的.XXX 文件更名为.OMO(提示：某些文件在更名前必须先改变其属性)。
2. 删除 PRACTICE 目录中的所有文件，并删除这个目录。
3. 迅速浏览 DOS 目录中的每个.TXT 文件，并打印与你的系统有关的文件。
4. 如果你有一个 OLD_DOS.1 目录，阅读其中的 README.NDW 文件。
5. 如果你有一个 OLD_DOS.1 目录，比较一些与你的当前 DOS 目录中有相同文件名的文件，它们可能是不太相同的文件。
6. 你有一些文件既出现在 DOS 目录，又出现在根目录中吗？例如 COMMAND.COM 可能出现在两个地方。比较出现在两个目录中的文件，看看它们是否真的相同。
7. 你的根目录中有几个 CONFIG.XXX 文件和 AUTOEXEC.XXX 文件吗？对它们进行 ASCII 比较以找出其中的不同之处，在这里请忽略空格和大小写。

第 17 章 格式化磁盘和取消格式化

简介： 在使用每张磁盘之前首先要格式化(除非买了预先格式化的盘)。同时，为了方便地纠正错误，DOS 在必要时也能对磁盘取消格式化。

17.1 总 览

DOS 在磁盘上存储数据之前，必须先准备好磁盘上的扇区，一个根目录，FAT 和其他管理信息。必须通过格式化(format)磁盘来准备它们。有时也可重新格式化一个磁盘以删除它的所有旧数据，这样就可重新使用这张盘了。而有时可能意外地格式化了一张存有有价值数据的盘，这时可以取消格式化磁盘来恢复它从前的数据。

本章将描述：

- 格式化软盘和硬盘的步骤
- 如何格式化软盘
- 如何重新格式化软盘
- 如何取消格式化软盘

因为格式会清除一张盘上的数据，在此不学习如何格式化硬盘。如果你正使用 DOS，显然你的硬盘已被格式化，可能不会有必要重新格式化硬盘。

注释： 不能格式化只读驱动器如 CD－ROM 驱动器里的盘或加上写保护的磁盘。

17.2 格式化的工作方式

在试图格式化时理解格式化如何工作是很重要的，因为在继续操作时需要作出一些决定。本节解释初次格式化软盘的过程，以及重新格式化软盘。同时也描述了硬盘格式化，虽然由于明显的原因你将不练习格式化硬盘。

17.2.1 格式化新软盘

新的软盘需要两级格式化。首先，它必须被物理格式化(也称为低级格式化)，它建立磁盘的扇区。然后，它必须被逻辑格式化(也称为高级格式化)，它安装 DOS 的系统信息到磁盘上：启动记录，空的根目录，和初始化的 FAT。(实际上，为了安全起见一个逻辑格式化安装两份复制的 FAT)。它也分配一个系列号帮助 DOS 内部确认磁盘。有两个可供选择的功能：

- 格式化可保存一个内部标志帮你辨认磁盘。大部分人更喜欢用标志如 FEB ORDERS 而不是 DOS 的系列号如 57A4－8C14。

- 格式化可以安装DOS核心程序(IO. SYS、MSDOS. SYS和COMMAND. COM),这样就可用这个软盘启动机器。

不是所有的磁盘都是好的,磁记录表面的缺陷会影响数据存储的可靠性。物理格式化检查磁盘表面,并记录坏点。逻辑格式化在FAT中标明这些坏点,这样DOS就不会在这里存储数据。

DOS的FORMAT程序对普通磁盘可进行低级和高级格式化。DOS 6可格式化从360KB 5 ¼英寸到2.88MB 3 ½英寸的所有软盘。它甚至可以为使用早期DOS版本的系统准备磁盘。例如,DOS 6可以格式化一个360KB盘为320KB盘让DOS 1.0系统使用。而且,在一定程度上,它可在高容量驱动器(如1.44MB驱动器)上准备一个较低容量的磁盘(如一个720KB的磁盘)。

答疑: 当我访问一个文件时,我得到消息Sector not found error(未找到扇区),不能访问数据。如何才能访问数据?

Sector not found error意味着驱动器控制器不再能读物理格式化安装的扇区信息。这常常因为物理格式化一个开始变坏的盘而发生,磁盘保存时间太长就会发生这种情况。即使只有一个扇区坏了,DOS遇到Sector not found error时也会拒绝访问整个文件。

DOS没有对这个问题提供一个好好的解决办法。最好的办法是使用其他软件如Norton Utilites,它包含的程序可用来在维护已有的数据的同时刷新硬盘和软盘上的物理格式化。通过刷新物理格式化几乎总能建立读取文件的能力。

在你恢复了软盘上的数据后,我的建议是拷到一张新盘,并把旧盘扔掉。软盘的价格比因为盘的物理问题丢失数据而造成的损失要便宜得多。

答疑: 当我试图访问一个文件时,得到消息Read error或Write error,DOS不会读或写文件。我如何才能访问数据?

Read error或Write error消息表示DOS在扇区中找到了错误,这可能是因为破坏了物理格式化或扇区中有坏点。这时,DOS拒绝访问整个文件,即使只有一个小小的部分破坏了。

如果你有DOS 6.2或更新的版本,运行磁盘上的SCANDISK寻找一个坏点并尽可能修复它周围的数据,确认选中表面扫描选项以使SCANDISK检查磁盘的表面。

如果仅有DOS 6.0或更早的版本,最好的选择是升级DOS得到SCANDISK或运行其他表面扫描工具如Norton Utilities的Disk Doctor,它可从一个坏扇区修复数据。

如果表面扫描不解决这个问题,则需要刷新磁盘上的物理格式化。DOS中没有一个工具可不破坏磁盘上的数据而做到这一点。这时需要一个其他软件包如Norton Utilities,它包含刷新硬盘和软盘上物理格式化的工具。

如果磁盘是软盘,不要连续使用它,恢复数据后,将所有文件拷到一个新盘上,扔掉旧盘。你不会想在不能恢复的情况下丢失另一个文件吧。

17.2.2 格式化新硬盘

硬盘格式化和软盘稍有不同。它首先做物理格式化，这与软盘格式化相同，但 DOS 的 FORMAT 程序不能做这个工作——因为硬盘种类太多。低级格式化常常在工厂里进行，如果因为某种原因必须重做，你可能不得不买一个特殊程序。（一些系统包含一个物理格式化程序，检查你的系统资料看看是否拥有）。

对于硬盘，第二个格式化步骤是分区，由 DOS 的 FDISK 程序完成，它将硬盘分成几个分区，虽然它们物理上在一张盘上，你可以认为它们是单独的盘。必须对一个硬盘进行分区；但如果愿意的话，可以在硬盘上只建立一个分区，它包含所有的扇区（大部分 DOS 用户这样做）。DOS 6 的最大分区大小是 2GB。FDISK 在硬盘的第一个扇区存储一个分区表（partion table）。分区表显示了哪个扇区属于哪个分区，DOS 在启动时使用这些信息。

预览： 第 54 章“FDISK”解释了如何用 FDISK 对一个硬盘进行分区。

每个 DOS 分区包含一个或多个逻辑驱动器。一个逻辑驱动器工作时就像一个物理驱动器如一个磁盘单元那样。每个逻辑驱动器有自己的启动记录、根目录和 FAT；而且，每个逻辑驱动器分配一个驱动器名，如 C 或 D。

许多经销商常常在将硬盘卖给顾客之前先分好区。这有两个优点：经销商可在硬盘上安装软件作为硬件包的一部分，它避免了计算机用户去做这个工作。不幸的是，如果你不喜欢经销商的分区方式，对于包含数据的硬盘，需花很长时间来改变分区。重新分区一个硬盘会使你丢失所有的数据。

预览： 如果真的需要重新分区，第 54 章描述了如何做这件事。

在硬盘上建立了一个或多个逻辑驱动器后，在其上存储数据之前必须使用 FORMAT 程序进行逻辑格式化，其过程与逻辑格式化一张软盘一样，只是时间长一点。逻辑格式化完成后，硬盘就可正常使用了。

17.2.3 重新格式化软盘

当你不再需要一个软盘上的数据时，可重新格式化磁盘，用作它用。重新格式化将一个磁盘恢复到最初的，没有使用过的状态。

注释： 在一个硬盘已被格式化和包含数据后，你可能不想再格式化，因为这样做会破坏所有的数据。如果硬盘被破坏且需要修复时，必须从头重新格式化，就像它是新的一样。

可在软盘上重做物理和逻辑格式化。当重做逻辑格式化时，会得到一个新的根目录和 FAT，以及一个新的内部标志、一个新的序列号和一个新的启动记录。重做物理格式化时，扇区被重新划分和测试。重做逻辑格式化不会从簇中擦除数据，它所做的全部工作只是标志簇可被新数据使用。但重做物理格式化时，表面测试实际上擦掉了簇中的所有数据，数据不能被重新恢复。

在某些条件下，DOS 的 FORMAT 程序通过保存取消格式化信息到盘上以避免意外擦掉有用数据。取消格式化文件，即镜象文件，向 DOS 的 UNFORMAT 程序提供恢复从前的

数据所需要的信息。

当重新格式化磁盘时，可选择三种格式化级：快速、安全和无条件。每一级有自己的特点，适用于一定的情况。

17.2.3.1　快速格式化

快速格式化只是简单地存储镜象文件和更换根目录与 FAT，不作扇区的可靠性检查。这基本上不花时间，但当磁盘上有些错误时，它不是理想的选择。因此，如果只是想重新使用磁盘存储新的数据，快速格式化常常是最好的选择。

快速格式化将删空磁盘为空，因为它的根目录不包含目录项。但实际上，从前的数据仍在簇中，在存放新数据之前，可用 UNFORMAT 进行恢复。

17.2.3.2　安全格式化

一个安全格式化除了完成快速格式化的操作外，还检查扇区的有效性。虽然这不同于物理格式化的表面测试，它可以找出产生读写错误的簇。有效性测试实际上不写入扇区，所以以前的数据还存在于磁盘上，必要时可取消格式化。

17.2.3.3　无条件格式化

无条件格式化在完成软盘物理格式后再对它进行逻辑格式化。物理格式化破坏了磁盘上的所有数据，因此磁盘不能被取消格式化。DOS 的 FORMAT 不能在硬盘上进行无条件格式化，这是因为 DOS 不能物理格式化一个硬盘，但可在软盘上做无条件格式化。

警告： 不要依靠无条件格式化来保护机密数据，一个拥用适当设备的专家仍能从一个物理重新格式化的盘中读出原来的数据。要删除分类数据，需要一个根据政府标准删除数据的程序。Norton Utilities 包含一个程序可做这样的事。

在两种情况下必须使用无条件格式化：

- 磁盘从未被格式化过，必须做无条件格式化划分扇区。
- 当需要改变磁盘容量时，需要新划分扇区。例如，如果一个磁盘格式化为 720KB 而现在想格式化到 1.44MB，必须做一次无条件格式化。

在其他情形，可能在重新使用一磁盘之前需要做无条件格式化刷新格式化。只要你确信以后不想再取消格式化这张磁盘，刷新旧盘的物理格式化总是一个好主意。（扔掉旧盘是一个更好的主意，磁盘很便宜，但数据不）。

17.2.4　使用 FORMAT 准备新盘

图 17.1 列出了 FORMAT 格式化一个新的，从未格式化过的磁盘的命令总结。

提示： 如果不在 FORMAT 命令中包含/U 选项，FORMAT 试着完成一个安全格式化，若磁盘未经格式化，则转到无条件格式化。只要你确信磁盘是新的和没使用过的，就可通过包含/U 到命令中来节约时间。

试着格式化一个新盘：

1. 取出一个新的，完全没使用过的软盘。如果你没有一个未使用过的盘，用一个包含你不再需要的数据的磁盘。

FORMAT

准备一张备用的磁盘

句法：

FORMAT drive [/V:label] [/S] [/U]

参数和选项：

drive	指定格式化的驱动器。
/V:label	指定格式化盘的卷标。卷标最多可达 11 个字符。如果省略/V，DOS 在格式化过程中提示输入卷标。
/S	使磁盘是可启动的。
/U	指定一无条件格式化。

注意：

FORMAT 还包含几个这里没有显示的增强型选项。它们将在本书后面的命令参考中描述，第 46 章"数据操作的其他技巧"将讲述如何使用这些增强选项。

图 17.1 FORMAT 命令总结

警告： *无条件格式化会破坏磁盘上任何存在的数据。*

2. 键入 FORMAT A：/U 或 FORMAT B：/U，这取决于所使用的驱动器。
   你会看到下列消息：

```
Insert new disk for drive d:
and press ENTER when ready.
```

3. 插入你想格式化的软盘并按 Enter 键。
   你会看到下面的消息：

```
Formatting 1.44M
           0 percent completed
```

第一行告诉你 FORMAT 准备格式化的磁盘的容量，你的驱动器容量可能与此不同。当磁盘以前从未格式化时，FORMAT 自动准备磁盘到驱动器的最大容量(即使不适合你的磁盘)，除非命令明确地指定正确的容量。完成百分比随 FORMAT 在磁盘上规划扇区而增长。完成时，屏幕显示：

```
Format complete.
Volume Label (11 characters, ENTER for none)?
```

4. 这时可为磁盘键入一个卷标。卷标在下一节中解释，现在可按 ENTER 越过卷标继续，可看到消息如下(盘不同，数据可能不同)：

```
1,457,664 bytes total disk space
    1,024 bytes in bad sectors
1,456,630 bytes avilable on disk
      512 bytes in each allocation unit.
    2,845 allocation units available on disk.
```

这些行按字节数和簇数告知存储容量。这个磁盘有一些坏扇区，它们占的字节数显示出来了。整个 bytes available on disk 显示了除了坏扇区以外还剩多少有用空间。

下一行，FORMAT 显示了 Volume Serial Number，它被分配给这个磁盘：

```
Uolume Serial Number is 2A46－17F8
```

最后，FORMAT 问你是否还要格式化另一张盘：

```
Format another (Y/N)?
```

5. 这时如果不想格式化另一张盘，键入 N 结束 FORMAT 程序。

返回命令提示符。

如果你在第 4 步回答了 Y，FORMAT 会自动转到第 3 步，这时就不能改变任何格式化参数如驱动器名或/U 选项符。

答疑： *我试图格式化一个磁盘，得到消息* Invalid media or track 0 bad disk availabe，Format failure。*我如何才能格式这个磁盘？*

最可能的问题是磁盘的 0 磁道被破坏了。因为 DOS 必须在 0 磁道保存启动记录、根目录和 FAT，如果这个磁道坏了，整个盘就不能再用了。唯一可做的事是将它扔掉使用另一张盘。

17.2.4.1　卷标

一个卷标是磁盘的一个电子标号。它保存在根目录中，显示在 DIR 清单和一些应用程序的信息中，卷标帮助你辨认磁盘。当你有许多软盘时，不妨使用卷标并贴外部标鉴以避免混淆。

前面例子中的格式化命令没有指定一个卷标，事实上可在原始命令中指定/V:label 越过此步。卷标中如有空格，应使用引号，如/V:"MY DISK"，否则会导致错误消息。空格被算在 11 个字符内，引号不算。如果指定/V:"MY DISK"，DOS 按 MY DISK 保存卷标。

提示： *如果不想要卷标，可键入/V:""(即引号中的一个 null(无标志)，在命令中包含一个 null 标志，就可避免在格式化过程中回答"Volume Label"询问。*

可用 VOL 命令查看卷标并用 LABEL 命令修改它，两个命令都将在第 18 章"增加的磁盘管理功能"中讨论。

17.2.4.2　系统磁盘

你可能每天都从硬盘启动。但如果驱动器 A 中有一个磁盘，则系统将试图从磁盘启动而不是从硬盘启动。通常情况下，在启动或重新启动时要保证驱动器 A 中没有磁盘。

但是，至少应该有一张软盘可从它启动，万一硬盘出现问题时就可用它启动。第 4 章"设置系统"，对制作紧急启动盘有完整的指导意见。如果你还没做，现在就可照着完成了。

试着在驱动器 A 中格式化一张系统盘。(如果前面的练习就用驱动器 A，可继续使用相同的磁盘)。将卷标定为 BOOT FLOPPY。

1. 键入 FORMAT A: /S /U /V:"BOOT FLOPPY"。

你将看到下面的消息：

```
Insert new disk for drive A:
and press ENTER when ready...
```

2. 磁盘已在驱动器中，按 Enter。

屏幕显示：

```
Formatting 1.44M
          0 percent completed
Format complete
System transferred
```

System transferred 消息告诉你系统文件已从启动驱动器拷贝到磁盘。

下面的消息告诉你磁盘的容量：

```
1,457,664 bytes total disk space
  18,432 bytes used by system
   1,024 bytes in bad sectors
1,438,208 bytes available on disk

      512 bytes in each allocation unit.
2,845 allocation units available on disk.
```

注意系统文件占用的空间已从有用空间减去。

下面，FORMAT 显示 Volume Serial Number，并询问是否格式化另一张盘：

```
Volume Serial Number is 2A46-17F8
Format another (Y/N)?
```

如果回答 Y，FORMAT 格式化另一张盘，带有系统文件和相同的卷标，无法改变任何参数。

3. 键入 N 结束 FORMAT 程序。

返回命令提示符。

答疑： 我试图格式化一张系统盘，得到消息 Insert DOS disk in drive and strike any key when ready。有什么东西错了吗？

没有任何问题。只是这次是从软盘启动，FORMAT 要你插入启动盘，以拷贝文件到新盘。你所要做的只是把启动盘插入指定驱动器（可能为 A:），按任一键继续即可。

17.2.5 使用 FORMAT 重新格式化一张盘

图 17.2 显示了重新格式化一张盘的命令参数。

FORMAT

准备一张盘以备重新使用。

句法：

FORMAT drive [/V:label] [/S] [/Q] [/U]

参数和选项：

drive	指定格式化的驱动器。
/V:label	指定格式化磁盘的卷标。
/S	使磁盘是可启动的。
/Q	进行快速格式化
/U	进行无条件格式化。

图 17.2　重新格式化命令总结

以后可看到，大部分命令实际上与前面的 FORMAT 命令相同，但可增加一/Q 选项进行快速格式化。如果不使用/Q 或/U，FORMAT 进行安全格式化。

试格式化在前面练习中格式化过的软盘。这次不再使用卷标。

1. 键入 FORMAT A：/V:""或 FORMAT B：/V:""，这取决于磁盘在哪个驱动器。可看到下列消息：

```
Insert new disk for drive d:
    and press ENTER when ready...
```

2. 按 Enter 键。

可看到下列消息：

```
Checking existing disk format.
Saving UNFORMAT information.
Verifying 1.44M
        0 percent completed.
```

因为命令没有指定/Q、/S 或/U，FORMAT 检查已存在的盘看是否已经格式化。当它发现磁盘已格式化时，保存 UNFORMAT 信息，然后进行安全格式化。

下一行告诉你 FORMAT 准备格式化多少空间，它可能有不同的容量。当磁盘曾被格式化时，FORMAT 自动设定相同容量的空间。

对于安全格式化，FORMAT 验证扇区的可靠性，但不会自动地物理格式化这个磁盘。它花很少的时间验证扇区，存在的数据不被破坏。当 FORMAT 验证扇区时，完成的百分数向上增长。结束后，你可看到消息：

```
Format complete.

      1,457,664 bytes total disk space
      1,457,664 bytes available on disk

          512 bytes in each allocation unit.
        2,845 allocation units available on disk.

Volume Serial Number is 2A46-17FB
```

```
Format another (Y/N)?
```

3. 键入 N 结束 FORMAT 程序。
   返回命令提示符。

如果不想使用缺省格式化级，可指定另一个级别，试着在相同磁盘上重新做一个快速格式化：

1. 键入 FORMAT A：/Q /V:""或 FORMAT B：/Q /V:""，取决于使用的驱动器。
   将看到下面的消息：

```
Insert new disk for drive d:
and press ENTER when ready...
```

2. 按 Enter 键。
   将看到下列消息：

```
Checking existing disk format.
Saving UNFORMAT information.
QuickFormatting 1.44M
Format complete.
        1,457,664 bytes total disk space
            1,024 bytes in bad sectors
        1,456,630 bytes available on disk

              512 bytes in each allocation unit.
            2,845 allocation units available on disk.

Volume Serial Number is 2A46-17F8

Format another(Y/N)?
```

   FORMAT 检查磁盘以确认它是否已被格式化过。如果没有，则不能进行快速格式化。然后保存 UNFORMAT 信息。接着，显示磁盘空间，它与当前容量相等。这时没有百分数显示，因为快速格式化不验证扇区。
3. 键入 N 结束 FORMAT 程序。

提示： 如果确定不想重新格式化磁盘，而且它的扇区情况良好，同时使用/Q 和/U 可能是最快的格式化。/Q 使 FORMAT 做一个快速格式化，/U 防止 FORMAT 存储取消格式化文件。同时，/Q 使/U 无效——FORMAT 不重新分配扇区。

答疑： 我试图做一个安全或者快速格式化，得到下列消息：

```
Drive d error. Insufficient space for the MIRROR image file.
There was an error creating the format recovery file.
This disk can't be unformatted.
Proceed with Format (Y/N)?
```

在安全或快速格式化时，FORMAT 将存储取消格式化文件，即镜象文件，这样以后若有必要，可取消格式化。当磁盘上没有足够的空间存储镜象文件时会得到这些信息。

如果你保证以后不会取消格式化磁盘，键入 Y 继续格式化而不保存取消格式化信息。这样以后就不能取消格式化了。

否则，键入 N 结束 FORMAT 程序。然后，删除磁盘上的一些文件以便挪出空间来保存 UNFORMAT 信息。（需要的空间大小取决于磁盘容量，对一个 1.44M 软盘可能需要 10KB 空间）。

预览： 第 46 章解释了 FORMAT 命令的增强功能，如早期 DOS 系统按每磁道 8 个扇区和指定一个非缺省的容量来格式化磁盘。

17.3　在 DOS Shell 中 FORMAT

DOS Shell 的 Disk Utilities 组中包含两个格式化选项。第一个称为 Quick Format，定义为 FORMAT %1 /Q。当打开它时，出现一个对话框指定驱动器名，缺省设置为 A:（如图 17.3 所示）。如果想格式化驱动器 B:用 B:代替 A:一旦选择 OK，设置为对话框中键入其他信息也包含在 FORMAT 命令中传到 DOS。例如，可以增加/V 选项定义卷标，/S 选项使磁盘可用来启动，或/U 选项用来取消镜象文件。

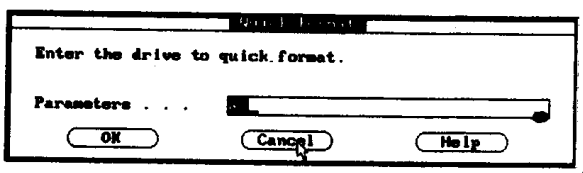

图 17.3　在 DOS Shell 中的 Quick Format 对话框

如果只有一个软驱，可设置为 FORMAT A:/Q，这样就不必处理对话框了，但同时也就没有机会增加选项到对话框的命令里。这可能是付出了较小的代价换取了取消对话框带来的方便。第 11 章“自定义 Shell”描述了如何定义和修改程序项。

第二个选项称为简单 Format，由命令 FORMAT %1 定义。当打开它时，一对话框提示设置驱动器，缺省为 A:。而且，在这儿可从对话框键入选项到命令中。如果只有一个驱动器，可重新定义此项为 FORMAT A:，以便不经过对话框。

在这两种情况下，如果要重定义选项，可选择 File 中的 Run 打开一个对话框键入 FORMAT 的其他选项。第 12 章“在命令提示符下工作”解释了如何使用 File，Run 命令。

17.4　使用 UNFORMAT 恢复一个重新格式化的磁盘

假设刚格式化了一个磁盘，当把它从驱动器中取出，一看标志，却发现用错盘了。UNFORMAT 可用来恢复数据（如图 17.4 所示）。

UNFORMAT 通过恢复以前的根目录和 FAT（存于一个镜象文件中）来恢复磁盘上的数据。只要重新格式化后没有增加任何东西，就可完全恢复以前的内容。但是，如果增加了一些东西，也许就会发现一些文件被破坏了。即使这样，总比丢失磁盘上所有的数据好。

取消格式化会遇到两个主要问题。第一个是 UNFORMAT 不能找到镜象文件。UNFORMAT 可能通过搜寻目录入口的簇来从头开始重建磁盘的目录结构，但这样处理不会产生最好的结果，它最多是救急方法。你将在第 47 章“高级补救技术”学习如何处理它。

另一个问题是由于你格式化磁盘后在其中加入了数据，甚至格式化后增加了 DOS 的核心程序文件，这使它成为一个可启动盘。这些新数据会覆盖你想恢复的以前的数据。当然你不能修复覆盖的数据。UNFORMAT 不能辨别这个问题，使用者必须自己留心如果格式化增加了数据，就有可能丢失一些原来的数据。而且，取消格式化时，又会丢失新装入的数据。

UNFORMAT

恢复一个以 FORMAT 重新格式化的盘。

句法：

UNFORMAT drive

参数：

drive　　指定取消格式化的驱动器。

图 17.4　UNFORMAT 命令总结

让我们增加一些数据到练习盘中，然后格式化，再取消格式化。

1. 从磁盘上拷一些文件到你的软盘上，任何文件都可。
2. 列出软盘目录内容，看看拷贝的文件。
3. 使用/Q 选项对软盘进行快速格式化。
4. 列出软盘目录内容。
   可看到其中没有任何文件。
5. 键入 UNFORMAT A:或 UNFORMAT B:，这取决于软盘所在的驱动器。
   显示的第一条消息是：

```
Insert disk to rebuild in drive d:
and press ENTER when ready
```

6. 盘已在驱动器中，只须按 Enter。
   下面，将看到一整屏幕消息：

```
Restores the system area of your disk by using the image file created ty the MIRROR command.
WARNING !!              WARNING!!
This command should be used only to recover from the inadvertent use of the FORMAT command or
the RECOVER command. Any other use of the UNFORMAT command may cause you to lose data!
Files modified since the MIRROR image file was created may be lost.
Searching disk for MIRROR image.
The last time the MIRROR or FORMAT command was used was at 22:14 on 10-16-94.
The MIRROR image file has been validated.
Are you sure you want to update the system area of your drive(Y/N)?
```

   这些消息指出通常在 DOS 中的 MIRROR 和 RECOVER 程序丢失了。只有 FORMAT 程序保存了一个镜象文件。
7. 键入 Y 继续取消格式化操作。
   你将看到下面消息：

```
The system area of drive d has been rebuilt.
```

You may need to restart the system.

8. 运行 UNFORMAT 后重新启动以清除驻留在系统缓冲区和缓冲器中的数据总是一个好主意。从驱动器 A 中移去盘，按 Ctrl+Alt+Delete。
   DOS 重新启动。

预览： 如果实在是无可奈何，也可在没有镜象文件的情况下对一磁盘取消格式化。但是，这时许多文件不能被完全恢复。第 47 章讲述了在镜象文件丢失后如何取消格式化。

17.5 自 测 练 习

在本章中已学习了如何对一个软盘进行格式化、重新格式化和取消格式化。也学习了格式化一个硬盘的步骤，尽管没有实践。如果想自己多多练习，可以下列步骤作为指南：

1. 对驱动器 A 或 B 中的软盘做无条件格式化。命名为 ON MY OWN1。
2. 对相同磁盘作一个安全重新格式化命名为 ON MY OWN2。
3. 增加一些文件到软盘上，做一个快速重新格式化。命名为 ON MY OWN3。
4. 取消格式化，恢复在第 3 步丢失的文件。
5. 确认已明白了如何准备硬盘投入使用的步骤。什么程序进行物理格式化？分区？逻辑格式化？

第 18 章 增加的磁盘管理功能

简介： 你可能不时地想要复制磁盘，而有时可能要修改在格式化过程中分配的标号。本章讨论完成这些任务的方法。

18.1 总 览

DOS 中有几个命令可用来管理磁盘，其中大多数只用于软盘。例如，可复制一个软盘，然后比较复制盘和源盘看它们是否完全相同。DOS 还包含两个浏览和改变磁盘卷标的命令。这些命令可用在软盘或硬盘上。

本章主要讲述：

- 在命令提示符下进行软盘复制
- 在 DOS Shell 中作复制拷贝
- 为 DOS Shell 定义磁盘拷贝程序项
- 比较两个软盘
- 浏览和修改一个磁盘的卷标

18.2 用 DISKCOPY 拷贝磁盘

无论何时购买了一个新的软件程序，都应通过备份所有的磁盘来保护投资。把源盘保存好，使用备份盘安装软件。作者推荐将这两套盘保存在不同的地方，这样就不必惧怕天灾人祸。

注释： 不能在只读驱动器如 CD-ROM 驱动器中复制磁盘。

图 18.1 总结了 DISKCOPY 命令，它用来拷贝磁盘。DISKCOPY 对磁盘的每个磁道作一个完全相同的拷贝。因为 DISKCOPY 通过拷贝完整的磁道来拷贝源盘的物理和逻辑格式，目标盘不必首先格式化。实际上，拷贝过程会取代磁盘上已存在的任何格式。

复制一个磁盘与拷贝一个磁盘上的所有文件到另一个磁盘效果很不相同。文件拷贝不复制源盘的系统区，它不删除目标盘上已有的数据(除了用与目标盘相同的文件名进行覆盖例外)，而且它将拷贝存储到可以找到有用簇的地方，可能与源盘的分布并不同；另一方面，逐磁道拷贝覆盖整个目标盘，包括开始的系统区域和结尾的任何空磁道，目标盘和源盘上的簇的顺序也完全相同，并保持源盘的任何文件碎片。

DISKCOPY

将一个软盘上的所有内容拷贝到另一个软盘。

句法：

DISKCOPY [source] [destination] [/V]

参数和选项：

source	指定放源盘的驱动器。
destination	指定放目标盘的驱动器。
/V	验证拷贝数据的正确性。

图 18.1　DISKCOPY 命令总结

当拷贝软件盘时，逐磁道拷贝可能很重要，因为只有这样应用程序的 Install 或 Setup 程序才能正确工作。例如，一些安装程序会首先检查插入的磁盘的卷标。如果卷标不是它所要求的，它会继续要求插入正确的磁盘。

注释： DISKCOPY 不能用来拷贝硬盘，只能拷贝软盘，而且必须有相同的尺寸和容量。

拷贝一个磁盘的格式可能产生一些问题，如果源盘有一些坏扇区空出，那些目标盘上的相同扇区也要被空出，即使它们实际上没坏。这不是一个大问题，只是浪费一点空间罢了。但是，如果情况相反——目标盘上有坏点而对应在源盘上却不是坏点。如果 DISKCOPY 发现这个区域不能写入而识别了这个问题，它将显示一个出错信息。如果它没注意到这个坏点，可能以后就会丢失一些数据。

答疑： 在复制一张软盘时，DISKCOPY 显示消息 Unrecoverable read error on drive d；side x，track xx；Target disk may be unusable. 出现什么问题了呢？

因为 DISKCOPY 一个磁道一个磁道拷贝，它不读源盘的 FAT。如果源盘的 FAT 中有坏点，DISKCOPY 无法辨识而继续拷贝。当它遇到一个读错误时，它显示这个消息。一般而论，坏点不包含有效数据，它不会在目标盘上产生问题。但是也有可能源盘上出现了新的坏点并且有一个或多个文件已被破坏。为了确认无错，可用 XCOPY 拷贝源盘上的所有数据到另一个磁盘。如果拷贝正确，则源盘没有问题。

答疑： 在复制磁盘时，DISKCOPY 显示消息 Unrecoverable write error on drive d；Side x，track xx；Target disk may be unusable. 出了什么问题？

在拷贝期间，DISKCOPY 在目标盘上发现了一个坏区，因而不能正确地复制一个或多个磁道。在这种情况下，目标盘不能再使用，用另一个目标盘再进行拷贝。

带坏点的盘并非完全不能用。在其上做一个无条件格式化可空出坏点区域。它可用来存储数据，但不能作为一个复制盘。

练习复制拷贝需要两张磁盘，一个包含数据，一个为空(或可重新使用的)。两个磁盘必须有相同的尺寸和容量，如两个 3 ½英寸 1.44M 盘。

警告： DISKCOPY 会破坏目标盘上的所有数据，这些数据不能通过任何方式恢复。因此，要确认目标盘不包含有用数据。

因为两张盘必须有相同尺寸和容量，DISKCOPY 的使用方式取决于所有的软驱的数目和类型：

- 如果只有一个驱动器，请参见“在 DOS 6.0 或更早的 DOS 中使用一个驱动器拷贝磁盘”或“在 DOS 6.2 或更新的 DOS 中使用一个驱动器拷贝磁盘”，具体取决于你的 DOS 版本。
- 如果有两个相同的驱动器，可使用两个驱动器进行拷贝，参看“用两个驱动器拷贝磁盘”。
- 如果有一个高密驱动器和一个相同尺寸的低密驱动器(3 ½英寸或 5 ¼英寸)，可使用两个驱动器拷贝一个低密磁盘，但必须只使用高密驱动器拷贝高密磁盘。根据你想拷贝的磁盘类型和 DOS 版本参看适当章节。
- 如果有两个不同尺寸的驱动器，如一个 3 ½英寸和一个 5 ¼英寸驱动器，不能同时用两个驱动器进行拷贝。参看“在 DOS 6.0 或更早的 DOS 中使用一个驱动器拷贝磁盘”或“在 DOS 6.2 或更新的 DOS 中使用一个驱动器拷贝磁盘”，取决于所用的 DOS 的版本。

提示： 实际操作中，很容易将两个盘混淆。源盘总应加上写保护，以避免意外地从目标盘拷贝磁道到源盘中。

18.2.1 在 DOS 6.0 或更早的 DOS 中使用一个驱动器拷贝磁盘

当用一个驱动器拷贝时，必须在相同驱动器中交换源盘和目标盘。对于使用 DOS 6.0 (或更早)的一驱动器拷贝，必须多次交换源盘和目标盘。DISKCOPY 告诉你什么时候插盘。

试使用驱动器 A 进行拷贝：

1. 写保护源盘。
2. 键入 DISKCOPY A: A:。

   DISKCOPY 提示 Insert SOURCE disk in drive A:,Press any key to continue。
3. 插入源盘按任一键。

   DISKCOPY 显示磁盘参数，如：

   ```
   Copying 80 tracks,18 sectors per track,2 side(s)
   Reading from source diskette...
   ```

   DISKCOPY 拷贝源盘上尽可能多的数据到内存，并显示如下消息：

   ```
   Insert TARGET diskette into drive A:
   Press any key to continue...
   ```

4. 插入目标盘，按任一键。

   DISKCOPY 写数据到目标盘，然后再次要求插入 SOURCE 盘。
5. 重复 3,4 直到拷贝完成。

DISKCOPY 分配一个新卷系列号到目标盘，并显示这个号码，然后问 COPY another diskette?。

6. 如果现在不想拷贝另一个磁盘，按 N 结束 DISKCOPY 程序。
7. 移去源盘的写保护，除非决定一直让它写保护。
8. 给目标盘贴上标签。如果是 5 ¼英寸软盘，保证在贴标签之前写标签，这样你就不会用笔压在磁盘上了。

18.2.2　在 DOS 6.2 或更新的 DOS 中使用一个驱动器拷贝磁盘

当使用一个驱动器拷贝磁盘时，必须在相同的驱动器中交换源盘和目标盘，用 DOS 6.2 和 6.22，只需交换一次盘。DISKCOPY 拷贝所有的源数据到一个临时文件中，然后拷贝到目标盘上。（第 43 章"DOS 及其运行环境"解释了 DOS 在哪里放置它的临时文件）。

答疑： 我试着做一个磁盘拷贝，却得到消息 Error creating image file; Diskcopy will revert to multiple-pass copy。

用 DOS 6.2 和更新的 DOS 的单驱动器拷贝时，DISKCOPY 从源盘拷贝所有数据到一个临时文件。但在某些情况下，DISKCOPY 不能创建临时文件。大部分情况是因为系统存储临时文件的磁盘满了。

因为它不能创建临时文件，DISKCOPY 将回到老式拷贝过程，它需要你在拷贝过程中交换几次磁盘才能完成拷贝。不会遇到任何麻烦，只是比平时多了一点点的不方便。

在你完成拷贝后，应该找出系统不能创建临时文件的原因。其他的 DOS 过程也需要创建临时文件，但不能和 DISKCOPY 共用临时文件。第 43 章对 DOS 中的临时文件作了完整的解释。

试着只使用驱动器 A 作一个拷贝：

1. 写保护源盘。
2. 键入 DISKCOPY A: A:。

   DISKCOPY 提示 Insert SOURCE disk in drive A:, Press any key to continue。
3. 插入源盘到驱动器 A 中按任一键。

   DISKCOPY 显示磁盘参数，如：

   Copying 80 tracks,18 sectors per track, 2 side(s)

   Reading from source diskette...

   DISKCOPY 拷贝源盘中的所有文件到一个临时文件，显示下面消息：

   Insert TARGET diskette into drive A:

   press any key to continue...

4. 插入目标盘到驱动器 A 中，并按任一键。

   DISKCOPY 将拷贝数据写入到目标盘，因为它将数据存在临时文件中，DISKCOPY 可迅速容易地拷贝另一个相同磁盘，然后显示下面消息：

Do you wish to write another duplicate of this disk?

5. 如果想作第二份复制盘，按 Y。但现在是练习，按 N。
   DISKCOPY 分配一个新的卷系列号到目标盘，并显示这个号码，然后它问 Copy another diskette?。
6. 现在不想拷贝另一个磁盘，按 N 结束 DISKCOPY 程序。
7. 移去源盘写保护，除非想让它保持写保护。
8. 给目标盘贴上标签，如果是 5 ¼英寸软盘，要保证在贴上之前写检签，这样就不会用笔压在磁盘上了。

18.2.3 用两个驱动器拷贝磁盘

当使用两个驱动器拷贝磁盘时，可把源盘放在一个驱动器，目标盘放在另一个驱动器中。如果两个驱动器有不同的容量，如一个 720KB，另一个 1.44MB，只能从高密驱动器往低密驱动器拷贝一个低密磁盘。一个高密驱动器也可写一个低密盘，但结果可能不可靠。

试着用两个驱动器进行磁盘拷贝：

1. 写保护源盘。
2. 如果两个磁盘容量相同，键入 DISKCOPY A: B:。如果驱动器 A 是高密而驱动器 B 为低密，键入 DISKCOPY A: B:，这样将在驱动器 B(低密)上写。如果驱动器 A 是低密而驱动器 B 是高密，键入 DISKCOPY B: A:，这样将在驱动器 A(低密)上写。
   DISKCOPY 显示下面信息：

   Insert SOURCE disk in drive x,
   Insert TARGET disk in drive y
   Press any key to continue.

3. 插入源和目标盘到指定驱动器，按任一键。
   DISKCOPY 显示磁盘参数：

   Copying 80 tracks,18 sectors per track,2 side(s)

   然后，在拷贝数据时显示消息：

   Reading from source diskette...
   Writing to target diskette...

   DISKCOPY 写拷贝数据到目标盘上。
   DISKCOPY 分配一个新卷系列号给目标盘，并显示这个号码。然后询问 Copy another diskette?。
4. 如果不想拷贝另一张盘，按 N，结束 DISKCOPY 程序。
5. 从源盘移去写保护，除非决定想保留写保护。
6. 给目标盘贴上标签，如果是 5 ¼英寸软盘，要保证贴上之前写好标签，这样就可避免用笔压在磁盘上。

答疑： 在用 DISKCOPY 复制磁盘时，得到消息 TARGET diskette bad or incompatible; Copy process ended。

这或者是因为目标盘的尺寸或容量不对，或者它在正常的区域包含表面缺陷。只能换一个目标盘继续复制过程。

18.2.4　验证拷贝

验证选项(/V)增加了一种保护拷贝的额外方式。驱动器控制器将自动验证它保存的所有数据，但在数据到达驱动器控制器之前可能会产生一些错误。(这些错误是不常见的，它们大部分由电源或硬件功能不正常产生)。DOS 验证功能可比较写入的拷贝和内存中的拷贝，并校正发现的错误。坦白地说，这种类型的错误是如此稀少，以至于可认为 DISKCOPY 不该花这个代价去寻找它。

注释： 如果是使 DOS 的一般验证功能有效(参见 VERIFY 命令)，则不需要/V 选项。它们功能相同。

DOS 验证不能找到在数据离开内存前产生的错误。在本章后面讨论的 DISKCOMP 命令可完成此种操作。

18.3　在 Shell 中拷贝磁盘

如果使用 DOS Shell，可从 Disk Utilities 组复制磁盘，名为 Disk Copy 的程序项可生成命令 DISKCOPY %1。选中它时，将打开一个对话框，这样就可键入驱动器名(如图 18.2 所示)。由图可见对话框设定 A:和 B:作为缺省驱动器名，可在选择 OK 之前编辑系统需要的驱动器名。

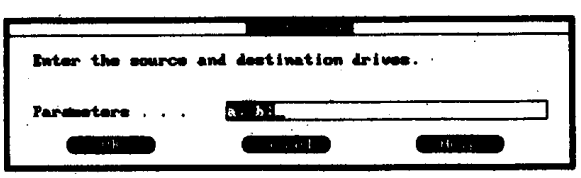

图 18.2　在 DOS Shell 中的 Disk Copy 对话框

如果让我选择，应该用一个编辑得适合于自己系统的程序项来代替当前项。这样就不必每次通过对话框指定磁盘驱动器，因为驱动器设置是不变的。第 11 章“自定义 Shell”解释了如何为 Shell 的程序清单定义和编辑程序选项。

如果只有一个驱动器，将定义从 DISKCOPY %1 改成 DISKCOPY A: A:。如果有两个相同大小和容量的驱动器，改变定义为 DISKCOPY A: B:取消对话框。

如果有两个相同大小但容量不同的驱动器，删除此项并创建另两项。第一项，可称为 Disk Copy Double Density，从高密盘往低密盘拷贝。第二项，可称为 Disk Copy High Density，只在高密驱动器上拷贝。

如果有两个不同大小的驱动器，删除这个定义并创建程序项 Disk Copy A:(定义为 DISKCOPY A: A:)，及 Disk Copy B:(定义为 DISKCOPY B: B:)。

18.4 用 DISKCOMP 比较两个磁盘

已经看到，DOS 的验证特点不能确保一个拷贝与原始盘完全匹配；它只验证被写的版本和内存中的版本。要对拷贝盘和原始盘进行完全的验证，必须使用 DISKCOMP 进行比较。

答疑： 我试图复制或比较磁盘，得到消息 Invalid drive specification；specified drive does not exist or is non-removable。

当试图拷贝一个不可移动磁盘时，就会出现这样的 DISKCOPY 或 DISKCOMP 消息。DISKCOPY 和 DISKCOMP 只处理可移动磁盘如软盘，它们不能处理 CD-ROM 驱动器，尽管 CD 是可移动的。

如果指定的驱动器是一个软驱，则可能是因为 DOS 在启动时不能正确找到这个驱动器。参看 DRIVER.SYS 和 DRIVPARM 寻找为 DOS 定义一个驱动器的方法。

DISKCOMP 命令几乎和 DISKCOPY 的格式一样，但不是拷贝磁盘而是比较磁盘。图 18.3 显示了命令格式。与 DISKCOPY 不同，DISKCOMP 在 DOS 6.2 中没有加以改进，因此，对于单驱动器比较，还必须交换多次磁盘。

DISKCOMP

比较两软盘。

句法：

DISKCOMP [drive1] [drive2]

参数和选项：

drive1	指定进行比较的软盘所在驱动器。
drive2	指定进行比较的另一软盘所在的驱动器。

图 18.3 DISKCOMP 命令总结

试比较刚刚拷贝的两个磁盘：

1. 如果做的是单驱动器拷贝，键入 DISKCOMP d: d:，d 表示驱动器名；如果做的是双驱动器拷贝，键入 DISKCOMP A: B:。(对于 DISKCOMP，它不考虑哪个驱动器是高密，哪一个驱动器是低密)。

   DISKCOMP 将提示何时插入哪一张盘。
2. 遵循屏幕指导。

   结束时，DISKCOMP 显示消息 Compare OK。然后，它问你是否想比较另一个。
3. 键入 N 结来 DISKCOMP 程序，然后回到命令提示符。

   如果 DISKCOMP 发现了一个错误，它为每个不匹配的磁道显示下面的消息：

   Compare error on side x, track xx

   如果 DISKCOMP 在你刚用 DISKCOPY 复制的拷贝中找到了不匹配磁道，这个文件

可能是防拷贝的。检查销售商的资料找出是否如此。如果它不是防拷贝的，问题可能是由于硬件功能失误如存储芯片失败，或者是由于拷贝过程中的电源起伏。试着再进行一次拷贝。如果不能用你的系统进行正常的磁盘拷贝，它可能需要修理。

DISKCOMP 一个磁道一个磁道地比较磁盘，而不是一个文件一个文件地进行。当你使用 COPY 或 XCOPY 拷贝一个磁盘的所有文件到另一个磁盘时，结果不是一个磁道一个磁道的拷贝，因此 DISKCOMP 会找到不匹配磁盘。

注释： 不能用 DISKCOMP 比较两个 CD，尽管 DISKCOMP 并不写磁盘。如果用 DISKCOMP 寻址一个 CD-ROM 驱动器，会得到消息 Cannot DISKCOMP to or from a network drive。(DISKCOMP 对这个问题显然有点混淆。尽管如此，含义却很清楚——你不能在这个驱动器上使用 DISKCOMP)。

18.5　用 VOL 和 LABEL 管理内部标号

进行格式化时，可为磁盘创建一个卷标，卷标保存在内部(根目录中)，出现在目录和树清单等地方。DOS 不使用卷标，它通过系列号辨认磁盘，卷标主要能给使用者带来方便。但其他程序可能会注意磁盘卷标。例如，一些 Install 或 Setup 程序会检查磁盘卷标来确保插入正确的磁盘。

一些应用程序在驱动器清单中显示卷标。例如，图 18.4 显示了 WordPerfect for Windows 6.0 的 File Open 对话框，可在 Drives 下面看到驱动器清单。硬盘的卷标显示在清单中。

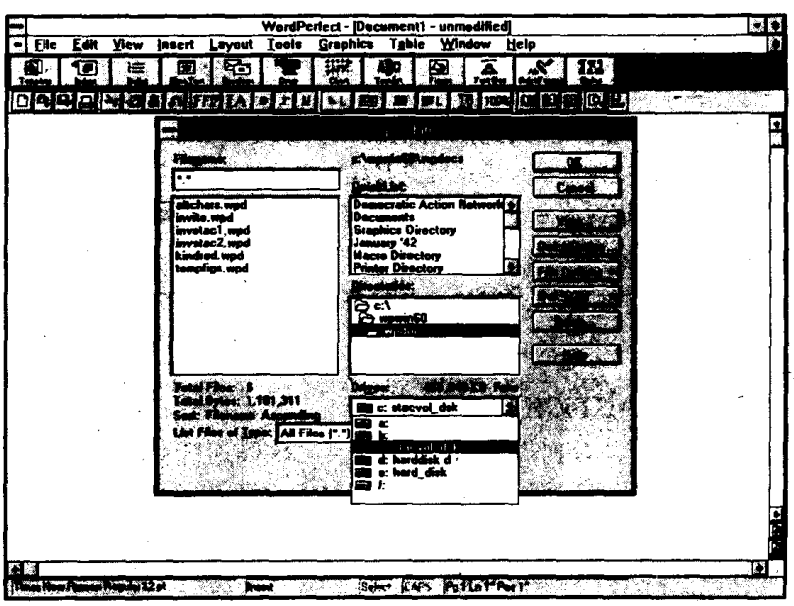

图 18.4　一个显示卷标及驱动器的典型驱动器清单

可用 VOL 命令显示一个磁盘的卷标(如图 18.5 所示)。试浏览你的磁盘卷标：

1. 键入 VOL 查看当前磁盘的卷标。

DOS 显示消息如下(你的卷标和系列号可能不同)。

```
Volume in drive C is MY HARD DRIVE
Volume Serial Number is 1CFB-7D56
```

2. 键入 VOL A:查看驱动器 A 中磁盘的卷标。

VOL

显示一个磁盘的卷标和系列号,如果存在的话。
句法:

VOL [drive]

参数和选项:

none	如果有的话,显示当前驱动器的卷标和系列号。
drive	指定显示卷标和系列号的驱动器。

图 18.5 VOL 命令总结

要改变卷标,可使用 LABEL 命令(如图 18.6 所示)。可在命令中包含新的卷标,或让 LABEL 进行提示。试改变驱动器 A 中磁盘的卷标为 PRACTICE:

LABEL

显示、更改磁盘卷标。
句法:

LABEL [drive] [label]

参数:

none	显示当前驱动器中磁盘卷标以进行更改或删除。
drive	指定所有显示或更改的磁盘所在驱动器,缺省值为当前驱动器。
label	给磁盘指定一最多包含 11 个字符的新卷标。

图 18.6 LABEL 命令总结

1. 键入 LABEL A: PRACTICE。

可听到驱动器在工作并看到灯亮,但不能从 DOS 得到任何确认消息。

2. 键入 VOL A:。

你没有在 LABEL 命令中使用引号,正如在 FORMAT 命令中一样,即使新卷标包含一个空格。这意味着不能通过在命令行包含一个空卷标来删除一个卷标。不用引号,就无法指定 null 卷标。但是,可以键入不带卷标文本的 LABEL 命令,等待 LABEL 提示输入卷标。

网络注意: 可以浏览网络驱动器的卷标,但不能改变它们。

注释: 可浏览只读驱动器中磁盘的卷标,但不能改变它。如果试图改变 CD 的卷标,将会得到消息 Cannot Label a network drive。

试从驱动器 A 中删除其中磁盘的卷标：

1. 键入 LABEL drive，不带参数。
   DOS 显示当前卷标并提示输入新卷标。
2. 按 Enter。
3. 当 LABEL 询问是否想删除当前卷标时，键入 Y。

18.6　自 测 练 习

在本章中，学习了如何复制磁盘、比较拷贝以及浏览与改变一个磁盘的卷标。如果想自己实践这些技术，可以使用下列步骤作为指南：

1. 对驱动器 A 的软盘作一个复制拷贝。
2. 如果另有一个驱动器 B，大小或容量与驱动器 A 不同，用驱动器 B 作一个复制拷贝。
3. 比较 1、2 的复制盘和源盘。
4. 改变两个复制盘的卷标为 ON MY OWN1 和 ON MY OWN2。

第 19 章 DOS Editor

简介： DOS 的 Editor 可用来处理其他字处理程序难以对付的文本文件。

19.1 DOS Editor 简介

DOS 总是包含一个文本编辑器，以便用户可以编辑一些简单的 ASCII 文本文件，如 AUTOEXEC. BAT 和 CONFIG. SYS，而不需使用其他字处理程序。但是，DOS 最初的编辑器非常不友好，以致于只有一些专门的程序员才喜欢它。DOS 5 最终推出了一个易使用的全屏幕图形接口编辑器，DOS Editor（或 EDIT）。

DOS Editor 不仅使创建和修改文本文件变得容易，而且可以很方便地浏览它们。用 DOS Editor，可打开一个资料如 README. TXT，来回滚动，打印全部或部分文本，增加自己的注释，等等。

如果你经常使用一个字处理程序，并已习惯于它的操作方式，可能会经常选用它，即使只是编辑文本文件。即便如此 DOS Editor 仍有其优势。DOS Editor 特别为文本文件而设计，大部分流行的字处理程序的资料文件都被赋予了某种格式，包含非 ASCII 格式字符。为了避免这种格式，必须特意以 ASCII 格式存储文件。如果忘了这么做而让字处理程序格式化了这个文件，DOS 就不能读它。如果这是一个. BAT 文件或 CONFIG. SYS 文件，就不能正常工作。

虽然 DOS Editor 没有一般字处理程序的全部功能，却有着比许多文本编辑器更多的方便之处。

- DOS Editor 的图形用户接口使使用者能使用鼠标或键盘编辑文本以及访问菜单和对话框
- DOS Editor 有一个完全的联机帮助系统可指导使用者进行不熟悉的操作
- DOS Editor 包括一个 Clipboard（剪贴板）。这样可在一个文档或文档间剪切、复制和粘贴文本
- 可搜寻文本，并用其他文本代替

本章主要讲述：

- 启动 EDIT
- 插入，删除和覆盖文本
- 剪切和粘贴文本
- 搜寻和取代文本
- 保存一个文件
- 打开一个编辑文件
- 打印文件

- 使用 Editor 时获取联机帮助

警告： DOS 的 Editor 是 QBASIC 编程系统的一个功能。如果删掉了 QBASIC.EXE，就不能使用 EDIT。

19.2　启动 DOS Editor

用几种方法启动 DOS Editor，可从 Shell 使用这些技术：

- 选中 Editor(在 Main 组)
- 在 DOS 目录中打开 EDIT.COM
- 释放一个 ASCII 文本文件到 EDIT.COM 上
- 打开一个.TXT 文件。(所有的.TXT 文件与 EDIT.COM 连接)
- 使用 File,Run 在 DOS 提示符下键入 EDIT 命令

在 DOS 命令提示符下，可使用 EDIT 命令启动编辑器(如图 19.1 所示)。如果不包含一个文件名，DOS Editor 开始时带一个空的，无标题文档。

EDIT

启动 DOS 的全屏幕编辑器，它可创建、修改和显示 ASCII 文本文件。

句法：

EDIT [filespec]

参数：

none	不带启动文件，使用缺省屏幕特征启动编辑器。
filespec	打开一个 ASCII 文本文件，如果文件不存在，创建一个文件；如果文件已存在，则显示它的内容。

图 19.1　EDIT 命令总结

试启动编辑器：

1. 如果不是在主 DOS 命令提示符下，退出当前程序，显示此命令提示符。
2. 键入 EDIT 启动编辑器。

   一个初始化对话框告诉你按 Enter 寻求帮助或按 Esc 到编辑屏幕。如果按 Enter，则显示一些初始帮助信息。按 Esc 时，则看到一个空白编辑屏幕，这时可开始创建一新文档。
3. 按 Esc 消除对话框。

提示： 当你想创建一个新文件时，可在 EDIT 命令中包含新的文件名启动 Editor。EDIT 立刻创建新文件，虽然它不包含文本。

图 19.2 显示了 DOS Editor 的屏幕示样，此时一文档被打开以备编辑。顶部是菜单栏，中间是主要部分，资料编辑区域。底端行是状态栏，两个滚动条提供了鼠标控制。

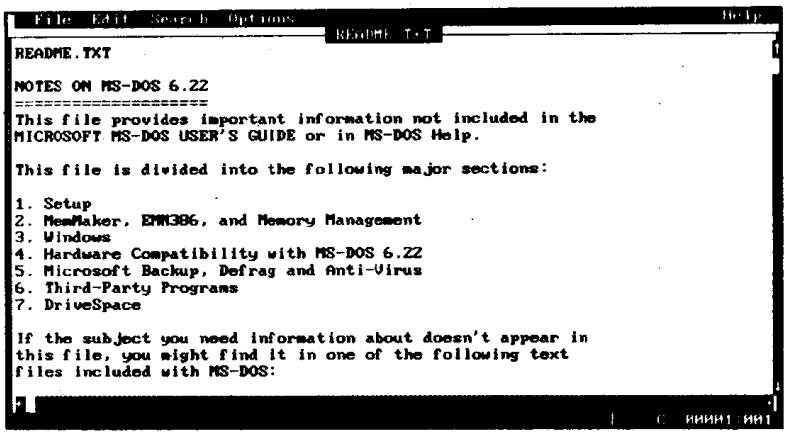

图 19.2 DOS Editor 屏幕

19.3 行操作

如果你已习惯于使用字处理程序，也就习惯于按段工作。但是，文本编辑器却按行而不是按段工作。一行最多 255 字符，以回车符结束。DOS Editor 不自动换行，如果一行长于显示区域，它继续进入屏幕两边，这时必须左右滚动才能看到整个行。使用水平滚动条就可右右翻卷，或简单地向一定方向移动光标即可。

试着键入一个长行查看屏幕滚动，键入林肯的葛底斯堡演说的第一个句子如下：

Four score and seven years ago,our fathers brought forth on this continent a new nation,conceived in liberty and dedicated to the proposition that all men are created equal.

你将看到当到达 new nation 时，屏幕开始滚动。

19.4 编辑文本

可通过插入、删除、覆盖、拷贝和移动来编辑一个已存在的文档，这意味着你需要掌握如何移动光标，选择文本，在 Insert 和 Overstrike 模式下转换，剪切和粘贴，搜寻和取代等。可用鼠标完成大部分工作。如果你更喜欢键盘，DOS Editor 不仅可用标准键盘的光标键如上下箭头键，而且可用 WordStar 控制键如 Ctrl+E 和 Ctrl+X。

提示： 键盘上的许多键与 Ctrl 组合时可执行各种功能，如光标移动功能。但如果先按 Ctrl+P，然后按 Ctrl+letter，则会产生一个特殊字符。例如，Ctrl+P 后面跟一个 Ctrl+Y 可在文档插入一个下箭头(↓)。

当你按 Ctrl+P 时，符号 ^P 出现在状态栏，它提醒你键入的下一个字符是一个特殊字符。一旦键入下一个字符，^P 从状态栏消失。

键入特殊字符的另一种方法是在键入一个数字代码时按住 Alt 键。可用这种方法键入系统代码页中的任何字符。第 49 章“处理另一个国家的信息”解释了代码页以及使用方法。

19.4.1　光标移动

用鼠标移动光标，可按需要翻卷然后单击适当位置。如果喜欢用键盘移动到所需位置。表 19.1 显示了光标移动键。注意 Wordstar 键给出了一些标准键没有的功能，不用担心这些不熟悉的组合键的记忆，用标准键总能到达想到的地方，只是有时花的时间长一点而已。

表 19.1　光标移动键

标准键	WordStar 键	功能
↑	Ctrl+E	上移一行
↓	Ctrl+X	下移一行
←	Ctrl+S	左移一个字符
→	Ctrl+D	右移一个字符
Ctrl+→	Ctrl+A	左移一个词
Ctrl+←	Ctrl+F	右移一个词
Home	Ctrl+Q,S	行的开始
End	Ctrl+Q,D	行的末尾
Ctrl+Enter	Ctrl+J	下一行的开始
(none)	Ctrl+Q,E	窗口顶端
(none)	Ctrl+Q,X	窗口底部
Ctrl+Home	Ctrl+Q,R	文件开端
Ctrl+End	Ctrl+Q,C	文件末尾

练习在自己创建的文件中移动光标：

1. 按 Home 跳到行首。
2. 按右箭头键向右移动光标。
3. 按 Ctrl+右箭头键向右一个词一个词移动光标。
4. 按左箭头键向左移动光标。
5. 按 Ctrl+左箭头向左一个词一个词移动光标。

如果配有鼠标，练习左右滚动，及单击一个地方将光标移到那儿。

19.4.2　插入文本

缺省状态下，处于 Insert(插入)模式。按 Ins 键或 Ctrl+V 在 Insert 和 Overstrike(覆盖)两种模式间转换。当处于 Overstrike 模式时，光标变成闪烁矩形。

要插入文本，将光标移到想插入的位置，要确保处于 Insert 模式，然后就可开始键入了。新字符将行的余下部分向右挤。当行到达 255 个字符的限制时，DOS Editor 铃响，并拒绝接收更多的字符。

试着在已键入的短行中插入一些回车：

1. 移动光标到 continent 一词的末尾。
2. 按 Enter。

EDIT 在词 continent 后插入一个回车，句子的剩余部分移到下一行。

3. 在词 proposition 后插入一个回车。

现在句子在屏幕上占 2½行，可阅读它的全部内容。

19.4.3 选择文本

有许多场合需要选择文本。例如，可能需要选择一块文本进行移动、拷贝或打印。可选择行的一部分；或者如果想选择多行文本，必须选择整行。

可用鼠标拖动光标来选择文本。从左到右拖动鼠标光标可选择行的一部分；如果上下拖动，可选择整行。如果要用键盘选择文本，移动光标时要按住 Shift 键。要取消选择的文本，单击任何地方即可，或放开 Shift 键后移动光标或按 Esc。

试着在创建的文件中选择文本：

1. 选 Ctrl+Home 移动光标到文件的开始。
2. 按住 Shift，按右箭头键直到你已选中了 Four score and seven。
   移动光标时，可看到选择高亮产生。
3. 按 Esc 退出选择。
   光标保留在词 Seven 的末尾。
4. 按住 Shift，按下箭头键一次选择开始的两行。
   两行作为一个整体被选中。
5. 按 Esc 退出选择。

如果配有鼠标，使用鼠标重复上面的练习。

19.4.4 删除文本

有几种方法可用来删除文本。表 19.2 显示了所能使用的不同的键。记住，如果按住一个键，它会重复执行功能从而删除一串文本。

表 19.2 删 除 键

标准键	WordStar 键	功能
Backspace	Ctrl+H	向左删一个字符
Delete	Ctrl+G	删一个字符(或删所选块)
(none)	Ctrl+T	删除当前词的剩余部分
Shift+Tab	(none)	删除所选行的领头空格

试删除一些文本：

1. 从第一行选择 score and seven。
2. 按 Delete 删除所选的文本。
   第一行现在是 Four years ago...。
3. 把光标放在词 conceived 的开头。
4. 按住 Delete，直到删完 conceived in liberty and(如果删多了，也不用担心)。
   第二行现在是 a new nation, dedicated to the proposition...。

19.4.5 剪切和粘贴

剪切将文本块从文件中删除，并把它保存在一个称为 Clipboard(剪贴板)的特殊内存区

域中。拷贝文本块到 Clipboard，是将块保存到 Clipboard 但不把它从当前位置删掉。你不能看见 Clipboard，但它总是包含剪切或拷贝的最后内容。可把 Clipboard 中的内容粘贴到文本任何地方。表 19.3 显示了剪切和拷贝块到 Clipboard 以及从 Clipboard 粘贴的操作键。如果喜欢用鼠标的话，可从 Edit 菜单执行这些操作。

表 19.3　Clipboard 键

键	功能
Shift+Del	剪切块到 Clipboard
Ctrl+Y	剪切当前行到 Clipboard
Ctrl+Ins	拷贝块到 Clipboard
Ctrl+Q,Y	拷贝从光标到文件末尾的内容到 Clipboard
Shift+Ins	从 Clipboard 粘贴块

如何在 DOS 的 Editor 中拷贝一个块到另一个地方

1. 选择块。
2. 选中 Edit Copy 或按 Ctrl+Ins。
3. 移动光标到目的地。
4. 选中 Edit Paste 或按 Shift+Ins。
5. 重复 3,4，拷贝块到合适的地方。

如何在 DOS 的 Editor 中移动一个文本块到另一个地方

1. 选择块。
2. 选中 Edit,Cut 或按 Shift+Del。
3. 移动光标到目的地。
4. 选中 Edit,Paste 或按 Shift+Ins。

试着在创建的文件中移动一些文本：

1. 选择 a new nation。包括 a 前的空格，以使它也被移动。
2. 按 Shift+Del 或选中 Edit,Cut 拷贝文本到 Clipboard。
   块从文件中消失。
3. 把光标放在 on this continent 前的空格处。
4. 按 Shift+Ins 或选中 Edit,Paste 从 Clipboard 插入块到文件中。
   现在第一行成为 our fathers brought forth a new nation on this continent...。
5. 向下移动光标到第一个空行的开头。
6. 粘贴文本两次。
   现在第四行成为 a new nation a new nation。

当移动和拷贝文本时，应考虑它周围的空格，一般情况下应包含其前后的空格，这取决于老位置和新位置的具体情况。在前面的例子中，选择前面的空格是因为这个词组后跟着一

个逗号。这样，当把光标放在适当的位置插入块时，也就会有一个空格跟在其后了。有时如果不能正确处理空格，则要在粘贴后对块进行调整。（在这个例子中必须修整当前第二行开始的逗号）。

19.5 搜寻和取代

假设使用 DOS Editor 创建了一个 19 页的 README. TXT 文档，完成后却发现使用 block 的地方应用 handle，要改正错误，可能需要多处改变。DOS Editor 的搜寻取代功能可自动做出修改。你要做的只是在 Change 对话框中填入适当的文本（图 19.3）。

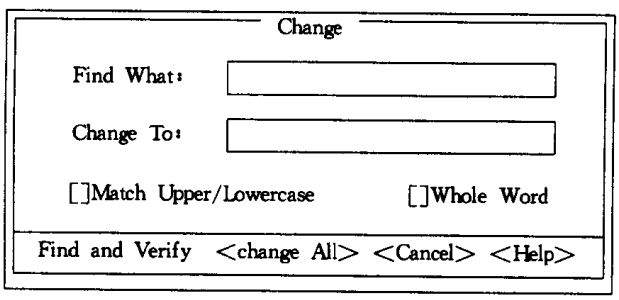

图 19.3 Change 对话框

可在 Find What 框中键入要被取代的串，然后在 Change To 框中键入取代串。

提示： 如果要删除一串而不是取代它，让 Change To 框为空即可。

试改变 nation 为 country：

1. 按 Alt+S 或单击 Search 拉下 Search 菜单。
2. 按 C 或单击 Change 选择 Change 命令。
   打开 Change 对话框。
3. 在 Find What 框中键入 nation。
4. 按 Tab 移动光标到 Change To 框，或单击 Change To 框。
5. 在 Change To 框中键入 country。
6. 按 Alt+C 或单击 Change All 选中 Change All 按钮。
   EDIT 将每个出现的 nation 改成 country。一个消息框告诉你修改完成。
7. 按 Enter 或单击 OK 关闭消息框。

如果没有选中 Match Upper/Lowercase，DOS Editor 在搜寻串时忽略大小写。也就是说，BLOCK、Block 和 block 被视为相同。但 DOS Editor 在取代串时只用在 Change To 框中指定的串，包括大小写形式。因此，如果想改变 BLOCK 为 HANDLE，Block 为 Handle，block 为 handle，必须使用三个独立的 Change 命令。

一旦选中 Match Upper/Lowercase，DOS Editor 只搜寻那些与 Find What 框中大小写一致的串。假设你想将所有的 DOSKEY 改成 Doskey，但想保留 DOSkey，必须选中 Match Upper/Lowercase 框以防止 DOSkey 也被改变。图 19.4 显示了如何为取代任务设置对话框。

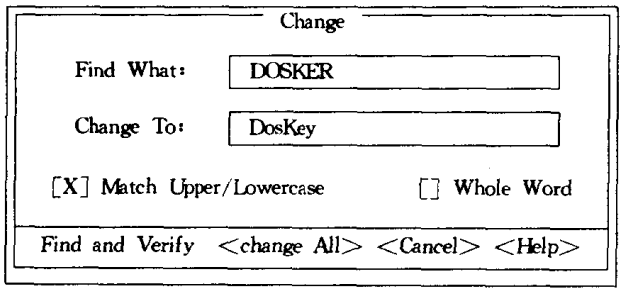

图 19.4　改变 DOSKEY 为 DosKey

Whole Word 选项也可能是必要的。例如，当你将 new 改成 country 时，你可能不想将 newly 改成 countryly 或 knew 改成 kcountry。如果选中 Whole Word，DOS Editor 只搜寻那些周围有空格或标点符号的 new。

填好框后，必须选择一个命令按钮启动取代操作。Find and Verify 将显示每个被找到的串，并让使用者自己决定是否取代它。你可能称它为笨方法，事实上，它慢但准确。Change All 不需要确认就可全部取代。这是很快的方法，但它可能做一些奇怪的取代，特别是在 Whole Word 未选中时。

提示： 避免在搜寻串中使用空格。一个包含空格的词组可能在资料中一行的结束处被分开，这样 DOS Editor 就不能找到它。

19.6　无替换搜寻

假设你不想取代 new，而只是想看一看。这时可以使用 Search Find 命令，它打开的对话框如图 19.5 所示。Find What 框和两个选项框与 Chang 对话框完全相同。当选择 OK 时，DOS Editor 寻找下一匹配块，并将其标志为一个块，然后结束搜寻。完成对这个串的搜寻后，按 F3(或选择 Edit，Repeat Find)搜寻下一个匹配块。

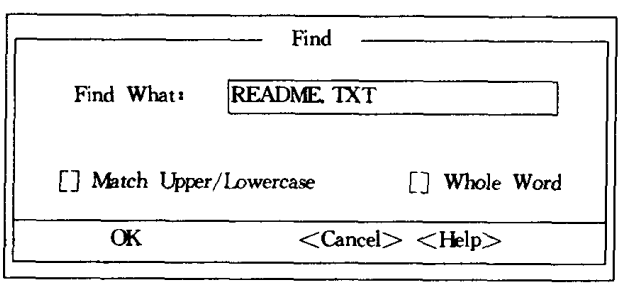

图 19.5　Find 对话框

试在你的文件中找到 country：

1. 选中 Search，Find 命令。

   Find 对话框打开，一个缺省值出现在 Find What 框中。

2. 在 Find What 框中删除缺省值，键入 country。

3. 按 Enter 或单击 OK。

EDIT 高亮显示下一个出现的 country，并把光标放在字母 C 下。

4. 按 F3。

EDIT 高亮显示下一个出现的 country。

5. 按 F3 多次。

每次 EDIT 都使找到的词呈高亮状态。

提示： Search Find 可能是在文档中长距离移动光标的一个方便的方法。

DOS Edit 记取最后键入 Find What 和 Change To 框中的文本。下次打开对话框时，最近的 Find What 文本作为缺省值出现。可使用这个文本，或编辑取代它。当打开 Change 框时，最近的 Change To 文本作为缺省取代文本出现。

19.7 文件管理

正如所见到的那样，启动 DOS Editor 时可打开一个文件，或者可用空白编辑区域创建一个新文件。File 菜单提供了打开和存储文件的命令，以及打印文件和退出 DOS Editor 命令(如图 19.6 所示)。

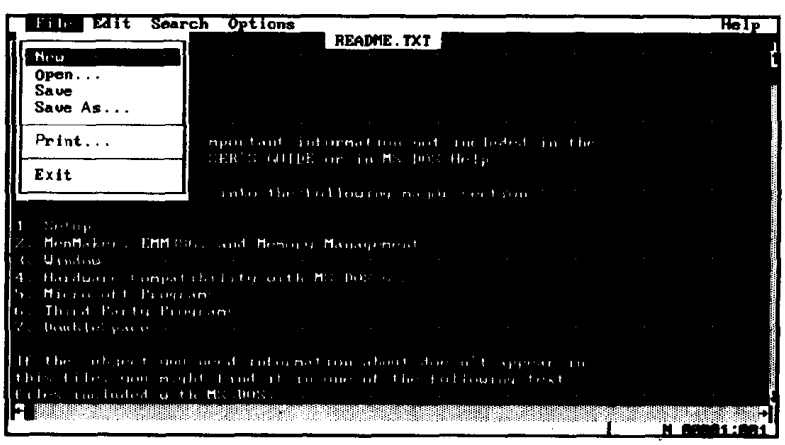

图 19.6 File 菜单

19.7.1 存储一个新文件

要存储一个从未存储过的无标题文档，可选择 File，Save 或 File，Save As...。Save As 对话框如图 19.7 所示。可选择存储文件的驱动器和目录，然后在 File Name 框中键入文件名；或者在 File Name 框中键入完整的文件名。

在图 19.7 的例子中，当前目录是 C:\DOS(见 File Name 下面的那行)。要改变驱动器，双击合适的驱动器名[-A-]、[-B-]等即可。与当前目录相关的目录出现在相同的对话框中，在这个例子中，清单中唯一的目录名是父目录(..)。如果当前目录有子目录，它们也将被列出。要改变当前目录，双击合适的目录。因为只有直接与当前目录相关的目录被列出，有时必须多次改变目录才能到达指定目录。

存储已创建的文件：

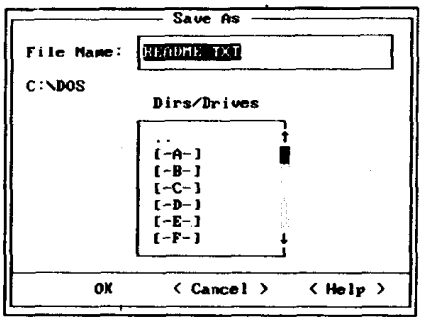

图 19.7　Save As 对话框

1. 选中 File,Save。

   打开 Save as 对话框。

2. 在 File Name 框中键入 C:\PRAC1. TXT。

3. 按 Enter 或单击 OK。

   DOS 在指定的目录中创建这个新文件。

注释： 不能在只读驱动器如 CD-ROM 驱动器中存储文件。

网络注意： 在本地驱动器上存储文件不会有什么麻烦;但能否在一个网络驱动器上存储文件,取决于对那个驱动器的访问权限。

可用新文件取代已存在的文件,只需在 File Name 框中输入这个文件的名字就行了,如果不在当前目录,要包含路径名。当 DOS Editor 询问是否想取代时,选择 Yes。

File,Save As 不仅仅可用于新文件,还可用于其他用途。例如,可打开 README. TXT,增加自己的注释,然后以 MYREADME. TXT 为名存储。README. TXT 将保持未被编辑的状态。MYREADME. TXT 就是一个新文件(除非它替代了一个已存在的文件)。

19.7.2　存储一个已经有名字的文件

File,Save 命令存储打开文件中的当前工作。例如,如果打开 AUTOEXEC. BAT,作一些改动,然后选择 File,Save,这些变化就被保存在 AUTOEXEC. BAT 中。这时不显示对话框。

作者忠告： 早存储,常存储。

如果系统死机或电源中断,只会丢失内存中的所有数据,而不会失去盘上的文件。不要等到完成整个文件后再存储,只要一开始创建文件,就要想着存储,然后每工作一小段时间就存储一次。我在文件的每一段后进行存储,然后才停下思考。

如果在 DOS Shell 或 Windows 下,在编辑器转到其他应用程序之前应先存储文件。因为系统总有可能死机,这样就无法再回到那个文件去存储它。

19.7.3　打开文件

可按需要打开和编辑任意多的文档,但一次只能处理一个文件。选中 File,Open 打开一个已存在的文档,在对话框中,可选择驱动器、目录和文件,或者也可以只在 File Name 中键

入文件名。

如果想创建一个全新文档，选择 File，New，此时文档无标题，除非首先存储它。无论何时打开另一个新的或已存在的文档，如果在最近的存储后又修改了当前文档，DOS Editor 在打开一个新文档之前询问是否存储它。

试着打开一个新文件：

1. 选中 File，New。
2. PRAC1. TXT 文件关闭，打开一个新的无标题文档。
3. 键入 Now we are engaged in a great civil war。

试回到 PRAC1. TXT 文件(不存储当前文件)。

1. 选中 File，Open。
   出现 Open 对话框。
2. 在 File Name 框中键入 C:\PRAC1. TXT。
   一个对话框警告你前面的文件还没存储，是否现在进行存储。
3. 按 Enter 或单击 Yes。
   出现 Save As 对话框。
4. 在 File Name 中键入 C:\PRAC2. TXT。
5. 按 Enter 或单击 OK。
   PRAC2. TXT 关闭，打开 PRAC1. TXT。

注释： *如果在选中 File，Exit 时有未经存储的变动，DOS Editor 也会询问是否想存储当前文件。*

19.8 打印文件

使用 File，Print 命令可打印当前文档。试打印 PRAC1. TXT：

1. 如果你的打印机是一个本地打印机(与网络上的连网打印机相对而言)，确认它已准备好。(参见第 9 章“Shell 中的其他文件管理”中的“通过 Shell 打印”一节)。
2. 选中 File，Print。
   一个对话框询问是想打印选择的文本还是整个文档，一个点表示缺省选项。在这种情况下，整个文档是缺省选项，因为没有文本被选中。
3. 按上下箭头键，移动点表示选择该项。把点放在 Complete Document 上。
4. 按 Enter 或单击 OK 打印文本。
   如果文件没有打印，参看第 12 章“在命令提示符下工作”的“重定向命令输出”中寻求问题的解决。

19.9 联机帮助

当执行 DOS Editor 操作时，它的 Help 系统提供屏幕指导，图 19.8 显示了激活 Help 的

DOS Editor 屏幕。一个 Help 菜单或条目出现在窗口的顶部，打开的文档出现在窗口底部。你可以将 Help 条目保留在屏幕上以指导工作。

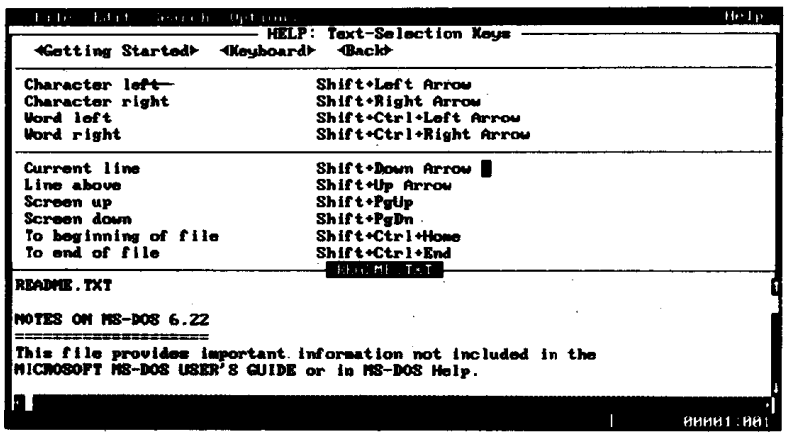

图 19.8　带有一个打开的 Help 窗口的 DOS Editor 屏幕

注释： EDIT.HLP 文件包含帮助信息。缺省情况下，DOS 首先在当前目录下寻找 EDIT.HLP，然后在程序搜寻路径下寻找。如果 EDIT.HLP 文件位于其他地方，使用 Options，Help Path 命令告诉 DOS 在哪儿可找到 EDIT.HLP。

DOS Editor 的帮助系统有三个主要部分：Keyboard，Getting Started 和 Survival Guide。Keyboard 部分解释了可在 DOS Editor 下使用的有关键，包括 Text Scrolling（文件滚动）Keys 和 Delete Keys。（图 19.8 显示的是 Text-Selection（文本选择）Keys）。Getting Started 部分是关于如何在 DOS Editor 中开始工作的内容，它包含了通常的编辑课题，如 Using a Dialog Box 和 MS-DOS Editor Options 等。Survival Guide 是关于一些最重要的操作的文档，如何拉下一个菜单、如何退出 Help 等。每个部分都提供了连接到其他两个部分的方法。

有几种方法可从 DOS Editor 获得 Help：

- 当 DOS Editor 初次启动而不打开一文档时按 Enter（而非 Esc）。
  打开 Survival Guide。
- 在 Help 菜单中选中一个命令。
- 在正在操作的命令或对话框上按 F1 可显示适当的帮助条目（如果没有关于这个命令或对话框的 Help 条目存在，则打开 Getting Started 条目）。
- 按 Shift＋F1 或在状态栏上单击 F1＝Help 可打开 Survival Guide。

试着打开 Keyboard Help：

1. 按 Alt＋H 或单击 Help 拉下 Help 菜单。
2. 按 K 或单击 Keyboard 选中 Keyboard 命令。
   Keyboard Help 库中的第一个条目被打开。

如果在一个对话框中按 F1，并没有真正进入 Help 系统，看到的只是关于这个对话框的解释，按 OK 将之关闭。但所有其他的方法都能将你带入 Help 系统，一旦到了那里，就能很容易地到达任何条目下。

高亮箭头包围的标志，如图中的<Getting Started>，是跳到其他题目的跳板，要使用这个跳板，可双击它，或在其上单击鼠标右按钮，或使用 Tab 键移到其上然后按 F1。

试跳到 Shortcut Keys 条目，要完成这个操作，可用 Tab 键移到其上后按 Enter，或在 Shortcut Keys 上单击鼠标右按钮，Shortcut Keys 条目就打开了。

状态栏显示了可在 Help 系统中使用的其他命令。按 F6 或单击 F6=Window 可在编辑窗口和 Help 窗口间切换。也可在想放光标的地方单击。按 Shift+F1 依次显示下一个 Help 条目(顺序由 Help 决定)，按 Alt+F1 在已浏览的条目清单中显示前一个条目。用这种方法可在 20 多个条目间来回转换，Back 跳板也可达到同样目的。可像打印文档一样打印一个 Help 条目。按 Esc 键可关闭 Help 窗口，回到全屏幕文档。现在关闭 Help 窗口：

1. 按 Esc 键关闭 Help 窗口。
2. 按 Alt+F，X 退出 Editor。

19.10 修改 Editor 屏幕

缺省状态下，Editor 屏幕在彩色显示器上显示蓝底白字。如果这个颜色搭配不适合于你，可选择另一种前后台的组合。选中 Options，Display，Editor 打开一个对话框，从中可选择想要的颜色。

如果需要的话，Options，Display 还可用来去掉滚动条和设置 Tab 键宽度。缺省时，Tab 键可移动 8 列，你可改变成其他数字。例如，如果设置成 10，Tab 键会移动光标到第 10、20、30 列处，依此类推。

19.11 EDIT 命令选项

当缺省设置不再满足你的工作需要时，可使用 EDIT 命令包含各种选项：

/B 如果配置的是一彩显，单色显示 Editor。

/G 用 CGA 显示器加速显示。(如果用这个选项，你的显示器出现雪花，去掉/G 重新启动 Editor)。

/H 每屏幕显示显示器可处理的最大数目行。

/NOHI 告诉 Editor 你的显示器没有高强度特点。(在 Compaq laptop 上不要使用这个选项)。

19.12 自 测 练 习

在本章中已学习了如何使用 DOS 的文本编辑器，下列练习帮助你实践使用 Editor 的使用方法：

1. 启动 Editor，开始编辑一个新文件。
2. 键入本章的简介作为新文件的文本。
3. 存储文件，名为 C:\PRAC3.TXT。

4. 打开 PRAC1. TXT，将其编辑为原来的内容来编辑它——葛底斯堡演说的第一个句子(参见本章中“行操作”一节的开头)。
5. 存储 PRAC1. TXT。
6. 在 Help 系统中查询 Delete Keys。
7. 打印 Delete Keys 项的内容。
8. 打印 PRAC1. TXT。
9. 退出 Editor。
10. 删除在本章中创建的文件：C:\PRAC1. TXT 到 C:\PRAC3. TXT。

第 20 章 基本批处理程序

简介： 要为从命令提示符下键入的命令创建自己的程序，只需加入一些特殊命令。批处理程序可大大节约时间、精力，并减少键入错误。

20.1 创建自己的程序

你经常反复使用一定数目的命令吗?可能你已厌倦了重复键入它们(有时还会产生键入错误)。前面已学习了如何用一些命令创建一个 DOSKEY 宏。保存和调回命令的另一种方法是创建一个批处理程序文件，它是一个扩展名为.BAT 的 ASCII 文本文件。

批处理程序提供了一些比宏更好的特点：

- 可在一个批处理文件中使用多个命令行。(一个 DOSKEY 宏必须放在一个命令行上)
- 可在磁盘上永久地保存批处理文件，关机或重新启动系统都不会将它们丢失
- 可把批处理文件拷给别人

批处理文件与 DOSKEY 宏相比较一个最主要的缺点是访问它的方式。因为批处理文件保存在磁盘上，DOS 必须搜寻它，像程序一样把它加载在内存中。因此，它要花较长的时间，而且 DOS 可能会找不到它。

在本章中，将学习如何：

- 创建一个批处理程序文件
- 运行一个批处理程序
- 清除命令提示符屏幕
- 注释一个批处理文件
- 通过一个批处理程序控制信息显示
- 暂时中止一个程序，直到用户按任意键继续
- 从一个批处理程序中调用另一个批处理程序
- 在一个批处理程序中使用可替换参数
- 读懂 AUTOEXEC.BAT 文件
- 在 DOS Shell 中使用批处理文件

20.2 创建一个批处理文件

一个批处理文件必须是一个带扩展名.BAT 的 ASCII 文本文件。大部分人使用 DOS 的文本编辑器创建他们的批处理文件。在文件的每行中键入一个命令。

试着创建一个批处理文件：

1. 启动编辑器，创建一个名为 C:\FINDBIN.BAT 新文档。
2. 键入下列命令：

```
DIR C:\*.BIN /S /B
```

这个命令列出硬盘中所有的.BIN 文件。

3. 存储这个文件，退出编辑器。

作者忠告： 在批处理程序中最好总使用完整的路径名。

一个批处理文件并不总运行在同一个驱动器和目录下，因此只能让每个文件名包含完整的路径以使 DOS 能找到必要的文件。例如，宁可使用 C:\DOS\README.TXT 而不使用\DOS\README.TXT。

提示： 可使用命令 DOSKEY /M> filename 从 DOSKEY 宏集中创建一个批处理文件。文件的内容如下：

```
macroname1=macrodefinition1
macroname2=macrodefinition2
```

等等。要将宏作为一个程序运行，可对文件进行编辑，删去每行的 macroname=。要使用这个文件作为定义这些宏的批处理程序，可编辑这个文件，在每行前插入 DOSKEY。

同样，可使用命令 DOSKEY /H > filename 从 DOSKEY 历史中创建一个批处理文件。你可能需要编辑文件，插入 ECHO 命令，删除不想要的命令，也可能要作一些其他变化。然后就能像编辑其他批处理程序一样编辑这个文件。

20.3　运行批处理程序

可像运行其他任何程序那样运行批处理程序，在命令提示符后键入它的文件名，可以带也可以不带扩展名。试运行 FINDBIN.BAT：

1. 键入 FINDBIN。

DOS 列出命令，紧接着是硬盘上的.BIN 文件。

20.4　定位批处理文件

当在命令提示符下键入一个命令如 FINDBIN，DOS 遵循标准的处理顺序：

1. 如果加载了处理键盘的输出的 TSR，它们先访问新键入的命令（按加载的顺序）。例如，若 DOSKEY 被加载，并且当前有一个名为 FINDBIN 的宏，DOSKEY 执行这个命令。
2. 如果没有 TSR 包括这个命令，则由 DOS 的命令处理器处理它。DOS 首先寻找名叫 FINDBIN 的内部命令。

注释： 内部命令建立在命令处理器（COMMAND.COM）内部，驻留在 DOS 内存

中。一些最常用的 DOS 命令，如 DIR 和 CD 就是内部命令。

3. 因为 FINDBIN 不是一个内部命令，DOS 下一步在当前目录中按顺序寻找名为 FINDBIN.COM，FINDBIN.EXE 或 FINDBIN.BAT 的程序。注意.BAT 文件位于这个顺序的末尾。
4. 如果 DOS 不能在当前目录找到一个 FINDBIN 程序，它将在给定的程序搜寻路径的第一个目录中寻找。如果程序文件不在这儿，DOS 就搜寻第二个目录，依此类推。在每个目录中，它首先寻找 FINDBIN.COM，然后是 FINDBIN.EXE，最后才是 FINDBIN.BAT。
5. 如果 DOS 没有找到一个 FINDBIN 程序，将显示消息 Bad command of file name，然后回到命令提示符。

可见仔细选择批处理文件名是非常重要的。要得到最好的结果，不应使 DOSKEY 宏、内部命令，或存在于搜寻路径中的.COM 或.EXE 程序重名。如果要经常使用这个程序应使它的名字尽可能简短，如 FB.BAT。或者使它易于记忆，如 FINDBIN.BAT。

提示： 为了尽量方便，可将所有的批处理文件放在一个目录中，并把这个目录放在搜寻路径靠开头的地方。

20.5 批处理文件的运行

当 DOS 运行一个批处理文件时，它按命令在文件中出现的顺序处理。对于每个命令，DOS 显示一个空行，然后显示命令提示符，填充命令，最后运行它。显示空行、命令提示符和命令称为 echoing the command（对命令作出响应）。DOS 在读文件末尾的空行时可能显示一个或多个命令提示符。最后，DOS 终止批处理文件，并显示一个命令提示符供使用。

所有这些命令和空命令提示符可能让一个没经验的用户觉得易混淆，在本章后面将学习如何删掉它们。

20.6 中止批处理文件

假设批处理文件中的一个命令显示了一个错误消息，而需要取消程序的剩余部分。可通过按中断组合键如 Ctrl+C 或 Ctrl+Break 中止一个批处理文件。中断时，批处理文件显示消息 Terminate batch job(Y/N)?，等待回答。如果按 N，程序中断当前命令，整个批处理文件结束，返回到命令提示符。一些命令可能根本不能中断，其他有的命令会立刻对中断键作出反应。有时，一个中断键只停止当前命令，需要重复取消每个命令，直到出现消息 Terminate batch job(Y/N)?。

注释： 当 DOS 的 Break 功能激活时，可快速中断程序。参看 BREAK 命令。

20.7 使用特殊的批处理命令

从命令提示符下键入的任何命令都可以放在一个批处理文件中，除 DOSKEY 宏外，

DOS 提供了批处理文件特别有用的几个命令。CLS、REM、ECHO、PAUSE 和 CALL 几乎总是用在批处理文件中，而很少使用在命令提示符下。

20.7.1　消除屏幕

CLS(Clear Screen)清除屏幕，并把光标放在显示器的左上角。这个命令不需参数。也可在命令提示符下键入 CLS 清除屏幕或隐藏工作。

试清除你的屏幕：

1. 键入 CLS。
   屏幕清除。

许多人常在一个批处理程序的开头使用 CLS 命令，这样在显示新消息之前可以清除原来的信息，也可以在启动程序的下一个命令之前清除前面命令的输出信息。

20.7.2　程序注释

REM 可在批处理文件中创建注释。这些注释帮助你描述一个程序的执行目的，紧随命令的选项的用途等等。试着在你的程序中增加 CLS 和一个注释，然后再运行：

1. 打开 FINDBIN.BAT 进行编辑。
2. 在文件的开始插入下列行：

```
REM This program lists all the .BIN files on your hard drive
CLS
```

3. 存储文件，退出 Editor。
4. 键入 FINDBIN 运行这个程序。
   这次，DOS 首先注释，然后在显示文件清单之前迅速清除屏幕。如果计算机运行得很快，它可能过得非常快而看不清楚。一会儿将学习如何控制显示。

注释： FINDBIN.BAT 现在如下所示：

```
REM This program lists all the .BIN files on your hard drive.
CLS
DIR C:\*.BIN /S /B
```

假设你有一个批处理文件 FRESHA.BAT，它对驱动器 A 中磁盘重做物理格式化而且不破坏文件。FRESHA.BAT 可能包含这些命令：

```
XCOPY A:\*.* C:\TEMPTEXT\ /E
FORMAT A: /U /V:""
XCOPY C:\TEMPTEXT\*.* A:\ /E
DELTREE /Y C:\TEMPTEXT
```

如果它包含一些注释，可能更容易理解。如下所示：

```
REM This program refreshes the disk on drive A
REM Prepared by D. Tabler, ext. 423

REM Copy the tree from A to temporary branch in C:
XCOPY A:\*.* C:\TEMPTEXT\ /E
```

```
REM Refresh the sectors:
FROMAT A:/U /V:""

REM Restore the tree:
XCOPY C:\TEMPTEXT\*.* A:\ /E

REM Remove the temporary branch from C:
DELTREE /Y C:\TEMPTEXT
```

REM 常用在一个批处理文件中，使命令临时失效。在命令前插入 REM 可使它成为注释，运行程序时 DOS 将忽略它。如果你的程序不能正确工作，可一次一行地将命令变成注释，直到找出问题为止。删去 REM 即可恢复命令。

20.7.3 控制信息显示

使用 ECHO 命令可打开或关闭命令响应。DOS 不能响应一个批处理文件中跟在 ECHO OFF 后面的命令。删除响应能得到更整洁的屏幕，从而不会让初学者觉得易混淆。

注意 当一个批处理文件运行结束时，响应将回到程序开始的状态。ECHO OFF 命令本身会响应，因为 DOS 在将响应关闭之前响应它。如果不想响应 ECHO OFF 命令，可将@放在其前面。这个符号删除一单个命令行的响应。

注释： 如果在命令提示符下键入 ECHO OFF，命令提示符消失，只要键入 ECHO ON 就可回到命令提示符。

如果要在程序运行时显示自己加上的信息，可使用如下格式的命令：

```
ECHO message
```

试着在 FINDBIN.BAT 程序中加入一些 ECHO 命令：

1. 打开 FINDBIN.BAT 程序中加入一些 ECHO 命令：
2. 在文件的开头键入下面的命令：

```
@ECHO OFF
```

3. 在 CLS 命令后插入下列命令：

```
ECHO * * * * * * * * * * * * * * * * * *
ECHO    A List of the .BIN Files on Drive C
ECHO * * * * * * * * * * * * * * * * * *
```

4. 存储文件，退出编辑器。
5. 运行这个文件。

这次命令不会作出响应，但在文件清单前面可看到 ECHO 消息。

注释： FINDBIN.BAT 现在如下：

```
@ECHO OFF
REM This program lists all the .BIN files on your hard drive.
CLS

ECHO * * * * * * * * * * * * * * * * * *
ECHO    A List of the .BIN Files on Drive C
```

```
ECHO * * * * * * * * * * * * * * * * * *
DIR C:\*.BIN /S /B
```

注意，这个程序使用 ECHO 和 REM 消息的方法有所不同。ECHO 既可注释程序，又可将消息显示给用户。当响应关闭时，REM 注释不再显示，因此只有在阅读程序时才能看到它。

有时可用一些空行设置关闭消息，键入 ECHO. 即可显示一个空行，要保证句点紧跟在 ECHO 后面，没有空格。也可使用空格将消息放在屏幕中间，显示星号强调一个消息，或者甚至在消息周围画一个框使它在屏幕上更为突出。组合这些技术，可创建一个批处理文件如下：

1. 在每个显示消息的 ECHO 后面键入 ECHO。
2. 运行这个程序。

每个消息行后将有一个空行。

注释： FINDBIN.BAT 现在如下：

```
@ECHO OFF
REM This program lists all the .BIN files on your hard drive.
CLS
ECHO * * * * * * * * * * * * * * * * * * *
ECHO.
ECHO A List of the .BIN Files on Drive C
ECHO.
ECHO * * * * * * * * * * * * * * * * * * *
ECHO.
DIR C:\*.BIN /S /B
```

20.7.4　暂停程序

PAUSE 命令，没有任何参数，可暂时停止批处理程序的进程，直到按任一键继续。一些程序有内建的暂停，例如 FORMAT 命令总是在开始格式化之前提供一个改变主意的机会；XCOPY 有一个特殊的/W (wait)选项，可用它在拷贝开始之前产生一个暂停，在针对软盘的批处理程序中，这个选项提供的改变软盘的机会是很有用的。尽管如此，许多命令还是立刻开始工作。如果需要有机会改变磁盘，放纸到打印机，在消息卷离屏幕之前阅读它们，或者决定是否继续程序，一个 PAUSE 命令可提供上述功能。

让我们将 FINDBIN.BAT 改成 FBINSYS.BAT，用它来寻找驱动器上的.BIN 和.SYS 文件：

1. 对 FINDBIN.BAT 作一个拷贝，命令为 FBINSYS.BAT。
2. 打开 FBINSYS.BAT 进行编辑。
3. 在结尾处增加下列行(提示：拷贝前面的行进行编辑，用 SYS 代替 BIN)。

```
ECHO.
ECHO Are you ready to see the .SYS files?
PAUSE
ECHO * * * * * * * * * * * * * * * * * * *
ECHO.
ECHO A List of the .SYS Files on Drive C
```

```
ECHO.
ECHO * * * * * * * * * * * * * * * * * * * *
ECHO.
DIR C:\*. SYS /B /S /P
```

4. 存储文件,退出编辑器。
5. 运行 FBINSYS。

这次,DOS 显示. BIN 文件,然后询问是否准备好看. SYS 文件,等待用户回答。

6. 按任意键。

DOS 逐个列出. SYS 文件。

注释: FBINSYS. BAT 现在如下:

```
@ECHO OFF
REM This program lists all the . BIN files on your hard drive.
CLS
ECHO * * * * * * * * * * * * * * * * * * * * * * * *
ECHO.
ECHO   A List of the . BIN Files on Drive C
ECHO. ECHO * * * * * * * * * * * * * * * * * * * *
ECHO.
DIR C:\*. BIN /S /B
ECHO.
ECHO Are you ready to see the . SYS files?
PAUSE
ECHO * * * * * * * * * * * * * * * * * * * * * * * *
ECHO.
ECHO A List of the . SYS Files on Drive
ECHO.
ECHO * * * * * * * * * * * * * * * * * * * * * * * *
ECHO.
DIR C\*. SYS /B /S /P
```

注释: 可在一个批处理文件中对输出结果重新定向。例如,可在前面的文件中将 DIR 输出重定向到打印机。但不能重定向批处理文件本身的输出。如果键入命令 FBINSYS >PRN,DOS 将忽略重定向。

20.7.5 调用其他批处理程序

可在一个批处理程序中启动另一个批处理文件。例如,与其在 FBINSYS. BAT 中包含 FINDBIN. BAT 的命令,不如直接调用 FINDBIN. BAT 试一试:

1. 打开 FBINSYS. BAT 进行编辑。
2. 删除第三行(CLS)到第九行(DIR)。
3. 在第三行(REM 后)插入到第九行(DIR)。

```
FINDBIN
```

4. 存储文件,退出编辑器。
5. 运行 FBINSYS。

DOS 显示 *.BIN 文件，然后结束。不会询问是否准备看 *.SYS 文件。

注释： FBINSYS.BAT 文件如下所示：

```
@ECHO OFF
REM This program lists all the .BIN files on your hard drive.
FINDBIN
ECHO.
ECHO Are you ready to see the .SYS files?
PAUSE
ECHO * * * * * * * * * * * * * * * * * * *
ECHO.
ECHO   A List of the .SYS Files on Drive C
ECHO.
ECHO * * * * * * * * * * * * * * * * * * *
ECHO.
DIR C\*.SYS /B /S /P
```

为什么 DOS 不完成 FBINSYS 程序呢？这是因为当你从一个批处理文件运行另一个批处理文件时，DOS 转到第二个程序后再也不回到第一个程序。在此例中，DOS 处理 FINDBIN 时，它启动这个程序，返回命令提示符，剩下的命令不再运行。

可使用 CALL 命令运行另一个批处理文件，而不完全转到其上。当调用一个批处理文件时，DOS 完成被调用的程序后回到最初的程序。试在 FBINSYS.BAT 中用 CALL 运行 FINDBIN。

1. 打开 FBINSYS.BAT 进行编辑。
2. 在 FINDBIN 命令前插入 CALL。
3. 存储文件退出编辑器。
4. 运行 FBINSYS。

这次，整个程序顺利运行完毕。

注释： FBINSYS.BAT 文件如下所示：

```
@ECHO OFF
REM This program lists all the .BIN files on your hard drive.
CALL FINDBIN
ECHO.
ECHO Are you ready to see the .SYS files?
PAUSE
ECHO * * * * * * * * * * * * * * * * * * *
ECHO.
ECHO   A List of the .SYS Files on Drive C
ECHO.
ECHO * * * * * * * * * * * * * * * * * * *
ECHO.
DIR C\*.SYS /B /S /P
```

20.8　使用可替换参数

在第 13 章“DOSKEY”中，学习了如何使用可替换参数使宏更为灵活。批处理文件也可

使用可替换参数,但它们使用百分号而不是美元符。

一个批处理程序最多可有 10 个可替换参数,从 0%到 9%。0%代表命令中的第一个词,它总是代表这个批处理文件的名字。(如果单独使用 0%,它常作为 ECHO 消息的一部分)。

让我们把 FINDBIN 变成 FINDFILE,运行这个文件时可输入想要的文件名):

1. 对 FINDBIN. BAT 作一个拷贝,命名为 FINDFILE. BAT。(确认拷贝的是 FINDBIN. BAT,而不是 FBINSYS. BAT)。
2. 打开 FINDFILE. BAT 进行编辑。
3. 在 DIR 命令中,将 *. BIN 改成%1,在命令结尾处加入/P,这样长清单会自动分页。DIR 命令如下所示:

```
DIR C:\1% /S /B /P
```

4. 编辑 REM 命令为:

```
REM This program lists all the %1 files on your hard drive
```

5. 编辑第 3 个 ECHO 命令为:

```
ECHO A List of the %1 Files on Drive C
```

6. 存储文件,退出编辑器。
7. 试运行新程序,键入 FINDFILE COMMAND. COM。
   DOS 列出硬盘上所有名为 COMMAND. COM 的文件。
8. 用下面文件名再试几次:

```
F*.COM
*.HLP
DOSSHELL.*
```

9. 如果键入 FINDFILE 不带文件名,将发生什么呢? 不妨一试。
   DOS 列出硬盘上的所有文件,一页接一页。不要忘记可按 Ctrl+Break 杀掉这个程序。

注释: FINDFILE. BAT 如下所示:

```
ECHO OFF
REM This program lists all the %1 files on your hard drive.
ECHO * * * * * * * * * * * * * * * * * * * * * * * *
ECHO.
ECHO A List of the %1 Files on Drive C
ECHO.
ECHO * * * * * * * * * * * * * * * * * * * * * * * *
ECHO.
DIR C\%1 /B /S /P
```

当你键入一个命令如 FINDFILE COMMAND. COM 时,DOS 用参数 COMMAND. COM 代替程序中任何地方出现的 1%。这样,实际运行的命令如下:

```
@ECHO OFF
REM This program lists all the COMMAND. COM files on your hard drive.
```

```
ECHO * * * * * * * * * * * * * * * * * * * * * *
ECHO.
ECHO A List of the COMMAND.COM Files on Drive C
ECHO.
ECHO * * * * * * * * * * * * * * * * * * * * * *
ECHO.
DIR C:\COMMAND.COM /B /S /P
```

可见将列出驱动器 C 上的所有 COMMAND.COM 文件。

这样 FINDFILE.BAT 变得更为灵活，因而也就比 FINDBIN.BAT 或 FBINSYS.BAT 用途更广。现在花点时间删除 FINDBIN.BAT 和 FBINSYS.BAT。可保留 FINDFILE.BAT，因为它可帮你找到硬盘上的任何文件。

20.9　AUTOEXEC.BAT 文件

几乎每个系统都有一个 AUTOEXEC.BAT 文件。当 DOS 启动或重启动时，它在启动驱动器的根目录中寻找这个文件。如果 AUTOEXEC.BAT 在那里，它在 CONFIG.SYS 之后运行，然后才出现命令提示符。

注释： 如果没有 AUTOEXEC.BAT，DOS 在启动时会询问日期和时间。AUTOEXEC 取消了日期和时间功能。如果你的系统无时钟功能，可在 AUTOEXEC.BAT 中包含 DATE 和 TIME 命令，这样在启动过程中可设置正确的日期和时间。

在安装过程中，DOS 6 放了几个命令到你的 AUTOEXEC.BAT 文件中。大部分使用 DOSKEY 的人把它作为 AUTOEXEC.BAT 中的一个命令进行启动。一个非常基本的 AUTOEXEC.BAT 文件如下所示：

```
@ECHO OFF
PATH C:\; C:\DOS
DOSKEY
```

大部分 AUTOEXEC.BAT 文件比这复杂得多。比如可从 AUTOEXEC 中包含启动防病毒和取消删除的程序命令，参见第 25 章“DOS 的防病毒程序”和第 26 章“UNDELETE 的使用”。

提示： 可在 AUTOEXEC.BAT 文件中加入 REM、ECHO、CLS 和 PAUSE 命令，使它使用起来更方便。

看一看你的 AUTOEXEC.BAT 文件：

1. 打开 C:\AUTOEXEC.BAT 进行编辑。
   现在应该理解文件中的 REM、CLS、PAUSE 和 ECHO 命令了。可能还会有不认识的命令，注意不要改变你不理解的命令，它们对你的系统的运行可能至关重要。
2. 重新浏览 PATH 命令。看它是否包含你想要的目录到搜寻路径中，或是否按使用频度顺序？如果不是，编辑或替换命令。但不要从搜寻路径中删去任何目录，除非确信不再需要了。(PATH 已在第 12 章“在命令提示符下工作”中作了详细解释)。
3. 如果想使用一个与现在不同的命令提示符，插入或替换 PROMPT 命令(命令

PROWPT 在第 12 章中解释过)。

4. 如果在引导时想自动启动 DOS Shell 或 Windows,在文件中的最后一行插入 DOSSHELL 或 WIN(注意,把该行放在程序的最后是很重要的),包括在命令提示符状态下启动该程序、所用的参数和选项。
5. 保存文件,退出编译器,重新引导以便 DOS 执行新的 AUTOEXEC.BAT 文件。

预览: 现在已经掌握了批处理文件的基本知识,加入其他命令可以使批处理文件更加灵活。你可以在批处理文件中作出这样的决定:对一组命令重复执行几次,向用户显示菜单等。第 37 章"批处理程序中的编程逻辑"解释了批处理文件的高级功能。

20.10 在 DOS Shell 中使用批处理文件

用户可以从 DOS Shell 中启动批处理文件,如果在文件清单中打开一个.BAT 项,该程序即被执行,但你却没有机会把可替换变量的参数包括进去。例如,在文件清单中双击 FINDFILE.BAT 文件。将显示磁盘驱动器上的所有的清单。当然,在 Shell 中选择 View,All Files 菜单项是一个更好的方法,因为通过它可以从文件清单中选取并处理文件。

如果使用 File,Run 菜单项来执行批处理程序,可以把命令行的参数包括进去,而且,如果你创建了一些在 DOS Shell 中经常用到的批处理文件,可以给它们定义相应的程序项。

20.11 自 测 练 习

在本章已经学习了编写批处理文件的基本知识,接下来的练习将有助于自测对它们的掌握程度:

1. 编写名字为 PRINT 下的批处理程序,用它打印驱动器%2 上的所有文件%1(提示:重新定向 DIR 命令的输出到打印机。用 ECHO 打印输出内容的标题,并在监视器上显示信息)。
2. 编写名字为 VAULT 的批处理程序,用它把所有%1 文件拷贝到你的最大容量软驱中的软盘(提示:使用 XCOPY 命令拷贝所有分枝的文件,需要的话在目标驱动器上建立目录。在程序运行期间,使用 ECHO 在屏幕上显示信息)。

第21章 系统配置

简介： CONFIG.SYS 文件对于系统来说是非常重要的，但许多 DOS 用户并不清楚其中命令的含义。本章将对其中的一些作出解释。

21.1 DOS的启动

当启动或重启动 DOS 时，它将其核心程序加载到内存中。然后它在引导驱动器的根目录中寻找一名为 CONFIG.SYS 的文件。CONFIG.SYS 中的命令帮助 DOS 设置或配置系统。大多数命令与硬件(尤其是内存)的定义和管理有关。以下是一些常用命令的功能：

- 决定为特定功能保留多少内存
- 决定使用哪个命令处理器
- 决定加载哪个设备驱动程序

执行完 CONFIG.SYS 中的所有命令之后，DOS 寻找并运行 AUTOEXEC.BAT。然后 DOS 将控制权移交给用户，或者是在命令提示符下，或者是在诸如 DOS Shell 或 Windows 的程序中。

本章将讲述：

- 保护启动文件(CONFIG.SYS 和 AUTOEXEC.BAT)
- 在启动出问题的情况下跳过启动文件
- 理解 CONFIG.SYS 文件的结构和内容
- 理解一般 DOS 设备驱动程序(SETVER.EXE、ANSI.SYS、HIMEM.SYS 和 EGA.SYS)的功能。
- 解释以下 CONFIG.SYS 命令

```
DEVICE
DOS=HIGH
BUFFERS
FCBS
FILES
SHELL
STACKS
```

21.2 启动文件的保护

你也许曾经对启动文件(CONFIG.SYS 和 AUTOEXEC.BAT)进行过编辑。这两个文件对于系统来说非常重要，用户应采取额外的措施来保护它们。作者建议在对它们进行任何改动之前，先进行备份拷贝。

启动文件所处的位置取决于是否使用磁盘压缩程序，以及如果是的话，它是否作为 DOS 核心程序的一部分被加载。在下列情况下，你的启动文件将位于引导驱动器(一般是驱动器 C)的根目录中：

- 未在引导驱动器中使用数据压缩程序
- 使用 DblSpace 或 DrvSpace 来压缩驱动器

如果引导驱动器是用 Stacker 3.0 及更早版本，或其他第三方压缩程序压缩的，DOS 将从宿主驱动器而不是压缩驱动器的根目录进行启动。在这种情况下，在压缩驱动器中也许也有 CONFIG.SYS 和 AUTOEXEC.BAT 的拷贝，因此必须确保已对宿主驱动器根目录中的启动程序进行了编辑。

如果引导驱动器是用 Stacker 3.1 及更新版本压缩的，并且使用的是 DOS 6 及以后版本，启动文件的位置取决于 Stacker 是随 DOS 核心程序一起加载的，还是从 CONFIG.SYS 加载的。要知道这一点，只需键入命令：

```
STACKER
```

随后将显示如下信息：

```
Stacker drive map:
    Drive A was drive A at boot time
    Drive B was drive B at boot time
    Drive C was drive C at boot time [E:\STACVOL.DSK=327.6MB]
    Drive D was drive D at boot time
    Drive E was drive E at boot time
```

如同例子中的那样，假定驱动器 C 为压缩驱动器，并且是从驱动器 C 启动的，若驱动器图显示 Drive C was drive C at boot time，则表示启动文件就在驱动器 C 的根目录中。如果驱动器图显示 Drive C was drive x at boot time，则表示系统实际上是从驱动器 x 启动的，而这也正是启动文件所在的位置。应保持并编辑 x:\CONFIG.SYS 和 x:\AUTOEXEC.BAT。

请现在就对这两个文件进行备份拷贝：

1. 确认系统的实际启动驱动器。
2. 切换到此驱动器的根目录。
3. 键入 XCOPY CONFIG.SYS CONFIG.SAV 备份 CONFIG.SYS 文件。
4. 键入 XCOPY AUTOEXEC.BAT AUTOEXEC.SAV 备份 AUTOEXEC.BAT 文件。

注释： *在对硬盘进行备份时，应包括这两个启动文件。DOS 的备份功能将在第 23 章“MSBACKUP 和 MWBACKUP”中讲述。*

对启动文件进行编辑后，应重新启动试验所作的变动。重启动迫使 DOS 读取并执行这两个启动文件。对 AUTOEXEC.BAT 和 CONFIG.SYS 所作的改动直到重启动时，才能发挥效用。

即使新配置看起来一切正常，也应将备份文件再保存几天，直到能够确定新设置更好些。如果决定返回到先前的启动文件，可使用 COPY 或 XCOPY 将当前文件替换为先前的文件。比如要恢复 AUTOEXEC.BAT，可键入如下命令：

COPY AUTOEXEC.SAV AUTOEXEC.BAT

21.3 跳过启动文件

对启动配置进行改动并不总是容易的事。有时一个微小的看起来无足轻重的变化会使配置的其他部分不能正常工作，以至于使DOS无法启动。问题的关键在于：如果不能启动，就无法打开编辑器，也就无法解决问题。不过倒是不一定非要使用紧急启动软盘，DOS提供了走出困境的另一种方案。

如果在改动了一个启动文件后，无法启动，可重新启动并在显示"Starting MS-DOS..."信息时，按下F8键。此键请求DOS逐行执行启动文件，从而使得用户可跳过文件中的任何一个命令。在DOS 6.0中，CONFIG.SYS逐行执行，而AUTOEXEC.BAT正常执行；在DOS 6.2及更新版本中，这两个启动文件都可逐行执行。

注释： DOS显示Starting MS-DOS...信息有好几秒钟时间，足够注意到这条信息并去按F8键了。

逐行执行启动文件有什么好处呢?这样就能知道哪些行是改动过的，可能的问题又是什么，从而可跳过这些行。也许最终只对系统的一部分进行了设置，但对于访问编辑器系统来说往往已经足够，这样就可以改动CONFIG.SYS和AUTOEXEC.BAT了。如果还是不能访问编辑器，至少可以用COPY命令将CONFIG.SAV拷贝为CONFIG.SYS，将AUTOEXEC.SAV拷贝为AUTOEXEC.BAT，即恢复启动文件的先前版本。COPY命令是一内部命令，一旦DOS可用，它就可用。恢复了启动文件的先前版本后就可以重新启动回到早先的系统之中。

试着逐行执行启动文件：

1. 退出正在运行的所有程序，回到初始命令提示符。
2. 按Ctrl+Alt+Delete重新启动。
3. 当见到信息Starting MS-DOS...时，按F8。

   可看到信息MS-DOS will prompt you to confirm each CONFIG.SYS command (MS-DOS将提示你确认每条CONFIG.SYS命令)。下一行，将见到第一条命令，跟着[Y,N]?。
4. 对每条命令按Y。

   有些命令在启动其对应的程序时，在屏幕上产生一些信息。如果在启动系统时遇到麻烦，应仔细检查每一条信息以寻找问题的迹象。在每条信息之后，DOS显示CONFIG.SYS的下一条命令，并询问[Y,N]?。

   处理完CONFIG.SYS后，DOS询问Process AUTOEXEC.BAT [Y,N]? 若按N，DOS跳过整个AUTOEXEC.BAT；若按Y，DOS 6.0按正常情况执行AUTOEXEC.BAT，而DOS 6.2则逐行执行AUTOEXEC.BAT。
5. 如果使用的是DOS 6.2及以后版本，对AUTOEXEC.BAT的每条命令按Y。

   最后将到达文件末尾并回到正常命令提示符下。

如果不能用F8启动系统，DOS还提供了一种救急措施：在见到Starting MS-DOS...信

息时，按下 F5 而不是 F8。此操作跳过整个启动文件，提供了一个非常初始的系统。许多设备驱动程序都未加载，因此可能无法使用鼠标或打印机。但是，可用 COPY 命令恢复先前的启动文件。然后，就可以重新启动并再次尝试了。

试着不经启动文件启动系统：

1. 退出所有正在运行的程序，回到初始命令提示符。
2. 按 Ctrl+Alt+Delete 重新启动。
3. 当见到信息 Starting MS-DOS...时，按 F5。
   可见到信息 MS-DOS is bypassing your CONFIG. SYS and AUTOEXEC. BAT files (MS-DOS 正在跳过你的 CONFIG. SYS 和 AUTOEXEC. BAT 文件)。然后可见到版本号和命令提示符。
4. 重启动以恢复正常配置。

注释： 跳过启动文件后，DOS 建立一缺省命令提示符 PG，以及一缺省搜寻路径 C:\DOS，这样就可以方便地访问 DOS 功能(如编辑器)。

21.4 CONFIG. SYS 命令的含义

CONFIG. SYS 是一个看起来有点像批处理文件的 ASCII 文本文件，它包含特定的配置命令。这其中的大多数命令只适用于 CONFIG. SYS，不能在其他地方使用。另一方面，在 CONFIG. SYS 中也不能使用通常的命令，如 DIR、FORMAT 等。

如同批处理文件一样，可使用 ASCII 编辑器创建或修改 CONFIG. SYS。与批处理文件不同的是，不能从命令提示符下运行 CONFIG. SYS。CONFIG. SYS 只能用于启动期间。

注释： CONFIG 运行在建立搜寻路径和改变当前目录之前。对于 CONFIG. SYS 中的那些不在引导驱动器根目录中的文件和程序，必须包括完整的路径名。

以下是 CONFIG. SYS 文件的一个示例：

```
DEVICE=C:\DOS\SETVER.EXE
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\ANSI.SYS
DOS=HIGH
BUFFERS=50
FILES=30
SHELL=C:\DOS\COMMAND.COM C:\DOS /P
STACKS=9,256
```

此例中包括六种不同类型的命令：DEVICE、DOS、BUFFERS、FILES、SHELL 和 STACKS。这些命令只能用于 CONFIG. SYS 中　注意普遍的句法规则：命令后跟着一等号(=)。等号的右边为文件名、参数和/或选项。

21.5 加载设备驱动程序

DOS 包括控制系统基本硬件单元的内置软件：键盘、磁盘驱动器、1MB 限以下的内存、

点阵打印机，等等。但是如果另有超出此基本配置的设备，如鼠标或扩充内存板，也许就会需要加载设备驱动程序来控制它们。也可加载一设备驱动程序对通常设备提供额外的功能，比如 ANSI.SYS 可为 DOS 命令提示符添上色彩。

某些设备驱动程序可仿真并不真正存在的设备。比如 RAMDRIVE.SYS 驱动程序可使用部分内存来仿真一硬盘；DRIVE.SYS 驱动程序使得一个软盘驱动器看起来像是两个。

DOS 6 包括几个可选的设备驱动程序，有关这些程序的详细描述参见附录 A“设备驱动程序”，它们中的一部分将在本章及后续章节中描述。另外，还可能有随硬件而来的设备驱动程序。还可购买比 DOS 的基本驱动程序功能更强的设备驱动程序。比如市场上的许多内存管理程序就有着 DOS 所无法提供的功能。

在 CONFIG.SYS 中，用 DEVICE 命令加载一设备驱动程序。此程序加载之后，就保存在内存之中，直到在 CONFIG.SYS 中删去此命令并重新启动系统。

21.6 DEVICE 命令的含义

DEVICE 命令的普通句法如下：

```
DEVICE=filespec [parameters] [switches]
```

DEVICE 命令必须包括设备驱动程序的文件名，然后是所有需要的参数和选项。记住如果文件不在引导驱动器的根目录中，文件名必须包括驱动器和路径名。

本章前面的 CONFIG.SYS 例子中加载了三个设备驱动程序：SETVER.EXE、HIMEM.SYS 和 ANSI.SYS。下面各节将分别介绍这些驱动程序。

21.6.1 SETVER 的使用

某些程序是专为 DOS 的某一特定版本而设计的。许多程序使用的技术在 DOS 早期版本中是得不到的。比如，为 DOS 3 所写的程序可能被设计得无法在其他任何 DOS 版本下运行。它可能会显示一出错信息，比如 Incorrect DOS Version (DOS 版本不正确)，然后退出。DOS 6 包括了 DOS 3 的大部分功能。需要较早版本 DOS 的程序在 DOS 6 下运行，往往不会出现什么问题。

SETVER 驱动程序(称为 SETVER.EXE)所包括的版本表格列出了这样一些程序所需要的版本。当这些程序的其中之一检查当前版本序号时，SETVER 执行操作并报告程序所需版本。你的 CONFIG.SYS 也许包括一 DEVICE 命令加载 SETVER.EXE，它是由 DOS 6 的 SETUP 程序安放在那儿的。要在 SETVER 表中添加一程序，可按以下句法键入一命令。

```
SETVER filename.ext version
```

比如，命令 SETVER PLACER.EXE 5.0 将 DOS 版本序号 5.0 报告给名为 PLACER.EXE 的程序。此命令在 SETVER 表中添加了一新入口。必须重启动以使更新后的表发挥效用。

21.6.2 HIMEM.SYS 的使用

如果系统中包括扩展内存，则需要加载一设备驱动程序来对它进行管理。HIMEM.SYS

是 DOS 6(及 Windows)提供的扩展内存管理程序。DOS 6 的 SETUP 程序可在 CONFIG.SYS 文件中放入一个 DEVICE 命令以加载 HIMEM.SYS。第 29 章"内存管理"在讲述如何使用扩展内存的同时讲述了如何使用和控制 HIMEM.SYS。如果 DOS 和 Windows 中都有 HIMEM.SYS,应选择最新版本的那一个。

21.6.3 ANSI.SYS 的使用

ANSI.SYS 设备驱动程序向监视器和键盘提供了一系列扩展功能。比如 ANSI.SYS 能使程序在命令提示符屏幕上显示色彩和图形,它还能重新定义键盘上各个键的含义。许多应用程序需要这些功能,离开了 ANSI.SYS 就无法正常工作。甚至某些 DOS 程序也需要 ANSI.SYS。第 48 章"控制台的控制"将讲述如何利用 ANSI.SYS 功能。

21.6.4 其他设备驱动程序

根据硬件的不同,系统可能还将加载其他几个设备驱动程序。比如,声卡、CD-ROM 驱动器、扫描仪、Fax/调制解调器、网络卡等都需要驱动程序。DOS 未提供这些驱动程序,它们是随硬件一起销售的具有版权的软件。用户可能需要根据硬件用户手册上的指示,自己在 CONFIG.SYS 中插入必要的 DEVICE 命令;或者硬件所带的设备程序将自动在 CONFIG.SYS 中插入适当的命令(应对 CONFIG.SYS 作一备份拷贝以防万一)。

如果使用数据压缩,则可能需要加载一个或多个驱动程序以支持数据压缩软件。这些驱动程序由压缩系统提供。DOS 为 DblSpace 和 DrvSpace 提供了特定的驱动程序,诸如 Stacker 之类的第三方压缩系统也提供各自的驱动程序。安装压缩软件时,在 CONFIG.SYS 中将插入适当的 DEVICE 命令。

21.7 将 DOS 加载到高端内存

通常情况下,DOS 将其自身加载到最先的 640KB 内存中(这一区域称为常规内存)。大多数非 Windows 应用程序也使用这部分内存,因此很容易用尽这部分内存。如果系统中有扩展内存,可将 DOS 加载到扩展内存以节省常规内存空间。具体的 CONFIG.SYS 命令如下所示:

```
DOS=HIGH
```

在 CONFIG.SYS 的 DOS=HIGH 命令之前,必须有一行命令来加载 HIMEM.SYS 设备驱动程序(或等效的内存管理程序)。

在 CONFIG.SYS 中,也许还会见到 DOS=UBM 或 DOS=HIGH,UMB 命令。UMB 参数使得 DOS 将高端内存(常规内存以上的内存区)为运行程序所用。这将在第 30 章"为 386、486 和奔腾机增加的内存工具"中解释。

21.8 缓冲器的使用

当 DOS 从磁盘读取数据时,它把数据存储在缓冲器(buffer)中。缓冲器就是 DOS 保留

作此用途的一部分内存。每个缓冲器的大小与磁盘扇区一致，通常是512字节。DOS根据需要填充缓冲器，提供程序所需数据。然后，它将这些数据传送给程序。

当应用程序请求数据时，DOS首先在缓冲器中寻找，看数据是否已经在那儿了。如果找到所需数据，就不必再访问磁盘，从而省下相当数量的时间。

DOS以同样的方式将输出磁盘数据在写入磁盘之前，先存入缓冲器。之后，如果需要重新读取这些数据(这比你想象的更为频繁)，就可以从缓冲器而不是磁盘读取了。

由于访问内存比访问磁盘快得多，缓冲可省下大量时间。BUFFERS命令用于告诉DOS所使用的缓冲器数目。一个合理的数目，从10到50将改善系统性能。如果使用得过多，搜寻所有缓冲器的时间可能比访问磁盘更长。同时过多的缓冲器也会占用更多的内存。

如果DOS被加载到扩展内存中，只要放得下，缓冲器也位于那儿。否则，它们将被放置在常规内存中。

BUFFERS命令能保留附加缓冲器从而使DOS实际上能预读一个文件。这一特性使系统运行得更快(详见命令参考中的BUFFERS)。

如果CONFIG.SYS中不包括BUFFERS命令，DOS将根据系统的常规内存量，使用一缺省数目。一拥有512KB到640KB(最大可能值)的系统，其缓冲器数目缺省值为15。除非使用诸如SMARTDRV.SYS之类的硬盘缓存系统，否则比缺省值更多一些的缓冲器将使系统工作的更好。关于硬盘缓存系统将在第31章"驱动器缓存"中讲述。

某些应用程序推荐特定数量的缓冲器以使操作更有效率。安装这些程序时，它们将在CONFIG.SYS中放置BUFFERS命令，改变已存在的BUFFERS，或者它们的文档中有此类说明要求用户完成这件事。

21.9 文件柄的分配

DOS为文件柄表保留了一块内存区。每个文件柄包含一打开文件的信息。FILES命令告诉DOS你所期望同时打开的文件的最大数目。某些应用程序打开的文件有一最小数目。在安装过程中，一个应用程序可能会在CONFIG.SYS中加入FILES命令，改变已存在的FILES命令，或者它的文档中有要求用户插入FILES命令的说明。

某些较早的应用程序使用文件控制区(FCBS:File Control Blocks)而不是文件柄。如果某个应用程序要求在系统中安装一些FCB，参见"命令参考"中的FCBS命令。

21.10 命令处理器的定义

命令处理器解释从命令提示符或批处理程序输入的命令。DOS所提供的命令处理器为COMMAND.COM。DOS预期在引导驱动器的根目录中找到这一程序。如果将COMMAND.COM换了个地方，或者欲使用其他命令处理器，可在CONFIG.SYS加入SHELL命令。下面的命令从DOS目录而不是根目录加载COMMAND.COM处理器：

```
SHELL=C:\DOS\COMMAND.COM C:\DOS /P
```

紧随SHELL=的第一个词标识命令处理器的文件名。另一个参数对于加载的命令处理

器来说则是唯一的。此例中的参数与 COMMAND.COM 有关。C:\DOS 参数告诉 DOS 在需要重新加载时，在哪儿可以找到 COMMAND.COM。/P 选项告诉 DOS 将 COMMAND.COM 作为永久命令处理器。

第 43 章"DOS 及其运行环境"在讲述如何控制 DOS 环境的同时，将讲述如何创建 SHELL 命令。

21.11 堆栈的分配

DOS 必须保留一定数量的内存用于执行硬件功能。这一部分内存称为堆栈(Stack)。可用 CONFIG.SYS 中的 STACKS 命令控制 DOS 分配堆栈空间的大小。通常 DOS 根据缺省值创建足够的堆栈空间，但某些程序可能会在 CONFIG.SYS 中插入 STACKS 命令，或修改已存在的值。除非你明白 DOS 使用堆栈的原理以及软硬件的需要，才可以改动已存在的 STACKS 命令。

提示： 如果某几个特定程序经常造成死机，或者经常发生 Stack error 或 Stack fault 错误，应向 Microsoft 产品支持队伍求助。这些问题可望用 STACKS 命令加以解决。

21.12 自 测 练 习

本章是对 CONFIG.SYS 文件的一个概述。你也许想看一下 CONFIG.SYS，看看究竟能明白其中的多少命令。使用 DOS 的编辑器浏览此文件。

1. 你能找到加载扩展内存管理程序以及将 DOS 加载到扩展内存的命令吗？
2. 你能找到为硬件设备，如 CD-ROM 驱动器或传真/调制解调器等加载的设备驱动程序吗？
3. 你的 CONFIG.SYS 包括加载 DOS 的读写缓冲器，及文件柄和堆栈的命令吗？
4. 其中有标识命令处理器的命令吗？
5. 你的系统加载了 SETVER.EXE 来控制版本序号，ANSI.SYS 来提供监视器和键盘的扩展功能吗？

不必为 CONFIG.SYS 中那些你不理解的命令焦虑。只要系统能正常工作，你的 CONFIG.SYS 就不会有什么问题。在后续章节中，将讲述改善系统配置的几条途径，尤其是有关内存使用的。

如果已有编辑器打开 CONFIG.SYS，确保在未作任何变动的情况下关闭它。

第四部分　系统管理

第 22 章　数据保护概述

简介： 使用 DOS 所包含的这些工具，可保护系统中的数据免受破坏。

22.1　总　览

正如将在这一章中所学到的，DOS 含有几个用于帮助用户避免和恢复诸如 FAT 表错误或意外删除等特定问题的工具。但是，DOS 提供的最重要的工具是 MSBACKUP(或 Windows 中相对应的 MWBACKUP)，这是一个防止数据丢失的最佳保护工具。该程序帮助你保存所有重要文件的最新备份，从而能在出现丢失时恢复它们。不管遇到什么不幸的事，例如硬盘驱动器有问题，Delete Sentry 清除一个文件而你又想恢复它，即使你的计算机被偷盗，你都可以通过数据的备份拷贝恢复它们，这些备份拷贝应小心地保存在其他地方。

无论是在命令提示符下，在 Shell 中或在 Windows 下，在你喜欢的任何环境中，DOS 6 使得建立和维护常规的备份工作非常容易。第 23 章“MSBACKUP 和 MWBACKUP”讲述如何建立和维护备份系统，第 24 章“Restore 和 Compare”讲述需要时如何恢复文件。

本章读起来有些像一个危险分类目录册，但在阅读时一定要记住，如果有一组良好的备份，一旦系统出现意外就能恢复。在本章中，你将会看到 DOS 提供的对付下列情况的工具：

- 病毒
- 软件故障
- 硬件故障
- 删除的文件
- 重新格式化磁盘
- 磁盘物理损坏

22.2　病毒的处理

病毒是一种程序，它偷偷地潜入你的系统中，隐藏在某处，由某一事件激发并做出意想不到的事情。有些害处小些的病毒显示一些愚蠢的信息或奏一段音乐。但是，[illegible]的病毒会破坏你的分区表或物理地重新格式化硬盘；或者清除硬盘上的所有数据。[illegible]个程序可帮助你保护系统抵御病毒侵害。

MSAV(Microsoft Anti-Virus)程序，它从命令提示符或 Shell 运行，扫描[illegible]检查病毒。它可以安全地从系统中清除许多病毒；对于其他一些病毒，则可

除被感染的文件(文件中隐藏病毒)。MWAV(Microsoft Windows Anti-Virus)是该程序的Windows版本。

VSAFE程序作为一个终止驻留程序(TSR)一直监视系统中类似病毒的活动,如物理地重新格式化硬盘。VSAFE锁住任何可疑的活动,警告用户发生的事情,并由用户决定下一步做什么。VSAFE能够在DOS,DOS Shell和Windows下运行。

第25章"DOS的反病毒程序"讲述DOS的反病毒程序。如果有规律地使用它们,系统就几乎可以免受病毒的侵害。但是总是有新型的病毒会偷偷逃过所有安全保护措施。而且,如果不坚持对病毒的监视(反病毒程序既占用空间也占用时间),那么受到病毒攻击的机会就会增加。备份是对付病毒的最后一道防线:即使病毒损坏了一部分数据,也可以通过备份盘恢复这些数据(在确信它们没有感染病毒后)。

22.3 软件运行失常的处理

任何程序在系统中都可能产生问题。回想过去的日子(即70年代末期),那时个人计算机一般只有16KB内存,可扩展到64KB,软件非常简单,也只有如此,否则它不能适应可用的内存空间。但是现在,程序可以非常复杂。像动画演播,音乐序列发生器,甚至日常的字处理程序或数据库管理器应用程序,都要很多人用几年的努力去开发。然后不幸的是,还是会出现错误。即使测试和调试过程相当严格,也可能漏掉程序某个角落的小错误。在现实世界中,这种情况多得更为复杂。在IBM PC兼容机世界中,存在着这么多硬件和软件的组合,以致于无法在可能使用的每个环境中都测试一下新的程序。

所开发的在早期DOS版本下运行的程序也可能会出现问题。Microsoft的技术人员在上千种系统上,用他们能够找到的各种硬件和软件测试每一个DOS版本,以使它没有错误。但是要达到百分之百的可靠性是不可能的。

小型的软件产品也有它们的问题。许多由一两个人开发的产品不可能像测试DOS那样进行测试。这些程序可能只在二、三种系统上测试,也可能是十几种,但不可能是上万种。另一方面,由于这些程序较小,并不复杂,这样就不太可能隐藏问题,尤其是如果它们具有较好的声誉并且在可靠的商店中销售就更是如此。然而,并不能保证这样的产品不会在你系统中产生问题。一般来说,一个程序越流行,它造成问题的可能性就越小。

有一个便宜软件的广阔世界:自由软件,公用软件和共享软件。它便宜的部分原因是由于开发者在包装、市场推广或发布方面没有花费资金。但是,有些开发者在开发和测试期间有时也会出现一些问题。其结果是在一个系统上运行良好的程序,在另外的系统上会产生问题。广泛应用的共享软件,如PKZIP和来自PKWARE的姐妹程序,在该领域已经过广泛的测试,这使得它与花费更高的经过包装的程序一样安全。

如果一个程序在你的系统中出现失误,它会使你丢失数据。它可能会冻结你的系统并强迫你在不希望的时候重新启动。或者,更糟糕的是,程序甚至会损坏该系统区或硬盘上的数据。

没有什么办法避免所有程序不会出现错误。但应避免使用可能会带来问题的那些程序。另外,更新备份文件是防范出错的最好手段。

22.4　硬件故障的处理

任何类型的硬件故障，从短暂的电源波动到死机，都会损坏系统区并使硬盘数据不能存取；或者，它会损坏一些特定的文件。有些硬件设备可保护你的系统不出现硬件故障。例如，波峰保护器可以避免电源波峰侵扰你的硬件。甚至可以买一个电池板装入系统，当电源中断时投入工作，从而可提供一分钟左右的时间保存所有文件并正确地关闭系统。如果在你所在的地区经常发生电源波动和断电，那么应向硬件销售商询问如何保护你的数据。

但是没有任何方法能保护你的系统免遭硬盘故障的危胁，而硬盘最终总会出现故障。当出现这种情况时，当你需要修理或更换硬盘时，就会为拥有一组备份来恢复文件而感到高兴（一个出色的磁盘技术人员能够从坏的磁盘上解救大部分甚至全部文件）。

22.5　删除的处理

你可能知道，错误地删除文件或删除一个文件后又想改变主意是经常会发生的事情。DOS 的 UNDELETE 工具在必要时能保护文件不被删除并且能帮助你恢复它们。第 26 章“UNDELETE 的使用”讲述如何建立和使用 UNDELETE。

然而，即使 DOS 的高级删除保护 Delete Sentry 也不能保证恢复每一个文件。再次提醒一组好的文件备份能够帮助你恢复那些由其他方法不能恢复的文件。使用删除保护只是保护那些还没有备份的文件的权宜之计。

22.6　重新格式化的处理

错误地重新格式化一张磁盘同样是经常发生的问题，DOS 的 UNFORMAT 工具是专为此情形设计的。但是，UNFORMAT 要考虑两个因素。首先，镜象文件必须存在。第二，格式化后磁盘上没有保存新的数据。UNFORMAT 没有这两个条件就不能工作，因此最好从备份中恢复丢失的文件。

22.7　受损磁盘的处理

目录树和文件分配表（FAT）是访问磁盘数据的基本条件。如果有一个受到损坏，就可能失去与一些，也许是全部文件的联系。这一重要的系统信息在许多情形下会受到损害。硬件故障，电源波动，病毒以及不等到驱动器灯灭就去重新启动系统都可能产生这一问题。

DOS 含有一个叫做 SCANDISK 的程序，它分析 FAT 表和目录结构，报告问题，并能修理它所发现的大部分问题。另外，它检测磁盘表面的物理缺损并抢救缺损区域的数据。SCANDISK 在第 27 章“CHKDSK 和 SCANDISK”讨论，它是一个维护工具，应该有经常运行以保持硬盘处于良好的运行状态。

DOS 6.2 以前的版本不包括 SCANDISK。一个早期的 DOS 程序叫做 CHKDSK（检查磁盘），用来检查和修理目录结构，但不能扫描磁盘表面。如果还没有更新到 6.2 或 6.22，就只

能使用 CHKDSK 而不是 SCANDISK 来维护磁盘。

CHKDSK 和 SCANDISK 不能处理每一个它们发现的问题，仍有可能必须从备份恢复受损的文件。

22.8 自测练习

明智的用户应有规律地使用本章中讲述的(以及下面几章中详细描述的)保护程序。VSAFE 和 Delete Sentry 应该在每次启动时作为 TSR 程序装入。MSAV 或 MWAV 应当每天运行，也可在每次启动时运行。CHKDSK 和 SCANDISK 不需要每天运行，每星期运行一次就可以了。最后，应每天使用 MSBACKUP 进行备份的更新。

第 23 章　MSBACKUP 和 MWBACKUP

简介： *好的备份系统是保护数据最重要的方法。DOS 6 提供了一个有用的工具，下面将学习如何使用它。*

23.1　文 件 备 份

DOS 6 用更现代的使备份更简单的 GUI 设施代替了早期 DOS 版本的备份工具。该程序有两个版本，一个在命令提示符下运行(MSBACKUP)，另一个则在 Windows 下运行(MWBACKUP)。新的备份工具包含的主要特性能使工作更为轻松：

- 它能够压缩备份数据以节省存储磁盘的空间
- 它能够与备份数据一起存储错误校验信息，这样即使备份文件受到损坏也能恢复你的数据
- 它能够在软盘已含有数据时发出警告
- 它维护一组分类目录以便跟踪备份文件
- 它能够比较备份的文件和硬盘上的文件并在出现不一致时发出警告
- 它能够备份到软盘，或到 DOS 驱动器\目录名下(诸如磁带驱动器或网络驱动器)
- 能够只用按动鼠标或键盘按键运行每天的备份
- 能够选择一个或多个文件从备份中恢复；也能恢复一个被删除的文件，返回到文件早期的版本，或在彻底崩溃之后恢复整个硬盘

本章讲述如何进行备份。将学习如何：

- 在 MSBACKUP(for DOS)和 MWBACKUP(for Windows)之间做出选择
- 启动 MSBACKUP 或 MWBACKUP
- 运行备份配置测试以使备份知道如何访问你的硬件
- 创建和管理设置文件，这样只需定义备份一次
- 使用备份设置文件备份文件
- 解释备份分类名称
- 管理备份分类

警告： *MSBACKUP 和 MWBACKUP 是 DOS 6 的新内容，不能用来恢复早期 DOS 版本备份的文件。DOS 6 把原来的 RESTORE 作为一个单独的程序，以便在需要从早期备份中恢复文件时使用。应该尽可能用新的 Backup 备份你的文件并删除原有的备份。然后，可以删除 C:\DOS\RESTORE.EXE。*

DOS 6.22 的压缩技术不同于 DOS 6.0 和 6.2 版本中的方法。如果你用 DOS 6.0 或 6.2 进行了备份，然后升级到 DOS 6.22，也许会在从早先做的备份中恢复文件时遇到麻烦。第

24 章“Restore 和 Compare”讲述快速恢复文件的条件，以及如何在其他情况下下恢复这些文件。与此同时，应该尽可能快地使用 DOS 6.22 进行新的完全的或全部的备份，以替代原有的备份。

23.2 练习准备

在本章中，你需要一些练习目录和文件用于操作练习。建立一个 PRACTICE 分枝而不是对实际文件进行操作是一个较好的方法：

1. 建立一个名为 PRACTICE 的目录，使它成为根目录的子目录。把 DOS 中所有的 .HLP 文件拷贝到 PRACTICE 中，把它们重新命名为. XXX 文件(注意：用一个 XCOPY 命令就能够完成这一工作)。
2. 建立一个名为\PRACTICE\EXER1 的目录并且拷贝所有的\DOS*.COM 文件到它里面，把它们重新命名为*.YYY。
3. 建立一个名为\PRACTICE\EXER2 的目录并且拷贝所有的\DOS*.INI 文件到它里面，把它们重新命名为*.ZZZ。

23.3 MSBACKUP 或 MWBACKUP 的选择

DOS 6 的备份程序包括两个版本：MSBACKUP(Microsoft Backup)在命令提示符或 Shell 中运行，而 MWBACKUP(Microsoft Backup for Windows)在 Windows 中运行。因此，不管你喜欢哪一个环境，都有一个备份工具可供使用。两个版本都采用了图形用户界面，使设置和运行备份程序都很容易。Windows 版本比 DOS 版本看起来更奇妙一些，正如你所希望的，它更易于使用，但功能相同。图 23.1 就是 MWBACKUP 的窗口，本章中其余部分的图例来自 MSBACKUP。

当你在配有 Windows 的系统上安装 DOS 6 时，SETUP 提供安装 MSBACKUP 或 MWBACKUP 的选择，也可以两者都选。你可能不需要两者都选，只安装使用其中一个可节省硬盘空间。如果以后想改变主意，可重新运行 SETUP 安装另一个版本。

注释： 为了避免罗嗦地经常说“MSBACKUP 或 MWBACKUP”，本章使用“Backup”表示其中任何一个。

23.4 开始备份

根据所使用的环境，有几种方法启动备份软件。现在尝试启动你的版本：

1. 如果想从 DOS 提示符进行备份，键入 MSBACKUP。
   如果显示信息 Backup requires configuration for this computer，则表明以前从没有使用过备份程序。阅读“备份配置”这一节的内容。如果没有出现该信息，跳到“备份功能的选择”一节。
2. 如果想从 DOS Shell 进行备份，则先启动 Shell，然后，打开 Disk Utilities 程序组并打

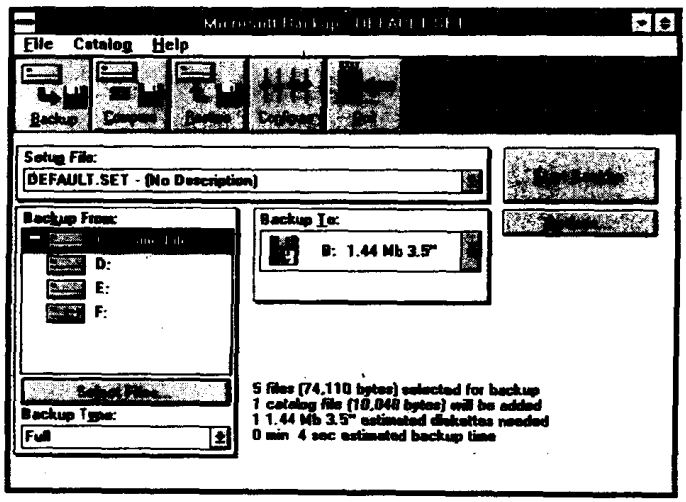

图 23.1　MWBACKUP 窗口

开 MSBACKUP 程序项。或者，可选中 File，Run，并在 Run 对话框中键入 MSBACKUP。

如果信息显示 Backup requires configuration for this computer，则表明以前从没有使用过备份程序。阅读“备份配置”一节的内容。如果没有出现该信息，跳到“备份功能的选择”一节。

注释： *MS Backup 程序项是定义好的，因此在任务切换时不能从该任务切换离开。一旦已经启动 MS Backup，在切换到另一个任务之前，应让它完成工作。*

3. 如果想在 Windows 下进行备份，启动 Windows。然后，打开 Backup 程序项（在 Microsoft Tools 组中）。

   如果信息显示 Microsoft Backup has not been configured. Do you want to automatically configure it now?，则表明以前从没有使用过你的备份程序。阅读“备份配置”一节的内容。如果没有出现这一信息，跳到“备份功能的选择”一节。

23.5　备 份 配 置

第一次启动任何一个版本的备份程序，都会看到一个告诉你还没有配置该软件的信息。应首先要做这一工作。在配置期间，Backup 检测系统以确保它能够产生可靠的备份。它涉及几种检测：

- 检查所有的硬件类型（视频、鼠标和磁盘驱动器）。如果它不能准确地识别它们，你还有机会进行纠正。
- 当改变软驱磁盘时，它进行检查看一看它是否能觉察到。如果能够，则不必在每次插入新磁盘时选择 OK 按钮。
- 检测内存和磁盘以便确认它能够快速可靠地阅读数据。
- 它用几张盘对数据进行试验备份，因此必须至少换盘一次。这一小备份会暴露与你

的系统中的某些部分的不兼容性：驱动器、内存，与 Backup 的过程相冲突的 TSR 程序等等。备份通过减慢其自身的速度能够避免一些不兼容性。其他问题（例如不兼容的 TSR 程序）则必须在使用 Backup 之前由你自己加以解决。

- 它对试验备份中的所有文件与硬盘上的原始文件进行比较。因此，这将会暴露在系统上进行可靠备份可能出现的问题。

23.5.1 开始检测

我们现在运行配置检测。下面的练习是针对 MSBACKUP 版本的。MWBACKUP 版本与之相类似。

1. 如果还没有做过这些，则像以前说明的那样启动 MSBACKUP 或 MWBACKUP。
   出现一个警告你需要做配置检测的信息。
2. 选中 Start Configuration 开始检测。
   在 Backup 为你进行选择时，将看到一些对话框。然后，Backup 显示有关视频和鼠标配置的确认（如图 23.2 所示）。

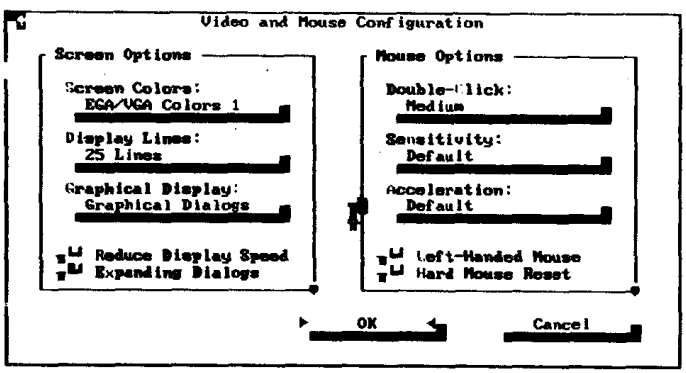

图 23.2 Video and Mouse Configuration 对话框

23.5.2 确认视频和鼠标配置

Video and Mouse Configuration 对话框提供改变不正确的配置项的机会，如果想在对话框顶层改变其中一项，如 Screen Colors，选择它后会出现另一个显示该选项的对话框。

在对话框底层的四个选项，如 Expanding Dialogs，是检验框，可以使之有效或失效。选择某项一次是使它有效；对话框中出现检验标记或 X 表明它有效。再次选择该项检验标记或 X 消失。

注释： 要选择 Video and Mouse Configuration 对话框或其附属对话框中的某项，可单击该项或输入 Alt 加上对应的高亮度显示的字母。例如，要改变 Screen Colors，单击标有 Screen Colors 下的方框，或者直接按 Alt+S。

注释： 在 Backup 中打开任何对话框时，按 F1 就会得到关于它包含的选项的屏幕帮助。

一般不需要改变该对话框中的大多数据。但是，当使用 MSBACKUP 时屏幕上出现“雪

花”，利用有些选项可以消除它。如果雪花的出现与鼠标移动有关，改变 Graphical Display 使用更少的图形。如果这样还不行，则使对话框底部的 Reduce Display Speed 有效。如果有任何其他的鼠标使用问题，例如，它不能准确移动，可以尝试使 Hard Mouse Reset 有效。

如果遇到有关双击鼠标的麻烦，可尝试不同的双击速度。选择慢速后，就不必太快按动按钮发送双击信息。该速度是为那些发现双击太快的用户而设置的。这对许多人来说并不合适，它会使在一行中两次单独的单击被解释为一次双击。如果你移动很快并且想让 DOS 与你保持一致，应选择快速的双击速度。

根据缺省设置，对话框并不是直接出现在屏幕上。它开始时是一个小方框并逐步扩大到其全部尺寸。如果你的系统较慢而且你不喜欢等待对话框在屏幕上扩大，应使 Expanding Dialog Boxes 无效。当禁止该选项时，对话框就会直接弹出到屏幕上。

如果想对 Video and Mouse Configuration 做改动，现在就可以按上述方法进行。然后，选中 OK 继续。

23.5.3　软驱变化行检测

在视频和鼠标配置之后，还有一些对话框出现。然后，Backup 准备进行软驱变化行检测。该测试确定当软盘驱动器中更换软盘时，Backup 是否能感觉到：

1. 将软盘从每个软盘驱动器中移开。
2. 准备好继续时选中 Start Test。

   Backup Devices 对话框显示 Backup 对软盘驱动器的分析（如图 23.3 所示）。每个驱动器按钮标有该驱动器的当前设置。

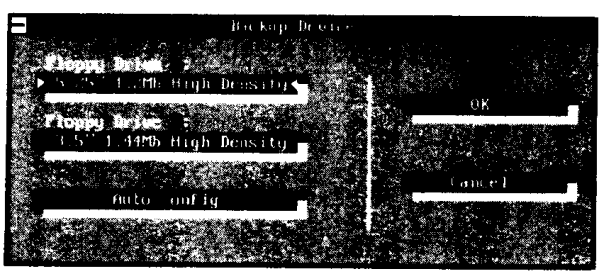

图 23.3　Backup Devices 对话框

3. 如果 Backup Devices 是正确的，选中 OK 继续下一项检测。如果不是，花时间纠正它们。要选择一个设备，选择该驱动器按钮并且从显示的列表中选择正确的描述。

23.5.4　软盘兼容性检测

下面，将看到 Backup 进行的一些其他检测。你所能做的就是坐在一边观察，直到它准备做软盘兼容性检测（如图 23.4 所示）。该检测做一个实验备份并进行比较。

有一个选项可以跳过这一检测，但如果你是第一次进行配置则不应跳过该项（如果出于某些原因重新运行配置检测，可以不重新做兼容性检测）。

1. 选中 Start Test 开始软盘兼容性检测。

   当 Backup 开始测试时会出现一系列对话框。（在本章中，将学习如何使用所有这些对

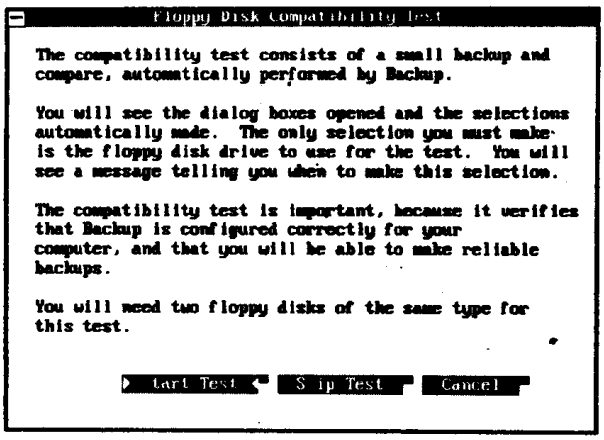

图 23.4 Floppy Disk Compatibility Test 对话框

话框,但现在还不到时候。)最后,Backup 要求你选择要检测的软驱以及容量。

2. 选择要检测的驱动器。要确保检测的正是要用于实际备份工作的驱动器。另一方面,即使你想备份到网络驱动器或其他设备,也必须选择对软驱的兼容性测试。记住必须为所选择的驱动器和容量提供两张磁盘。

提示: 如果有两个相同的软驱,可以选择使用两者。这将加速整个备份过程,因为当 Backup 写到一个软驱时,可以在另一个软驱中更换磁盘。Backup 不必等待更换磁盘。

选择驱动器后,将会出现一些对话框。然后,Backup 要求你插入第一张磁盘。

3. 在指定驱动器中插入第一张磁盘并选择 Continue(磁盘可以是格式化过的,也可以不是)。

Backup 检查该磁盘,如果磁盘含有数据将显示一个警告框。可替换该磁盘并选择 Retry,或简单地选中 Overwrite 继续使用当前磁盘。

警告: Backup 替换磁盘上的所有已存在的数据。既使只备份一个文件,所有已存在的数据也将受到损坏。

下面,当备份进行时,将看到一个进程屏幕。该进程屏幕在本章随后的"运行备份"一节中讲述。最后,要求你插入第二张磁盘。

如果 Backup 的配置可以觉察到磁盘的变化,插入下一张盘的信息直接出现在进程屏幕上,而不是在一个对话框中。当 Backup 等待新磁盘时,驱动器灯将一直亮着。使用 DOS 时当驱动器灯亮着的时候是你应当更换磁盘的唯一时间。

如果在几秒钟内不能更换磁盘,Backup 停止对更换磁盘的检测并显示一个对话框。当 Backup 不能探测到驱动器中更换的磁盘时,它也显示一个对话框。

4. 插入第二张磁盘。如果一个对话框出现在屏幕上,选中 Continue。

当备份完成时,Backup 显示一个小结对话框。Backup 检查磁盘并在该磁盘含有数据时显示一个警告框。这时可以替换该磁盘并选择 Retry,或者简单地选中 Overwrite 继续使用当前磁盘。

5. 选择 OK 继续。

然后，Compare 功能开始，像以前一样，出现一系列对话框。

最后，Compare 要求你插入第一张备份盘。

6. 插入第一张备份盘(从第 3 步得到)，并选中 Continue。

Compare 读取备份文件并把它们与硬盘上的原始文件相比较。完成之后，它要求插入第二张备份磁盘。

7. 插入第二张备份磁盘(从第 4 步得到)。如果一个对话框出现在屏幕上，选中 Continue。

最后，Compare 显示一个 Compare Summary 对话框。

8. 选中 OK。

出现一告诉你兼容性测试成功的信息(如果在检测期间 Backup 或 Compare 遇到问题，见下面一节)。

9. 选中 OK。

Configure 对话框出现在屏幕上。

10. 选中 Save。

Backup 保存配置和兼容性检测的结果。

注释： *保存检测结果是非常重要的。如果不这样做，下次启动备份程序时就必须重新运行检测。但是，你不必保存刚刚创建的备份软盘。可以立即重新使用它们。*

11. 选中 OK 退出 Configure 对话框。(如果你使用 MWBACKUP，跳过这一步)。

MSBACKUP 的主菜单出现。

23.5.5 配置问题的处理

当然，配置检测并不总是一帆风顺的。如果 Backup 遇到麻烦，它将报告该问题并在可能的情况下建议一种解决方案。例如，它可能请求你清除一个 TSR 程序并再试一次。

如果你不能理解错误信息，逐字将其拷贝下来，并记住是在配置过程中哪里出现的(如“就要启动试验备份时”)，并且打电话给 Microsoft 产品支持部门。他们应该能够帮助你确认问题和完成系统配置。

如果兼容性检测失败，这时仍能继续进行备份，但是不应这样做，因为它们是不可靠的。在确认并找出问题后，应当重新运行兼容性检测。

如何重新运行兼容性检测

1. 选中 Configure，打开 Configure 对话框。
2. 选中 Compatibility Test 按钮。
3. 按屏幕上的指示进行操作。

如果配置检测的其他任何部分失败，必须在使用备份程序之前，确定并解决问题。大多数情况下，问题出在硬件中或有不兼容的 TSR 程序。配置检测期间接收到的信息应当能够帮助你确认问题，而且会给出一些解决它的建议。

每次启动备份程序时都出现配置警告框，直到成功地通过除兼容性检测之外每一个检测为止。当系统已经通过所有其他配置检测，但兼容性检测失败时，另一个警告框会提出警告。

答疑： 当 Windows 在 Enhanced Mode 下运行时，MWBACKUP 不能通过兼容性检测，但是它可以在 Standard Mode 下通过检测。

有些 Windows 应用程序安装了与 MWBACKUP 相冲突的设备驱动程序。因此使用 MSBACKUP 对你来讲可能更容易些。但是，如果确实想使用 Windows 版本，可在不安装任何其他应用程序的情况下重新启动 Windows，并再次尝试。

如果它仍然不能工作，把文件 WIN. INI 作为 WIN. SAV 文件拷贝下来，然后编辑 WIN. INI 删除每一个以 LOAD＝或 RUN＝开始的行。然后，重新启动 Windows 并再次尝试。

如果在编辑 WIN. INI 之后 Windows 不能启动，把 C:\WINDOWS\WIN. SAV 拷贝到 C:\WINDOWS\WIN. INI(恢复原有的 WIN. INI)，然后重新启动 Windows。然后，编辑 WIN. INI 只删除一项，重新启动 Windows，并进行兼容性检测。最后，你会发现与兼容性检测相冲突的项。

在通过兼容性检测后，记录下从正常的 Windows 配置中删除的那些项。每次运行 Backup 时必须删除相同的项。

23.5.6 重新检测系统

如果改变了系统配置，可使用相同的步骤重新运行兼容性检测。在某些情况下，应该重新运行该检测程序，例如，安装一个新的 TSR 程序或改变了一个 TSR 的配置。

注释： 当安装一个新的 TSR 程序时，许多人往往忘了要重新运行配置和兼容性检测。两个备份程序都会检查是否配置有改变，如果是，则警告你需要进行新的检测。

如何重新配置软盘驱动器

1. 选中 Configure，打开 Configure 对话框。
2. 对于 MSBACKUP，选择 Backup Devices，然后按 Auto Config。
   对于 MWBACKUP，按 Auto Floppy Configure。
3. 按屏幕上的指示进行操作。
4. 如果 Backup 错误地识别驱动器，改正它们。

23.6 备份功能的选择

完成配置检测后，就要准备进行第一个备份。从现在起，启动 MSBACKUP 时，将看到能够选择 MSBACKUP 四个功能(Backup，Restore，Compare 和 Configure)或 Quit 的对话框(如图 23.5 所示)。

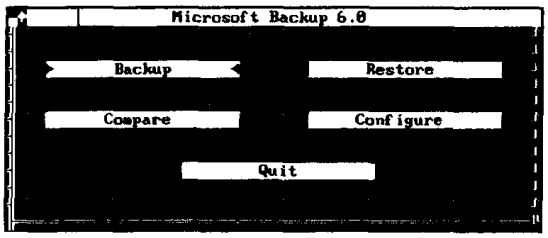

图 23.5　MSBACKUP 的主菜单

注释： 对于 MWBACKUP，功能作为按钮出现在窗口靠近上面的地方。在前面所示的图 23.1 中能够看到它们。

Backup 功能把从硬盘中选出的文件归档到诸如软盘或另一硬盘驱动器的备份介质中。Backup 能够压缩备份的数据以节省空间，而且可配置它到最大速度，最大可靠性，或在两个极端之间的某个位置。

Compare 功能把备份文件与硬盘上对应的原始文件相比较。Compare 能验证备份的准确性，确认备份后文件有否改变。

Restore 功能使备份的文件返回到硬盘。如果硬盘出现故障，可使用 Restore 把所有文件放到新的或修复的驱动器上。同时，也可以用 Restore 恢复一个被意外删除的文件，或者返回到丢弃的文件早先版本。

Compare 和 Restore 将在第 24 章中讲述。本章讲述如何使用 Configure 和 Backup 功能。从主菜单中选择 Backup 就可以启动 Backup 功能，然后出现 Backup 对话框（如图 23.6 所示）。

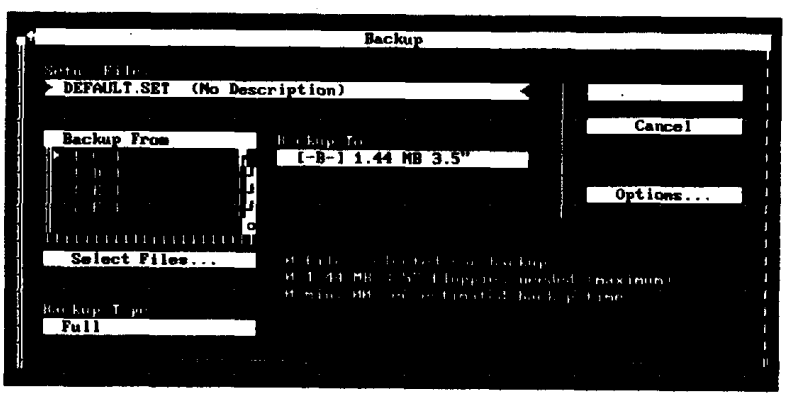

图 23.6　MSBACKUP 的 Backup 对话框

23.7　Setup 文件操作

当你定义一个想要执行的备份时——选择要备份的文件，选择接收备份文件的驱动器，选择不同的备份选项等等，应把你的工作保存在一个 Setup（配置）文件中。下一次想要运行同样的备份时，所要做的就是打开 Setup 文件并按 Start Backup 按钮。因为每天可能重复相同的备份，Setup 文件使你在第一天之后只需按几个键或几下鼠标就能完成所有的工作。

Backup Setup 文件有许多其他优点。它们能与其他用户共享，或保存在网络驱动器上以便确保工作组中的每个人执行相同的备份标准。在 PC 机上可以有多个 Setup。例如，如果几个人共享一台计算机，每个人有各自的 Setup 文件。对每一个重要的应用程序，也可以分别有一个 Setup 文件。然而，很多人只使用一个 Setup 文件进行备份。

如何创建一个新的备份 Setup

1. 打开一个与你想要创建的相类似的 Setup 文件。
2. 选中 File，Save Setup As。
   Save Setup As 对话框出现。
3. 指定文件名以及新的 Setup 文件说明，并选中 OK。
   对话框关闭，并且 Backup 打开新的 Setup。
4. 为新的 Setup 选择文件。
5. 为新的 Setup 选择选项。
6. 为新的 Setup 选择备份类型。
7. 选择进行备份的驱动器。
8. 选中 File，Save Setup 保存新的 setup 文件。

每个新的 Setup 文件必须基于一个已有的文件。Backup 提供了一个叫作 DEFAULT.SET 的 Setup 文件，可用它建立你的第一个 Setup 文件。根据缺省设置，DOS 在相同目录中保存新的 Setup 文件作为当前的 Setup。如果基于 DEFAULT.SET 建立新的 Setup 文件，C:\DOS 是缺省目录。现在，将建立一个练习 Setup 并保存在 C:\DOS 中。

基于 DEFAULT.SET 创建一个新的名为 PRACTICE 的 Setup 文件。

1. 启动 Backup 功能。DEFAULT.SET 自动打开。
2. 选中 File，Save Setup As。出现一个询问新的 Setup 名字的对话框，显示 DEFAULT.SET 作为缺省名。缺省目录是 C:\DOS。
3. 键入 PRACTICE.SET。
4. 单击 Description 框，或按 Alt+C 跳过它。
5. 在 Description 框中键入 Practice with DOS pragram files。
6. 选中 Save。
   对话框关闭并且 PRACTICE.SET 成为当前 Setup 文件。

至此，PRACTICE.SET 与 DEFAULT.SET 有着相同的定义。可修改它建立你所需要的 Setup。

23.7.1 为 Setup 选择文件

在一个 Backup Setup 中必须指定欲备份的文件。例如，可以备份驱动器 C 中的所有文件，所有非程序文件，或某一目录中的所有文件。DEFAULT.SET 并不选择任何文件，因此，当你基于 DEFAULT.SET 建立了一个新的 Setup 文件时，新的 Setup 一开始也不选择任何文件。

如何为 Backup Setup 选择文件

1. 打开 Setup 文件。
2. 若要选择某一驱动器上的所有文件，在 Backup From 框中选择该驱动器(注意：双击该驱动器名，或高亮度显示该驱动器并按空格键)。
   Backup 对话框显示所选择的文件数(加上一个分类文件)。在驱动器名字旁边有一个箭头和 All files 字样。
3. 要想根据文件名包括或排除文件，选择 Select Files。
   Select Backup Files 对话框打开。
4. 选中 Include 或 Exclude 按钮。
   Include 或 Exclude 对话框打开。
5. 在相应的框中输入路径名和文件名。
6. 根据你是否想要包括或排除整个分枝中的文件，允许或禁止 All Subdirectories。
7. 选中 OK 关闭 Include 或 Exclude 对话框。
   所选择的文件数出现在 Select Backup Files 对话框的底部，而且在文件列表中所选择的文件旁边有一个检验标记。如果备份类型是 Incremental 或 Differential，则只有具有正的文档属性的文件有一个检验标记。其他所选择的文件有一个点，因为选择它们是为了不被备份，除非把备份类型改为 Full。
8. 重复第 3 到 7 步，直到已经包括了或排除了所有必要的文件。
9. 要想选择或退选单个目录和文件，再次选中 Select Files。
10. 要想选择或退选整个目录，滚动目录树直到找到它为止，然后，双击该目录，或者高亮度显示它并按空格键。
    该目录名变为红色表明它已被选择或退选。如果被选择，它旁边出现一个实的箭头，并且文件列表中的所有文件有检验标记，这表明它们已被选择。
11. 要想选择或退选某个文件，高亮度显示所在目录名，然后，双击文件列表中的某个名字，或高亮度显示它并按空格键。
    该目录名变为红色表明它的某些文件已经被选择或退选。一个 V 字形标记出现在该目录旁边表明它的文件只有一些被选择。如果选择某个文件，则此文件名旁有一个检验标记，如果是退选的文件，则没有检验标记。
12. 要想根据日期、属性或文件名(用于拷贝保护的文件)排除文件，选中 Special 按钮。
    Special Inclusions 对话框出现。
13. 填充相应的对话框。
14. 选中 OK 关闭对话框。
    文件列表中被排除的文件的旁边有一个点标记。

有些人对硬盘上的每个文件进行备份，这样在必要时，就可以很简单地恢复整个硬盘。另外一些人喜欢只备份那些安装应用程序之后创建或修改的文件(如文档文件)，而不是原始的程序文件，它们通常并不改变并且在必要时可从原始磁盘装入。跳过程序文件将大大节省备份时间和介质空间，但使得完全恢复硬盘变得困难些。Backup 可以很容易地设置成这两种方法。无论选用哪种方法，都可删去不必要的文件，如便笺、个人备忘录、工作清单等。贯穿本章，“选中的文件”(selected files)是指已选择的欲备份的文件集合，无论它是不是包括硬盘上的所有文件。

23.7.1.1 选择整个驱动器

在 Backup 对话框中，Backup From 框列出硬盘驱动器。如果想备份一个驱动器上的所有文件，选择该框中的该驱动器。可选择想要的任何驱动器，也就是说，能够在一个 Setup 文件中备份系统中的每一个文件。

注释： 选择一个驱动器可以通过双击该驱动器名或按 Alt+K 选择 Backup From 框，键入该驱动器名以高亮度显示该驱动器并按空格键。当整个驱动器被选择时，All files 出现在该驱动器名字旁边。

注释： CD-ROM 驱动器同样出现在驱动器列表中，可选择它们进行备份，但一般没有必要这样做。如果 CD 上的文件被损坏，不可能从备份恢复它们，因为不能写入到 CD。

如果你不想选择一个驱动器上的所有文件，就不要在 Backup From 框中选择驱动器名字。

23.7.1.2 包括和排除文件

大多数人并不想备份整个驱动器。正如早些提到的，许多文件不需要备份，在它们上面不必浪费时间和空间。在 Setup 中选择文件的另一种方法是根据文件名包括或排除它们。例如，可包括所有 DOC 文件，但排除 MEMO*.DOC 和 TEMPY.DOC。

图 23.7 显示在一个 Setup 中用于包括和排除文件的对话框。正如所看到的，可为每一种你想要包括或排除的文件类型指定一个路径和一个整体的或特定的文件名。如果允许 Include 或 Exclude All Subdirectories，在整个特定的分枝中所有所示的文件名相匹配的文件被选择。一遍又一遍地使用这些对话框，可建立完整的包括和排除列表。

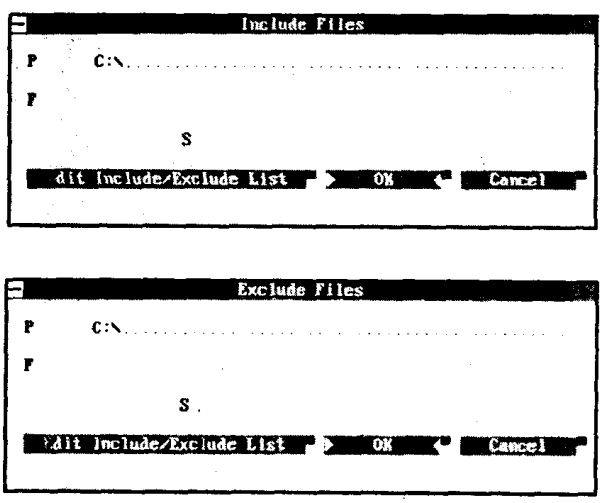

图 23.7 Include 和 Exclude 对话框

在此练习 Setup 中，将包括 PRACTICE 分枝中除 D*.*和 M*.*文件之外的所有文件。

注释： 包括和排除的顺序是重要的。每个新的说明将覆盖所有前面的说明。所以，

在这个练习中，必须在排除 D*.* 和 M*.* 之前包括 *.*。

1. 单击 Select Files 或按 Alt+L。

   Select Backup Files 对话框打开(如图 23.8 所示)。

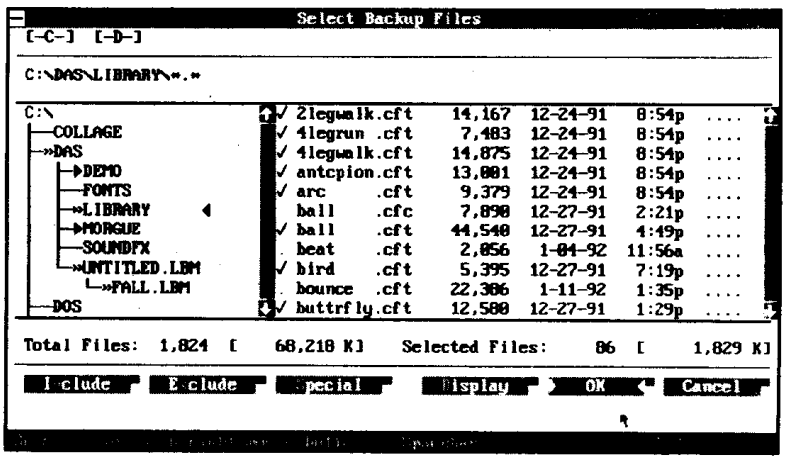

图 23.8　Select Backup Files 对话框

2. 选中 Include。

   Include Files 对话框打开。缺省路径为当前目录，缺省文件名为 *.*。

3. 编辑缺省路径读 C:\PRACTICE。同时允许 Include All Subdirectories。

4. 选中 OK 返回到 Select Backup Files 对话框。

   Include Files 对话框关闭。注意 Select Backup Files 对话框 Select Files 域中的数已经改变。同时还要注意目录树中 PRACTICE，EXER1 和 EXER2 旁边的箭头，它表明这些目录当前已被选择。

5. 要想看 PRACTICE 目录中所选择的文件，可把高亮度显示移到目录树中的 PRACTICE。按向下箭头移动高亮显示，或者单击目录树中的 PRACTICE。

   在文件列表中，一个检验标记表示一个选择的文件。现在，所有文件被选择，所以它们都有检验标记。

6. 现在选中 Exclude，以排除 D*.* 和 M*.* 文件。

   Exclude Files 对话框打开。

7. 确保路径是 C:\PRACTICE。

8. 在 File 框中键入 D*.*。同时允许 Exclude All Subdirectories。

9. 选中 OK。

   对话框关闭。Selected Files 数改变。PRACTICE，EXER1 和 EXER2 目录旁边的实箭头改变为 V 形标记，表明这些目录中只有一些文件被选择。还能在文件列表中看到 D*.* 文件没有检验标记。

10. 对 M*.* 重复第 6 至 9 步。

注释： 在 Include 或 Exclude 对话框中选择 Edit Include/Exclude List，可检查和编辑整个 Include 和 Exclude 列表。可编辑的内容包括修改列表中的入口，拷贝入口以便使用它做为一个新入口的基础，以及从列表中删除入口。

11. 选中 OK 关闭 Select Backup Files 对话框。

Backup 对话框返回。在屏幕右下角，Backup 显示所选择的文件数，所需的软盘估计数，以及备份所需的大概时间。还要注意驱动器列表中驱动器 C 旁边的箭头，及 Some files 提示，表明驱动器 C 上的一些文件被选择。

注释： *Backup 对话框中的文件数总是比选择的文件数多 1 个。附加的文件是备份分类，它必须写到备份介质。分类将在本章的"处理备份分类"一节中解释。*

23.7.1.3 覆盖 Include/Exclude 列表

Include 和 Exclude 对话框创建了一个建立 Setup 的基本文件选择的 Include/Exclude 列表。在 Select Backup Files 对话框中可通过选择或退选某些目录或文件来覆盖它们。

要想切换目录或文件选择状态，双按它，或者高亮度显示它并按空格键。

试一试退选 PRACTICE 目录中的某些文件：

1. 选中 Select Files。

   Select Backup Files 对话框打开。
2. 高亮度显示 PRACTICE 目录。

   文件列表显示 PRACTICE 目录内容。
3. 可重新排列文件清单以便使所有选择的文件显示在清单的开端。这使查看所选择的文件更为容易。选中 Display 按钮。

   Display Options 对话框打开。
4. 允许 Group Selected Files。

   检验标记出现在检验框中。
5. 选中 OK。

   对话框关闭。现在文件清单首先显示选择的文件。
6. 要想退选清单中的第一个文件，双击它，或者移动高亮度显示到它上面并按空格键。

   退选该文件后，检验标记从文件名处消失。应该还可看到 Selected Files 数减 1。
7. 选择 OK 关闭对话框。

当覆盖一个文件或目录的选择时，就是把它从 Include/Exclude 列表控制中删除。可通过再次选择它使它的状态在被选择和不被选择之间切换。目录中的任何一个文件选择被覆盖，目录树中的该目录即变为红色。不幸的是，MSBACKUP 中的文件名并不变为红色，因此有时确认哪些文件是用这种方法操作的很是困难(在 MWBACKUP 中受影响的文件名变为红色)。

当一个目录名是红色时，可使整个目录返回到 Include/Exclude 列表控制，从而使目录中所有人工选择和退选都无效。其方法是在目录名上同时按两个鼠标按钮，或者高亮度显示目录名并按 Alt+空格键。在 MWBACKUP 中，文件列表中每个选择被覆盖的文件以红色显示，用相同方法可以使单个文件返回到 Include/Exclude 列表控制中。

提示： *如果确实出现文件选择错误，可通过删除名为 Setup.SLT 的文件使它们全部无效。它删除多个文件和目录选择，但不涉及 Include/Exclude 列表。*

作者忠告： *当为 Backup 选择或退选文件时，应尽可能使用 Include/Exclude 列*

表。

选择单个目录和文件是相当容易的，这可能使你非常愿意用这种方法进行所有的文件选择，但与建立 Include/Exclude 列表相比，它并不是个好方法。Include/Exclude 列表影响以后添加到驱动器的目录和文件，而选择文件则只影响当前的文件。选择或退选一个目录则仅影响该目录中当前的和将来的文件。

23.7.1.4　进行特定排除

还有另外一种方法可以排除覆盖每个其他方法的文件。图 23.9 显示了在 Select Backup Files 对话框中按 Special 按钮时打开的 Special Selections 对话框。这一功能使用户可以根据日期、属性或文件名排除文件。

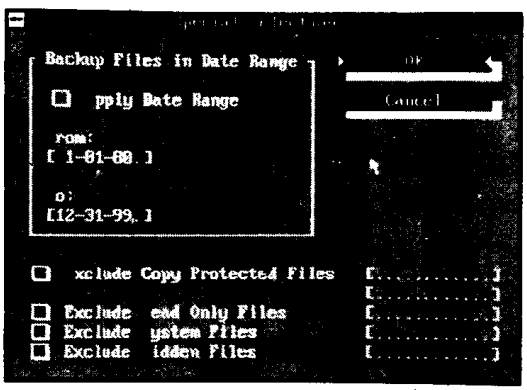

图 23.9　Special Selections 对话框

23.7.1.5　根据日期排除文件

当在 Backup Files In Date Range 区中键入日期范围时，也就排除了所有该范围以外的文件。如果想排除 9/10/92 之前创建或修改的文件，在 From 框中键入 9/10/92，并且在 To 框中键入任何将来的某一日期(最晚是 12/31/99，它也是缺省值)。还需要选中 Apply Date Range 框。当选中 OK 关闭对话框时，就施行这一排除操作，但另一方面直到你关闭 Select Backup Files 对话框时，选择的文件数才会更新。

在该对话框中输入日期范围并不自动地包括该日期内的所有文件。它只是排除超出该范围以外的文件。仍需用其他方法选择包括的文件。

23.7.1.6　根据属性排除文件

通过检验相应的框，可排除隐含、系统或只读文件。

23.7.1.7　排除拷贝保护的文件

为了防止侵权盗版，有些应用程序采取措施保护它们的文件免于被拷贝。如果想拷贝或备份它们，可能会使整个应用程序失效。通常，应用程序的文档会对拷贝保护发出警告。如果你有这样的应用程序，可通过检验 Exclude Copy Protected Files 框并且在框中从左到右最多输入 5 个文件名，使它的文件避免被拷贝。

23.7.2　其他选择功能

用 Display 按钮可打开一个对话框，在这里可为文件列表选择排列顺序，或者为文件列

表键入文件名过滤器。

23.7.3 选择备份类型

大多数人喜欢循环做他们的备份，对所有所选文件进行备份，而不管自从上一次备份以来是否改变。这一方法叫作全备份(full Backup)，在必要时，可从中恢复所选文件的完整集合。

除了全备份，在通常情况下，使用归档属性做为限定，只备份那些自上一次备份以来被修改或创建的文件。这些部分备份与全备份相比只需很少的时间和介质空间；通过用一些部分备份与上一次全备份相结合，就可以完成对全部选择文件的完全覆盖，选择想要的备份类型就可以在全部和部分备份之间切换。

备份通常提供三种备份类型：Full，Incremental(递增)和 Differential(差异)。Full backup 备份所有选择的文件，不考虑它们的归档属性。它还关闭每一个它备份的文件的归档属性。从而开始一个新的备份循环。根据系统的情况，可每周做一次，每两周做一次，或者当 Incremental 或 Differential 备份花很长时间时做一次全备份。

在全备份之间，可进行 Incremental 或 Differential 备份。这些部分备份只归档上一次全备份以来被修改的那些文件。Differential backup 归档那些所选的具有归档属性的文件，并且不关闭它们的归档属性。每次 Differential 备份均归档自从上次全备份以来创建或修改的所有选择的文件。这样，上一次全备份加上最近一次 Differential 备份就代表了恢复所选择文件到硬盘上需要的文件完整集合。

Incremental backup 归档具有真的归档属性的所有选择文件，并且关闭它们的属性。换句话说，每次 Incremental 备份只归档那些自上次全备份或 Incremental 备份以来被创建或修改的文件。如果每天运行一次 Incremental 备份，每天的运行只备份当天创建或修改的文件。为了具有一个完整的文件集合，必须保留上次的全备份，并加上随后每一次的 Incremental 备份。

作者忠告： 不要在一个 Setup 中混合 Differential 和 Incremental 备份类型。

每一种部分备份都有其优缺点。Differential 备份可以使你每天使用相同的磁盘，但当循环进行时备份时间很长。Incremental 备份要求每天使用新的磁盘，但每天的备份只需很短时间。如果在一个循环期间多次修改一个文件，Incremental 备份将保存该文件的每个版本；Differential 备份则只保存最近的版本。如果不想返回到早期的各个版本，每天只在相同的几个文件上工作，Differential 备份就具有了优势。但是，如果每天处理不同的文件(尤其是在对其中的许多文件进行改动时)，Incremental 备份就是较好的选择。在任何情况下，当 Differential 备份时间太长，或已堆积了许多 Incremental 备份磁盘时，就应该开始新的全备份循环。

如果在同一个 Setup 中把 Incremental 和 Differential 备份相混合了，最后将只得到备份文件的不完整集合。如果在开始做 Differential 备份之后，发现更喜欢 Incremental 备份，或相反，在切换到另外一种部分备份之前应先做一次全备份。

一旦为一组文件创建了一个备份 Setup，对全备份和部分备份应使用同样的设置。在 Full 和 Incremental 或 Differential 之间根据需要切换 Backup Type。这样，就能保证每天选择

的是相同的文件。

在该试验备份中，试着做一个全备份。

1. 选中 Backup Type 按钮。

   Backup Type 对话框打开，显示 3 种备份类型。

2. 选择　　并选中 OK。

   对话框关闭。Backup Type 按钮现在是 Full。

当 Backup Type 是 Incremental 或 Differential 时，只有那些具有归档属性的所选文件被备份。检查 Backup File Selection 对话框中的文件列表，那些旁边有一个点的被选文件将不会被备份，因为它们已经由 Special Inclusions 对话框排除，或者因为它们的归档属性是关闭的。

注释： *当你在全部和部分备份之间切换时，将看到欲写入文件数有较大的改变。*

23.7.4　标识备份驱动器

在 Backup 对话框中，可用 Backup To 框选择把备份写到何处。如果备份到软盘，要保证选择通过兼容性检测的驱动器。

对这一试验备份，将使用在兼容性检测期间使用的相同的软盘驱动器：

1. 选中 Backup To 按钮打开 Backup To 对话框。

   所有软盘驱动器列在该框中，加上 MS-DOS Drive and Path。

2. 选择在兼容性检测期间检测过的软盘驱动器。

提示： *如果有两个相同的软盘驱动器，Backup 含有同时使用两者进行备份的选项，应选择这一选项以加速备份。*

要想备份到另外的设备类型，它必须是可用 DOS 驱动器和路径名访问的。例如，如果想备份到网络驱动器的一个目录，应选中 MS-DOS Drive and Path，然后在随后出现的正文框中键入路径名。

23.7.5　备份选项的选择

Backup 有几个选项用于调整速度、可靠性和最小介质用法。例如，可选择压缩备份数据，它不但少占介质空间，还可以节省备份时间。还能选择在备份进行时，把所有写入的数据与原始文件比较验证；该类型的验证会占用很多的备份时间，但能保证软盘可用于文件恢复。其他选项包括用口令保护备份的文件，无条件格式化所有备份磁盘等等。

在阅读每个选项之前，选中 Options 按钮打开对话框看一看完整的列表。Disk Backup Options 对话框打开(如图 23.10 所示)。

23.7.5.1　Verify Backup Data(验证备份数据)

Verify Backup Data 使备份速度减慢但能确保结果可靠。当每个文件写入时，它很快地被再读出并与原始文件相比较。错误被立即纠正。如果由于目标介质上有损坏的地方，Backup 尝试几次都不能写一次正确的拷贝，它请求改变介质以便其再次尝试。

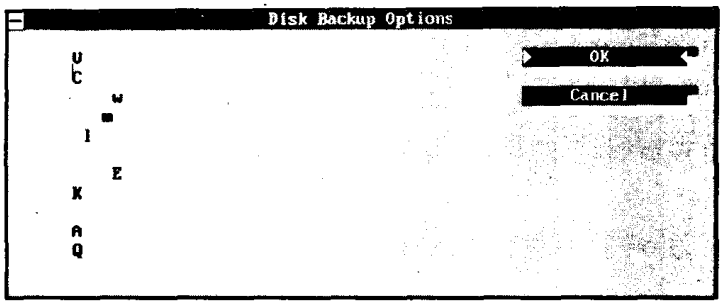

图 23.10 Disk Backup Options 对话框

注释： 这是比 DOS 的一般验证性能更彻底的验证。不要认为 DOS 验证能胜任验证备份文件。

另一种验证备份文件的方法是用 Compare 功能把它们与原始文件相比较，这将在第 24 章讲述。该功能在可能的情况下还能修正错误。但是，如果一直等到备份完成才检查错误，如果有错误不能修复，有必须重新运行整个备份过程。而当你在进行过程中验证备份，那么即使遇到不能纠正的错误时，也不必重复整个备份。作者建议始终选用 Verify 选项。

23.7.5.2 Compress Backup Data(压缩备份数据)

该选项可节省目标介质的空间和备份时间。Backup 尽可能地压缩要备份的数据，做这一工作不需占用更多的处理器时间。因为它不占用附加时间进行压缩，并且因为它最终只写很少的数据，最后结果是备份用的时间更少。

正如你看到的，Backup 对话框中的 Backup 显示备份需要多少张磁盘。但是，这一估计不考虑压缩。如果使用压缩，可能只需要大约估计的磁盘数的一半(实际数量与要备份的文件类型有关，有些类型比其他的更易压缩。)

警告： 不要忘记 DOS 6.22 的压缩方法与早先版本不兼容。如果你已经使用 DOS 6.0 或 6.2 的压缩选项，有关如何恢复文件的详细内容参见第 24 章中“运行 Restore”一节中的问题解答。如果备份文件要恢复到另一台使用 DOS 6 早先版本的计算机上，则不要使用压缩选项。

23.7.5.3 Password Protect Backup Sets(口令保护备份集合)

如果备份保密的数据，可使用口令保护它，这样非授权人就不能通过存取备份磁盘得到你的文件。当选中该选项时，Backup 要求你输入最多为 7 个字符的口令。当输入它时你不能看到该口令，因此必须输入两次，第二次确认第一次的输入。口令与大小写有关，即 Dagwood 与 DAGWOOD 不同。

当对备份赋予一个口令时，一旦想要访问备份文件，必须提供口令，不管是比较它们或是恢复它们都是如此。然而，不需口令就可以删除或覆盖它们。因此，口令并不能保护有些人对数据的故意破坏。

23.7.5.4 Prompt Before Overwriting(覆盖前的提示)

做备份时很容易插错磁盘。一旦选中 Prompt Before Overwriting 选项，如果磁盘已含有数据 Backup 会发出警告。你可以看到磁盘上的数据内容，并且能够选择覆盖它或更换磁盘。

这一步是花费时间的，它使 Backup 花一些时间来分析每一张新的磁盘，然后等待你的响应。如果想使用前面的备份磁盘，而保证不会犯错误，可取消该选项缩短操作时间并更少地涉及用户操作。作者建议总是使用该选项，因为很容易弄混磁盘从而覆盖某些重要的东西。

23.7.5.5 Always Format Diskettes(总是格式化磁盘)

一张新的物理格式化的磁盘总是比要在其上覆盖先前数据的磁盘可靠。当选中 Always Diskettes 选项时，Backup 无条件格式化每一张使用的磁盘。否则，它只格式化未被格式化的磁盘。但是，正如所知道的那样，无条件格式化要用很长时间。等待 Backup 格式化很多张磁盘是一件很枯燥的事。你也许并不认为增加可靠性值得花费这么多时间。

注释： *当你选择备份到 MS-DOS 驱动器和路径时，那些仅与软盘相关的选项是无用的而且不能选择，如* Always Format Diskettes。

23.7.5.6 Use Error Correction(使用错误纠正)

选择 Use Error Correction 选项时，Backup 在备份介质上写入特殊的错误纠正代码。该代码帮助 Compare 和 Restore 功能恢复数据，即使在介质受到损坏时。它不占用很多附加时间或空间，所以大多数人认为它是值得的。作者建议使用它；但是，如果想把备份时间和介质的使用降到最少，可关闭该选项。

23.7.5.7 Keep Old Backup Catalogs(保持旧的备份目录)

Backup 记录它做备份时的目录；目录记载 Compare 和 Restore 功能使用的备份内容。根据缺省设置，如果不删除它们，它们就一直保存在硬盘上。但是，如果禁止该选项，当 Backup 创建新的目录时，它自动地从硬盘删除旧的目录。(Backup 目录将在本章后面“处理备份目录”一节中讨论。)

注释： *Backup 也在备份介质上存储目录的拷贝，所以总能在从硬盘上删除它之后从介质上恢复目录。*

23.7.5.8 Audible Prompts(声音提示)

当备份进行时，如果你呆在一边查看屏幕，你并不需要 Backup 在让你做某些事(如变换磁盘)时发出声音。但是，如果你想做点其他事，应选取 Audible Prompts 选项，这样 Backup 能够呼叫你注意信息。

23.7.5.9 Quit After Backup(备份后退出)

如果想在完成备份后立即退出 Backup，可选中 Quit After Backup 选项。如果作为批处理程序的一部分运行 Backup，如 AUTOEXEC.BAT 或一关机例程，那么这一选项非常有用。如果禁止它，备份完成时，它返回到主 Backup 菜单。

至此你已了解了这些选项，现在应选择那些在你的试验备份中使用的选项：

1. 允许下列选项：

   Verify Backup Data

   Compress Backup Data

   Prompt Before Overwriting Used Diskettes

Use Error Correction on Diskettes

Audible Prompts

检验标记出现在每个允许的选项旁边。

2. 禁止所有其他选项。

禁止的选项没有检验标记。

3. 选中 OK 关闭对话框。

至此已经完成了定义第一个备份 Setup。正如你已看到的，它花费了一些时间，但是你只需对每个 Setup 做一次。从此，只需打开 Setup 并选择 Start Backup 即可。下面将看到如何进行这一工作。

23.7.6 管理 Setup 文件

一旦按需要建立好 Setup，就应保存它以便将来使用。如果已经用 File，Save Setup As 创建了 Setup 文件，所需要做的只是选中 File，Save Setup 保存 Setup 文件的任何改动。改变当前 Setup 后，选择 File，Save Setup 保存这一变化。如果做临时改变，例如只对该备份将备份类型从 Differential 变为 Full，则不必保存该 Setup，当你从 Backup 退出时，改变将会被遗忘。

现在选中 File，Save Setup 保存试验备份的 Setup。

当你开始 Backup 时，根据缺省设置 DEFAULT.SET 被打开。有三种方法打开一个不同的 Setup。可选择 File，Open Setup，并选择想要打开的 Setup。或者，按 Setup File 按钮显示 Setup 文件列表，选择需要的 Setup，并选中 Open 打开它。然而，在启动命令中包括 Setup 名字将更方便，例如：

```
MSBACKUP DAILY.SET
```

注释： 如果在 Setup 文件名中省略扩展名，Backup 假定为.SET，因此可以省略扩展名。

答疑： 当我打开一个 Setup 时，Backup 用很长时间去读未包括在 Setup 中的磁盘驱动器。为什么它做这个，我如何能够使其停止？

有些时候，在 Setup 中包括来自那个驱动器的文件。一旦那样做了，当打开 Setup 时 Backup 总是读该驱动器。当该驱动器是 CD-ROM 驱动器时，这是非常令人恼火的，因为 Backup 迫使你在该驱动器(否则不断重复错误信息)放一张 CD 以便打开该 Setup。

如果涉及 DEFAULT.SET，每次开始 Backup，都要等待读附加的驱动器，除非在开始 Backup 的命令中指定另外一个 Setup。这也是令人不愉快的事。

要想停止读该驱动器，应从 Setup 中特别排除该驱动器。在 Backup From 驱动器列表中(在 Backup 对话框中)，选择该驱动器以一次选择该驱动器中的所有文件，然后再选择它就是退选它们。这样做了之后，在驱动器名旁边应出现一个箭头，但入口不应当显示 Some files 或 All files。

如果为备份建立 DOS Shell 或 Windows 程序项，可把 Setup 文件建立在每个程序项定义中。例如，如果创建一个叫作 Daily Backups 的程序项，可定义该项为 MSBACKUP DAI-

LY. SET 或 MWBACKUP DAILY. SET。

23.8 运行备份

当准备备份文件时，启动 Backup，打开 Setup 文件，检查以确保备份类型是所需要的那种，然后按 Start Backup。如果 Start Backup 按钮变淡，则表示根据当前 Setup，没有要备份的文件，这可能是选择了 Incremental 或 Differential 类型，并且没有需备份的被修改了的文件。

如何运行备份

1. 启动 Backup。
2. 选中 Backup 功能。
3. 打开需要的备份设置。
4. 选中 Select Files，并在必要时调整所选文件。
5. 确认或改变 Backup Type。
6. 选中 Start Backup，并按屏幕上的指示进行操作。
   当执行备份时，要保证写下备份目录的名字。
7. 工作完成时退出 Backup。
8. 标记备份介质并小心存储。

现在试着运行你的试验备份：

1. 操作中需要一些软盘。可重新使用兼容性检测中已使用过的软盘。

警告： Backup 替换磁盘上所有已存在的数据。即使你只备份一个文件，所有已存在的数据也将被破坏。

2. 选中 Start Backup 按钮。
   屏幕上显示进展情况(如图 23.11 所示)。然后，对话框要求你插入 #1 盘。

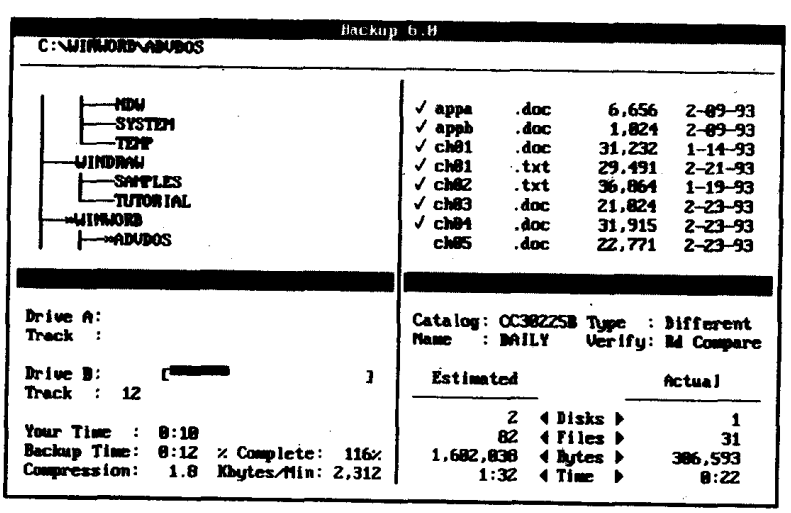

图 23.11　Backup 进展屏幕

3. 从进展屏幕上用一张纸写下目录名字(在图 23.11 中,它是 CC30225B)。以后会需要该目录。
4. 在备份驱动器中插入磁盘并选中 Continue。
   Backup 检查磁盘并在磁盘含有数据时显示一警告框。可替换该磁盘并选择 Retry,或者简单地选择 Overwrite 继续使用当前磁盘。
5. 需要时插入另外的磁盘。
   当所有数据被备份时,Backup 显示一个总结对话框。
6. 选中 OK 关闭总结框。
   返回 Backup 主菜单。

图 23.11 显示的是备份进行中出现的屏幕。正如所看到的那样,该屏幕帮助用户了解备份的进行情况。估计的和实际的时间与空间需求的不同(在右下角)是由备份选项造成的,如压缩和格式化磁盘的需要,Backup 等待更换磁盘的时间,Backup 需纠正的错误等等。

答疑: *Backup 在备份的中间死机,该怎么办?*

这往往是因为设备驱动程序或 TSR 程序与 Backup 相冲突。试一试在没有那些对系统不必要的 TSR 或设备驱动程序的情况下启动系统。例如,可去除鼠标驱动程序,病毒监视程序,FAX-MODEM 驱动程序,CD-ROM 驱动程序等等。但是,如果硬盘驱动器是压缩的,要确保装入压缩软件。

有几种方法可以从启动文件回避这些项:

- *重新启动并按 F8 逐行执行 CONFIG.SYS,以及 AUTOEXEC.BAT。第 21 章"系统配置"讲述怎样做这一工作。*
- *编辑 CONFIG.SYS 和 AUTOEXEC.BAT 在所有不必要的命令前插入 REM。第 20 章"基本批处理程序"讲述怎样做这一工作。*

在成功地运行 Backup 之后,应确保恢复原有的配置。以后每次要运行 Backup 时,必须禁止运行相同的程序。(第 38 章"启动 DOS"将讲述怎样建立 CONFIG.SYS 文件以便在启动时可从多种配置中进行选择)。

23.9 处理备份目录

每次运行备份时,Backup 创建一目录记载日期、时间、设置和备份中包含的文件。在第 24 章中将要学到,Compare 和 Restore 功能都要使用该目录。Backup 在硬盘及备份介质上保存该目录。

图 23.12 表明怎样解释 Backup 赋予每个目录的名字。从目录名中能够辨识出日期、驱动器和备份类型,但是,不能得知所使用的设置。Backup 在硬盘上留有另一个目录,叫做主目录(master catalog),它标识属于每个 Setup 的目录。被称为 Setup.CAT 的主目录只简单地列出属于该 Setup 的所有当前目录的名字。

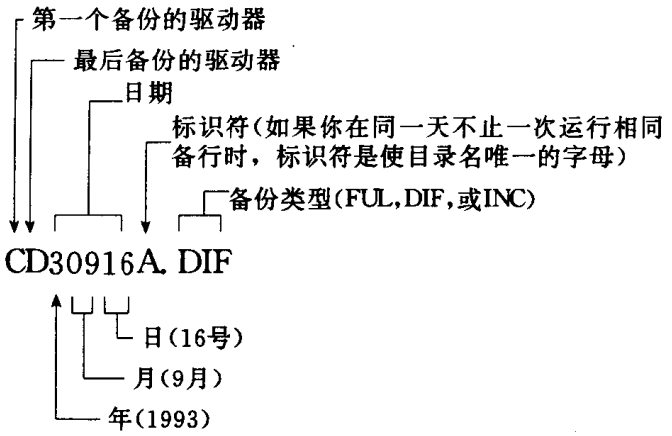

图 23.12　目录名字剖析

一般的主目录内容如下：

```
All word processing files onC
CC50724B. FUL     7-24-95     4:59a
CC50805A. DIF     8-05-95     7:53a
```

第一行总是显示设置说明，从而能够确认正在查看正确的目录。接着，每个目录被逐行列出。此例中，只有其中两个，一个是 1995 年 6 月 24 日的 Full 备份，另一个是 1995 年 5 月 5 日的 Differential 备份。

看一看到目前为止所创建的主目录：

1. 退出 Backup，保存所有 Setup 的变动。
2. 键入 TYPE C:\DOS\PRACTICE.CAT 在监示器上显示该目录。
   如果得到 File not found 信息，则表示目录没有保存在 DOS 目录中，这可能因为你是在网络上。不必着急看 PRACTICE.CAT。很快将学习怎样快速定位你的目录。

每次 Backup 为 Setup 建立一个新的目录时，它把目录名添加到 Setup 的主目录中。如果你自己维护目录，当你删除旧的目录时，也应该从主目录删除它们的名字。(如果禁止 Keep Old Backup Catalogs，Backup 将自动从主目录删除它们)。

23.10　保存备份磁盘

一旦进行了备份，就应该标记备份磁盘并把它们放在安全的地方。标签上应包括 Setup 名字和目录名字(如果发现目录名字难于解释，还可在标签上写上日期和备份类型)。现在试着标记你的试验备份磁盘：

1. 用下列信息标记每一张磁盘：
   Backup Setup Name: PRACTICE.SET
   Catalog Name:xxxxxxxx.FUL
   (可选)Date:xx/xx/xx
   (可选)Backup Type:FULL

注释： 如果做备份时忘记写下目录名字，可从备份介质中找到该名字。详见本章后面的“备份文件的管理”。

把该备份磁盘保存在哪里？作者建议进行复制拷贝。把一组放在靠近计算机的地方以便在需要恢复某些文件时可以很容易恢复。另一组保存在防水的装置中，或把它放到其他地方，这样就不会在出现火灾、水灾或其他灾害时丢失所有的备份。

有时在修改一个文件后，你可能又改变了主意。例如，可能在电子表格中输入一个错误的图形或把一个错误的更新记入数据库。如果有规律地做备份，从备份中加以恢复可以很容易地返回到文件的早先版本。如果几天前做了错误的修改就必须返回到早先的备份来恢复该文件。如果没有保存旧的备份，就不能够得到未修改的文件。

你还能遇到想恢复一个很久以前删除的文件的情形，这时必须从删除以前做的备份中恢复它。可能需要返回到几个月前的备份去找该文件。如果不保存旧的备份，就不能恢复必要的文件。

如果希望能够恢复文件的早期版本，或恢复被删除的文件，比如说 6 个月前的文件，那么应该保存旧的备份介质。我们并不建议你保存许多的备份，但至少应考虑保存一段时间。必须自己决定什么方法最适合于你的数据。

如果你决定保留很长一段时间旧的备份，就不必在硬盘中保留它们的目录。Backup 能够从备份介质恢复目录。

作者忠告： 直到有了下一个更好的备份时，才删除原先的备份。

既使你从不需要返回到早先的文件，也应该在完成了一新的更好的备份以后，再覆盖旧的备份盘。例如，如果每星期一上午做一次全备份，直到第三个星期一之后才可使用第一个星期一的备份盘。也就是说，如果第二个星期一的备份不能使用，仍可返回到第一个星期一的备份。类似地，如果每天做一个不同的备份，直到星期四才可以重新使用星期二的备份盘，这样当星期三的备份不能使用时仍可返回到星期二的备份。

23.11 备份文件的管理

Backup 在硬盘上建立. SET、. SLT、. CAT、. FUL、. INC 和. DIF 文件。在备份介质上，它在称为 catalog. nnn 的文件中尽可能多地装入备份文件，这里 catalog 是目录名，nnn 是序列号。例如，集合中第一个或仅有的文件可能叫作 CC50107A. 001。

如果备份到 DOS 目录，则只创建一个备份文件，名字叫作 catalog. 001。所有备份的文件都组合在这个文件中(用 Restore 和 Compare 可以解开它们)。比如，指定的备份目录叫作 JNFBACK，Setup 为 DAILY，且目录名为 CC50105A. FUL，Backup 建立下面的分枝：

```
JNFBACK (the specified directory)
¦
¦-- DAILY (subdirectory)
        ¦-- CC50105A. FUL (subdirectory)
                ¦-- CC50105A. 001 (the backup file)
```

当你想删除旧的备份时，该分枝结构使你能方便地找到备份文件。从两个不同的 Setup 备份到相同路径名几天以后，你的目录树就会像如下的情形（备份文件没有显示，只有子目录）：

```
JNFBACK
|
|-- DAILY
|       |-- CC50105A.FUL
|       |-- CC50106A.DIF
|       |-- CC50107A.DIF
|
|-- SPECIAL
|       |-- CC50105B.FUL
|       |-- CC50106B.INC
|       |-- CC50107B.INC
```

正如所看到的，要找出并删除 1995 年 1 月 6 日所做的 Differential 备份非常容易。

如果备份到软盘，Backup 对每张盘创建一个文件，分别命名为 catalog.001；catalog.002 等等，并且在第张盘中装入尽可能多的文件。它填充每张盘到最后一字节，必要时它把最后的文件放到下一张磁盘。只有最后一张磁盘可能没有完全装满数据。像 PRACTICEFUL 那样，Backup 对每张盘赋予一个 Setup type 标签。

23.11.1　Setup 和目录保存在哪里

根据缺省设置，Backup 把目录和 Setup 文件保存在备份程序所在的目录中，在非网络系统上一般是 C:\DOS。如果不在网络上，并且你是该计算机的唯一用户，就会发现让 Backup 使用缺省目录是最方便的。

如果你与他人共享计算机，可能想把你的目录和 Setup 文件保存在不同的目录中，这样在使用 Backup，Compare 或 Restore 时，每个人就只能看到他们自己的文件。通过设置叫作 MSDOSDATA 的环境变量就可以做到这一点，它标识所有 Backup 文件（以及 Microsoft Anti-Virus 文件，这将在第 25 章“DOS 的反病毒程序”中学习）的缺省目录。例如，要为叫作 C:\JNFBACK 的目录设置 MSDOSDATA，可键入下列命令：

```
SET MSDOSDATA=C:\JNFBACK
```

当 MSDOSDATA 完成设置后，Backup 在指定目录中自动寻找欲打开的目录或 Setup 文件，并且自动地在那里保存新的 Setup 或目录。

注释： *在第 43 章“DOS 及其运行环境”中将学习怎样使用诸如 MSDOSDATA 的变量。在第 38 章将学习怎样在启动期间，根据进行启动的人的不同设置 MSDOSDATA。*

如果在网络上，可能需要使用 MSDOSDATA 为你的 Setup 和目录文件访问不同的驱动器或目录。例如，你的备份软件可能在一个只允许读和执行的驱动器上。网络管理人员或工作组负责人应该能告诉你如何设置 MSDOSDATA。实际上，你的 AUTOEXEC.BAT 可能已经建立了起来，在每次起动时它会正确地定义环境变量。

你能够很容易地找出备份程序使用的缺省目录：

1. 启动备份程序并选中 Backup 功能。
2. 选中 File,Open Setup。
   Open Setup 对话框打开,列出根据缺省设置可用的 Setup。
   缺省目录在对话框的顶部,Dir:的旁边。

23.11.2 Setup 文件的打印

.SET 文件包含除了所选择或退选的单个目录和文件以外的全部设置信息,这些目录和文件保存在.SLT 文件中。使用 File,Print Setup 命令可打印.SET 文件,但不能打印.SLT 文件,因为它不是 ASCII 文本文件。

注释: 如果使用 MWBACKUP 而不是 MSBACKUP 打印 Setup,可选择图形或文本模式的打印输出。

现在试着打印 PRACTICE.SET 文件:

1. 确保打印机已准备好。
2. 启动 Backup 并打开 PRACTICE 设置。
3. 选中 File,Print Setup。
   开始打印 Setup。

在 Setup 文件中你能够识别许多项,因为它们正是在定义 Setup 中你所选择的结果。例如,可以看到表明允许错误纠正选项的行:

```
Error correction:          Yes
```

在 Setup 文件中还有不能从 Backup 控制的变量。例如,会发现影响文件列表外观的行:

```
Show File Size:          Yes
Show File Data:          Yes
```

可编辑.SET 文件,使用 EDIT 这样的 ASCII 文本编辑器改变这些变量设置,它不能用其他方式改变。

注释: Setup 文件的 MWBACKUP 版本含有一些不同于 MSBACKUP 版本的变量。

注释: 如果要编辑.SET 文件,应保留一个备份拷贝,这样在你不喜欢改变的结果时还可以返回到原来的版本。

23.11.3 删除 Setup

Backup 的 File 菜单含有删除备份 Setup 的命令。.SET 和.SLT 文件可用一次操作加以删除。不能删除当前的 Setup,因此必须保证在删除一个 Setup 之前有另外的 Setup 被打开。试一试删除在本章中建立的 PRACTICE Setup:

1. 确保 PRACTICE 不是当前的设置。如果是,打开 DEFAULT 设置。
2. 选中 File,Delete Setup。
   Delete Setup 对话框打开,列出缺省目录中所有的 Setup。

3. 选择想要删除的 Setup。

警告： *该功能删除具有检验标记的 Setup。一定要双击该 Setup 名字，或高亮显示它并按空格键选择它。在继续之前它的旁边应当有一个检验标记。*

4. 选中 Delete 按钮。

   一个对话框要求你确认选择。确保对话框显示的是 PRACTICE Setup。如果不是，选中 Cancel 并返回到第 2 步。

5. 选中 Delete 删除该 Setup。

23.11.4　删除目录

Keep Old Backup Catalogs 选项控制在建立新的目录时，Backup 是否自动删除原有的目录。如果禁止该选项，Backup 根据下面规则删除目录：

- 当制作一个新的全备份时，所有该设置的旧目录均从缺省目录中删除，并且它们的入口也从主目录中删除，只保留新的全备份目录。
- 当制作一个 Differential 备份时，所有先前的该设置的 Differential 目录均从缺省目录中删除，并且它们的入口也从主目录中删除。只保留全备份目录和新的 Differential 目录(如果有该设置的 Incremental 目录，它们并不被删除)。
- 当制作一个新的 Incremental 备份时，不删除任何目录。

如果使 Keep Old Backup Catalogs 选项有效，Backup 不自动删除任何目录。要删除目录时，必须自己去做。

当删除备份本身时，应删去相应的目录，另一方面，如果要将备份保留几个月，其目录不必保留这么长时间，删除一个目录不会影响恢复文件。如果目录从硬盘上删除之后需要从备份集恢复一个文件，可从备份集本身恢复目录。用户必须自己决定将目录保留多长时间。当删除一个目录时，不要忘记从主目录中删除它的入口，这可用 EDIT 进行编辑。

23.12　通过软盘拷贝大型文件

如果需要把一个相当大的文件拷贝转移到另一台计算机中，假定即使使用数据压缩也不能用一张软盘容纳这个文件。再假定也不能将两台计算机连到一起，用网络连接，或通过调制解调器传送该文件，并且这两台计算机只有通常的软驱，没有可移动的硬盘或其他高容量设备。你必须找出通过软盘转移该文件的方法。

COPY 和 XCOPY 都不能把一个文件分别装入两张以上的磁盘，但 Backup 能够。所需做的只是备份文件到一个软盘驱动器，然后恢复该文件到另外一台计算机。

如果打算使用压缩选项，要保证两台计算机使用互相兼容的 Backup。也就是说，如果计算机 A 使用 DOS6.0 或 6.2，计算机 B 也必须使用 DOS 6.0 或 6.2，而不是 DOS 6.22(除了在第 24 章“运行 Restore”一节的问题解答中讲述的情形以外)。如果计算机 A 使用 DOS 6.22，计算机 B 也必须用 DOS 6.22。如果 DOS 版本不兼容，不要使用压缩选项。

如何通过软盘把大型文件拷贝到另一台计算机上

1. 启动 Backup。
2. 选中 Backup 功能。
3. 创建一个新的 Setup。(不要使用 Default Setup,否则 Backup 在每次从现在开始备份时都要读整个驱动器。见本章前面"Setup 文件的管理"一节中的问题解答。)
4. 对新的 Setup,只选择想要拷贝的文件。还要选择要拷贝到的软驱,以及全备份类型。

注释:不要使用压缩选项,除非计算机使用兼容的 Backup 版本。

5. 开始备份并按屏幕上的指示进行操作。
6. 在另一台计算机上,启动 Backup。
7. 选中 Restore 功能。
8. 从磁盘上恢复目录。
9. 因为目录只包含一个文件,可在 Restore Files 框中简单地选择整个驱动器。
   对话框应显示有一个文件被选择恢复。
10. 选择选项把文件恢复到相同的驱动器和目录,或按需要到不同的驱动器和目录。
11. 选中 Start Restore 并按屏幕上的指示进行操作。

23.13 自测练习

下一章将讲述怎样从备份比较和恢复文件。除了本章中做的以外,还需要制作一些备份进行练习。下面几项可帮助你创建必要的备份:

1. 把所有文件从 EXER1 拷贝到 PRACTICE,重新命名它们为.XXX 文件。(替换已有的同名文件。)
2. 重新定义 PRACTICE Setup,用 1/1/94 和 1/1/95 之间的日期备份 PRACTICE 分枝中所有文件。排除 EXER1 目录中名字以 C 开始的文件。
3. 再次运行 PRACTICE 备份,但把备份类型改为 Incremental。
4. 把所有文件从 EXER2 拷贝到 PRACTICE,重新命名它们为.XXX 文件。(替换已有的同名文件)。
5. 运行另一个 Incremental PRACTICE 备份。
6. 保留全备份磁盘和两次 Incremental 磁盘以便在下一章中使用。

第 24 章　Restore 和 Compare

简介： 可使用备份来恢复被删除的文件，或从病毒侵犯或硬盘崩溃的系统中恢复文件等等。

24.1　恢复和比较

有一组好的备份集合，就为几乎任何类型的硬盘急救做好了准备。假如错误地删除一个分枝并且不能恢复它，可从备份恢复整个分枝——目录和文件；假如修改一个电子表格但随后决定取消修改，可从备份恢复未修改的版本；假如硬盘坏了而必须买一个新的，在格式化之后能把所有备份的目录和文件恢复到新盘上。

Compare 功能在它检测到一个文件的备份版本与硬盘上的版本有差别时通知用户。如果在备份或恢复时没有使用验证选项，那么验证文件不失为一个好方法。一般情况下，验证选项是更好的选择，因为它在备份进行时标识和纠正错误。Compare 只能够指出问题，必须采取其他步骤（如重新做备份或恢复）解决它。

Compare 还能识别一个文件自从备份以来是否被移动或删除。它能判断备份文件是否仍在原始硬盘驱动器目录中。

还可用 Compare 来确定自上次备份以来文件是否被修改或损坏。Compare 能够确定不同计算机上的文件是否一致；备份计算机 A 上的文件，然后把它们与计算机 B 上的文件相比较。（如果能够用某种方法连接两台计算机，用 FC 比较文件会更容易。当没有可能连接或并网这两台计算机时，可用 Backup 和 Compare。）

按照下面的步骤比较或恢复备份中的一个文件，一组文件，或全部文件。打开备份集合目录，选择要恢复或比较的文件，并响应提示。

本章主要讲述：

- 打开一个备份集合目录
- 选择要恢复或比较的文件
- 在一个文件的几个版本中进行选择
- 选择 Restore 或 Compare 选项
- 进行 Restore 或 Compare 工作

24.2　从 DOS 早期版本制作的备份恢复文件

如果需要从更新 DOS 之前做的备份中恢复文件，会遇到一些问题。如果曾用 DOS 5 或更早版本做了备份，然后升级到 DOS 6 的某一版本，不能使用 MSBACKUP 或 MWBACKUP 从旧备份中恢复文件。

有关怎样从这些备份恢复文件的说明，参见本章后面的“从 DOS 6 以前做的备份恢复文件”一节。

如果使用 DOS 6 或 6.2 制作备份，随后升级到 DOS 6.22，如果下面条件有一个为真，从早先的备份中恢复不会有任何答疑：

- 做备份时没有使用压缩选项(压缩选项在 Backup 的早先版本中是缺省的，除非特意关闭压缩选项，否则文件将被压缩)。
- DOS 升级时 Backup 程序未同时升级。
- 在 DOS 6 或 6.2 下安装了 DblSpace 并且没有把压缩驱动器升级到 DrvSpace。
- 旧的 Backup 程序在硬盘上仍然可以使用，并且能够运行除 DOS 6.22 以外的 Backup 版本(例如，它可能在 OLD_DOS.1 中)。
- 在 DOS 目录中，仍然有 DBLSPACE.BIN，并跟有一个叫做 DRVSPACE.MR1 的新文件(DRVSPACE.MR1 只包含在 DOS 6.22 非英语版本中)。

如果这些条件没有一个为真，要从早先备份中恢复文件将需要采取一些附加措施。有关具体步骤的说明参见本章后面“运行 Restore”一节中的问题解答部分。

24.3 恢复文件

在 MSBACKUP 的主菜单中，或在 MWBACKUP 对话框中选择 Restore 按钮即可启动 Restore。本章假定使用的是 MSBACKUP。如果使用的是 MWBACKUP，其差别也能很快适应。现在尝试启动 Restore：

1. 如果备份程序还没有开始，启动它。
2. 选中 Restore 按钮。

Restore 对话框打开(如图 24.1 所示)。

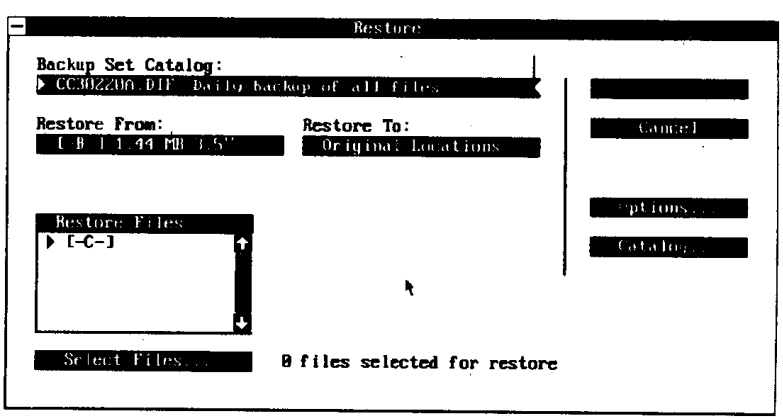

图 24.1 Restore 对话框

24.3.1 打开目录(Catalog)

Backup Set Catalog 按钮显示当前备份目录的名字和说明。缺省目录为最近所建立的。

如果不想使用该目录，可以从目录列表中选择另一个。

试着打开第 24 章“MSBACKUP 和 MWBACKUP”中建立的 PRACTICE 备份的目录（如果 PRACTICE 目录已经打开，仍可做一下这个练习以学习怎样打开另一个目录）。

1. 选中 Backup Setup Catalog 按钮（它是一含有当前备份目录名字的方框）。
   Backup Set Catalog 对话框打开（如图 24.2 所示）。每一个目录名旁边的说明可帮助识别标记目录的 Setup。可在列表中看到主目录（如 DAILY.CAT）和每一个目录。

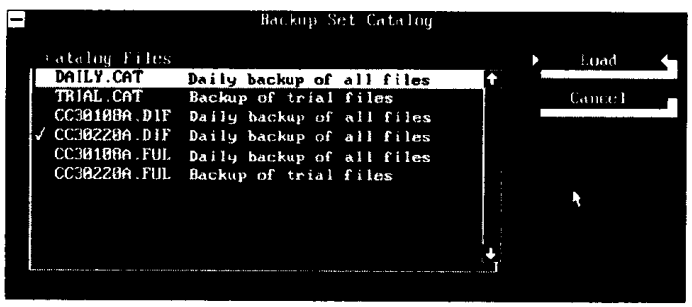

图 24.2　Backup Setup Catalog 对话框

   注意当前打开的目录旁边有一个选中标记。
2. 找出使用 PRACTICE Setup 做的最后一次备份的目录。它的名字应该是 CCxxxxxx.INC（不要使用主目录，将在后面加以说明）。
3. 要打开目录，你必须双击它，或高亮度显示它并按空格键。选中标记移动到选择的目录，表明它现在是打开的目录（如果选中标记不移动，再试一次）。
4. 选中 Load 完成对话。
   对话框关闭并且 Restore 对话框返回。对话框中的信息（如驱动器列表和 Restore From 框）反映打开目录的内容。目前为止，没有文件被选择并且 Start Restore 按钮是暗的。

如何打开一个目录

1. 在 Restore 对话框中，按 Backup Set Catalog 按钮。
   Restore 列出所有目录。
2. 通过双击或高亮度显示并按空格键选择一个目录。
   Restore 在所选目录旁边放上一个检验标记。
3. 选中 Load 按钮。

24.3.2　选择文件

当选择欲恢复的文件时，不一定要处理包括/排除列表，可在 Restore 屏幕上选择整个驱动器，或按 Select Files 按钮操作驱动器列表、目录树和文件列表。

答疑： *Restore 或 Compare 驱动器列表中未显示所有硬盘驱动器。为什么它不能找到我的所有硬盘驱动器？*

Restore 和 Compare 只显示那些受当前备份目录影响的驱动器。例如，如果当前目录只含有从驱动器 C 和 D 备份的文件，则只有这两个驱动器出现在驱动器列表中。

24.3.2.1 选择整个驱动器

当选择整个驱动器时，备份目录中该驱动器的所有目录和文件被选择。如果有不止一个驱动器出现在 Restore Files 框中，可以选择它们中的任何一个或全部。当选择一个驱动器时，并不是选择当前在该驱动器上的所有文件，而是选择从该驱动器备份的并且出现在备份目录中的所有文件。

如何选择欲比较或恢复的文件

选择一个完整的驱动器：

1. 双击 Restore Files 框中的驱动器名字，或高亮度显示驱动器名字并按空格键。

选择或退选一个目录：

1. 选中 Restore 对话框中的 Select Files。
   Select Restore Files 对话框打开。
2. 在目录树中，双击该目录名，或高亮度显示它并按空格键。

选择或退选单个文件：

1. 在 Restore 对话框中选中 Select Files。
   Select Restore Files 对话框打开。
2. 在目录树中，高亮度显示该目录。
   目录的内容列在文件列表中。
3. 双击每一个文件名，或高亮显示它并按空格键。

根据日期范围排除文件：

1. 在 Restore 对话框中选中 Select Files。
   Select Restore Files 对话框打开。
2. 选中 Special。
   Special Inclusions 对话框打开。
3. 使 Apply Date Range 有效。
4. 在 From 和 To 框中键入日期范围。

排除拷贝保护的文件：

1. 在 Restore 对话框中选中 Select Files。
   Select Restore Files 对话框打开。
2. 选中 Special。
   Special Inclusions 对话框打开。
3. 使 Exclude Copy Protected Files 有效。
4. 在 Exclude Copy Protected Files 右边的正文框中最多键入 5 个文件名(它可以是普遍的或特定的)。

根据属性排除文件：

1. 在 Restore 对话框中选中 Select Files。

   Select Restore Files 对话框打开。

2. 选中 Special。

   Special Inclusions 对话框打开。

3. 根据需要使 Exclude Read-Only Files、Exclude System Files 或 Exclude Hidden Files 有效。

试一试选择所有从驱动器 C 备份的文件：

1. 如果有鼠标，双击驱动器列表中的驱动器 C。如果使用键盘，按 Alt＋I 选择 Restore Files 框。因为只有一个驱动器出现在框中，它是自动高亮度显示的。按空格键选择它。

   一个实箭头和词组 All files 表明该驱动器上所有文件已被选择。可看到选择的文件数的改变并且 Start Restore 按钮可使用。

2. 重复步骤 1 退选该驱动器。

   箭头消失，选择的文件数变为 0，并且 Start Restore 按钮变暗。

24.3.2.2　选择或退选单个文件

选中 Select Files 按钮时，可逐个选择要恢复的目录或文件。如果首先选择一整个驱动器，可使用 Select Files 退选指定的目录或文件。

试着选择一些要恢复的文件：

1. 选中 Select Files 按钮。

   Select Restore Files 对话框打开(如图 24.3 所示)。即使备份的文件只来自 PRACTICE 分枝，目录树也显示备份时硬盘驱动器的整个目录结构。

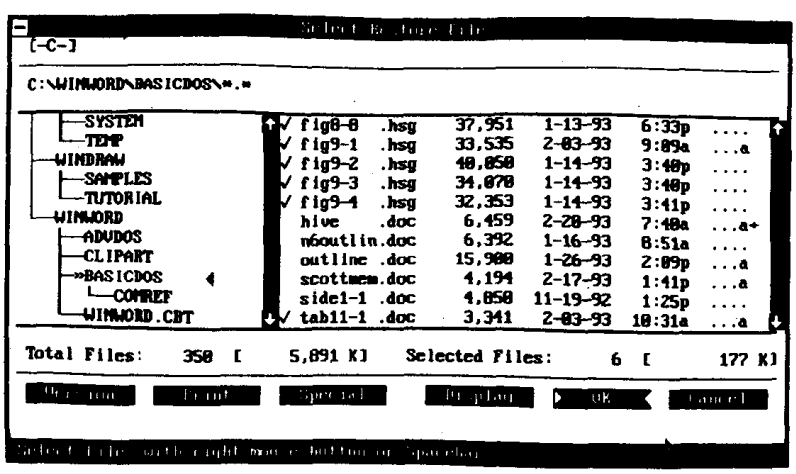

图 24.3　Select Restore Files 对话框

注释： 驱动器列表、目录树和文件列表均取自备份目录。它们反映被备份的目录结构和文件，而不是硬盘驱动器的当前结构。

2. 通过按上下箭头键或通过单击不同目录在目录树中移动高亮度显示条。
   注意大多数目录中没有文件，这是因为没有从这些目录备份任何文件。
3. 现在移动高亮度显示条到 PRACTICE 目录。
   文件列表显示从该目录备份的所有文件。
4. 双击 PRACTICE 目录，或按空格键选择整个目录。
   一个实箭头出现在目录树中该目录名字的旁边；并且右下角的 Selected Files 域中显示已经选择的文件数。
5. 移动高亮度显示条到 EXER1 目录。
6. 双击两个文件。如果使用键盘，按 Tab 键把高亮度显示条移到文件列表中，按下箭头键高亮度显示一个文件，并且按空格键选择它。
   检验标记出现在所选文件的旁边，一个 V 形标记出现在目录树中目录名字的旁边，并且右下角的 Selected Files 域显示已经选择的文件总数。

提示： 可选中 Print 按钮打印该目录文件的内容。

7. 选中 OK 返回到 Restore 对话框。
   注意词组 Some files 出现在驱动器 C 的旁边。还能看到所选择的文件数。

注释： 与 Backup 功能不同，Restore 并不保存文件选择。每次启动 Restore，必须重新选择想要恢复的文件。

24.3.2.3 恢复文件旧版本

当打开主目录时，一个文件可能有几个版本。例如，假定在一个星期的每一天都修改 HIVE.DOC，每天做 Incremental 备份，在主目录中就会有 HIVE.DOC 的多个版本。

根据缺省设置，文件列表只在目录中显示每个文件的最新版本，但该文件旁边的加号表明还有多个版本可供选择。Version 按钮，只适用于主目录，可用来打开文件列表中当前高亮度显示的文件的版本列表，允许你想要选择的版本。

如果你做了第 23 章中的所有练习，在 PRACTICE 主目录中起码有一些文件有两个版本。现在看一看这些文件：

1. 打开 PRACTICE 主目录。
2. 选中 Select Files。
3. 选中 PRACTICE 目录。
   注意一些文件旁边的符号＋。
4. 高亮度显示带符号＋的文件。
   Version 按钮变为可选。
5. 选中 Version 按钮。
   一个对话框显示目录中所有版本列表。通过它们的日期和时间可识别版本的不同。
6. 通过双击或高亮度显示并按空格键选择最早的版本。
   在选择的版本旁边出现一个检验标记。
7. 选中 OK 关闭 Version 对话框。
   选中的版本出现在文件列表中并被选择恢复。

如何返回到文件的早期版本

1. 启动备份程序并选中 Restore。
2. 打开备份集合的主目录。
3. 选中 Select Files。
   Select Restore Files 对话框打开。
4. 高亮度显示含有该文件的目录。
5. 高亮度显示该文件。
   如果有符号＋出现在文件旁边，则在目录中有该文件的几个版本存在并且 Version 按钮可选。
6. 选中 Version。
   Version List 对话框打开。
7. 双击想要的版本，或高亮度显示它并按空格键。
   一个检验标记出现在所选择版本的旁边。
8. 选中 OK。
   选中的版本出现在文件列表中。
9. 选中 OK 返回到 Restore 对话框。
10. 选中 Start Restore。
    Restore 告诉你插入哪一张盘。

24.3.3　Restore 选项

Restore 提供几个选项确保恢复成功。但是，在考虑选项之前，应先看一看当前打开的是哪一个备份 Setup，因为任何选项的改变都保存在当前 Setup 中。

24.3.3.1　从 Restore 窗口改变 Backup Setup

启动 MSBACKUP 时，除非在启动命令上包括另一个 Setup 文件名，否则就根据缺省设置打开 DEFAULT.SET。如果直接进入 Restore 功能，在这里看不到备份 Setup 的名字，也许你会认为当前备份 Setup 与当前备份目录有关，但并不总是如此。

打开一个目录并不会打开其 Setup。例如，可能有 DEFAULT.SET 作为 Setup 文件打开，但 DAILY.CAT(从 DAILY Setup)作为目录文件打开的情形。在这种情况下，对选项所作的任何改变都保存 DEFAULT.SET，而不是 DAILY.SET 中。

如果你大多数时间使用相同的恢复选项，不管涉及到什么目录，则可把它都保存在 DEFAULT.SET 中。这样，就不必为恢复选项打开另外一个 Setup。但是，要使恢复选项与特定的 Setup 相关，就必须在设置选项前打开该 Setup，当要使用这些选项时再次打开它。

从 Restore 对话框打开一个 Setup 与从 Backup 对话框打开一个 Setup 稍有不同。因为在对话框中没有 Setup，必须使用 File 菜单。

尝试打开 PRACTICE Setup：

1. 选中 File，打开 Setup。
   Open Setup 对话框打开。
2. 双击 PRACTICE Setup 文件，或高亮度显示它并按空格键。
   检验标记出现在所选择的 Setup 文件旁边。

3. 选中 Open 打开 Setup。

如果在其他 Setup 打开时改变 Restore 选项，Restore 询问是否想保存对该 Setup 的改变。

你可以回答 yes 保存它们，或 No 放弃它们。然后，新的 Setup 打开并对 Restore 选项建立自己的设置。

现在准备为已经打开的 Setup 选择选项：

1. 选中 Options 按钮打开 Restore Options 对话框(如图 24.4 所示)。在浏览选项时保持其处于打开状态。

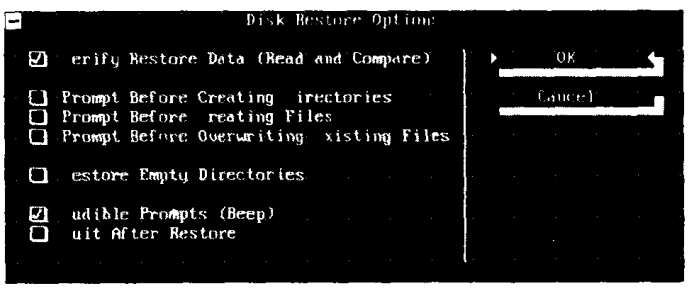

图 24.4 Disk Restore Options 对话框

24.3.3.2 Verify Restore Data(验证恢复数据)

当 Verify Restore Data 有效时，Restore 花时间把每一个恢复的文件与备份的版本相比较并纠正所发现的错误。

注释： Compare 功能可以在恢复完成后使用，来验证恢复的文件，这将在本章后面的“Compare”一节中讲述。

24.3.3.3 Prompt before Creating Directories(创建目录前提示)

当向一个不再存在的目录恢复一个或多个文件时，Restore 为恢复的文件重新创建该目录。当 Prompt before Creating Directories 无效时，Restore 创建目录时没有警告。允许该选项时 Restore 则要求用户确认每一个它必须创建的目录。如果选中 No，Restore 跳过属于该目录的文件。

24.3.3.4 Prompt before Creating Files(创建文件前提示)

假定想要恢复从目录中删除的文件。根据缺省设置，Restore 把文件返回到该目录并且没有警告。但是如果使 Prompt before Creating Files 有效，当 Restore 必须在目录中创建一个文件时，将请求用户的允许。如果选中 No，Restore 跳过该文件。

24.3.3.5 Prompt before Orerwriting Existing Files(覆盖已有文件前提示)

当一个文件被恢复时，如果目标目录已经含有一个相同名字的文件，Restore 将用恢复的文件覆盖已存在的文件。禁止 Prompt before Overwriting Existing Files 可使 Restore 覆盖已有的文件并且没有警告。当允许该选项时，Restore 在覆盖一个文件时将请求用户的允许。如果选中 No，备份版本不被恢复。

24.3.3.6　Restore Empty Directories(恢复空的目录)

Backup 总是记录一个驱动器的整个目录结构,即使有些目录不包含要备份的文件时也是如此。使 Restore Empty Directories 有效,可重新创建一个驱动器先前的目录结构,包括空目录。如果禁止该项,Restore 只创建恢复文件所在的目录。

24.3.3.7　Audible Prompts(音响提示)

就像 Backup,Restore 为备份介质提供的提示一样,根据所选择的选项,它也可在显示其他提示时发出声响。如果想让 Restore 在显示提示时发出声响,应使 Audible Prompt 有效。

24.3.3.8　Quit after Restore(恢复后退出)

如果使 Quit after Restore 有效,当恢复完成时备份程序自动终止,返回到启动备份时正在使用的程序。如果没有使该选项有效,当恢复完成时返回到 Restore 对话框。

24.3.3.9　恢复选项的选择

应使 Disk Restore Options 对话框在屏幕上打开。试一试为 PRACTICE Setup 选择选项:

1. 使下列选项有效:

   Verify Restore Data
   Prompt before Creating Directories
   Prompt before Overwriting Existing Files
   Audible Prompts

   检验标记出现在允许选项的旁边。
2. 禁止所有其他选项。
   被禁止的选项没有检验标记。
3. 选中 OK 关闭对话框。

现在通过选中 File Save Setup 可在 Setup 中保存这些选项;或者等待直到退出 Restore,或者切换到另一个 Setup。如果想退出或改变 Setup 并且不保存选项的改变,Restore 将询问是否想把它们保存到当前 Setup(该信息告诉你 Setup 的名字)。你所需要做的只是选中 Yes 保存它们或者选中 No 放弃它们。对这一练习,可等待直到退出保存选项。

24.3.4　运行 Restore

在打开目录,挑选文件并选择 Restore 选项后,准备运行 Restore。

1. 按 Start Restore 开始恢复文件。
   Restore 以目录名字请求备份介质;如果涉及软盘,它还指出特定的磁盘号。例如,它可能要求备份集合 CC50510A. FUL 的 #2 盘。
2. 在指定驱动器中插入要求的磁盘并选中 Continue。
   图 24.5 显示的是恢复期间进展情况框。上半部分是 Restore 当前正在工作的目录和文件。下半部分显示 Restore 的进展情况和结果。

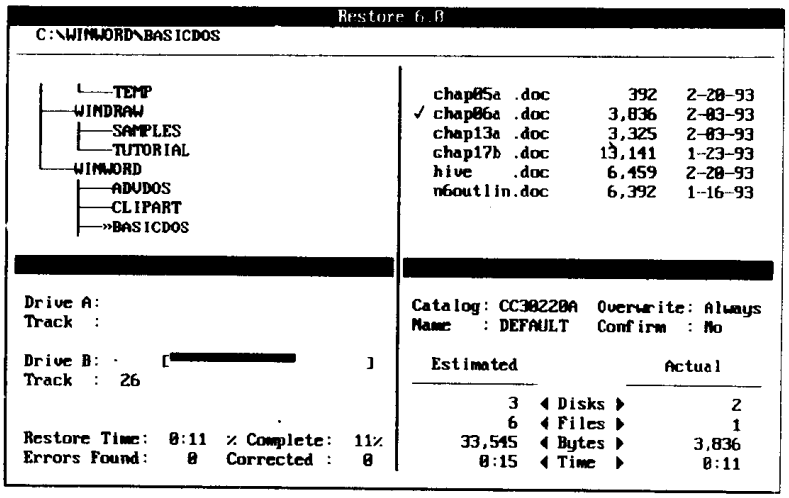

图 24.5 Restore 进展框

注释： 如果 Backup 在备份磁盘上保存错误校正码，Restore 能够检测并校正由于坏扇区，读/写错误和其他物理问题引起的错误。

最后，一个总结框显示有多少个文件被恢复，多少个文件被跳过(例如，由于选择不覆盖一个已存在的文件)，备份文件中有多少错误被检测和校正，如此等等。

3. 选择 OK 关闭对话框。
4. 退出备份程序。

Restore 询问是否保存对当前 Setup 的改变。

5. 选中 Yes 把当前 Restore 选项保存在 PRACTICE Setup 中。

答疑： 我想恢复用 DOS 6.0 或 6.2 备份的某些文件，但是 Restore 说它不能读压缩的文件。

DOS 6.22 使用与 DOS 6.0 和 6.2 不相同的压缩方法。除非仍然在磁盘驱动器上使用 DblSpace，否则它不能够读在早期 DOS 版本下备份的文件。可以使用下列方法恢复旧的备份：

- 如果在磁盘驱动器上仍然有 OLD_DOS.1 目录，切换到该目录并尝试运行 MSBACKUP。如果能运行 MSBACKUP 的先前版本，它就能够读旧的备份文件。
- 如果仍然有 DOS 先前版本的安装盘，可把先前程序文件拷贝到硬盘驱动器并运行先前的备份程序。有些必要的文件压缩在安装盘上，必须使用 EXPAND 程序把它们拷贝到硬盘驱动器。按下面步骤进行：

1. 创建一个叫作 OLDBACK 的目录(或其他你喜欢使用的名字)。
2. (对 1.2MB 盘上的 MS-DOS 6.2，继续到第 3 步。)对 1.44MB 盘上的 MS-DOS 6.0 或 MS-DOS 6.2，在相应驱动器中插入旧 DOS 版本的 Setup 磁盘，并输入下面命令，这里 d 是含有 Setup 盘的驱动器名字：

```
XCOPY d:*.OVL C:\OLDBACK
EXPAND d:MSBACKUP.EXE C:\OLDBACK
```

```
EXPAND d:MSACKUP. HLP C:\OLDBACK
EXPAND d:MSBCONFG. HLP C:\OLDBACK
```

3. 对 1.2MB 磁盘上的 MS-DOS 6.2，在相应驱动器插入 Setup Disk 2 磁盘并输入下面命令，这里 d 是含有 Setup 磁盘的驱动器名字：

```
XCOPY d:*.OVL C:\OLDBACK
EXPAND d:MSBACKUP. EXE C:\DOS
```

然后，在驱动器中插入 Setup Disk 并且输入下面命令：

```
XCOPY d:*.OVL C:\OLDBACK
EXPAND d:MSACKUP. HLP C:\OLDBACK
EXPAND d:MSBCONFG. HLP C:\OLDBACK
```

4. 切换到 C:\OLDBACK 并启动 MSBACKUP。

从现在起，当你想恢复由先前 DOS 版本备份的文件时，切换到 C:\OLDBACK 并从这里启动 MSBACKUP。当你想恢复由 DOS 6.22 备份的文件时，要确保不在 C:\OLDBACK 中；从其他目录启动 MSBACKUP 或 MWBACKUP。

5. 如果没有先前 DOS 版本的安装盘，但硬盘上有 DBLSPACE. BIN 和 DRVSPACE. MR1 文件，仍然可以从旧的压缩备份恢复。如果你的 DOS 6.22 没有 DRVSPACE. MR1，要想得到它，请寄回“DOS 用户指南”后面的赠券。

24.3.5　Restore 的一般使用方法

可能经常需要使用 Restore 恢复多个文件的最新版本。例如，假定错误地删除了一个目录中所有的.DOC 文件，并且不能可靠地使删除无效。就需要用 Restore 恢复它们。所要做的是启动 Restore，打开校正目录，选择所需文件，选取需要的选项，并选中 Start Restore。然后，按要求插入磁盘。

如何恢复每个文件的最新版本

1. 启动备份程序。
2. 选中 Restore 按钮。
3. 为包含所需文件的 Setup 打开主目录。
4. 选中 Select Files 按钮。
5. 选中要恢复的文件，并选中 OK 返回到 Restore 对话框。
6. 如果需要，选中 Options 看一看，并且为恢复设置选项。
7. 选中 Start Restore。
8. 按要求插入备份介质。
9. 恢复完成后，选中 OK 清除总结报告。

24.3.5.1　从旧的备份恢复文件

如果需要恢复一个很久以前删除的文件，可能需要返回到一个早先的目录。如果该目录也已经从硬盘驱动器删除，从备份介质可以将其恢复。在确定文件删除之前的哪一个备份集包含该文件的最后版本时，可能会有一些麻烦。这时必须恢复几个目录，直到找到正确的那个。一旦发现正确的目录，选取该文件并选中 Start Restore。

如何恢复一个目录已从硬盘驱动器删除的文件

1. 选中 Catalog 打开 Select Catalog 对话框。
2. 选中 Retrieve 从备份介质恢复目录。
   Retrieve Catalog 对话框打开。
3. 如果从软盘恢复，为备份介质选择软盘驱动器。当 Restore 发出请求时，在驱动器中插入备份集的最后一张磁盘并选中 Continue。
   如果从 DOS 目录恢复，在正文框中键入驱动器和路径名。要保证在路径中包括目录文件所在目录的所有子目录，这样 Restore 准确地知道所需要的是哪个目录。一个完整的路径名应是这样的：
   L:\JNFBACK\DAILY\CC50429A.FUL。
   如果路径名不以 CATALOG.TYPE 形式的一个子目录名结尾，Restore 显示 Path name is entered incorrectly(路径名键入出错)信息。
   Restore 把目录恢复到 Backup 目录的缺省目录中并选择它，但并不打开它。
4. 选中 Load 装入目录。
   Restore 对话框返回。恢复的目录被打开，但还没有文件被选中。
5. 选取要恢复的文件。
6. 检验选中的选项，如果需要可作适当变动。
   建议选择 Verify Restore Data 选项。
7. 选中 Start Restore 并按屏幕上的指示进行操作。Restore 从备份集把选取的文件恢复到硬盘驱动器。

24.3.5.2 恢复整个驱动器

如果硬盘驱动器出现故障并且需要替换整个驱动器，就必须首先格式化硬盘驱动器并重新安装 DOS(这样 MSBACKUP 才可用)。还应重新安装在备份中没有包含的其他软件。然后，就可以准备恢复备份。

硬盘驱动器上已没有目录文件，因此必须从备份介质恢复它们。不幸的是，这意味着不会有任何主目录文件，并且识别每个文件的最新版本也会变得较为困难。

要保证所恢复的文件均为最新版本的最简单方法是恢复最后一次全备份，随后按时间顺序恢复部分备份的文件。如果一直做 Differential 备份，所要做的是恢复最后一次全备份，随后恢复最后一次 Differential 备份，这样工作就完成了。如果一直做 Incremetal 备份，恢复最后一次全备份，随后是恢复在它之后的第一次 Incremental 备份，再后面是恢复第二次 Incremental 备份，如此下去，直到按时间顺序恢复所有的 Incremental 备份。如果一个文件有几个版本被备份，这么做能保证后面的版本覆盖前面的。

如何恢复整个驱动器

1. 格式化驱动器并重新安装 DOS，重新安装那些没有备份的软件。
   重新安装 DOS 使 MSBACKUP 可用，这样就可以从备份恢复文件。
2. 启动 MSBACKUP 或 MWBACKUP，并选中 Restore 功能。
3. 选中 Catalog 打开 Select Catalog 对话框。
4. 选中 Retrieve 从备份介质恢复目录文件。
   Retrieve Catalog 对话框打开。

5. 如果从软盘恢复文件，为备份介质选中软盘驱动器。当要求插入磁盘时，在驱动器中插入最后一张盘并选中 Continue。

   如果是从 DOS 目录恢复，选中 MS-DOS Drive and Path，并在正文框中键入最后一次全备份的驱动器和路径名。要保证在路径名中包含目录文件所在目录的所有子目录，这样 Restore 准确地知道想要的目录。完整的路径名应如下所示：

   L:\JNFBACK\DAILY\CC50429A.FUL。

   如果路径名不是以 CATALOG.TYPE 形式的子目录名结尾，则 Restore 显示 Path name is entered incorrectly 信息。

   Restore 把目录文件恢复到备份目录文件的缺省目录中。

6. 选中 Load 装入该目录文件。

   Restore 对话框返回。恢复的目录文件打开，但是还没有文件被选中。

7. 选中驱动器上的所有文件。

8. 建议选择 Verify Restore Data、Restore Empty Directories 和 Audible Prompts 选项。

9. 选中 Start Restore 并按屏幕上的指示进行操作。Restore 从备份集把所有文件恢复到硬件驱动器。

10. 如果在最后一次全备份之后做 Differential 备份，只需对最后一次 Differential 备份重复步骤 3 到 9。

    如果在最后一次全备份之后做 Incremental 备份，按时间顺序对每一次 Incremental 备份重复步骤 3 到 9。

11. 如果做的备份基于不止一个 Setup，对每一个 Setup 重复步骤 3 到 10。

如果一直基于不止一个 Setup 做备份，需要恢复来自所有 Setup 的备份才能完成恢复硬盘驱动器的工作。完成所有 Setup 时，除有一点不同以外，硬盘驱动器完全恢复到最后一次备份的情况。可能恢复了某些已经删除的目录和文件，所要做的只是再次删除它们。

24.3.5.3　把文件恢复到不同的驱动器和目录

如果使用备份把文件拷贝或移动到另外一台计算机，例如，从办公室到家里，目标计算机可能有着不同的目录结构。在 Restore To 域中选择 Other Drives 或 Other Directories 可以把文件恢复到另外的驱动器和目录。

选中 Other Drives 后，Restore 请求和被恢复文件相关的每一个目录的驱动器名和路径名。选中 Other Directories 时，Restore 只请求每一个目录的路径名。

如果只是把文件恢复到另一台计算机，不一定要使用 Other Drives 和 Other Directories。如果想为被恢复的文件提供不同的驱动器名或目录名时，需要 Other Drives 和 Other Directories 选项。例如，如果文件来自驱动器 C 并且要恢复到另一台计算机的驱动器 C 时，不必使用 Other Drives。如果文件恢复到另一台计算机上的同名路径，不必使用 Other Directories。

如何把文件恢复其他驱动器和目录

1. 启动 MSBACKUP 或 MWBACKUP，并选中 Restore 功能。
2. 选中 Catalog 打开 Select Catalog 对话框。
3. 选中 Retrieve 从备份介质恢复目录文件。

   Retrieve Catalog 对话框打开。

4. 为备份介质选中软盘驱动器。当要求插入磁盘时，在驱动器中插入备份集的最后一张盘并选中 Continue。
   Restore 把目录文件恢复到备份目录文件的缺省目录中。
5. 选中 Load 装入目录文件。
   Restore 对话框返回。被恢复的目录文件打开，但还没有文件被选中。
6. 选中要恢复的文件。
7. 建议选择 Verify Restore Data、Prompt before Creating Directories、Prompt before Overwriting Existing Files 和 Audible Prompits 选项。
8. 选中 Restore To 按钮。
   Restore To 对话框打开。
9. 选择 Other Drives 或 Other Directories 并选中 OK。
10. 选中 Start Restore 并按屏幕上的指示进行操作。Restore 请求驱动器和/或目录名替换备份集中的驱动器和目录名。

24.3.5.4 重新建立丢失的目录文件

让我们设想一个最糟糕的情形：你想从旧的备份中恢复某些文件，目录文件已经从硬盘驱动器删除，而且备份集的最后一张软盘也被丢失或受损坏。没有该备份的目录文件并不会丢失所有的文件，可以请求 Restore 检查备份集并基于它的内容建立新的目录文件。只需选中 Catalog 按钮，选中 Rebuild，为备份集选中驱动器，并按指令插入磁盘。Restore 读整个备份集重新建立该目录文件。

警告： 如果备份集中的某些磁盘丢失或损坏，想要恢复的文件就可能不能用了。

如何重新建立目录文件

1. 启动 MSBACKUP 或 MWBACKUP，并选中 Restore 功能。
2. 选中 Catalog 打开 Select Catalog 对话框。
3. 选中 Rebuild。
   Restore 请求备份集的驱动器名字。
4. 为备份介质选中软盘驱动器。
   Restore 请求备份集中的第一张磁盘(除非已为备份集选择了一个 MS-DOS 目录)。
5. 按要求插入磁盘。
   Restore 在备份目录文件的缺省目录中重新建立目录文件
6. 选中 Load 装入该目录文件。
   Restore 对话框返回。被恢复的目录文件被打开，但还没有文件被选择。
7. 一旦目录文件被打开，可与其他恢复任务一样选中文件和选项。

24.4　Compare

图 24.6 所示的是选择 Compare 功能时打开的对话框。从实际操作角度来看，比较差不多与恢复一样：打开一个目录文件，选择要比较的文件，选择选项，并响应提示。Compare 只提供两个选项：Audible Prompts 和 Quit after Compare。

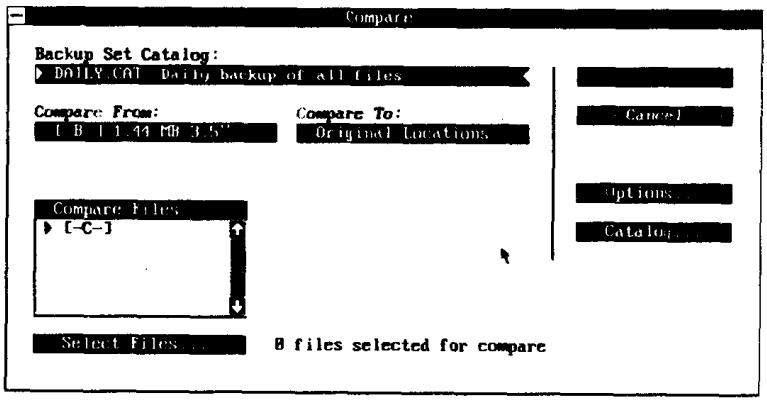

图 24.6　Compare 对话框

当遇到备份介质上的物理问题并且有错误校正码时，Compare 校正错误。它并不校正文件的备份版本与硬盘驱动器版本的不一致，而只显示警告框，以便让用户知道有问题存在。用户应自己记录下 Compare 报告的不匹配情况。最后的总结报告并不列出不一致文件的名字，它只显示有多少个不一致的文件。

如何比较备份的文件与它们的原始文件

1. 启动 Backup。
2. 选中 Compare 按钮。
3. 打开所需的目录文件。
4. 选择要比较的文件。
5. 按要求建立比较选项。
6. 选中 Start Compare。
7. 按要求插入备份介质。

24.5　从 DOS 6 以前做的备份恢复文件

如果需要从用早期 DOS 做的备份中恢复文件，使用图 24.7 中介绍的 RESTORE 命令。例如，假定你想从早期备份集把所有 CHAP*.TXT 文件恢复到驱动器 C。应输入如下命令：

```
RESTORE A: C:\CHAP*.TXT /S
```

RESTORE

恢复由早先 DOS 版本(DOS 2.0 到 5.0)BACKUP 备份的文件。

句法:

RESTORE drive1 target [/S] [/P] [/N]

参数和选项:

drive1	标识含有备份文件的驱动器。
target	标识要恢复的文件,以及要恢复到的驱动器。target 必须包括驱动器名,它可通过包括路径和文件名字指定文件。
/S	恢复文件到所有子目录。
/P	在恢复只读文件,或自从上一次备份以来已改变的文件时发出指示。
/N	只恢复目标驱动器上已不存在的那些文件。

图 24.7 RESTORE 命令总结

24.6 自 测 练 习

本章介绍了怎样比较和恢复由 MSBACKUP 或 MWBACKUP 做的备份文件。如果想要练习一下这些方法,使用下面几项做为指导:

1. 从硬盘驱动器删除整个 PRACTICE 分枝并恢复它。
2. 把恢复的分枝与备份版本相比较,以确保它们相同。
3. 恢复 PRACTICE 分枝中一个文件的旧版本。

在继续到下一章之前,你可能想从硬盘上删除所有在第 23 章和本章中建立的文件:

1. 从硬盘驱动器删除 PRACTICE 分枝。
2. 删除 PRACTICE Setup。
3. 删除 PRACTICE Setup 的目录文件(提示:打印 PRACTICE.CAT 主目录文件做为要删除目录文件的指引)。

第 25 章　DOS 的反病毒程序

简介： 病毒已不再是传说中的灾难。它们是造成许多 PC 机用户巨大麻烦的实际问题。

25.1　为什么要进行病毒防护？

这是一个病毒是多么有害的例子：一些人在他们的计算机感染病毒时还不知道它的存在，在某一天，计算机不能启动了。当他们试着从急救启动磁盘启动，病毒又安装到磁盘上却仍然不为人知。他们决定看一看启动盘在其他机器上是否能工作。每次启动另一台计算机时，病毒感染新的计算机并且使其硬件受害。最后，4 个硬盘在几分钟内被毁坏。

最后他们知道某些异常的事在活动着，他们叫来专家，他告诉他们废弃受感染的软盘，以及所有其他试过的软盘。然后，重新格式化硬盘，重新安装 DOS，并且从备份恢复所有文件。由于他们有经常备份的公司规定，所以很快恢复了运行。

病毒通过许多途径侵入初始系统。它可能从联机服务装载或随其他新软件装入。它可在系统是新的时候就已经侵入。在病毒发作之前，它可能已在系统间传播并占据系统很长时间。许多病毒等待某些引发事件（如创建者的生日），然后进行活动。

Microsoft 在 DOS 6 中包含几个反病毒工具。如果正确地使用它们，这些工具几乎可以在病毒进入计算机的同时识别出它，阻止它做任何有害的事，并且从系统清除它。在本章将学习：

- 了解病毒扫描程序和病毒监视程序的区别
- 确定何时使用病毒扫描程序
- 确定何时使用病毒监视程序
- 对驱动器扫描已知病毒
- 对驱动器扫描未知病毒
- 监视系统类似病毒行为的活动
- 如果检测或扫描到病毒该做什么
- 从 Microsoft 更新具有最新病毒信息的病毒扫描软件

25.2　Microsoft 反病毒功能

DOS 6 的反病毒程序以两种方式保护系统。病毒扫描程序扫描内存和磁盘中的驻留病毒，而病毒监视程序监视类似病毒行为的所有系统活动。这两种程序都占用系统时间，可选择反病毒选项使所用时间最少或最大限度地保护系统。

25.2.1 病毒扫描程序

有两个病毒扫描程序分别是 MSAV(Microsoft Anti-Virus)和 MWAV(Microsoft Windows Anti-Virus)。在执行 Setup 时,可以安装其中一个或两个都安装。通常不需要安装两个程序。如果喜欢使用 DOS,则安装 MSAV,如果使用 Windows,就安装 MWAV。这两个程序有一些不同,但两者都能够检查已知的和未知的病毒。

25.2.1.1 已知病毒

已知病毒是已被反病毒专家发现和分析的病毒。他们已经给它命名,如 Stoned 或 Michelangelo。他们已有了识别它的标志——病毒程序中含有的可用于标识它的一组字节。一个病毒可能有几种已知的变形——基本相同程序的细微变化。目前为止已经有超过一千种的病毒。它们中的许多还未曾看到过并且可能已灭绝,但是没有办法去预测何时或何地病毒会再次出现。

Microsoft 的反病毒程序通过检查内存和磁盘中上千种已知病毒特征的方法扫描已知病毒。这种操作在大的硬盘上可能要占用几分钟的时间。也可以把查找限制在最可能被病毒感染的区域:硬盘分区表,软盘的启动记录和程序文件。几乎没有什么病毒能把自己隐藏在其他地方。

扫描程序能够消除许多已知的病毒而且不会损坏被感染的文件或系统区,这叫作清除一个病毒。然而,有些病毒在它们侵入时损坏文件或系统区,当然,扫描程序对此无能为力。这样的文件需要删除,通常可从备份恢复它们或者从原始程序磁盘重新安装。

25.2.1.2 未知病毒

由于最新的病毒在未被检测、分析、命名,并且加入到病毒扫描程序之前还是未知的。因此,扫描未知病毒比扫描已知病毒更为重要。

扫描程序通过查看程序文件中表明它们可能已被病毒感染的变化来检测未知病毒。大多数程序文件在它们安装后并不改变,所以任何改变都可能表明已被病毒感染。

在驱动器上第一次使用 Microsoft 反病毒扫描程序时,它在每一个目录中创建一个叫作 CHKLIST.MS 的文件,记录该目录中所有程序文件的大小、属性和日期/时间。它还记录一校验和,这是通过文件内容计算得到的独一无二的数值。如果内容改变,校验和也随之改变,从而在文件大小、属性和日期/时间仍保持不变时,扫描程序也能知道文件已有变化。

> **警告:** 当扫描程序记录 CHKLIST.MS 数据时,如果有未知病毒存在于系统中,它将不能被标识。值得庆幸的是,随后讨论的 Microsoft 病毒监视程序能够防止它做任何有害的事。

随后每次扫描把 CHKLIST.MS 记录的数据与程序文件相比较,任何不一致都会导致扫描程序报告一个验证错误并让用户决定如何处理它。

许多程序变化是合法的。在改变程序选项或重新配置屏幕显示等操作时,有些程序将这些变化写入到自己的程序文件中。如果有类似这样的程序,很快就要学到如何忽略针对它们的验证错误。

如果更新应用程序到新的版本,下一次扫描对改变的程序文件将产生验证错误。因为已经知道更新了该程序,可以请求扫描程序用新的值更新 CHKLIST.MS。

如何没有进行升级或重新配置而产生了验证错误，应对此给予高度重视。可以选择删除该文件，或者先停止扫描，以进行进一步的调查，并决定采取何种措施。

25.2.2　监视病毒

Microsoft 的反病毒监视程序 VSAFE 必须作为终止驻留程序装入。VSAFE 可在 DOS、DOS Shell 和 Windows 下运行。它监视所有的系统活动以查找病毒。但同时它却使整个系统速度减慢，装入 VSAFE 即选择了安全而不是速度。作为折衷的方案，可只监视最有可能被病毒侵犯的活动，如修改硬件分区表或在硬盘上做物理格式化等。

当 VSAFE 监视到可疑的活动时，它冻练该活动并显示一个选择框，询问是否继续下去或停止。如果你知道该活动是安全的，例如，使用 FDISK 重新设置硬盘分区表，可以告诉 VSAFE 让该活动继续进行。如果不是，则停止该活动，重新启动从内存清除潜在的病毒，并且立即运行病毒扫描程序确定并删除该病毒。

25.3　何时使用反病毒程序

如果想最大限度地保护系统，应该在每次启动时扫描内存和程序文件。如果使用 MSAV，则可以在 AUTOEXEC.BAT 中添加 MSAV 命令。对于 Windows，可以在 StartUp 组中安装 MWAV，这样在每次启动 Windows 时它就运行。

病毒经常隐藏在软盘的引导区中。当插入一张来自其他系统的软盘时，不管它是 软件或者从朋友那里借来的旧磁盘，却应该在从它引导，从它拷贝，从它安装一个程序，启动它上面的程序，或以某种方式使用它之前扫描它。换句话说，一旦把它插入磁盘驱动器中就扫描它(随后你会看到如何设置 VSAFE 从而一旦在系统中插入软盘即自动地进行这一操作。)

一旦接收到新的或租借来的计算机就应扫描它，因为代理商在发货之前可能并没有扫描计算机。它或许已经在商店或先前租借的用户那里感染了病毒。

当计算机遇到有可能导致病毒入侵的情况，应该重新扫描系统：

- 如果电子公告牌服务(即使是众人皆知的 CompuServe 或 Prodigy)，或任何类型的联机服务，尤其是从它卸载某些内容。
- 如果通过 INTERLNK 或类似软件把你的计算机与其他机子相连。
- 如果进入网络并从网络驱动器拷贝内容到你的一个本地驱动器，或运行位于网络驱动器上的程序。
- 如果在删除某一文件被病毒感染的版本后从它的备份恢复文件(备份可能也已经被感染)。
- 如果安装或更新软件，尤其是软件原始包装已被启封。

注释： *安装或更新软件时的扫描使扫描程序有机会对新的文件记录 CHKLIST.MS 数据。*

每当产生这些事件之后的扫描都要花费许多时间，而且大多数人并不中意这样严格的处理。如果选择较少次数的扫描，但同时计算机系统又常常与外界发生关系，一定要安装并有规律地使用 VSAFE。可在 AUTOEXEC.BAT 中插入 VSAFE 命令以便每次引导时启动

它。

25.4 MSAV 的使用

MSAV 命令可以直接在批处理程序如 AUTOEXEC.BAT 的 DOS 提示符下键入，或者从 DOS Shell 使用 File，Run 选项。现在尝试启动 MSAV：

1. 退出所有程序并进入 DOS 命令提示符。
2. 键入 MSAV 启动程序。

   对话框打开(如图 25.1 所示)。

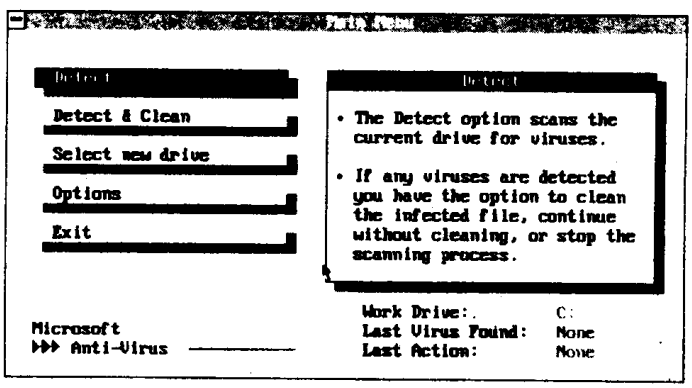

图 25.1 Main Menu 对话框

如何扫描硬盘驱动器中的病毒

1. 在 DOS 命令提示符下键入 MSAV。

   MSAV 对话框打开。
2. 如果工作的驱动器(在右下角)并不是硬盘驱动器，按 F2。
3. 从列表中选择硬盘驱动器。

   MSAV 读选中的驱动器并且工作驱动器变为新选择的驱动器。
4. 选中 Options 打开 Options 对话框。
5. 选择想要使用的选项，并禁止那些不想使用的选项。

   在有效的选择旁边有检验标记出现。
6. 选中 OK 关闭对话框。
7. 如果已使 Prompt While Detect 选项有效，或者选择 Detect，或者选择 Detect & Clean 并开始扫描。

   如果已使 Prompt While Detect 选项有效，选择 Detect 报告病毒但不清除它们，或者选择 Detect & Clean 报告病毒并清除可以清除的病毒。
8. 如果已使 Prompt While Detect 选项有效，应响应显示的病毒选项。
9. 当总结报告显示在屏幕上时，选中 OK 关闭对话框。

25.4.1　选择驱动器

从 Main Menu 对话框中，一次只能扫描一个驱动器。在对话框的右下角 Work Drive：(工作驱动器)字样旁显示所选驱动器。启动 MSAV 时，即根据缺省值选择当前驱动器。可按 Select New Drive(重新选择驱动器)按钮改变所选驱动器。

注释： 可按适当的功能键激活 MSAV 对话框底部所标的功能。

下面练习一下改变驱动器：

1. 在 A 驱动器中插入一个磁盘(其内容无关紧要，但必须是格式化过的)。
2. 按 F2 键选中 Select New Drive 功能。在对话框左上部出现一驱动器表。
3. 双击驱动器 A 或键入字母 A。
   MSAV 读盘，Work Drive 变换为 A。
4. 接着换回 A 驱动器。

25.4.2　选项设置

MSAV 给出了一系列选项，从而使扫描工作更为灵活。在读这些选项的同时不妨打开屏幕上的选项对话框：

1. 单击 Options(选项)按钮或键入字母 O。
   打开 Options Setting(选项设置)对话框(如图 25.2 所示)。

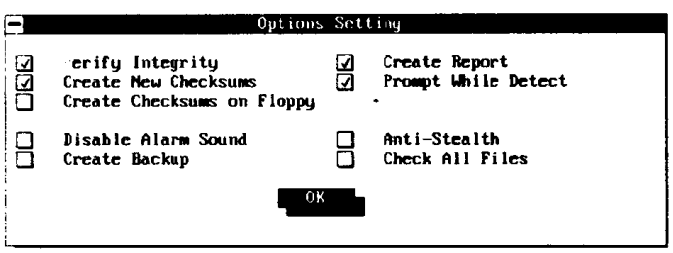

图 25.2　MSAV Options Setting 对话框

第一组三个选项控制扫描未知病毒。下面那组两个选项为混合选项。第三组控制 MSAV 是否自动进行操作。最后一组专为一些非同寻常的情形而设置。一个对号表示选中选项。

25.4.2.1　Veify Integrity(确保完整性)

当选中此选项时，MSAV 在扫描过程中将查找未知病毒。退选此项可节省大量扫描时间但却必须为漏过了某种未知病毒而冒风险。其缺省设置为选中该项，建议选中该项。

25.4.2.2　Create New Checksums(创建校验和)

当该选项被选中时，MSAV 为在扫描过程中找到的每一个新程序文件创建一 CHKLIST.MS 入口。如果在第一次运行 MSAV 前就取消该选项，则 MSAV 因不能创建 CHKLIST.MS 文件而失去扫描未知病毒的功能。保持该选项处于选中状态可使系统在加入新软件后不断刷新 CHKLIST.MS 文件。

25.4.2.3 Create Checksums on Floppy(创建软盘查找记录)

通常 MSAV 只查找硬盘上的未知病毒,因为 CHKLIST.MS 系统只对频繁使用的软盘作出反应。如果想让 MSAV 查找通常使用的软盘上的未知及已知病毒,选中此项,MSAV 就会创建必要的 CHKLIST.MS 文件。

25.4.2.4 Disable Alarm Sound(去除警告声)

通常 MSAV 在遇到已知或可疑的未知病毒时,会发出蜂鸣声。尽管还能引起使用者注意,但却是一种十分令人厌烦的声响。选中该选项,就能去除蜂鸣声。

25.4.2.5 Create Backup(创建备份)

该选项使 MSAV 为受感染的文件创建一后缀为.VIR 的备份。对于那些不能删除又感染了未知病毒的文件,可考虑选择此功能。将文件转化为.VIR 形式使得它不可执行,从而也就阻断了病毒进入内存的通路。这样受感染的文件就可以安全得以保存了,当然将它从硬盘上移到专门隔离软盘上也许更为可靠。也许时隔不久这种未知病毒就变为已知的了,这时就能清除它而恢复原有文件。在本章后些部分将讲述如何添加新病毒定义。

25.4.2.6 Create Report(创建报告)

一般情况下 MSAV 在扫描最后显示一总结报告。此总结包含了扫描的详细结果,如扫描文件的总数、感染已知病毒的文件数目以及清除病毒文件的数目等。此总结不涉及未知病毒,一旦关闭对话框它也就随之消失了。Create Report 选项使 MSAV 在盘上生成一更为详尽的报告,除了总结信息外,还包括一份可能包含未知病毒的文件清单。图 25.3 给出了一份存盘报告的示范。它被存为 ASCII 文本文件以 MSAV.RPT 为名保留在被扫描驱动器的根目录下。可用 EDIT 观看、打印。

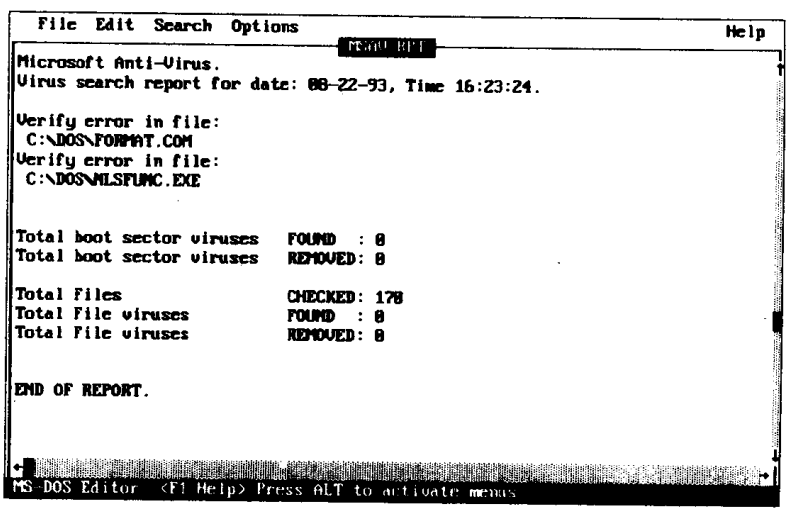

图 25.3 MSAV.RPT 示样

在本章后些时候将讲到如何不用对话框扫描病毒,这对于想从 AUTOEXEC.BAT 运行 MSAV 是必要的。一旦对话框被关闭,MSAV.RPT 就成了查获病毒的唯一证据。

25.4.2.7　Prompt While Detect(查获提示)

该选项控制 MSAV 是将病毒报告给使用者还是自动进行处理。当选中该选项时，MSAV 每遇到一个病毒就显示一提醒框。若去除该选项，则 MSAV 的操作取决于在初始化扫描过程时选的是 Detect 还是 Detect & Clean 按钮。如果选的是 Detect，MSAV 只是在报告中注明此病毒然后往下继续；如果选的是 Detect & clean，则它还会清除所能消除的病毒。

注释： 如果报告表明找到了病毒但未清除，则应选中该选项重新扫描，这样就可处理被感染的文件了。

25.4.2.8　Anti-Stealth(反隐病毒)

受隐病毒(Stealth virus)感染的文件其大小、日期/时间标志、属性及校验和均不发生改变。扫描隐病毒仅比普通病毒多花几秒时间，却可带来额外的保护。建议始终使用此选项。

25.4.2.9　Check All Files(检查所有文件)

此选项使 MSAV 除了扫描程序文件外同时扫描数据文件。由于大多数病毒只感染程序文件，可以关掉此选项以节省扫描时间。但如果希望得到更多的保护，则应选中此项。

至此讲述了有关选项的一些知识，接着进行一下实际操作：

1. 使下列选项有效：

   Verify Integrity
   Prompt While Detect
   Anti-Stealth

   当选中某选项时，它旁边会出现一对号。
2. 使其他选项无效。
3. 选中 OK 关闭对话框。

25.4.3　运行扫描程序

选完选项后，开始进行扫描。

1. 选中 Detect。

   Main Menu 对话框相当于一进程屏幕。本屏幕右下角显示扫描驱动器、找到的最近一个病毒及对此采取的措施。

   由于 Prompt While Detect 被设定为有效，在扫描过程中将出现一些提醒框(作为一个典型的例子，如图 25.4 所示)。一旦出现病毒警告，选中 Stop 以停止扫描。参阅下一节“病毒处理”，然后继续练习。

   如果未找到任何病毒，扫描结束将出现一总结对话框。
2. 选中 OK 关闭对话框。
3. 按 F3 退出扫描程序。

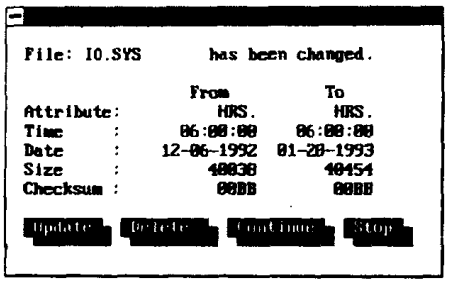

图 25.4 病毒警告对话框示例

25.5 病毒处理

图 25.4 展示了一典型的未知病毒警告框。如果文件是被合理改动的，并且希望以最新的数值刷新 CHKLIST.MS，可选 Update。如果怀疑文件感染了病毒则可选 Delete 删除文件。选 Continue 忽略显示信息而继续扫描。选 Stop 停止扫描，进而调查文件改动的原因。

若 MSAV 找到了一个已知病毒，则 Virus Found 对话框将显示病毒名称，并允许使用者在可能的情形下清除病毒(Clean)，不处理病毒而继续扫描(Continue)，或删除文件(Delete)。如果无法清除病毒，应删除文件并从备份中恢复它。在选中 Delete 按钮前记下文件名，以免结束扫描后忘了此文件名。从备份恢复文件后，应再扫描一遍驱动器以确保恢复的文件未感染病毒。

25.6 其他 MSAV 命令

Delete 命令删除所选驱动器中的全部 CHKLIST.MS 文件。偶尔为了去除过时的文件入口，可进行这样的操作。在下次扫描驱动器时使用 Create New Checksums 选项即可创建新文件。

List 命令可给出 MSAV 所能查找的已知病毒清单。当需要确认此扫描程序能否查找、清除某流行的病毒时，这样一份清单自然能带来不少方便。如果有病毒不在此清单中，不妨刷新病毒定义以包括它。

25.7 在批处理模式运行 MSAV

图 23.5 总结了 MSAV 命令，其中包含了一些可能有用的选项。

可通过以上/N 或/P 选项，去掉 MSAV 窗口而直接开始扫描。/P 使得 MSAV 在命令提示符屏幕中显示进程信息，如 Scanning memory for viruses(扫描内存中的病毒)...，Scanning files for viruses(扫描文件中的病毒)...等等。最后显示扫描总结报告。

MSAV

扫描内存及磁盘病毒，并尽可能清除病毒。

句法：

MSAV[drive... |path|/A|/L] [/S|/C] [/R] [/A] [/L] [/N] [/P] [/F]

参数和选项：

none	不需打开 Microsoft Anti-Virus 窗口即可选择驱动器和选项。
drive	认明所要扫描的驱动器，扫描其中所有的程序文件。
path	认明某一分枝的顶部，扫描这一分枝中的所有程序文件。
/A	扫描除 A 和 B 以外的所有驱动器。
/L	扫描除 A、B 和网络驱动器外的所有驱动器。
/S	扫描但不清除病毒。此为默认设置。
/C	扫描并清除所有发现的病毒。
/R	创建一报告文件(MSAV. RPT)列出所检查的文件、找到的病毒、清除的病毒的数目，以及可能包含未知病毒的文件名。
/N	关闭扫描时的信息显示。
/P	以非图形的命令行模式运行 MSAV。
/F	关闭扫描文件名显示。此选项仅能与/N 或/P 合用。

注释：

本书最后的"命令参考"描述了某些不太常用的选项。/VIDEO 包含了一系列视频及鼠标选项，/video-mouse 则涉及视频/鼠标选项。

图 25.5　MSAV 命令简介

而/N 选项使得 MSAV 显示/MSAV. TXT 文件的内容，如果不存在这个文件，则只显示 working... 最后也不显示任何报告。如果发现病毒，只是设定退出码 86(关于退出码请参看第 37 章"批处理程序中的编程逻辑")。如果是为初学者在批处理文件中设置 MSAV，不妨加上/N。可将任何有关信息放在 MSAV. TXT 中。对于单用户系统来说，应把 MSAV. TXT 存放在 MSAV. EXE 所在目录中，而对于多用户系统，只要能为用户设置分立的块，就能为不同的用户创建不同的 MSAV. TXT 文件。在配置块中，设置一名为 MSDOSDATA 的环境变量以指向 MSAV. TXT 所处位置。这样 Peter 的 MSAV. TXT 就可在 PETERDIR 目录中，而 Judi 的 MSAV. TXT 就可在 JUDIDIR 目录中。

注释： MSDOSDATA 同样影响 MSBACKUP 的工作方式，具体参见第 23 章"MSBACKUP 和 MWBACKUP"。

同时加上/P 和/N 选项，MSAV 将显示扫描的文件名，除非再加上/F 选项。

MSAV. INI 文件控制有效选项，但某些选项，如 Prompt While Detect 和 Disable Audible Alarm 不适用于批处理模式。可用任何 ASCII 编辑器编辑此文件，当然也可在 MSAV 窗口中交互式设置选项，并在退出前进行存储。/R 选项用来创建磁盘报告而不管 Create report 的设置状况。

假如你要在批处理模式下运行 MSAV，去除 MSAV. TXT 以外的所有显示信息，同时创建一磁盘报告，扫描所有的当地驱动器，清除找到的病毒，并去除文件名显示，可用下列命令：

```
MSAV /N /R /L /C /F
```

提示： 如果将上述命令包括在一批处理文件中，那么紧随其后的命令可以是 IF ERRORLEVEL 86 以判定是否查出病毒。

MSAV 的某些参数可使之自动运行，即使不在批处理模式下。在这种情况下，MSAV 窗口一出现就开始扫描，用户没有任何机会选择选项或按按钮。而一旦扫描结束，MSAV 又在你读屏幕报告前就自动结束。可达到此目的命令行参数为/S、/C、路径，或其他指定一个以上扫描驱动器的参数。

25.8 MWAV 的使用

图 25.6 是一个典型的 MWAV 对话框，它看起来与 MSAV 相似，除了前者上面总有可选的驱动器清单，每次扫描均可选择多个驱动器，但不能创建磁盘报告。同时 MWAV 还可擦除删去的文件。擦除文件是用零来覆盖文件簇，从而达到彻底清除的目的。这样一来就从簇根除了病毒。MWAV 选项存储在 MWAV. INI 中。

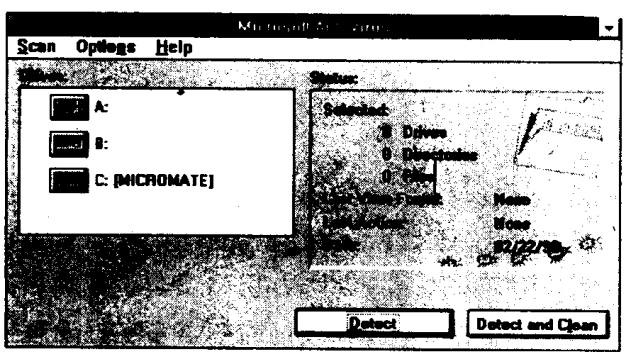

图 25.6 MWAV 对话框

25.9 VSAFE 的使用

VSAFE 作为一 TSR 是 DOS 的病毒监视器，用来监控系统的所有操作，寻找病毒存在的迹象。它可找到最近一次扫描后进入系统的病毒，并立即发挥作用。它也可找到扫描程序漏过的病毒。

但是，这也带来一些副作用。病毒监视器会占用运行其他程序的时间，尽管在 486 及以上机型中，感觉不到速度的降低。同时它还占用一定的内存。第 29 章“内存管理”和第 30 章“为 386、486 和奔腾机的增加的内存工具”讲述了如何为应用程序释放更多的内存。使用这些技术，或许可找到足够的内存来加载 VSAFE 及其他的 TSR 和应用程序。

VSAFE 命令用来加载 VSAFE TSR。可在 AUTOEXEC. BAT 中插入适当的命令以使

每次启动时加载 VSAFE。当然，如果你不打算老用它，也可在需要监视系统操作时在命令提示符下加载它。

试着加载 VSAFE TSR：

1. 退出任何使用程序，回到初始命令提示符。
2. 键入 VSAFE 加载 TSR。

   显示信息确入已加载 VSAFE，并告知占用的内存。

如果要在 Windows 下运行 VSAFE，必须加载 MWAVTSR. EXE 驱动程序，从而使 VSAFE 能在 Windows 屏幕中显示警告框。由于 MWAVTSR. EXE 是一 TSR，必须在启动 Windows 时加载它，而不是在启动后。如果使用的是 **Windows** 3. 1 及以后版本或 Norton desktop for Windows，可在 StartUp 组中添加 MWAVTSR. EXE 程序项。或按以下步骤在 WIN. INI 文件中添加 MWAVTSR. EXE：

怎样从 WIN. INI 加载 MWAVTSR. EXE

1. 使用诸如 EDIT 之类的 ASCII 文本编辑器编辑 WINDOWS\WIN. INI。
2. 找到[Windows]部分。
3. 找到 load＝行(如果没有，插入这样一行)。
4. 在此行末添加 MWAVTSR. EXE。
5. 存储文件并退出编辑器。
6. 如果正在运行 Windows，退出并重新运行以载入此驱动程序。

25. 9. 1　VSAFE 选项

加载了 VSAFE 后，就可按 Alt＋V 热键在 DOS 命令提示符屏幕中浏览、选择 VSAFE 选项。如果正在 Windows 中，找到 VSAFE**Manager**窗口(启动 Windows 后即被打开)并选中 Options 按钮即可。

试着打开 Options 对话框。

1. 按 Alt＋V 键。

   打开选项对话框。
2. 在浏阅选项时，将对话框保留在屏幕上。

图 25. 7 显示了 Windows 屏幕中的 Options 对话框。在 DOS 中与此类似。这些选项控制 VSAFE 的监视程度。图中所选的是缺省设置。所做的任何变动都会对以后的运行产生影响。

25. 9. 1. 1　HD Low-Level Format(硬盘低级格式化)

选中此选项使 VSAFE 对任何进行低级(物理)格式化硬驱的企图发出警告。

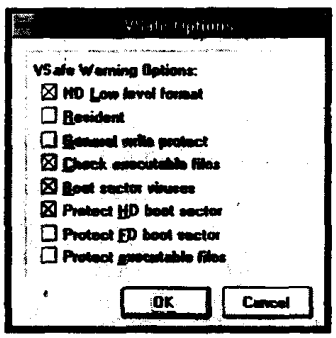

图 25.7 VSAFE Options 对话框

25.9.1.2 Resident(驻留)

此选项被选中后,一旦某个程序试图使其成为内存驻留的,VSAFE 即发现警告。如果愿意,不妨一试:

1. 按 2 选中此项。确认对话中此选项旁出现 X 号表示此选项有效。
2. 按 Esc 关闭对话框回到命令提示符。
3. 键入 DOSKEY/REINSTALL 命令以安装 DOSKEY TSR 的新拷贝。
   计算机发出蜂鸣声(有点像电话铃声)并显示 VSAFE Warning Message。
4. 选中 Stop 阻止加载 TSR。
   回到命令提示符。

25.9.1.3 General Write Protect(一般性写保护)

此选项使 VSAFE 对程序试图写盘发出警告。对日常使用而言,这种保护措施可能有点过份。

答疑: 当我试图启动 Windows 时,显示一硬盘出错信息,或硬盘写保护信息。

如果载入的 VSAFE 选中的了 General Write Protect 选项,就会出现这种问题。Windows 在加载过程中需在硬盘上创建临时文件,而此 VSAFE 选项禁止此类操作。

选中 Cancel 关闭对话框,然后退出 Windows。你必须对一系列出错信息作出回答,因为 Windows 在退出过程中无法存储其组文件。

回到 DOS 命令提示符后,按 Alt+V 打开 VSAFE 选项对话框。然后按 3 关闭 General Write Protect 选项(确认去除了 X 号)。现在可以重新启动 Windows 了。

如果在未使用 VSAFE 的情况下得到这条信息,那可能是因为硬盘出了些问题。退出 Windows,运行 SCANDISK 找出问题所在。

25.9.1.4 Check Executable Files(检查可执行文件)

此选项用来使 VSAFE 检查每个加载的可执行文件是否包含已知及未知病毒(只有在以前扫描过这些未知病毒从而创建了 CHKLIST.MS 文件的情况下,才能检查未知病毒)。

25.9.1.5　Boot Sector Viruses(引导区病毒)

此选项使 VSAFE 检查插入系统的每张软盘的引导区中的已知病毒。

25.9.1.6　Protect HD Boot Sector(硬盘引导区防护)

此选项使 VSAFE 对程序写硬盘引导区(包括文件分配表)的企图发出警告。

25.9.1.7　Protect FD Sector(软盘引导区防护)

此选项与上一次选项相似,只是指针对软盘引导区。

25.9.1.8　Protect Executable Files(可执行文件防护)

此选项使 VSAFE 对任何修改可执行文件的企图发出警告。

25.9.1.9　VSAFE 发现可疑操作后采取的措施

VSAFE 发现所监视的可疑操作后,自动阻断此操作并显示一个与图 25.9 类似的警告框。用户可以选择中止或继续操作。在此例中,由于 VSAFE 在监视 Check Executable Files 选项时发现程序文件有所改动,还可选择刷新 CHKLIST.MS 文件。

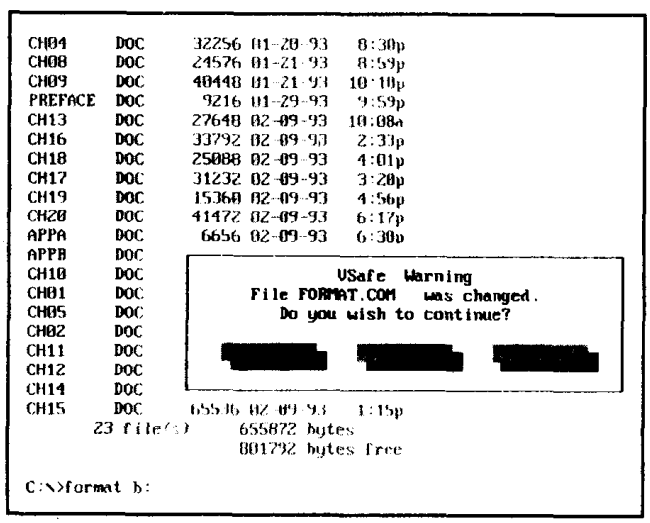

图 25.8　VSAFE 警告框

若选中 Stop 按钮,VSAFE 阻断操作。有时会因此而出现一些出错信息,尤其是在合法程序中。如果怀疑此操作不合法,关闭文件和程序,重新启动清除内存,然后扫描病毒。如果并未发现什么病毒,但 VSAFE 仍发现有可疑操作,应向 Microsoft Product Support Team (Microsoft 产品支持队伍)求助。

25.9.2　卸载 VSAFE

在某些情况下可能需要使 VSAFE 完全失效。比如,在安装或升级 DOS 及 Windows 时,必须卸载 VSAFE,否则 SETUP 程序无法顺利工作。可用紧随 VSAFE 命令后的/U 选项卸载 TSR,但这必须是加载的最后一个 TSR。

如何卸载 VSAFE

1. 退出 Windows 或其他外壳。
2. 键入命令：

   VSAFE /U

3. 若显示信息告知由于有其他 TSR 是在 VSAFE 后安装的而不能卸载 VSAFE，并且 VSAFE 不是通过 AUTOEXEC. BAT 加载的，重新启动以卸载 VSAFE。
4. 若/U 选项无法卸载 VSAFE，并且 VSAFE 命令包含在 AUTOEXEC. BAT 中，进行以下操作。
   a. 编辑 AUTOEXEC. BAT。
   b. 在 VSAFE 命令前插入 REM。
   c. 保存文件退出编辑器。
   d. 重新启动。
   e. 如果要再次加载 VSAFE，去掉命令前的 REM，保存文件，退出编辑器，重新启动即可。

试着卸载 VSAFE：

1. 退出正运行的 Windows。
2. 键入命令：

   VSAFE/U

3. 若显示信息告知由于有其他 TSR 是在 VSAFE 后安装而不能卸载 VSAFE，并且 VSAFE 不是通过 AUTOEXEC. BAT 加载的，重新启动以卸载 VSAFE。
4. 若/U 选项无法卸载 VSAFE，并且 VSAFE 命令包含在 AUTOEXEC. BAT 中，进行以下操作：
   a. 编辑 AUTOEXEC. BAT。
   b. 在 VSAFE 命令前插入 REM。
   c. 保存文件退出编辑器。
   d. 重新启动。

注释： 若要再次加载 VSAFE，去掉命令前的 REM，保存文件，退出编辑器，然后重新启动。

25.9.3 VSAFE 的加载选项

图 25.9 对 VSAFE 命令进行了总结。/n+和/n-或控制 VSAFE 选项的开和关。其具体数字与前面的监视选项列表中的相对应。比如，/1+使 VSAFE 监视低级格式化，而/3-使 VSAFE 不再监视写磁盘操作。

VSAFE

加载(或卸载)VSAFE TSR,连续监视计算机的类病毒操作。

句法:

VSAFE [/option+|-]...[/U]

参数和选项:

/option+|-　指定 VSAFE 监视病毒的方式,选项后的+或-用来控制开关。

/U　从内存中清除 VSAFE,不能与其他选项合用。

注释:

本书最后的"命令参考"介绍了一些不太常用的选项。如/NE 和/NX 防止 VSAFE 加载到扩充和扩展内存中去,/Akey 和/Ckey 用来设定 VSAFE 热键,/N 使 VSAFE 监视网络驱动器,而/D 关闭校验和等。

图 25.9　VSAFE 命令简介

25.10　更新病毒清单

新病毒不断出现,比起三年前来,现在的用户更容易碰到新病毒。因此,随时刷新已知病毒清单就变得至关重要。Microsoft 建立了一布告牌以通告最新的病毒信息。参看 DOS User's Guide 以获取有关布告牌及下载最新信息的办法。

一旦加入了这些最新信息,反毒软件就能识别这些新病毒,但却无法清除它们,要想清除这些新病毒,必须安装最新版本的反毒软件。DOS User's Guide 包括了用于订购新版反毒软件的订单。

25.11　自 测 练 习

本章讲述了如何用 MSAV 或 MWAV 及 VSAFE 保护系统免受病毒侵扰。如果想自己练习这些技术,可以将以下练习作为指引:

1. 扫描所有的硬盘寻找已知和未知病毒。
2. 扫描一些软盘寻找已知病毒。
3. 加载 VSAFE,选择其中一些选项。保持 VSAFE 的加载状态,进行下面的练习。
4. 如果想在每次启动时扫描一个或多个驱动器,编辑 AUTOEXEC.BAT。为使扫描自动进行,勿忘使用某些 MSAV 选项。
5. 如果想在每次启动时运行 VSAFE,在 AUTOEXEC.BAT 中插入 VSAFE 命令,并使用适当的选项。
6. 如果要在 Windows 下运行 VSAFE,编辑 WIN.INI 以加载 MWAVTSR.EXE。

第 26 章 UNDELETE 的使用

简介： 从备份中恢复被删除的文件无疑是最理想的。但另一方面，Delete Sentry 程序抓取文件删除并将删除的文件转移到 SENTRY 隐目录中，保存数天之久。从那儿可用 UNDELETE(恢复)程序以近乎 100%的准确性恢复文件。

26.1 文件的删除与恢复

这也许是寻常的一幕：你花了整个上午为下午的营销会准备电子报表，但在外出吃午饭前，不小心按错了键，删除了整个文件。如果安装了 UNDELETE，只需轻轻几次键击就可恢复文件。否则，或许文件仍能恢复，但花的力气却要多一些。

DOS 通过将文件的目录项标记为可再使用的来删除文件(它将文件名的第一字母变为 $\sum$)。DOS 同时改变文件的 FAT 项，将它们标记为零以表示未使用的串。但是它并不真正抹去串中的数据以目录的剩余项。

要恢复一个文件，DOS 必须恢复仍包含原始数据的文件串。如果其他文件占用这些串，则 DOS 无法恢复删除的文件。删除防护程序尽力保持原始串的状态(或至少追踪它们被用在何处)。

DOS 6 提供了两种级别的保护措施。高级保护 Delete Sentry 影响正常的删除过程，它将文件从原先的目录转移到某一特定隐目录，并保持一段时间。低级保护删除追踪，保存一删除文件串的清单，如果这些串未被再次使用，就可恢复文件，但删除追踪并不能保持这些串的状态。如果以上两种措施均未采用，那么也许能使用删除目录项的信息来恢复文件，前提是 DOS 未再次使用它。

这些保护措施能在 DOS、DOS Shell 和 Windows 下工作。DOS 与 Windows 的不同只在于恢复文件的方式，而不是保护措施的实质。

本章将讲述如何用 Delete Sentry 在短时间内保护删除的文件及如何恢复用各种方法删除的文件。将学习：

- 安装并设置 Delete Sentry
- 恢复 Delete Sentry 保护的文件
- 恢复删除追踪保护下的文件
- 恢复 DOS 目录项删除的文件
- 清除 Delete Sentry 目录
- 清除 Delete Sentry

作者忠告： 使用删除保护而不依赖于它。

不要用删除保护来取代经常的备份操作。正如本书一再强调的，良好的备份习惯是防止数据丢失的最好办法。它能起到 Delete Sentry 所不能起到的作用：

- 文件由拷贝或转移操作所取代
- 文件是在很早以前删除的，数据已不再在盘上
- 文件是因软硬件错误、病毒或者诸如火灾、水灾之类的外部灾祸遭受破坏的

仅在还未来得及进行备份的情况下用诸如 Delete Sentry 之类的删除保护系统保护文件。

26.1.1　Delete Sentry 的使用

如果要用 Delete Sentry 或 Deletion-tracking(删除追踪)，必须首先加载 UNDELETE TSR 并指定保护的驱动器。图 26.1 为加载 UNDELETE 的命令总结。

UNDELETE

使删除保护有效或无效，并报告保护状况。

句法：

UNDELETE /LOAD|/U|/S[drive]...|/T[drive:-entries]... | /STATUS

参数和选项：

/LOAD	以默认删除保护方法加载 UNDELETE。
/U	卸载 UNDELETE TSR。
/S [drive]	加载 UNDELETE TSR，并将 Delete Sentry 设置为当前和默认的删除保护方法；drive 指定 Sentry 保护的驱动器。
/T [drive]	加载 UNDELETE TSR，并将删除追踪设定为当前和默认的删除保护方法；drive 指定追踪保护的驱动器。
entries	指定驱动器中删除追踪文件入口的最大数目，可从 1 到 999。其默认值视驱动器大小而定。
/STATUS	显示当前保护方法保护的驱动器。

图 26.1　UNDELETE 命令总结(加载 TSR)

可用以下命令通过加载 TSR 设置 Delete Sentry：

```
UNDELETE /S [drive]...
```

比如可用以下命令安装 Delete Sentry 保护驱动器 D：

```
UNDELETE /SD
```

试着为引导驱动器安装 Delete Sentry：

1. 由于 Delete Sentry 为 TSR，在安装前应退出所有的程序和外壳。确认处于初始命令提示符下。
2. 键入 UNDELETE /SC(如果 C 是引导驱动器)。

   系统将作出反应：

```
UNDELETE loaded.
Delete Protection Method is Delete Sentry.
```

```
Enabled for clrives:C
```

网络注意： *要在网络驱动器上使用 Delete Sentry，必须对该驱动器的根目录有读写、创建、删除文件的权限。*

UNDELETE 程序保存保护驱动器的清单。下次加载 Delete Sentry 时，即使未指定驱动器 C，它仍受保护。假设下次加载 Delete Sentry 时添上驱动器 D：

```
UNDELETE /SD
```

程序作出的反应表示 C 和 D 均受 Delete Sentry 保护：

```
UNDELETE loaded.
Delete Protection Method is Delete Sentry.
Enabled for drives:C D
```

也可以通过指定两个/S 选项，同时设定 C 和 D：

```
UNDELETE /SC /SD
```

当指定/S 时未包含驱动器名时，Delete Sentry 将当前驱动器加到保护驱动器清单中。比如 C 是当前驱动器，下面的命令等效于 UNDELETE /SC：

```
UNDELETE /S
```

若打算使用 Delete Sentry，首先用/S 选项指定每个想用保护的驱动器进行加载，然后只需在 AUTOEXEC.BAT 中插入下面命令就可以相同的设置加载 Undelete 了：

```
UNDELETE /LOAD
```

根据默认设置，Delete Sentry 并不保护所有的被删除文件。它只保护具有正的归档属性（在最近一次备份后做过改动）的文件。另外它也不保护 DOS 为了自身用途创建和删除的文件，如.TMP（临时）文件。

26.1.2 检查驱动器清单

如果忘了 Delete Sentry 所保护的驱动器名，可用/STATUS 选项列出。试着进行此练习：

1. 键入 UNDELETE /STATUS。
   显示下述信息：

```
Delete Protection Method is Delete Sentry.
Enabled for drive:C
```

下节将讲述如何从 Delete Sentry 恢复文件。由于刚加载上此程序，它可能还未抓取到被删除文件。在继续下面的练习之前，先删除几个文件，以便在下节中练习文件恢复。

1. 在根目录下创建一名为 PRACTICE 的子目录，并从 DOS 目录中拷贝所有的.HLP 文件，将扩展名换为.XXX（可用一个 XCOPY 命令完成此操作）。
2. 换到 PRACTICE 目录。
3. 删除 PRACTICE 目录中的所有文件，但不删除目录本身。

26.1.3 列出可恢复文件

如何使用 Delete Sentry 恢复文件

1. 按以下句法键入命令列出可恢复文件：

UNDELETE [filespec] /LIST

如果已运行过 Delete Sentry，UNDELETE 自动列出可由 Delete Sentry 恢复的文件。

2. 决定列表中需要恢复的文件。
3. 再次键入 UNDELETE 命令，不加/LIST 选项。

UNDELETE 一次列出一个可恢复文件，并询问是否进行恢复。

4. 对每个不想恢复的文件，键入 N。
5. 对每个确实想要恢复的文件，键入 Y。

如果恢复文件与同一目录中的一个已存在文件重名，UNDELETE 要求提供一新名称。

6. 完成文件恢复后，按 Esc 退出 UNDELETE。

也可用 UNDELETE 命令由 Delete Sentry 恢复文件(见图 26.2)。

UNDELETE

恢复被删除文件；列出可恢复文件；删空 SENTRY 目录。

句法：

UNDELETE [filespec] [/DT|/DS|DOS] [/LIST|/ALL] [/PURGE[drive]]

参数和选项：

none	用最有效的方法恢复当前目录中的所有被删除文件。
filespec	指定想要恢复的文件，缺省值为当前目录中所有被删除的文件。
/DT	仅恢复列在删除追踪文件(PCTRACK.DEL)中的文件，每个文件恢复时均需确认。
/DS	仅恢复列在 Delete Sentry 目录(SENTRY)中的文件，对每个文件均需确认。
/DOS	仅恢复 DOS 目录中的文件，每个文件均需确认。
/LIST	列出可恢复文件，但并不直接进行恢复。列出哪些文件取决于 filespec 和恢复方法。
/ALL	不经确认直接恢复文件。恢复的文件取决于文件名和恢复方法。
/PURGE[drive]	删空指定 drive 中的 SENTRY 目录，缺省值为当前驱动器。

图 26.2 UNDELETE 命令总结(恢复和清除文件)

/LIST 选项使 UNDELETE 列出所有的可恢复文件。练习此操作：

1. 确认正在 PRACTICE 目录中。
2. 键入下面命令：

```
UNDELETE /LIST
```

屏幕显示下列信息：

```
Directory: C:\PRACTICE
```

```
File Specifications: *.*
Delete Sentry Control file contains 5 deleted files.
Deletion-Tracking file not found.
MS-DOS directory contains 6 deleted files.
Of those 4 files may be recovered.
Using the Delete Sentry method.

DEFRAG      XXX     9,227  5-31-94  6:22a  ...A  Deleted:  2-11-95  11:13a
DOS SHELL   XXX   161,323  5-31-94  6:22a  ...A  Deleted:  2-10-95  10:35a
HELP        XXX   296,844  5-31-94  6:22a  ...A  Deleted:  2-10-95  10:35a
MEMMAKER    XXX    17,237  5-31-94  6:22a  ...A  Deleted:  2-11-95  11:13a
QBASIC      XXX   130,881  5-31-94  6:22a  ...A  Deleted:  2-11-95  11:13a
```

UNDELETE 用各种恢复方法寻找文件，然后选择最好的方法(在此例中是 Delete Sentry)列出可恢复文件。报告的第一部分显示了三种恢复方法的结果，这之中只有 Delete Sentry 是重要的。

Delete Sentry 不仅显示文件的目录信息，还包括删除的日期和时间。如果列出的文件有重名，这一信息就显得分外有用。

注释： ...A 表示文件非系统、非隐含、非只读，但在删除时具有正的归档属性。

注释： 无论当前是否加载 Delete Sentry TSR，UNDELETE /LIST 均可正常工作。只要驱动器中存在 SENTRY 目录，UNDELETE 就能显示可恢复文件。

如果在命令后加上一文件名，UNDELETE 仅列出与此样本名匹配的文件。试着仅列出 MEMMAKER.XXX 入口：

1. 键入 UNDELETE MEMMAKER.XXX /LIST。
   UNDELETE 仅报告名为 MEMMAKER.XXX 的文件。

如果在同一目录中多次创建、删除了同名文件，将列出多个文件，它们的日期和时间标志有助于决定恢复哪个文件。

可在命令后加上一路径后，文件名可加可不加，以列出另一目录中的可恢复文件。

26.1.4 用 Delete Sentry 进行文件恢复

假定要恢复 UNDELETE 列出的文件，可键入不带/LIST 的相同命令。试着恢复 PRACTICE 目录中的一些文件：

1. 键入 UNDELETE MEMMAKER.XXX 恢复指定文件。此命令产生的报告的开头部分与上一个例子中的相同，只是不列出所有文件。它一次显示一个文件并询问是否进行恢复：

   ```
   MEMMAKER XXX 17,237 5-31-94 6:22a...A Deleted: 2-11-95 11:13a
   This file can be 100% undeleted. Undelete(Y/N)?
   ```

   如果回答 N，UNDELETE 继续下一个与文件名匹配的文件。
2. 键入 Y。
   显示下列信息：

File successfully undeleted.

如果目录中包含一个欲恢复文件重名的文件，则显示：

The filename already exists. Type a different filename.
Press F5 to bypass this file.

可键入一新文件名，或按 F5 跳过它。如果键入文件名 MEMMAKER. YYY，屏幕显示：

MEMMAKER YYY　　17,237　5-31-94　6:22a　..A　Deleted:2-11-95 11:13a
File successfully undeleted。

注释： 若有多个被删除文件与键入的文件名相符，UNDELETE 逐一列出这些文件，并访问是否进行恢复。不必浏览整个列表，恢复了想要的文件后，按 Esc 退出 UNDELETE 即可。

26.1.5　删空 SENTRY 目录

如何删空 SENTRY 目录

若使用的是 Windows：

1. 在 Microsoft Tools 组中打开 Undelete。
2. 选中 Drive/Dir 按钮。
   打开 Change Drive and Directory(转换驱动器和目录)对话框。
3. 选中欲清除文件所在目录。
4. 选中 OK。
   文件清单显示所选目录中所有的被删除目录和文件。
5. 选中欲从 SENTRY 中清除的文件(注意：只有具有 Perfect rating 的文件才在 SENTRY 目录中)。
6. 选中 File，Purge Delete**Sentry** File。

若使用的不是 Windows：

1. 切换到欲清除的驱动器。
2. 键入 UNDELETE /PURGE。
3. 显示确认清除文件信息。
4. 键入 Y 清除文件。
   驱动器中的所有文件被清除。若不使用 Windows 就不能有选中地清除文件。

删除文件的主要原因之一是为了释放更多的磁盘空间。用户也许会认为 Delete Sentry 装满了使系统堵塞的被删除文件，这并不完全正确。每次加载 Delete Sentry 时，它自动清除 SENTRY 中七天前存入的文件。一旦 SENTRY 占据了驱动器的 20%以上，Delete Sentry 自动删除较早的文件，直到不超过这个限度(较早文件是指具有较早的删除日期/时间的文件)。同时，如果 DOS 得不到足够的空间存储新数据，Delete Sentry 将清除较早文件以留出足够的空间。

警告： 自动清除仅在加载 Delete TSR 的情况下才能进行。如果有一个 SENTRY 目录却不总是加载 Delete Sentry，则在 DOS 试图写入文件的时候，有可能出现 Disk Full 的出错信息。这时应再次加载 TSR 以得到所需空间。

可用 UNDELETE /PURGE 命令清除 SENTRY 中的所有文件。将显示下列信息：

Delete Sentry control file contains 123 deleted files.

Confirm parging of SENTRY files on drive C (Y/N)?

回答 Y 删空 SENTRY，回答 N 不删空它。文件记数显示了保存在此驱动器上的所有删除文件。使用 DOS 版本的 UNDELETE 无法清除单个文件或目录，可在 Windows 版本下做这件事。只需选择欲清除文件并在 File 菜单中选中 Purge Delete Sentry File 命令即可。

26.1.6 SENTRY 中的内容

SENTRY 是一通常不在 DOS 目录清单中显示的隐藏目录。但在 DOS 或其他应用程序中，都有一些浏览隐藏目录内容的办法。选用合适的方法浏览 SENTRT 目录，则会见到如下稀奇古怪的文件名称：

```
#A1B2C3E MS      17 01-13-93      6:42a
#A1B2C3F MS      89 01-13-93      5:18p
#A1B2C3G MS      339 02-09-93     9:37a
#A1B2C3H MS      421 02-09-93     9:40a
```

尽管名称看起来古怪，可它们正是被删除的文件。SENTRY 为它所保存的文件指定类名称，另有一控制文件联系原始文件名、目录与 SENTRY 目录中的类名。

26.2 UNDELETE 的目标目录

如果恢复被删除目录及其文件

若在非 Windows 下使用 Delete Sentry：

1. 创建与被删除目录同样名称和路径的新目录。
2. 像通常一样恢复目录文件。

若使用的是 Windows：

1. 在 Microsoft Tools 组中打开 Undelete。
2. 选中 Drive/Dir 按钮。
   打开 Change Drive and Directory 对话框。
3. 选中被删除目录的父目录。
4. 选中 OK。
   文件清单显示父目录中所有被删目录和文件。
5. 选中欲恢复目录。
6. 若 Undelete 按钮可选，选中它恢复目录(若此按钮不可选，则表示此目录无法恢复)。
7. 再次选中 Drive/Dir 按钮，选中恢复目录。
   关闭对话框时，文件清单显示恢复目录中的被删文件。
8. 选中欲恢复文件并选中 Undelete 按钮。

注释： 若未使用 Windows 或 Delete Sentry，则无法恢复目录。

DOS 版本的 UNDELETE 总是在与被删文件相同的目录中恢复文件。如果已删除或重

新命名了一个目录，则无法恢复其中的文件，除非再创建一个相同名称和路径的新目录。如果使用 DELTREE 删除了整个分枝，则所有的文件被移入 SENTRY 中，在恢复文件前必须重建目录。

26.3　当 UNDELETE 无法恢复文件时

即使有 Delete Sentry，也有可能无法恢复某个文件。Delete Sentry 或许早已把它清除掉了，特别是在磁盘已经很满，DOS 经常需要从 Delete Sentry 得到空间时。另外带有扩展名 .TMP、.VM?、.WOA、.SWP、.SPL、.RMG、.IMG、.THM 和 .DOV 的文件不受 Delete Sentry 的保护(Delete Sentry 将这些文件视为临时文件)。并且作为缺省设置，UNDELETE 仅保护具有正的归档属性的被删文件。至于其他文件可从它们的备份进行恢复。

提示： 如果想删除一组文件，使得 UNDELETE 不再保存它们以留出更多的空间，在删除前关闭其归档属性即可。

如果 Delete Sentry 无法恢复一个文件，可从即备份重建它，或者使用 DOS 目录方法恢复它，具体方法留待本章稍后讲述。

26.4　删除追踪的使用

如何使用 Deletion-Tracking 恢复文件

1. 如果在此驱动器上仅使用了 Deletion-Tracking，可用下述命令列出可恢复文件：

   UNDELETE [filespec] /LIST

   如果在此驱动器上用了 Delete Sentry，则应加上/DT 选项迫使 UNDELETE 使用删除追踪方法：

   UNDELETE [filespec] /LIST /DT

2. 从清单中选中所要恢复的文件。
3. 再次键入 UNDELETE 命令，不加/LIST 选项。

   UNDELETE 逐一列出可恢复文件，并询问是否恢复它。当由于出现串丢失而使某个文件无法完全恢复时，UNDELETE 发出警告。
4. 对每个不需恢复的文件，键入 N。
5. 对每个需要恢复的文件，键入 Y。

   如果恢复的文件与同一目录中的某个已存在的文件重名，UNDELETE 要求提供一新名称。
6. 恢复了所有想要的文件后，键入 ESC 退出 UNDELETE。

对于从 DOS 5 升级上来的用户，也许会把删除追踪作为保护被删文件的一种老办法。受删除追踪保护的驱动器在其根目录下有一名为 PCTRACKR.DEL 的系统文件。PCTRACKR.DEL 保存了一被删文件串清单，恢复文件即是将 DOS 未再次使用的串恢复原状的过程。

如果有了 Delete sentry，几乎毫无必要再删除追踪。对于保护被删文件来说，Delete Sen-

try 干得要漂亮得多。只要时常加载它，它能有自动清除较早文件，从而使占用磁盘空间这一事实并不引入注意。当然对于一直使用 DOS 5 的删除追踪并且不想换个方法的用户来说，也可在 DOS 6 下使用这种方法。在 UNDELETE 命令用/T 代替/S 就可加载删除追踪。不能在同一命令中同时使用/T 和/S 选项，也就是说，不能同时加载 Delete Sentry TSR 和 Deletion-Tracking TSR。下面的命令加载了删除追踪 TSR，并将驱动器 C 加入其驱动器清单中。

```
UNDELETE /TC
```

注释： Deletion-Tracking 文件的入口缺省数目取决于驱动器的大小。要增加或减少 Deletion-Tracking 文件中的入口数，只需在/T 选项后加上 entries。比如，要在 D 盘的 Deletion-Tracking 文件中设置 500 个入口，只需在加载 UNDELETE 时使用/TD-500 选项。

26.4.1 列出受 Deletion-Tracking 保护的文件

答疑： 我在 DOS Shell 中删除了一些文件，UNDELETE 未将它们作为可恢复文件列出来。我该如何恢复这些文件？

如果你使用的是删除追踪而不是 Delete Sentry，则 DOS Shell 或 Windows 中删除的文件不受保护。你无法使用删除追踪方法恢复它们。不过也许可用 DOS 目录方法完全或部分恢复它们。在 UNDELETE 命令中加上/DOS 选项，看看能否用这种方法恢复文件。

如果使用的仅仅是删除追踪方法，对于恢复刚擦除的文件来说，也许可以做到安全有效。但如果在擦除文件后，计算机进行了某些操作，即使只有短短几分钟，删除追踪所恢复的可能只是些错误的数据（具体原因在下节中阐述）。在这种情况下，明智的方法是从备份恢复文件。如果没有备份，那么只好采用下述方法。

当在使用删除追踪的系统中键入 UNDELETE /LIST 命令时，将显示与下面类似的信息：

```
Directory: C:\PRACTICE
File Specifications: *.*

Delete Sentry control file not found.
Deletion-tracking file contains      5 deleted files.
Of those,          5 file have all clusters available,
                   0 files have some clusters available,
                   0 files have no clusters available.
MS-DOS directory contains      6 deleted files.
Of those,      4 files may be recovered.

Using the Deletion-tracking method.

DEFRAG      XXX      9,227  5-31-94  6:22a  ...A  Deleted:  2-11-95  11:13a
DOSSHELL    XXX    161,323  5-31-94  6:22a  ...A  Deleted:  2-10-95  10:35a
HELP        XXX    296,844  5-31-94  6:22a  ...A  Deleted:  2-10-95  10:35a
MEMMAKER    XXX     17,237  5-31-94  6:22a  ...A  Deleted:  2-11-95  11:13a
QBASIC      XXX    130,881  5-31-94  6:22a  ...A  Deleted:  2-11-95  11:13a
```

这些信息表示使用的是删除追踪而不是 Delete Sentry。在指定目录中有 5 个文件可用删除追踪恢复。UNDELETE 已将删除追踪文件中的串清单与 FAT 进行了比较，并确定这些文件的所有串都未被占用。可用如下命令恢复一些文件：

```
UNDELETE D*.XXX
```

UNDELETE 逐一显示文件，并询问是否恢复它：

```
DEFRAG XXX 9,227 5-31-94 6:22a ...A Deleted: 2-11-95 11:13a All of the clusters of this file are available. UNDELETE(Y/N)?
```

若回答 N，UNDELETE 继续下一个文件，若回答 Y，则显示信息：

```
File successfully undeleted.
```

26.4.2 部分恢复文件

有时删除追踪仅能恢复文件的一部分。在这种情况下，当键入 UNDELETE /LIST 时，显示的信息与下面的类似：

```
Deletion-tracking file contains 3 cleleted files.
Of those,       1 files have all clusters available,
                1 files have some clusters available,
                1 files have no clusters available.
```

文件列表显示如下：

```
  T4          CAT        17  1-13-93  6:42a  ...A  Deleted:  2-12-94  5:45a
* BIGTIME     ABC    39,152  3-10-93  3:10a  ...A  Deleted:  2-12-94  7:32a
* * ZIGZAG    BMP       630  3-10-93  3:10a  ...A  Deleted:  2-12-94  7:29a
* indicates some clusters of the file are available.
* * indicates no clusters of the file are available.
```

ZIGZAG.BMP 已无法恢复，因为已无法得到它的串。一些 BIGTIME.ABC 串已无法恢复，不过另一些可恢复。在决定是否恢复文件的一部分时，应考虑下列因素：

- 可恢复的部分并不一定是文件的开头，甚至不一定是连续的。它也许是一些分布在各处的串。
- 总的来说，从备份恢复文件是最好的办法，即使备份有点过时。
- 部分恢复不包括文本的文件是不明智的，因为其结果毫无用处。

警告： *不要试图部分恢复一个程序文件，运行这样的程序只会给系统带来危害。*

- 如果要恢复的文件来自字处理器、桌面印刷系统或数据库管理器，这些应用程序为文档生成的特殊标题信息可能会丢失，如果这样，用户就不能在原始应用程序中打开该文件进行编辑，但可以试着把它作为 ASCII 文本文件编辑，清除不必要信息，尽可能地恢复文本。

即使 UNDELETE 报告文件完全可恢复，其最终结果也不一定完全正确。也许另一个文件已覆盖了某些串，只不过又被删掉了而已。由于这些串对于 UNDELETE 来说看起来是可得到的，它就报告说可恢复整个文件。事实上，它从这些串上恢复的是其他文件的数据。

26.4.3 恢复在 DOS 5 下删除的文件

如果刚从使用删除追踪的 DOS 5 升级上来，在 PCTRACKR.DEL 中也许仍有 DOS 5 下删除的文件。这时如果换用 DOS 6 的 Delete Sentry，UNDELETE 不一定能恢复删除追踪法文件。如果要恢复一个升级并换用 Delete Sentry 前删除的文件，应在 UNDELETE 命令中加上/DT 选项迫使 UNDELETE 访问 PCTRACKR.DEL 文件。假设几天前刚在 DOS 5 下删除了 GRFALLS.DRW，如果串完好无损，可用下面命令恢复：

```
UNDELETE GRFALLS.DRW /DT
```

提示： 使用 Delete Sentry 大约一周后，可删除 PCTRACKR.DEL。它的数据已变得非常过时而无可用之外。PCTRACKR.DEL 是一系统文件，位于所保护的各个驱动器的根目录下。在删除它或使用 DELTREE PCTRACKR.DEL 前，必须关闭其系统属性。

26.5 DOS 目录方法的使用

如何从 DOS 目录恢复文件

1. 如果在某个驱动器上从未使用过 Delete Sentry 或删除追踪，可用以下命令列出可恢复文件：

   UNDELETE [filespec] /LIST

   如果使用过 Delete Sentry 或删除追踪，则在命令中加上/DOS 选项迫使 UNDELETE 使用 DOS 目录方法：

   UNDELETE [filespec] /LIST /DOS

2. 决定表中需恢复的文件。
3. 重新键入不带/LIST 选项的命令。

   UNDELETE 逐一列出可恢复文件，并询问是否恢复它。
4. 对每个不想恢复的文件键入 N。
5. 对每个需要恢复的文件键入 Y。

   UNDELETE 要求提供文件名的第一个字符。
6. 键入文件名的第一个字符。

   如果键入的第一个字符不引起文件重名，UNDELETE 恢复此文件，否则 UNDELETE 要求提供另一个首字符。
7. 恢复了所有想要的文件后，按 Esc 结束 UNDELETE。

通过搜寻目录中的被删文件项来恢复文件是最不可靠的方法。UNDELETE 只在找不到 SENTRY 目录或 PCTRACKR.DEL 文件时才使用这种方法。即使指定一特定文件名，该文件只在 DOS 目录中而不在 SENTRY 或 PCTRACKR.DEL 中出现，只要在此驱动器上建有 SENTRY 目录或 PCTRACKR.DEL 文件，UNDELETE 就不使用目录方法。

如果一个文件已从 SENTRY 或 PCTRACKR.DEL 中清除，只要它的原目录项仍然存在，就可以用目录方法恢复它。可在 UNDELETE 命令中加上/DOS 选项迫使 DOS 使用目

录方法，如：

```
UNDELETE \TEST1\*.* /DOS /LIST
```

屏幕显示的信息与下面类似：

```
Directory: C:\

    File Specifications: *.*
        Delete Sentry control file not found.
        Delete-tracking file not found.
    MS-DOS directory contains 5 deleted files.
        Of those,    3 files may be recovered.

    Using the MS-DOS directory method.

* *   ? ONFIG      TMP       778        1-05-93  10:24a    ...A
      ? OC2D31     TMP     7,059        12-05-93  4:278    ...A
* *   ? MFO3OD     TMP        18        12-05 93  4:27p    ...A
      ? OREAD      TXT '   22,016         12-06-92  6:00a  .....
* *   indicates the first cluster of the file is unavailable and can't
    be recovered with the UNDELETE command.
```

清单包括 C:\TEST1 目录中所有的名字以∑起头的文件。清单中的？号表示那个字符。即使一个文件的目录项仍然存在，但如果它的第一个串被另一个文件占用的话，它就无法恢复，如表中 * * 所示。

清单中也许还有些出乎意料的文件名，Delete Sentry 和删除追踪不保护临时文件，但目录方法却可以显示它们。此清单显示 DOS、Windows 及其他应用程序创建又删除的临时文件，它们以类名称如～MS396AX. TMP 显示。

假如要恢复几个文件，可用下面不带/LIST 的 UNDELETE 命令：

```
UNDELETE C:\TEST1 \*.* /DOS
```

此命令产生的屏幕显示的开头与带/LIST 的相似，只是逐一列出文件，并询问是否恢复：

```
? REP      BAT     63 12-03-92  9:54a     UNDELETE (Y/N)?
```

如果回答 N，UNDELETE 接着处理下一个文件，如果回答 Y，它提示进行下面的操作：

```
Please type the first character for ? REP      .BAT:
```

一旦键入字符所构成的文件名与目录中其他文件不重复，屏幕显示：

```
File successfully undeleted.
```

注释： UNDELETE 列出无法恢复的文件，这意味着任何可能恢复它们。

26.5.1 被删目录

一旦一个目录被删除，DOS 版本的 UNDELETE 就无法使用目录方法恢复其中的文件，即使创建一个同样名称和路径的目录也于事无补。因为被删除的原有目录项不在新目录中。Windows 版本的 UNDELETE 或许能像恢复文件一样恢复被删目录。

26.5.2 目录方法的缺陷

目录方法所显示的 File successfully undeleted 信息可能具有欺骗性。UNDELETE 在找到文件的目录项和第一个串后，仍需寻找文件的其余串。过的时间越长，在磁盘上进行的操作越多，UNDELETE 就越有可能找不到正确的串，特别是在文件比较破碎的情况下。

只有在删除之后立即进行恢复，才有把握用此方法准确恢复文件。如果已在系统上进行了某些操作，不如从备份建立文件更为可靠。

警告： 不要运行用 DOS 目录方法恢复的程序，如果它包含了一些错误信息，有可能损坏系统。

26.6 无提示恢复文件

如果在 UNDELETE 命令中包括一特定文件名，则 UNDELETE 直接恢复此文件而无进一步的提示。比如 UNDELETE /LIST 显示有下列文件可恢复：

```
Directory: C:\PRACTICE
File Specifications: *.*
Delete Sentry control file contains      5 deleted files.
Deletion-Tracking file not found.
MS-DOS directory contains   6 deleted files.
Of those,   4 files may be recovered.
Using the Delete Sentry method.
DEFRAG      XXX       9,227  5-31-94    6:22a  ...A  Deleted:   2-11-95   11:13a
DOSSHELL    XXX     161,323  5-31-94    6:22a  ...A  Deleted:   2-10-95   10:35a
HELP        XXX     296,844  5-31-94    6:22a  ...A  Deleted:   2-10-95   10:35a
MEMMAKER XXX         17,237  5-31-94    6:22a  ...A  Deleted:   2-11-95   11:13a
QBASIC      XXX     130,881  5-31-94    6:22a  ...A  Deleted:   2-10-95   11:13a
```

如果想要直接恢复 DEFRAG.XXX，可键入 UNDELETE DEFRAG.XXX。

如果想要直接恢复所有的文件而不加提示，可键入 UNDELETE /ALL。如果使用的只是 DOS 目录方法，UNDELETE 自动设定每个文件名的首字符，比如用＃号。如果用文件名和/DOS 则仍会出现提示。

26.7 在 DOS Shell 中恢复文件

DOS Shell 的 Disk Utilities 目录中包含 Undelete 程序项。选择此项可打开一对话框，在其中可键入 UNDELETE 的命令参数。/LIST 选项是唯一的默认参数，如果要添加一个文件名，应确保插入在/LIST 选项前，因为在 UNDELETE 命令中文件名必须在选项之前。按 Enter 则出现 DOS 命令提示符屏幕，就可开始工作了。

26.8 Windows 的 Undelete 操作

图 26.3 显示了 Undelete 窗口，从 Microsoft Tools 组中选中 Undelete 就可打开它。其中

的文件清单显示了用各种方法可恢复的文件。当状态为 Perfect(完美)时,表示此文件受 Delete Sentry 保护;Lesser(稍差)状态表示文件受删除追踪保护;如果文件名的第一个字符是一问号,则表示使用的是 DOS 目录方法。文件清单有时还显示一些无法恢复的目录中的文件,它们的项变暗淡,且状态为 Destroyed(受损)。

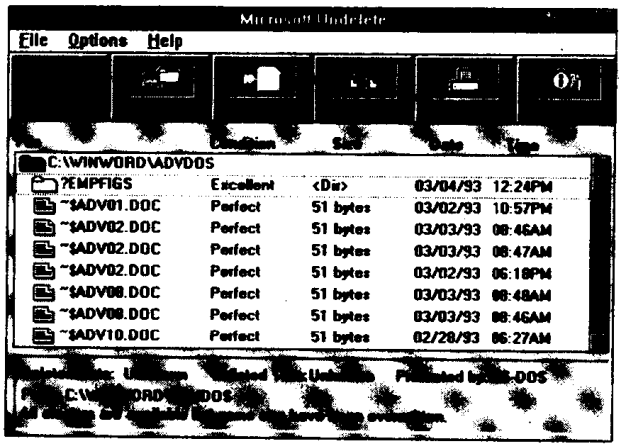

图 26.3　Windows Undelete 窗口

欲恢复文件,只需在文件清单中选定文件然后选中 Undelete 按钮即可。可按 Drive/Dir 按钮转换驱动器及目录。

26.8.1　Windows Undelete 的特点

用 Find 按钮可打开一对话框,在其中可键入文件名或文本进行搜寻,搜寻范围包括整个当前驱动器。同样也可搜寻一组文件,如 pbrush(Windows Paintbrush)组搜寻所有的 *.PCX 和 *.BMP 文件。

Sort By 按钮用于选定文件清单的分类标准。Print 按钮用于在默认打印机上打印当前文件清单。按 Info 按钮所打开的对话框则提供了关于当前选中文件的附加信息,如首串数。

图 26.4 显示了 File 菜单,其中的大多数命令与所见的按钮功能相同,不过也有一些是其特有的。Undelete To 用于选择存放恢复文件的目录。Purge Delete Sentry File 命令则用于清除 Delete Sentry 中选中的文件。

图 26.5 显示了 Options 菜单。选中 Select By Name 用于键入恢复或清除的文件名。Unselect By Name 则用于键入退选文件的样本名。

Configure Deletion Protection 选项可用来打开一对话框,在其中可选择 Delete Sentry、Delete Tracker 或 Standard(即 DOS 目录项方法)。如果选中了其中之一,则出现一个附加的对话框可用于设定保护的文件,文件在 SENTRY 中保存的天数等选项。Undelete 根据用户的选择来刷新 UNDELETE.INI 文件,但为了使之有效,必须重新启动系统。

26.8.2　目录的恢复

可用 Windows Undelete 恢复被删除的子目录。图 26.3 中名为? EMPFIGS 的项就是一个被删目录,从图标和大小栏可看出来。这个目录未用任何方法加以保护,不过可用 DOS 目

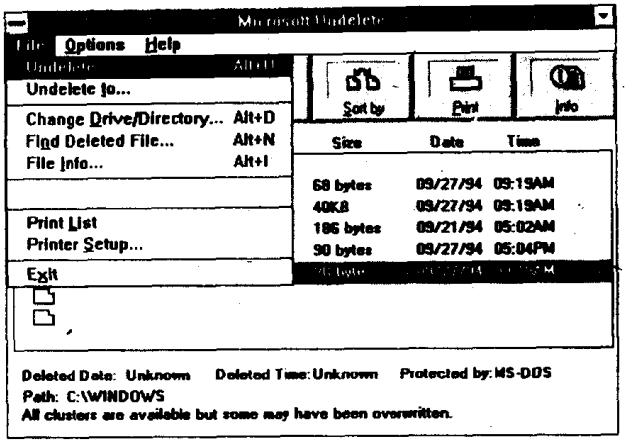

图 26.4 Windows Undelete File 菜单

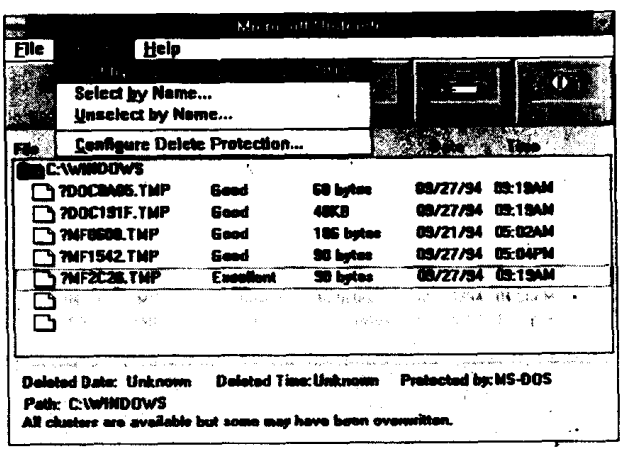

图 26.5 Windows Undelete Options 菜单

录方法加以恢复。目录被恢复后，其中的文件也就自然被恢复了。事实上，它们可用 Delete Sentry 进行保护。

26.9 UNDELETE.INI 文件的设定

如何配置 Undelete

如果使用的是 Windows：

1. 在 Microsoft Tools 组中打开 Undelete。
2. 选中 Options Configure Delete Protection。
   打开 Configure Delete Protection 对话框。
3. 选择 Delete Sentry、Delete Tracker 或 Standard。
4. 选中 OK 按钮。
   如果所选的是 Delete Sentry 或 Delete Tracker，则打开一配置对话框。

5. 用对话框配置 UNDELETE。
6. 选中 OK 存储新配置。
   出现提示信息，建议重新启动系统以使配置生效。
7. 关闭所有的应用程序，退出 Windows，重新启动。

如果使用的不是 Windows：

1. 用 ASCII 文本编辑器如 EDIT 打开 C:\DOS\UNDELETE.INI 进行编辑。
2. 逐行改动文件以配置成合适的 Undelete。
3. 保存文件，退出编辑器。
4. 键入 UNDELETE /U 下载 UNDELETE。
5. 键入 UNDELETE /LOAD 加载新配置的 UNDELETE。

UNDELETE 将所有的配置信息保存在名为 UNDELETE.INI 的文件中。可用 ASCII 文本编辑器如 Editor 编辑此文件以改变配置。必须分外小心不要改动文件的基本结构，所要做的只是改变不同项的设置。比如，Delete Sentry 被设置为保护驱动器 C 和 D，则在 UNDELETE.INI 中可找到如下部分：

```
[sentry.drives]
C=
D=
```

如果要去除对驱动器 D 的保护，只需删除 D=行。但小心不要删除或更改头[Sentry.drives]。

如果不考虑归档属性而保存文件，将下行中的 FALSE 改为 TRUE 即可：

```
archive=FALSE
```

要改变 SENTRY 可占用的盘区百分比，可改变下行中的数字：

```
percentage=20
```

要改变文件在 SENTRY 中保存的天数，则可变动下行中的数字：

```
days=7
```

下面一行决定了保存以略过的文件类型。*.* 告知 Delete Sentry 保存所有的文件。而后的每一个 filespec 去除对某一特定类型文件的保护。换句话说，这行就是让 Delete Sentry 保存除.TMP、.VM、.WOA 等以外的所有文件。

```
Sentry.files=*.* -*.TMP -*.VM -*.WOA -*.SWP -*.SPL -*.RMG -*.IMG
-*.THM -*.DOV
```

你可以自己添加行中的例外文件类型。比如，若不想保护 *.BAK 文件，则可在行后加上-*.BAK。

26.10 自测练习

本章讲述了如何使用 UNDELETE 保护文件及在必要时恢复文件。以下练习可帮助你熟悉这些技术：

1. 加载 Delete Sentry TSR 保护启动驱动器。
2. 检查目前 UNDELETE 保护的驱动器。
3. 创建一练习目录，拷贝一些文件到其中，删除它们，然后进入恢复。
4. 如果使用了 Windows 并安装了 Windows 版本的 Undelete，试着在 Windows 中删除和恢复文件及目录。
5. 如果不想继续使用 Delete Sentry，下载此 TSR 并清除 SENTRY 目录(若经常使用 Delete Sentry，则不必这么做)。

第 27 章　CHKDSK 和 SCANDISK

简介： CHKDSK 对驱动器来说有两大功能：用于解决可能导致数据破坏的问题及挽回浪费的空间。SCANDISK 除了这些功能以外还可从物理损坏的扇区中挽救数据。

27.1　CHKDSK 和 SCANDISK 的用途

CHKDSK 和 SCANDISK 检查驱动器的 FAT 和目录结构，寻找可能导致无法访问文件的错误和不一致之处。它们可解决找到的大部分问题。它们还可找到不属于任何文件又未被删除的簇(即丢失的簇 lost clusters)，用户可自行决定删除它们，还是将它们转换为文件。用此方法可挽回驱动器中上千字节的空间。

SCANDISK 比 CHKDSK 新得多，在识别和解决 FAT 及目录结构问题方面也要好得多。它可用来检测磁盘表面，识别损坏簇，并将尽量多的数据移到正常簇中，同时阻断坏簇以防再次使用。

SCANDISK 是 DOS 6.2 新引入的，如果你已升级到 DOS 6.2 或更新版本，则不必再用 CHDKSK 修理磁盘，尽管有时它的报告看起来更为方便。如果使用的是稍早的 DOS 版本，则还得用 CHKDSK 来诊断和解决问题。

警告： 应设法拥有一磁盘表面扫描工具。如果不打算升级到 DOS 6.2，可在第三方应用软件包中找到出色的表面扫描工具，如 Norton Utilities 的 Disk Doctor。在安装数据压缩程序，如 DblSpace 或 DrvSpace 前扫描磁盘表面尤为重要。

本章讲述了如何使用 CHKDSK 和 SCANDISK。你将了解如何：

- 确定何时使用磁盘检查工具
- 运行 CHKDSK
- 运行 SCANDISK
- 处理 CHKDSK 或 SCANDISK 报告的一般问题

27.2　使用 CHKDSK 或 SCANDISK 的时机

应定期运行 CHKDSK 或 SCANDISK 检查错误及丢失的簇。对于全时使用系统的用户来说，应每周运行一次 SCANDISK 或 CHKDSK。对于使用较少的系统，每两周或每月一次比较合适。当然在遇到下列问题的迹象时，应即时运行 CHKDSK 或 SCANDISK。

- 文件数据丢失
- 文件中出现其他数据
- 原来运行正常的程序开始出问题

- 文件从目录中丢失
- 目录中包括无意义信息
- 目录突然丢失
- 程序破碎为一连串临时文件

27.3 检查驱动器问题

CHKDSK 和 SCANDISK 会因打开的文件而出错，因此在运行它们之前应结束所有运行的程序(除 TSR 外)。特别是应退出 DOS Shell 或 Windows，而后在初始命令提示符下运行 CHKDSK 或 SCANDISK。

试对驱动器 A 中的软盘使用 CHKDSK：

1. 在驱动器 A 中插入一格式化盘。
2. 退出任何程序或外壳回到初始命令提示符。
3. 键入下面命令开始 CHKDSK：

```
CHKDSK A:
```

CHKDSK 检查磁盘并产生一报告。它也许会报告磁盘的一些问题，但并不纠正它们。所出的问题将在下节解释。

不管是否报告问题，报告的余下部分与下面的例子相似。前几行以字节为单位描述了磁盘空间状况：

```
1,457,664 bytes total disk space
  923,648 bytes in 27 user files
  534,016 bytes available on disk
```

接着的几行以分配单元(簇)显示了磁盘空间状况：

```
  512 bytes in each allocation unit
2,847 total allocation units on disk
1,043 available allocation units on disk
```

最后两行描述了内存的使用状况，但仅仅是最基本的 640KB：

```
655,360 total bytes memory
611,248 bytes free
```

最后如果使用的是 DOS 6.2 或更新版本，将显示信息建议使用 SCANDISK：

```
Instead of using CHKDSK, try using SCANDISK. SCANDISK can reliably detect and fix a
much wider range of disk problems. For more information, type HELP SCANDISK from the com-
mand prompt.
```

注释： *在 DOS 的早期版本中，没有内存报告。第 29 章"内存管理"讲述了找出可用内存的更好办法。*

SCANDISK 比 CHKDSK 更新更好，但如果它发现第三方压缩软件如 Stacker 存在，就无法修理任何磁盘(甚至包括未压缩的)，并且无法进行表面扫描。因此如果使用了第三方软件，请跳过下面的练习(SCANDISK 与 DOS 的磁盘压缩程序 DblSpace 和 DrvSpace 兼容，如果使用的是这些程序，可继续下面练习)。

若有 DOS 6.2 或更新版本，试着在同一磁盘上运行 SCANDISK：

1. 退出所有程序和外壳，回到初始命令提示符。
2. 用下述命令开始 SCANDISK：

SCANDISK A:

打开 SCANDISK 窗口，SCANDISK 每完成一项测试就勾掉一条目（如图 27.1 所示）。

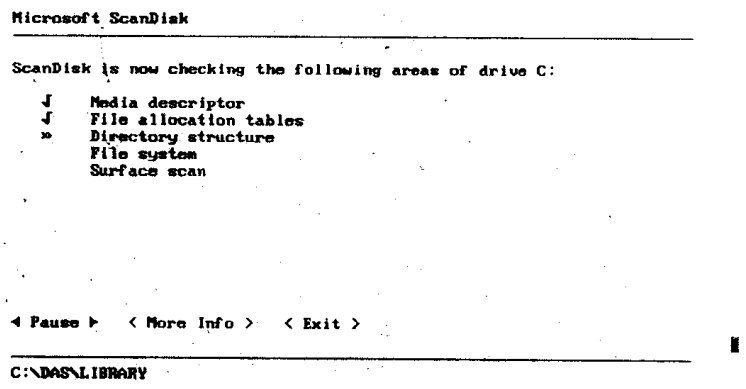

图 27.1　SCANDISK 窗口

一旦出现出错信息，选中 Stop 中止扫描阅读下一节。如果不结束扫描，则 SCANDISK 结束检查 FAT 和目录结构并询问是否进行磁盘表面扫描。

3. 软盘的表面扫描需花几分钟时间，如果想试一下表面扫描，选中 Yes；否则，选 No。

如果选择进行表面扫描，进程屏幕将显示 SCANDISK 的工作进程。

完成 SCANDISK 后，将显示一总结报告。用户可选择显示一更详细的报告或退出。

4. 选中 Exit 结束 SCANDISK。

27.3.1　磁盘问题的处理

CHKDSK 和 SCANDISK 报告的最常见错误为丢失的分配单元(lost allocation unit)。当 FAT 项表明一串属于某文件，而相应的链不能找到特定文件的目录项时，就会发生这种错误。

簇丢失有多种方式，任何类型的 FAT 或目录结构错误都有可能引致簇丢失。假定某目录项受损而不再指向文件的第一个簇，这样这个文件的簇就不再依附于此目录项，CHKDSK 和 SCANDISK 就视之为丢失的分配单元。假定一个文件的第二个簇的 FAT 项受损而不再指向第三个簇，于是第三个簇及以后的簇就丢失了。由于硬件的错误操作，电流波动及未正确停放的磁头等原因，常会造成这类错误。

也可能在不引起磁盘任何其他错误的情况下，存在丢失的簇。在 DOS 删除文件时，中断操作是引起簇丢失的最普遍的情形。如果在 DOS 删完被删文件的所有 FAT 项前，重新启动、取出磁盘或关闭电源，剩下的项就成了丢失的分配单元。当运行某程序时造成死机，不得不用重新启动代替正常的文件关闭时，也会造成簇丢失。

产生丢失簇的可能性要比想象的多得多。系统运行几个月后，它们会占据上千字节的磁盘空间。这些簇不能用于存放新数据，只是浪费了存储空间。CHKDSK 和 SCANDISK 可用于释放这些空间。

CHKDSK 和 SCANDISK 为从丢失簇恢复空间提供了两种办法：将它们转换为文件或删除它们。如果未缺失任何数据，并且 CHDKSK 或 SCANDISK 也没有报告驱动器有其他问题，删除这些簇既方便又安全。如果缺失了一些数据，或者 CHKDSK 报告有其他问题，则应将这些簇恢复为文件，以在可能的情况下置放并挽救丢失的数据。

答疑： 在用 SCANDISK 或 CHKDSK 恢复文件时，得到信息 Insufficient room in root directory，是怎么回事？

每个根目录有其特定的项数，其最大值取决于磁盘大小。SCANDISK 或 CHKDSK 可能找到上百个丢失的链，特别是在任其堆积或 FAT 及目录结构中约其他错误导致上百个文件受损时。所有的这些文件都存放于根目录中。

一旦得到这条信息，就应检查并处理所有的 FILEnnnn.CHK 文件。尽量删除这些文件，对于需要保存的，将它们拷贝到其他目录中去。然后再键入 SCANDISK 或 CHKDSK /F 命令恢复更多的丢失分配单元。

27.3.2 使用 CHKDSK 处理簇丢失

CHKDSK 在两行题头后报告发现的问题。下面显示的是一条报告簇丢失的信息：

```
Volume MICROMATE created 10-27-1994 10:30a
Volume Serial Number is 1A4F-4156
Errors found,F parameter not specified
Corrections will not be written to disk

  45 lost allocation units found in 13 chains.
    92160 bytes disk space would be freed
84,768,768 bytes total disk space
    88,064 bytes in 6 hidden files
   126,976 bytes in 41 directories
71,376,896 bytes in 1793 user files
13,084,672 bytes available on disk
     2,048 bytes in each allocation unit
  41,391 total allocation units on disk
   6,389 available allocation units on disk
 655,360 total bytes memory
 627,424 bytes free
```

在这个例子中，CHKDSK 在 13 条链中找到了 45 个丢失的簇。一旦清除这些簇就可得到 92 160 字节(90KB)的可用空间。如果将它们转换为文件，CHKDSK 将创建 13 个文件，每条链一个。

在上面的命令中，CHKDSK 不恢复丢失的簇，也不处理任何其他问题。如果想这么做，必须在命令后加上/F 选项，即：

```
CHKDSK /F [drive]
```

使用了/F 选项，就会看到下列信息：

```
Volume MICROMATE created 10-27-1992 10:30a
```

```
Volume Serial Number is 1A4F-4156
    45 lost allocation units found in 13 chains.
Convert lost chains to files (Y/N)?
```

如果键入 N，CHKDSK 删除丢失的簇，释放空间用于新数据。如果键入 Y，CHKDSK 将每个丢失的链转换为根目录中名为 FILEnnnn.CHK 的文件，其中 nnnn 是从 0000 开始的序号。然后就可以分别检查每个文件，决定是保留还是删除它。检查文件可用 DOS 的 EDIT 命令或其他字处理软盘。非文本文件自然是不可读的，即使是文本文件也可能只是部分可读。在存放文件前应删去无用部分。

27.3.3　使用 SCANDISK 处理簇丢失

SCANDISK 发现丢失的簇后，显示 Problem Found 对话框，如图 27.2 所示。选中 Save 可将簇恢复为文件，或 Delete 删除簇。其结果与用 CHKDSK 一样。也可选择 Don't Fix It 略过此问题或 More Info 找出有关丢失的簇的更多信息以利决择。

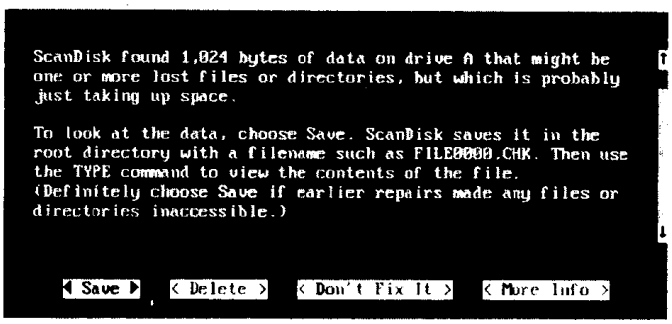

图 27.2　Problem Found 对话框

对话框中只显示了一部分信息，要读余下的部分可用滚动条或 PageDown 进行滚动。也可选择 More Info 以阅读关于问题更详细的解释。

无论何时 SCANDISK 纠正磁盘的错误，它都将恢复信息存储起来，因此如果不喜欢，可改回原状。若选图 27.2 中的 Save 或 Delete，则打开图 27.3 所示的对话框。选中 Drive A（或可能的话，Drive B）存储恢复信息或用 Skip Undo 略过。此处的选择控制了 SCANDISK 接下来要做的事。如果随着扫描的继续，发现了更多的错误，那时已没有机会决定是否存储恢复信息了。也可在作出决定前选择 More Info 参阅有关恢复功能的更多信息。

若选择存储恢复信息，必须在驱动器 A 或 B 中插入一张空白的格式化软盘。SCANDISK 将把恢复信息存放在软盘上。

仅在运行 SCANDISK 后，驱动器未进行任何变动时，才能恢复 SCANDISK 的变化。即使添加、展开、压缩了一个文件，FAT 就会发生变化，恢复信息（其中包括旧 FAT）就无法得到了。牢记这一限制。可用下面的命令恢复 SCANDISK 的变化：

```
SCANDISK /UNDO undo-drive
```

其中的 undo-drive 指包含了恢复盘的驱动器，而不是修复的驱动器。

27.3.4　交缠文件

CHKDSK 不能而 SCANDISK 能纠正的第二种常见错误也许是交缠文件（Cross-linked

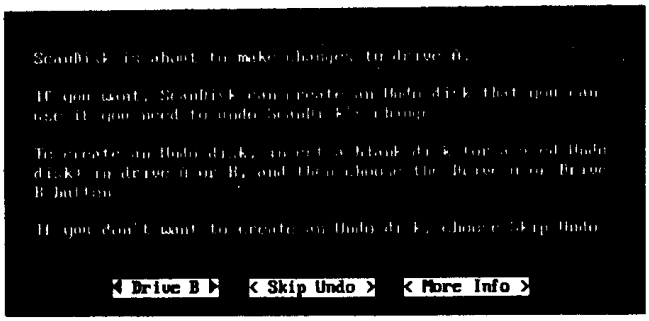

图 27.3 Create Undo Disk 对话框

files)，这是指两个文件共有某些簇。比如，FAT 显示 FILEA 占据簇 201、202、203 和 204；FILEB 占据了簇 1096、1097、203 和 204。簇 203 和 204 不可能同属两个文件，一定是有一条链出错了。可能性最大的就是 FILEB 应从 1097 链接到 1098 及以上。

CHKDSK 无法修复交缠文件，因为它无法猜测链的对错。修复交流文件最好的办法当然是从备份重新建立，但如果没有备份并且出问题的不是程序文件，则也许至少可用下述的一种办法恢复文件。

SCANDISK 完成操作中的步骤 2 和 3，如果用户要求修复交缠文件的话。使用者仍需检查和编辑修复的文件并处理所有其他的 CHK 文件。

预览： CHKDSK 和 SCANDISK 所能发现和纠正的错误非常少见，因此留到第 47 章“高级补救技术”讲述。

如何用 CHKDSK(如果没有 SCANDISK 的话)恢复交缠文件

警告： 不要对交缠的程序文件采取此操作。运行错误修复的文件会损害硬盘。删除出错文件，重新进行安装。(这对扩展名为.COM、.EXE、.BIN、.DLL、.SYS、.OVL 的文件适用。)

1. 将所有丢失的分配单元恢复为文件，然后退出 CHKDSK。(将在稍后回到恢复的文件。)
2. 对每一个交缠文件作一拷贝，并给每个拷贝一个新名，但扩展名与原来的相同。新文件不再是交缠的，但每一个都包含相同簇的拷贝。
3. 删除交缠文件，以清除 FAT 中的问题。
4. 用编制原来文件的应用程序检查每个拷贝。
   也许会发现一个文件是好的，而另一个不是。
5. 编辑受损文件移去多余的数据。注意使用最初创建此文件的应用程序。
6. 检查 CHKDSK 恢复的 FILE nnnn.CHK 文件。如果其中包含受损文件的数据，用相同的应用程序插入到合适的位置。
7. 处理其余 FILE nnnn.CHK 文件。
8. 将恢复的文件换回原名。

27.4 过时的 CHKDSK 功能

CHKDSK 是早期的 DOS 程序，它包括一些曾经非常方便但现在已被更好的程序取代的功能。除了报告安装的内存量外，这还包括两个过时的功能：

- 它可报告文件的碎片化，但 DEFRAG 更好。
- 它可列出盘上的所有文件，但 DIR 和 TREE 更好。

27.5　SCANDISK 的表面扫描

当磁盘初次低级格式化时，DOS 探测其表面，识别不能可靠存储数据的点，并将其在 FAT 中标识出来。但格式化后的磁盘在使用过程中，坏区会增长。当出现新的坏点时，将显示信息 Disk read error（磁盘读错误）或 Disk write error（磁盘写错误）。即使大多数的文件是完好的，不用 SCANDISK，也无法访问存在坏区中的文件。SCANDISK 可逐扇区读取文件，挽回尽量多的扇区。这样失去的仅仅是存放在真正损坏的扇区中的数据。

SCANDISK 在检查完 FAT 和目录结构后，可进一步扫磁盘表面（见图 27.4）。如果发现问题，它就把所有能够准确读取的数据从坏簇移到别的地方。它同时在 FAT 中对坏簇进行标记以不再使用它们。在大多数情况下，SCANDISK 无法挽回整个文件，却可挽回一大部分。当然如果存有备份的话，不如删除挽回的文件，而从备份加以恢复。否则就不得不保留挽回的文本文件，并重新输入丢失的数据。作为一条准则，永远不要运行可能不完整或不正确的程序文件，这可能对硬盘造成损害。

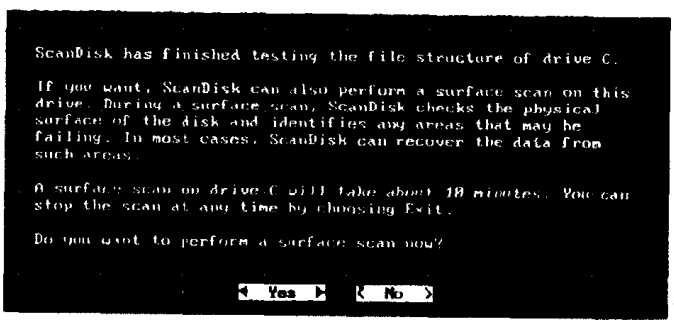

图 27.4　SCANDISK 的表面扫描

不要等到来了出错信息才扫描硬盘表面，经常运行 SCANDISK，在导致数据丢失前，就把问题解决掉。对于一般的使用状况来说，应每月检查一次硬盘表面。

27.6　自 测 练 习

本章讲述了如何使用 CHKDSK 和 SCANDISK 检验磁盘，纠正某些常见的错误。以下练习有助于你进一步熟悉 CHKDSK 和 SCANDISK：

1. 用 CHKDSK 检查硬盘。挽回丢失的分配单元，修复所有的交缠文件。找出硬盘分配单元的大小。（如果报告中有其他的错误，在试图纠正以前参阅第 47 章。）
2. 如果未使用第三方压缩软件，用 SCANDISK 扫描硬盘。包括进行表面扫描并纠正报告的错误。

第28章 系统优化

简介： DOS 6提供了一系列的工具用于改善系统的总体操作性能。

28.1 DOS的功能

学会使用操作系统，而不仅仅只是依赖于各种应用程序的好处之一在于可采取步骤使系统运行得更快，更有效率。DOS 6可用于充分利用内存从而拥有更多加载、运行程序的空间，可使硬盘的容量几乎扩大两倍，可提高磁盘驱动器的运行速度，从而加快整个系统的速度。

本章综述了各种用于优化系统的DOS工具，各个工具的细节则留到随后的几章讲解。

28.1.1 更有效的内存管理

DOS系统中的内存空间总是显得拥挤，即使有着数以兆字节的扩展或扩充内存。习惯上程序必须在常规内存(conventional memory)中运行，不管系统安装了多少内存，它的大小总是限制在640KB。在加载了DOS、一些驱动程序以及其他一些TSR后，常规内存中所剩的空间就不足以运行一些大的应用程序，如桌面印刷系统。对于286计算机，DOS 6提供了部分解决常规内存拥挤的方法，对于386及以上机，情形就大大不同了。

最早的PC限于1MB的内存，除非添加一扩充内存板。286及以上机可将内存扩展到几兆字节。对于拥有扩展内存的系统，DOS6提供了名为HIMEM的驱动程序用于访问和控制扩展内存区。甚至可将DOS加载到扩展内存中，从而省下45KB或更多的常规内存用于其他程序。第29章“内存管理”讲述了各种类型的内存——常规、扩展及扩充，并描述了如何加载和使用HIMEM。

对于386及以上机，DOS 6提供了更多释放常规内存空间的方法。EMM386驱动程序用于在高端内存(upper memory)中加载和运行程序。高端内存是指在常规和扩展内存之间的区域。要是有一些程序需要用扩充内存，EMM386还将用扩展内存仿真扩充内存。DOS 6甚至包括一个名为MEMMAKER的程序，用来设置系统使用HIMEM和EMM386，这样用户就不必自己做这件事了。第30章“为386、486和奔腾机增加的内存工具”讲述了如何使用EMM386和MEMMAKER。

注释： 如果使用Windows，那么会发现第30章所讲述的技术同样能改善Windows的操作水准。

28.1.1.1 硬驱缓存

相对于内存和微处理器来说，硬盘的操作要慢得多，系统不得不等待数据在其中读写。缓存(caching)程序通过将磁盘数据存放在内存中缓解这一问题，从而使DOS能对内存而不是硬盘进行读写。

DOS 6 中的SmartDrive 缓存程序用于在扩展内存中创建一缓存区。有了 SmartDrive，应用程序的运行速度将得到显著提高。SmartDrive 将操作数据存放在缓存中，下一次使用相同的数据时，SmartDrive 就从缓存而非硬驱中读取。而从缓冲中读取数据要比从硬盘中快得多。第 31 章“驱动器缓存”讲述如何加载和使用 SmartDrive。

注释： *对于使用 Windows 的用户，第 31 章讲述了平衡 SmartDrive 和 Windows 的内存需要的技术。*

28.1.1.2 RAM 驱动器的使用

减少磁盘访问的另一途径是创建 RAM 驱动器，除了位于内存中而不是磁盘中，它很像一个磁盘驱动器，它的速度却要快上 100 倍。可以像使用其他驱动器一样使用 RAM 驱动器，但是一旦重新启动或关机，就会丢失其中的数据。因此必须在启动后将数据加载到其上，并在重启动或关机前将重要数据转移到一真实驱动器。由于停电和死机的可能性总是存在的，大多数人只在 RAM 驱动器中存放临时文件，而将重要的数据文件存放在更安全的地方。

DOS 6 的 RAMDRIVE 驱动程序可用于创建 RAM 驱动器。用户可以自己控制每个 RAM 驱动器的大小及位于常规、扩充还是扩展内存。DOS 为每个 RAM 驱动器分配一个驱动器名，就像对真实驱动器那样。第 32 章“RAM 驱动器”讲述了如何创建和使用 RAM 驱动器。

28.1.2 去除文件碎片

DOS 经常被迫使文件碎片化以适合盘上的可用空间。碎片化并不损坏文件，但使驱动器的读取时间加长，因为它不得不更频繁地移动读写头。如果硬盘上的文件碎片化较严重，则可以明显地感觉到程序的运行速度放慢了。

DOS 6 的 DEFRAG 程序可用于去除文件的碎片。DEFRAG 通过移动簇的位置将文件碎片收拾在一起，同时适当调整 FAT 和目录结构。大约每月进行一次去除硬盘碎片化的操作是保持系统处于最佳状态的步骤之一。第 33 章“去除文件碎片”讲述了使用 DEFRAG 的方法。

注释： *提高 Windows 操作性能的方法之一是建立 Permanent Swap File(永久交换文件)。DEFRAG 可成为创建这一文件的至关重要的工具，第 33 章将讲述这一点。*

28.1.3 数据压缩

不管硬盘有多大，总有装满的时候。压缩驱动器可释放大量空间以存放更多的文件。数据压缩将文件中重复的数据予以压缩从而使之占据较少的空间。比如，假定单词“the”在某一文件中多次出现，就可以用一代码替换除第一次出现外的所有“the”，并使此代码指向第一个词即可。

文件压缩的程度取决于它所包含的重复数据的数量。程序文件包括的重复数据通常较少，因此程序文件的压缩程度相当有限。文本文件则包含了大量重复信息。一些常用字符串如“the”、“and”、“are”等可能出现上百次。文本文件通常可压缩到少于原大小的一半。完全

由黑、白点组成的黑白位图文件最具可压缩性。

DOS 6.0 和 DOS 6.2 的早期版本提供了名为 DblSpace 的磁盘压缩程序。DOS 6.22 用 DrvSpace 代替了 DblSpace。第 34 章“DriveSpace 和 DoubleSpace”讲述了如何使用 DblSpace 和 DrvSpace。

注释： 改善 Windows 操作状况的方法之一是为它提供 Permanent Swap File，但 Permanent Swap File 却是不可压缩的。第 33 章讲述了如何在一压缩驱动器上为 Permanent Swap File 留出空间。

第29章 内存管理

简介： 即使拥有4MB或更多RAM的PC机，运行中可很容易出现内存不够的情况。DOS 6包括的几个工具可帮助用户最大限度地使用内存。

29.1 DOS的内存之争

如果拥有4MB、8MB或更多的内存，怎么还会用完内存空间呢？答案在DOS的基本结构中，它仅用最开始的640KB RAM(即常规内存)来加载和运行应用程序，不管拥有多少RAM都是如此。如果让设备驱动程序和TSR(比如PRINT和DOSKEY)驻留在常规内存中，则一旦想加载流行的字处理和电子报表程序，很容易出现Out of Memory信息。

出色的内存管理的目标就是从常规内存中移去尽量多的东西，从而最大限度地增加可用于应用程序的空间。DOS提供了多个工具做这件事。本章讲述了可用于任何系统的工具。第30章"为386、486和奔腾机增加的内存工具"讲述了为386及以上机型添加的工具。

本章将讲述如何：

- 识别不同类型的内存
- 用MEM命令显示内存布局
- 加载DOS的扩展内存管理程序HIMEM.SYS
- 将DOS从常规内存中移出

还将学习以下术语：

- 常规内存
- 高端内存区(HMA：high memory area)
- 扩充内存
- 扩展内存
- RAM
- ROM
- 高端内存

注释： Windows使用内存的方式与DOS不同。如果拥有4MB以上RAM，出现Windows Out of Memory信息则可能是堆栈空间不移。退出，重启动，重新运行Windows通常可以解决这一问题。

29.2 内存的概念

内存是计算机的存储设备，程序运行时即驻留在此。内存的基本部分是在母板上靠近微

处理器的一系列条状物，这样数据就能在两者间快速传送。内存为全固态结果，不像磁盘，它没有可移动部分阻碍运行速度。

不可能在磁盘或其他存储设备上运行程序，运行程序必须载入内存。当键入一命令，如WORD(启动 Microsoft Word for DOS)，DOS 将 Word 程序载入常规内存。当加载一个内存驻留程序，如 TSR 或驱动程序，它就被保存在内存中直到特意移去它或重新启动。如果在常规内存中加载了足够的内存驻留程序，想象运行其他程序，如 Word 时，就无法加载它。

许多人把内存和磁盘空间混为一谈。磁盘驱动器为程序和数据文件提供了长期存放的手段，内存则是一短期存储区域，DOS 在其中存储运行的程序。尽管程序必须加载入内存，其处理的数据却可存放在一稍慢的设备中，比如磁盘。尽管如此，如果一程序能将其数据也存入内存，就可加快运行速度，因为不需等待数据在较慢设备中读写。

29.2.1 内存种类

最早的个人计算机拥有 16KB 的内存，用户可以以 16KB 为增量进行扩充，直到最大值 64KB。无论是程序所能完成的任务，还是它们与用户交流的方式都大大受到了限制。彩色监视器、图形用户界面、鼠标在那时是不可想象的。那些早期程序运行非常之慢，并且将大部分数据存放在外部存储设备中。

到了七十年代晚期，Intel 公司开发的 8088 微处理器芯片可访问 1MB 内存，这在当时是开创性的。IBM 公司在最早的 IBM PC 中采用了 8088。从此个人机迅速兴起，都是以 8088 和 1MB 内存为基础。

IBM 和一家新兴小公司 Microsoft 为这些新机器开发了一种名为 Disk Operating System (DOS：磁盘操作系统)的操作系统。Inter、IBM 和 Microsoft 决定把 1MB 的内存区分为两部分，如图 29.1 所示。最开始的 640KB 称为常规内存，用于运行 DOS 及其应用程序。这一区域是以前个人机内存的十倍。余下的 384KB 称为高端内存，用于 DOS 访问硬件设备，如监视器。

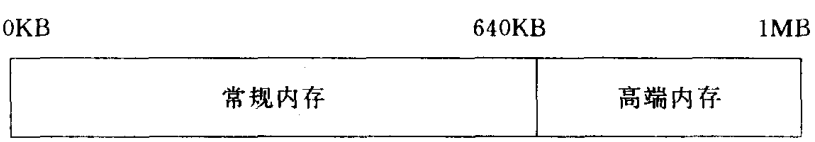

图 29.1 常规内存和高端内存

有两种内存设备，RAM 和 ROM。RAM 即随机访问内存(random-access memory)，可读取和写入。可一遍一遍地向它写入数据，新数据自然取代老数据。尽管如此 RAM 也有其弱点，一旦断电，其数据就丢失了。

ROM 即只读内存(read-only memory)，其存储的数据由制造者写入，可重复读取，但不能再次写入。ROM 芯片通常包含一基本程序，如基本输入输出操作系统(BIOS：basic input-output operating system)，它提供了读写硬件数据的基本指令。即使关闭电源，ROM 中的信息仍得以保护。因此在启动的早期，其他程序还来不及加载到 RAM 中，ROM 中的信息已准备好了。

常规内存总是由 RAM 组成的，因为它必须可重复加载程序和数据。高端内存可包括一

部分 RAM，一部分 ROM，取决于系统的设备。BIOS 芯片，一般指 ROM-BIOS，位于高端内存中。高端内存空了不少部分，用于安装其他设备，如声象卡、传真/调制解调器等。

一旦有 640KB 的内存，人们就想法尽量利用它。程序变得越来越大，越来越快，也越来越奇妙。内存驻留程序——设备驱动程序和 TSR，如 ANSI.SYS 和 PRINT 开始出现了。当加载了内存驻留程序后，它就待在内存中了，加载的其他程序只能进入剩下的空间。不到几年，640KB 的内存又显得太小了。

29.2.2 扩充内存

最早解决内存拥挤问题的方案是采用扩充内存(见图 29.2)。这是位于分立板上的一系列 RAM 条，与其他硬件设备地位相仿。如图所示，扩充内存是通过高端内存进行访问的。注意每次只能访问其中的一部分，不过可以在各部分间切换利用整个空间。被称为扩充内存管理器(expanded memory manager)的设备驱动程序控制对扩充内存的访问和切换。扩充内存比常规或高端内存要慢一点，但比磁盘驱动器还是要快得多。

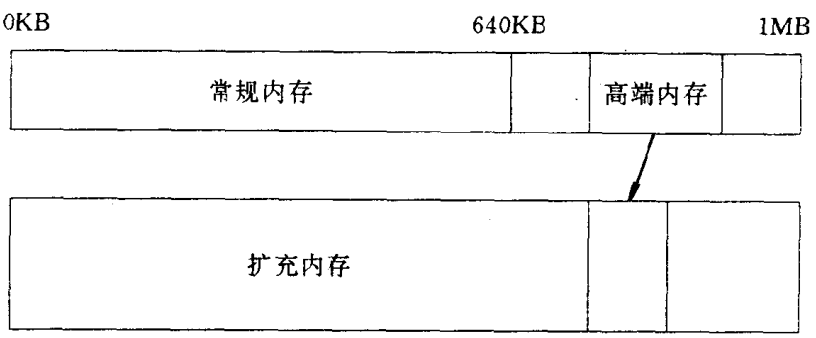

图 29.2 扩充内存

即使系统拥有扩充内存，也不一定用得上。程序必须专门加以设计才能利用扩充内存。如果没有这样的程序，扩充内存就无可用之处。

因为扩充内存是一个独立的设备，DOS 不能直接访问，所以不能直接在那里装载运行程序，但用户可以把数据文件保存在那里。例如，需要 600KB 常规内存的字处理程序可以把它正在处理的文档存放在扩充内存中。

29.2.3 扩展内存

除了扩充内存外，Intel 还开发了另一种解决 640KB 限制的方法。286 芯片除了更快更好外，还可直接访问 16MB 内存。超过 1MB 的内存全都是 RAM，称作扩展内存。386 和 486 将这一限制推得更远，可有几 GB 的内存(1GB 大于 10 亿字节)。从理论上说，可在此内存空间加载运行程序，但令人失望的是 DOS 必须坚持原先的设计方案，无法直接访问额外的空间。DOS 仍然得受 640KB 内存的限制，除了一些有意思的情况例外，本书稍后将详细解释。这就是即使有几兆字节的内存仍会不够的原因。

注释： 一些新设置的操作系统可充分利用整个扩展内存，包括 OS/2、Windows NT 和 Windows 95。在不久的将来，会有更多的此类操作系统出现。最终 DOS 可能会被这些更先进的操作系统所取代。

当然这并不是说扩展内存就没什么用处。DOS 可将扩展内存作为独立的设备，并通过高端内存访问它，其细节由扩展内存管理器(extended memory manager)处理。DOS 不能在扩展内存中加载和运行程序，但有一个例外——高端内存区，这将在下一节中解释。

386 及以上个人机至少拥有 1MB 的 RAM。最开始的 640KB 为常规内存，接下来的 384KB 为扩展内存(高端内存不使用存储器芯片中的 RAM，它只访问硬件卡中的内存)。因此除非是在 286 或更老的机器上，扩展内存的数量总是多于(或等于)384KB。

29.2.4 高端内存区

由于 DOS 对内存寻址的特点，使得它可直接访问的内存并不恰好是 1MB，而是要大约 64KB。当 PC 机拥有扩展内存时，其中的一小部分(大约 64KB)可由 DOS 直接寻址，因而可在其中运行程序。DOS 的名为 HIMEM.SYS 的扩展内存管理器将这部分扩展内存当作高端内存区(HMA：high memory area)，每次可在其中加载运行一个程序。一般情况下，将 DOS 加载入 HMA，释放 45KB 或更多的常规内存用于其他程序。

注释： *扩展内存、HIMEM.SYS 和 HMA 只能用于 286 及以上 PC 机。最早的 PC 和 XT 机可用扩充内存板，但不是扩展内存。*

29.3 浏览内存布局

可使用 MEM 命令浏览所拥有的内存类型及使用方式。要观看 PC 机的内存布局，只需键入不带任何参数的 MEM。

将看到以下信息：

Memory Type	Total=	Used +	Free
Conventional	640K	112K	528K
Upper	0K	0K	0K
Reserved	384K	384K	0K
Extended (XMS)	3,072K	3072K	0K
Total memory	4,096K	3568K	528K
Total under 1 MB	640k	112K	528K
Largest executable program size	527K	(540,096 bytes)	
Largest free upper memory block		(0 bytes)	

上例中系统有 640KB 的常规内存，其中 528KB 未被占用。它有 384KB 的高端内存，由于它被用于最原始的目的——访问与硬件匹配器相联系的 RAM 和 ROM，它出现在 Reserved 的旁边。因为高端内存不能用于程序运行，它未算在 Total under 1MB 中。下章将学习如何使高端内存用于程序运行。

系统有 3MB 的扩展内存，全被占用了，无扩充内存。全加在一起，此系统共有 4MB 内存，其中 528KB 未被占用。可执行程序最大为 527KB，这正是最大可使用常规内存(看起来还有 1KB 用于别处了)。报告最后一行提到的 upper memory blocks(高端内存块)将在第 30 章解释。

不加任何选项的 MEM 可给出关于可用内存量的总体状况，但不知道到底加载了什么程序。在 MEM 命令中加上/C(classify:分类)选项可列出有关程序的详细情况。如果监视器每屏只能显示 24 行，还可加上/P 选项使报告分页。试一下这些操作：

1. 键入 MEM /C /P。

   MEM 显示报告的第一页，当前载入内存的程序。

2. 若最后一行提示 Press any key to continue，按一键阅读报告的余下部分。

报告以当前加载入内存的程序表为开始。具体的显示可能与下例的有所不同，取决于当前使用的驱动程序、TSR 及其他程序。

```
Modules using memory below 1 MB:
Name           Total  =     Conventional  +   Upper Memory
-------  ----------------  ----------------  ----------------
MSDOS      61,037  (60K)     61,037  (60K)      0  (0K)
SETVER        784   (1K)        784   (1K)      0  (0K)
ANSI        4,208   (4K)      4,208   (4K)      0  (0K)
COMMAND     5,760   (6K)      5,760   (6K)      0  (0K)
MOUSE      13,616  (13K)     13,616  (13K)      0  (0K)
GRAB       23,760  (23K)     23,760  (23K)      0  (0K)
GRAPHICS    5,872   (6K)      5,872   (6K)      0  (0K)
Free      540,208 (528K)    540,208 (528K)      0  (0K)
```

从上表中可见加载内存的有 MSDOS、SETVER、ANSI 等。MSDOS 占用了 60KB，SETVER 占用了 1KB 等等。还有 520KB 的常规内存可用于运行其他程序。

注释： MEM 报告内存中的程序不包括其自身，因为一旦结束报告显示，它也就中止了。当你见到报告时，MEM 已退出了内存，这块空间可用于其他程序。

29.3.1 加载 HIMEM

由于 DOS 不能直接访问扩展内存，要使扩展内存为 DOS 所用，必须有一管理器。DOS 包括一名为 HIMEM 的扩展内存管理器。如果系统有扩展内存，就可用 HIMEM 进行访问和控制。

注释： 也可用第三方扩展内存管理器，如 QEMM 或 386MAX。这些产品比 HIMEM 功能更多。如果已使用了第三方管理器，跳过本章余下部分。如果还不确定，看一下刚才显示的 MEM 报告是否有 QEMM 或 386 MAX，或者问一下安装系统的人。

HIMEM 不仅能创建和控制 HMA，它还能管理其余的扩展内存。从而使其中运行的程序能在 DOS 下存储和使用数据，而不互相冲突。可在 CONFIG.SYS 中用以下命令加载 HIMEM：

```
DEVICE=C:\DOS\HIMEM.SYS
```

注释： 必须确保加上正确的 DOS 目录路径，当 DOS 执行 CONFIG.SYS 时，还没有搜寻路径，因此必须告诉 DOS 在哪儿可以找到想要的文件。

大多数人把 HIMEM.SYS 放在 CONFIG.SYS 的最前面，这样 HIMEM 可立即发挥作用，使扩展内存可用于 CONFIG.SYS 加载的其他程序。

29.3.2 在 HMA 中加载 DOS

HIMEM 允许有一个程序使用 HMA。如果载入 HMA 的是一个 10KB 的小程序，则未充分释放常规内存。保证使用 HMA 的程序大小合适的方法之一是加载 DOS，可用如下的 CONFIG.SYS 命令：

```
DOS=HIGH
```

将这个命令紧跟在 HIMEM 后成为 CONFIG.SYS 中的第二个命令，以保证 DOS 在其他程序以前进入 HMA。节省的常规内存的多少取决于 DOS 配置的方式。最多可省下 64KB。还有一小部分 DOS 可待在常规内存中。

如果系统有扩展内存并使用第三方管理器，它还使用 HIMEM 吗？DOS 还在 HMA 中吗？可在 MEM /C 的报告中寻找答案：

- 在内存报告中寻找称为 Extended (XMS)的一行。如果没有这么一行，则意味着没有扩展内存，可跳过本章余下的部分(及第 30 章)。
- 在加载程序表中，寻找 HIMEM。如果没有，则意味着没有使用 HIMEM。下一个练习将学习如何加载它。
- 报告的最后一行是否说 MS-DOS is resident in the high memory area (MS-DOS 驻留在高端内存区)？如果是这样，就表示系统已在 HMA 中加载。如果不是，下一个练习将学习如何加载。

如果拥有扩展内存但却没使用 HIMEM 或在 HMA 加载 DOS，完成以下练习，学习如何加载它们。

警告： 如果你的系统使用的是第三方扩展内存管理器如 QEMM 或 386MAX，不要做这个练习。

1. 切换到引导驱动器的根目录。
2. 做一个名为 CONFIG.SAV 的 CONFIG.SYS 保险拷贝。
3. 用 ASCII 文本编辑器如 DOS 的 EDIT 打开 CONFIG.SYS 以备编辑。
4. 如果其中没有为 HIMEM.SYS 设置的 DEVICE 命令，在文件的第一行插入下面一行：

   ```
   DEVICE=C:\DOS\HIMEM.SYS
   ```

5. 如果 DOS 没有加载到 HMA 中，紧接着 HIMEM 命令插入以下命令：

   ```
   DOS=HIGH
   ```

6. 存储文件，退出编辑器。
7. 重新启动，使 DOS 执行新的 CONFIG.SYS。

   如果 DOS 无法启动，见下面的答疑。
8. 如果 DOS 顺利启动，运行修改后的 CONFIG.SYS，键入 MEM /C /P 浏览新内存布局。

可以看到HIMEM加载到了常规内存中。注意由于DOS被加载到HMA，有了更多可用的常规内存。同时可以看到扩展内存在使用上的区别。有了HIMEM，大多数扩展内存都变成可利用的了。有64KB的HMA被占用了，报告的最后一行显示现在DOS位于HMA中。

答疑： 我在CONFIG.SYS中加了一个命令以加载HIMEM，现在无法启动。出了什么问题，如何解决。

这显然是因为HIMEM的缺省设置不适合于你的系统。HIMEM包括许多参数以使之适合特定的硬件配置（在附录A"设备驱动程序"中详细叙述）。最为可能的是应使用/MACHINE参数，如果这也不行，试一下其他参数（试验参数不会造成任何损害）。如果你自己不知道该使用哪些参数，你的硬件销售商或制造商，或Microsoft的Product Support Team也许会帮助你。

为了正确启动，先按Ctrl+Alt+Delete，然后在见到显示信息Loading MS-DOS...时按F8。跳过加载HIMEM.SYS的命令，执行CONFIG.SYS及AUTOEXEC.BAT中的所有其他命令。而后可以试着按上面的建议对CONFIG.SYS进行修改。别忘了改动CONFIG.SYS后必须重启动以使之生效。

答疑： 我试着运行一个程序，却得到信息：Packed file corrupt。我的程序出了什么问题了。

你的程序没有问题。当你在HMA中加载DOS时，就释放了最前面的64KB常规内存用于其他程序。某些程序加载到这块低内存区时，就不能运行。解决问题的方法是用LOADFIX命令将程序加载到64KB线以上。只需在开始程序的命令前加上LOADFIX，如：

```
LOADFIX AUDIT README.TXT
```

29.4 自测练习

本章介绍了有关PC机内存的众多术语，确保理解了以下每个术语：

- 常规内存(conventional memory)
- 高端内存区(high memory area)
- 扩充内存(expanded memory)
- 扩展内存(extented memory)
- RAM
- ROM
- 高端内存(upper memory)

第 30 章 为 386、486 和奔腾机增加的内存工具

简介： 对于 386 及以上机而言，可用 DOS 6 中的工具将大多数驱动程序和 TSR 从常规内存中移出去。

30.1 DOS 和 386、486、奔腾机

DOS 6 包括多个工具用于 386 及以上 PC 机的内存优化。比如要是系统没有扩充内存，而某个程序却要求有一些，就可将部分扩展内存作为扩充来使用。还可在高端内存的未被占用部分加载、运行程序，以释放更多的常规内存。DOS 甚至提供了一个自动设置 CONFIG.SYS 和 AUTOEXEC.BAT 以充分利用高端内存的程序。

本章将讲述如何充分利用这些工作，将学习：

- 用扩展内存仿真扩充内存
- 将设备驱动程序和 TSR 加载到高端内存而非常规内存中
- 在必要时使用 MEMMAKER 优化高端内存

注释： 如果你有一用途更广的内存管理器，如 QEMM 或 386MAX，则不再需要本章讨论的这些功能。跳过本章，参阅你的内存管理器手册以学会充分利用 PC 内存。

30.2 提供扩充内存

由于几乎所有的新型计算机拥有扩展内存，只有很少一部分有扩充内存，最新的应用程序很少有需要扩充内存的。但如果有一些老的应用程序，如旧版的 Ventura Publisher，就需要一些扩充内存。如果不能加载某应用程序，特别是在硬件升级前能顺利使用的情形下，应检查一下它的文档，看它是否需要扩充内存。

DOS 包括的名为 EMM386 的驱动程序可用于向只有扩展内存的系统提供扩充内存。在不加任何参数和选项的情况下加载 EMM386，它将 256KB 的扩展内存用作扩充内存。如果有足够多的扩展内存，程序可临时要求并获得更多的扩充 。程序使用扩充内存完毕后，除了基本的 256KB，其余内存又转换回扩展内存。

可在 CONFIG.SYS 用以下命令加载 EMM386：

```
DEVICE=C:\DOS\EMM386.EXE
```

EMM386 要求有 HIMEM，因此这条命令必须跟在加载 HIMEM 的命令之后。如果把 EMM386.EXE 命令紧跟在 DOS=HIGH 命令之后（或者紧跟在 HIMEM.SYS 命令之后，如果没使用 DOS=HIGH 的话），这样 EMM386 就能在其他程序之前先占上 256KB 的扩展

内存。

如果想进行将部分扩展内存转换为扩充内存的练习，按以下步骤操作：

警告： 如果使用的是第三方扩展内存管理器，如 QEMM 或 386MAX，不要尝试这个练习。

1. 切换到引导驱动器的根目录。
2. 做一名为 CONFIG.SAV 的 CONFIG.SYS 保险拷贝。
3. 用 ASCII 文本编辑器如 DOS 的 EDIT 打开 CONFIG.SYS 以备编辑。
4. 在加载 HIMEM 的那行插入下面这行：

   ```
   DEVICE=C:\DOS\EMM386.EXE
   ```

5. 重启动。

   如果 DOS 因某种原因无法启动，参见第 29 章“内存管理”中的“在 HMA 中加载 DOS”一节中的答疑部分。

6. 键入 MEM。

   MEM 报告应显示现在有了 256KB 扩充内存(少了 256KB 扩展内存)。同时注意 EMM386 占用了一些常规内存和保留内存。

7. 要去除扩充内存，可用 CONFIG.SAV 代替 CONFIG.SYS 并重启动。

30.2.1　使用高端内存块

第 29 章讲到被称为高端内存的从 640KB 到 1MB 的内存为 DOS 保留作硬件访问之用。但只要很少的系统会真正使用整个 384 KB 内存。未使用的空间是宝贵的，因为不像扩充内存和扩展内存，DOS 可在其中运行程序。EMM386 的第二项功能是在未使用的高端内存空间加载运行程序。大多数人把所有的设备驱动程序和 TSR 加载到高端内存中，以留出多于 600KB 的常规内存运行应用程序。

图 30.1 显示了系统高端内存的布局实例。设备内存和 ROM-BIOS 并不一定要占据连续的位置。未使用的空间可分为块称为高端内存块(UMB:upper memory block)。

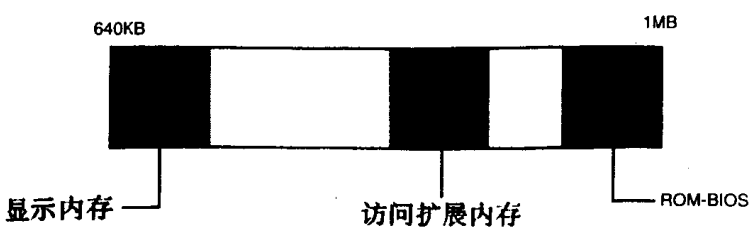

图 30.1　典型的高端内存配置

高端内存中的空白块无与之相联系的 RAM，它们只是一些保留空间。EMM386 从扩展内存中得到足够的 RAM 以支持高端内存加载所需程序。如果因为不存在 UMB 或它不能得到足够的扩展内存，从而使 UMB 对于某程序不够大，DOS 将程序加载到常规内存中，不发出任何警告。在这种情形下，就不能通过将程序加载到高端内存释放常规内存空间。

对于 386 及以上 PC 机来说，也许早就使用了高端内存。MEM 报告显示：

1. 键入 MEM /C /P。

MEM 显示内存使用报告。报告的开始部分显示哪些程序加载到了常规内存中，哪些加载到高端内存中：

模块使用的内存不超过 1MB：

Name	Total =	Conventional +	Upper Memory
MSDOS	17,741 (17K)	17,741 (17K)	0 (0K)
HIMEM	1,168 (1K)	1,168 (1K)	0 (0K)
EMM386	3,120 (3K)	3,120 (3K)	0 (0K)
COMMAND	2,928 (3K)	2,928 (3K)	0 (0K)
SETVER	640 (1K)	0 (0K)	640 (1K)
MTMCDS	10,320 (10K)	0 (0K)	10,320 (10K)
SHARE	17,904 (17K)	0 (0K)	17,904 (17K)
MSCDEX	27,952 (27K)	0 (0K)	27,952 (27K)
SMARTDRV	28,832 (28K)	0 (0K)	28,832 (28K)
Free	630,403 (615K)	630,403 (615K)	0 (0K)

在本例中，SETVER、MTMCDS、SHARE、MSCDEX 和 SMARTDRV 占据了高端内存，其他程序则在常规内存之中。

某些程序不能加载到高端内存中，包括 MS-DOS 中的一小部分，如 HIMEM，EMM386 和 COMMAND 等。如果你的系统将几个其他程序加载到了常规内存中，或根本未使用高端内端，MEMMAKER 可帮助你充分利用高端内存。MEMMAKER 的用法将在下节讲述。

注释： 即使你的系统内存目前布局非常合理，也请阅读下一节（不过不用做练习）。往系统添加了驱动程序或 TSR 后，就需要运行 MEMMAKER 了。

30.3 MEMMAKER 的使用

设置和使用高端内存块需要在 CONFIG. SYS、AUTOEXEC. BAT 和 SYSTEM. INI（若使用 Windows）中添加许多命令。DOS 6 的 MEMMAKER 可完成这一工作。MEMMAKER 分析系统状况，指出使用 UMB 的最佳方案，并在 CONFIG. SYS、AUTOEXEC. BAT 和 SYSTEM. INI 中插入必要的命令。

MEMMAKER 将所有的驱动程序和 TSR 命令归入启动文件以将相应的程序加载到高端内存中。在运行 MEMMAKER 前，编辑 CONFIG. SYS 和 AUTOEXEC. BAT，插入所有的所需加载的驱动程序和 TSR 命令提供给 MEMAKER（仅在了解 SYSTEM. INI 文件的工作方式的情况下，才可以改动它）。当然不用急着加载所有的驱动程序和 TSR，以后要想加载更多的驱动程序和 TSR，还可运行 MEMMAKER。

MEMMAKER 包括取消其操作的选项，因此放心大胆地去试吧。如果不喜欢最终的结果，取消它们即可。

1. 退出任何程序和外壳，回到初始命令提示符。
2. 在 DOS 命令提示符下键入 MEMMAKER。

MEMMAKER 显示 Welcome 屏。

3. 读完 Welcome 屏后，按 Enter 继续。

   MEMMAKER 提供常规和快速设置选择。

4. 按 Enter 选择 Express(快速)。

注释： 用常规设置可在高端内存中挤入更多的程序，不过大多数用户不必这么精打细算。

   MEMMAKER 询问系统是否需要扩充内存。

5. 按 Enter 回答 No(如果肯定程序需用扩充内存，按空格键将 No 换成 Yes，然后按 Enter 回答 Yes)。

   MEMMAKER 开始进行内存布局，在操作的同时，随时通知用户所发生的事情。为了试验最佳设置方式，它需多次重启动系统，不过每次启动前，都会发出警告。

答疑： 我的系统在运行 MEMMAKER 中死机了，该怎么办？

某些系统在 MEMMAKER 试图重新启动时死机，比如在 MEMMAKER 尝试的技术不适合系统时。一旦出现这种情况，只好关闭电源，然后打开重启动。系统重启动后，MEMMAKER 继续控制系统，MEMMAKER 将建议用户选择选项 Try again with conservative settings(用保守的设置再进行尝试)。这时应选择这个选项，除非愿意冒再次死机的风险尝试更多的新选项。

   当 MEMMAKER 准备好用系统的新设置进行最后的重启动时，它提示注意启动过程中的出错信息。

6. 仔细阅读启动信息。可在过程中按 Pause 以得到更多的阅读时间(在暂停后按任一键继续)。

   最后一次重启动后，可看一下 MEMMAKER 是否发挥了作用。

7. 若回答 Yes，MEMMAKER 显示一表比较内存的旧用法和新用法。从中可获知到底节省了多少常规内存。

8. 按 Enter 接受新设置并退出 MEMMAKER。也可按 Esc 取消 MEMAKER 的操作，恢复原来的系统设置。

9. 键入 MEM /C /P 看一下加载到高端内存的程序。

   下面列出的是一种典型的情形：

Modules using memory below 1 MB:

Name	Total =	Conventional	+ Upper Memory
MSDOS	16,621 (16K)	16,621 (16K)	0 (0K)
HIMEM	1,152 (1K)	1,152 (1K)	0 (0K)
EMM386	3,120 (3K)	3,120 (3K)	0 (0K)
COMMAND	3,680 (4K)	3,680 (4K)	0 (0K)
SETVER	816 (1K)	0 (0K)	816 (1K)
ANSI	4,240 (4K)	0 (0K)	4,240 (4K)
MOUSE	13,616 (13K)	0 (0K)	13,616 (13K)
GRAB	23,744 (23K)	0 (0K)	23,744 (23K)
GRAPHICS	5,872 (6K)	0 (0K)	5,872 (6K)
SMARTDRV	27,264 (27K)	0 (0K)	27,264 (27K)
Free	713,952 (697K)	630,656 (616K)	83,296 (81K)

```
Memory Summary:
Type of Memory          Total     =     Used      +     Free
----------          ----------      ----------      ----------
Conventional      655,360  (640K)    24,704  (24K)   630,656  (616K)
Upper             158,848  (155K)    75,552  (74K)    83,296  (81K)
Reserved          234,366  (2,29K)  234,368  (229K)        0  (0K)
Extended (XMS)  3,145,728 (3,072K) 1,429,504 (1,396K) 1,716,224 (1,676K)
------              ----------      ----------      ---------
Total Memory    4,194,304 (4,096K) 1,764,128 (1,723K) 2,430,176 (2,373K)
Total under 1 MB  814,208  (795K)   100,256  (98K)    713,952  (697K)
Largest executable program size                      630,560  (616K)
Largest free upper memory block                       67,952  (66K)
MS-DOS is resident in the high memory area.
```

将这些数据与第 29 章的比较，它们来自同一系统，只不过是运行 MEMMAKER 前的。可使用的常规内存跳到了 616KB。当前在常规内存中的程序——一小部分 DOS、HIMEM、EMM386 和 COMMAND，就是无法在高端内存中运行的。其他驱动程序和 TSR 都加载到了高端内存，还剩下 81KB 的空间。

预览： 如果需要比 MEMMAKER 的快速设置创建的更多的高端内存空间，参见第 44 章“腾出更多的内存空间”，其中讲述了如何进行设置。

30.4 MEMMAKER 所做的改动

下面是一个经过 MEMACKER 修改的 CONFIG.SYS 的典型例子：

```
DEVICE=C:\DOS\HIMEM.SYS
DOS=HIGH
DEVICE=C:\DOS\EMM386.EXE NOEMS
BUFFERS=15,0
FILES=50
DOS=UMB
DEVICEHIGH /L:1,12048=C:\DOS\SETVER.EXE
STACKS=9,256
DEVICEHIGH /L:1,9072=C:\DOS\ANSI.SYS
SHELL=C:\DOS\COMMAND.COM C:\DOS\ /E:1024 /P
```

下一节解释了 MEMMAKER 对这一文件所做的更动。

30.4.1 加载 HIMEM 和 EMM386

MEMMAKER 在 CONFIG.SYS 中插入一些命令以加载 HIMEM 和 EMM386。如果 HIMEM 命令早就存在于文件之中，MEMMAKER 不改动它，只是也许将它移到一个更好的位置。本例中的 EMM386 命令有 NOEMS(no expanded memory：不要扩充内存)选项，从而只要求高端内存支持，而不要扩充内存支持。如果告诉 MEMMAKER 需要扩充内存，则它用 RAM 选项代替 NOEMS 以获取这两种支持。

提示： 如果既不用 RAM 也不用 NOEMS，EMM386 提供扩充内存支持，而不提

供高端内存支持，正如在第 29 章讲述的那样。

预览： EMM386 命令包括一系列的参数和选项以控制诸如扩充内存的最小和最大量等问题。“命令参考”详细描述了这些高级选项。

30.4.2 DOS 命令

注意文件中包含了两条 DOS 命令。DOS=HIGH 命令将 DOS 加载到 HMA 中，如第 29 章所述。MEMMAKER 只处理高端内存，不会将 DOS=HIGH 插入到 CONFIG.SYS 中。你必须在运行 MEMMAKER 之前或之后，自己加上这条命令。

DOS=UMB 命令将高端内存块提供给 DOS。MEMMAKER 将这条命令插入到 CONFIG.SYS 中。也可将这两条命令合并为 DOS=HIGH,UMB。

30.4.3 DEVICEHIGH 命令

MEMMAKER 将大多数 DEVICE 命令转换为 DEVICEHIGH 命令，从而将设备驱动程序加载到高端内存块中。/L 选项告诉 DOS 在高端内存块中加载程序的确切位置，以充分利用空间。

30.5 AUTOEXEC.BAT 文件

运行 MEMMAKER 后，你的 AUTOEXEC.BAT 文件将与下面的类似，尽管包含的命令会有所不同。

```
PATH C:\DOS;C:\WINWORD;C:\NDW;C:\WINDOWS;C:\HSG
CD MOUSE
LH /L:1,20304 MOUSE BON
CD \HSG
LH /L:1,23744 GRAB
CD \
PROMPT $P$G
SET TEMP=C:\WINDOWS\TEMP
LH /L:1,20560 GRAPHICS
```

MEMMAKER 将加载 TSR 的命令几乎全转化成了 LH(Loadhigh：加载到高端内存)命令，从而将 TSR 加载到高端内存。/L 和/S 选项控制程序加载的确切位置。

LH 不仅仅能用于 AUTOEXEC.BAT 文件，任何时候加载 TSR 都可用到它。比如想临时在命令提示符下键入 VSAFE 命令来使用 VSAFE，就可键入 LH VSAFE，从而使 VSAFE 加载到高端内存，而不是常规内存中。可用 LH 将 DOSKEY、PRINT 和任何其他 TRS 加载到高端内存中。

注释： 运行 MEMMAKER 后，可能会得到信息 Packed File Corrupt。不必担忧，没有什么可怕的错误，解决方法非常简单。参见第 29 章名为“在 HMA 中加载 DOS”一节的答疑部分。

注释： /L 选项使高端内存中的程序更为紧凑。MEMMAKER 还可与/L 选项一起加上/S 选项。这些选项较为复杂，使用不当会造成死机。幸运的是，并不需要把

它们搞清楚。如果有足够的空间，LH 命令不用/L 和/S 选项也可把程序加载到高端内存，否则 DOS 忽略 LH 而将程序加载到常规内存中。

如果要把 TSR 加到 AUTOEXEC.BAT 中，把驱动程序加到 CONFIG.SYS 中，可自己插入 LH 和 DEVICEHIGH 命令。如果这些新程序不能顺利地进行高端内存而被加载到了常规内存中(用 MEM /C /P 可看出)，MEMMAKER 可将它们挤进去。可根据需要多次运行 MEMMAKER，改变系统的配置。

30.5.1 取消 MEMMAKER

如果用 MEMMAKER 重新配置系统后，某一程序无法运行，可用下面命令恢复原先的配置：

```
MEMMAKER /UNDO
```

这时，可按照下列步骤利用高端内存：

1. 在 CONFIG.SYS 中插入命令加载 HIMEM 和 EMM386(如果不需要扩充内存，可加上 NOEMS 选项，否则用 RAM 选项)。
2. 插入 DOS=UMB 命令。
3. 将其余 DEVICE 命令换为 DEVICEHIGH 命令(不要尝试用/L 或/S 选项)。
4. 在 AUTOEXEC.BAT 文件中，在加载 TSR 的命令前插入 LH(不要尝试用/L 或/S 选项)。

如果仍然有问题，可逐一改变 DEVICEHIGH 或 LH 命令，将对应的驱动程序或 TSR 加载到常规内存中。每进行一个更改就重新进行启动，看看是不是已解决问题。最后总能找到引起麻烦的程序。也许可向其制造商要一个与高端内存相容的新版本。

30.6 自测练习

如果在学习本章的过程中已优化了你的 PC 机内存，则不必再做一遍。只需确保你已经知道如何运行 MEMMAKER 以及在添加了新驱动程序和 TSR 时，如何使用 LH 和 DEVICEHIGH 命令。

第 31 章　驱动器缓存

简介： 除了硬件升级外，提高系统运行速度的最好办法是进行驱动器缓存。

31.1 有关缓存

磁盘缓存(disk cache)是指为保存 DOS 从磁盘中取写的数据而保留的一内存区。第 21 章“系统配置”中所学的缓冲区(buffer)其实就是一种缓存。DOS 和 Windows 提供的磁盘缓存程序 SmartDrive 可完成比缓冲器更出色的缓存工作。

注释： SmartDrive 要求有扩展内存，如果没有，可跳到第 32 章“RAM 驱动器”。

在内存中移动数据比访问磁盘快得多。磁盘缓冲也许是改善系统操作状况的最佳方案，因为它大大减少了磁盘访问次数。

本章描述了如何使用 Smart Drive，将学习如何：

- 理解 SmartDrive 的用途
- 安装 SmartDrive
- 决定是否需要用 SmartDrive 进行双缓冲
- 指定进行缓存的驱动器及缓存程度
- 迫使 SmartDrive 延迟写入
- 理解 SmartDrive 的显示信息

警告： 如果已经使用了第三方驱动器缓存软件，如 NCache，不要再用 SmartDrive。将磁盘数据缓冲两次会降低系统运行速度并导致数据丢失。如果安装了第三方软件，跳过本章。

31.2 SmartDrive 的用途

SmartDrive 将其置于 DOS 和磁盘设备之间，因而可处理由 DOS 发出的读写要求(如图 31.1 所示)。当 DOS 要从磁盘读取数据时，SmartDrive 加入了进来。如果想要的数据已经在缓冲中，SmartDrive 把数据从缓存中传递给 DOS，省下通常的读盘时间。如果想要的数据不在缓存中，DOS 直接从磁盘读取数据。

图 31.1　SmartDrive、DOS 和驱动器

SmartDrive 将数据拷贝到缓冲中用于将来的读写。如果缓存满了，新数据将取代部分现存数据。使得 SmartDrive 灵活机智的原因之一是它知道如何识别并替换缓冲中最不常见的

数据。

注释： 取代缓冲中的数据并真正造成数据丢失。数据仍存在磁盘上可再次读取。取代缓存数据仅仅意味着下次访问这些数据将花费较多的时间。这也就是 SmartDrive 替换最不常用的数据的原因。

SmartDrive 也可进行写缓存的操作。当 DOS 要将数据写入磁盘时，SmartDrive 插了进来，将数据写入缓存中，需要的话替换缓存中已有的数据。SmartDrive 从缓存中把数据写入磁盘。

SmartDrive 并不一定立即把要写入的数据传送给磁盘。它将数据保留到不再进行键入、打印或磁盘访问等操作时。在这样一种间隔中，SmartDrive 将缓冲数据写入磁盘，首先写入最早的部分。这一延迟写入(delayed writing)方式通过将优先权给予数据读取和处理改善了系统操作性能，在系统不太繁忙时才进行写入操作。

- 当数据已延迟了 5 秒钟。
- 当 SmartDrive 需要缓冲空间来写入新数据。
- 当按 Ctrl+Alt+Del(SmartDrive 在系统重启动前，立即写入所有的延迟数据)。
- 当键入命令 SMARTDRV /C(/C 选项迫使 SmartDrive 写入所有的延迟数据)。

DOS 6.2 还增添了一种写入情况：一旦出现命令提示符时。这一措施保证了一旦程序结束，其数据在其他程序开始前被写入。它同时也减小了在延迟数据写入前关机的可能性。

如果在有延迟数据未写入的情况下，系统断电会出现什么问题呢?将失去所有未来得及写入磁盘的缓存中的数据。比如正在进行文字处理，就有可能失去早已存储的文档的一部分或所作的改动。如果正在更新一数据库或电子报表，就有可能失去部分所作的更动。用户必须自己作出决定：为了加快速度是否值得冒失去数据的风险。

如果正在进行商业交易型应用程序，数据的部分缺失是很难对付的，因为一项新的交易更新了多种记录。比如在一个帐目接收系统中，一项新条目会影响顾客记录、到期发货单以及帐目录记录。如果只有一项记录得到更新，系统就出了问题，那么这些记录将不再同步从而导致麻烦。由于这个原因，作者建议在使用涉及多重更新的交易应用程序时，不要使用写入缓存操作。

同样不应对软驱进行延迟写入，这容易造成在写入延迟数据前换盘的错误，而丢失数据。

31.3 缓存和缓冲区

DOS 也可将读写数据存放在数据缓冲区中。即使有了缓存程序，DOS 仍需将某些信息保存在缓冲区中。由于 DOS 在访问 SmartDrive 前首先所有的缓冲区，因此在一缓存系统中，减少缓冲区数目可明显节省时间。如果使用了 SmartDrive，应改变 CONFIG.SYS 中的 BUFFERS 命令，只使有 10 到 15 个缓冲区。

31.4　启动 SmartDrive

在启动 SmartDrive 前必须加载扩展内存管理器，如 HIMEM. SYS。在可能的情况下，SmartDrive 加载到高端内存(不必指定 LH)。如果启动时已通过 CONFIG. SYS 加载了 HIMEM，现在就可以启动 SmartDrive 了。

注释： SmartDrive 不能加载两次。如果已加载了 SmartDrive，再次键入 SMARTDRV 命令并不造成任何损害，而只是显示启动信息不再加载 SmartDrive。

1. 要启动 SmartDrive，可键入下面命令：

```
SMARTDRV
```

系统的响应取决于具体配置，下面是一种典型的情形：

```
Microsoft SMARTDrive Disk Cache version 5.01
Copyright 1991,1993 Microsoft Corp.

Cache size:1,048,576 bytes
Cache size while running Windows: 262,144 bytes
         Disk Caching Status
drive   read cache   write cache   buffering
-----------------------------------------------
A:      yes          no            no
B:      yes          no            no
C:      yes          yes           no

Write behind data will be committed before command prompt returns.

For help,type "Smartdrv /?".
```

其中的缓存大小为缺省设置，取决于系统中扩展内存的数量。此例中的系统在加载 SmartDrive 时还有 1MB 到 2MB 的扩展内存。在启动 Windows 时，SmartDrive 将放弃一些扩展内存，因为 Windows 必须有 2MB 以上的扩展内存才能使状态最佳。

在缺省情况下，硬驱(包括用 INTERLNK 链接的硬驱)为读写缓存；软驱仅为读缓存；CD-ROM 在 DOS 6.2 和 6.22 中是读缓存的，在更早版本中则不是，它们当然不是写缓存的。其他类型的驱动器，如 RAM、网络及压缩驱动器不能缓存。

注释： 只有在加载 SmartDrive 前先加载了 MSCDEX (DOS 的 CD-ROM 驱动程序)，SmartDrive 才能对 CD-ROM 驱动器进行缓存。如果是从 AUTOEXEC. BAT 加载 SmartDrive 和 CD-ROM 驱动程序(如 MSCDEX)的，必须确保 MSCDEX 命令在前面。

31.4.1　SmartDrive 的双缓冲(double-buffering)功能

在 SMARTDRV 的普通设置下，某些驱动器不能工作得很好。为此，SMARTDRV 还可使用一种称为双缓冲的技术，它尽管使用速度略有降纸，却能使系统免受数据丢失之害。双缓冲就是在常规内存中创建一小缓冲区，并使送往缓冲的数据首先经过引缓冲区。这就防止了某些特定类型的驱动器直接访问缓存而造成的地址混淆。

注释： 不遵照 Microsoft 的 Virtual DMA Service(VDS)标准的总线总裁器(bus-mastering controller)驱动器需要双缓冲。包括某些 SCSI 驱动器、ESDI 驱动器等等。

不要自动安装双缓冲，某些驱动器不能在双缓冲下工作。可从 SMARTDRV 的磁盘缓存状态看出系统是否需要使用双缓冲。只要缓冲列中有一行显示 yes，就应使用双缓冲。如果有一行显示－，则表示 SMARTDRV 无法确定是否需要，为保险起见，不妨装上。可用 CONFIG. SYS 中的如下命令通过加载 SMARTDRV. EXE 驱动程序来安装它：

```
DEVICEHIGH=C:\DOS\SMARTDRV. SYS DOUBLE_BUFFER
```

此时仍需用 SMARTDRV 命令来启动 SMARTDRV，正如前面讲述的那样。

31. 4. 2 选择缓存方式

在 SmartDrive 命令中，可选择每个驱动器的缓存方式来代替缺省设置。在 SmartDrive 命令中添加 drive 参数可打开该驱动器读取缓存，但不影响写入缓存。Drive＋打开该驱动器的读写缓存，Drive－则关闭该驱动器的所有缓存。不能在关闭读取缓存的情况下打开写入缓存。

假定你有两个软驱(A 和 B)以及三个硬驱(C、D 和 E)。如果你想让驱动器 B 和 C 仅为读缓存，A 和 E 为读写缓存，D 无缓存，可用下面的命令启动 SmartDrive：

```
C:\DOS\SMARTDRV A+ B C D - E+
```

系统将作出响应：

```
Microsoft SMARTDrive Disk Cache version 5.01
Copyright 1991,1993 Microsoft Corp.

Cache size: 1,048,576 bytes
Cache size while running Windows: 262,144 bytes

                Disk Caching Status
drive   read cache   write cache   buffering
-----------------------------------------
A:        yes          yes           no
B:        yes          no            no
C:        yes          no            no
D:        no           no            no
E:        yes          yes           no

Write behind data will be committed before command propt returns.
For help,type Smartdrv /?.
```

由于驱动器 B 和 E 为缺省缓存设置，可略去这两个驱动器而用下面的命令：

```
C:\DOS\SMARTDRV A+ C D-
```

31. 4. 3 安装 SmartDrive 后改变驱动器缓存

加载 SmartDrive 后，可用 SmartDrive 命令改变驱动器的缓存类型。比如要关闭驱动器 A 的所有缓存，可用如下命令：

SMARTDRV A－

此命令仅改变驱动器 A 的缓存类型，但不影响其他驱动器。试着关闭引导驱动器的写缓冲及驱动器 A 的所有缓存：

1. 对当前驱动器缓存状态作一记录，必要时进行恢复（提示：可按 PrintScreen 打印 SmartDrive 的启动信息）。
2. 键入 SMARTDRV A－ C

   在结果显示中，Disk Caching Status（磁盘缓存状态）表将如下所示：

```
         Disk Caching Status
drive   read cache   write cache   buffering
-----------------------------------------------
B:        yes          no            no
C:        yes          no            no
```

3. 键入 SMARTDRV 命令恢复先前的磁盘缓存状态。

31.4.4　SmartDrive 结果显示

用带/S（表示 status，状态）选项的 SmartDrive 命令可显示驱动器的缓存状态。此报告将包含比 SmartDrive 启动信息更多的内容。试一下此操作：

1. 键入 SMARTDRV /S。

   系统响应如下：

```
Microsoft SMARTDrive Disk Cache version 5.01
Copyright 1991,1993 Microsoft Corp.

Room for 128 elements of 8,192 bytes each
There have been 5,727 cache hits
       and 3,111 cache misses

Cache size:1,048,576 bytes
Cache size while running Windows:262,144 bytes
         Disk Caching Status
drive   read cache   write cache   buffering
-----------------------------------------------
A:        yes          no            no
B:        yes          no            no
C:        yes          yes           no

For help,type Smartdrv /?.
```

正如所见到的那样，SmartDrive 不仅告知缓存设置，同时给出最近的缓存统计。Hit（击中）表示 SmartDrive 在缓存中找到了数据从而避免了访问磁盘；Miss（漏过）则表示相反。5700 比 3100 的 Hit 和 Miss 比例已经非常好了，这表示 SmartDrive 避免了大量的磁盘访问。有时 Hit 比例可达到 9 比 1 甚至更高，这取决于正在进行的过程。

31.4.5　迫使 SmartDrive 写入缓存数据

/C 选项可用来迫使 SmartDrive 写入所有的缓存数据，如下所示：

1. 键入 SMARTDRV /C。

若 SmartDrive 写入缓存数据，则可见到驱灯闪烁。

如果设置为写缓存且未升级到 DOS 6.2，在停下磁头，系统断电前应先使用此命令。如果将一软驱设置为写缓存，则在换盘前应先使用此命令。

某些时候系统死机而来不及键入 SMARTDRV /C 命令。如果能用 Ctrl+Alt+Del 重启动，则不会有任何麻烦；如果只能用重启动按钮或关闭系统，则会丢失所有的延迟数据。

31.4.6 Windows 和 SmartDrive

DOS 和 Windows 都带有 SMARTDRV.EXE 文件。如果有 Windows 3.1 或更新版本，比较 Windows 目录和 DOS 目录中的 SMARTDRV.EXE 文件的日期/时间标签，使用最新版本的 SmartDrive。

注释： *运行 Windows 时，不能启动 SmartDrive。必须在启动 Windows 前加载* SmartDrive。

31.5 控制缓存大小

SmartDrive 命令包括用于控制缓存大小的参数，如图 31.2 所示。可规定运行 Windows 时的缓存初始大小和最小大小。如果只运行 Windows，可将最小缓存设置为 0 以让其自由控制。如果要在 Windows 和 DOS 间交替运行，则应保留一些缓存空间，这样启动 Windows 时，就不会丢失所有的 DOS 数据。表 31.1 显示了缓存的缺省大小。

表 31.1 缓存的缺省大小

扩展内存	初始缓存	最小缓存(当 Windows 运行时)
最大到 1MB	所有的扩展内存	0(没有缓存)
最大到 2MB	1MB	256KB
最大到 4MB	1MB	512KB
最大到 6MB	2MB	1MB
6MB 或更多	2MB	2MB

缓存越大越好，因此不妨将它设置得尽量大。比如要是有 3MB 的扩展内存不必用于其他用途，就可用如下命令安装 SmartDrive：

```
SMARTDRV 3072 512
```

如果扩展内存放不下指定的缓存，SmartDrive 可调整缓存大小使之与可用空间匹配。

31.5.1 调整 SMARTDRV

参数/E、/L 和/B 可用来平衡 SmartDrive 的效率和它使用的内存数量。这些参数只能在启动 SmartDrive 的命令中使用，一旦已加载了 SmartDrive，就不能改变这些特性了。

如果要使 SmartDrive 使用较少的常规内存，可将/E 设置得小于其最小值 8KB。这一设置使 SmartDrive 在同一时间移动较少的数据，因而降低了速度；但另一方面也减少了所占

用的常规内存。也可通过减小预读(read-ahead)缓冲区(其缺省大小为 16KB)来减少常规内存的使用。如果经常要访问未碎片化的顺序文件,增大预读缓冲器可提高 Hit 比率,当然这将占用较多的常规内存。

SMARTDRV

SmartDrive 的启动和配置,在扩展内存中创建磁盘缓存,写缓存数据。

句法:

SMARTDRV [drive[+|-]...] [/E:element] [InitCache] [WinCache] [/B:buffer] [/L] [/S]

参数与选项:

none	若还未加载 SmartDrive,以缺省值加载程序 TSR 部分。显示驱动器的缓存状态。
drive	识别所要控制缓存的盘驱(驱动器名后不加冒号)。
+|-	+使该驱动器的读写缓存均有效;-使该驱动器的读写缓存均无效。见有关缺省设置注释。
/E:element	设定 SmartDrive 一次移动缓存的数量,以字节为单位;可以是 1024,2048,4096 或 8192;缺省值为 8192。值越大,SmartDrive 使用的常规内存也就越多。
InitCache	设定 SmartDrive 启动及未运行 Windows 时的缓存大小,以 KB 为单位。其缺省值取决于系统的内存大小。
WinCache	设定启动 Windows 时缓存减到的最小值,以 KB 为单位。缺省值取决于系统内存(见注释)。
/B:buffer	指明预读缓冲区的大小可以有多个选择,缺省值为 16KB。
/L:	阻止 SmartDrive 自动加载到高端内存中,即使有可用的 UMB。
/S:	显示有关 SmartDrive 的附加信息。

图 31.2　SMARTDRV 命令简介

在通常情况下,只要可能 SmartDrive 就加载到高端内存中。在某些系统中,特别是在需要双缓冲的情况下,将 SmartDrive 加载到高端内存实际上反会降低速度。如果正使用双缓冲和高端内存块,并且有足够的常规内存,不妨试用一下/L 选项,看看能否提高系统的操作水准。

注释:　SMARTDRV 命令包括几个附加选项。/R 清除缓存中内容。/Q 抑制加载 SmartDrive 时的信息显示,而/V 则要求显示所有的信息。/S 显示附加的状态信息。

仅在 DOS 6.2 中,/F 在每次命令结束出现命令提示符时写入延迟数据,而/N 则相反。/F 为缺省设置,在使用/N 后可用/F 选项恢复缺省设置。/U 使 SmartDrive 不加载 CD-ROM 缓存模块,而只加载程序的其余部分,这可省下一点内存空间。/X 使所有驱动器的所有缓存功能无效。

以上所有选项将在"命令参考"中详细解释。

31.6　用 SMARTMON 衡量 SmartDrive 的性能

DOS 6.2 包括一个称为 SMARTMON 的 Windows 应用程序用以显示 SmartDrive 的性

能，以使调整其配置。图 31.3 显示了启动 SmartMon 出现的窗口。在缺省设置下，此窗口总是位于其他应用程序之上，因而可在进行其他工作时监测缓存性能。

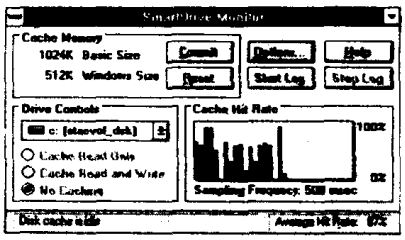

图 31.3 SmartDrive Monitor 窗口

Cache Memory 控制组显示缓存的大小，及在 Windows 下的大小。这些数值不能在 SmartMon 窗口中进行改动。要改变缓存大小必须在调整 SMARTDRV 命令后重新启动。Commit（执行）按钮迫使 SmartDrive 写入所有的延迟数据，它等效于 SMARTDRV /C 命令。Reset（重新设置）按钮用于清除缓存，等效于 SMARTDRV /R 命令。

Drive Controls（驱动器控制）组用于开关各驱动器的缓存功能。Cache Hit Rate（Cache Hit 比率）频繁地（缺省为每 500 毫秒）对缓存击中比率进行取样，并以直方图方式显示结果。在执行不同操作时，可见到此图形的变化。

状态条显示两条信息，左边的显示缓存是闲还是繁忙，右边的显示平均击中比率。随着操作不同，这两条信息将作出相应调整。

提示： SmartMon 窗口可能会挡住正使用的其他应用程序显示的信息。有时你会奇怪怎么某应用程序不再工作，而事实上可能你没看见出错信息。如果不希望 SmartMon 窗口总是在上面，可拉下控制菜单，单击 Always On Top 以退选此特性。或者也可以使之最小化。最小化图标仍将显示缓存击中比率，只是从显示当前百分数（在彩色监视器中为红色）切换到显示平均数（黑色）。

31.6.1 工作记录

可单击 Log 按钮记录下缓存的工作状况。一个典型的工作记录文件如下所示：

ticks（时钟频率数），	total（总计），	hits（击中数）
71276556，	193252，	145827
71278369，	193276，	145847
71280401，	193291，	145859
71303799，	193795，	146341

第一列显示测量时间，以自启动始的时钟频率数表达，第二列显示当时的磁盘访问总数，第三列则是缓存击中数。

在缺省情况下，工作记录存储在\DOS\SMARTMON.LOG 中。如果已存在这样一个文件，新的工作记录数据将被添加到文件末尾；如果不存在这样的文件，SmartMon 将创建一个。工作记录将持续进行直到单击 Stop Log 按钮或到达一时间限。缺省时间限为两小时；如果记录两小时的缓存操作，记录文件将非常宠大。

31.6.2　SmartMon 选项

Options 按钮打开图 31.4 所示的对话框。Cache Hit Rate 组控制着 SmartMon 测量击中率显示结果的频度。Log File 组决定文件名和记录时间限。Drive Control 组决定是否将在 Drive Controls 组中所作的所有变动自动记入 AUTOEXEC.BAT 中的 SMARTDRV 命令。也可指定另一文件来存储这些变化。

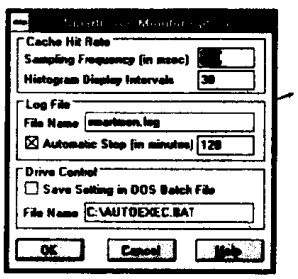

图 31.4　SmartDrive Monitor Options 对话框

31.7　磁盘缓冲区

没有扩展内存就无法使用 SmartDrive，但却可在常规内存中开辟一些磁盘缓冲空间以稍加速磁盘访问。磁盘缓冲区就像较逊色的缓存。

DOS 总是在将数据传送到发出请求的程序前，先把它从磁盘读到一缓冲区。类似地，DOS 在将程序中的数据写入磁盘前，也总是先写到一缓冲区中。如果只有一个缓冲区，它必须担负所有的读写任务，因此，DOS 下次要读取的数据已经在此缓冲区中是不太可能的。如果有 30 个缓冲区，在其中找到所需数据的几率则可惊人地提高。如果有 99 个缓冲区(最大值)，这一几率会更高，但一般情况下，系统性能将下降，因为 DOS 在缓冲区中搜寻所需数据的时间将比访问磁盘更长。

如果还配置一些预读(look-ahead)缓冲区，DOS 在程序请求读某一扇区时，会读额外的扇区。当未碎片化文件顺序读取时，这一操作可增加在缓冲中下一个所需扇区的可能性。

图 31.5 显示了 CONFIG.SYS 中 BUFFERS 命令的句法。如果用户不自己设定 BUFFERS，DOS 将依据系统配置来设置缓冲区的缺省数目。可以肯定地说，缺省设置总是不尽人意的。如果使用了功能更完善的缓存系统，如 SmartDrive，将缓冲区的数 0 减为 15 左右。如若不然，根据表 31.2 来确定设置的缓冲区数。如果打算按顺序读取一些未碎片化的文件，可包括一些预读缓冲区。

表 31.2　推荐的缓冲区设置

磁盘大小	缓冲区数
<40MB	20
40MB 到 79MB	30
80MB 到 119MB	40
<TE>>119MB	50

BUFFERS

设定磁盘缓冲区数目。

句法：

BUFFERS=read-write [,look-ahead]

参数：

read-write	指定读写缓冲区的数目。范围为 1 到 99。缺省值(如果略去 BUFFERS 命令)取决于系统配置。
look-ahead	指定预读缓冲区的数目。范围为 0 到 8。缺省值为 0。如果 read-write 或 look-ahead 超过了其范围,look-ahead 设置为缺省值 0。

图 31.5 BUFFERS 配置命令总结

一般说来缓冲区是和 DOS 位于同一位置。若 DOS 加载到 HMA,缓冲区同样在 HMA 中创建。但如果 HMA 中没有足够的空间来设置所要求的缓冲区,它们就被移到常规内存中。通常可在 HMA 中与 DOS 一起设置 40 个缓冲区,每个占一扇区,半 KB。若系统中的常规内存资源较为紧张,不妨试一下 HMA 中最多能容纳多少缓冲器。MEM /D /P 命令可用来判断它们是否在常规内存中。报告中有 BUFFERS=nn 这么一行,若此区域的大小不仅为 512 字节,则表示缓冲区在 HMA 中。

31.8 自测练习

依据以下步骤在系统中建立 SmartDrive：

1. 若已决定使用 SmartDrive,但还未在 AUTOEXEC. BAT 中增加相应命令,现在加上。
2. 若 AUTOEXEC. BAT 中早已包含了 SMARTDRV 命令,仔细调节此命令以控制驱动器缓存和缓存大小。
3. 若系统需要双重缓冲,但 CONFIG. SYS 并未加载 SMARTDRV. EXE,现在加上必要的命令。
4. 键入一命令写所有的延迟数据。
5. 显示 SmartDrive 缓存报告。
6. 若有 Windows,启动 SMARTMON 测量不同缓存选项的效率(为了试验各种缓存大小,必须多次重启动)。

第 32 章　RAM 驱动器

简介： 若有一些闲置的内存空间，可通过创建一 RAM 驱动器来提高系统运行速度。同时本章也介绍了这种做法的弊端。

32.1　有关 RAM 驱动器

如果在每次启动的时候，在内存中创建一 RAM 驱动器，并将最常用的程序和文件拷入此驱动器，由于突破了驱动器的硬件限制，将大大提高这些程序的运行速度。当然这么做也会引起一些麻烦。首先每次关机前必须记得把 RAM 驱动器中所有改动过的数据拷入一真实驱动器。其次一旦系统死机，将失去所有还未存盘的数据。

因此大多数不在 RAM 驱动器中保存任何重要的数据。RAM 驱动器最常见的用法是保存由 DOS、DOS Shell、Windows 以及诸如 Microsoft Word 等应用程序创建的临时文件。Windows 在运行中要写入和删除大量的临时文件，设置一 RAM 驱动器将明显提高系统性能。

把临时文件保存在 RAM 驱动器中，一旦系统死机，这些临时文件将随 RAM 的清除而消失。另一方面若把临时文件保存在硬驱中，它们将弃置在硬盘上，以后还可找到它们。这两种方案哪一种更好呢？系统一旦死机，则必定会失去来不及存盘的工作。如果所使用的应用程序可创建可读的临时文件，则有可能找到正确的临时文件，从而挽回本将失去的数据。但如果临时文件不可读，则它们没有任何用处，只会浪费磁盘空间。你必须根据自己对有关应用程序的了解做出决定：是将临时文件保存在硬盘中以利于必要时挽回，还是存放在 RAM 驱动器中以得到自动清除之便利。

相对于所占用的内存空间，SmartDrive 缓存比 RAM 驱动器更能提高系统性能。不要轻易在扩展内存中创建 RAM 驱动器，除非有足够的空间满足所有其他程序的需要并使 SmartDrive 有了至少 2MB 的缓存。

本章讲述如何创建和使用 RAM 驱动器，将学会如何：

- 在启动期间创建一 RAM 驱动器
- 让 DOS 和 Windows 将临时文件存放在 RAM 驱动器中

32.2　何处放置 RAM 驱动器

在拥有 4MB 及以上扩展内存的 PC 机中，放置 RAM 驱动器最合适的地方便是扩展内存。如果有真正的扩充内存而不是用 EMM386 仿真的，又不用于其他用途，则用它作 RAM 驱动器也是个好主意。否则的话，还是将它放在扩展内存中为好。不要把它放在常规内存中，这一空间非常宝贵。

32.3 RAM 驱动器的设置

可从 CONFIG.SYS 加载 RAMDRIVE.SYS 来设置 RAM 驱动器。如果要完全使用缺省设置，则不必在命令中包括任何参数和选项。但如此一来，RAM 驱动器被放置在常规内存中了。可使用/A 选项将 RAM 驱动器放置在扩充内存中，或用/E 将之放在扩展内存中。

如果有足够多的扩展内存(至少 4MB)，试着创建一 RAM 驱动器：

1. 作 CONFIG.SYS 的安全拷贝 CONFIG.SAV。
2. 用 ASCII 文本编辑器打开 CONFIG.SYS 以备编辑。
3. 键入如下命令(若从 CONFIG.SYS 加载 HIMEM，将此命令放在 HIMEM 和 DOS=HIGH 命令后面的某个地方)。

```
DEVICEHIGH=C:\DOS\RAMDRIVE.SYS /E
```

4. 重启动并注意观察启动信息(提示：可在启动期间按 Pause 键阅读信息，按任意键可继续启动)。

RAMDRIVE 的启动信息与下面的相似：

```
Microsoft RAMDrive version 3.07 virtual disk D:
    Disk size: 64K
    Sector size: 512 bytes
    Allocation unit: 1 sectors
    Directory entries: 64
```

在本例中，DOS 为 RAM 驱动器指定驱动器名 D。

RAM 驱动器的缺省大小为 64KB，每扇区 512 字节，根目录中有 64 个项。64KB 的 RAM 驱动器对于某些应用程序已足够了，比如 Microsoft Word for DOS，但对于 Windows 程序却无帮助。Windows 需要至少 2MB 空间来存储临时文件；若没有 2MB，Windows Print Manager 在打印任务中混淆这些数据，从而使打印结果毫无用处。

可在 RAMDRIVE.SYS 命令中包括一大小参数来控制 RAM 驱动器的大小。应以 KB 为单位来指定大小，从 4KB 到 32 767KB(32MB)。下面的命令在扩展内存中建立了一 2MB 的 RAM 驱动器，每扇区 512 字节，根目录中有 64 个项：

```
DEVICEHIGH=C:\DOS\RAMDRIVE.SYS 2048 /E
```

尽管 RAM 驱动器在扩展内存中，RAMDRIVE 程序却被加载到常规内存中，占据了大约 5KB 的空间。若系统被设置为使用 UMB，可用 DEVICEHIGH 代替 DEVICE，从而将它加载到高端内存中。

注释： 如果仅用 DEVICEHIGH 命令不能将 RAMDRIVE 放入高端内存，也许可用 MEMMAKER 将它挤进去。在 CONFIG.SYS 中加入 DEVICE 命令，然后运行 MEMMAKER 重新配置高端内存。

32.4　RAM 驱动器的使用

通常情况下，RAM 驱动器的用法与其他驱动器没什么区别。在 DOS Shell、MSAV、BACKUP 中，RAM 驱动器同样出现在驱动器清单中。图 32.1 显示了 RAM 驱动器在 Shell 屏幕中的样子。可用 LABEL 命令为 RAM 驱动器指定一卷标，在此驱动器中拷贝、移动、删除文件等等。甚至可用 DriveSpace 或 DoubleSpace 压缩它。

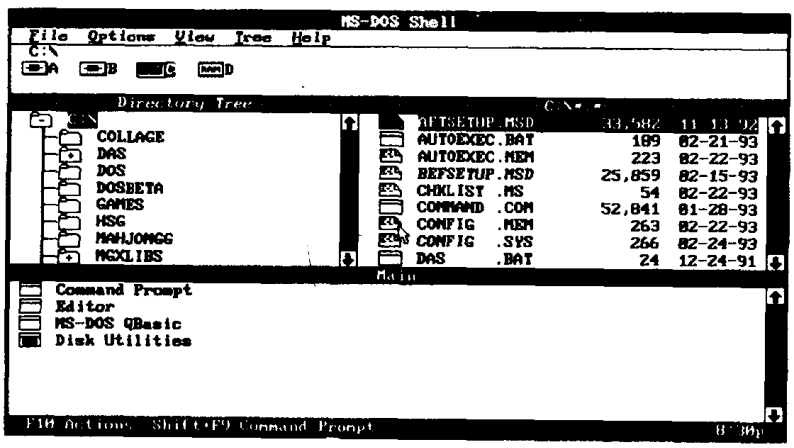

图 32.1　Shell 中的 RAM 驱动器

尽管如此，以下操作对于 RAM 驱动器却是不可行的：

- 不要格式化它，一旦创建即可使用。
- 不要缓存它，因为它在内存之中，RAM 驱动器总是以最大速度运行的。缓存实际上只能降低其速度。
- 不必扫描其中的病毒，扫描内存也就扫描了 RAM 驱动器。
- 不必为 RAM 驱动器清除碎片化；当不涉及读写磁头时，碎片化并不造成任何妨碍。
- 不要对 RAM 驱动器使用 DISKCOPY 和 DISKCOMP，它们只和软盘有关。
- 如果单纯是为了临时文件而使用 RAM 驱动器，则不要用 Delete Sentry 或删除追踪保护它。

32.5　将临时文件引向 RAM 驱动器

DOS 和 Windows 用变量 TEMP 告知何处存放临时文件。可用 SET 命令设置 TEMP 变量，如下所示：

```
SET TEMP=D:\
```

此命令设置 D 驱动器的根目录来存放临时文件。可将此命令放入 AUTOEXEC.BAT 中，从而将临时文件存放在 RAM 驱动器中。

32.6 覆盖缺省参数

RAMDRIVE 命令包括多个参数用于控制驱动器大小、扇区大小、目录项数等等(如图32.2所示)。

RAMDRIVE.SYS

使用一块计算机的随机访问内存(RAM)来模拟一硬盘驱动器。

句法:

DEVICE[HIGH]=[path] RAMDRIVE.SYS [DiskSize[SectorSize[NumEntries]]] [/E|/A]

参数和选项:

path	标识 RAMDRIVE.SYS 文件所在位置。缺省值为引导驱动器的根目录。
DiskSize	以 KB 为单位设定 RAM 驱动器所占内存大小。有效值为 4 到 32767,或可用(总是较小)。缺省值为 64。
SectorSize	以字节为单位设定扇区大小。可以是 128 256 或 512(缺省值)。Microsoft 建议使用 512。如果在命令中包括 SectorSize,必须同时包括 DiskSize。
NumEntries	设定 RAM 驱动器根目录中所能创建的文件和目录数。可以是 2 到 1024,其值最大为扇区大小值。缺省设置为 64。如果在命令中包括 NumEntries,必须同时包括 SectorSize 和 DiskSize。
/E	在扩展内存中创建 RAM 驱动器。确保使用此选项(或/A)将 RAM 驱动器排除在常规内存之外。
/A	在扩充内存中创建 RAM 驱动器。

图 32.2 RAMDRIVE 命令总结

要在扩展内存中创建一 3MB 的 RAM 驱动器,可键入以下命令:

```
DEVICE=C:\DOS\RAMDRIVE.SYS 3072 /E
```

若要指定根目录最大项数为 256,应在命令中加上扇区大小和项数:

```
DEVICE=C:\DOS\RAMDRIVE.SYS 3072 /E 512 256
```

32.7 自测练习

若有足够的内存空间,并希望尝试一下 RAM 驱动器,可进行以下练习:

1. 在 CONFIG.SYS 中插入必要命令加载 RAMDRIVE.SYS。
2. 重新启动并决定 DOS 指派给 RAM 驱动器的名称。
3. 若要将 RAM 驱动器用于临时文件,在 AUTOEXEC.BAT 中插入必要的 SET 命令。
4. 重启动使 SET 命令发挥作用。

第 33 章　去除文件碎片

简介： 大约每过一个月，应运行 DEFRAG 优化硬盘，缩短访问时间。

33.1　有关文件碎片化

簇或分配单元是 DOS 用来存储文件的最小单位。比如有一个文件 UPTIME.BAT，大小为 2 500 字节。若磁盘簇大小为 512 字节，则一个 2 500 字节的文件需要 5 个簇。

DOS 不一定将 UPTIME.DAT 存储在 5 个相邻的串中。当你往一空盘上加数据时，DOS 逐个定入文件，每个文件占据相邻的簇。但删除操作使这一结构出现了空洞，DOS 可将新文件写入这些空洞。随着磁盘逐渐变满，DOS 开始难以找到地方写入整个文件，于是不得不将它们碎片化，将数据存放任一空簇。

当加大一个文件从而需要更多的簇，而下一个相邻簇已被另一个文件占据时，也会发生碎片化。DOS 必须向前搜寻下一个可用的簇，这样就产生了一个新碎片。当在磁盘上进行文件删除、写入、修改等操作时，就有可能产生越来越多的碎片化文件。

读取一个碎片化文件对 DOS 来说并没什么困难。文件分配表(FAT)将它指向文件的每个簇，即使这些簇并不相邻。但为了访问一个碎片化文件，驱动器的读写磁头必须多次移动以找到文件的所有簇。每移一次，就必须等待转到合适的扇区。移动读写磁头是访问驱动器最慢的一部分，它大大增加了读取文件的时间。若硬盘中有许多碎片化文件，进入系统的速度就会降低。

可使用磁盘优化程序，如 DOS 6 的 DEFRAG 来减少或消除盘驱的碎片化。这些优化程序有两条途径来消除碎片文件：

- 根据需要移动数据，将每个文件的簇移到一起。
- 移动簇，使空白簇位于磁盘末尾。这一操作使得已使用的簇和空白簇各成一块。当 DOS 往磁盘上添加新文件时，将它们写入相邻簇的可能性就会增大(但对于文件扩充仍存在问题)。

本章讲述如何用 DEFRAG 优化硬盘。将学习如何：

- 进行运行 DEFRAG 前的准备工作
- 启动 DEFRAG
- 选择优化方法
- 完成优化操作

注释： 不能在网络驱动器、用 INTERLNK 创建的驱动器，或诸如 CD-ROM 之类的只读驱动器上使用 DEFRAG。如果试图在一个用 DoubleSpace、DriveSpace，或诸如 Stacker 之类的第三方压缩程序压缩过的驱动器上运行 DEFRAG，压缩软件将

打断命令并执行它自身的去除碎片化功能。

33.2 运行 DEFRAG 前的准备工作

在运行 DEFRAG 前，必须使磁盘作好准备。遵循以下操作步骤，从而在本章中练习使用 DEFRAG。准备好驱动器 C，除非它是一个压缩驱动器。如果这样的话，那就准备另一个局域硬盘。如果没有其他的局域硬盘，则不需要学习 DEFRAG，可跳过本章。

如何优化硬盘

警告： 不能用 DEFRAG 对网络驱动器、只读驱动器、用 INTERLNK 创建的驱动器，或压缩驱动器进行优化。以下步骤仅适用于局域未压缩驱动器。

1. 准备对硬盘进行 DEFRAG：
   a. 删除所有不需要的文件。
   b. 删除 PCTRACKER.DEL。
   c. 运行 SCANDISK 或 CHKDSK /F 纠正所有报告的问题。
   d. 整理目录结构。
2. 退出正在运行的所有程序，回到初始命令提示符。
3. 键入 DEFRAG /B。
   DEFRAG 检测内存，然后访问驱动器名。
4. 选择需优化的驱动器。
   DEFRAG 对此驱动器进行分析，并推荐一种优化方法。
5. 选中 Optimize 以接受推荐的方法。
   DEFRAG 在 DEFRAG 屏幕上显示其工作进程。程序结束时，DOS 重新进行启动。
6. 选中 Configure 以选择不同的优化方法。
   拉下 Optimize 菜单。
7. 选中 Optimization Method。
   出现 Select Optimization 对话框。
8. 选择 Optimization Method。
   再次出现 Optimize 菜单。
9. 选中 Begin Optimization。
   当 DEFRAG 对所选驱动器进行操作时，在 DEFRAG 屏幕上显示其进程。程序结束后，DOS 重新进行启动。

如果现在就想试一下 DEFRAG，请遵循以下步骤：

1. 删除所有不需要的文件，以空出尽量多的空间。表 31.1 列出了一些可能可删去的 DOS 文件。
2. 如果使用了删除追踪，可删去 PCTRACKR.DEL。去除碎片化之后，这一文件就没有什么用处了，因此不如先删掉它，以留出更多的空间。PCTRACKR.DEL 是在根目录中的一个隐含文件，因此在删除前必须用 ATTRIB 去除它的隐含属性。
3. 运行 SCANDISK 或 CHKDSK，恢复所有丢失的分配单元并纠正此驱动器所有的目

录结构问题。第 27 章“CHKDSK 和 SCANDISK”详细介绍了如何使用 SCANDISK 或 CHKDSK 完成这一操作。去除丢失的分配单元可释放上千字节的硬盘空间，从而使文件变得更紧凑，离目录和 FAT 表更近。

SCANDISK 和 CHKDSK 同样可检查驱动器的目录树和 FAT 表中的问题。它们可纠正大多数发现的问题(必须以 CHKDSK 命令中包括/F 选项)。DEFRAG 不能处理目录结构或 FAT 表中的错误，因此必须在运行 DEFRAG 前纠正它们。

4. 使那些对于系统基本功能并非绝对必要的 TSR 无效。特别注意应使所有可能写盘的 TSR 无效，比如用来自动刷新文件的 TSR。照着以下步骤使 TSR 无效：
   a. 用 ASCII 编辑器打开 CONFIG. SYS。
   b. 在欲使之无效的命令前插入 REM。
   c. 存储 CONFIG. SYS。
   d. 对 AUTOEXEC. BAT 进行同样的处理。
   e. 重启动。

注释： 运行 DEFRAG 后，去掉插入的 REM 以恢复那些 TSR。

警告： 如果当 DEFRAG 正在识别簇时，TSR 进入写盘，则某些文件可能遭受无法修复的损坏。

5. 整理目录结构。这个过程也许比较费时，但却能得到较短的路径名，并使子目录保持一合理长度，因此还是值得的。在 Windows 或 DOS Shell 中这一操作最为简便，因为也同时看到整个目录树以及一个或多个目录清单，并且还能在需要时四处拖动文件。如果需要重新组织目录结构，不妨现在就完成它，这样在优化文件时就能用上新目录结构。

表 33. 1　可删除的文件

文件名	描　述
HIMEM. SYS	如果你的系统没有扩展内存或所使用的是第三方内存管理程序(如果使用了 DOS 的 HIMEM. SYS，就可删除 Windows 版本，反之亦然。)对 HIMEM 的解释参见第 29 章“内存管理”
LOADFIX. EXE	如果没有扩展内存或未将 DOS 加载到 HMA 中，可删除此程序。有关 LOADFIX 的解释参见第 29 章
EMM386. EXE	如果你的计算机并非 386 或 486，或者使用了第三方内存管理程序，可删除此程序。有关 EMM386 的解释参见第 30 章“为 386、486 和奔腾机增加的内存工具”
RAMDRIVE. SYS	如果没用它来创建 RAM 驱动器，就可删除此文件。有关 RAM 驱动器的解释参见第 32 章“RAM 驱动器”
SMARTDRV. SYS	如果没使用它，或系统无扩展内存，或使用了第三方缓存系统，或使用了 Windows 提供的更新版本，可删除此程序。(如果使用的是 DOS 版本，则可删除此程序的 Windows 版本)。有关 SMARTDRV 的解释参见第 31 章“驱动器缓存”

（续表）

文件名	描 述
DRIVER. SYS	如果不需要重新定义和命名盘驱，可删除此文件。有关 DRIVER. SYS 的解释参见第 51 章“配置磁盘和网络”
GRAPHICS. COM GRAPHICS. PRO	这些文件用于在 DOS 中打印图形屏幕，如果不需要这一功能，就可删除它们。有关 GRAPHICS. PRO 和 GRAPHICS. COM 的解释参见第 50 章“DOS 的打印功能”
NLSFUNC. EXE KEYB. COM COUNTRY. SYS DISPLAY. SYS	如果在美国并且不需要国际语言支持，可删除此文件。有关国际语言支持的解释参见第 49 章“处理另一个国家的信息”
DEBUG. EXE	对于非程序员可删除此文件
MSHERC. COM	如果不打算用 QBasic 或没有 Hercules 图形匹配器，可删除此文件
*. TXT	如果不再需要这些 DOS 目录中的文档文件，就可删除它们
FASTOPEN. EXE	如果不打算用它在 RAM 中对硬盘目录进行缓存，就可删除它。有关 FASTOPEN 的解释参见命令参考
SHARE. EXE	如果不在网络上并且不使用诸如 Windows 之类的多任务系统，就可删除此文件。有关 SHARE 的解释参见第 51 章
DOSSHELL. * EGA. SYS	如果不打算使用 DOS Shell，就可删除这些文件。有关 DOS Shell 的解释参见第二部分
ASSIGN. COM，BACKUP. EXE，COMP. EXE，CV. COM，DELIN. EXE，EXE2BIN. EXE，GRAFTABL. COM，JOIN. EXE，4208. CPI，5202. CPI，LCD. CPI，MIRROR. COM，MSHERC. COM，PRINTER. SYS，PRINTFIX. COM，REMLINE. BAS	这些文件在 DOS 5 中还留着，到了 DOS 6 则已废弃不用了。但如果需要，它们还是可以在 DOS 6 中运行的。4201. CPI 也许还有用，其他文件要是不用的话，就删除它们

提示： 如果安装的 DOS 6 不太完全，则没有上面最后的一系列文件。如果出于某种原因需要它们，可从 Microsoft 公司订购。有关订购信息参见《User's Guide》。

33.3 启动 DEFRAG

准备好硬盘后，就可能开始运行 DEFRAG 了：

1. 退出运行中的 Windows、DOS Shell 及其他程序。

警告： 不要在有文件打开的情况下运行 DEFRAG，它可能会把文件的簇当作可用空间而覆盖它们。

2. 在初始命令提示符下，用以下命令启动 DEFRAG：

```
DEFRAG /B
```

/B 选项告知 DEFRAG 在结束时重新启动系统。有可能还有一些程序在同时运行，通常是一些保存文件分配信息的 TSR。DEFRAG 使旧分配信息失效，如果有程序试图

以过时的信息来访问磁盘，文件会造成损坏。重启动清除由这些程序维护的缓冲区，迫使它们重读驱动器。

DEFRAG 做的第一件事就是检测系统内存以确保能可靠地高速移动数据。如果系统未通过此检测，就不能使用 DEFRAG，直到解决这一问题。

DEFRAG 然后要求选择一驱动器（如图 33.1 所示）。

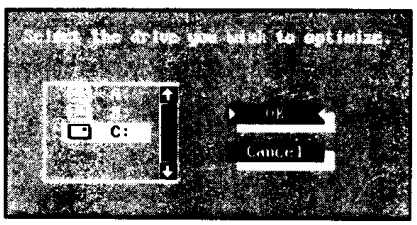

图 33.1　Drive Selection 对话框

如果出现的对话框与图 33.1 中的不同，则意味着你的驱动器可能压缩过，因此访问的是压缩程序的去除碎片化功能。要继续去除碎片化操作，应参照压缩程序提供的提示（对于 DoubleSpace 和 DriveSpace，参见第 34 章“DriveSpace 和 DoubleSpace”）。

如果还有一个局域驱动器，并希望继续本章的练习，准备好该驱动器从步骤 1 重头开始一遍，但注意应在启动命令中包括驱动器名，比如 DEFRAG D: /B。

注释： 在目前的情况下，屏幕上下部的 Status 和 Legent 框没有什么意义。屏幕中部则显示正在进行的操作。

提示： 可在 DEFRAG 的任何阶段按 F1 获取帮助信息。

3. 选中驱动器 C 并选中 OK。

   DEFRAG 对盘驱进行分析并推荐一种优化方法（如图 33.2 所示）。DEFRAG 将在 No Optimization，Unfragment Files Only 或 Full Optimization（去除文件碎片化，并将空区放在一起）三项中推荐其一。

4. 选中 Optimize 对盘驱进行优化（如果推荐的是 No Optimization，选中 OK）。

   在 DEFRAG 工作时，可阅读下一章节。结束操作后，由于在 DEFRAG 命令中包括了 /B 选项，DOS 将自动重启动。

33.4　Drive Map 诠释

屏幕背景是一驱动图。右下角的 Legend 框解释了如何理解此图。在图 33.2 中，每个图块表示 35 个簇。由于此图总是代表整个盘驱，因此这一数字取决于驱动器大小。

DEFRAG 不能移动或重写系统和隐含文件。比如防拷贝程序文件一旦安装后就不能移动。其他包括一些系统文件，比如 DOS 核心程序文件必须占据盘中的特定位置。驱动器图中标有 X 的块表示其中至少有一些簇包含着不可移动文件。

注释： 若某块中包含着各种类型簇的混合，DEFRAG 使用最高优先权的符号。从高到低，优先权依次为 bad(B)，unmovable(X)，used 和 unused。比如某块中包含一

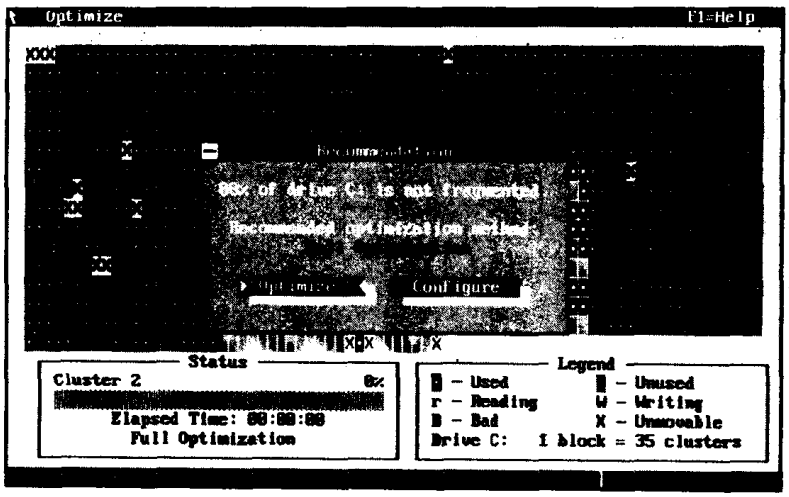

图 33.2 Recommendation 对话框

些不可移动区和未使用区，它的符号为 X(表示不可移动)。

33.5 优化进程

DEFRAG 开始优化时，其图动态显示正在进行的操作。可从图中看到块正在读取(用 r 表示)和写入(用 W 表示)，同时使用和未使用的块在图中移动(当然坏簇和不可移动簇没有移动)。图 33.3 显示了优化过程中屏幕的样子。可在第二行右边末端附近看到 r，这表示 DEFRAG 正在读取此块。

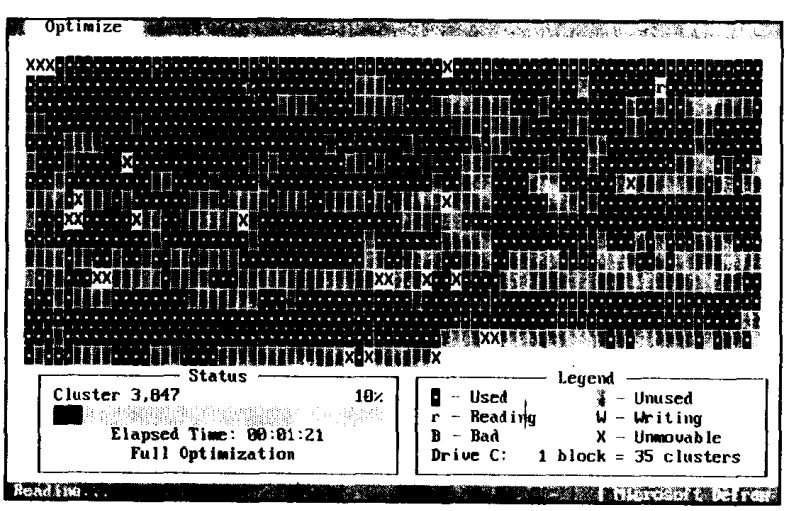

图 33.3 Optimization 进程屏幕

此例中的 Status 框显示此驱动器的 10%已去除了碎片化，正进行到第 3847 个簇，耗时 1 分 21 秒。因为 10%的任务用了 1⅓秒，可以推测整个过程需 12 分钟——足够用来休息一会儿。优化方法显示在 Status 框的底部。去除碎片化操作结束后，出现一对话框(如图 33.4

所示)。

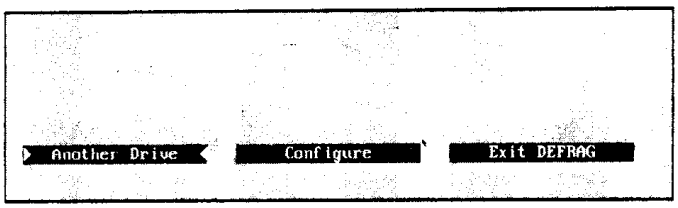

图 33.4　Optimization Complete 对话框

33.6　Optimize 菜单

用 DEFRAG 优化磁盘时,可在仅去除文件碎片化(Unfragment Files Only)和重新组织未使用的空间(Full Optimization)之间作出选择。偶尔这两种目标互不相容。当去除文件碎片化时,DEFRAG 不可避免地在不可移动文件旁边留下一些空洞。而重新组织未使用空间时,DEFRAG 可能会将一个文件分块存储至这些新洞。当然稍有碎片,并不会引起性能的明显降低。去除碎片的目的在于去除大量的碎片。

大多数情况下,可使用推荐的优化方法。但如果愿意的话,也可选择另一方法。比如,假定你要将一个大数据库拷到盘上,并希望它占据相邻簇以缩短访问时间。这时无论推荐的是何种方法,都应使用 Full Optimization 将所有的未使用空间移到磁盘末尾。

如果想覆盖掉推荐方法,在 Recommendation 对话框中选中 Configure(如图 33.2 所示)。对话框关闭并拉下 Optimize 菜单。

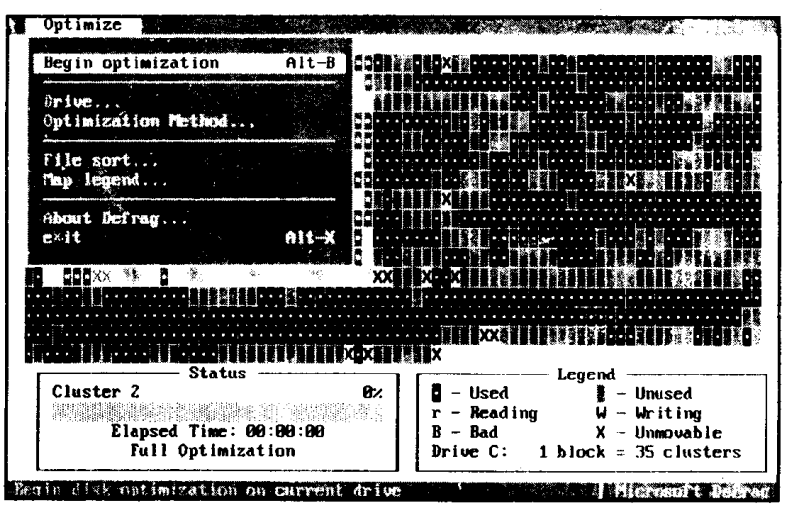

图 33.5　Optimize 菜单

用 Optimization Method 命令可打开如图 33.6 所示的对话框。在 Select Optimization Method 对话框中选择一方法后,又回到 Optimize 菜单,并可从中选中 Begin Optimization 以启动优化过程。

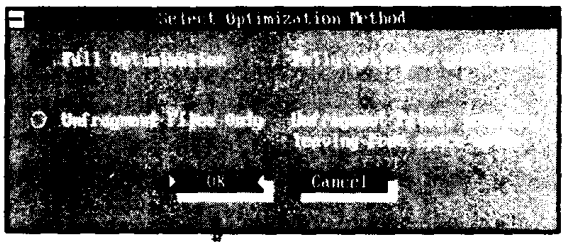

图 33.6 Select Optimization Method 对话框

33.7 目录项的分类

DEFRAG 有时可方便地进行另一项操作——对目录项进行分类。这与 DIR 的分类功能不同，后者只是对由它产生的文件清单进行分类而不是对保存在磁盘中的项。DEFRAG 实际上移动了目录中的项。

如何对目录项进行分类

1. 如同“如何优化硬盘”所解释的那样准备好驱动器。
2. 退出所有正运行的程序，回到初始命令提示符。
3. 键入 DEFRAG /B。
   DEFRAG 测试内存，然后询问驱动器名。
4. 选中欲优化驱动器。
   DEFRAG 分析此盘驱并推荐一种优化方法。
5. 选中 Configure。
   打开 Optimize 菜单。
6. 选中 File Sort。
   出现 File Sort 对话框。
7. 选择排序关键词和顺序并选中 OK。
   回到 Optimize 菜单。
8. 若不想使用所推荐的方法，选择一优化方法。
9. 选中 Begin Optimization。
   DEFRAG 在对所选驱动器进行操作的同时，在 DEFRAG 屏幕上显示其进程。结束后，DOS 重启动。

在某些情况下，在执行其他操作前，也许先需要对目录项进行分类。比如正准备将一系列文件拷到软盘中寄给一用户，可能需要将文件按字母顺序排列起来。又如正用一诸如 PKZIP 之类的第三方压缩程序准备文件以通过调制解调器传送，PKZIP 按它在目录中找到的顺序将文件添加 ZIP 文件中，并以同样的顺序展开它们。

要对目录项进行分类，应选中 Optimize 菜单中的 File Sort。然后选择 File Sort 对话框中的分类顺序。可按名称、扩展名、日期和时间，或大小的上升或下降顺序进行分类。

33.8　其他 DEFRAG 命令选项

图 33.7 对 DEFRAG 命令进行了总结。前面已学过如何用/B 选项在优化后重新启动计算机。/V 选项和/SKIPHIGH 选项同样提供在 DEFRAG 菜单和对话框中没有的功能。

DEFRAG

重组驱动器中的文件，优化文件操作性能；对目录项进行分类。

句法：

DEFRAG [drive] [/F|/U] [/Sorder] [/V] [/B] [/SKIPHIGH]

参数和选项：

none	打开 DEFRAG 对话框，选择驱动器和选项。
drive	指定优化驱动器。
/F	去除文件和未使用空间的碎片后(完全优化)。
/U	仅去除文件碎片化。
/Sorder	对目录项进行分类；order 指定分类方法：
	N　按名称(字母顺序)
	N-　按名称(逆字母顺序)
	E　按扩展名(字母顺序)
	E-　按扩展名(逆字母顺序)
	D　按日期和时间，较早的在先
	D-　按日期和时间，较晚的在先
	S　按大小，较小的在先
	S-　按大小，较大的在先
/V	移动数据后进行验证。
/B	DEFRAG 结束后重新启动。
/SKIPHIGH	将 DEFRAG 程序加载到常规内存中。

图 33.7　DEFRAG 命令总结

如果想让 DEFRAG 验证写入的数据，可使用/V 选项；但如果用了 DOS 的验证功能，则不必再用/V 选项。/SKIPHIGH 可用来阻止 DEFRAG 加载到高端内存中。一般情况下，应让 DEFRAG 在高端内存中运行，因为这更经济。但如果系统出了些问题，则可能需要用/SKIPHIGH 选项迫使 DEFRAG 加载到常规内存中。还有一些选项在“命令参考”中详细解释。

/H 选项用来使去除碎片化过程包括隐含文件，但是必须记住许多隐含文件是不应移动的(可将文件设为系统文件从而阻止它被移动)。/LCD 或/BW 用来在 LCD 或单色模式下显示 DEFRAG 的对话框。如果系统不能正确显示图形或屏幕上出现雪花点，/G0 用来抑制 DEFRAG 对话框中的图形显示，所有这些选项均在“命令参考”中作了详细描述。

33.9 以批处理模式运行 DEFRAG

可在 DEFRAG 命令中包括驱动器名和优化方法，从而避免打开 DEFRAG 对话框（如图 33.7 所示）。比如要对驱动器 D 进行一带验证的完全优化，可键入下面命令：

```
DEFRAG D: /F /V /B
```

键入此命令后，DEFRAG 即开始优化操作。对于要从批处理程序（比如一个每周运行一次的清理程序）去除一驱动器的碎片化来说，这是一个好方法。

注释： Norton Untilities 的 SpeedDisk 优化程序提供了多个 DEFRAG 所没有的功能。特别是，它可在 Windows 的后台工作，与此同时可继续其他工作。

33.10 自测练习

作为本章练习的一部分，已对引导驱动器进行了去除碎片化操作。如果还有一些其他局域硬盘，现在就为它们去除碎片化。从现在开始，不要忘了经常（也许每月一次）去除盘驱的碎片化。

第 34 章　DriveSpace 和 DoubleSpace

简介： 对于许多人来说，DOS 6 最吸引人的功能是数据压缩，它使硬盘所存储的数据量增加一倍。

34.1　DOS 6 和数据压缩

用完硬盘空间是早晚的事。买一块新硬盘又过于昂贵。如果你的硬盘用尽了其空间，而同时又还能用上一段时间——硬盘的平均寿命为 3 到 5 年，压缩其数据或许是解决问题的方法。一个优秀的磁盘压缩程序通常可将盘驱所能存储的数据量翻倍。DOS 6.0 和 6.2 包括了一称为 DoubleSpace 的压缩程序，DOS 6.22 用 DriveSpace 取代了 DoubleSpace。

警告： DOS 6.2 为 DoubleSpace 增加了一些重要的安全措施。如果你是用 DOS 6.0 安装了 DoubleSpace，并且还未进行升级，请立即这么做。可升级到 DOS 6.2 或 6.22。

注释： 如果你用的是 DOS 6.2 版本的 DoubleSpace，可不经转换到 DriveSpace 而升级到 DOS 6.22。

尽管这两个数据压缩程序的内部原理颇不一致，但它们的用户界面却是相同的。除了程序名本身以外，它们的命令、窗口和对话框都是一样的。

本章讲述如何使用这两个程序，将学习如何：

- 用 DriveSpace 或 DoubleSpace 压缩引导驱动器
- 理解用 DriveSpace 或 DoubleSpace 创建的驱动器
- 压缩附加驱动器
- 浏览有关压缩驱动器的报告
- 从 DoubleSpace 转换到 DriveSpace

注释： 使用的数据压缩程序不能多于一个。如果已经使用了诸如 Stacker 之类的第三方数据压缩程序，就不应将 DriveSpace 或 DoubleSpace 加入系统。可将系统从第三方程序转换到 DriveSpace 或 DoubleSpace，但似乎没有什么理由非要这么去做，既然目前的程序工作得好好的。

34.2　数据压缩的工作方式

数据压缩的策略基于大多数程序包括了大量重复之处这一事实。对于一连串字节，数据压缩程序仅在它在文件中第一次出现存储它，其他则用一指向第一串字节的代码代替。代码所占据的空间要比它所取代的数据少，因此整个文件也就变小了，从而盘驱中有了更多的可

存储其他文件的空间。

比如下面的句子是某文件中的第一个句子，DriveSpace 或 DoubleSpace 将取代标有方括号的字符串：

Data compression schemes [ta]ke advan[ta]ge of t[he] fact [th] [at] most fil[es]includ[e] [a]lot [of] repeti[ti] [on].

即使从这个小例子也可以看出，随着文件的加长，越来越多，越来越长的字符串可用代码替换。当 DOS 需要读取这些文件时，DriveSpace 或 DoubleSpace 将它解压缩，将各个代码以适当的字符取代。同样的数据压缩策略也为 DOS 6 的备份程序所采用。

DriveSpace 或 DoubleSpace 同时还通过以扇区而非簇来分配文件，清除浪费的空间。在压缩硬盘上的一个小文件可能只需 512 字节的空间，而不是 2048 字节或更多。对盘上每个这样的小文件都可省下 1.5KB。

34.3 压缩驱动器

第一次运行 DriveSpace 或 DoubleSpace，它创建一压缩驱动器，无论是外表还是操作都像是一个真正的驱动器，但事实上却不是。它实际上是一个称为 Compressed Volume File (CVF：压缩体积文件)的文件，它驻留在真正的盘驱中，后者被称为宿主(host)驱动器。CVF 是宿主驱动器根目录中一系统、隐藏、只读文件。

CVF 和宿主驱动器有各自的驱动器名，由 DriveSpace 或 DoubleSpace 指定。假定在你压缩了驱动器 C 后，DriveSpace 或 DoubleSpace 将字母 C 指派给 CVF，将 K 给宿主驱动器。这之后一旦向驱动器 C 写入文件，DriveSpace 或 DoubleSpace 压缩此文件并放入 CVF。当从驱动器 C 读取文件时，DriveSpace 或 DoubleSpace 将其解压缩。如果愿意的话，仍可在驱动器 K 的剩余空间里读取未压缩文件。

某些文件是不可压缩的，DriveSpace 或 DoubleSpace 将它们存放在宿主驱动器中。其中包括 Windows 交换文件，IO.SYS，MSDOS.SYS，DBLSPACE.BIN 和一些 DriveSpace/DoubleSpace 文件。这些文件通常是系统或隐藏文件，因此如果列出的仅是非常系统和非隐藏文件，寄主驱动器看起来就像是空的。它们都是些很重要的文件，不要删除其中的任何一个。

34.4 已存在驱动器的压缩

当用 DriveSpace 或 DoubleSpace 压缩一已存的驱动器时，它在宿主驱动器中创建一 CVF，压缩盘驱中的全部或大部分文件，并将它们存入 CVF。然后 DriveSpace 或 DoubleSpace 将原宿主驱动器名指派给 CVF，而将一新名指派给宿主驱动器。这一驱动器指派过程，称为安放 DriveSpace 驱动器，使得可用原驱动器名访问 CVF 中的数据。这样就不必改变已习惯了的批处理程序、命令、宏指令和路径名称。

每次启动或重启动时，DriveSpace 或 DoubleSpace 必须重新安放压缩驱动器。在启动过程中，可看到 DriveSpace 或 DoubleSpace 信息。有些时候，当你使用一 DriveSpace 或 DoubleSpace 功能如 DEFRAGMENT 时，会见到一条信息说 DriveSpace 或 DoubleSpace 正在重新安放驱动器。

34.5　设置 DriveSpace 或 DoubleSpace

第一次运行 DriveSpace 或 DoubleSpace，它将执行以下任务：

- 设置核心程序，DRVSPACE. BIN 或 DBLSPACE. BIN，每次启动时进行加载
- 在引导驱动器的根目录中安装 DriveSpace 或 DoubleSpace 的系统文件（不管压缩的是哪个驱动器）
- 在 CONFIG. SYS 中安装一命令，使核心程序置于内存中合适位置
- 在 DOS 6. 2 或 6. 22 中，运行 SCANDISK 检查将要压缩的驱动器

第一次压缩硬盘上的一个驱动器时，以上的这些操作自动进行。不能不压缩硬盘而设置 DriveSpace 或 DoubleSpace。

警告： 在对驱动器进行压缩之前，先进行扫描是非常重要的。否则有可能失去驱动器中的数据。在安装这些数据压缩之前，应确保已升级到了 DOS 6. 2 或 6. 22。DOS 6. 0 不包括 SCANDISK，不能首先检查驱动器。如果已在升级到 DOS 6. 2 或 6. 22 前安装了 DoubleSpace，应备份驱动器中所有数据（非常重要）并立即运行 SCANDISK。

如果已使用了其他类型的数据压缩系统，就不要再安装 DriveSpace 或 DouleSpace。如果要转换到 DOS 的数据压缩程序，应先下载系统中的其他类似程序，具体操作参见 DOS 目录中的 README. TXE 文件的第 7 部分。

34.6　DOS 数据压缩初步

不要仅仅为了好奇而安装 DoubleSpace 或 DriveSpace。安装过程耗时较多，并且总有一定的可能性使安装失败，从而不得不从备份中恢复盘驱中的数据。同时，如果并不打算继续使用此压缩，移去 DoubleSpace 或 DriveSpace 的过程耗时更长，并且同样可能导致失败。

不能压缩过满的驱动器。为了压缩引导驱动器，DriveSpace 和 DoubleSpace 至少需要 1. 2MB 的自由空间。如果没有这么多空间，可将一些文件移到软盘中以释放空间。驱动器压缩后，盘驱中将会有更多的空间，这时可再移回来。

如果你还不清楚该如何安装数据压缩程序，本节将指导你完成这一过程。下面的步骤是针对如何安装 DOS 6. 22 的 DriveSpace 的，如果要安装 DOS 6. 2 的 DoubleSpace，步骤几乎相同，除了用 DBLSPACE 命令代替 DRVSPACE 以外（再次提醒：不要在 DOS 6. 0 下安装 DoubleSpace）。

如果要在此时安装 DriveSpace 或 DoubleSpace，进行以下步骤的操作：

1. 更新备份，以确保对欲压缩驱动器上的每个重要文件进行了备份。
2. 退出 Windows 或 DOS Shell。
3. 如果通常使用网络，连接到此网络及其驱动器上，从而使 DriveSpace 获知你通常使用的所有驱动器名。

4. 键入下面命令：

DRVSPACE

DriveSpace 以一 Welcome 屏幕致以问候，而后显示 Setup 选择屏幕（如图 34.1 所示）。

Express 设置可自动压缩驱动器 C。

如何初次压缩硬盘

1. 备份欲压缩驱动器。
2. 退出所有的程序，回到初始 DOS 命令提示符。
3. 键入 DRVSPACE。
   出现 Welcome 屏幕。
4. 选中 Continue。
   DriveSpace 访问采用 Express 或 Custom Setup。
5. 选中 Express Setup。
   DriveSpace 再给一次改变主意的机会。
6. 按 C 继续。
   DriveSpace 在压缩驱动器的同时，显示其进程。

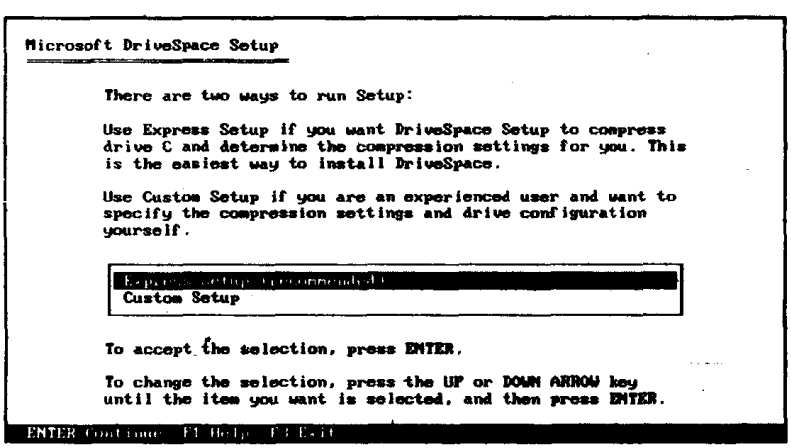

图 34.1 Setup 选择屏幕

注释： 如果不想压缩驱动器 C，必须选中 Custom Setup 来压缩另一个驱动器。此练习中使用了 Express Setup。如果所选的是 Custom Setup，在继续第 5 步操作前，必须选择欲压缩的驱动器。

选择 Express Setup 后，一信息警告你别忘了在继续操作前，对驱动器备份。这时可选择退出（从而备份盘驱）或 Continue。

5. 如果已在步骤 1 中备份了文件，选中 Continue。
   随后的屏幕显示给出最后一次退出的机会。
6. 按 C 继续。

注释： 如果在驱动器 A 中有一磁盘，DriveSpace 将要求你拿开它，以使 DriveSpace 在需要时可重新启动系统。

34.6.1　压缩过程

DriveSpace 在对盘驱进行压缩时，显示一系列的信息和操作屏幕。你所能做的就是观察（或者干脆走开）。即使关闭计算机电源也不能取消压缩操作，一旦再打开电源，DoubleSpace 将从它中断的地方继续开始。

首先 DriveSpace 运行 SCANDISK 检查盘驱，此时必须确保纠正每一个发现的问题。然后真正开始压缩，它将及时通知用户压缩进程——当前正在压缩哪个文件，花去的时间和预期结束的时间，驱动器中未压缩部分的百分比，等等。

压缩完成后，DriveSpace 在压缩驱动器上执行自身的 DEFRAG 功能并调整 CVF 的大小。结束后，DriveSpace 显示一终止屏幕，比较驱动器中自由空间的初始值和压缩后值；同时显示压缩率和压缩驱动器所花的时间。读完这一屏幕后，只需按 Enter，DriveSpace 即重新启动系统。至此，设置结束，新的压缩驱动器随时可投入使用。

注释： 2 比 1 的压缩率表示在未压缩文件中两字节的内容，在压缩文件中仅需一字节。

提示： 在安装 DriveSpace 或 DoubleSpace 后，可运行 MEMMAKER 将压缩程序从常规内存移出来。

34.6.2　压缩其他驱动器

如何压缩其他驱动器

1. 备份欲压缩的驱动器。
2. 退出所有的程序，回到 DOS 初始命令提示符。
3. 键入 DRVSPACE。
   打开 DriveSpace 窗口。
4. 选中 Compress Existing Drive。
   DriveSpace 显示未压缩驱动器清单。
5. 选择欲压缩之驱动器并按 Enter。
   DriveSpace 显示缺省压缩设置。
6. 若接受这些设置，按 Enter。
   DriveSpace 再给一次机会改变主意。
7. 按 C 继续。
   DriveSpace 在压缩驱动器的同时，显示一进程屏幕。

初次运行 DriveSpace 或 DoubleSpace 压缩驱动器后，键入 DRVSPACE 或 DBLSPACE 命令可得到如图 34.2 所示的屏幕。此屏幕列出了系统中所有的压缩驱动器，并提供了一些滚动式菜单执行各种 DriveSpace 功能。要想压缩另一个驱动器，单击菜单条中的 Compress 或按 Alt+C 来选中 Compress 菜单（如图 34.3 所示）。然后选中 Existing Drive 选项。

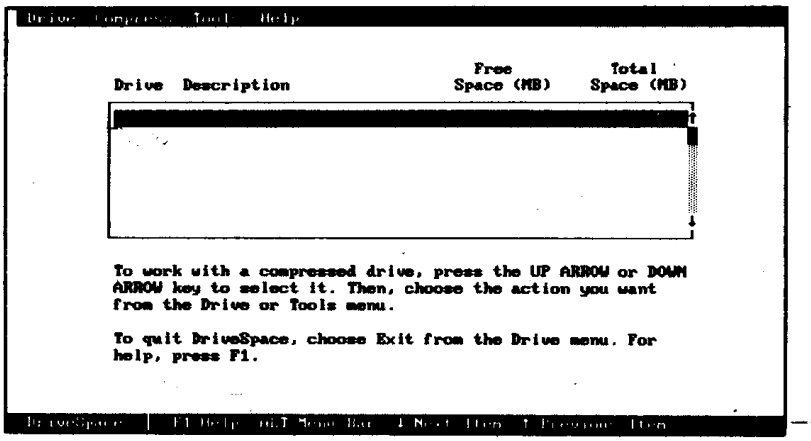

图 34.2 选择压缩驱动器

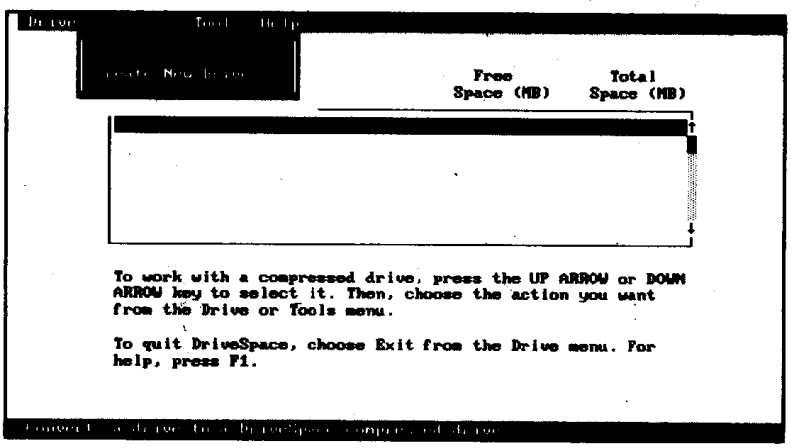

图 34.3 Compress 菜单

34.6.3 选择一驱动器

选择 Compress Existing Drive 后，所出现的屏幕如图 34.4 所示。屏幕中的指示指导你如何选择欲压缩驱动器。所列的驱动器即所能压缩的驱动器。你也许会奇怪为什么其他驱动器不在此清单中。这是因为网络驱动器、CD-ROM 驱动器及过满的驱动器都是不可压缩的。压缩引导驱动器，DriveSpace 至少需要 1.2MB 的自由空间，其他驱动器需有 664KB 或更多。

提示： *要压缩自由空间不够多的驱动器，可暂时把其中的一些文件移到另一个驱动器或软盘中。压缩后，就有足够的空间把它们移回来了。*

图 34.5 显示了下一个显示的屏幕。在缺省设置下，DriveSpace 在宿主驱动器中留出 2MB 的未压缩空间保存不能压缩的文件，当然也可设置一个与之不同的值。同样也可改变 DriveSpace 指派给宿主驱动器的名称。但是除非有充足的理由，否则不如接受 DriveSpace 所建议的设置。根据屏幕上的指示来修改或接受所显示的设置。

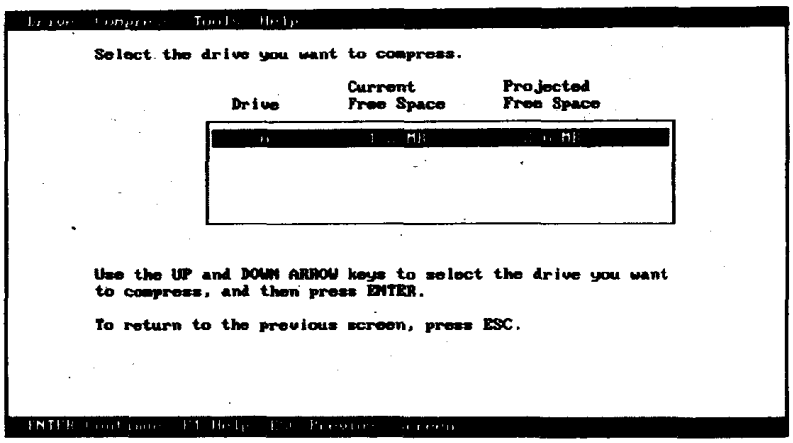

图 34.4　选择一欲压缩驱动器

注释： *尽管在此屏幕中提到了一新驱动器，它实际上是有了一个新名称的原宿主驱动器。*

接受了压缩设置后，DriveSpace 提醒你对文件进行备份，然后再一次给予退出的机会。如果选择继续，DriveSpace 将以与压缩第一个驱动器类似的方式压缩此驱动器。

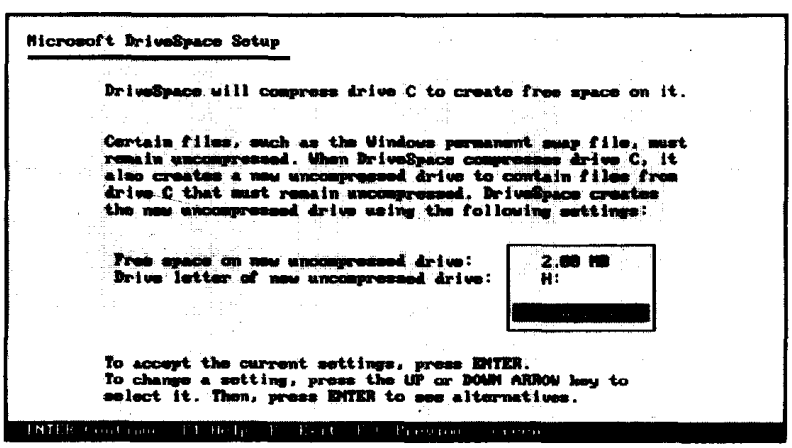

图 34.5　压缩设置

34.7　与其他 DOS 应用程序的关系

DriveSpace 和 DoubleSpace 与 DOS 其他应用程序的关系被设计得较为自然。比如，可在 DOS Shell 等的驱动器清单中见到新压缩的驱动器以及宿主驱动器。可在压缩驱动器中创建新目录，移动和拷贝文件，删除和恢复文件，等等。

但是不能用 FORMAT 命令格式化压缩驱动器，必须使用 DriveSpace 或 DoubleSpace 自身的格式化功能（见“命令参考”）。

必须像对其他驱动器一样，经常备份压缩驱动器。可用 SCANDISK 扫描并修理压缩驱

动器，也可用 Microsoft Anti-Virus 程序检查病毒。可用 SmartDrive 缓存宿主驱动器，这一过程将同时缓压缩驱动器。

由于组织方式的不同，碎片化只能降低未压驱动器的速度，对压缩驱动器却没有类似的作用。压缩驱动器并不需要经常化地去除碎片化。事实上，不能在压缩驱动器上运行 DEFRAG。如果要对压缩驱动器去除碎片化，必须使用压缩程序自身的去除碎片化功能，这已在第 33 章“去除碎片化”中作了介绍。

34.8 从 DoubleSpace 转换到 DriveSpace

正如前面所提到的，如果你正使用 DOS 6.0 的 DoubleSpace，应立即进行升级。如果已升级到 DOS 6.22，则应转换到 DriveSpace。你的系统中是否有转换程序取决于所安装的 DOS 6.22 的版本。如果没有这一程序，可设法从 Microsoft 公司订购，详见 DOS User's Guide。

注释： 如果 DRVSPACE 命令并没能开始下述的转换过程，则说明你的系统中没有转换程序。

有了转换程序后，就可以键入 DRVSPACE 命令，并按屏幕上的指示行事了。DriveSpace 将所有能找到的 DoubleSpace 空间转换为它自己的格式。必须首先对文件进行备份！这一过程可耗时 3 小时或更长，可在结束一天的工作后开始它。

转换完成后，你会发现 DBLSPACE 程序文件仍存在于驱动器中。不要删除它们，DriveSpace 需要它们完成两项功能：

- 用于恢复使用 DoubleSpace 时，用 Backup 的压缩选项备份的文件
- 用于访问在可移动存储介质（比如软盘）上用 DoubleSpace 压缩的文件，这一介质在转换到 DriveSpace 的过程中，不在系统内部

SCANDISK 同样需要 DoubleSpace 程序文件来扫描和修理未转换到 DriveSpace 的 DoubleSpace 存储空间。

34.9 压缩驱动器的管理

DriveSpace 菜单提供了多个管理压缩驱动器的命令。比如，可检查目录和 FAT 表结构，并浏览有关压缩驱动器的统计信息。选中 Drive Info 即可显示驱动器信息。比如驱动器中可用的自由空间数量。大部分所显示的信息的含义是不言自明的，但自由空间数值（Space free figure）需要略加解释。

如果未压缩驱动器有 2MB 的自由空间，即此驱动器上有 2 097 152 个未使用字节，那么你能加入的数据量就是这么多，不可能再多。对于压缩驱动器来说，则有一点不同。在压缩驱动器中，DriveSpace 或 DoubleSpace 预测可加入驱动器中的未压缩数据量。比如 DriveSpace 或 DoubleSpace 报告有 20.13MB 的自由空间，这表示它经推算得知能添加 20.13MB 的未压缩文件。这一推算是根据预测的压缩率给出的，所以并不总是正确的。也许只能添加 15MB 的程序文件，因为它们一般不能达到 1.8:1 的压缩率。另一方面，可能可添加 50MB 甚至更多的图形文件，因为它们的压缩率可大到 10:1。

34.10　DIR 和压缩驱动器

当在压缩驱动器上使用其他 DOS 命令时，它们将与 DriveSpace 或 DoubleSpace 相协调。比如在压缩驱动器上使用 DIR，它将报告驱动器中可用空间的预期值，而不是真实值。可在 DIR 命令中加上/CH 选项，来显示文件压缩率。报告将与下面的内容类似：

```
Volume in drive D has no label
Volume serial Number is 13EF-2218

Directory of D:\FRENCH
.            <DIR>        02-21-94      12:28P
..           <DIR>        02-21-94      12:28P
CHKLIST      MS               54  12-22-94   3:58P     4.0 to 1.0
FLASHCRD     BAS          30,896  05-05-94  10:42a     2.3 to 1.0
FLASHCRD     EXE          91,084  05-05-94  10:43a     1.3 to 1.0
VOCAB        BAS          31,265  05-05-94  11:42a     2.2 to 1.0
VOCAB        EXE          91,242  05-05-94  11:42a     1.3 to 1.0
WORK         BAS          13,054  05-04-94  11:59a     1.6 to 1.0
                  1.4 to 1.0 average compression ratio
        8 file(s)     257,595 bytes
                   21,102,592 bytes free
```

在本例中，CHKLIST.MS 的压缩率为 4:1，而两个 EXE 文件仅为 1.3:1。此清单中文件（不是整个驱动器）的平均压缩率为 1.4:1。

注释： DIR /CH 对用第三方程序，如 Stacker，压缩的驱动器同样有效。DIR 的/C 选项也能显示压缩率，但不要使用它。/CH 选项可给出更准确的结果。

清单中显示的文件大小为未压缩大小，这样就可看出一个文件如果加载到内存中或移动、拷贝到未压缩驱动器中需占用多少空间。

34.11　DriveSpace 和 DoubleSpace 的附加功能

本章的讨论只触及 DriveSpace 和 DoubleSpace 的表面，刚够用来创建和使用压缩空间。当你希望了解更多的知识时，可以下面所列的内存作为参考：

- 创建和使用空的压缩空间
- 在软盘上创建和使用压缩空间
- 压缩空间的去除碎片化
- 删除压缩驱动器
- 重新格式化压缩驱动器
- 列出所有驱动器的信息，而不只是压缩驱动器的
- 改变压缩驱动器的预期压缩率
- 改变压缩驱动器的大小
- 去除驱动器的压缩状态

这些功能在第 45 章“再谈 DrvSapce 和 DblSpace”中解释。

34.12 自 测 练 习

如果在本章中安装了 DriveSpace 或 DoubleSpace，应确保学会以下操作：

- 显示 DriveSpace 或 DoubleSpace 产生的报告
- 在目录清单中包括压缩信息

第 35 章　系统优化综述

简介： 本章将讲述如何把 HIMEM、SmartDrive、VSAFE、UNDELETE、BACK-UP……各种 DOS 功能综合起来，成为一协调的系统以优化操作，保护数据。

35.1　检查 CONFIG.SYS 和 AUTOEXEC.BAT

随着加入 PC 机的应用程序的增多，CONFIG.SYS 和 AUTOEXEC.BAT 会变得非常混乱。你可能自己往其中加了一些命令，但更多的命令是由各种应用安装程序加入的。应常常检查一下 CONFIG.SYS 和 AUTOEXEC.BAT，以确认你明白所有的命令的意义，并确保这些命令的顺序合适，没有那些已删除程序留下的过时命令。如果使用了高端内存，应将尽可能多的程序加载到高端内存中。如果不是，再运行 MEMMAKER，以达到此目的。

本章以两种典型的系统为例：一较小的使用扩展内存但不用高端内存，基于 286 的系统，和一较大的装备有 CD－ROM 驱动器的奔腾系统。你的系统或许介于这两者之间，当然也有可能有着更多的附属设备，但不管怎么说，这儿所举的例子将有助于分析自己的启动文件。

在阅读本章时，不妨向自己提下列答疑：

- 我已充分利用了我的 PC 机的内存资源吗？
- 我的启动文件包含了一些不必要的命令吗？
- 所有的命令顺序合理吗？
- 有一些我所不能理解的命令吗？

35.2　286 系统的设置

对于一配有 2MB 扩展内存的 286 计算机来说，下面的例子可以说是设计合理的 CONFIG.SYS：

```
DEVICE=C:\DOS\SETVER.EXE
DOS=HIGH
DEVICE=C:\DOS\SETVER.EXE
DEVICE=C:\DOS\ANSI.SYS
BUFFERS=15
FILES=50
STACKS=9,256
```

本例中使用的命令解释如下：

- DEVICE=C:\DOS\HIMEM.SYS 首先加载扩展内存驱动程序，从而可为其他程序提供扩展内存支持。

- DOS=HIGH 在其他程序之前，将 DOS 加载到 HMA 之中。
- DEVICE=C:\DOS\SETVER.EXE 由 DOS 的 SETUP 程序安装。你的系统不一定需要 SETVER。
- DEVICE=C:\DOS\ANSI.SYS 加载 ANSI.SYS 驱动程序，因为此系统中至少有一个程序需要此驱动程序(你的系统也许需要它，但如果不使用它的话，不妨删掉它)。
- BUFFERS=15 仅设置了 15 个缓冲区，因为此系统使用了从 AUTOEXEC.BAT 加载的 SmartDrive，下面将会看到这一点。
- FILES=50 设置了 50 个文件柄，因为此系统中有一个程序需要它们。也许你需要的文件柄会少一点，但应在 CONFIG.SYS 文件中保留 FILES 命令。
- Stacks=9,256 由某些需要它的程序安装，不能改动或删除它(除非你不再拥有那个程序了)。

此系统的 AUTOEXEC.BAT 文件至少应包含以下命令(你的 AUTOEXEC.BAT 也许包含更多的命令，用以设置 TSR 和其他想要的功能)。

```
@ ECHO OFF
PATH C:\DOS;C:\WINDOWS; C:\
DOSKEY
SMARTDRV
UNDELETE /LOAD
VSAFE
MSAV
```

这些命令的目的如下所述：

- @ECHO OFF 关闭文件中包含的其余命令的命令响应。
- PATH 建立此系统的程序搜寻路径。
- DOSKEY 设置 DOSKEY 程序以记录命令史。
- SMARTDRV 加载 SmartDrive 缓存系统，设置为扩展内存中的 1MB 缓存(缺省值)及缺省缓存参数。
- UNDELETE /LOAD 根据当前配置加载 Delete Sentry 删除保护系统。
- VSAFE 加载 VSAFE 病毒监视程序。
- MSAV 设置 Microsoft Anti-Virus 扫描程序用以立即扫描系统中的病毒。此命令只打开第一个对话框，用户必须结束扫描并终止 MSAV。

35.3 386 及其以上系统的设置

对 386、486 及奔腾 PC 机来说，可使用 EMM386 提供高端内存支持并将驱动程序和 TSR 加载到 UMB 中。运行 MEMMAKER 可设置 EMM386。以下是用 MEMMAKER 修饰的 CONFIG.SYS 的典型例子。

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS
BUFFERS=15,0
FILES=50
```

```
DOS=UMB
LASTDRIVE=E
FCBS=4,0
DOS=HIGH
STACKS=9,256
DEVICEHIGH /L:1,5888=C:\DOS\RAMDRIVE.SYS 3092 /E
DEVICEHIGH /L:1,12048=C:\DOS\SETVER.EXE
DEVICEHIGH /L:1,6696=C:\DEV\MTMCDS.SYS /D:MSCD001 /P:300 /A:0
DEVICEHIGH /L:1,9072=C:\DOS\ANSI.SYS
DEVICEHIGH=C:\DOS\DRVSPACE.SYS /MOVE
```

这些命令完成了以下任务：

- DEVICE=C:\DOS\HIMEM.SYS 在所有程序前加载扩展内存驱动程序。
- DEVICE=C:\DOS\EMM386.EXE 提供高端内存支持，NOEMS 选项抑制了此程序的扩充内存设施(没有此选项，EMM386.SYS 将把 256KB 的扩展内存转换为扩充内存)。
- Buffers=15 仅设置 15 个缓冲区，因为正如下面将看到的，此系统从 AUTOEXEC.BAT 加载了 SmartDrive。
- Files=50 设置 50 个文件柄，因为系统中有一程序需要它们。你需要的文件柄也许会少一些，但应在 CONFIG.SYS 文件中包括 Files 命令。
- DOS=UMB 使高端内存块为 DOS 所用。
- LASTDRIVE=E 保留 5 个驱动器名空间。
- FCBS=4,0 保留用 FCB 方法追踪 4 个文件的空间。
- DOS=HIGH 加载 DOS 到 HMA 中。
- Stacks=9,256 由需要它的程序安装，不应修改或删除(除非不再拥有那个程序了)。
- DEVICEHIGH...RAMDRIVE.SYS 在扩展内存中创建一 3MB 的 RAM 驱动器。此 RAM 驱动器用于 DOS 和 Windows 临时文件，正如从 AUTOEXEC.BAT 文件确定的那样。
- DEVICEHIGH... SETVER.EXE 加载 SETVER 程序到高端内存中。
- DEVICEHIGH... MTMCDS.SYS 加载 CD-ROM 的设备驱动程序(这一驱动程序与 CD-ROM 一起出售，并不是 DOS 的一部分)。
- DEVICEHIGH...ANSI.SYS 加载 ANSI.SYS 到高端内存。
- DEVICEHIGH...DRVSPACE.SYS 在安装 DriveSpace 时插入。

MEMMAKER 修饰了加载 RAMDRIVE.SYS、SETVER.EXE、MTMCDS.SYS、ANSI.SYS 和 DRVSPACE.SYS 的命令，从而将这些驱动程序加载到高端内存的特定位置。

此系统的 AUTOEXEC.BAT 文件如下所示：

```
@ECHO OFF
PATH C:\DOS; C:\WINDOWS; C:\

REM Direct temporary files to the RAM drive:
SET TEMP=D:\

SMARTDRV 4096
LH /L:1,53968 UNDELETE /LOAD
```

```
LH /L:1,63088 VSAFE
LH /L:1,65288 SHARE
LH /L:1,52700 MSCDEX /D:MSCD001
```

这些命令完成了以下任务：

- @ECHO OFF 抑制本文件中其他命令的命令响应。
- PATH 建立程序搜寻路径。
- SET，正如命令所指出的那样，将 DOS 和 Windows 临时文件引向 RAM 驱动器 D。因 RAM 驱动器有 3MB 大小，它有足够的空间来存放 Windows 临时文件。在此 16MB 系统中，还有足够的扩展内存用来建立一 4MB 的 SmartDrive 缓存。
- SMARTDRV 加载 SmartDrive 缓存系统，以缺省缓存参数在扩展内存中建立一 4MB 的缓存。
- LH...UNDELETE 加载 UNDELETE TSR 到高端内存。
- LH...VSAFE 加载反病毒监视程序到高端内存。
- LH...SHARE 加载一称为 SHARE 的程序，它用以避免同时在 Windows 中运行的应用程序争抢同样的文件。SHARE 在 DOS 软件包中，如果经常使用 Windows，不妨加载它。
- LH...MSCDEX 启动 DOS 的 CD-ROM 软件。

在开始 Windows 时，还将执行另外两个命令。Windows StartUp 组（未在此处显示）加载 MWAVTSR.EXE 程序支持 VSAFE 并启动 MWAV 病毒扫描程序，这样系统一启动，就能扫描内存和硬盘中的病毒。

35.4 每日关机程序

你不仅应考虑启动程序，还应考虑关机前该做些什么。比如，也许需要运行一不同的或渐增的备份。如果在 DOS 6.0 下使用 SmartDrive 的延迟写入，可能需要在断电前写入延迟数据（DOS 6.2 在显示命令提示符前写入所有的延迟数据。因此只要小心别在命令提示符以外的地方造成死机，就不需要处理延迟写入）。

如果使用了 MSBACKUP，可建立一 SHUTDOWN.BAT 文件自动运行备份程序。可用 DOS Editor 创建此文件，它可以和下面所写的一样简洁：

```
MSBACKUP DAILY.SET
```

MSBACKUP 命令启动程序并打开第一个对话框。你仍需在对话框中完成备份。回到命令提示符后，即可关机。

35.5 每周关机程序

每过一周，应运行一更广泛的关机程序，进行完全备份，运行 SCANDISK 和 DEFRAG（必须在局域硬盘，包括压缩盘驱中运行的 SCANDISK）。如果使用 MSBACKUP，批处理程序如下所示：

```
MSBACKUP DAILY.SET
SCANDISK
DEFRAG /B
```

当 MSBACKUP 程序进行 DAILY 设置时，将备份类型重新设置为 Full 以完成一完全备份。当 MSBACKUP 询问是否将变动存入 DAILY.SET 时，选择 No，从而使保存的仍是 Differential 或 Incremental 类型。

35.6　后续章节简介

一直到本章，都是本书的基本部分。为了更好地控制 PC 机，保护数据，并能从麻烦中复原，至此已学习了所有的 DOS 功能。概括如下：

- 使用 DOS Shell(如果没有更先进的外壳的话)来管理磁盘、目录、文件和程序
- 在 DOS 命令提示符下管理磁盘、目录、文件和程序
- 用 Delete Sentry、Backup、Microsoft Anti-Virus 等程序保护文件
- 用 EMM386、MEMMAKER 等更好地利用 PC 机内存资源
- 用 DEFRAG、DriveSpace 等更好地利用硬盘

本书的以下部分将对某些功能作更详细的描述，特别是那些能显著提高系统性能的选项。比如，将解释如何压缩一软盘，重新引导命令输出的更多方式，以及如何为多重配置设置 CONFIG.SYS 和 AUTOEXEC.BAT。

还将学习至今还未介绍的 DOS 功能，比如：

- 写自己的批处理程序
- 控制 DOS 环境
- 通过 INTERLNK 连接两台计算机
- 设置 PC 的国际化用途
- 刷新旧 BASIC 程序到 QBASIC
- 用 FDISK 分配硬盘

35.7　自 测 练 习

浏览你自己的 CONFIG.SYS 和 AUTOEXEC.BAT 文件。现在应当明白这两个文件中命令的用途了。

第五部分 DOS 高级应用

第 36 章 命令行状态下的 DOS 应用

简介： 通常说来，对 DOS 命令理解得越透彻，使用计算机的效率也越高。高级命令通常更加强大有效，因而可以提供像 DOS Shell 和 Windows 程序项一样功能强大的批处理程序，使 DOS 的效应在命令行状态下也能得到尽善尽美的发挥。

36.1 最大限度地发挥 DOS 命令行及 DOS Shell 的作用

很多用户都已掌握了命令行下的基本命令以及 DOSKEY 所提供的许多特殊功能。但是，除此以外，还有很多工具可以使我们更有效地使用这些命令。本章中将介绍以下内容：

- 输入重定向、输出重定向以及管道命令
- 用 MORE 命令查看分页显示的文件或输出
- 用 FIND 命令从文件或输出信息中查找指定的行
- 用 SORT 命令对文件排序
- 用多重命令定义 DOS Shell 的程序项
- 使用可替换参数使 DOS Shell 程序项更灵活适用
- 调整 DOS Shell 程序项所占用的内存量
- 理解 DOSSHELL.INI 的内容

36.2 输入输出重定向

假设需要打印当前驱动器中根目录中的内容，我们只需用“>”符号将 DIR 命令的输出重定向至 PRN 设备。具体操作如下：

1. 置打印机为 ready 状态。
2. 转至根目录。
3. 键入 DIR /A－D > PRN（/A－D 选项列出所有文件，包括系统文件和隐含文件，但不包括子目录的内容）。

注释： 请参见第 12 章中表 12.1 中所介绍的 DOS 设备名。

如果不想打印而只想将目录内容存入一个文件中，可采用下列方式作重定向：

1. 键入 DIR /A－D > DIRSAVE.TXT 并执行。

DOS 将在当前目录下建立一个名为 DIRSAVE.TXT 文件。这是一个普通的 ASCII

文本文件，可用 TYPE、EDIT 查看，也可用 PRINT 打印。

2. 为显示 DIRSAVE. TXT 中的内容，可执行命令 TYPE DIRSAVE. TXT。这样 DOS 便会显示出文件的内容。

提示： 对重定向生成文件赋以扩展名. TXT。TXT 扩展名与 DOS Shell 中的 EDIT 及 Windows 中的 Notepad 相关联，用户可很方便地查看 TXT 文件。

假若只想向 DIRSAVE. TXT 中加入目录内容，则不能用“>”符，因为那样将使得原有内容被取代。表 36. 1 中列出了所有的重定向标识符。“>>”可提供向文件尾附加内容的功能。具体地说：

1. 运行 DIR /A－D DOS >> DIRSAVE. TXT（注意：可用 DOSKEY 调出前面用到过的一行 DIR 命令并编辑修改之）。
2. 打印出 DIRSAVE. TXT 文件。

此时，由于文件很长，屏幕无法完全容纳其内容。为了看到整体，必须将文件断成页，这一功能在下一部分会介绍到。

表 36. 1　重定向符

符号	描述
<source	将来自外设或文件的输入作重定向
>destination	将输出定向到文件或外设；目标文件被覆盖
>>destination	将输出重定向到指定的文件或外设；若目标文件已存在，新内容将附加在其后
command1|command2	将 command1 的输出作为 command2 的输入（管道）

注释： 表 36. 1 中所列的符号是 DOS 的保留字符。不能在用户自己设定的目录名或文件名中使用它们。

36. 2. 1　用 MORE 作分页显示

为了查看 DIRSAVE. TXT 的细节，可用 EDIT 打开它，并编辑之。如果只需查看其内容，更简单的方法是用 TYPE 命令。可结合使用 TYPE 及 MORE 命令将输出作分页显示。

关于分页显示

1. 用下列命令可将输出内容置成分页形式：

command|MORE

2. DOS 会将输出内容放在一个临时文件中（若输出内容较多，将花费一定时间），然后显示输出的第一页，最后一行显示－－MORE－－。
3. 要查看下一页，只需按一下任意一个键（用 Ctrl+Break 可中止查看并回到 DOS 命令行方式）。

MORE 不带参量。缺省方式下，它从键盘读取文件并将其一页页显示在监视器上。可用两种方法将 MORE 的输入重定向为其他的来源，一是使用重定向符页“<”，其次可用管道命令符“|”。具体操作为：

1. 键入 MORE < DIRSAVE. TXT。
   DOS 显示 DIRSAVE. TXT 文件的第一页，并在底部显示——MORE——。
2. 按任意键以查看一页。
   DOS 则显示出 DIRSAVE. TXT 的第二页。
3. 继续按键直到查看完整个文件。
4. 键入命令 TYPE DIRSAVE. TXT|MORE。
   DOS 将显示 DIRSAVE. TXT 文件的第一页。
5. 按任意键以查看下一页。
6. 若不想继续，可用 Ctrl+Break 结束显示并加到 DOS 命令行方式。

管道命令同样适于那些不具备以/P 选项作分页显示功能的命令。例如对 ATTRIB 及 TREE：

1. 键入 ATTRIB DOS*.* | MORE。
   DOS 将列出 DOS 子目录下所有文件的属性，而且是分页显示的。
2. 键入 TREE \ |MORE。
   DOS 将分页显示当前硬盘上的目录树。

36.2.2 临时管道文件

为实现管道功能，DOS 必须建立两个临时文件，比如 ABOABOBH 及 ABOABOBN。当命令执行完毕或用 Ctrl+Break 结束时，临时文件将被自动删除。但是，在命令执行过程中，若对计算机作热启动或冷启动，临时文件将废弃在硬盘上，必须由用户自行删除。

可以用 SET TEMP=path(通常将此命令置于 AUTOEXEC. BAT 中)来设置临时文件放置的路径，否则 DOS 将把它们放在当前目录中。假若临时文件存放于速度相对较慢的磁盘上，遇到较大的输出量时，将消耗很多的等待时间。因为 DOS 向盘上写入临时文件时，用户不得不停下来等待。在第 32 章“RAM 驱动器”中将介绍一种方法，可使临时文件存储到虚盘中，这样可大大提高速度。

如果磁盘被写保护了，或者空间不足，管道命令将无法建立临时文件，因而就无法执行(关于细节，请参看第 43 章“DOS 及其运行环境”)。

36.2.3 向文本文件中附加一些内容

用户有时需要用重定向功能向某一文本文件(比如说 DIRSAVE. TXT 文件)头部插入一些东西。我们知道，ECHO text 命令可将 text 显示在显示器上。若对 ECHO text 作重定向，text 也可以被重定向到某一文件或设备中。可以用 ECHO 及重定向功能将文本内容置于某个文件的首部。特别地，“ECHO”(ECHO 后面加一个句号点)将输出一个空行。对“ECHO”作重定向，可向文件中加入一个空行。

用下面的方法可试验一下有关操作：

1. 键入 ECHO Root directory of drive C: > DIRSAVE. TXT。
   DOS 将以 Root directory of drive C:这段文字覆盖 DIRSAVE. TXT 文件，而且不作任何警告。

警告： “>”重定向符在不作任何提示的情况下覆盖掉同名文件。为了安全起见，应使用“>>”，除非确信要覆盖掉同名文件。第 37 章“批处理程序中的编辑逻辑”将介绍检测某一文件是否已存在的方法，并给出相应的命令。

2. 键入 ECHO. >> DIRSAVE. TXT。
   文件尾部将出现一空行。
3. 键入 DIR /A－D >> DIRSAVE. TXT。
   DOS 将向文件尾部加入目录表。
4. 键入 ECHO. >>DIRSAVE. TXT 两次。
   文件属部将加入两个空行。
5. 键入 ECHO DOS directory on dirve C: >> DIRSAVE. TXT。
   DOS 将向文件尾部加入这段文字。
6. 键入 ECHO. >> DIRSAVE. TXT。
   文件尾部将加入一空白行。
7. 键入 DIR /A－D DOS >> DIRSAVE. TXT。
   DOS 将向文件尾部加入 DOS 目录表。
8. 键入 TYPE DIRSAVE | MORE 来查看文件。
   请注意我们向文件中插入的内容。

36. 2. 4　输入输出重定向的机制

并非所有的输入输出重定向都能用重定向符控制，一般而言，只有那些来源于键盘的输入，以及发送到显示器的输出，才能作为重定向的内容。

更准确地说，许多 DOS 命令需请求 I/O(输入/输出)服务。当某一命令调用 I/O 服务时，DOS 便在一个名为 CON 的设备上作读写操作，这一设备通常被定义为键盘或显示器。

重定向功能可以与任何请求 I/O 服务的应用程序结合使用，这些应用程序可以是 DOS 提供的命令或用户程序。但是，对那些自己管理 I/O 任务而不使用 DOS 标准 I/O 服务的应用程序，重定向功能将无法实施。

要注意的是，即使是 DOS 应用程序，也并非都能与重定向功能结合使用。例如，TYPE 及 COPY 不使用标准输入，但由于它们自身就能从文件或设备上读取数据，因此输入重定向是不必要的(但是 TYPE 的输出能被重向)。类似地，PRINT 命令的打印输出不能被重定向到文件中，也是因为它不使用标准 I/O 服务。PRINT 命令输出的内容中仅有要求用户确定的那一部分信息可以被重定向，这部分信息就是通常显示在屏幕上的那些文件。

36. 2. 5　标准错误输出

DOS 执行中发现错误时，例如缺少文件时，将向标准错误输出设备(但不是标准输出设备)输出错误信息，并由 DOS 显示在控制台上。由于重定向功能只对标准输出起作用而不影响标准错误输出，因此出错信息将不能被重定向。为检验这一点，可如下操作：

1. 执行命令 DIR XXXX. XXX。
   DOS 的响应包括两种类型的输出，前三行是标准输出，最后一行是标准出错输出：

```
Volume in drive C is HARD DRIVE
Volume Serial Number is 171D-78E4
Directory of C:\
File not found
```

2. 再试试重定向时的情形。若执行 DIR XXXX. XXX > NUL 将输出重定向至名为 NUL 的设备，标准输出将消失，屏幕上只显示 File not found。

用户也许很难理解这些特性的用途，但是掌握这些细节的确可得到一些便利。例如，可以写一个 DOSKEY 的宏，用于判定文件是否存在：

```
DOSKEY ISFILE=DIR $1 $G NUL
```

这一命令定义了一个名为 ISFILE 的宏，可用来查找参数项 $1 中给出的文件名，但将标准输出抛弃掉（在 DOSKEY 的宏定义中，必须以 $G 来代替“>”）。若文件不存在，将显示 File not found；否则无任何输出。

注释： 实际上，用户是可以对所有的键盘输入及显示器输出作重定向的，只需用 CTTY 命令重新定义 CON。这一技术将在第 48 章“控制台的控制”中介绍。

尚需注意的另一点是，有些命令用标准输出来输出出错信息（比如 CHKDSK 等），而某些命令则用标准出错输出方式来输出普通信息（MORE 即是一例）。所有这些并非程序员任意指定的，而是依需要而行。以 CHKDSK 为例，用户使用重定向处理 CHKDSK 的输出时，通常是为了得到它的一个打印件或文件副本，而且希望将出错信息包括在其中。

36.2.6 将输出重定向到 PostScript 打印机

PostScript 打印机需要特殊的驱动程序，DOS 无法用其所需的模式打印。在通常情况下，这些打印机可设置成为仿真的标准并行口打印机（像一般的 Epson 和 IBM 打印机），这样 DOS 即可直接对其进行控制了（除了不能对其响铃及啸声功能加以利用）。对于 PostScript 打印机，可查看一下它的文档，找到仿真设置的方法。

36.2.7 用 ECHO 简化命令的运行

仅仅运行 DATE 命令而不带参数时，DOS 将显示当前日期并要求输出新的日期设置。用户必须以回车应答以保持当前设置，例如：

```
C:\>DATE
Current date is Sat 08-27-1994
Enter new date (mm-dd-yy):
```

即使只想看一看系统日期，也必须用一个多余的回车来结束 Date 命令，这是很令人不快的。尤其是在批处理文件中执行时，更有诸多不便。幸而可以采用管道命令来简化这些不必要的操作。DOS 可从管道而不必从键盘读取这一回车信号，因此不必停下来等待用户输入。操作如下：

1. 键入命令 ECHO. |DATE。
   DOS 将显示日期并自动回到命令行状态。
2. 对 TIME 命令亦可如法炮制。键入 ECHO. |TIME。

DOS 将显示当前时间并自动回到命令行状态。

这一技巧可用于 DEL 命令中。若想以 DEL *.* 删除目录中的所有文件，DOS 将访问 Are you sure(Y/N)?。用户必须以 Y 应答才能继续执行。若用 ECHO Y|DEL *.* 命令以管道的方式向其输入一个 Y，DOS 将直接执行而不必再等待用户从键盘输入(注意：除非的确想删除目录中所有的文件，否则不要轻易尝试这一命令)。

36.3　DOS 的文件过滤功能

DOS 将三个命令设计成过滤器(过滤程序从标准输入读入数据，作一定操作后再输出到标准输出设备)。这三个 DOS 命令是 MORE，SORT 和 FIND。MORE 用于将输出分页，SORT 用于将输出按序排列，FIND 用于查找包含特定字符串的行。MORE 命令前面已介绍过，在下面几小节中将介绍 SORT 和 FIND 命令的用法。

36.3.1　行的概念

前述三种文件过滤器都是以数据行为单位进行操作，而数据行一般以回车符作为分隔符。在像批处理文件、CONFIG.SYS 文件一类的 ASCII 文件中，数据行是很明显的。但是，许多文字处理软件并不是每行加一个回车符，而是在段落结束时才加上。尽管这些软件在显示文件时能将段落分行显示，DOS 却认为每段仅包含一行文字(除非文件被存储为 ASCII 方式)。因此，若要对文字处理软件产生的文件作为过滤处理时，必须预先知道其文件的存储格式。

36.3.2　用 SORT 命令对行排序

SORT 命令原来用于对 DIR 命令的输出排序。然而新版本的 DIR /O 命令会自动对目录信息排序，而不必以 SORT 来辅助。SORT 现在一般用于处理简单的列表，如一些索引文件，而对复杂的工作，往往要用较为复杂的排序软件。

图 36.1 列出了 SORT 命令的格式。若不带参量直接运行 SORT，计算机将从标准输入设备读入数据行，并将其排序后输出。后面的图中列出了从键盘输入数据行的方法。

SORT

对 ASCII 类型的数据行排序然后输出结果。

句法：

SORT [/R] [/+n] [<source] [>destination]

参数和选项：

source	需作排序的数据源；缺省时为 CON，从键盘读入数据直至用户键入 CTRL+Z。
destination	指定输出内容的去向；缺省时为 CON，即将输出显示到屏幕上。
/R	将输出倒序排列。
/+n	对从第 n 列开始的内容排序。

图 36.1　SORT 命令总结

从 CON 设备输入文件

大多数 DOS 命令需要从 CON 设备输入一些字符，如"Y"或"N"，等等。对 DOS 过滤程序而言，需输入的不仅是几个字符，而是整个文件(除非将文件重定向为它们的输入)。此外，CON 也可作为 COPY 命令的输入源，这种情况下，所需的也是一整个文件。文件一般以文件结束符结尾，这一结束符的 ASCII 码为十六进制的 1A，在屏幕上则显示为 ^ Z。

在 EDIT 命令出现之前，大多数 DOS 用户都熟知如何从 CON 设备输入文件，因为毕竟这是输入简单文件的最便捷的方法。而这也是最难的文件输入手段，用户每次只能处理一行内容，而无法跳回前一行作任何删改。

在 EDIT 出现之后，用户已根本不必从 CON 设备输入文本文件；但是，若未装入 EDIT，或希望尝试一下这一输入方法，可继续阅读以下说明：

1. 输入命令 MORE、SORT、FIND 或 COPY 后，DOS 将把光标置于下一行的行首，等待用户的输入。
2. 仔细地输入每一行文字，并在每次回车换行前检查并改正可能存在的错误。一旦按回车后就无法再回去改了(MORE 和 FIND 都立即处理当前行，然后停下来等待下一行输入，由此引起的结果常常令一般用户感到诧异)。
3. 输入完最后一行后，可按 F6 或 Ctrl+Z 为文件加上结束符(FIND 和 SORT 命令都要求文件结束符出现在新行的行首)。输入结束后，DOS 再对文件作处理。

36.3.2.1 重定向 SORT 命令的输入输出

用户一般将某个文件重定向为 SORT 的输入，并用管道命令符获得其输出，然后又将这些输出重定向别的文件或某个命令中。假如要对一个名为 DIRSAVE.TXT 的文件进行排序，可作如下操作：

1. 输入 SORT < DIRSAVE.TXT|MORE。
   DOS 将按字母表顺序对文件内容进行排序，并一行一行地将结果显示出来(这种排序是没有意义的，但从中可以了解 SORT 命令是如何工作的)。

注释： DOS 中的 HELP 文件介绍说不必用重定向给出 SORT 的输入文件，其实不然。用户必须用"<"重定向符给 SORT 提供输入文件。

若不想将结果输出在屏幕上，而是想将其存作文件，可输入命令 SORT < SAVEDIR.TXT > SORTDIR.TXT。

36.3.2.2 SORT 的排序序列

SORT 的排序序列由当前代码页决定(代码页的概念将在第 49 章中作介绍)。表 36.2 列出了 101 标准键盘在英语代码页方式下的排序序列(先第一列，而后第二列，其余列依次类推)。SORT 不区分字母大小写。

表 36.2　SORT 的代码序列(标准英语方式下的代码)

(space)	*	4	>	h	r	\
!	+	5	?	i	s	]
"	,	6	@	j	t	^
#	-	7	a	k	u	-
$	.	8	b	l	v	`
%	/	9	c	m	w	{
&	0	:	d	n	x	|
'	1	;	e	o	y	}
(	2	<	f	p	z	~
)	3	=	g	q	[	

36.3.2.3　SORT 命令的局限

SORT 命令简捷方便,但不幸的是,它还是有许多功能上的局限:

- SORT 仅按字母表顺序排序,若试图用其对数值文件排序,一般是不行的。
- 对操作对象仅能以位置作为标识,不能区分以逗号或制表符分隔开的字段,而后者正是大多数数据库系统广泛采用的方式。
- 每一行被视为单独的一个记录,占据多行的记录无法被标识为一个字段。例如,占据几行的有关姓名及地址的记录,在 SORT 中会被视为不同的几个记录。
- SORT 只能将一行中从头至尾所有的内容作为一个单位进行排序,用户无法设定用于排序的代码长度,也无法指定首选及次之的排序字段。

36.3.2.4　SORT 无法对多重字段进行排序的原因

对上节中最后一项有些用户可能会产生一些疑问,因为看起来可以先对某个字段进行排序,再对另一个字段进行排序,这样便可得到以后者为首选字段的排序结果。实际上,这种技巧只能在某些偶然的情况下奏效。设想有如下的记录序列,其中包括名字及住址信息,希望对姓名进行排序,对姓名相同的情况,再对城市名排序。假设按城市名排序后结果如下:

```
JONES    MARY    45 PINE ST        GRAND RAPIDS,MI
ADAMS    PETER   2314 FIR ST       LAS VEGAS,NV
JONES    MARY    316 LOCUST ST     PITTSBURGH,PA
ADAMS    PETER   16 FOREST RD      SILVER SPRING,MD
ADAMS    PETER   5064 ELM ST       TULSA,OK
```

如果能够用 SORT 对记录中的 1 至 20 列内容排序,结果将是所希望得到的,即对同姓名的记录,其顺序将由城市名而定:

```
ADAMS    PETER   2314 FIR ST       LAS VEGAS,NV
ADAMS    PETER   16 FOREST RD      SILVER SPRING,MD
ADAMS    PETER   5064 ELM ST       TULSA,OK
JONES    MARY    45 PINE ST        GRAND RAPIDS,MI
JONES    MARY    316 LOCUST ST     PITTSBURGH,PA
```

实际上 SORT 操作的对象不仅限于只包含姓名的前 20 列,而是包括所有列。因此,对

姓名相同的记录，顺序将由街道地址而非城市名所决定：

ADAMS	PETER	16 FOREST RD	SILVER SPRING,MD
ADAMS	PETER	2314 FIR ST	LASVEGAS,NV
ADAMS	PETER	5064 ELM ST	TULSA,OK
JONES	MARY	316 LOCUST ST	PITTSBURGH,PA
JONES	MARY	45 PINE ST	GRAND RAPIDS,MI

显而易见，SORT 无法实现上例中所需的操作。

36.3.2.5 关于其他排序软件

如前所述，SORT 仅能完成较为简单的排序，对更复杂的任务，则需使用其他排序软件。几乎所有的文字处理软件、数据库软件以及制表软件均包含排序功能，其中 WordPerfect 功能较全。这些排序程序都可以对各种格式的文件进行排序，其中包括一般的 ASCII 文件。

36.3.3 用 FIND 命令定位文本行

FIND 可用于在文件中查找包含特殊短语的文本行(如图 36.2 所示)。例如，可使用 FIND 查找包含“Norton Utilities”的文本行。

FIND

查找文件中所包含的特定内容。

句法：

FIND [/V] [/C] [/N] [/I] "String" [filespec...]

参数和选项：

"String"	指定需搜索的字符串(引号不可省略)。
filespec	指定需作搜索的文件名。
/V	显示出含有指定字符串的所有行。
/C	显示出包含指定字符串的文本行的总行数。
/N	显示时列出行号。
/I	不区分大小写。

注释：

若不指定文件名，FIND 将从 DOS 标准输入设备中获取输入，即通过键盘或管道。

图 36.2 FIND 命令总结

用 FIND 可实现许多其他功能。通过管道命令与其他命令综合使用，可有选择地显示某些命令的输出。比如，可只列出 DATE 命令输出中的日期行，或选择输出 DIR 命令给出的有关剩余磁盘空间的内容。

36.3.3.1 按内容查找文件

假若想找出一封有关坏软盘的信件，却又记不起存有这封信的文件名，则可用 FIND 来查找包含“compact disc”的文件。假设所需文件可能为 DISCLET、CDRETURN 及 CD-

COMP，则可进行如下查找：

```
C:\>FIND "compact disc" /I DISCLET CDRET CDCOMP
_____ C:\DISCLET
_____ C:\CDRET

I have just received the enclosed compact disc,opened it,and listened to it only once. Unfortunately,
it skips in several places. I am returning it to take advantage of your "Full Replacement" guarantee.
Over the last few years,I have purchased more than 100 compact discs from service and this is the
first problem I have ever encountered.

_____ C:\CDCOMP

It will be some time before I can afford a new CD ROM drive. At the present,I have two high-densi-
ty diskette drives (one of each) and a 40MB hard disk. By the time I can afford a compact disc-style
drive,maybe they'll be able to write as well as read. Wouldn't that be great!
```

FIND 将不包含“compact disc”的行也列出来了，其实是因为对此种格式的文件，一段即是一行。此例中，DISCLET 不含有与“compact disc”有关的内容，另外两个文件均含有这一关键词组。由于加入了/I 选项，因此按不同大小写形式存在的“Compact disc”都将被找出来。

用户可按下述步骤练习一下在 .TXT 文件中查找包含“CD”关键词的有关内容：

1. 转到 DOS 子目录下。
2. 键入 DIR *.TXT。

   DOS 将列出目录中所有的 .TXT 文件，这些文件名将在下面的命令中用到。
3. 键入 FIND /I "CD" 及刚刚列出来的 .TXT 文件名，比如说可能是 FIND /I "CD" README.TXT COUNTRY.TXT NETWORKS.TXT OS2.TXT。

   DOS 将显示出所有包含“CD”关键词的文本行，这一信息将便于用户阅读或打印。

36.3.3.2　在所有文件中查找特定字符串

不幸的是，FIND 命令中不能使用通配符，因而无法方便地让它搜索目录中的全部文本文件。用户必须逐一输入文件名，而且不能超出每个命令行内容必须少于 127 个字符的限制。值得安慰的是，第 37 章中介绍了如何将 FOR 命令与 FIND 结合使用来达到前述的目的。

36.3.3.3　在 DOS 命令的输出中查找特定的行

若通过管道将某一命令的输出转向为 FIND 的输入，用户可选择特定的行作为最后输出，而放弃其余的内容。比如说用户希望在使用 TIME 或 DATE 命令时只显示日期及时间，则可进行如下操作：

1. 键入 ECHO.|TIME|FIND "Current"。

   DOS 将仅显示出当前时间，而忽略其他信息。
2. 键入 ECHO.|DATE|FIND "Current"。

   DOS 将仅显示当前日期，并不输出其他信息。

利用此技巧可实现许多别的应用，例如：

1. 键入 DIR|FIND "free" 可查看当前盘的剩余空间。
2. 键入 CHKDSK|FIND "disk space" 可查看当前盘的总容量。

3. 键入 MEM|FIND "executable"可查看当前最大可执行程序的大小。
4. 键入 SMARTDRV/STATUS|FIND "cache"可查看当前缓冲区的命中率(此命令执行后将显示出两行,其中一行并非我们所需)。
5. 键入 DOSKEY /H|FIND "DIR"可查看记录在 DOSKEY 中包含 DIR 的所有命令行。

提示: 上述一些技巧时常可用到,对这些命令,可将其加入到 DOSKEY 的宏定义中。比如可定义一个名为 ST(System Time)的宏。

```
DOSKEY ST=ECHO. $B TIME $B FIND "Current"
```

36.3.3.4 在目录列表中找出特定文件

尽管 DIR 命令可列出所有满足通配符的文件名,但是否能找出所有 4 月 1 日产生的文件,或是找出文件名中含有"RA"二字母的文件呢? 通过管道将 DIR 的输出重定向到 FIND 的方法可使以上想法得以实现。下述方法可列出包含"ME"的文件名:

1. 转到 DOS 子目录。
2. 键入 DIR|FIND "ME"。
   DOS 将列出 HIMEM. SYS,MEM. EXE,MEMMAKER. EXE 以及 README. TXT 这几个文件名。

注释: 在缺省情况下,FIND 大小写是敏感的,因而在处理 DIR 的输出时,引号中的被搜索内容都应该用大写表示,除非在 DIR 中使用/L 选项或在 FIND 使用/I 选项。

3. 用 DIR *.EXE|FIND "ME"可列出包含"ME"的所有. EXE 文件名。

以上方法在针对多个目录使用时往往会导致一些不便,因为 DIR 命令只列出文件名而不指出详细路径。这一点可通过在 DIR 命令中加入/S 及/B 选项来解决。

这样,用户可用 DIR \ /S /B|FIND "ME"|MORE 来列出当前盘各目录中包含"ME"的文件名。这一操作可能需要花一小段时间才能完成。DOS 先对 DIR 的输出建立一个临时文件,再将其作为 FIND 的输入。

36.4 关于 DOS Shell

在本书第二部分中已介绍过在 Shell 中执行程序及定义程序项的方法。下面将探讨这些方面的高级技巧,并介绍 DOS Shell 的初始化文件 DOSSHELL. INI 的一些特点。

36.4.1 定义程序项

在 Shell 中,用户自定义程序项的优劣将决定 Shell 使用效率的高低。通过向 Shell 中加入常用程序项,用户可方便快捷地执行各种应用程序。程序项的存在同时也方便了初学 DOS 的用户,既易于掌握又灵活多变。

图 36.3 列出了定义程序项时需用到的两个对话框,在第一个中选择 Advanced 按钮即会弹出第二个。在各项中,仅有 Program Title 及 Commands 的内容是必须定义的,其余皆可

依需要设定或忽略。

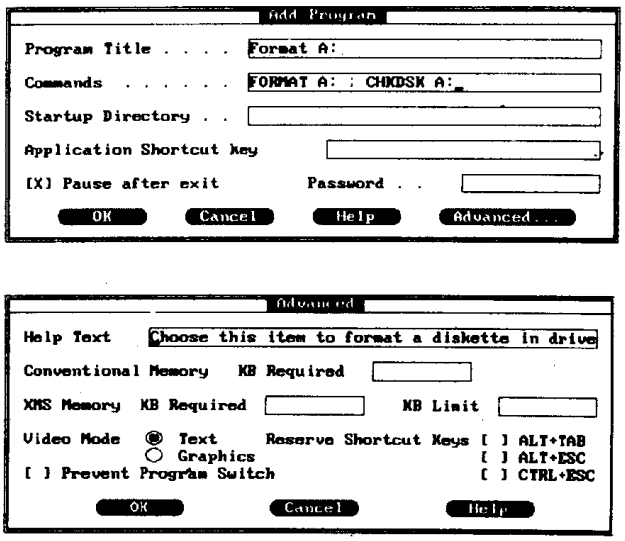

图 36.3　定义程序项

36.4.1.1　在一条命令行中综合使用多条命令

上图的例子中显示了在一个命令域中执行多条命令的方法，只需在两个命令之间以两个空格及一个分号分隔即可。在上例中，执行该程序项后 DOS 将首先对 A 盘作格式化，而后运行 CHKDSK 命令。程序项中允许出现可替代的参数，当然这些参数必须满足命令格式的需要。单个命令及其参数总长度不得超过 127 个字符，而整个域的长度限制在 255 个字符以内。

在 DOS Shell 中可试着做一些练习：

1. 启动 DOS Shell 并激活程序列表。
2. 选择 File 中的 New 选项，弹出 New Program Object 对话框。
3. 选择 Program Item 并选择 OK，弹出 Add Program 对话框。
4. 选定 Program Title 项后输入 Virus Scan Floppy。
5. 在 Command 项中输入 MSAV A：C：/R;MORE <A:\MSAV.RPT。
   该命令将检查 A:盘上是否带有病毒程序，并将带有的病毒清除后将结果存到该盘上。执行结束前将显示一个关于检查结果的报告。
6. 选择 OK 关闭对话框。
   至此，一个新的程序项将出现在程序列表中。
7. 在 A 驱动器中置入一张格式化过的软盘。
8. 执行 Virus Scan Floppy 程序项。
   DOS 将对该软盘作病毒检查并列出结果。

36.4.1.2　内存需求

图 36.3 中的 Advanced 对话框中有几项内容涉及到有关内存需求的问题，这些项在任务切换时非常重要。

Convertional Memory KB Required 项用于设定程序装入时所需的最少内存，缺省状态下为 128KB。当程序装入内存时，此项设置并不影响 DOS 为其分配的内存量，因为 DOS 总是将所有可用内存分配给当前执行的应用程序。但在任务切换时，这项值将决定 DOS 需向磁盘上写入的内存量，从而直接影响到可同时载入的任务总数，以及任务切换所需的时间。若确知某个应用程序仅需 7KB 即可执行，则应在 Conventional Memory KB Required 项中输入 7，以免 DOS 在任务切换时要为其保存 128KB 的内存。

DOS Shell 一开始运行便接管 HIMEM.SYS 管理的全部扩展内存，自行处理有关内存管理的任务。XMS Memory KB Required 及 XMS Memory KB Limit 标识了任务切换时某一应用程序所需的最大及最小扩展内存量，缺省时分别为 0KB 及 384KB。当某一任务被切换为等待状态时，其所占据的扩展内存将仍旧保留，而不是像常规内存那样被暂存到磁盘上。做任务切换时，那部分扩展内存将不能为其他应用程序所使用。因而，为某一程序预留多少扩展内存将直接影响到能够同时装入的程序的多少。出于这一原因，若能确定某一应用程序所需的扩展内存大小，应将其设定在有关 XMS 的两个域中。

显示方式往往也会影响到分配给显示缓冲区的内存大小。一般应使用文本(Text)方式(即使某些应用程序需要在图形方式下运行)，除非在 CGA 上执行一些有特别需要的程序。在图形方式下，显示缓冲区要占据较大的内存。

36.4.1.3 任务切换机制中的一些限制

有些应用程序与任务切换机制相冲突，为了避免这种现象发生，可人为地通过 Advance 对话框设置一些限制。设定 Enable Prevent Program Swith 一项后，用户便无法作任务切换，除非先关闭前一个程序才能开始执行下一个程序。由于某些程序需用一些与任务切换键相同的热键，为执行这些程序且不至在切换时出错，用户必须在 Reserve Shortcut Keys 中设置有关的热键。

36.4.1.4 可替换参数

在 DOS Shell 中，用户可像在 DOSKEY 的宏定义中一样使用可替换的参数。这些参数由%1 至%9 九个符号代表。若 Add Program 对话框的 Commands 域中有可替换的参数存在，DOS 将弹出类似图 36.4 的对话框，以定义程序执行时为方便用户输入参数而弹出的对话框。

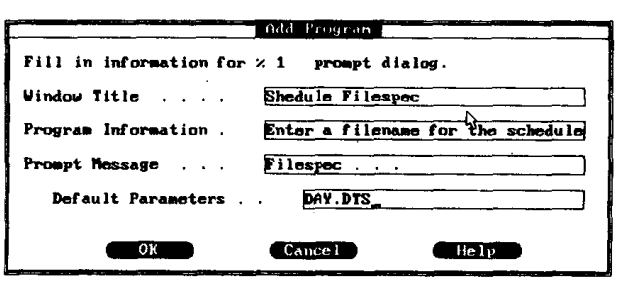

图 36.4 为程序项设定参数的对话框

前三个域中的内容决定对话框中的文本信息，最后一个域标明缺省时的参数值。在%1 至%9 九个符号之外，还有两个特殊的参数符：%F 及%L。在缺省情况下，%F 用程序项开

始执行时文件列表中的当前文件名作为参数，%L 用该程序项以前最后一次执行时的参数作为参数。当然，用户可自行输入新的参数。

可按下述方法试着在 Virus Scan Floppy 程序项中加入参数：

1. 加亮显示程序列表中的 Highlight Virus Scan Floppy 项。
2. 选择 File，Properties 项。

   将弹出 Program Item Properties 对话框，并显示 Virus Scan Floppy 程序项的有关设置。
3. 在 Commands 域中以%1 替换 A:，使其成为

   MSAV %1 /C /R ; MORE < %1\MSAV.RPT
4. 选择 OK。

   DOS Shell 将弹出 Add Program 对话框，要求用户定义设定参数时所需的对话框。
5. 在 Windows Title 框中输入 Virus Scan Floppy。
6. 在 Program Information 框中输入 Enter the name of the drive to be scanned。
7. 在 Prompt Message 框中输入 Drive name...。
8. 在 Default **Parameters** 对话框中输入 A:。
9. 选定 OK。

   对话框关闭，系统回到 DOS Shell 状态。
10. 执行 Virus Scan Floppy 程序项。

    系统将弹出 Virus Scan Floppy 对话框并以 A:作为缺省参数。
11. 选择 OK。

    程序将正常执行。

36.4.2　关于 DOSSHELL.INI

DOSSHELL.INI 对 DOS Shell 的启动起着至关重要的作用。它可以用像 DOS 的 EDIT 一类的文字编辑器来修改。图 36.5 显示了 EGA/VGA 方式下该文件的首部，其中有一段警告信息：不要让编辑器截断那些较长的文本行，否则该文件将失效。还有一些需要注意的是：

- 修改该文件前应先做备份，以防意外的损坏。
- 用文字处理软件编辑后存盘时，应确保其被存作 ASCII 方式的文本文件。
- 切忌修改分隔符，例如[]及{}，除非确信操作无误。

36.4.2.1　Savestate 部分

Savestate 部分显示的是 Shell 的一些重要参数(如图 36.6 所示)。更动其中某些参数(例如 screenmode 和 displayhiddenfiles)，或在菜单中重设某些参数，可明显看到 DOS Shell 执行状态的变化。

36.4.2.2　Programstarter 部分的程序项列表

programstarter 部分包含有关程序列表项的定义。图 36.7 显示了 Manage Daily Schedule 这一程序组的定义，其热键为 Ctrl+D，执行的命令为：

<MS Last> SCHEDULE %1

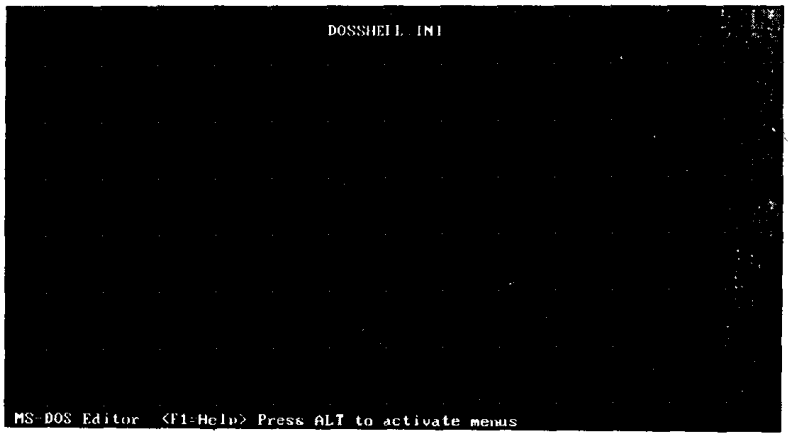

图 36.5 DOSSHELL.INI 的首部

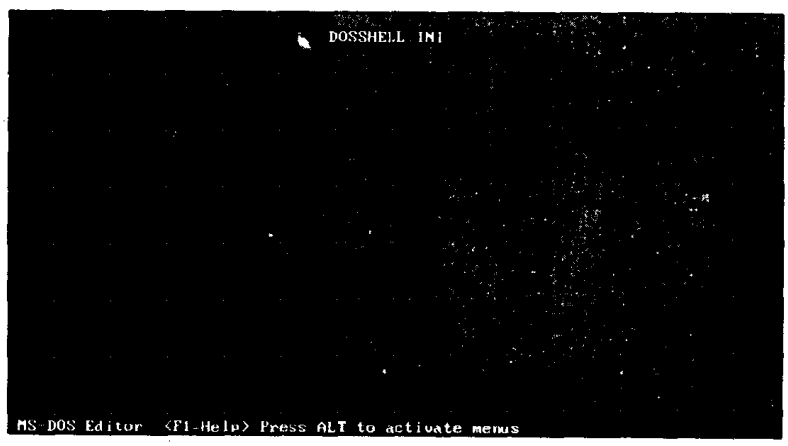

图 36.6 DOS Shell 的参数取值状况

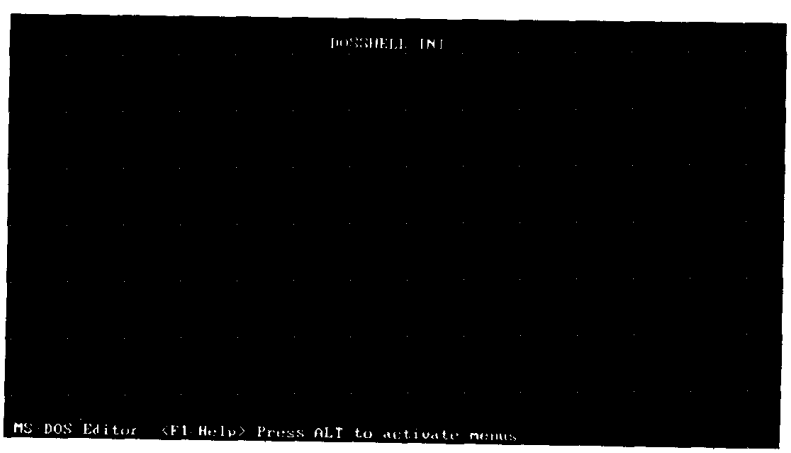

图 36.7 程序组及程序项定义部分

36.4.2.3　DOSSHELL.INI 中为程序设定口令的方法

上图中的例子里口令实际为“A Stitch in Time”。DOSSHELL.INI 仅仅将小写字母依字母顺序后移六位再转换成大写字母，大写字母则保持原样不变。设定口令并非想阻止那些真的想破译密码的人，而只是为了给某些程序项加上执行口令，其作用使用户将在实践中有所体会。

36.4.2.4　颜色

在 DOSSHELL.INI 文件 programstarter 部分的尾部，为 DOS Shell 的各显示部分设定了颜色（见图 36.8）。用户可通过修改该部分内容替换各部分的色彩。设定的方法显而易见，只需试几次就明白。

可用 16 种颜色为红，白，棕，绿，蓝，青，品红，黑以及它们各自对应的亮色（如亮白，大红等等）。在多数显示器上，brightblack 为灰色，brightbrown 为黄色，brightwhite 为白色，而 white 为浅灰色。

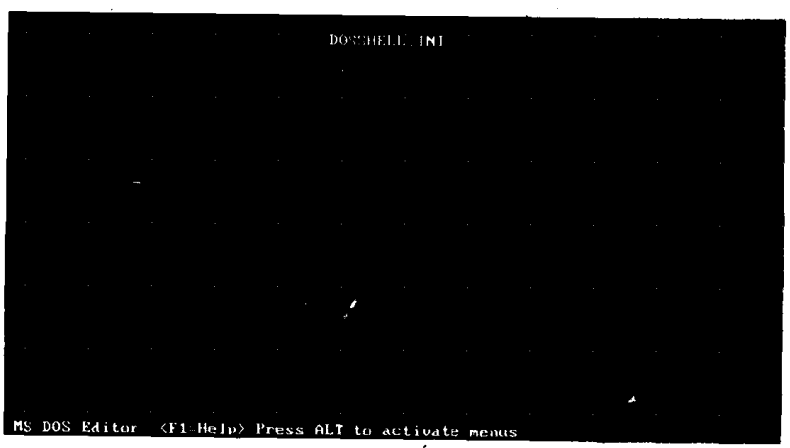

图 36.8　屏幕色彩定义

36.4.2.5　文件关联

DOSSHELL.INI 的最后一部分设置的是文件关联的情况，这些是一般用户不大注意的。DOS 安装完毕时的文件关联状态见图 36.9，这些是由 Microsoft 预先设定好的。用户可通过这部分内容得知当前有效的文件关联状况。

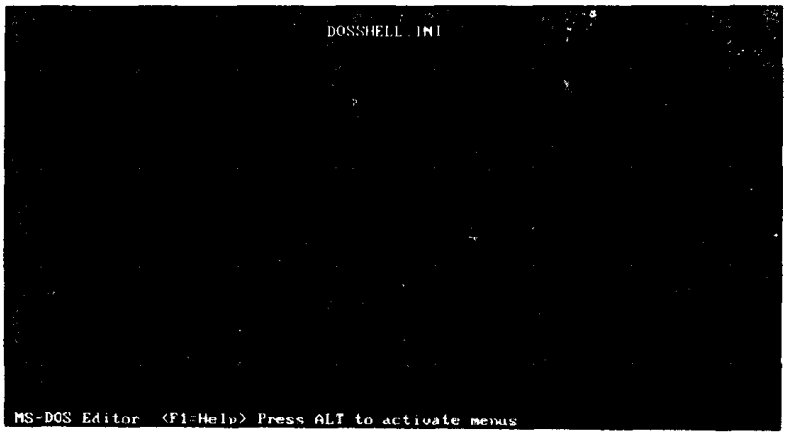

图 36.9 文件关联的定义

36.5 自 测 练 习

本章讲述了可在 DOS 命令提示符下和 DOS Shell 内使用命令的许多技巧。用户可用以下内容自测一下自己运用这些技巧的能力。

1. 创建一个文本文件，它含有硬盘上所有隐含文件和系统文件的清单，还包括系统日期，系统时间，以及在文件首部的注释。
2. 创建一个 DOSKEY 宏，用于显示系统中任何程序占用的内存量（提示：联合使用 FIND 和 MEM/C 命令，并使用一个可替换参数）。
3. 创建一个 DOSSHELL 程序项，由它调用 Vault File 把当前选中的文件移至驱动器 A 中的磁盘上，并把它的名字加入到名为 VLIST. TXT 的文件中（该文件位于启动盘的根目录下），然后对 VLIST. TXT 进行排序。
4. 查阅 DOSSHELL. INI 文件。查阅程序定义和文件关联。
5. 删除本章中创建的而且你不再想保留的程序项和文件。程序项有：Virus Scan Floppy，Vault File。文件有：DIRSAVE. TXT，VLIST. TXT。

第 37 章　批处理程序中的编程逻辑

简介： 除了本书早先介绍的关于批处理程序的基本功能之外，DOS 还具有另外一些可以在批处理程序中提供逻辑判定的功能。

37.1　逻辑编程引论

在一个批处理程序里，可以包含与其他批处理程序相连接的命令，可以读取键盘的输入，可以做出判断，也可以建立循环来重复执行某些指令。对于有经验的编程人员来说，虽然 DOS 所提供的命令已经很熟悉了，但与 BASIC C 所能提供的功能相比，却多少有些笨拙。在这一章里，将介绍逻辑编程的一些基本概念，特别对于从未编过的读者有益。

本章介绍 DOS 的如下功能：

- 与其他的批处理程序连接
- 检测当前状态并决定下一步的行为
- 在批处理程序中建立循环
- 显示菜单并从键盘读取用户的选择

37.2　批处理文件的链接

批处理程序一旦变得很复杂，把一个批处理文件与另外一个连接在一起将是非常方便的。假如现在已经有了一个批处理程序 LATT.BAT，它的作用是显示软驱 A 盘上所有文件的属性。但是，如果要建立一个批处理文件来重写 A 盘，就必须首先把 A 盘上所有的文件以及它们的属性都列出来。由于已经有了一个 LATT.BAT 文件，要完成上述作用，可以把 LATT.BAT 中的命令拷贝到这个新的 BAT 文件里，也可以简单地把它们两者连接起来。

利用连接来实现上述功能具有以下优点：

- 可以节省磁盘空间而不必把一个文件拷贝到另一个文件里。
- 一旦改进了 LATT.BAT 批处理程序，那么所有与它连接的程序都得到了改进。

这种方法的主要缺点是链接本身需要时间，所以链接可能会降低程序执行的速度；并且链接以后对 LATT.BAT 的修改可能会对与之相连的其他程序造成不适当的影响；还有，如果 LATT.BAT 被移动，被重新命名，或者被删除，那么整个链接就不再有效。

在本节的练习里将使用批处理文件 LATT.BAT，下面先建立这个文件：

1. 在 C 根目录下创建一个新的子目录 BATCH。
2. 转到 C:\BATCH 目录下。
3. 创建一个新文件 LATT.BAT 并用一个文本编辑器(如 EDIT)编辑它。

4. 敲入并存储下列内容：

```
@ ECHO OFF
ECHO Press any key to see an attribute list of the files on drive A
PAUSE
ATTRIB A:\*.* /S|MORE
```

5. 在驱动器 A 格式化一张新盘(警告：不要使用含有所需文件的软盘)。
6. 用 XCOPY 从 C 盘拷贝所有 C*.COM 文件到 A 盘，使用/S 选项以包含 C 盘的所有子目录。
7. 键入 LATT 试一下这个批处理程序。

DOS 将把盘里所有文件的属性都列出来。看下一节时，请把盘从驱动器中取出来。

37.2.1 链接和调用

图 37.1 显示了用来连接一个批处理程序的两种不同方法。当用链接(Chaining)时，比如从 Batch A 到 Batch B 的链接，这时程序整个地转到 Batch B，而先前的那个程序 Batch A 完全终止了。当被链接的程序 Batch B 结束时(没有链接其他程序)，整个批处理过程结束，重新回到 DOS 控制状态。相反的，当在 Batch A 里调用(Calling)另一个程序 Batch B 时，只是暂时地转到 Batch B 中，当它结束后，又重新回到 Batch A 中，开始执行调用指令下面的其他指令。

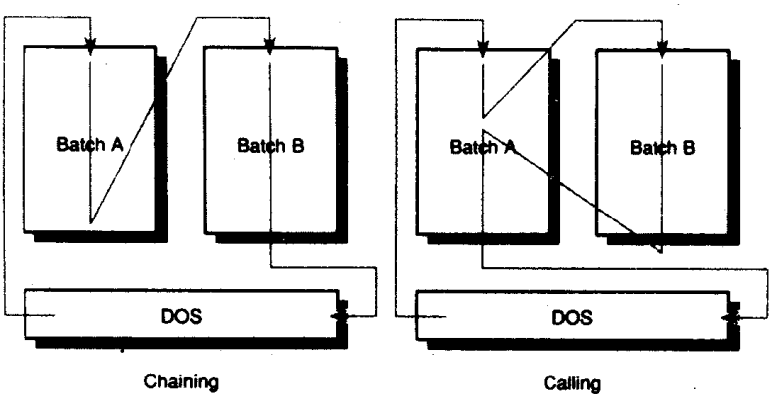

图 37.1 链接和调用其他 BAT 文件

37.2.2 链接

如果在一个批处理程序里包含这样一个命令，它只是用来执行另外一个批处理程序而没有用特殊的指令，此时就实现了链接，而先前那个批处理程序自动地终止了。试一试下面的几个步骤：

1. 确认当前目录为 C:\BATCH。
2. 创建一个新文件 PR.BAT，输入以下内容：

```
@ ECHO OFF
ECHO Linking to LATT.BAT
LATT
```

```
ECHO Returen from the link
```

3. 试执行程序 PR. BAT：

在屏幕上出现以下信息：

```
Linking to LATT. BAT
Press any key to see an attribute list of the files on drive A
Press any key to continue...
[以下是文件的属性列表]
```

注意，屏幕上并未出现信息“Return from the link”，这是由于程序链接到 LATT 以后再也不回到 PR. BAT。

注释： 在批处理程序里执行其他类型的可执行文件如 COM，EXE 时，并没有发生链接。在刚才的这个例子里，ECHO、PAUSE、ATTRIB 都是可执行程序，但并没有发生链接。只有批处理程序之间的连接才发生了如上所述的链接。

37. 2. 3　调用

调用其他批处理程序，用下面的命令格式：

```
CALL batch-command
```

试用调用代替链接：

1. 在文件 PR. BAT 里，在 LATT 前面插入 CALL。
2. 执行 PR。

这次，屏幕上将出现 PR. BAT 程序中的最后一句话：Return from the link.

在 DOS 的命令行方式下执行命令时可以在 CALL 命令里包含任意参数。例如，如果 LATT 被重新改写，而带有一个可替换的命令行参数来表明所选择的驱动器，那么就可以执行下面的命令：

```
CALL LATT A:
```

注释： 如果 CALL 命令是在一个批处理程序中调用另外的批处理程序，则命令中的重定向将被忽略。

注释： 在 DOS Shell 中定义一个程序项时，也可能要使用 CALL 指令。如果在程序项中执行一个批处理程序但没有用 CALL 时，DOS 将转向这个批处理程序并且不再回到程序项。当然，如果命令表中只有这一个批处理文件，或者这个批处理文件是表中的最后一个，不用 CALL 也不会有什么问题。但是，如果不是上面所说的那种情况，那么就应该在程序列表中用 CALL 来执行批处理程序以使这个批处理程序结束时能返回到程序列表中。

37. 2. 4　传递可替换的命令行参数

在连接命令里可以使用可替换的命令行参数；DOS 在执行连接前先完成参数的替换。在下面这个例子里，从调用程序向被调用程序 FMT. BAT 传递了两个参数：

```
CALL FMT %3 TOTMASTER
```

在通常情况下，%3 将被执行这个程序的命令的第三个参数所取代。如果这个参数是 A:，则这个 CALL 命令执行时变成：

```
CALL FMT.BAT A: TOTMASTER
```

这时，被调用的批处理程序 FMT.BAT 就把 A:作为它的第一个参数，而 TOTMASTER 被当作第二个参数。FMT.BAT 本身并未意识到参数 A:来源于调用程序的%3 参数。

可替换的命令行参数并不自动地由连接程序向被连接程序传递。与 DOS 命令行方式下执行一个程序的情况一样，在连接命令里必须包含向被连接程序传递的所有参数，并且这些参数要有正确的顺序和格式。

37.2.5 多重链接

一个批处理程序被连接调用后，仍可以连接调用其他的批处理程序。在嵌套的批处理程序中可以使用的连接调用次数是不受限制的。程序 A 可以链接(chain)程序 B，而程序 B 又可调用(Call)程序 C，程序 C 又可链接程序 D，如此反复，形成一个多重嵌套的批处理程序。如果把多重批处理程序限制在几个嵌套连接的程序系列中，那么对这些程序的管理，修改以及调试都会变得非常容易。

37.2.6 递归调用

批处理程序可以连接自己，这种情况称为递归。当递归调用时，一个批处理程序被反复执行直至其他事件发生而终止。可以在批处理程序中设置判断，条件满足则跳出递归，或者简单地按 Ctrl+Break 键而终止这个程序的执行。下面的练习是建立一个屏幕保护程序，它在屏幕上快速地显示不同的东西。

注释： *屏幕保护程序的作用是防止屏幕总是显示同样的东西。否则，时间一长，就有可能"烧坏"屏幕。*

1. 创建一个 BAT 文件 SS.BAT，在其中输入：

```
@ECHO OFF
ECHO Out to lunch
ECHO Will be back eventually
CALL SS
```

2. 执行 SS.BAT。

   上面两条信息将迅速卷离屏幕。

3. 按 Ctrl+Break 终止这个程序的执行。

   DOS 提问 Terminate batch job (Y/N)?

4. 按 Y 中止 SS.BAT。

37.2.7 中断有链接的批处理程序

按 Ctrl+Break(或 Ctrl+C)可以中止一个嵌套执行的 BAT 程序。此时 DOS 将显示下列信息：

Terminate batch job (Y/N)?

键入 Y 将中止所有的嵌套程序；如果键入 N，则只中断 DOS 正在执行的那个命令。

37.3　BAT 程序中的分枝结构

在一个 BAT 程序中可以根据一些具体的条件判断来决定下一步干什么。比较典型的条件有：

- C:\AUTOEXEC.BAT 是否存在?
- 在这之前的命令是否执行完?
- 用户是否按了 Y 或者 N?
- %1 参数是否是/S?

37.3.1　分枝的种类

BAT 程序里的分枝既可以相当简单，也可以非常复杂(当然，它可能达到真正的程序语言如 BASIC 和 C 等所能达到的复杂程度)。

37.3.1.1　旁路(Bypass)分枝

图 37.2 是一个最简单的分枝，条件判断若为真则执行子程序 X，否则就越过 X。这个简单分枝被称为旁路分枝，因为真和假两种情况中只有一种情况要执行子程序 X，另一种情况则只是越过它进入到下一步。另外，这两种情况最终都将归结到同一地方。

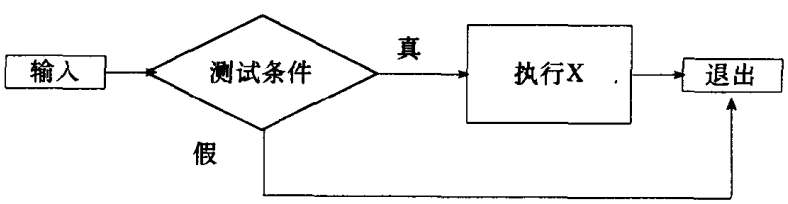

图 37.2　旁路分枝示意图

典型的旁路分枝示例如下：

- 如果 C:\AUTOEXEC.BAT 存在，则更名为 C:\AUTOEXEC.OLD。
- 如果当前命令失败，则给出警告信息。
- 如果用户按 Y，则格式化 A 盘。
- 如果%1 参数是 SORT，则对源文件排序。

37.3.1.2　选择分枝

图 37.3 显示了另一种常见的 BAT 分枝结构。在这个程序里，如果条件为真则执行一个子程序 X，否则就执行另外一个子程序 Y。

典型的选择分枝示例如下：

- 如果 C:\AUTOEXEC.BAT 存在，则把它拷到 A 盘；否则，创建一个新文件 A:\AUTOEXEC.BAT。

- 如果当前的拷贝操作成功，则删掉源文件；否则，显示错误信息并退出。
- 如果用户选择 1，则显示 AUTOEXEC. BAT；否则，显示 CONFIG. SYS。
- 如果%3 参数是/P，则打印目标文件；否则就在屏幕上显示这个文件。

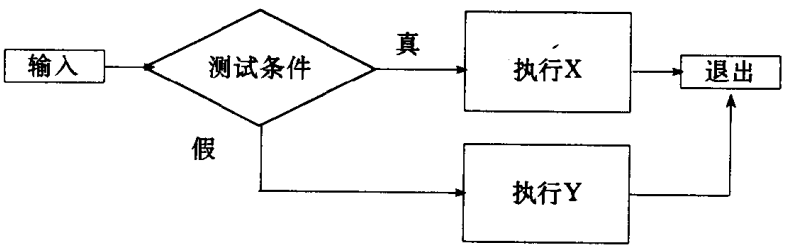

图 37.3 选择分枝示意图

37.4 IF 命令

在批处理程序里用 IF 命令来建立有分枝结构的程序。图 37.4 给出 IF 指令的具体格式。

IF

当(且仅当)条件 IF 为真时执行命令，否则跳过命令。

句法：

IF [NOT] condition command

参数：

condition	用来判断的条件表达式，有以下几种情况：
ERRORLEVEL n	执行程序的返回码不小于 n 时表达式值为真。
EXIST filespec	指定文件存在时表达式为真。
string1==string2	两个字符串相等时表达式为真。
command	表达式为真时所要执行的命令。

注释：

NOT 把条件表达式的判断值(真或假)取反，所以条件表达式为假时命令被执行。

图 37.4 IF 命令总结

37.4.1 程序返回码

程序结束时都将设置一个返回码。返回码的取值范围是 0 到 255，返回码的值表明程序执行是成功了还是没有成功。如果没有成功则返回码的值是错误码。返回码没有确定的含义，它的意义由程序自己设定。返回码 0 总是用来表明程序已成功执行。例如，FORMAT 程序提供的返回码如下：

0 磁盘格式化成功(没有错误)。

3 用户按 Ctrl+Break 或 Ctrl+C 键退出 FORMAT。

4　　发生不能确定的错误，磁盘格式化没有成功。

5　　格式化出现了问题，向用户报告，用户取消了 FORMAT。

不幸的是，并非所有的 DOS 命令都设定了自己的返回码，因此在批处理程序里检查 DOS 命令的返回码可能会不成功。附录 B 中列出了一些 DOS 命令的返回码，未列入表的 DOS 命令没有设定返回码。一般的应用程序都设有返回码，可以在它们的说明文件中查到。

37.4.2 ERRORLEVEL 条件

IF 命令里，ERRORLEVEL 条件检查执行程序的返回码，如果返回码大于或等于 IF 命令里所设定的数值(如 IF ERRORLEVEL 3 中的 3)，则条件表达式的判定值为真。

下面的程序格式化 A 盘，若不成功则显示 Try again with another floppy。

1. 把本章以前练习所用的磁盘插入驱动器 A。
2. 创建一个新文件 FMT.BAT，其内容如下：

```
@ ECHO OFF
FORMAT A: /Q /U /V:""
IF ERRORLEVEL 1 ECHO Try again with another floppy
```

3. 执行 FMT.BAT。
4. FORMAT 提示按任意键继续时，按 Ctrl+Break 键可以中断 FORMAT 命令。
   此时，屏幕出现如下信息：Try again with another floppy。
5. 重新执行 FMT，这次不要中断它。
   由于这次 FORMAT 的返回码是 0，因此屏幕上将没有任何信息。

注释： 由于 FORMAT 未用返回码 1 和 2，所以在这个 IF 命令里可以写成 IF ERRORLEVEL 3。在大多数情况下，必须写成 IF ERRORLEVEL 1，才能正确地区分返回码 0(完全成功)和其他返回码。

在 IF 命令里可以用 NOT，它将把条件的判定值取反。也就是说，返回码比所指定的数值小时条件判定值为真。下面的程序格式化软盘且在成功时显示信息：Remove the floppy from drive A：

```
FORMAT A:
IF NOT ERRORLEVEL 1 ECHO Remove the floppy from drive A:
```

使用 IF ERRORLEVEL 时，很容易被其中的逻辑关系弄得糊里糊涂，特别是在用了 NOT 时。像图 37.5 中的逻辑表可能会有帮助。它可以帮助用户分析一个命令，判断它在何种情况下是真，何种情况下是假，并且可以看到在每种情况下将发生什么。

条件	IF ERRORLEVEL3	
退出码	0,1,2	3,4,...
结果	假	真
执行任务	继续	中断程序执行

条件	IF NOT ERRORLEVEL 2	
退出码	0,1	2,3,...
结果	真	假
执行任务	删除文件	显示信息

图 37.5 IF 指令逻辑表示例

37.4.2.1 返回码的短暂性

返回码的存在时间很短,其他程序的执行将改变返回码的数值,即使一个程序没有设定自己的返回码,它也会把返回码重新置成 0。在 DOS 命令中,有些命令(如 CALL 和 ECHO 等)不改变返回码的数值,但其他的 DOS 命令将改变返回码的数值(特别是那些非批处理程序)。因此,如果需要返回码的话,一定要立即用,否则它很快就会被改掉。

IF 指令不改变返回码的数值。因此,可以在一个命令行里使用多个 IF 指令来判断不同命令(如 FORMAT 或 XCOPY 等)的返回码。

37.4.2.2 用 IF 指令建立旁路分枝

IF 指令可以用来建立一个简单的旁路分枝。如果条件判断为真,则执行一单个指令;相反,若判断结果为假,则越过这个指令直接跳到下一指令。如果需要在条件判断为真时执行更多的指令而不只是一个指令,在通常情况下,可在 IF 指令里 CALL 来调用另一个批处理程序,而在这个批处理程序里包含了所要执行的那些指令。例如:

```
IF ERRORLEVEL 1 CALL BADFORM
```

我们上面所讲述的都是简单旁路分枝的情况,如果要实现选择分枝程序的编程,必须使用 GOTO,下一节里我们将具体介绍。

37.4.2.3 用 GOTO 编写分枝选择程序

要完成这样的工作:最后一个命令成功则调用 GOODFORM,否则就调用 BADFORM。如果像下面这样编程是不行的:

```
IF NOT ERRORLEVEL 1 CALL GOODFORM
CALL BADFORM
```

在上面的程序里,不管判断的结果是真还是假,都将调用 GOTO 指令使 DOS 从程序的当前位置跳到所指定的其他地方(如图 37.6 所示)。可以用 GOTO 语句来完成选择分枝程序的编程。条件判断为真时,就跳到程序的其他部分;相反,若判断结果为假时,则执行 IF 语句的下一条指令。

GOTO

在批处理程序中使用，使 DOS 跳到标号所指定的语句。

句法：

GOTO label

参数：

label　语句标号，指出 DOS 所要跳到的语句，它必须是同一程序里用:label 标明的一个语句标号。

图 37.6　GOTO 批处理命令总结

```
IF NOT ERRORLEVEL 1 GOTO GOODFORM
REM IF DOS reaches the next line,the Format Command
failed for some reason.
...
...(处理 FORMAT 出错的指令)
...
GOTO NEXTSTEP
:GOODFORM
...
...(FORMAT 成功则执行的指令)
...
...
:NEXTSTEP
...
...
...
```

在上面的程序段里，如果 IF 语句的条件表达式为真(即返回码为 0)，DOS 就跳到同一程序里用 GOODFORM 作标号的语句，在 GOODFORM 这个部分有一些语句要执行。最后，程序进行到 NEXTSTEP，开始执行后面的指令，若条件表达式为假，则不执行 GOTO 语句，DOS 执行 IF 语句下面的语句进入处理返回码大于 0 的情况，在这个分枝的最后，由:GOTO NEXTSTEP 命令，使 DOS 越过 GOODFORM 分枝程序段，直接执行后面的指令。:GOTO NEXTSTEP 语句是非常重要的，没有这个语句，会在条件表达式为假的情况下仍然执行 GOODFORM 程序段，从而不符合这个程序的要求。

提示： GOTO 语句必须指向以冒号和语句标号为开头的命令行。对于一个带冒号的命令行语句，DOS 将忽略掉除前面 8 个字符外的其他所有字符。因此，可以在一个带语句标号的命令行里加上一些解释性的文字来说明程序段的逻辑结构(而不会引起副作用)，使它具有注释的作用。应该注意的是，:PRACTICE1 和:PRACTICE2 都会被 DOS 读作:PRACTIC，并且它们都会被认为是可以标识的。GOTO 语句遇到这种情况将转向它首先碰到的那个:PRACTIC。

GOTO 语句还可用来建立多重选择的分枝程序。下面程序段的目的是显示 FORMAT 命令执行后的返回码：

```
FORMAT A:
```

```
IF ERRORLEVEL 5 ECHO Exit code is 5
IF ERRORLEVEL 4 ECHO Exit code is 4
IF ERRORLEVEL 3 ECHO Exit code is 3
IF ERRORLEVEL 0 ECHO Exit code is 0
```

结果当返回码为 4 时，屏幕上将显示：

```
Exit code is 4
Exit code is 3
Exit code is 0
```

由于这些 IF 指令一个接一个，DOS 按顺序执行它们，并不管前一个 IF 语句的判断结果。如同上面的情况所述，IF ERRORLEVEL 5 的判断为假，它后面的信息不会被显示。IF ERRORLEVEL 4 的判断为真，它后面的信息将被显示；同样，IF ERRORLEVEL 3 和 IF ERRORLEVEL 0 的判断都将取真值，它们后面的信息都会被显示。所以，最终出现了上面那个结果。显然，这并不是我们所要的那个结果。

修改 FMT. BAT 程序，让它对 FORMAT 命令的所有可能的返回码分别处理：

1. 修改 FMT. BAT，使其中包含以下命令行：

```
@ ECHO OFF
FORMAT A: /Q /U /V:""
IF ERRORLEVEL 5 GOTO HANDLE 5
IF ERRORLEVEL 4 GOTO HANDLE 4
IF ERRORLEVEL 3 GOTO HANDLE 3

REM If DOS reaches this line,the exit code must be 0
ECHO Exit code is 0
GOTO NEXTSTEP

:HANDLE3 This routine handles an exit code of 3
ECHO Exit code is 3
GOTO NEXTSTEP
:HANDLE4 This routine handles an exit code of 4
ECHO Exit code is 4
GOTO NEXTSTEP

:HANDLE5 This routine handles an exit code of 5
ECHO Exit code is 5

:NEXTSTEP The program ends here
```

2. 执行 FMT. BAT。
3. FORMAT 提示按任意键继续时，按 Ctrl+Break 键可以中断 FORMAT。
   屏幕上将出现信息：Exit code is 3。
4. 再次执行 FMT. BAT，这次不要中断它。
   屏幕上将出现信息：Exit code is 0。

在上面这个程序段中，IF 语句检测返回码必须按照从高到低的顺序，这样才能使它首先碰到的判断为真的情况是返回码与所设定数值相等的情况。否则，就有可能出现错误。

37.4.3 测试文件是否存在

有时，我们在进行下一步工作之前有必要知道一个文件是否存在。例如，很多程序执行

时都要先检查一下 AUTOEXEC.BAT 是否存在，下面就是一例：

1. 创建文件 TEST.BAT，其中包括以下内容：

```
@ ECHO OFF
IF EXIST A:\AUTOEXEC.BAT GOTO PRESENT
ECHO There is no AUTOEXEC.BAT file on drive A
GOTO ENDING

:PRESENT
TYPE A:\AUTOEXEC.BAT|MORE

:ENDING
```

2. 执行 TEST.BAT。

   屏幕上出现信息：There is no AUTOEXEC.BAT file on drive A。

3. 向 A 盘拷贝文件 AUTOEXEC.BAT。

4. 再次执行 TEST.BAT。

   这次屏幕上将打印出 A:\AUTOEXEC.BAT 的内容。

IF [NOT] EXIST 语句也可用来建立一个旁路分枝程序。下面这个语句的作用是，如果 C:\根目录下设有 AUTOEXEC.BAT，则把 A:\AUTOEXEC.BAT 拷到 C:\目录下：

```
IF NOT EXIST C:\AUTOEXEC.BAT COPY A:\AUTOEXEC.BAT C:\
```

提示： 如果要检查当前目录下是否存在子目录 TEMP，用指令 IF EXIST TEMP 是不行的，因为 EXIST 指令不能用来检查一个目录是否存在。但是，可以在子目录名之后用特殊文件名 NUL(任何目录都有这个文件)来检查是否存在这个子目录：

```
IF EXIST TEMP\NUL command
```

37.4.4　字符串比较

初看起来字符串的比较似乎一点用处都没有：IF APPLES==APPLES 很明显是真值，而 IF APPLES==ORANGES 是假值。但是，由于参加比较的字符串可以是可替换的参数，因此就能够用这种方法检查从命令行中输入的参数。

下面这个程序段检查命令行的第一个参数是否是/F。如果是，在向 A 盘拷文件之前先重新格式化 A 盘；如果不是，则拷文件之前不进行格式化：

```
@ ECHO OFF
IF "%1"=="/F" FORMAT A:
COPY C:\MYAPP\*.* A:
```

37.4.5　空字符串的比较

虽然有些情况下可以不用双引号来标识进行比较的两个字符串，但使用引号是一个好习惯。在上面的程序段中，如果没有用引号而且把参数%1 忽略掉了，上面的 IF 语句就变为：

```
IF==/F FORMAT A:
```

条件表达式==/F 是 DOS 所不认识的，这会造成一个系统出错，从而中断程序的执

行，它后面的程序没有执行就夭折了。给两个比较的字符串加上引号就可以避免这种情况的发生。如果用了引号，语句可以被接受：

```
IF "" == " /F" FORMAT A:
```

当然，由于一个空的字符串不可能等于"/F"，因此，FORMAT A:的指令也不会被执行，但不会中止整个程序的执行。

注释： 如果使用了引号，比较的两个参数都必须用。

可以用空字符串来检查一个命令行参数是否是空的。检查%3 参数是否为空用下面的语句：

```
IF "%3" == "" GOTO CONTINUE
```

如果用户未输入第三个参数，DOS 将把上述语句替换成：

```
IF "" == "" GOTO CONTINUE
```

一个空字符串当然与一个空字符串相等，因此条件表达式为真，DOS 执行 GOTO CONTINUE 指令。

编一个小程序，根据用户是否用了 P 作为命令行的第一个参数来决定是打印还是屏幕显示 AUTOEXEC. BAT 文件：

1. 创建一个批处理文件 PD. BAT：

```
@ ECHO OFF
IF "%1" == "P" GOTO PRINT-IT
TYPE C:\AUTOEXEC. BAT|MORE
GOTO ENDING
:PRINT-IT
TYPE C:\AUTOEXEC. BAT>PRN
:ENDING
```

2. 执行 PD. BAT 不带任何参数。
   PD 将在屏幕上显示 AUTOEXEC. BAT 文件。
3. 执行 PD. BAT 带上参数 P：PD P。
   PD 将打印 AUTOEXEC. BAT 文件。

37.4.6 嵌套 IF 语句

在 IF 语句里再使用 IF 指令，就称为 IF 的嵌套。只有当外层的 IF 语句判断为真时才执行内层的 IF 语句。在下面的语句里，当前的返回码的数值既小于 5 又大于或等于 4 时，才执行 GOTO 命令(也就是当返回码等于 4 时)：

```
IF ERRORLEVEL 4 IF NOT ERRORLEVEL 5 GOTO CODE4
```

当 IF NOT ERRORLEVEL 4 语句判断为真时才执行 IF NOT ERRORLEVEL 5 语句，只有当这两个 IF 语句的判断都是真时，GOTO 语句才被执行。

下面的例子里，若%1 参数既不是 Y 又不是 N，则执行 GOTO 命令：

```
IF NOT "%1"=="Y" IF NOT "%1"=="N" GOTO BADINPUT
```

在字符串的比较中必须注意字母大小写的不同。在上面这个语句里，小写的 y 或 n 与大写的 Y 或 N 并不相同，若用户键入的是 y 或 n 则 GOTO 语句将被执行。下面的修改将忽略大小写的区别：

```
IF NOT "%1"=="Y" IF NOT "%1" =="y"
IF NOT "%1"=="N" IF NOT "%1"=="n"
GOTO BADINPUT
```

37.5　循 环 语 句

如果程序段可以被重复地执行若干次，就称为循环。有两类循环——活循环和死循环。一个活的循环设置了出口，一旦条件判断为真就中止循环。死循环则不然，它将被无限期重复执行除非被外在地中止(如按 Ctrl+Break 键或重启动)。大多数的死循环是由于用户的错误造成的，当然有时候也可以利用这种死循环来达到一些有意义的目的。

图 37.7 显示了一个(活)循环语句的逻辑结构。程序执行一个命令(如 FORMAT 命令)然后检查条件表达式是否为真(如检查返回码是否为 0)。如果条件表达式为真，则继续执行循环体内的命令。只有条件表达式为真时，程序才跳出循环，接着执行后面的语句。也就是说，循环体内的命令被反复执行直到条件表达式为真。一个活的循环语句在循环体内执行命令时必然影响到所设置的作为出口的条件，这样才使它下次被检查为真，然后跳出循环。

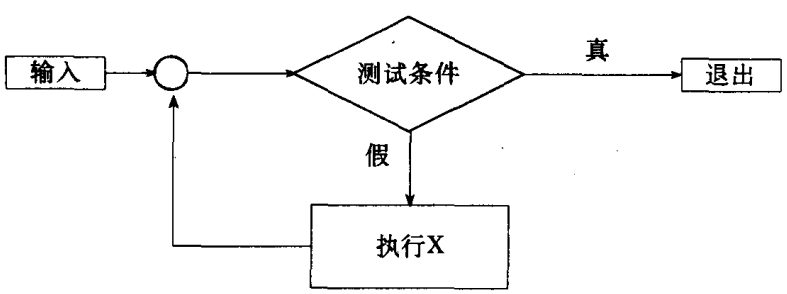

图 37.7　循环语句结构图

在某些循环语句里，需要在进入循环体之前就检查条件是否为真。这样就有可能使循环体内的指令未被执行，条件就满足(判断为真)，从而结束这个循环语句的执行。

37.5.1　循环语句的示例

下面这个程序执行 FORMAT 直到它格式化成功。在通常情况下，这个循环只被执行一次(格式化成功)；如果盘是坏盘或出现了其他什么问题，循环体将被重复执行一次或多次：

1. 创建名为 RF.BAT 的批处理文件：

```
@ ECHO OFF
:FORMAT LOOP
FORMAT A: /Q /O /V:""
IF NOT ERRORLEVEL 1 GOTO ENDING
REM If DOS reaches this line,FORMAT was not successful
```

```
ECHO FORMAT failed for some reason.
ECHO To try again,press any key
ECHO To quit,press Ctrl+Break
PAUSE
GOTO FORMATLOOP
:ENDING
```

2. 执行 RF.BAT。
3. FORMAT 提示按任意键继续时若按 Ctrl+Break 键屏幕上将出现以下信息：

```
ECHO FORMAT failed for some reason.
ECHO To Try again,press any key
ECHO To quit,press Ctrl+Break
```

4. 按任意键继续执行 FORMAT。
5. 重复第 3 步。
6. 按 Ctrl+Break 键中止这个批处理程序的执行。

37.5.2 调用 SHIFT 命令的循环

能够让一个命令重复执行多次有时是很方便的。假如能在一个删除命令里删除好几个没有关联的(比如,与同一通配符匹配),这无疑是一个非常省事的功能。SHIFT 命令可以帮助实现这个功能。

SHIFT 命令不带任何参数,它的作用是使所有的命令行参数依次前移一个位置,比如,原来的%1 参数变成了%0,原来的%2 参数变成了%1,如此等等。如果把 SHIFT 命令用于循环语句,那么每执行一次循环,都将会掉一个命令行参数(最前面的那个,即当前的%0 参数)。

我们来建立可以删除任意多个文件的一个批处理程序：

1. 创建一个批处理文件 MDEL.BAT,它包含以下内容：

```
@ ECHO OFF
:DELLOOP Deletes any number of files.
IF "%1"=="" GOTO ENDING
DEL %1
SHIFT
GOTO DELLOOP
:ENDING
```

循环开始时检查第一个参数是否为空。如果是空的,程序跳到 ENDING 并结束;如果不是空的,DEL 命令将删掉这个参数所标识的那个文件,并且把所有参数向前移一位(这样,下一个参数将成为%1),接着程序回到循环的开头,执行下一次循环。

2. 拷贝下列文件到 A 盘,并修改它们的文件名：

```
C:\DOS\README.TXT(改名为 A:\A.XXX)
C:\DOS\HELP.HLP(改名为 A:\B.XXX)
```

3. 执行命令 MDEL A:\A.XXX A:\B.XXX。

DOS 将把 A 盘的这两个文件删掉。

SHIFT 命令在同一个程序里可以操作几个以上的命令行参数。对大多数的 BAT 程序而言，九个参数已足够了。当然，原则上 SHIFT 命令可操作参数的数目是不受限制的。

37.5.3 FOR 循环

另外一个实现循环的方法是用 FOR 语句(如图 37.8 所示)。刚接触到 FOR 循环语句，也许用户并不喜欢它那笨拙的令人讨厌的格式，可是一旦用户熟悉这种格式，就会发现用 FOR 语句来构成循环具有莫大的益处。FOR 语句的一个最大优点就是能在命令行方式下直接运行；它是唯一能在命令行方式下进行重复操作的命令。

FOR

为可变参数的每次取值重复执行一个命令。

句法：

FOR %X|%%X IN (set) DO command

参数：

%X|%%X	可变参数，必须是非数字的字符。它用在 command 所表示的命令里作为参数。在命令行方式下用%X，而在批处理程序里用%XX。
(set)	可变参数的取值集合。可变参数将取括号内的所有参数。圆括号是必不可少的。
command	要循环执行的 DOS 命令，包括它的参数和选项。

图 37.8　FOR 命令总结

例如，在命令行方式下，可以用 FOR 循环命令来删除任意多的文件：

1. 把下列文件拷到 A 盘，并重新命名：

   C:\DOS\README.TXT(重新命名为 A:\A.XXX)
   C:\DOS\HELP.HLP(重新命名为 A:\B.XXX)

2. 键入命令 FOR %F IN (A:\A.XXX A:\B.XXX) DO DEL %F。
   DOS 首先用 A:\A.XXX 替换 DEL %F 命令中的%F，再执行命令 DEL A:\A.XXX。然后，它用 A:\B.XXX 来替换%F，再执行 DEL A:\B.XXX。

注意，在批处理程序里，必须用两个%来标识变量，即用%%X 而不是%X。否则，DOS 会去掉 X 前面的%从而不能正确执行。

提示： 由于在批处理程序里，DOS 用%作特殊的功用。因此，执行批处理程序时，首先去掉所有单个出现的%符号，即使它不是与数字相连来表示一个可替换参数。所以，如果要想在批处理程序里仍然保留%符号，则必须在前面再加上一个%符号。如果要在批处理程序里使用文件 TEN%ERS，则必须双写%。如要打印这个文件，则用下面的命令：

```
PRINT TEN%%ERS
```

DOS 在执行这个命令前用%代替%%从而能正确执行。

37.5.3.1 FOR 语句中使用通配符

取值集(set)里使用通配的文件标识符时,DOS 针对每一个符合匹配的文件分别执行这个命令。如果有 100 个文件与之匹配,则 DOS 将执行 100 次这个命令。例如,现在有一个批处理程序 STAMP 可以修改单个文件的时间/日期标志,如果要把子目录 MYDOCS 下的所有文件都作上述修改,可以用下面这个命令:

```
C:\> FOR %X IN (C:\MYDOCS\*.*) DO CALL STAMP %X
```

在 FOR 循环语句里必须使用 CALL 来调用批处理程序(即使在命令行方式下也要如此)。每次调用 STAMP 时,符合匹配的文件名作为它的第一个参数(STAMP 程序用%1 来使用这个参数)。

注释: 如果 FOR 循环语句里不是用 CALL 命令来调用一个批处理程序,而是直接执行批处理程序,那么这个循环命令可能只被执行一次就中止了。

37.5.3.2 FOR 语句和 FIND 命令

DOS 中的 FIND 命令不能用带通配符的文件名作为参数多少令人有些遗憾。怎样才能用 FIND 命令对当前目录中的所有文件(或 .DOC 和 .TXT 文件)进行搜索呢?答案是联合使用 FIND 和 FOR 命令。FOR 循环语句把符合(SET)取值集合中描述的文件名作为参数传递给 FIND 命令,这样就可以使 FIND 每次都是以不带通配符的文件名作为参数,从而达到在所有文件中搜寻的目的。

下面这个命令的作用是找出 DOS 子目录中所有含字符串"CD"的 .TXT 文件:

```
FOR %F IN (C:\DS\*.TXT) DO FIND "CD" /I %F.
```

执行上述这类命令,要随时准备使用 Pause(或 Ctrl+S)使输出分页显示。由于 FOR 循环语句中不能使用管道,所以用 MORE 命令来分页显示输出结果是不可能的。

提示: FOR 语句和 IF 语句联合使用时,可以提供一种简单方法来检查各种可能的情况。下面的指令当%1 参数是 Y,y,N,n 时实现跳转:

```
FOR %%X IN (Y,y,N,n) DO IF "%1"=="%%X" GOTO GOODINPUT
```

37.6 CHOICE 命令

许多批处理程序需要用户回答 Yes-No 的简单问题,或者是显示一个菜单,由用户输入所要选择的选项前的代号(一般为字符)。用户通过键盘敲入字符(标识用户的回答或选择),程序再根据用户的选择执行相应的命令。CHOICE 命令可以用来实现这个功能。图 37.9 示出了 CHOICE 命令的命令格式,它在监视器上显示一些信息和提示符,等待用户敲入字符,然后根据这个字符设置返回码的数值。

举例来说,批处理程序里如果出现如下命令:

```
CHOICE Do you want to prepare another disk
```

屏幕上将出现以下信息:

```
Do you want to prepare another disk [Y,N]?
```

如果用户键入 Y 或 y,CHOICE 把返回码设为 1;如果敲入 N 或 n,则设为 2;如果用户按 Ctrl+Break 或 Ctrl+C 键,则返回码为 0。其他输入则产生警告声,CHOICE 将一直等到用户敲入正确的字符(即上述几种情况)。

CHOICE

批处理程序让用户在诸多选项中进行选择。

句法:

CHOICE [/C:list] [/N] [/S] [/T:char,sec] [text]

参数和选项:

/C:list	用户可从键盘上敲入的字符的一个字符表,冒号是任选的。
/N	这个选项使 CHOICE 不显示提示符。
/S	这个选项使 CHOICE 区分大、小写,否则,将忽略键盘上输入字符的大小写。
/T:char,sec	此选项使 CHOICE 确认缺省值 char 之前等待 sec 秒的时间。char 必须是 list 表中的某一个字符,sec 取值范围从 0 到 99。在这个参数中不要使用空格。
text	提示符前所要显示的文本信息。

图 37.9　CHOICE 命令总结

下面的程序格式化 A 盘并向它拷贝一些文件,如果原盘里有文件存在则提醒用户注意:

1. 创建批处理文件 FD.BAT,其内容是:

```
@ECHO OFF
IF NOT EXIST A:\*.* GOTO FORMAT
ECHO The disk contains these files:
DIR A:\ /S /B /P
CHOICE Do you want to format the disk?
IF ERRORLEVEL 2 GOTO FORMAT
IF ERRORLEVEL 1 GOTO NEXTDISK

:FORMAT
FORMAT A: /Q /V:""
XCOPY C:\DOS\README.TXT A:
GOTO ENDING

:NEXTDISK
ECHO Try again with another disk

:ENDING
```

2. 确认 A 驱动器里的软盘是本章前面章节练习所用的软盘。
3. 执行 FD.BAT。

   FD 格式化 A 盘,并向 A 盘拷贝文件 README.TXT。
4. 再一次执行 FD.BAT。

   这次,FD 将警告说软盘中有文件存在并问是否要格式化。
5. 键入 N。

屏幕显示:Try again with another disk。

37.6.1 CHOICE 命令的参数

使用/C:list 定义一个供选择的字符表代替缺省时的 Y 和 N。例如,如果要求用户通过键入 A,B 或 C 来选择一个菜单的选项(菜单可以用 ECHO 命令来显示),用选项/C:ABC 来完成。如果这个命令同时包含选项/S,则小写的 a,b 和 c 不会被接受。执行这个命令,CHOICE 将在屏幕显示[A,B,C]?,如果用户键入 A,则返回码被设为 1;如果键入 B,返回码则设为 2;键入 C,返回码则设为 3。

如果不用参数 text,则 CHOICE 除显示提示符之外不再显示其他信息(当用 ECHO 已经显示了一个菜单以后,就可以这样做)。实际上,还可以用/N 选项来隐藏提示符的显示。

/T 参数,使 CHOICE 在用户指定时间之后确认一缺省的选择。如果在 10 秒之内用户仍未敲入选择的情况下缺省地选 A,则可用/T 参数来完成/T:A,10。

37.7 用 DOSKEY 宏指令替换 DOS 命令

DOSKEY 宏指令的优先级高于其他任何程序,包括内部命令(.COM 和 .EXE)和批处理程序。可以利用 DOSKEY 宏指令的这个特性来操纵 DOS 命令。例如,如果想要在拷贝文件时总是用 XCOPY 而不是用 COPY,则可以建立一个 COPY 宏指令来取代 DOS 中正常的 COPY 命令,使用户每次执行 COPY 命令时,就相当于执行 XCOPY 命令:

```
DOSKEY COPY=XCOPY $ *
```

注释: DOSKEY 宏指令定义中的 $ * 参数代表从命令行中输入的所有参数。因此,定义完上面的宏指令之后,所有 COPY 命令使用的参数和选项都将被传给 XCOPY 命令作为参数。

如果只是想使某些命令失效以免没有经验的人误用,可以简单地把它替换为一条信息。下面就是这样一个例子:

```
DOSKEY FDISK=ECHO Please don't use this command; call J. North on extension 457.
```

这些宏指令并不能完全使它们所替代的那些命令失效。如果在 COPY 和 FDISK 命令前加上空格,仍然可以执行它们(与没有定义宏时一样)。这是因为 DOSKEY 忽略那些以空格开头的 DOS 指令。

提示: BAT 批处理程序比 DOSKEY 宏指令灵活得多,也许用户更希望用批处理程序来执行 DOS 命令。但是批处理程序的优先级低于 DOS 的内部命令(.COM 和 .EXE),如何能达到上述目的呢?在这里,可以用 DOSKEY 宏定义来执行这个 DOS 命令,使每次执行它时调用另一个批处理程序。例如,下面这个宏定义将重新定向 FORMAT 命令,以指向一批处理程序 FORMHELP.BAT 来指导没有经验的用户使用 FORMAT 命令:

```
DOSKEY FORMAT=FORMHELP.BAT
```

37.8　软盘上的批处理程序

DOS 执行批处理程序时，一次读一个命令并执行这个命令，当它执行完这个命令以后就回到这个批处理文件里，读取下一条命令，如此反复。因此，当试图执行一个软盘上的批处理程序时，情况可能会变得很复杂。如果在一个命令的执行过程中取走了程序所在的软盘而插入了其他软盘，命令执行完之后，DOS 将要求重新插入那张软盘以继续执行那个批处理文件。屏幕上将会出现：

```
Insert disk with batch file
Press any key to continue...
```

即使批处理程序中已没有命令要执行，DOS 仍要显示这些信息，因为它并不能确认所要执行的批处理文件已经空了，直到它从这个文件中读出指令而发现了文件末尾的标志。

如果只是移走了包含有正在执行的批处理文件的那个软盘而并没有插入其他盘，则屏幕上将会显示如下信息(当 DOS 读盘而发现驱动器为空时总是出现这个信息)：

```
Not ready reading drive A
Abort,Retry, Fail?
```

看到这个信息，插入有批处理程序的那张软盘并按 R，就可以继续执行该批处理程序了。

37.9　批处理程序可能被破坏

批处理程序可以从软盘上启动，(不论你知道或者怀疑这件事)，让我们来看看，如果软盘上的批处理程序试图处理同一张软盘时将会 发生什么情况。在批处理程序里，有些指令可能会从当前的指令序列中跳转到程序的其他地方去执行，(参看图 37.6)。当条件表达式的值为真时程序跳到其他地方；只有当值为假时，程序才执行 IF 语句后的命令：

```
IF NOT ERRORLEVEL 1 GOTO GOODFORM
REM If DOS reaches the next line,the FORMAT command
failed for some reasom
...
...(处理 FORMAT 失误时的命令)
...
GOTO NEXTSTEP
<MS Last>:GOODFORM
...
...(处理 FORMAT 成功时的命令)
...
<MS Last>:NEXTSTEP
...
...
...
```

在上面的程序段中，如果条件表达式为真，表明返回码为 0(即格式化 A 盘成功)，这就有可能破坏这个 BAT 文件。因此，有必要在批处理程序中提供警告信息使用户有机会换出

原来的 A 盘或取消这个程序。

预览： 第 43 章“DOS 及其运行环境”将介绍一些功能使用户执行批处理程序时，每次只执行一行命令。这样可以使用户找到程序中出错的命令或程序段。

37.10 自 测 练 习

完成下列编程：

1. 写一个消除软盘碎片的程序(使软盘上的剩余空间不再是分散的)。程序必须先列出软盘的文件并确认用户是否愿意继续下去(提示:把软盘上的所有文件都拷到一个临时的子目录下(不要忘了隐含文件和系统文件),重新格式化软盘,再把所有文件都拷回来,最后把那个临时的子目录删掉)。
2. 编写一程序,用当前的时间和时期更新软盘上所有文件上的时间和日期标志。(提示,用 FOR 循环对每一个文件执行 COPY 命令,不要忘记在 COPY 命令里包含/B 选项以免文件被错误地截去了一部分。先用一些不重要的文件来检测和调试程序)。当确认程序能正确工作而且没有截尾误操作之后,把它用于其他任何目录下的文件,用目录的路径名作为命令的参数。还应该注意的是,不要把这个程序用于需要保留时间/日期标志的文件。
3. 编写一个程序,显示刚执行完的指令的返回码,任何其他程序运行完之后,运行你的程序能显示它的返回码。
4. 删除本章建立的而且你不再需要保存的文件。

第 38 章　启动 DOS

简介： DOS 6 在启动文件——CONFIG.SYS 和 AUTOEXEC.BAT 中引入了一些新特征，使得用户可以对启动流程进行更多的控制。

38.1　控制启动流程

在过去的 DOS 版本中常常因为用户错误地修改 CONFIG.SYS 和 AUTOEXEC.BAT 使系统无法正常工作，因而不得不用软盘启动。但在 DOS 6 下，用户仍可从硬盘启动，执行启动文件中的可用部分，而跳过错误部分。这使得运行和调试更为容易。

DOS 6 的另一项新功能是在 CONFIG.SYS 中可以有几种不同的系统配置，在启动时用户可以根据需要选择其中之一。如果为运行不同的应用程序而需要不同的系统配置，或者是若干用户共用一台计算机而每人对系统有不同要求，那么现在启动时选择不同配置变得更加容易。

本章介绍如何利用 DOS 6 的新特征，内容如下：

- 启动时完全跳过启动文件
- 设置 CONFIG.SYS 中的某些命令为可选的
- 设置 CONFIG.SYS 使用户可以通过菜单选择几种不同配置中的一个

38.2　跳过启动文件

如何在启动时跳过所有启动文件

若要跳过 CONFIG.SYS 和 AUTOEXEC.BAT：

1. 启动或重启动。
2. 当看到提示 Starting MS-DOS... 出现时，按下 F5 键。

若要完全跳过 CONFIG.SYS、AUTOEXEC.BAT 和 DRVSPACE.BIN 或 DBLSPACE.BIN：

1. 启动或重启动。
2. 当看到提示 Starting MS-DOS... 出现时，按下 Ctrl+F5 键。

DOS 将显示如下信息：

You pressed CTRL＋F5 or CTRL＋F8 to bypass DRVSPACE.BIN. Therefore, none of your compressed drives are available.
MS-DOS is bypassing your CONFIG.SYS and AUTOEXEC.BAT files.

DOS 启动时将不执行这些文件。用户仍然可以使用 DOS 内部命令(如 COPY),但是不能访问任何压缩盘上的程序。如果 DOS 目录位于压缩盘上,那么类似 EDIT 之类的外部命令也无法使用。另外,通常情况下可以装入的驱动程序和内存常驻程序也不存在。

每次执行 CONFIG.SYS 和 AUTOEXEC.BAT 中的一行命令:

1. 启动或重启动。
2. 当看到提示 Starting MS-DOS... 出现时,按下 F8 键。
   DOS 依次显示 CONFIG.SYS 的每一行命令并跟随提示[Y,N]?。(如果 CONFIG.SYS 中含有启动菜单,首先显示菜单)。
3. 对于 CONFIG.SYS 的每一行,输入 Y 执行该行,输入 N 则跳过该行。
   执行过 CONFIG.SYS 后,DOS 会询问 Process AUTOEXEC.BAT[Y,N]?。

注释: 用户可在任何时候按 Esc 键执行 CONFIG.SYS 和 AUTOEXEC.BAT 中剩余的命令。在任何时候按 F5 键可以跳过启动文件中的剩余命令。

4. 输入 Y 则每次执行 AUTOEXEC.BAT 中的一行命令,输入 N 则跳过整个 AUTOEXEC.BAT。
   如果输入 Y,DOS 每次显示 AUTOEXEC.BAT 的一行并提示[Y,N]?。
5. 输入 Y 每次执行一行命令,输入 N 则跳过一行命令。

跳过 DRVSPACE.BIN 或 DBLSPACE.BIN 并每次执行 CONFIG.SYS 和 AUTOEXEC.BAT 中的一行命令。

1. 启动或重启动。
2. 当看到提示 Starting MS-DOS... 出现时,按下 Ctrl+F8 键。

   因为跳过了文件 DRVSPACE.BIN 或 DBLSPACE.BIN,所以不能装入压缩盘。如果 CONFIG.SYS 和 AUTOEXEC.BAT 位于压缩盘上,DOS 将无法执行它们。DOS 会询问是否执行 AUTOEXEC.BAT;因没有 AUTOEXEC.BAT 文件,若回答 Y,则 DATE 和 TIME 命令将被执行。
   如果 CONFIG.SYS 和 AUTOEXEC.BAT 位于非压缩引导盘上,它们将按前面所说的过程逐行执行。

在本书前面的第 21 章已经讲过如何在启动时按 F8 键逐行执行 CONFIG.SYS 和 AUTOEXEC.BAT,跳过用户认为存在问题的命令行。这一方法可以帮助用户分离出错误的命令行。但如果仍然行不通,就可能需要完全跳过 CONFIG.SYS 和 AUTOEXEC.BAT,启动最基本的系统,恢复原来的启动文件,然后重新启动。

当启动 DOS 6 时,信息 Starting MS-DOS... 会在显示器上停留几秒钟。要跳过 CONFIG.SYS 和 AUTOEXEC.BAT,按下并放开 F5 键或者按住一个 Shift 键不放。屏幕上显示如下信息:

MS-DOS is bypassing your CONFIG.SYS and AUTOEXEC.BAT files.

完成这一步之后,如果 COMMAND.COM 不在启动盘的根目录下,DOS 就找不到它。若是如此,DOS 会询问到哪里寻找命令解释器。输入完整的文件名,包括驱动器和路径名,如 C:\DOS\COMMAND.COM。

现在试一下跳过启动文件:

1. 退出正在运行的程序并回到最初命令解释器的提示符下。
2. 按下 Ctrl+Alt+Delete 键重启动。
3. 当看到提示 Starting MS-DOS...时，按下 F5 键。
   屏幕显示 MS-DOS is bypassing your CONFIG. SYS and AUTOEXEC. BAT files。
4. 再次重启动回到正常状态。(这一次不要跳过启动文件)。

如果系统在按 F5 键之后正确启动，命令查找路径中的唯一目录是 C:\DOS(这样用户就可以用 DOS 实用程序如 EDIT 来修改启动文件)。需要安装设备驱动程序的设备(如鼠标)不能工作。因为没有内存管理器，程序也不能使用扩展或扩充内存。最坏的情况下，如果用户为读写硬盘而必须从 CONFIG. SYS 中装个某个实用程序——比如一个第三方软件商编写的不能作为系统核心安装的硬盘管理或压缩程序——那么将无法访问硬盘，当然也不能访问 DOS 目录。但是仍然可以使用 COPY 命令，如果在修改前保存了 CONFIG. SYS 和 AUTOEXEC. BAT，就可以恢复它们。例如，如果把这两个文件保存为 CONFIG. SAV 和 AUTOEXEC. SAV，命令 COPY CONFIG. SAV CONFIG. SYS 和 COPY AUTOEXEC. SAV AUTOEXEC. BAT 将恢复以前的启动文件，然后就可以成功地重新启动。

如果用户使用了 DrvSpace 或 DblSpace，那么压缩软件随 DOS 内核一同装入。按 F5 键不能跳过它，所以仍可访问压缩盘。然而，如果用户怀疑是压缩软件引发了问题，可以在启动后看见提示 Starting MS-DOS...时按 Ctrl+F5 或 Ctrl+F8 键。按 Ctrl+F5 键跳过 DrvSpace 或 DblSpace 以及两个启动文件。按 Ctrl+F8 键跳过 DrvSpace 或 DblSpace 并逐行执行 CONFIG. SYS 和 AUTOEXEC. BAT 中的命令(像按 F8 键一样，随后按 Esc 键可以不带提示执行启动文件的剩余部分，或按 F5 键跳过剩余部分)。

提示： DOS 在显示 Starting MS-DOS...时会暂停两秒钟的时间，如果不喜欢或确实不需要这一暂停，可以在 CONFIG. SYS 中插入命令 SWITCHES /F 消除它。如果需要为其他缺乏经验的用户设定启动文件并且不想让它们被跳过，可以在 CONFIG. SYS 中加入 SWITCH /N 命令。但这并不能阻止用户按 Ctrl+F5 或 Ctrl+F8 键来跳过 DoubleSpace 或 DriveSpace 和启动文件。使用命令 DBLSPACE /SWITCHES=N 或 DRVSPACE /SWITCHES=N 禁止 Ctrl+F5 和 Ctrl+F8。

38.2.1　带确认执行 CONFIG. SYS 命令

可以将一些 CONFIG. SYS 命令设置成可选的，这样每次启动时 DOS 会提示用户确认这些命令是否执行。在命令名字的后面紧跟一个问号(?)就可以做到这一点。比如，可以将如下命令加入 CONFIG. SYS 中以建立一个虚拟盘，并由用户来决定是否安装：

```
DEVICE? =C:\DOS\RAMDRIVE. SYS
```

如果 MEMMAKER 用/L 和/S 选项修改了该命令，问号仍保留在 DEVICEHIGH 后面：

```
DEVICEHIGH? /L:1,8902 /S=C:\DOS\RAMDRIVE. SYS
```

38.3 使用系统设置语句块

DOS 6 为 CONFIG.SYS 增加了一套新命令，使用户可以显示一个菜单，从键盘输入选择，然后根据所选菜单项执行一段命令。设想一种典型情况，用户有一个程序需要扩充(expanded)内存。用户并不打算每次启动时都借用扩展(extended)内存，除非要使用那个程序。启动时设置 CONFIG.SYS，屏幕上显示：

```
MS-DOS 6.22 Startup Menu
========================

   1. Set up expanded memory
   2. Don't set up expanded memory

Enter a choice:2          Time remaining: 05
F5=Bypass startup files F8=Confirm each line of CONFIG.SYS and AUTOEXEC.BAT [N]
```

如果选择第一项，CONFIG.SYS 借用一些扩展内存生成扩充(expanded)内存。如果选择第二项，不产生扩充(expanded)内存。如果两项都不选，可能是用户开机后走开了，那么当剩余时间计数到 0 时缺省选择项 2 被选中。

图 38.1 列出了产生前文菜单的 CONFIG.SYS 文件。首先注意到文件被分成若干块，叫做语句块，它以方括号括起来的名字开头，例如[MENU]和[COMMON]。

```
[MENU]
MENUITEM=EXPAND,Set up expanded memory
MENUITEM=NORMAL,Don't set up expanded memory
MENUDEFAULT=NORMAL,5

[COMMON]
DEVICE=C:\DOS\HIMEM.SYS
DOS=HIGH
BUFFERS=10,0
FILES=30

LASTDRIVE=E
FCBS=4,0

[EXPAND]
DEVICE=C:\DOS\EMM386.EXE RAM

[NORMAL]
DEVICE=C:\DOS\EMM386.EXE NOEMS

[COMMON]
DOS=UMB
DEVICEHIGH /L:1,12048=C:\DOS\SETVER.EXE
STACKS=9,256
SHELL=C:\DOS\COMMAND.COM C:\DOS\ /P
DEVICEHIGH=C:\DOS\DRVSPACE.SYS /MOVE
```

图 38.1 CONFIG.SYS 的多重设置

38.3.1　系统设置语句块

[MENU]语句块包含生成菜单的命令。[COMMON]语句块包含的命令无论选中哪一项菜单都将被执行。当选中菜单的第一项时,[EXPAND]块的命令得到执行;其中的 EMM386 命令用 RAM 选项设定高端(UPPER)内存和扩充(EXPANDED)内存。选中菜单第二项时,[NORMAL]语句块的命令被执行;其中的 EMM386 命令用了 NOEMS 选项而只设定高端(UPPER)内存。

语句块的顺序很重要,第一个[COMMON]块安装了 HIMEM.SYS。在[EXPAND]或[NORMAL]块装入 EMM386 前必须做这一步。然后,第二个[COMMON]块把一些驱动程序装入高端内存;这些命令只能在装入 EMM386 后执行。

38.3.2　定义菜单

要定义一个菜单,必须以[MENU]为标题生成一个语句块。对于菜单中的每一项,按如下格式加入命令:

MENUITTEM=blockname[,menu-text]

blockname 标志了当这一菜单项被选中时要执行的命令块。blockname 最多可达 70 个字符长,并可包含除下列字符外的任何可打印字符:

空格,\/;=[]

menu-text 是菜单中显示的文本,最多有 70 个字符。如果不定义菜单文本,将显示块名 blockname。

如果用户在设定的时间内不做选择,那么由 MENUDEFAULT 命令确定一个缺省选择。命令格式如下:

MENUDEFAULT=blockname[timeout]

blockname 是缺省执行的语句块。timeout 是选择缺省项前系统等待时间的秒数,范围从 0 到 90。如果忽略了 timeout 项,DOS 将一直等待。当用户没有选择菜单而直接按下回车键时,缺省项也被选中。

如果想试一下生成用户自己的启动菜单,按照以下步骤:

1. 把 CONFIG.SYS 拷贝成 CONFIG.SAV。
2. 打开 CONFIG.SYS 并编辑之。
3. 在文件开头处插入下面的命令。

```
[MENU]
MENUITEM=Peter,Peter(default)
MENUITEM=Judi
MENUITEM=Scott
MENUDEFALUT=Peter,5
[COMMON]
```

4. 在文件结尾加入下面命令行:

[PETER]

[JUDI]

[SCOTT]

注释： 这个例子显示一个菜单，但实际上三个选择并没有什么不同。

5. 保存文件并退出编辑器。
6. 退出所有程序(如 DOS Shell)回到第一级命令解释器下。
7. 重新启动计算机。

   DOS 显示菜单并暂停。
8. 按数字 2 选择 Judi 项。

   DOS 完成启动过程。
9. 再次重启动。

   DOS 显示菜单并暂停。
10. 不做选择，等到计时结束。

    5 秒钟后，DOS 选择缺省项完成启动。
11. 把 CONFIG.SAV 拷贝成 CONFIG.SYS，恢复原来的 CONFIG.SYS。

38.3.3 子菜单

再设想另一种普遍情况。三个用户共用一台计算机。Peter 要处理许多国际事务，因而需要设定键盘、显示器和打印机以配合各种文字。Scott 做一些多媒体方面的工作，要装入多媒体设备的驱动程序。Judi 有时做多媒体工作，有时要用到数据库，这一工作在有 RAM 虚拟盘时可提高速度，有时这两样工作都要做。

Judi 所有可能的选择可以不在主菜单中选择，而是在选择 Judi 的系统设置后再显示第二个菜单，即称为“子菜单”。要定义子菜单，在主菜单中以如下格式插入命令：

```
SUBMENU=blockname[,text]
```

blockname 指向另一个为这一子菜单定义菜单项的语句块。

修改后的 CONFIG.SYS 可能如下：

```
[MENU]
MENUITEM=Peter,Peter (default)
SUBMENU=Judi
MENUITEM=Scott
MENUDEFAULT=Peter,5

[COMMON]
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS
DOS=HIGH
BUFFERS=10,0
FILES=30
LASTDRIVE=E
FCBS=4,0
DOS=UMB
DEVICEHIGH /L:1,12048 =C:\DOS\SETVER.EXE
```

```
STACKS=9,256
SHELL=C:\DOS\COMMAND.COM C:\DOS\ /P
DEVICEHIGH=C:\DOS\DRVSPACE.SYS /MOVE

[PETER]
COUNTRY=351,860,C:\DOS\COUNTRY.SYS
DEVICEHIGH=C:\DOS\DISPLAY.SYS CON=(EGA,437)

[JUDI]
MENUITEM=Multimedia
MENUITEM=Database
MENUITEM=Both
MENUDEFAULT=Database,5

[SCOTT]
DEVICEHIGH=C:\DEV\MTMCDS /D:MSCD001 /P:300 /A:0
DEVICEHIGH=C:\DEV\MTMSBS

[MULTIMEDIA]
INCLUDE SCOTT

[DATABASE]
DEVICEHIGH=C:\DOS\RAMDRIVE.SYS /E

[BOTH]
INCLUDE SCOTT
INCLUDE DATABASE

[COMMON]
```

在[MENU]块中，注意到第二个选项是 SUBMENU 命令而不是 MENUITEM。[PETER]块中的命令用来设定国际字符。这些命令将在第 49 章中解释。

[SCOTT]块为多媒体硬件设备安装驱动程序，这些都不是 DOS 命令。它们必须与 PC 上安装的硬件相配。[JUDI]块的 MENUITEM 命令生成一个子菜单。对于子菜单的每一项都有单独的语句块。

[MULTIMEDIA]块中用到了一个以前没有见过的命令 INCLUDE。"INCLUDE 块名"使得指定块作为本块的一部分来执行。在本例中，INCLUDE 命令使[SCOTT]中的所有命令被包含进来。也可以在编辑时简单地把命令拷贝到本语句块中，但本例的做法使以后修改容易一些，因为只需在一处改动它。[DATABASE]块中的命令建立 RAM 虚拟盘。最后，[BOTH]块中包含了多媒体和 RAM 盘的命令。

文件最后的[COMMON]块并不是错误。即使它是空的，这也是一个很好的想法。这样，在用户加入新命令到 CONFIG.SYS 中时，新语句在[COMMON]块中而不是最后一个可选择的语句块。

38.3.4　在菜单中加入颜色

在定义菜单或子菜单的同时，还可以定义文本和屏幕背景的颜色。MENUCOLOR 命令为菜单屏定义颜色：

```
MENUCOLOR=text[,background]
```

用于定义颜色的代码见表 38.1。

表 38.1 颜色代码

代码	颜色	代码	颜色
1	黑	9	灰
2	蓝	10	亮蓝
3	绿	11	亮绿
4	青	12	亮青
5	红	13	亮红
6	洋红	14	亮洋红
7	棕	15	黄
8	白	16	亮白

缺省的背景色是黑色。在白色背景上显示亮蓝色文本，可在[MENU]或子菜单块中插入下面命令：

```
MENUCOLOR=10,8
```

预告： 本书第 43 章将说明如何在 AUTOEXEC.BAT 中处理不同的系统设置。

38.4 自测练习

本章介绍如何在启动期间跳过启动文件，以及如何在 CONFIG.SYS 中使用多重设置。以下内容帮助用户独立实践这些技术。

1. 重启动并跳过启动文件。
2. 修改 CONFIG.SYS 使之在蓝色背景上显示白色菜单。菜单中至少包括三项，其中一个菜单引出一个子菜单(但不要改变系统设置)。
3. 恢复以前的 CONFIG.SYS。

第39章 目录管理

简介： *DOS提供了一些高级的目录技术，使用户具有更强的管理目录结构的能力。*

39.1 高级目录管理技术

用户已经学习了一些基本的目录管理技术，但还有一些更高级的技术，可以使用户管理更复杂的目录结构。这些技术中的一些可以节省时间，例如删除一个子目录中的文件而不删除其本身。还有一些技术用以解决一些特殊问题，例如一个程序只能在某个驱动器的根目录工作，就可以使硬盘的目录看起来像另一个驱动器的根目录。在这一章，用户将学到。

- 在子目录外删除文件而不删除子目录
- 在批处理文件中保存和恢复搜索路径
- 将目录名用一个驱动器名代替及用驱动器名代替目录名
- 找到一个被代替路径的真正名字

39.2 目录结构的建议

为使用方便和快速的文件存取，用户可以有多种方法来优化目录结构。

若要保持路径短小，则避免使用大于三级或四级的子目录。路径越长，敲入错误的机会就越大。另外，DOS命令行的长度不能超过127个字符，如果要拷贝下列文件就会发生超出DOS命令行长度的问题：

C:\JNFILES\WP\DOCS\DOSBOOK\FIGURES\FIG3-1.PCX

到

D:\STORAGE\BACKUPS\JNF\DOCS\DOSBOOK\FIGURES\SAVE3-1.PCX

如果所有的子目录都放在磁盘开始的部分（即搜索路径中最前面的路径），并且那些经常使用的文件（如DOS命令文件）紧跟其后，硬盘上文件的存取时间就会缩短。这就是说读写磁头不需为寻找目录和相应文件而快速移动。然而，光靠DOS命令来管理磁盘是很困难的。如果由于硬盘上的目录和文件的增多而使磁盘存取速度下降，那么用户可以用第三方提供的实用程序来恢复磁盘的存取速度，例如Norton Utilities的SpeedDisk。

注释： *另一个影响磁盘存取速度的是文件碎片，它在第33章"去除碎片化"中讨论。*

39.3 用 DELTREE 删除文件和子目录

提示： 如果在一个批处理文件中创建一个临时目录，然后用 DELTREE 删除这个目录，一定要用 /Y 开关来使 DELTREE 命令忽略确认步骤。使用 /Y，DOS 就不用要求用户确认是否删除一个他们从未见过的目录。

在前面（在第 14 章“目录管理”），我们已经学习了怎样用 DELTREE 删除整个目录。但是 DELTREE 命令的独特优点是只删除一个目录中的文件而不删除目录本身。

39.3.1 清除一个目录中的所有文件而保留这个目录

怎样清除一个目录中的所有文件而保留这个目录

1. 进入要被清除的目录。
2. 输入 DELTREE /Y *.*。

DOS 会删除这个目录中的所有文件，包括它所有的子目录及其中的文件。

DELTREE 不会删除当前目录，但它将删除其中的所有文件，包括它的子目录。可以用它的这个特点来清除一个目录，下面用户可以自己试试：

1. 创建一个名为 PRACTICE 的目录并且把\DOS*.TXT 文件拷贝进去。
2. 创建一个名为 EXER1 的 PRACTICE 的子目录并且把\DOS\D*.SYS 文件拷贝进去。
3. 创建一个名为 EXER2 的 PRACTICE 的子目录并且把\DOS\M*.HLP 文件拷贝进去。
4. 在这一部分，需要用 PRACTICE 子目录做几个练习，所以把整个子目录复制到\EXERCISE。
5. 进入到\EXERCISE。
6. 输入 DIR /S /P 来看整个子目录的内容。
7. 输入 DELTREE /Y *.* 清除整个目录只保留子目录 EXERCISE。

注释： DELTREE 有个小毛病，必须把/Y 放在文件前面而不是后面，否则 DELTREE 就忽略它。

8. 输入 DIR /S 就会看到目录已经空了。

如果在根目录中清除目录，DELTREE 就清除整个目录树，只有根目录保留。

39.3.2 删除文件但保留目录

因为 DELTREE 不删除当前目录，所以也不删除进入当前目录的路径。就是说，它不删除父目录，父目录的父目录，等等。可以用这个特点来删除一个子目录中的所有文件而保留目录本身。只要进入最低层的子目录然后用 DELTREE 删除那个子目录。

注意，这一动作将删除所有不在当前路径的子目录。用子目录 PRACTICE 做个例子。如果把 EXER1 做为当前目录来删除子目录 PRACTICE，结果是 PRACTICE 和 EXER1 都空了却没有被删除；然而 EXER2 由于不在当前路径而被删除。用户可以自己试试：

1. 把\PRACTICE 复制到\EXERCISE。
2. 进入\EXERCISE。
3. 输入 DIR /S /P 来看这个目录的内容。
4. 进入 EXER1。
5. 输入 DELTREE /Y.. 来删除这个子目录。
6. 回到父目录。
7. 输入 DIR /S 就会看到 EXERCISE 和 EXER1 仍然存在但已经空了。
   注意 EXER2 已经被删除。

39.4　重新组织一棵目录树

DOS 包含一些可使用户重新组织目录树的命令，像转换目录的名字或在相互间没有正常联系的目录之间建立临时的连接。在这些命令中，只有 SUBST 还在使用。其他的命令——APPEND，JOIN 和 ASSIGN（它们已经从 DOS 6 中撤消了，但仍可能出现在 DOS 目录里）——已经永不需要了。在一定 的条件下，APPEND 会导致数据丢失。这一章只讲解如何使用 SUBST，其他的列在“命令参考”中。

39.4.1　为什么使用 APPEND 是危险的

APPEND 命令看起来非常有用——不管当前目录是什么它都可以使某些目录中的文件对用户来说就像在当前目录。APPEND 命令和 PATH 命令非常相似，只不过它是对数据文件而言的。然而，当用户修改拼接目录中的文件时，就会发生问题，如图 39.1 所示。

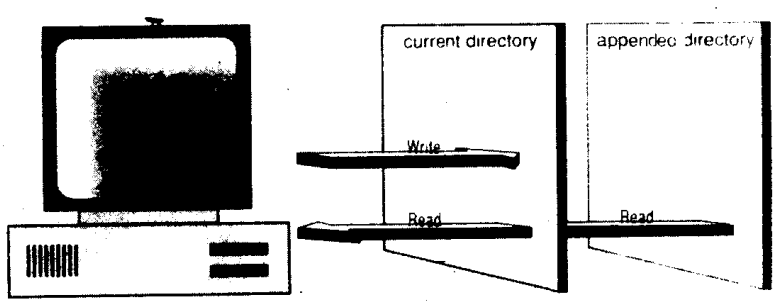

图 39.1　用 APPEND 命令拼接的目录

当一个应用程序要通过保留一个文件的新拷贝（而不是重写已经存在的拷贝）来更新一个文件时，上面的问题就会发生。对于应用程序而言，这个文件就像在当前目录中，而只有 DOS 知道它所在的真正目录。因此，应用程序就在当前目录中保存文件的更新版本，而 DOS 又没有把它放回原来的目录，总的结果是新版本放到了不正确的目录中，而在正确目录中的那个原始版本没有被更新。这种情况不仅发生在用户使用的数据文件上，也发生在对应用程

序的功能非常重要的维护文件上(如 DOSSHELL. INI)。

用户可能认为这是非常罕见的问题,而事实上大多数应用程序都以这种方式来更新文件——所有主要的字处理程序,桌面出现程序,图形制作程序等等。最常见的一个例外是数据库管理程序,它代替原来的文件而不是保存一个新拷贝来的更新文件。

APPEND 命令提供了一点方便,但却带来更大的危险。

39.4.2 使一个目录看起来像一个驱动器

假定对于一个版本很旧的程序,它希望它的程序文件在驱动器 A 上,而数据文件在驱动器 B 上。(对于老式的 PC 机这是非常普通的,因为它们大多数没有硬盘。)最好的办法是更新软件,但如果不可能就只好用现有的软件了。与其在速度慢的软盘上工作,倒不如骗过程序而在硬盘的目录中工作。SUBST 命令可以把一个路径名用一个驱动器名代替(参见图 39.2)。例如,下面的命令就用驱动器 A:代替 C:\ASHAM:

```
SUBST A: C:\ASHAM
```

SUBST

给路径赋予盘名。

句法:

SUBST [drive:path] | [drive:/D]

参数和选项:

none	列出当前替代的情况。
drive	指定作替代后的盘名。
path	指定路径。
/D	撤消指定的替代。

注释: 以 F 命令只能用于物理上存在的磁盘而不能用于操作替代盘:ASSIGN、BACKUP、CHKDSK、FDISK、FORMAT、LABEL、MIRROR、RECOVER、RESTORE、SYS、DISKCOPY 以及 DISKCOMP。若硬行操作,DOS 将给出错误信息。

图 39.2 SUBST 命令总结

在 PC 机上试试这种技术:

1. 在驱动器 A:中插入任意一张格式化的磁盘。
2. 输入 DIR A:,看看 A 盘上有什么。
3. 输入 SUBST A: C:\PRACTICE。

   DOS 不确认这个替换操作。
4. 再输入 DIR A:。

   这时就会看到,DOS 把 C:\PRACTICE 中的内容当作是 A 盘的内容。注意,现在将不能存取真正的 A 盘上的内容。每一个指向 A 盘的操作都被 C:\PRACTICE 替换了。

5. 输入不带参数的 SUBST 就可以看到所有被代替的路径的一个表。
   用户将看到这样的信息 A:=> C:\PRACTICE。
6. 输入 SUBST A: /D 来中止这个替换操作。
7. 再输入 DIR A:。
   这时看到的是真正 A 盘的内容。

注释： SUBST 不能引用当前的驱动器。如想用当前的驱动器代替一个路径，必须首先转到另外的驱动器。同样的，要中止当前驱动器名的代替也要先转到其他驱动器。

39.4.3　用 SUBST 创建一个虚拟驱动器

SUBST 还能解决应用程序不认识路径的问题(例如 WordStar 的早期版本就是这样)。可以用 SUBST 把一个假驱动器名赋给一个目录。例如，把驱动器名 F:赋给目录 C:\WIDGETS，就输入 SUBST F: C:\WIDGETS。被赋予的驱动器名不必是系统中真正存在的驱动器。如果用了一个不存在的驱动器名，就相当于创建了一个虚拟驱动器；就是说，虽然它实际上是个目录，但它的作用就和单独的驱动器一样。

39.4.4　为 SUBST 选择驱动器名

在前面的例子中用了不存在的驱动器名。用户可以用真实的驱动器，诸如 A 或 B。但这样在代替操作有效期间就不能使用真正的驱动器。

选择的驱动器名必须是有效的。虽然 DOS 提供了 26 个驱动器名(A 到 Z)，但所有这些名字并不是自动生效。在缺省情况下，DOS 允许使用五个驱动器名(A 到 E)或者实际使用的那些驱动器的驱动器名(包括 RAM 驱动器)，而不管它有多少个驱动器名。用户必须在 CONFIG.SYS 文件中使用 LASTDRIVE 命令使更多的驱动器名有效(参见图 39.3)。

LASTDRIVE

指定可以采用的最大驱动器字母，因而也就指定了系统可以访问的最大驱动器数：

句法：

LASTDRIVE=x

参数：

x　　指定驱动器字母，由 A 到 Z。表明系统中可使用的最大驱动器字符。

图 39.3　CONFIG.SYS 中的 LASTDRIVE 命令总结

假设用户的机器是普通配置——两个软驱和一个硬盘。对 SUBST 和 ASSIGN 命令而言，驱动器 D 和 E 是有效的。但是如果有四个硬盘呢(从 C 到 F)？在缺省情况下，SUBST 和 ASSIGN 命令就没有虚拟驱动器名可用。如果在文件 C:\CONFIG.SYS 中加入下列命令，就可以有两个虚拟驱动器名(G 和 H)可用。

```
LASTDRIVE=H
```

DOS 在内存中维护一个驱动器名字表，每个驱动器名字占将近 100 字节内存空间。用户可以通过仅申请实际需要的驱动器名字来节省可用内存。如果在 CONFIG.SYS 中加入 LASTDRIVE＝Z 而使所有的可用驱动器名字有效，而实际上只需使用四个驱动器名字，就浪费了 2K 的内存空间。

网络注意： 如果用户的 PC 机属于一个网络，就需要 LASTDRIVE 命令创建一个足够大的驱动器名字表，以便用户可以登录到网络驱动器上。网络管理员会告诉用户合适的 LASTDRIVE 命令。

39.4.4.1 用 SUBST 转换驱动器

怎样转换驱动器 A 和 B

1. 进入驱动器 C(或其他非 A 或 B 的驱动器)。
2. 如果要用名字 A 进入驱动器 B，输入 SUBST A：B:\。

注释： 只要替换有效，用户就不能进入真正的驱动器 A。

3. 如果想用名字 B 进入驱动器 A，输入 SUBST B：A:\。

注释： 只要替换有效就不能进入真正的驱动器 B。

4. 要中止替换，先转换到任何非替换的驱动器上，然后输入 SUBST A：/D 或 SUBST B：/D。

SUBST 还能解决这类问题，假设用户要从 3 ½英寸的软盘上安装一个应用程序，而且只能用 B 驱动器，但是应用程序必须从驱动器 A 上安装。用户可以在安装之前用驱动器 A 代替驱动器 B 而使安装顺利进行。

```
C:\> SUBST A: B:\
C:\> A:
A:\> INSTALL
```

在安装程序执行完毕后，用户必须在中止替换之前离开驱动器 A：

```
A:\> C:
C:\> SUBST A: /D
```

提示： 如果用户需要用一张只适合驱动器 B 的软盘启动机器，就要做更复杂事情。用户不得不交换机箱内 A、B 驱动器的电缆，对 286 以上的机器，还要运行 SETUP 程序来重新定义驱动器(参见硬件手册或从销售商那里得到启动并使用 SETUP 程序的方法)。

39.4.5 TRUENAME 命令

当用户用 SUBST 掩盖了真正的目录结构以后，可能对当前目录是什么或一个文件在哪个目录中产生混淆。DOS 有一个未公开的命令叫 TRUENAME，它能显示一个目录或文件的真正路径(见图 39.4)。

为得到当前目录的真实路径，只需直接执行 TRUENAME，结果如下：

```
A:\> TRUENAME
```

```
C:\> LOCKOUT\
A:\>
```

为得到某一文件的路径，可将文件名作为参数执行 TRUENAME 命令：

```
A:\> TRUENAME LOCKOUT.DAT
C:\> LOCKOUT\LOCKOUT.DAT
A:\>
```

TRUENAME

显示文件或目录的真实路径。

句法：

TRUENAME [drive|path|filespec]

参数：

none	显示当前目录及磁盘的真实路径。
drive	显示 drive 的真实路径。
path	显示 path 所指定的目录的真实路径。
filespec	显示指定文件的路径。

图 39.4 TRUENAME 命令总结

即使当前目录中不存在名为 LOCKOUT.DAT 的文件，上述结果仍将如此。TRUENAME 只管找出当前路径，而不管文件是否存在。

39.5 自测练习

本章讲述了使用 DELTREE 及 SUBST 管理目录的方法。读者可作一些实际操作以巩固前述命令使用的技巧：

- 删除目录\PRACTICE\EXER2 中的全部文件。
- 删除 PRACTICE 目录中的内容。
- 以驱动器 A 替代\PRACTICE 目录。
- 找出驱动器 A 上文件的真实路径。
- 撤消替代。
- 确定 PC 上最多可用的盘名数。

完成以上训练后，删除 PRACTICE 目录。

第 40 章 高级拷贝技术

简介： DOS 的拷贝命令可以用于普通的拷贝工作，但是每条命令还提供 DOS 或 Windows 中没有的独一无二的特殊功能。

40.1 处理一些特殊拷贝任务

前面已经讲了如何利用 DOS Shell 拷贝文件，涉及了 COPY 命令，XCOPY 命令以及 Microsoft 的 Backup 程序。COPY 和 XCOPY 还有一些特殊用途，它们可以完成一些不常遇见的拷贝任务。例如，COPY 可以用来合并文本文件以及改变文件的日期和时间，XCOPY 可以根据文件的归档属性、日期和时间有选择地进行拷贝。

另外，REPLACE 命令能比较两个目录并选择拷贝文件，它既可以把文件追加到目标目录中，也可以替换目标目录中的同名文件。

本章将讲解如何使用这些 DOS 拷贝命令的高级特性，具体内容有以下几个方面：

- 在批处理文件中用 XCOPY 命令拷贝一组文件到几个软盘
- 用 XCOPY 剪裁和移动目录
- 用 COPY 合并文件
- 用 COPY 改变文件的日期/时间
- 用 REPLACE 命令向目标目录中追加文件
- 用 REPLACE 命令替换同名文件

40.2 XCOPY 命令

图 40.1 示出了 XCOPY 命令的句法（有人也称 XCOPY 命令为外延拷贝、交叉拷贝或扩展拷贝）。

XCOPY

拷贝文件和子目录。

句法：

XCOPY source [destination] [/A|/M] [/D:date] [/P] [/W]

参数和选项：

source	指明准备拷贝的文件。
destination	指明要拷贝到哪一个目录中，缺省值是当前目录。

/A	只拷贝归档属性为 1 的文件，而且不改变文件的归档属性。
/M	只拷贝归档属性为 1 的文件，但将归档属性删除。
/D:date	只拷贝指定日期以后所创建或修改过的文件。
/P	在拷贝每个文件之前，都会询问是否拷贝该文件，等用户回答“Y/N”之后，才决定是否拷贝。
/W	显示一个暂停信息，等用户按下任一键后再继续拷贝。

图 40.1　XCOPY 命令总结

40.2.1　XCOPY 和 DISKCOPY

有时候需要进行整片软盘地拷贝，DISKCOPY 可以把文件从一个软盘上逐个磁道地拷贝到另外一片软盘上，两片软盘必须具有同样的格式和大小。XCOPY 也可用于整片盘的拷贝，与 DISCOPY 不同，它按文件逐个进行拷贝，因此具有下述优点：

- 源盘和目标盘不必具有相同的容量，例如可以把文件从 1.2MB 的软盘上拷贝到 1.44MB 的软盘上。
- XCOPY 不拷贝源盘的格式，而 DISKCOPY 却要拷贝。例如，DISKCOPY 会拷贝源盘上的坏扇区标志，从而把目标盘上并没有坏的扇区当作坏扇区。
- 如果目标盘上有坏扇区，DISKCOPY 就不能执行，但 XCOPY 仍可以执行。
- DISKCOPY 将毁坏目标盘上的所有文件，XCOPY 则不。
- 如果源盘中文件存放空间很分散(即存在文件碎片)，用 DISKCOPY 拷贝时，目标盘中仍会存在文件碎片。当用 XCOPY 拷贝时，只要目标盘是空的，拷贝后目标盘中文件存放空间就集中在一起。

但不能用 XCOPY 复制软件包的安装盘，因为复制安装盘时，必须与源盘相同地逐个磁道地复制，因此应当用 DISKCOPY 命令。

40.2.2　避免磁盘空间不足时命令中断

如果目标盘容纳不下所要拷贝的文件时，COPY 和 XCOPY 命令都要中断，这是它们的不足之处。例如，如果要拷贝 *.DOC 文件到 A 盘，而才拷到一半时磁盘就满了，则 COPY 和 XCOPY 命令就会中断而回到 DOS 提示符下。怎样处理这种情形呢？如果再放一片软盘到 A 驱动器并键入 COPY 或 XCOPY 命令，那么以前已经拷贝过的文件又会再拷贝一次，因为 COPY 和 XCOPY 不知道哪些文件已经拷贝过。这时候用户只能逐个文件地进行拷贝。

不过有几种办法可以解决这个问题。第一个办法是利用 DOS Shell，它可以在换掉磁盘后继续拷贝。另一个办法是用第 23 章所讲的 Backup 程序。第三个办法是把 XCOPY 和归档属性连在一起用，不过由于 XCOPY 改变了源文件的归档属性，因此会影响以后的备份(backup)操作，但这却是解决问题的最佳方法。

怎样用 XCOPY 拷贝大量文件到软盘

1. 可以先用 backup 备份文件，该操作会删除文件的归档属性。
2. 键入 ATTRIB filespec +A 命令打开驱动器上所有文件的归档属性。如果要拷贝整个目录下的文件，在该命令后加/S 选项。
3. 键入 XCOPY filespec d：/M /W 命令，开始把文件拷贝到驱动器 d。如果要拷贝目录下的所有文件，在命令后加/S 选项；/M 选项表示根据文件的属性选择拷贝文件，当拷贝完毕后，这些属性被删除。
   XCOPY 暂停并显示信息 Press any key to begin file(s)。
4. 重新在驱动器中放入一个空的格式化过的软盘，按任意键。
   XCOPY 开始拷贝文件，最后显示信息 Disk Full，命令中断。
5. 重复第 3 和第 4 步，直到所有文件拷贝完。

做该练习时是将大量文件从 DOS 目录下拷到软盘上，至少需要两片空的格式化过的软盘。

1. 先用 backup 备份 DOS 目录下的所有文件，因为该练习会删除文件的归档属性。
2. 改变目录到 DOS 子目录下。
3. 键入 ATTRIB *.* +A 命令打开目录下所有文件的归档属性。

注释： XCOPY 不能拷贝隐含或系统文件。假如要拷贝这些隐含或系统文件，必须在命令后加－H 和－S 选项以删除文件的系统属性。但首先得键入 ATTRIB *.* >PRN(或一个临时性文件)以建立文件属性档案，拷贝完成后，用该档案来确定哪些文件需要恢复它们的隐含或系统属性。

4. 键入 XCOPY *.* A：/M /W 命令，便开始拷贝文件到驱动器 A 上(如果打算把文件拷到驱动器 B 上，就用 B：代替 A：)。
   因为用了/W 选项，因此 XCOPY 暂停并显示信息 Press any key to begin copying file(s)。
5. 放一片空的格式化过的软盘到驱动器中，并按任意键。
   XCOPY 开始向磁盘中拷贝文件，最后将显示信息 Disk full 并中断拷贝。
6. 键入 ATTRIB *.* 命令，查看一下目录下文件的属性。
   注意那些已经拷贝了的文件的归档属性被删除了(因为用了/M 选项)，但没有拷贝的文件仍然有归档属性。
7. 重复第 3 和第 4 步，直到所有文件拷贝完。

完成该操作过程的批处理文件如下所示。%1 是要拷贝的文件名。该操作包括拷贝隐含文件和系统文件，并打印出所有源文件的原始属性，这样可以根据打印结果恢复一些文件的属性。

```
@ECHO OFF
REM Record the current attributes in case we want to
REM restore them when the job is done.
ECHO Current attributes for the %1 files: > PRN
ECHO. >PRN
ATTRIB %1 /S > PRN
```

```
REM Turn on the archive attributes for all %1 files.
REM Also,turn off their hidden and system attributes.
ATTRIB %1 /S +A -H -S

:COPYLOOP
REM Copy all the %1 files that have positive archive
REM attributes. Turn off attributes as you copy:
ECHO Insert the next target diskette in drive A:
PAUSE
XCOPY %1 A: /S /M

REM Repeat loop if XCOPY terminated unsuccessfully
REM for any reason,such as a disk full error.
IF ERRORLEVEL 1 GOTO COPYLOOP

:ENDING
ECHO * * * * * * * * * * * * * * * * * * * * * * * *
ECHO *                                               *
ECHO * The archive,hidden,and system attributes       *
ECHO * have been removed from the source files.       *
ECHO * Check the printout and restore attributes as    *
ECHO * needed.                                        *
ECHO *                                               *
ECHO * * * * * * * * * * * * * * * * * * * * * * * *
```

如果在该批处理文件中再加入一些命令，可以使其更完美：

- 设置%2，以接受目标驱动器的名称和路径。
- 设置%3，以接受 Y 或 N 来判断是否拷贝隐含和系统文件；或者设置 CHOICE 命令，从键盘上接受 Y 或 N。

40.2.3　剪裁和移动

剪裁和移动是树形目录操作术语，在 PC 机中用于描述把整个目录或目录的一部分，从一个父目录移动到另一个父目录下。如果用户手中没有其他专用软件来作剪裁和移动，就可以用 XCOPY 和 DELTREE 命令完成。

怎样剪裁和移动目录

1. 如果源目录下有隐含和系统文件，键入命令 ATTRIB source-path\directory-name*.* /S>PRN，建立隐含和系统属性档案，以便随后恢复。
2. 键入 ATTRIB source-path\directory-name*.* /S 命令，删除文件的隐含和系统属性。
3. 键入 XCOPY source-path\directory-name destination-path\directory-name /S 命令，拷贝目录及其下的文件到另外的驱动器上。如果源目录名称和目标目录名称相同，那么它们的子目录名称也相同。
4. 键入 DELTREE source-path\directory-name 命令，删除源目录及其下的文件。
5. 利用打印出的属性档案，恢复拷贝的文件的隐含和系统属性。

在下面的练习中，读者先建立一个名为 PRACTICE 的目录作为 DOS 目录的子目录，然后把它移到根目录下。

1. 建立一个目录\DOS\PRACTICE,然后拷贝所有的\DOS*.TXT 文件到 PRACTICE 目录下。
2. 键入 XCOPY \DOS\PRACTICE*.*\PRACTICE\命令,把目录\DOS\PRACTICE 用 XCOPY 拷贝到目录\PRACTICE 下。

注释: 在 XCOPY 命令中,必须指明目标目录。如果仅指明根目录作为目标目录,DOS 将拷贝目录\DOS\PRACTICE 下的所有文件到根目录下,但不会建立一个新目录。

如果要移动整个目录树而不仅是一个目录,就必须在命令后加上/S 选项。

3. 键入 DIR \PRACTICE\命令,查看新建的目录以及其下面的文件。
4. 键入 DELTREE /Y \DOS\PRACTICE 命令删除旧目录。

这样便完成了目录的剪裁和移动。

在下一部分中仍然用\PRACTICE 目录为练习对象。

40.3 合并文件

COPY 命令可用于合并文件,图 40.2 示出了使用 COPY 命令合并文件时的句法。

COPY(合并文件)

把两个或两个以上的文件合并成一个文件。

句法:

COPY [switches] source [+...][destination] [switches]

参数和选项:

source	源文件。
destination	目标文件。
/A	拷贝文件时,将文件视为 ASCII 文本文件来处理。
/B	指明要拷贝的文件是二进制文件,按二进制文件处理。

注释:

/A 和/B 选项作用到其后的所有文件,直到重新遇到/A 或/B 选项为止。

图 40.2 COPY 命令总结(用于合并文件)

练习合并目录\PRACTICE 下的*.TXT 文件:

1. 将目录转到\PRACTICE 下。
2. 键入 COPY README.TXT + WHATSNEW.TXT + NETWORKS.TXT COLLECT.TXT

在 DOS 进行操作时,将在屏幕上显示如下信息:

```
README.TXT
WHATSNEW.TXT
```

```
NETWORKS.TXT
        1 File(s) copied
```

信息告诉用户只拷贝了一个文件，因为只产生(或替换)了一个新文件。

3. 键入 DIR *.TXT 命令查看一下新建立的文件及原来的文件，注意 COLLECT.TXT 的大小是 README.TXT、WHATNEW.TXT 和 NETWORKS.TXT 文件的总和，也可从 COLLECT.TXT 的日期和时间上看到它的确是刚刚创建的文件。

下面的命令能产生同样的结果，但有时也与预期的结果不符：

```
C:\DOCS>COPY *.TXT COLLECT.TXT
```

因为源文件只用了一个文件通配符，没有用＋号，COPY 将把所有与通配符匹配的源文件合并到一个目标文件中去，这有可能把用户不需要的文件也包括进去。在目标文件中各源文件出现的顺序与它们在目录中出现的顺序一样，也就是说，如果 NETWORKS.TXT 在目录中第一个出现，那么在目标文件 COLLECT.TXT 中也是第一个出现。

注释： 当 DOS 创建一个新文件的时候，它先用第一个可用的目录项，该目录项可能是以前删除了的文件留下的，因此当前目录下的目录项的顺序是任意的。即使用户先写入 CHAPTER1，再与入 CHAPTER2，最后写入 CHAPTER3，这三个文件在目录中出现的顺序也可能是任意的。
当用户用 DEFRAG(参见第 33 章)优化驱动器时可以根据某个标准对文件进行排序。

40.3.1　附加文件

有时候用户希望用附加的方式合并文件，例如，如果打算把文件 WHATSNEW.TXT 和 NETWORKS.TXT 附加到 README.TXT 后面而不另外再建一个新文件，则可按如下操作：

1. 确认正处于\PRACTICE 目录下。
2. 键入 COPY README.TXT＋WHATSNEW.TXT＋NETWORKS.TXT /Y 命令(选项/Y 用于避免 COPY 对 README.TXT 的重写)。
   屏幕上将显示下列信息：

```
README.TXT
WHATSNEW.TXT
NETWORKS.TXT
        1 File(s) copied
```

3. 键入 DIR *.TXT 命令，查看 COPY 执行的结果，请注意 README.TXT 的大小、日期和时间。

注释： 在这种情形下，如果 COPY 找不到文件 README.TXT，它就把 NETWORKS.TXT 合并到 WHATSNEW.TXT 中去(如果 WHATSNEW.TXT 存在的话)。当命令中没有指明目标文件时，COPY 就使用命令中最左端的文件作为目标文件。

下面的命令也是合法的，但有可能结果不同，因为所指明的文件可能与用户要求不一致，而且它们的顺序也与要求不同。该命令的目标文件是当前目录中符合通配符 *.TXT 的第一个文件：

```
C:\DOCS>COPY *.TXT
```

还可以在该命令的后面指明目标文件，这样就把所有具有扩展名 TXT 的文件合并到了 README.TXT 中。命令如下：

```
C:\DOCS>COPY *.TXT README.TXT
```

答疑： 假如合并文件时得到如下显示：Content of destination lost before copy，是否损坏了文件？

如果源文件之一被用作目标文件，DOS 就打开该文件并开始把其他的源文件附加在它后面。当第一个文件附加完后，该目标文件就不能再变回去了。如果 DOS 在指定的目录中找到了目标文件，而且发现它也符合源文件的通配符时，它就会发现不能把目标文件附加在自身的后面，因此不会引起误操作。

这条信息看起来具有破坏性，但实际上操作进行得很正常，因为目标文件已经被改动，它不能再被用作源文件。事实上，用户可以忽略这条信息。

40.3.2 各种不同类型文件的合并

因为 README.TXT，WHATSNEW.TXT 和 NETWORKS.TXT 是 ASCII 码文本文件(无格式)，它们在合并时不会发生问题。其他类型文件的合并就不那么容易，下面分段讲述。

40.3.2.1 数据文件

假设文件 PART1.DOC，PART2.DOC 和文件 PART3.DOC 是由字处理程序生成的，并且以字处理器固有的格式存储，文件可能包括文件头和文件尾。当用户打算将它们合并时，由于所有的文件头和文件尾都包含在文件中，因此会产生一个原字处理程序处理不了的很复杂的文件。当今一些应用软件如桌面印刷系统、数据库等所生成的文件在用 COPY 合并时也存在同样的问题。

要合并这样的文件，不能用 COPY 命令，而要用原应用软件中提供的子功能。它们在合并文件时会首先去掉文件头和文件尾，合并后再把文件头和文件尾加上去。

40.3.2.2 二进制文件

和数据文件相似，不能用 COPY 合并诸如可执行文件这样的二进制文件。合并两个可执行程序模块并不能得到一个可执行文件，相反，可能会产生一个有损自己数据的程序。因此，要合并二进制文件，得用专门的工具软件，例如，可以用 Link 连接程序合并两个可执行程序模块。

作者忠告： 用生成文件的原始应用软件合并文件。

只有在合并 ASCII 文本文件时，COPY 命令才最便利，对于其他类型的文件，最好采用生成文件的应用软件的子功能来完成文件的合并。

40.3.3　更新日期/时间

当用 DOS 拷贝文件时，它并不更新文件的日期/时间标记，因为时间和日期指明了文件的版本。但在合并文件时，由于新产生的目标文件是属于新版本，因此 DOS 会为它指定当前的时间和日期。

不过，可以如下使用 COPY 命令，造成合并文件的假象，让 DOS 为文件指定当前的时间和日期：

```
COPY source-filespec /B +,, [destination]
```

注意源文件只能使用一个文件名，不能用通配符，因为使用通配符会发生真正的合并操作。所以当用户准备更新一组文件的时间和日期时，应该逐个进行，也可以用批处理命令 FOR 形成循环来依次完成。

加号后面的两个相连的逗号强迫 DOS 停止合并操作。如果命令中包含了目标文件，DOS 便把源文件拷贝到目标文件中，并且只更新目标文件的日期/时间，如果没有指明目标文件，则更新源文件的日期/时间。

警告： 选项/B 很重要，它防止 DOS 损坏文件，本章后面"/A 和/B 选项的基本操作"一节中会对此作解释。因此千万不要丢掉选项/B，否则会丢掉数据。

注释： 在 DOS 6.2 和 6.22 环境中，用户可以使用/Y 选项来覆盖拷贝保护。

练习更新 NETWORKS.TXT 文件的日期/时间：

1. 键入 DIR NETWORKS.TXT 命令，查看它原来的日期/时间。
2. 键入 COPY NETWORKS.TXT /B+,,命令，用系统的当前时间更新文件原来的日期/时间。
3. 键入 DIR NETWORKS.TXT 命令，再次查看文件的日期/时间。

提示： 下面的批处理程序可用于随意更换一组文件的日期/时间。其中%1 是新时间，%2 是新日期；%3 到%9 是要改变日期/时间的文件名，它也可以是通配符。

```
@ECHO OFF
TIME %1 > NUL
DATE %2 > NUL
FOR %%F IN (%3 %4 %5 %6 %7 %8 %9) DO COPY /B %%F+,,
ECHO The system time and date have been changed.
ECHO Please reset the correct time:
TIME
ECHO Please reset the correct date:
DATE
```

40.3.4　匹配合并

COPY 命令还能合并具有相同的文件名但扩展名不同的文件，例如下面的命令：

```
COPY *.TXT+*.ADD *.NEW
```

该命令能够把 FORM.TXT 和 FORM.ADD 合并到 FORM.NEW 中，或者把

TEACH. TXT 和 TEACH. ADD 合并到 TEACH. NEW 中，等等。如果当前目录中存在 TEACH. ADD 文件但不存在 TEACH. TXT 文件，或者相反，DOS 将显示错误信息并且不生成 TEACH. NEW 文件。

下面是另外一些匹配合并使用方式：

- 源文件可以不止两个：

```
COPY *.1+*.2+*.3+*.3+*.4 *.ALL
```

- 可以省略目标文件而把合并结果放到第一个文件中：

```
COPY *.TXT+*.ADD
```

40.3.5 /A 和/B 选项的基本操作

在拷贝和合并文件时，/A 和/B 选项有时显得很重要，为了正确使用它们，应当懂得一些关于 DOS 文件存储方面的知识。

40.3.5.1 簇和无效空间

簇是 DOS 在磁盘上存在文件时的存储单位。每一个磁盘中族的大小是固定的，它和磁盘的容量以及格式化磁盘的程序有关。例如，低容量(360KB)的软盘每簇中有 512 个字节，但大容量如 360MB 硬盘每簇中有 4096 个字节(或更多)。簇的大小在格式化时便已设定好。

DOS 以簇为单位存储文件。如果一个文件不满一簇，剩下的空间便称为无效空间。假如一个磁盘的簇大小是 2048 字节，一个一字节的文件也将占据一整簇，剩余的 2047 个字符便是无效空间。一个 15KB 的文件将占据 8 个簇，最后一簇有 1KB 无效。无效空间是被浪费的空间，它不能被分配给别的文件。无效空间中可能存储有以前文件的内容。

40.3.5.2 访问文件时如何回避无效字节

当 DOS 进行文件拷贝、打印、合并或其他操作时，它都得先读取文件。读取文件时应当只读有效的文件数据而忽略无效字节。DOS 有两种方法来确认簇中合法数据结尾之所在：

- 在 ASCII 码方式下以文件结束符(EOF)作为结尾标志——文件结束符是 ^ 2 或十六进制数 1A，它作为最后一个字符出现在文件中。当以 ASCII 码方式读取文件时，DOS 扫描它所读的数据，一旦遇到十六进制数 1A，便停止读取操作(如果没有文件结束符(EOF)，DOS 就用文件的大小作为判断决定文件的结尾。)
- 在二进制方式下，DOS 以目录项中记录的文件大小作为判断。如果目录项中指明文件长 3 267 字节，DOS 将从磁盘上读取刚好 3 267 个字节的数据，其中任何十六进制的 1A 都被当作普通数据读取。

ASCII 码方式存在着一个问题，那就是仅在文件最多只有一个 EOF 符号，而且 EOF 符号位于文件的结尾时它才能正常工作，但事实上有许多文件，如二进制文件、工作表、图形文件及其他一些应用文件都不是如此，它们往往含有 EOF 符号，而且是它们数据的一部分。因此，ASCII 方式只有在文件为真正的 ASCII 文件，也就是说文件中只有 ASCII 字符而没有格式信息时，它才能可靠地工作。但二进制方式在任何情况下都能正常工作。

DOS 在写文件时也有 ASCII 码和二进制两种方式。以 ASCII 码方式写文件时，DOS 在文件的结尾加上 EOF 符号，但二进制方式不这样做。

一些 DOS 命令(特别是老的版本)采用 ASCII 码方式,而另一些则采用二进制方式。例如 TYPE 和 PRINT 命令,它们使用 ASCII 方式处理 ASCII 码格式的文件;而 XCOPY 和 MOVE,不论是何种类型的文件,它们都用二进制方式处理。

40.3.5.3　拷贝方式

COPY 命令既可以在二进制方式下工作也可以在 ASCII 码方式下工作,但大多数在二进制方式下工作。当文件只能是 ASCII 码文件,例如,合并文件以及把文件拷贝到 CON 或 PRN 时,它才采用 ASCII 码方式。但用户可以用/A 和/B 选项强迫 COPY 命令采取用户所希望的方式。

用户可以在源文件名之前或之后加上/A 选项,迫使 COPY 以 ASCII 码方式读取源文件;也可以在目标文件之后加上/A 选项强迫 COPY 以 ASCII 码方式写文件并在文件末尾加上 EOF 符号。EOF 符号对 DOS 程序并不重要,但也许对其他一些应用软件必不可少(现在大多数软件都不需要 EOF 符号)。

用户可以在源文件名之前或之后加/B 选项,强迫 COPY 以二进制方式读取源文件。如果用户需要合并二进制文件,就不得不用/B 选项,这样 COPY 才能用二进制方式读取源文件。否则 COPY 将按缺省值 ASCII 方式进行合并。

用户也可以在目标文件之后加/B 选项,让 COPY 以二进制方式写文件而不在文件结束时写上 EOF 符号。例如,如果用户希望合并二进制文件并且不希望 DOS 在文件结尾加上 EOF 符号时,就需要加上这个选项。

40.3.5.4　更新非 ASCII 码文件的日期

如果打算更新非 ASCII 码文件的日期/时间,则需要加上/B 选项。如果没有用/B 选项,在文件合并过程中,DOS 就从第一次遇到十六进制数 1A 的字节开始把文件截断;没有选项/B,DOS 还要在文件末尾加上 EOF 符号。在文件名前面加上了/B 选项,DOS 将拷贝整个文件,为文件指定新的日期/时间,并且在文件末尾不加 EOF 符号。下例是更新名为 DEFRAG.COM 的文件的日期/时间:

```
COPY /B DEFRAG.COM+,,
```

答疑:　当笔者在用 COPY 命令更新文件的日期/时间时,曾经丢失过文件的一部分。

当用 COPY 命令的文件合并功能改变文件的时间/日期,如果在命令中不使用/B 选项,则 DOS 自动选择 ASCII 码方式进行合并,在这种方式下,COPY 一旦遇到文件结束符就停止操作。如果文件中有非 ASCII 码数据,该过程很有可能截断文件。文件一旦毁坏,用户不能再用 DOS 应用程序进行修复,因此必须从文件备份中恢复。如果被截断的数据是文本数据,或许可以用修复专用程序从磁盘上未用的簇中恢复一部分,例如,使用 Norton Utilities' UnErase。

为了避免出现这样的问题,当用 COPY 命令更新文件的时间/日期时,一定记住加上/B 选项。

作者忠告:　更新文件日期/时间时,一定使用/B 选项。

当更新 ASCII 码文件的时间/日期时,不一定要使用/B 选项,但养成使用它是一种

好习惯，因为在操作非 ASCII 码文件时，选项/B 是必需的。用户也可以定义一个 DATESTMP 宏命令来完成 COPY /B %1+,, ，然后键入 DATESTMP DEFRAG.COM 命令即可。

40.3.5.5 向设备拷贝文件

在把非 ASCII 码文件拷贝到一个设备时，一定要使用/B 选项，因为缺省时，无论是何种设备，COPY 均用 ASCII 码方式拷贝，这样对于非 ASCII 码文件，操作便不能正确进行。

40.4 REPLACE 命令

REPLACE 命令有两个特别的功能，两者都通过对源目录和目标目录的比较来选择文件：

- 把存储于源目录但目标目录中没有的文件追加到目标目录中去(在这儿 REPLACE 有些像恢复目标目录中丢失了的文件)。
- 替换在源目录和目标目录中均存在的文件(在这儿 REPLACE 起用替换项的作用)。

图 40.3 示出了 REPLACE 命令的句法。

REPLACE

从一个目录到另一个目录进行文件替换成添加。

句法：

REPLACE filespec path [/P] [/R] [/W] [/S] [/U] [/A]

替换目标目录中与源目录文件同名的文件，或把目标目录中不存在但源目录中存在的文件添加到目标目录中去。

参数和选项：

filespec	源文件。
path	目标目录。
/P	在替换一个文件之前，系统显示提示信息以确认是否一定要替换。
/R	表示只读属性的文件也可以替换。
/W	在寻找源文件之前，系统等待按下任意键后才开始工作。
/S	对目标目录及其所有子目录进行逐一搜索，寻找要被替换的文件。文件名相同者被替换。
/U	检查源目录和目标目录中文件的日期/时间，然后用源目录中日期较新的文件取代目标目录中日期较旧的文件。如果目标目录中的同名文件日期较新，则不予取代。
/A	将存于源目录中而不存在于目标目录中的文件添加到目标目录中。该选项不能同/R、/S 或/U 选项一起使用。

图 40.3 REPLACE 命令总结

40.4.1　追加文件

假如用户删除了几个在 C:\PROPS 目录下的文件而随后又改变了主意，又假如已经不能恢复这些文件了，但在网络驱动器上有这些文件的备份，(如果这些备份文件是用 backup 程序生成的，它便不能直接用于 REPLACE 操作，但用户可以用 RESTORE 进行恢复。)则使用 REPLACE 命令是恢复它们的最容易的方法。下述命令能选择通配符为 *.PLS 的文件进行恢复，这些文件存在于 G:\PROPS 中但不存在于 C:\PROPS 中。

```
REPLACE G:\PROPS\*.PLS C:\PROPS /A /P
```

由于使用了/P 选项，REPLACE 将显示每一个文件名并让用户决定是否拷贝该文件，这有助于避免不需要的恢复。

练习删除并恢复一些\PRACTICE 目录下的文件。

1. 删除\PRACTICE\NETWORKS. TXT 和 WHATSNEW. TXT。
2. 键入 REPLACE \DOS*. TXT \PRACTICE /A /P 命令。
3. 对每个提示性显示都回答 Y。

作者忠告： 记住使用/P 选项。

功能较强的 REPLACE 命令会追加一些用户并不打算要的文件，甚至重写一些新的同名文件。选项/P 能给用户提供一个选择文件的机会，并能准确地给出有关这类操作的预警。

40.4.2　替换已经存在的文件

假如用户在硬盘的几个不同区域安装了一套应用软件，而现在又接到了新版本，这时便可使用 REPLACE 命令。下述命令能够搜寻并替换 C 盘上所有与 A 盘上文件同名的文件。

```
REPLACE A:C:\ /R /S /R
```

练习用 DOS 目录下的原始文件替换 PRACTICE 目录下的同名文件。

1. 键入 REPLACE \DOS*. TXT \PRACTICE /P。
   在替换每个文件之前，DOS 都显示提示信息，征得用户同意后再进行替换。
2. 键入 DIR \PRACTICE 命令，查看 REPLACE 命令执行情况。注意 README. TXT 文件被恢复到其原始文件的大小和日期/时间，NETWORKS. TXT 也被恢复其原始文件的日期/时间。

REPLACE 的/U 选项使得只替换那些日期比源文件旧的文件。

40.5　自 测 练 习

本章讲明了 XCOPY、COPY 和 REPLACE 命令的几个高级技巧，读者可以根据下面几条提示自行练习：

1. 把一组文件拷贝到一组软盘上。

2. 移动\PRACTICE 目录，使它变成 DOS 目录的子目录。
3. 把目录\DOS\PRACTICE 下的所有文件合并到一个文件中，该文件取名 ONMY-OWN. TXT。
4. 更新文件\DOS\PRACTICE\README. TXT 的日期/时间。
5. 写一个 FOR 循环命令，更新\DOS\PRACTICE 目录下的所有文件的日期/时间。
6. 删除\DOS\PRACTICE 目录下的一些文件，然后再用 DOS 目录进行 REPLACE 操作。
7. 在完成本章练习之前，删除\DOS\PRACTICE 目录。

第 41 章　INTERLNK 命令

简介： DOS 6 新设置的 INTERLNK 命令可以把两台计算机连接起来，使其中一台计算机能访问另一台计算机。例如，可以把其中一台计算机上的文件拷贝或移到另一台计算机上，或者执行装在另一台计算机上的程序，也可以使两台计算机共用一台打印机。

41.1　使用 INTERLNK 命令连接两台计算机

INTERLNK 命令主要用于通过笔记本计算机来访问台式计算机，这样就可以很方便地传送或打印两台计算机中的文件。但是，只有两台计算机距离较近而且没有连网时才便于使用。

用 INTERLNK 命令连接的两台计算机，其中一台称作服务器，另一台称作客户机（见图 41.1）。当处于连接状态的时候，服务器不能做其他任何事情，唯一可做的是通过它的键盘键入中止连接的命令，所有的工作都由客户机完成，它可以像使用自己的配置一样使用服务器的驱动器或并行口。为了做到这一点，INTERLNK 命令将给服务器的驱动器或通信口设定名字。例如，假如客户机只有一个硬盘 C，那么 INTERLNK 命令就将名字 D 分配给服务器的硬盘 C，这样，客户机就可以采用把文件从 C 盘拷到 D 盘的方式把文件从自己的硬盘拷到服务器的硬盘上。

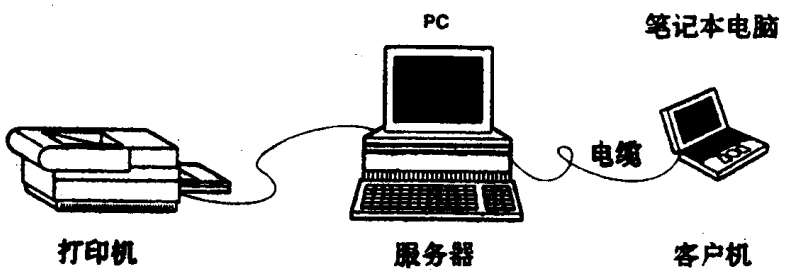

图 41.1　客户机服务器连接

客户驱动器和端口
A：
B：
C：
D：＝服务器驱动器 C：
E：＝服务器驱动器 D：
LPT1：＝服务器端口 LPT1

本章主要讲述：

- **创建两台计算机的连接**
- **从客户机访问服务器的驱动器和通信口**

- 中止两台计算机的连接

41.2 INTERLNK 建立过程

主要通过三个步骤建立两台计算机之间的连接：

- 首先，用缆线连接两台计算机。
- 接着，在服务器上执行 INTERSRV 程序。这步使计算机处于服务器状态。
- 最后，在客户机上通过 CONFIG.SYS 装入 INTERLNK.EXE 驱动程序。经过这三步后，便完成了连接过程。

如果需要经常把笔记本计算机和台式 PC 计算机相连以便来回传送文件，可以在笔记本计算机的 CONFIG.SYS 文件中建立交替配置，这样便能够选择在引导时是否装入 INTERLNK.EXE；或者在 DEVICE 命令中的 DEVICE 或 DEVICEHIGH 之后加一个问号(?)，在每次引导时 DOS 便会问是否打算装入 INTERLNK.EXE(所有这两种技巧均在第 38 章"启动 DOS"中有解释)。假如笔记本计算机的高端内存是空着的话，每次引导时都可以装入 INTERLNK.EXE，而不管是否打算和台式计算机连接。

当没有用缆线把客户机和另外一台计算机连接，或者另一台计算机没有处于服务器状态时，INTERLNK 命令便不会生效。在这种情况下如果要将两台计算机连接起来，就得再走一步——在客户机上键入 INTERLNK 命令。

注释： 如果需要经常把两台计算机相连接，也可以使用其他软件，例如 Norton pcANYWHERE。该软件还能使用户在家中通过电话线使用 modem 控制办公室里的计算机。

41.3 用缆线连接两台计算机

可以通过串行口或并行口把两台计算机连起来，但两台计算机的通信口类型必须一致。

不过，所使用的通信口的序号可不一致，例如，可以用缆线将客户机的 LPT1 和服务器的 LPT3 口连接。

一般采用下面三种缆线：

- 3 线串行缆线
- 无调制解调器的 7 线缆线
- 双向并行缆线

现在可以将两台计算机连接起来了：

1. 选择适当的缆线。
2. 连接两台计算机的串行口或并行口。

41.4 建立服务器

图 41.2 示出了 INTERSVR 命令的使用情况，用该命令使计算机处于服务器状态。如

果使用缺省设置，则仅键入 INTERSVR 即可，这时服务器的所有驱动器和通信口都可以重新定向。假如只让 C 盘重新定向，则键入下述命令：

```
INTERSVR C:
```

或者通过禁止 A 和 B 驱动器的重定向，亦即使用：

```
INTERSVR /X=A /X=B
```

注释： INTERSVR 命令不能重定向网络驱动器，CD-ROM 驱动器，或其他任何使用于重定向界面的驱动器。

INTERSVR

装入 INTERLNK 服务器软件，重定向驱动器和打印机接口；把 INTERLNK 文件从一台计算机拷贝到另一台计算机。

句法：

```
INTERSVR[drive...] [/X=drive...]
[/LPT[n] |/COM[n]] [BAUD:rate] [/V] [/RCOPY]
```

参数和选项：

none	装入服务器程序并显示服务器屏幕。
drive	指明用于重定向的驱动器。缺省时，INTERSVR 将重定向所有的驱动器。
/X=drive	标明不用于重定向的服务器上的驱动器。
/LPT	表明缆线连接在并行口上。
/COM	表明缆线连接在串行口上。
n	指明缆线连接的端口序号。
/BAUD:rate	设置数据传送的最大波特率，可用的值有 9600，19200，38400，57600 和 115 200，缺省值是 115200。
/V	防止与服务器时钟冲突。
/RCOPY	把 INTERLNK 程序从一台计算机拷贝到另一台计算机上。

图 41.2　INTERSVR 命令总结

怎样在启动客户机之前连接两台计算机

1. 用缆线连接两台计算机。
2. 在服务器计算机上键入 INTERSVR 命令。

   在服务器计算机的屏幕上会显示驱动器和端口表，但由于两台计算机还没有连接起来，所以并不显示新的驱动器和端口名。
3. 编辑客户机的 CONFIG.SYS 文件，以便能装入 INTERLNK.EXE。假如已经编好了的话，这一步可以省略。

4. 重新启动客户机，以确认 INTERLNK.EXE 已经装入。
   INTERLNK.EXE 搜索用缆线连接的端口并建立连接，显示重定向了的服务器上的驱动器和端口，同时服务器屏幕上也映象出新的驱动器和端口列表。
5. 现在便可以按显示在屏幕上的名字使用服务器上的驱动器和端口。
6. 当打算中止连接时，先按服务器键盘上的 Alt+F4，然后便可以取下缆线了。
7. 如果打算清除客户机内存中的 INTERLNK.EXE 程序以供他用，可先编辑 CONFIG.SYS 文件，删除装入 INTERLNK.EXE 的命令，然后重新启动计算机。

如果在命令行中没有给出端口名字，INTERSVR 程序将搜索服务器所有端口，以寻找缆线连接的位置，这样便会占用更大的内存空间，因为它需要装入一系列操作串行口和并行口的程序。但是，这不大可能出现内存不足的问题，因为在 INTERSVR 执行期间，服务器不能执行其他任何命令。假如串行口之一连有鼠标，则最好避免这种搜索，因为在这种情形下 INTERSVR 会破坏鼠标驱动程序。假如 COM1 端口上接有鼠标，想用 COM2 和客户机相联，则应该键入如下命令：

```
INTERSVR /COM2
```

如果是 COM1 上连有鼠标而用 LPT2 和客户机连接，则可以用选项/LPT 以使 INTERSVR 只搜索并行口，或者干脆就指明/LPT2。

当使用串行口连接时，INTERSVR 传送数据的速度是 115200 波特率。假如这个速度对自己的计算机来说太快了，就可以用/BAUD 选项来设置一个较低的速度。

提示： 如果连接时计算机出现死机，这可能是由于 INTERLNK 和计算机的时钟发生冲突，可以在下一次连接时加/V 选项试一试。

现在便可以建立服务器了：

1. 如果缆线连接到 COMn，则键入 INTERSVR/COMn。
2. 如果缆线连接到 LPTn 上，则可以键入 INTERSVR/LPT 或者 INTERSVR/LPTn。

图 41.3 示出了在第一次装入 INTERSVR 但还没有建立起连接时服务器屏幕上的显示内容，所显示出的驱动器和端口将用于重新定向，但还没有接收到客户机分配给它们的名字。在 INTERLNK 建立连接后，有关客户机的那一列也将填上内容。请注意屏幕右下角的 Alt+F4 命令，它可以用于中止连接。

41.5 建立客户机

图 41.4 示出了插入文件 CONFIG.SYS 中用于安装 INTERLNK.EXE 的命令，INTERLNK.EXE 驱动程序。INTERLNK 一旦装入，它就开始搜索服务器。假如找到了，就建立两机的连接，随后就可以访问服务器系统了；如果没有找到，它仅仅把自己装好然后等待。假如希望 INTERLNK.EXE 只有在找到服务器后才装入，可以使用/AUTO 选项。

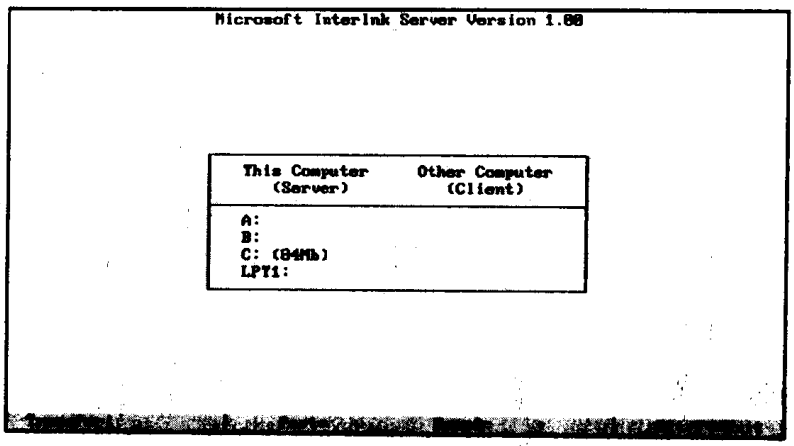

图 41.3　INTERSVR 屏幕

INTERLNK.EXE

建立客户机和服务器之间的连接，重新定向服务器的驱动器。

句法：

DEVICE[HIGH]=[path] INTERLNK.EXE [/DRIVES:n] [/NOPRINTER]
[/COM[n]|/LPT[n]] [/AUTO] [/NOSCAN]
[/BAUD:rate] [/V]

参数和选项：

path	指明 INTERLNK.EXE 文件的存放位置，缺省时是引导盘的根目录。
/DRIVES:n	指明重定向给客户机的服务器的驱动器数，缺省值是 3。如果 n=0，INTERLNK 只重定向打印机。
/NOPRINTER	禁止打印机重定向。缺省时，INTERLNK 重定向所有可用的并行打印机端口。
/COM[n]	指明客户机上的与服务器连接的通信口，如果没有指定 n，INTERLNK 将对所有的串行口进行搜索，以确定连接的位置。
/LPT [n]	指明客户机上的与服务器连接的通信口。如果没有指明 n，INTERLNK 将对所有并行口进行搜索，以确定连接的位置。
/AUTO	使用该选项时，INTERLNK.EXE 只有在找到服务器后才把自己装入内存。缺省时，不论是否能建立连接，INTERLNK.EXE 都会装入内存。
/NOSCAN	把 INTERLNK.EXE 装入内存，但不建立连接。缺省时，INTERLNK.EXE 一装入内存便开始试图与别的计算机建立连接。
/BAUD:rate	设置串行通信的最大波特率。可用的值有 9 600，19 200，38 400，57 600 以及 115 200。缺省值为 115 200。
/V	防止和计算机的时钟冲突。如果两台计算机通过串行口连接，在用 INTERLNK 接通驱动器或打印机端口时，假如两台计算机之一出现死机，就应当用该选项。

图 41.4　INTERLNK.EXE 命令总结

注释： 在通常的 CONFIG.SYS 文件中插入 INTERLNK.EXE 时可能改变后面的其他驱动程序的设置，例如改变 RAMDRIVE.SYS 和 DRIVER.SYS 命令的设置，如果不把它放在它自己的位置上，就最好把它放到 CONFIG.SYS 文件的最后，这样，即使改变了驱动程序的设置也不会使批文件和 DOSKEY 宏命令失效。

41.5.1 标识连接的端口

跟服务器一样，如果客户机的串行口接有鼠标，就不能让 INTERLNK.EXE 扫描所有的串行口。使用/LPT 选项使搜索限制在并行口，或者标明/COMn 以标识一个特殊的串行口。使用选项还可以节约内存空间，因为只需要装入控制相应端口的软件。如果不需要连接服务器的打印机，可以使用/NOPRINTER 选项，这样便可以节约更多的内存。

如果不知道该使用哪个端口，可以使用/NOSCAN 选项以禁止搜索。随后就得用 INTERLNK 命令建立连接。

现在可以做一些练习。先装入 INTERLNK.EXE：

1. 打开\CONFIG.SYS 文件，用 ASCII 文本编辑器进行编辑。
2. 如果缆线连接在客户机的 COMn 端口上，便在装入 EMM386.EXE 的命令后面加上下述命令：

   ```
   DEVICEHIGH=C:\DOS\INTERLNK.EXE /COMn
   ```

3. 如果缆线连接在客户机的 LPTn 端口上，便在装入 EMM386.EXE 的命令后面加上下述命令：

   ```
   DEVICEHIGH=C:\DOS\INTERLNK.EXE /LPTn
   ```

4. 存储文件，退出编辑器，重新启动客户机。

INTERLNK.EXE 装入后，它便把缆线连接的端口定位并建立和服务器的连接，这时便可以在客户机的显示器上看到如下的报告(具体计算机上的驱动器和端口名称可能与此不同)：

```
Microsoft INTERLNK Version n.nn
port=LPT1
Drive letters redirected:3 (D:through F:)
Printer ports redirected:1 (LPT2:)

This Computer      Other Computer
  (Client)           (Server)

-------          ----------
     D:    equals      A:
     E:    equals      B:
     F:    equals      C:    (360Mb MS-DOS 6.22)
  LPT2:    equals   LPT2:
```

该表说明电缆连接在客户机的 LPT1 端口上；客户机以驱动器名 D 使用服务器的驱动器 A，以名字 E 使用服务器的驱动器 B，以名字 F 使用服务器的驱动器 C，以及用名字 LPT2 使用服务器的 LPT2 端口。同时还可以从该表中看出服务器的 C 盘容量为 360 兆字节，它是用 DOS 6.22 进行分区的。在服务器的屏幕上也显示一个相应的

列表，该表的显示一直保持到连接中断为止。

注释： 只需一台计算机安装 DOS 6.0，另外一台安装 DOS 3.0 或 DOS 3.0 以的版本即可。必要时可以把必需的 INTERLNK 文件拷贝到没有安装 DOS 6.0 的计算机上，在本章后面“拷贝 INTERLNK 软件”一节将讲如何操作。

5. 尝试键入操作服务器硬盘的命令。在下面所示的两个命令中，d:代表服务器的硬盘名称(参见驱动器分配表)。

```
DIR d:
TREE d:
```

6. 现在不要急于中断连接，请先阅读下面一节。

41.5.2　重定向驱动器

缺省值时，INTERLNK 重定向服务器的三个驱动器；使用选项/DRIVE 时便需指明驱动器数目。除非 INTERSVR 命令指明了用于重定向的驱动器，INTERLNK 都将从 A 开始重定向服务器的驱动器。例如，假如只想操作服务器的 C 驱动器，就应当按如下方式使用 INTERSVR 命令：

```
INTERSVR C:
```

而 INTERLNK.EXE 命令应按下面的方式：

```
DEVICE [HIGH]=C:\DOS\INTERLNK.EXE /DRIVES:1
```

假如服务器有三个硬盘驱动器(C,D 和 E)，而用户打算都对它们进行操作但不操作软驱，那么在 INTERLNK.EXE 命令中不需要指明任何参数，因为缺省值就是 3，但必须在 INTERSVR 命令中指明这些将重定向的驱动器，如下所示：

```
INTERSVR C: D: E:
```

提示： 如果只想使用服务器的打印机端口，便在 INTERLNK.EXE 命令中写上 /DRIVES:0 选项。

假如需要在 INTERSVR 命令中包含/V 选项，也可以把它写在 INTERSVR 命令的后面。

41.6　用 INTERLNK 建立两台计算机之间的连接

如果在启动客户机时 INTERLNK.EXE 不能建立两机的连接，在服务器已建立后，便可以用 INTERLNK 命令完成这项工作(如图 41.5 所示)。

即使连接已经建立了，也可以在任何时候键入 INTERLNK 命令以改变驱动器名称的设置。例如，如果想把服务器的 C 驱动器名称改为 D 而不是 F，以便批文件能够正常工作，可以键入下述命令：

```
INTERLNK D:=C:
```

INTERLNK 显示新的驱动器名分配表以确认所作的改变。但是，不能在表中加入新的

驱动器名，只能使用原表中已存在的名字。如果想删去第二个驱动器名，键入 INTERLNK D:=，则重定向被取消。

INTERLNK

建立客户机和服务器之间的 INTERLNK 连接，重新定向服务器的驱动器。

格式：

INTERLNK [drive1=[drive2]]

参数：

none	建立连接，显示状态报告，包括当前驱动器和打印机的状态。
drive1	指定客户机给服务器驱动器(drive2)分配的名字。
drive2	指定服务器上用于重定向的驱动器名，客户机通过使用 drive1 对它进行操作。

图 41.5 INTERLNK 命令总结

客户机已经启动后如何连接两台计算机

1. 如果还没有装入 INTERLNK.EXE 程序，那么编辑客户机的 CONFIG.SYS 文件，加入装载命令。
2. 重新启动客户机，以确认 INTERLNK.EXE 已经装入。在引导过程中，INTERLNK 会显示它不能建立连接的信息。
3. 现在可以用缆线将两台计算机连接起来。
4. 在服务器上键入 INTERSVR 命令。
   在服务器屏幕上将出现一个驱动器和端口列表，但是，由于还没有连接好，还不会显示相应的新驱动器的名称和端口名称。
5. 在客户机上键入 INTERLNK 命令。
   INTERLNK 寻找缆线连接的端口然后建立连接，显示新的服务器上的驱动器和端口列表，并在服务器的屏幕上显示相应的内容。
6. 现在可以用显示在屏幕上的名字访问服务器的驱动器和端口。
7. 在服务器键盘上按下 Alt+F4，中止连接，然后取下缆线。
8. 如果需要从内存中清除 INTERLNK.EXE 程序以供他用，可以先编辑 CONFIG.SYS 文件，删除相应的装载命令，然后重新启动计算机。

提示： 在连接状态下键入没有参数的 INTERLNK 命令，则在客户机上重新显示驱动器名字列表。

如果需要在 Windows 中通过并行口打印文件，可以用 Windows Control Panel 把并行口分配给 LPTn.DOS。

不过，不能在重定向的驱动器上使用进行扇区和簇等物理操作的软件，例如不能运行 FORMAT，UNFORMAT，CHKDSK 或者 DEFRAG，甚至 UNDELETE 也被禁止使用，只能运行与目录和文件有关的软件。

41.7　中 止 连 接

如果不再需要服务器，就在服务器键盘上按下 Alt＋F4，连接便中止，服务器计算机也回到正常状态。

现在按下 Alt＋F4，看连接是否中止。

41.8　拷贝 INTERLNK 软件

可以将 INTERLNK 软件包拷贝到任何装有 DOS 3.0 或更高 DOS 版本的计算机中。既可以用磁盘传递这些文件，也可以用有/RCOPY 选项的 INTERSVR 命令完成这项工作，但这时必须用 7 线电缆或无调制解调器的缆线。当两台计算机已经用缆线连接好后，便可以在装有 DOS 6 的计算机上键入下述命令：

```
INTERSVR /RCOPY
```

选项/RCOPY 必须单独使用。INTERSVR 通过建立目标计算机而进行文件拷贝（目标计算机上必须存在 MODE 命令）。

41.9　自 测 练 习

本章已经讲明了如何通过 INTERLNK 将两台计算机连接起来。如果准备自己动手操作，可参考下面的步骤：

1. 建立客户机，装入 INTERLNK.EXE 程序。
2. 在启动客户机以前，建立服务器，这样在客户机启动过程中便可以建立连接。
3. 再试试另外的方法，这次先建立客户机后再建立服务器，这样必须使用 INTERLNK 命令才能建立连接。
4. 从客户机的 CONFIG.SYS 文件中删去 INTERLNK.EXE 命令。如果准备保存它，可以考虑设置一个选择命令或者另外的配置。

第 42 章 在 DOS 上的 Windows

简介： 用户可以在 Windows 多任务环境中使用 DOS 应用程序，这样还能利用 Windows 的增强功能。

42.1 DOS 和 Windows 相互支持

Windows 在 DOS 的支持下工作，所以本书中所讲的很多 DOS 特性也能使 Windows 更加完美。最明显的例子是 SmartDrive，它是 DOS 的磁盘高速缓存系统，它也能加快 Windows 应用程序的执行。本章第一部分将总结一些能使 Windows 更快更好运行的 DOS 特性。

很多人经常周期性地轮流换用 Windows 和 DOS 的应用程序。用户可以在 Windows 下运行大多数常用 DOS 应用程序并更好地利用 Windows 的多任务环境。也就是说，用户不必退出 Windows 就能调用 DOS 应用程序，在 Windows 工作台上就能启动 DOS，并能在 DOS 和 Windows 应用程序间来回切换。在 386 增强模式下，用户可以在 DOS 和 Windows 之间拷贝文件。本章将介绍如何在 Windows 下运行 DOS 应用程序。

本章包括以下几方面内容：

- 用 DOS 工具完善 Windows 的运行环境
- 在 Windows 下运行一个或多个 DOS 应用程序
- 在 Windows 下用 PIF 编辑器定义 DOS 应用程序的运行环境
- 当 Windows 在 386 增强模式下运行时，在 DOS 应用程序和其他应用程序之间拷贝和粘贴文件

42.2 优化 Windows

和 DOS 一样，Windows 需要尽可能多的内存。另外，由于它把硬盘当作虚拟内存，所以也需要尽可能多的硬盘空间。用户所作的任何释放内存的事情——特别是常规内存——以及硬盘空间，都有助于提高 Windows 的运行。

42.2.1 释放内存

首先，用 DOS 将尽可能多的程序清除出内存。第 29 章“内存管理”以及第 30 章“为 386、486 和奔腾机增加的内存工具”中，都说明了怎样把 DOS 移到 HMA（高内存区）以及把大多数设备驱动程序和 TSR 移到高端内存，使常规内存保持有 600KB 可用。第二，确认没有装入驱动程序、TSR 或者其他任何不需要的程序。检查 CONFIG.SYS 和 AUTOEXEC.BAT 文件，假如它们载入了不再使用的软件，就把这些装入命令删除。另外，确认没有分配多余的文件、缓冲区、驱动器名。第 21 章“系统配置”中讲了 CONFIG.SYS 文件的内容。

如用现在装有 RAM 驱动器，用户应当考虑是否还让它驻留内存。一个小于 2MB 的 RAM 驱动器对 Windows 没有好处。事实上，在使用 Print Manager 时还有危害。在运行时，Windows 需要使用这些内存空间。因此，如果没有 8MB 的内存，就不要装入 RAM 驱动器。如果有 8MB 内存，便可以装入。如果内存有 12MB 或更多，就可以把 RAM 驱动器的大小设置到 4MB。在 AUTOEXEC.BAT 文件中设置下列命令，可以把 Windows 的临时文件重定向到 RAM 驱动器(其中 E:是 RAM 驱动器的名称)：

```
SET TEMP=E:\
```

第 32 章“RAM 驱动器”讲了什么是 RAM 驱动器以及如何把 TEMP 定向到 RAM 驱动器。

42.2.2　释放磁盘空间

Windows 的对换文件和临时文件都需要磁盘空间。如果用户的磁盘空间太拥挤，就会降低 Windows 的运行速度。用户应当周期性地检查磁盘上的目录结构，删除那些不必要的文件。

提示： *在启动 Windows 之前，查看一下驱动器上形如～xxxxxxx.TMP 和～WOAxxxx.SWP 的文件，它们是 Windows 的临时文件，一旦 Windows 被挂起并重新启动计算机时它们就作废了。因此，用户可以放心地删除它们，但删除必须在 DOS 提示符下进行。重要的是：在 Windows 运行过程中绝对不要删除 .SWP 和 .TMP 文件！*

运行 SCANDISK(或 CHKDSK)程序，处理失去的链。它们能把驱动器上成千字节的空间集中在一起。第 27 章“CHKDSK 和 SCANDISK”中讲了丢失的链。

用 DEFRAG(或其他软件)重整硬盘，把空余的空间集中在一起，这样能使 Windows 更快地生成或访问临时文件和交换文件。第 33 章中讲了 DEFRAG 命令。

也可以考虑压缩硬盘上的文件，以留给 Windows 更多可用的空间。第 34 章“DriveSpace 和 DoubleSpace”以及第 45 章“再谈 DrvSpace 和 DblSpace”中讲了 DOS 的压缩功能。

注释： *用户不能压缩 Windows 的对换文件。但是如果压缩磁盘后装入 Windows，DOS 的压缩功能将为对换文件留出足够的空间。*

提示： *如果装有 Windows 程序的驱动器太拥挤而另一个驱动器拥有足够的空间，用户可以告诉 Windows 把临时文件和对换文件建立在另一个驱动器上。例如下面的命令告诉 Windows 把临时文件建在 D:\TEMP 下。*

```
SET TEMP=D:\TEMP
```

在 SYSTEM.INI 的[386Enh]部分中，下面的说明告诉 Windows 把 386 增强模式的临时交换文件放到 D 盘上：

```
PagingDrive=D
```

上面的说明项并不影响用户的永久性对换文件。

在实模式和标准模式下，Windows 为 DOS 的应用程序建立对换文件。在 SYS-

TEM. INI 的[NonWindowsApp]中，下述说明项告诉 Windows 把 DOS 对换文件放到 D:\DOSSWAP：

```
Swapdisk=D:\DOSSWAP
```

另一个释放磁盘空间的办法是把临时文件定向到 RAM 驱动器，对此前面已经谈及。

42.2.3 使用 SmartDrive

SmartDrive 的调整磁盘缓存功能对 Windows 和对 DOS 一样重要，在起动 Windows 之前一定记住先装入 SmartDrive。如果用户手中有其他磁盘高速缓存系统，使用时必须保证和 Windows 的版本兼容。

Windows 需要扩展内存，扩展内存甚至比磁盘高速缓存系统更重要。如果扩展内存不足，Windows 的运行会严重受阻，用户便不能装入更多的应用程序。在这种情况下，它将从 SmartDrive 缓冲区中借用尽可能多的内存。当用户装入 SmartDrive 时，就说明或暗示了两个内存大小。第一个是 DOS 下的磁盘高速缓存区内存大小，第二个是 Windows 下的最小高速缓存区大小。缺省值是根据计算机的内存大小，经过仔细考虑后设定的(见表 42.1)。用户没有必要另行设定。

表 42.1 SmartDrive 高速缓存区大小

扩展内存	DOS 高速缓存区	Windows 最小缓存区
1MB	所有扩展内存	None
2MB	1MB	256KB
4MB	1MB	512KB
6MB	2MB	1MB
大于 6MB	2MB	2MB

42.3 在 Windows 下使用 DOS 应用程序

如果用户既要用 Windows 应用程序又要用 DOS 应用程序，很可能首先想到要退出 Windows 后才能调用 DOS 功能。但不必这样做，因为在 Windows 下就能够调用 DOS 功能，特别是在 386 增强模式下更是如此。

在 386 增强模式下，用户可以在 DOS 功能中加进 Windows 的特性。用户可以和在 DOS 中一样，在整个屏幕上运行 DOS 应用程序，或者在一个可以规定其大小、可以移动和缩小等的窗口中运行，也可以同时运行几个 DOS 应用程序，而且即使控制已离开了这些程序，它们可能仍在继续运行。DOS 应用程序可以使用 Windows Clipboard(剪贴板)，所以用户可以在 DOS 和 Windows 之间来回拷贝数据。如果某个程序失效，用户也可以在不重新启动计算机的情况下删除它。

在实模式和标准模式下，用户只能起动一个 DOS 应用程序，而且是在满屏幕下运行。当这个程序在运行时，其他 DOS 应用程序就被挂起。一旦用户离开而切换到 Windows 应用程序时，它也被挂起。在这种方式下的好处在于当用户从一个应用程序切换到另一个应用程序时，程序从原来停止的地方开始运行，而不会从头再来。但如果用户先退出 Windows 再运行

DOS 程序，每次调用都会从头开始。

注释： 很多 DOS 应用程序也有允许用户执行 DOS 命令的功能，例如 File，Run 命令或 DOS Shell 命令。在 Windows 下运行这些程序时不能使用这些功能，因为它将阻碍整个应用程序的执行。在这种情形下，一般采用切换到别的 Windows 任务下调用不同的 DOS 命令。

42.3.1　启动 DOS 应用程序

练习在 Windows 下使用 DOS 编辑器：

1. 启动 Windows。
2. 在 Program Manager 中，拉下 File 菜单，选择 Run 项。
   打开 Run 对话框。
3. 键入 EDIT \DOS\README.TXT，按下 Enter 键。
   DOS 编辑器打开 DOS 的 README.TXT 文件。工作时编辑器占据整个屏幕。
4. 按 Alt＋Tab 键，退出编辑器，回到 Program Manager。
   注意，在工作台的左下角显示出了非常小的 MS-DOS 编辑窗口。
5. 按下 Alt＋Tab 键或者把鼠标移到编辑器小窗口那儿，双击鼠标器，又会重新回到编辑器内。
6. 退出编辑器，关闭 DOS 任务，回到 Windows 工作台。

警告： 不能在 Windows 下运行 CHKDSK、SCANDISK、DEFRAG 等能识别驱动器目录结构的 DOS 应用程序，也不能运行 SUBST、JOIN、APPEND 和 ASSIGN 等直接操作目录的应用程序。SHARE 和 FASTOPEN 也不能在 Windows 下运行。不能在启动 Windows 后才装入磁盘高速缓存系统，应当在启动 Windows 之前装入，不要在 Windows 下运行 Undelete 和 Backup 程序。
一些常驻内存的程序可通过 Windows 调用，但大多数不能。如果用户希望在 Windows 下运行 TSR，请与制造商联系弄清楚是否兼容，并弄清怎样在 Windows 下访问 TSR（常驻内存的程序）。

提示： 当任务占据满屏幕时，用户常常忘记自己处于 Windows 下。用户可以设置环境变量 WINPMT 为自己定义一个提示符，以提醒是在 Windows 下运行 DOS 应用程序。该环境变量必须在 Windows 启动之前设置好，具体用法和 PROMPT 命令相似。例如下面的命令设置一个 DOS 应用程序提示符，该提示符先显示信息 DOS running under Windows... type EXIT to return to the desktop，然后是当前驱动器和当前目录。

```
SET WINPMT DOS running under Windows... type EXIT to return to the desktop $P$G
```

如果装入 ANSI.SYS，就可以利用它的命令给屏幕加上色彩和图形，或者定位光标等。它们都可以当作 WINPMT 定义的一部分。

还有一些在 Windows 下启动 DOS 应用程序的方法：

- 从 File Manager 下选择 File 菜单中的 RUN
- 用鼠标双击该文件名
- 在 File Manager 中用鼠标双击程序名
- 在 File Manager 中把数据文件拖至程序名。(例如,将 README.TXT 拖到 EDIT.EXE 上)
- 为应用程序创建一个程序项再打开之

注释: *在 Applications 程序组内有一个 MS-DOS Editor 的程序项。*

所有这些方法的结果是一样的,但它们都存在着一个问题。很多 DOS 应用程序在启动后只显示一个信息便自动结束了。如果用前面所讲的方法之一运行下面的程序,就会看到将发生什么情况:

1. 选择文件,执行并打开 Run 对话框。
2. 键入 MEM 命令显示内存的使用情况。

   如果能在程序结束回到工作台之前读完这些信息,那么你一定是一个相当迅速的读者(或者可能计算机很慢)。

提示: *用户如果正在使用剪贴板 Clipboard,但由于内存不够,Windows 不能装入新的应用程序,则可以拷贝一个字符到 Clipboard 里面去以减少它占用的内存。有时候这样做可以使用户能够装入新的程序。*

有一种方法,可以使 DOS 应用程序在程序结束时不会自动中断任务。打开 MS-DOS Prompt 程序项(在 Main 程序组中),用户便能看见一个命令提示屏幕。用户可以在一行中运行几个程序,但必须键入 EXIT 以结束任务。试试用 MS-DOS Prompt 运行 MEM:

1. 打开 Main 程序组。
2. 打开 MS-DOS Prompt 程序项。

   这时便出现一个命令提示屏幕。注意屏幕上端方框中的信息(见图 42.1,该图为 386 增强模式下的显示)。不幸的是,这条信息在用户键入命令后便卷过屏幕了。

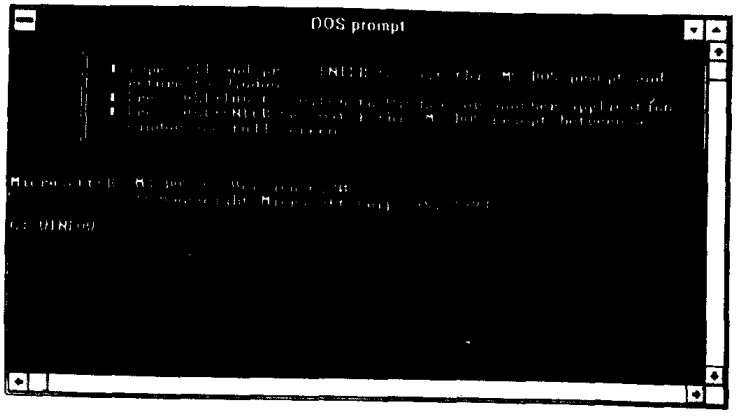

图 42.1 MS-DOS 在 Windows 下的命令提示屏幕(386 增强模式)

注释： 在实模式和标准模式下，方框中的信息不包括最后一项。因为在这两种模式下 MS-DOS 应用程序要占据整个屏幕。可以按 Alt＋Enter 键重新回到一个窗口。

3. 键入 MEM 命令。
   这时，用户就能读结果 3。
4. 试运行其他一些不影响 Windows 的命令，如 VER、BREAK、PATH 和 VERIFY。
5. 键入 EXIT 命令结束任务，回到工作台。

42.3.2　在 Windows 下执行 DOS 任务

如果是在 386 增强模式下运行 Windows，用户可以在 Windows 下用整个屏幕显示 DOS 任务，或仅仅把它们显示在一个小窗口中。如果在一个小窗口中显示，对阅读会造成一定的难度，但便于查看其他的任务以及便于在任务间切换。如果是处于 386 增强模式下，可以试试在一个窗口中运行 DOS 的 EDIT 程序。

注释： 如果用户不能肯定 Windows 所用的模式，可以拉下 Program Manager 的 Help 菜单，选择 About Program Manager 一项。在对话框的底部就显示出所用的模式。

1. 启动 DOS EDIT 程序，编辑\DOS\README. TXT。
2. 在 EDIT 下，按下 Alt＋Enter 键。
   工作台回到 Windows，但同时也在 EDIT 下(如图 42.2 所示)。

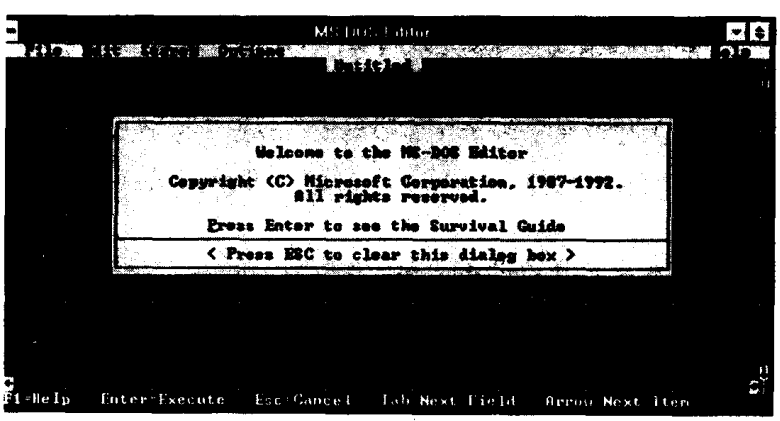

图 42.2　在 Windows 工作台上的 DOS 窗口

3. 试执行以下任务：
   移动窗口
   让窗口变小
   让窗口变大
   使它缩到最小
   恢复其大小
4. 是否注意到窗口最大时也不会充满整个屏幕？那是因为不需要整个屏幕就可以完成

所有命令的提示。试把窗口扩到最大。

窗口最大时也不会遮住工作台的显示。

5. 注意任务间切换的容易性。用鼠标单击 Program Manager 窗口。

Program Manager 窗口就变成了活动窗口。

6. 再次打开 EDIT,编辑一个新文件。

新的编辑任务占满了整个屏幕。

7. 按下 Alt+Enter 键。

现在可以看见工作台上有两个编辑任务。

8. 改变两个窗口的大小,以便能一次看见它们(提示:不能把其中一个扩到最大)。
9. 激活第一个编辑任务。
10. 按 Alt+Enter 键使它回到整个屏幕。
11. 按下 Alt+Tab 键,激活第二个任务。
12. 按下 Alt+F,X ,中断任务。
13. 激活另一个编辑任务,并且也把它中断。

注释: 在 Windows 中显示 DOS 应用程序将占用更多的系统资源(内存和处理器时间)。如果屏幕上显示"Out of Memory"的信息,请将 DOS 任务重新回到满屏幕状态下。

提示: 用户可以控制 DOS 窗口中文本显示尺寸大小,这也决定窗口的最大值。按下 Alt+Spacebar 键拉下窗口的 Control 菜单,选择 Font 一项,可以从对话框中选择 10 种字体,最小的 4×6,最大的 16×12。

答疑: 如果在 Windows 下运行 DOS 应用程序,鼠标不能工作时怎么办?

如果一个应用程序在 DOS 下运行时不能使用鼠标,那么在 Windows 下时也不能使用鼠标。但是,用户可以用鼠标拉下窗口的 Control 菜单并从中选择一些项。另外,当用户选择 Control 菜单的 Edit Mark 功能时,可以用鼠标选择文件文本并把它拷贝到剪贴板中。

如果 DOS 应用程序本身就可以使用鼠标,但在 Windows 下运行时却不行,则是由于用户的鼠标驱动程序有误,应当重新装上能在 Windows 环境中运行的鼠标驱动程序。

42.3.3 程序信息文件(PIF)

Windows 通过检查 PIF 而确定 DOS 应用程序的配置,PIF 包含了程序运行时需要多少内存以及哪些文本需要显示等信息。

Windows 把 PIF 看作是一个可执行文件。如果用户打算运行一个 DOS 程序,则可以执行 PIF 文件或程序本身,执行 PIF 和执行其他程序一样。例如,用户可以用鼠标在 File Manager 下双击程序使之运行,或者在 File 中键入它的名字,亦可编一个程序去执行它。

当用户执行一个 .COM、.EXE 或 .BAT 等 DOS 应用文件时,Windows 检查当前目录和程序搜索路径,找到有文件名字的 PIF。如果找到了,就执行 PIF;如果没有找到,就采用缺省 PIF 启动程序。缺省 PIF 称为_DEFAULT.PIF,在本章后面将会讲到。

Windows 为一些常用的 DOS 应用程序提供 PIF：EDIT. PIF、DOSPRMPT. PIF（用于前面用到过的命令提示）以及 QBASIC. PIF 和 _DEFAULT. PIF。很多新的 DOS 应用程序在它们程序包中包括了 PIF，如果用户打算在 Windows 下面运行这些应用程序，可以把它们的 PIF 拷贝到硬盘上。用户也可以定义自己的 PIF 或者编辑已存在的 PIF。用户一般用 PIF 编辑器来建立或改动 PIF。

42. 3. 4　PIF 编辑器

图 42. 3 示出了一个在 PIF 编辑器中的 PIF 例子。在本例中，PIF 编辑器处于 386 增强模式下，所编辑的是 EDIT. PIF，其中给出了 MS-DOS Editor 的配置。图 42. 4 是当用户选择 Advance 按钮时，显示的 Advanced 对话框。

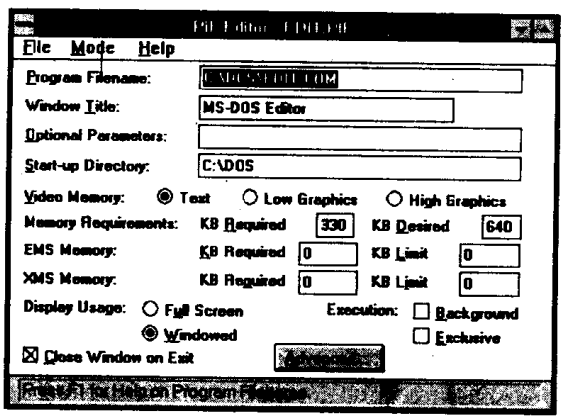

图 42. 3　PIF 编辑器中的 PIF 实例（386 增强模式）

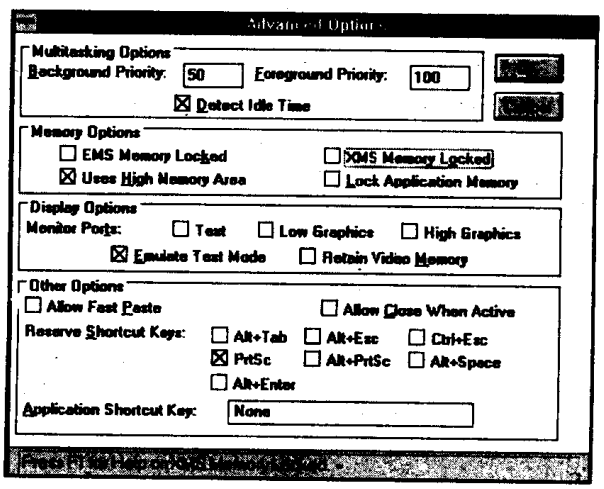

图 42. 4　在 386 增强模式下的 Advanced PIF 选项

用户可以练习使用 PIF 编辑器：

1. 打开 Main 程序组，运行 PIF Editor。
   屏幕上出现 PIF 编辑器窗口，但没有 PIF 文件被打开。
2. 选择文件并打开。

Open 对话框中列出了所有当前目录(Windows)下的 PIF 文件。

3. 用鼠标选择 EDIT. PIF 文件。

PIF 编辑器窗口中便显示出 EDIT. PIF 中的信息。如果用户是在实模式或标准模式下运行 Windows,则显示的信息和图 42. 3 及图 42. 4 中显示的有所不同。

42. 3. 4. 1 Program Filename 字段

PIF 中所必需的内容仅仅是 Program Filename(程序文件名)字段的内容,在执行 PIF 文件时它告诉 Windows 要运行的程序。它可以引用批文件(. BAT)或一个可执行文件(. COM 或 . EXE)。但必须全部写出程序的文件名说明符,包括路径和扩展名。如果程序是处于 Windows 下面或处于搜索路径中的目录下,就可以省略路径说明。

如果可能的话,最好给 PIF 文件取和程序名相同的名字,比如 EDIT. COM 的 PIF 文件名为 EDIT. PIF。如果用程序名执行一个 DOS 应用程序而不是用 PIF 文件——例如在 File Manager 窗口下用鼠标双击 EDIT. COM——Windows 将寻找同名的 PIF 文件。如果没有找到,就用缺省 PIF 文件(称为_DEFAULT. PIF),这有可能不是 DOS 应用程序所需的最好配置。

提示: 用户可以在该字段中设置环境变量而不是文件名,当 Windows 执行 PIF 时它将会使用环境变量的值作为程序文件名。

例如,假设用户根据需要用 EDLINE. EXE、EDIT. COM 或 NED. EXE 编辑文本文件,则可以在程序名(Program Filename)字段中写上%EDITOR%。下面写在 AUTOEXEC. BAT 中的命令将用来设置 EDITOR 环境变量,缺省值是 EDIT. COM。

```
CLS
ECHO Text editors...
ECHO.
ECHO        A EDIT. COM
ECHO        B EDLIN. EXE
ECHO        C NED. EXE
ECHO.
CHOICE /C:ABC /T:A,5 Choose an editor
IF ERRORLEVEL 1 IF NOT ERRORLEVEL 2 SET EDITOR=C:\DOS\EDIT. COM
IF ERRORLEVEL 2 IF NOT ERRORLEVEL 3 SET EDITOR=C:\DOS\EDLIN. EXE
IF ERRORLEVEL 3 IF NOT ERRORLEVEL 4 SET EDITOR=C:\DOS\NED. EXE
IF ERRORLEVEL 4 SET EDITOR=C:\DOS\EDIT. COM
```

当然,用户可以在 AUTOEXEC. BAT 执行完毕后重新设置 EDITOR,但必须在启动 Windows 之前设置。在启动 Windows 以后,用户只能在第二个命令提示符后键入 SET 命令。当用户离开第二命令环境后,该变量就会被访问(第 43 章"DOS 及其运行环境"解释了第二命令提示符和环境变量)。

42. 3. 4. 2 Window Title 字段

Window Title(窗口标题)字段中的内容是执行 PIF 文件时显示在窗口标题栏的内容。如果用户不提供标题,Windows 就用程序的名字(没有扩展名)。如果用户在执行 PIF 的文件中加了对程序的说明,则该说明就将覆盖该字段。

提示: 用户可以在该字段中说明一个环境变量而不是标题。当 Windows 执行该

PIF 文件时就把该变量当作标题。

42.3.4.3 Optional Parameters 字段

用 Optional Parameters(可选参数)定义命令中的参数。例如,如果要建立一个 PIF 文件来编辑\DOS\README.TXT,则在该字段中键入\DOS\README.TXT。

如果想在 PIF 执行时在对话框中显示参数,在这个字段中输入?。如果在执行该 PIF 的程序项的属性中也有可选参数,这将覆盖用户所输入的可选参数。假如用 File、Run 执行 PIF,则包含在命令行中的参数也会覆盖用户输入的参数。

提示: 用户可以在该字段中说明一个环境变量而不是参数和选项,在 Windows 执行 PIF 时,它将用该变量的值作为可选参数。

42.3.4.4 Start-up Directory 字段

Start-up Directory(启动目录)字段中说明 Windows 启动程序以前应该转换到的目录。如果执行该 PIF 的程序项中包含了 Start-up Directory,则会覆盖用户输入的目录项。

提示: 用户可以在这个字段内说明环境变量而不是目录路径名。在执行 PIF 文件时,Windows 把这个变量的值当作目录路径。

42.3.4.5 Video Memory 字段

Video Memory(视频内存)字段中说明当离开任务时保留多少内存来存放屏幕上的内容。文本方式要求的内存最少,而高分辨率图形方式所需的内存最多。一般说来尽可能少地保留内存而把其他内存释放以供它用,因此,建议用户选择 Text(文本)方式,除非用户知道的确需要 High Graphics(高分辨率图形)或 Low Graphics(低分辨率图形)方式。如果用户发现切换任务时出现障碍,或者一部分显示的内容丢失,或者屏幕上出现 Out of Memory(内存耗尽)的信息,那么就中止该应用程序然后再换一种高一些的显示方式。

在 386 增强模式下,PIF 的一个高级选项是 Retain Video Memory(保留视频内存)。如果用户激活这个选项,Windows 就不会把视频内存中的内容存放到磁盘上的对换文件中,而当用户再回到这个任务时,也不需要再把这些内容从磁盘上读回来。这有利于更快的切换任务。但是,它释放的内存相应也少。因此用户必须知道是否有足够的扩展内存来这么做。如果用户这么做了但发现执行其他任务时有困难,就不能激活这个选项。

42.3.4.6 内存说明字段

这一组字段用于说明需要多少常规内存、扩充内存(EMS)和扩展内存(XMS)。对于每种类型的内存,用户都可以取其最大值或最小值。表 42.2 示出了这些内存的缺省值。

表 42.2　Windows PIFs 的缺省内存说明

Type	所需的内存数	KB
常规内存	128KB	640KB
扩充	0KB	1024KB
扩展	0KB	1024KB

注释： 在 386 增强模式下，在程序需要时，Windows 自动把扩展内存转变为扩充内存。所以如果 PC 没有扩充内存时，只要有扩展内存，程序就可以把所有扩展内存当作扩充内存使用。

在实模式和标准模式下，由于一次只能运行一个 DOS 应用程序，而且 Windows 把所有的扩充内存都分配给它使用，所以 PIF 中不出现 EMS 设置项（所谓扩充内存，是装在扩充板上的内存，由扩充内存驱动器管理的内存）。在实模式和标准模式下，Windows 不会把扩展内存转变为扩充内存（EMM386. EXE 被忽略）。

在最小设置情况下，称为 KB Required（所需的内存数），不需要决定分配给程序多少内存。如果内存充足的话，程序运行时需要多少内存，Windows 就分配给它多少内存（包括常规内存和扩充内存）。但是如果内存不够，Windows 就分配给程序尽可能多的内存。假如内存设置得太少，那么 Windows 就不会装入任何程序并且显示内存不足的信息。如果内存实在太小，Windows 在试图装入程序并执行时就会死机。

最大设置（称为 KB Desired 或 KB Limit）决定分配给程序的最大常规内存、扩充内存和扩展内存。－1 表示允许程序占用尽可能多的内存，直到可用的内存用完为止。建议在 KB Desired 字段中填－1，表示分配给程序尽可能多的常规内存，这可以让 Windows 把其他内存释放出来供程序使用。在一般情况下，在 EMS Memory 和 XMS Memory 项中填 0，当然，如果用户知道的确需要扩充内存和扩展内存，就不能填 0。

注释： 一些在 DOS 下运行时需要扩展内存的应用程序，在 Windows 下运行时却不能用扩展内存，因为它们不符合 Windows 访问扩展内存的标准。如果用户手中有这样的应用程序，最好在 DOS 下运行。

若选用 386 增强模式的 Advance Options 等选项，用户可以防止 Windows 把应用程序的一些内存中的内容输送到磁盘上的对换文件中去。这些选项称为 EMS 内存锁定（EMS Memory Lock）、XMS 内存锁定（XMS Memory Lock）和应用程序内存锁定（Lock Application Memory）（对于常规内存而言）。激活这些选项将加速任务对换操作，但会抢夺其他任务的内存，因而也会使 Windows 的运行减慢或阻碍其他应用程序的装入。在一些情况下，锁定扩展内存有损一个应用程序的运行，因为它需要的内存大于它所能得到的，因而必须在内存和硬盘之间不停地交换数据，以提供额外的内存。

高址内存区使用选项（Use High Memory Area）也出现在 Advance Options 对话框的同一部分。这一选项使用缺省值。如果一个应用程序自己知道如何使用 HMA（高址内存区），该选项又是激活了的，而且 HMA 又没被使用，那么该应用程序就会访问另外 64KB 的内存空间。这三个条件一般不会同时存在，但激活该选项不会有害处。

42. 3. 4. 7 Display Usage 字段

Display Usage 字段决定应用程序是占用满屏幕还是只占用一个窗口。窗口方式要用更多的系统资源，但方便于多任务状态。不管选择何种方式，都可以按 Alt＋Enter 键在这两种方式间进行相互切换。

注释： 一些应用程序只能以全屏幕方式运行（因为在启动时常常要显示标志图形或其他图形）。如果用户试图把它转换为窗口方式，就会出现错误信息。

42.3.4.8　执行和多任务优先权

Execution 选项决定在另外一个程序激活后，原来的应用程序是否还执行下去。例如，如果正在运行的程序是涉及长时间计算、排序或数据传送的程序，用户可能希望在该程序运行过程中能在计算机上做其他工作(由于原来的程序运行时用掉了一部分系统资源，所以新的工作进行得相当慢，但这总比坐着等强)。要想在一个应用程序正在运行过程中执行另一个程序，就激活 Background 选项。

如果想把所有的控制权交给一个应用程序而使其他应用程序不能在后台执行，就激活 Exclusive 选项。该选项对一些对时间比较敏感的程序，如游戏和电话通信程序等来说十分重要，这些程序在和别的程序分时运行的情况下就不能正常工作。但是，在大多数情况下，把 Foreground Priority(前台优先权)设置到最大值 1000 是一个较好的办法，它可以给应用程序尽可能多的时间。利用 Foreground Priority 选项可以让 Windows 在程序处于空闲状态(亦即等待用户)的时候，把时间分配给其他任务。

在 Advanced Options 对话框中，Background Priority(后台优先权)和 Foreground Priority 决定 Windows 把多少时间分配给任务(相对于其他正在运行的程序而言)。最小值是 0 而最大值是 10000。缺省时，Background Priority 是 50，而 Foreground Priority 是 100。

任务的优先权数决定它在任务列表中的位置，并不是要求一定要分享多少时间百分数。如果任务 A 的优先权数为 50 而任务 B 的优先权数是 100，而且不再有其他任务，那么任务 B 的优先权最高。即使它的优先权是 10000，它也不会比优先权是 100 时分享更多的时间。当有很多任务同时执行时，优先权数为 10000 者的最优先。

Detect Idle Time(检测空闲时间)选项使 Windows 能停止把时间分配给那些当前没有做任何事情的程序——例如，这个程序仅仅等待用户去激活它并输入命令。Detect Idle Time 选项能使其他应用程序运行更快，但也许会给空闲着的程序带来问题，因为一些程序要周期性地更新时间，如同带有一个定时闹钟。一些新的 DOS 应用程序知道了 Windows 这一特点，如果它们需要处理时间，就会向 Windows 发出忙的信号。一般情况下，用户都应当激活这个选项，除非知道应用程序和它不兼容。

提示： 用户可以在应用程序正在运行的时候改变其执行配置。拉下 Control 菜单，选择设置以打开对话框(如 42.5 所示)，于是用户能选择 Exclusive(独占方式)或 Background(后台方式)选项，并且改变应用程序的优先权。

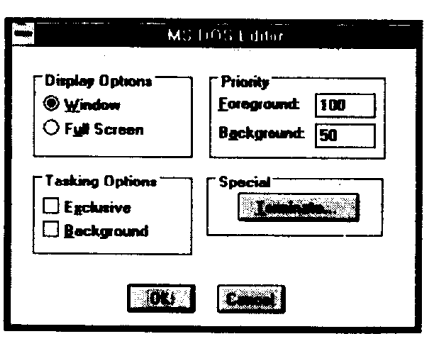

图 42.5　在 Windows 下运行的 DOS 应用程序的配置对话框

42.3.4.9 Close Windows On Exit 选项

用户已看到过 Close Window on Exit(退出时关闭窗口)选项的结果。如果用户是在为一个显示最后信息的应用程序建立 PIF 文件,那么关闭这个选项,使 DOS 提示符停留在屏幕上,直到用户键入 EXIT 命令。

提示: 如果用户在装入 DOS 应用程序时有问题,可以关闭 Close Window on Exit 选项,以便能看见应用程序在结束之前显示的信息。

42.3.4.10 Display Options 选项

在 386 增强模式下,Advanced Options 对话框中的 Display Options 部分包含了与视频监视器有关的几个选项。其中 Retain Video Memory(保留视频内存)选项已经在 Video Memory 部分讲过了。

Monitor Ports(监视端口)选项决定 Windows 是否装入额外的 TSR 来监视视频卡。当视频卡为 EGA(不是 VGA 或更高)时只有很少一些应用程序需要这些选项。如果用户在激活其他任务后再回到原来应用程序时,屏幕上显示出无意义的字符,则需要激活这个选项,否则一般不需要激活它。在这种情况下,还可以通过选择 High Graphics(高分辨率图形)项来解决问题。

注释: 监视端口严重降低 Windows 的运行速度,除非不得已,不要选择该选项。

Emulate Text Mode(仿真文本方式)选项可以使文本方式的应用程序运行更快。只要不是显示器或鼠标工作不正常,在文本方式的应用程序中都可以激活这个选项。

42.3.4.11 拷贝和粘贴

在 386 增强模式下,用户可以通过使用 Windows Clipboard(剪贴板),在 DOS 应用程序和 Windows 应用程序之间拷贝和粘贴文本。如果用户发现在把文本粘贴到一个特定的 DOS 应用程序时出现文本丢失,就得针对此应用程序关闭 Allow Fast Paste(允许快速粘贴)选项。否则让它处于激活状态,以利于提高粘贴速度。

下面的练习将给用户展示怎样在 Windows 下运行的 DOS 应用程序中拷贝和粘贴文本。如果不是处于 386 增强模式下工作,则应跳过该练习。

1. 启动 Applications 程序组中两个 MS-DOS Editor 程序项拷贝。
2. 在一个 Editor 项中,打开一个文件,取名\DOS\README. TXT。
3. 按下 Alt+Spacebar 键,弹出 Editor 项的 Control 菜单。
   即使该项占据了整个屏幕,当在 Windows 下运行时它都有 Control 菜单。按 Alt+Spacebar 键,切换回窗口方式,拉下 Control 菜单。
4. 从 Control 菜单中选择 Edit。
   子菜单出现。
5. 从子菜单中选择 Mark 项。
   标题栏中便显示 Mark MS-DOS Editor。该标题告诉用户应用程序的功能已被覆盖,而且正处于 Windows 的拷贝功能中。
6. 拖动鼠标标记一部分文本。
   在标记文本的时候,标题栏便显示 Select MS-DOS Editor。

7. 按 Enter 键，把选择的文本拷贝到 Windows 剪贴板。

   文本上的标记消失，标题栏回到原来状态。

8. 激活其他 Editor 任务。

9. 把光标移动到要粘贴文本的地方。

10. 用鼠标拉下 Control 菜单或按 Alt+Spacebar 键。

11. 选择 Edit。

    出现子菜单。注意 Paste 命令已能调用。

12. 选择 Paste。

    Windows 把文本从剪贴板粘贴到文件中光标所在的位置。

13. 关闭两个 Editor 任务，但不要存盘。

注释： 与 Windows 应用程序不同，在选择文本时用户不能滚动 DOS 窗口，所以一次最多只能选择一整屏幕的文本。

42.3.4.12　Allow Close when Active 项

缺省时，用户不能在没有正常退出的情况下关闭 DOS 应用程序的窗口。例如，不能弹出窗口的 Control 菜单并选择 Close。用鼠标选择窗口的 Control 菜单也无效。在程序正常关闭所有已打开的文件之前，不允许用户中断程序运行。

但是，一些应用程序并不进行文件操作，因此也用不着这样的保护。如果能从外界关闭该应用程序窗口而不通过先激活它并正常退出后再关闭，则整个操作过程要快一些。例如，如果用户建立了一个 PIF 来执行 MSD 或 HELP，则可能不需要退出该应用程序就能安全关闭其窗口。激活 Allow Close when Active(允许激活时关闭)选项，就可以通过关闭其窗口而中断应用程序。

42.3.4.13　为应用程序的使用预留快捷键

表 42.3 中示出了 Windows 自己给一些键的功能定义，这些功能将覆盖所有应用程序对这些键所分配的功能。Windows 应用程序自己避免了这些键，但 DOS 应用程序可能与 Windows 不一样，它可能把其中之一或几个键用作其他的目的。

表 42.3　Windows 的快捷键

快捷键	功能
Alt+Enter	在全屏幕和窗口方式之间来回切换 DOS 应用程序的运行方式
Alt+Esc	在任务列表中从上一个任务切换到下一个任务
Alt+Tab	在任务列表中切换到下一个任务
Ctrl+Esc	显示任务列表
Alt+Spacebar	弹出窗口控制(Control)菜单
PrintScreen	把屏幕上的内容拷贝到 Windows 的剪贴板上
Alt+PrintScreen	把活动窗口的内容拷贝到 Windows 剪贴板上

如果应用程序用了其中一个或几个键，则用户可以通过激活 PIF 窗口中 Reserve Shortcut Keys 项下的适当选项，以便覆盖 Windows 的功能定义。

提示： 为了让 DOS 应用程序可以在打印机上打印屏幕，而不是把屏幕上的内容写到 Windows 的剪贴板上，就得激活 Reserve Shortcut Keys(预留快捷键)下的 PrtSc。用户仍然可以通过使用 Alt+PrintScreen 键把活动窗口的内容加到剪贴板上。

42.3.4.14 为应用程序指定快捷键

用户可以给应用程序指定一个快捷键。当应用程序在后台运行时，按下该键就可以激活应用程序。要指定快捷键，先用鼠标选择 Application Shortcut Keys(应用程序快捷键)的文本框，并键入用户想用的组合键形式。用户可以用 Ctrl 和/或 Alt 与其他所有的字符键和功能键组合，甚至也可以和 Shift 键组合。例如 Ctrl+F5，Alt+Shift 或者 Alt+Ctrl+P。如果用户所键入的组合键无效，在准备关闭 Advanced Options 对话框时 Windows 将显示出警告信息。

提示： 一旦用户定义了快捷键，就不能简单地认为把它从文本框中抹去就失效了。可以把光标移到框中并按下 Shift+Backspace 键，框中显示字 None，便表示先前所定义的快捷键被消除。

一个应用程序的快捷键可能会和使用这些键的其他一些应用程序冲突。例如，在 Windows 中，Ctrl+I 表示使用斜体或不使用斜体。虽然还有其他一些方法用于激活斜体，但用 Ctrl+I 是最方便的方法之一。假如用户在 Windows 下运行一个 DOS 应用程序，而 Ctrl+I 又是它的一个快捷键，那么，在这种情况下，Ctrl+I 就不再具有选用或不选用斜体的功能了，它仅让用户切换回 DOS 应用程序。很多用户会发现这很不方便。

要避免这种冲突，在定义快捷键时，尽量不要选用那些 Windows 用了的键组合。一般采用比较偏的组合形式比较安全，例如使用 Ctrl+Alt 再加上一个符号键如 * 或[等。也许用户会发现用其他方法切换任务会更方便。

42.3.5 实模式和标准模式下的 PIF 选项

如果 Windows 程序在实模式或标准模式下运行，PIF 窗口所显示的内容有所不同，例如图 42.6 所示。有几个选项和 386 增强模式相同：Program Filename(文件名)，Window Title(窗口标题)，Optional Parameters(可选参数)，Start-up Directory(启动目录)，XMS Memory(扩展内存)和 Close Window on Exit(退出时关闭窗口)。有三项相似，但选择范围要小些：Video Mode(视频方式)，Memory Requirement(内存需求)以及 Reserve Shortcut Keys(预留快捷键)。很多选项，例如 Display Usage 和 Execution 不能用，因为在实模式和标准模式下，一次不能运行几个 DOS 应用程序。另一方面实模式和标准模式下的 PIF 也有几个 386 增强模式所没有的选项，这些选项将在下面几段中讲述。

如果用户的系统有时需要在实模式或标准模式下运行，有时又需要在 386 增强模式下运行，这时候往往希望建立的 PIF 可以在两种模式下都能运行。在 PIF 编辑器窗口中，可以用 Mode 菜单在两种模式间切换。这只改变 PIF 编辑器窗口，而不是当前 Windows 的运行模式。可试试下面的模式切换：

1. 启动 PIF 编辑器。
2. 拉下 Mode 菜单。

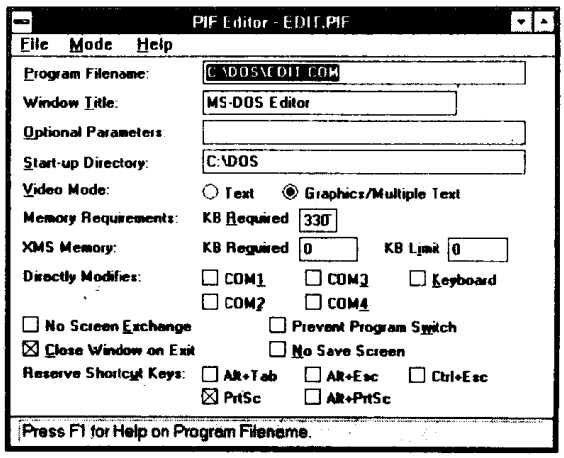

图 42.6　PIF 编辑器中的 PIF 文件实例(实模式和标准模式)

当前模式被选择符号标记。注意 Standard 意味着实模式和标准模式两种方式。

3. 选择另一种模式。

对话框中将警告用户所选择的模式与当前模式不符。

4. 选择 OK。

窗口中变成所选择模式下的 PIF 格式。

当用户改变模式时,PIF 编辑器记住 Program Filename,Window Title,Start-up Directory 和 Close Window on Exit 等项中的内容,其他一些共同的参数,如 Optional Parameters 和 XMS Memory 等中的内容不被记忆下来,因此改变模式时必须重新定义。

注释： 当使用 File,Save 或 File,Save As 等命令存储 PIF 时,两种模式都被存储下来。如果只编辑了其中一种模式,在存储时另一种同样也被存储,不过采用的是缺省值。

42.3.5.1　存储屏幕映象

在实模式和标准模式下,只能使用两种显示方式:文本方式和图形/多重文本方式。如果应用程序要在视频内存区中存储多页文本,(这样做可使显示操作更快地进行),就必须选用图形/多重文本方式(Graphics/Multiple Text)选项,以便能让系统保留足够的内存空间来存储这几页文件,在用户离开该应用程序时不至于丢失。注意在实模式和标准模式下没有保留视频内存区的选择项,在用户离开应用程序时,视频内存区将被刷新,所以分配足够的空间存储视频内存中的内容是很重要的。

如果用户是为一个以文本操作为主的程序定义 PIF,而且不确定它是否以多页方式工作,就先选择文本方式。如果发现在离开该程序或返回时出现数据丢失等反常情况,就需编辑 PIF,把显示方式改为图形/多重文本方式。

很多新版本的 DOS 应用程序与 Windows 兼容,因而不需要存储它们的内存映象。当它们在 Windows 下被重新激活时,会自动地刷新屏幕。通过选择 No Save Screen 项,可以告诉 Windows 在任务切换时不必再存储或恢复屏幕映象,从而节约了时间和内存空间。

注释： 如果回到应用程序时是一个空屏,可禁止应用程序的 PIF 中的 No Save

Screen 项。

42.3.5.2 使用直接访问 COM 端口和键盘的应用程序

和 Windows 不太兼容的应用程序可能会认为它们自己取得了整个 PC 的控制权，因而会直接访问监视器、键盘和端口。Windows 管理时要访问监视器和并行口，因此在应用程序要访问 COM 端口或键盘时，必须告诉 Windows。选择 Directly Modifies 部分中的适当选项，便可以警告 Windows 在使用同一外设时，可能会和其他程序发生冲突。

在访问键盘时，任何直接访问键盘的程序都会使 Windows 的快捷键失效，而不管用户是否激活了键盘选项。激活选项的结果仅仅是节约了一些内存空间，因为它告诉 Windows 不必装入那些操作快捷键的功能块。

42.3.5.3 在实模式和标准模式下打印屏幕

在实模式和标准模式下，按 Print Screen 键把整个屏幕的内容拷贝到 Windows 的剪贴板上，而按 Alt+PrintScreen 键仅把窗口中的内容拷贝到剪贴板上。在前面已经讲过了如何覆盖这些功能而让这些键恢复常用的 DOS 中的功能(见 Reserve Shortcut Keys 项这一部分)。如果用户不必把屏幕拷贝到剪贴板上，可以激活 No Screen Exchange 选项。激活该选项还可以节约部分内存空间。

42.3.5.4 阻止程序切换

激活阻止程序切换选项(Prevent Program Switch)可以节约一小部分内存空间。因为该选项被激活时，用户在回到工作台和其他应用程序时，必须退出现在的应用程序。

42.3.5.5 PIF 缺省值

当在 Windows 下装入没有 PIF 的 DOS 应用程序时，Windows 便使用 PIF 的缺省值。图 42.7 显示了缺省 PIF，其名字是_DEFAULT.PIF。程序文件名_DEFAULT.BAT 是一个哑名，其实没有 DEFAULT.BAT 这个文件。哑名仅仅表示在任何 PIF 中，该项中必须有一个值。另外，缺省 PIF 使用所有 PIF 值。如果改变该 PIF 中的任何值——例如改变 Display Usage 中的内容为窗口方式(Windowed)——都会影响使用 DEFAULT.PIF 的任何应用程序，因为它们没有自己的 PIF。

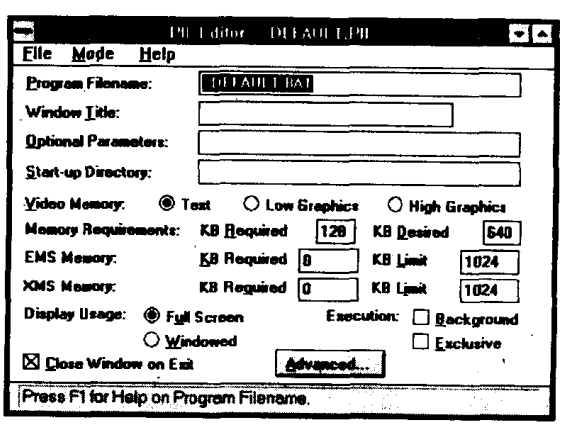

图 42.7 PIF 缺省值

42.4　自 测 练 习

这章讲了哪些 DOS 功能可以帮助完善 Windows，以及怎样在 Windows 下运行 DOS 应用程序。下面几项建议可以帮助用户在 Windows 下作使用练习：

1. 如果要在 Windows 下运行 DOS 应用程序，则检查每个程序，看看它们是否有 PIF。如果有，看一看 PIF，确定一下有没有要改变的选项。如果没有 PIF，可以考虑为其定义一个 PIF，以便在运行时不使用缺省 PIF。
2. 确信自己知道怎样做下面一些事情：
   a. 在 Windows 下启动应用程序。
   b. 在全屏幕方式和窗口方式间切换该应用程序。
   c. 在 386 增强模式下，在 DOS 应用程序和其他应用程序之间拷贝和粘贴文本。
3. 为在 Windows 下运行的 DOS 命令提示屏幕定义一个提示符，这样用户就不会忘记自己处于多任务状态下。在 AUTOEXEC.BAT 中输入一个命令，以定义每次启动 DOS 时的提示符。
4. 弄清哪些 DOS 应用程序不能在 Windows 下使用。

第 43 章 DOS 及其运行环境

简介： DOS 的运行环境使用户能够设置命名变量，从而更方便地和很多应用程序及实用程序进行通信。用户亦可在批文件中使用这些变量。

43.1 DOS 的运行环境

DOS 保留了一个内存缓冲区，称为环境，来存储一些关键的变量，例如路径和提示符。其他程序也可以在 DOS 环境中存储和访问变量，例如，Microsoft Windows 使用 TEMP 变量来指定其临时文件。

用户也可以在批处理文件中建立并访问环境变量。事实上，为了让一个批处理文件获得新的环境，有时候得启动第二个命令解释程序。

本章将阐述怎样控制和访问 DOS 环境，用户将了解：

- 显示环境中的所有变量
- 识别 DOS 自动设置的环境变量
- 设置环境变量以控制 DOS 实用程序，如 DIR 和 XCOPY
- 建立和使用自己的环境变量
- 使用 CONFIG 变量，在 AUTOEXEC.BAT 中建立可替换配置
- 控制原始命令解释程序的装入
- 控制环境的大小
- 装入第二命令解释程序
- 重定向批处理文件的输出
- 通过有条件地执行，检查一个批处理文件

43.2 环 境 变 量

环境变量都有名称和值。在下面的例子中，TEMP 是变量名，C:\DOS 是它的值：

```
TEMP=C:\DOS
```

变量名和其值都可以是任何长度，只要不超过环境中剩余的空间就行。它们也可以是任何字符，包括空格。DOS 总是以大写字符的形式来翻译和存储变量名，但对值就以输入的值为准进行存储。

43.2.1 显示环境变量

图 43.1 示出了 set 命令，该命令用于控制环境变量。没有任何参数的 SET 就显示环境中的当前内容。如果要查看一下当前环境中的变量，就键入没有参数的 SET 命令。

DOS 将显示环境中的所有变量的列表。

SET

设置和改变环境变量的值，或者显示当前值。

句法：

SET [variable=[string]]

参数：

none	显示所有环境变量的当前值。
variable	指定要设置的环境变量。
string	说明环境变量的新值，如果省略，则该环境变量的当前值取消。

图 43.1　SET 命令总结

提示： 要查看一个特定变量的值而不是整个列表，键入 SET|FIND /I "name"命令，例如 SET|FIND /I "PROMPT"命令。

43.2.2　DOS 的环境变量

DOS 使用了一系列环境变量。有一些变量是由 DOS 自动设置，另外一些则由用户设置而 DOS 仅使用它们。

43.2.2.1　PATH

当用户键入包含当前搜索路径的 PATH 命令时，PATH 变量就被设置了。如果用户不设置 PATH，DOS 就使用缺省值 PATH=C:\DOS。DOS 在搜索程序和文件时，就会访问 PATH 变量。如果环境中没有 PATH 变量，就没有搜索路径。

43.2.2.2　PROMPT

当用户键入 PROMPT 命令时，PROMPT 变量就被设置了并显示当前设置的命令提示符。当 DOS 显示命令提示符时就会自动去访问 PROMPT 变量。如果环境中没有 PROMPT 变量，DOS 便显示缺省命令提示符，缺省命令提示符就是当前驱动器名(不是当前目录)后面跟一个右方括符。PROMPT 命令和它使用的符号的含义已在第 12 章中讲述过。

43.2.2.3　TEMP

TEMP 变量必须用 SET 命令建立。该变量告诉 DOS 在哪里存储临时文件(诸如管道文件)。如果没有 TEMP 变量可用，DOS 就把它的临时文件放在当前目录下。Microsoft Windows 也使用 TEMP 变量来存放它的临时文件。Windows 需要至少 2MB 的空间来存储它的临时文件，因此如果用户把 TEMP 定向到没有足够空间的驱动器上，在使用 Windows 的打印管理器(Print Manager)子功能时便会出现障碍。

如果用户在扩展或扩充内存中建立一个至少 2MB 的 RAM 驱动器，并且把 TEMP 定向到该 RAM 驱动器上，那么那些使用临时文件的 DOS 程序将会运行得更快。采用这种方法，DOS 就可以在不进行真实磁盘访问的情形下写入其临时文件。但是，SmartDrive 能使情况变量比使用 RAM 驱动器更好，所以不要从 SmartDrive 猎取内存空间来建立 RAM 驱动

器。不过，如果用户还有其他程序不使用的扩充内存，或者系统中的扩展内存大于 4MB，在 SmartDrive 建立 2MB 高速缓存区后还剩下 2MB，则可以建立 RAM 虚拟磁盘。如果所建立的 RAM 驱动器的名字为 E，那么在 AUTOEXEC. BAT 文件中键入下面的命令：

```
SET TEMP=E:\
```

43. 2. 2. 4 DIRCMD

DIRCMD 变量必须用 SET 命令设置，该变量建立 DIR 命令的缺省参数。一旦键入 DIR 命令，DOS 便去访问 DIRCMD 变量。如果环境中没有 DIRCMD 变量可访问，DOS 就使用 DIR 的缺省参数和选项。

用户可以在 DIRCMD 中定义 DIR 选项的任何组合。很多用户喜欢 DIR 按字符顺序把整个目录中的内容进行排序，并且将所有的子目录列在最前面，如果愿意这样，就可以按下面的步骤去做：

1. 打开\AUTOEXEC. BAT 文件，准备编辑。
2. 在 AUTOEXEC. BAT 中的某个地方写入下述命令：

```
SET DIRCMD=/0
```

3. 存储该文件并退出编辑器。
4. 重新启动计算机。
5. 键入 DIR 命令，验证其结果。

43. 2. 2. 5 COPYCMD

COPYCMD 变量建立 COPY、XCOPY 以及 MOVE 命令的缺省参数。该变量最先用于设置/-Y 选项，以使 COPY 等命令忽略 DOS 的缺省复写保护。但建议用户不要这样做。即使用户临时需要忽略复写保护，可以在命令后面加上/-Y 选项。

43. 2. 2. 6 MSDOSDATA

MSDOSDATA 变量用来说明 Backup 的建立和目录文件所在的缺省驱动器，也用来说明 Microsoft Anti-Virus 的 MSAV. TXT 文件所在的缺省目录。在第 23 章“MSBACKUP 和 MWBACKUP”中解释了怎样用 MSDOSDATA 来备份文件，在第 25 章“DOS 的反病毒程序”中解释了怎样在 Microsoft Anti-Virus 中使用它。

43. 2. 2. 7 COMSPEC

COMSPEC 变量由 DOS 设置，用来展示命令解释程序的存放地点。在一些其他程序覆盖了命令解释程序的常驻内存部分，而 DOS 又需要将这部分重新装入时，就会去访问 COMSPEC 变量。在本章后面将会看到 DOS 怎样根据该变量的值作出决定。

43. 2. 2. 8 CONFIG

当 CONFIG. SYS 使用启动菜单和多配置块时，DOS 设置 CONFIG 变量来给在启动过程中所选中的块命名。用户可以通过在 AUTOEXEC. BAT 中访问该变量来剪裁配置的其余部分。在本章后面“使用 CONFIG 变量”一节中会讲些使用 CONFIG 的例子。

43. 2. 3 建立环境变量

下面的命令建立一个名叫 TEMP 的环境变量，并把它的值设置成 C:\DOS：

```
C:\> SET TEMP=C:\DOS
```

因为变量名和值都可以有空格，因此必须保证在等号前后没有空格，除非用户希望在名和值中包含空格。DOS 把“TEMP”和“TEMP　”(有一个尾随的空格)看作两个不同的变量。当寻找存放临时文件的地方时，DOS 访问“TEMP”而不是“TEMP　”。

试建立一个名为 MYNAME 的环境变量：

1. 键入 SET MYNAME=name 命令，其中 name 是自己的名字。
2. 键入 SET 命令，查看新设置的变量。

在环境空间用完之前，SET 命令不返回任何确认信息。如果环境空间用完了，就显示下列信息：

```
Out of enviroment space
```

在本章后面将阐述如何扩大环境空间。

43.2.4　改变和删除环境变量

要改变一个环境变量的值，只需在 SET 命令后面跟上相同的变量名和一个新的值。要删除一个环境变量，在 SET 后面写上环境变量的名字，但在等号后面什么也不写，如下面的命令所示：

```
SET TEMP=
```

注释： 重新启动计算机时将删除所有的环境变量，但那些由 DOS 自动设置以及在 AUTOEXEC.BAT 中设置的除外。

43.2.5　访问环境变量

用户可以在批处理文件中，用与可替代参数相仿的方法访问环境变量，例如下面的格式：

```
%variable-name%
```

例如，如果在批处理文件执行过程中把目录转到名为 TEMP 的目录下，则可使用下述命令：

```
CD% TEMP%
```

注释： 不能在命令提示符下或在 DOSKEY 宏命令中访问环境变量。

试编写一个简短的批处理程序来访问 MYNAME 变量：

1. 通过文本编辑器如 EDIT 打开一个名为 HELLO.BAT 的新文件，准备编辑。
2. 写入下述程序：

```
@ ECHO OFF
ECHO Hello %MYNAME%
```

3. 存盘并退出编辑器。
4. 键入 HELLO，执行该批命令。

DOS 会显示 HELLO name。

43.2.5.1 附加一个变量

当一个批处理文件在用户的系统上建立一个新程序或一个新的功能时，它可能需要附加一些环境变量，如 PATH、PROMPT 或 COPYCMD，但不彻底替换当前的变量值。用户可以通过在 SET 命令中包含变量参数，而把变量的当前值合并到新的值中去。例如，(这也是常用的技巧)下面的命令用 PATH 变量把 C:\PROGS 目录加到当前搜索路径的末尾：

```
PATH=%PATH%;C:\PROGS
```

表达式%PATH%使 PATH 变量的当前值(当前搜索路径)被填到命令中去，然后把 C:\PROGS 附在它后面。用户亦可很方便地把 C:\PROGS 插到当前搜索路径的前面。

注释： 很多应用程序的安装程序使用这个技巧把应用程序所在的目录路径加到搜索路径中去。一些应用程序把目录加到搜索路径的前面而不是最后，如果不符合自己的要求，用户可以检查 AUTOEXEC.BAT 中的 PATH 命令，然后作调整以适应系统的要求。

43.2.5.2 保存和恢复环境变量

一些批处理命令需要临时性地改变环境变量的值。例如，它可能需要临时改变程序的搜索路径。一个比较优秀的程序在中止前会自己恢复环境变量先前的值。用户可以在运行程序之前把环境变量的当前值存储起来，并在程序结束后把它们恢复。下面的例子说明了怎样保存和恢复 COPYCMD：

```
REM Save the current prompt:
SET SAVECOPYCMD=%COPYCMD%
REM Make sure we didn't run out of environment space
IF %SAVECOPYCMD%=%COPYCMD% GOTO CONTINUE
.
. (commands to deal with inadequate environment space)
.
:CONTINUE
. (the remainder of the program)
.
REM Restore the previous prompt:
SET COPYCMD=%SAVECOPYCMD%
REM Remove unneeded variable:
SET SAVECOPYCMD=
```

同样的方法可用于保存和恢复任何环境变量。

注释： 如果在命令 SET COPYCMD=%SAVECOPYCMD%被执行之前批处理程序被中止了，例如用户按下了 Ctrl+Break 键——变量 COPYCMD 之值不会被恢复。

43.2.5.3 环境变量和可替换参数的比较

使用环境变量而不使用数字形式的可替换参数如%1，有以下好处：

- 可以使用尽可能多的变量(只要不超出环境大小的限制)。

- 环境变量比一个可替代参数的有效期更长久，它在整个批处理程序工作期间都有效，而且还能被其他批处理程序访问。

43.2.5.4　使用 CONFIG 变量

当用户在 CONFIG.SYS 中设置多个配置块时，DOS 设置 CONFIG 环境变量来指示启动过程中所选择的块。用户可以通过 AUTOEXEC.BAT(或其他批处理程序)访问该变量，根据所选择的配置进行分枝。例如，如果 CONFIG.SYS 中有三块：PETER、JUDI 和 SCOTT，可以按下列方法在 AUTOEXEC.BAT 中分枝到不同的启动路线：

```
@ECHO OFF
    commands for all blocks
GOTO %CONFIG%
:PETER
    commands that should be executed for the PETER block
GOTO ENDING
:JUDI
    commnads that should be executed for the JUDI block
GOTO ENDING
:SCOTT
    commands that should be executed for the SCOTT block
:ENDING
```

提示： 如果块名是用户的名字，也可以用 CONFIG 为用户提供一些信息。例如，可以显示如下的欢迎信息：

```
ECHO Hello, %CONFIG%... the PC has been set up for your configuration.
```

也可以使用 PATH 命令中包含以用户的名字作为目录名的路径：

```
PATH=%PATH%;C:\%CONFIG%
```

甚至可以在命令提示符中包含用户的名字：

```
PROMPT %CONFIG%'s configuration $_$P$G
```

43.3　控制命令解释程序和环境

在引导过程中，装入命令解释程序后就建立环境。缺省时，DOS 的命令解释程序 COMMAND.COM 在装入时便保留了 256 字节的环境。要控制命令解释程序和环境的大小，必须在 CONFIG.SYS 中写上 SHELL 命令(见图 43.2)。

当 CONFIG.SYS 中没有出现 SHELL 命令时，DOS 从根目录下装入 COMMAND.COM。如果 COMMAND.COM 不在启动盘的根目录下，DOS 便查看\DOS 目录。DOS 根据 COMMAND.COM 所在的地方设置 COMSPEC 变量的值。

答疑： 当试图启动计算机时，DOS 给出下列出错信息：Bad or missing command interpreter，Enter correct name of Command Interpreter (such as C:\COMMAND.COM)，出了什么错？怎样才能完成启动过程？

出错的原因在于 DOS 没能在启动盘的根目录下找到 COMMAND.COM，或者也

未在该驱动器的\DOS 目录下找到，CONFIG.SYS 中也未有 SHELL 命令指明命令解释程序所在。这有可能是 COMMAND.COM 文件受到损坏或被删除了，或者 SHELL 命令不在 CONFIG.SYS 中，也有可能硬盘受到某些损坏，以致于 DOS 不能访问\DOS 目录。如果不告诉 DOS 在哪里能找到命令解释程序，就会永远完不成启动过程。

如果用户经常在 DOS 目录或根目录中使用 COMMAND.COM 文件，该文件可能丢失或遭破坏，或者 DOS 不能访问硬驱。如果这样，用户必须用含有命令解释程序的紧急启动软盘重新启动，然后把它拷贝到启动驱动器的根目录，再重新启动。

如果用户经常把 COMMAND.COM 放到别的目录下，或者经常使用别的命令解释程序如 NDOS，就必须键入命令解释程序文件名的全名。如果 DOS 完成了启动过程，编辑 CONFIG.SYS 以插入(或修复)适当的 SHELL 命令，然后再次从硬盘启动(如果 DOS 按上面所说的方式装入命令解释程序，便不会执行 AUTOEXEC.BAT，所以用户就不得不根据系统的缺省设置使用最小的系统)。如果已经输入了文件名而 DOS 还继续询问时，就不得不用软盘重新启动，以便能确定问题之所在，并加以解决。

SHELL

装入命令解释程序。

句法：

SHELL＝filespec [parameters] [switches]

参数和选项的内容取决于装入哪个命令解释程序。下面是针对 COMMAND.COM 而言的：

SHELL＝[path] COMMAND.COM [path] [device] [/E:n] [/P]

参数：

filespec	指定欲使用的命令解释程序。
path	如果命令解释程序不在启动盘的根目录下，便指明它所在的路径。
device	指明用于标准输入输出的设备。缺省值是 CON。
/E:n	指明环境大小的字节数，可以从 160 到 32768，缺省值是 256。
/P	设置该命令中的命令解释程序为第一级命令解释程序。

图 43.2 SHELL 配置命令总结

因为缺省的环境大小是 256 字节，因此，如果除了 DOS 所用的环境变量外，用户还自己建立和使用了一些环境变量，就很容易把环境的空间用尽。

在下述情况时，使用 SHELL 命令：

- 如果用户不想使用 DOS 的 COMMAND.COM 而使用自己的命令解释程序。
- 命令解释程序既不在启动盘的根目录也不在\DOS 目录下。
- 用户准备改变环境的大小。

43.3.1 环境大小

要获得一个大些的环境，用户必须在 CONFIG.SYS 中使用带有/E 选项的 SHELL 命

令。要使用缺省的命令解释程序并建立 1KB 字节的环境，可使用下述命令：

```
SHELL=COMMAND.COM /E:1024 /P
```

警告： 环境设在常规内存中，它是可用的空间，因此不要把环境设置得过大而超过了实际所需。

警告： 如果没有写入/P 选项，而随后又键入了 EXIT 命令，就会出现死机。如果使用了选项/P，EXIT 就被忽略。

43.3.2　存放在其他地方的命令解释程序

如果命令解释程序不在启动盘的根目录中，用户就必须两次说明其位置：

- 必须包括和第一个参数(COMMAND.COM)连在一起的路径，以便 DOS 在启动过程中能知道在哪儿可找到命令解释程序。
- 必须写入路径(不包含文件说明)作为第二个参数，以便 DOS 在需要装入它的暂住内存部分时能找到命令解释程序之所在。该参数也使 DOS 在环境中设置变量 COMSPEC 的值。

如果要使用 C:\DOS\COMMAND.COM 作为命令解释程序，并使用缺省的环境大小，便须在 CONFIG.SYS 中使用下述命令：

```
SHELL=C:\DOS\COMMAND.COM C:\DOS /P
```

第二个参数使 DOS 建立下面的环境变量：

```
COMSPEC=C:\DOS\COMMAND.COM
```

43.3.3　使用第三方的命令解释程序

可以不使用 DOS 的 COMMAND.COM 作为命令解释程序。例如，使用 Norton Utilities 中的 NDOS 程序，该程序比 COMMAND.COM 具有更多的特色。SHELL 用于建立像 NDOS 这样的其他命令解释程序，在这种情形下，COMMAND.COM 不会被装入。

43.4　装入第二级命令解释程序

有时，解决某个问题的最好办法便是装入第二级命令解释程序，即重装 COMMAND.COM 的另一个实例。例如，如果用户临时性的需要更多的环境空间，便需要改变到另一个命令解释程序；或者用户打算重新定向批处理文件的输出，所有这些都可以通过装入第二级命令解释程序完成。下面几段中将阐述这个问题。

COMMAND 命令把 COMMAND.COM 作为第二级命令解释程序(见图 43.3)。下面的命令启动 COMMAND.COM 作为第二级命令解释程序，并设置 2KB 大小的环境；由于没有其他路径说明，COMSPEC 变量的值将保持不变。

```
COMMAND /E:2048
```

只要不使用/P 选项，第二级命令解释程序便是第一级的子程序。第一级命令解释程序

仍然驻留内存，并在结束第二级解释程序后，又恢复控制权。如果要结束第二级命令解释程序而回到第一级命令解释程序去，可键入下述命令：

EXIT

COMMAND

启动新的 DOS 命令处理器 COMMAND.COM。

句法：

COMMAND [path] [ctty] [/E:n] [/Y[/C command] | [/K filespec]]

参数和选项：

path	指明命令处理程序的路径。
ctty	指明命令输入及输出的设备，缺省值是当前使用的设备。
/E:n	说明新设置的环境大小的字节数，从 160 到 32768。
/Y	表明在执行/C 或/K 选项中指定的批处理程序时，一次只执行一行，并且询问是否执行该行。
/C command	装入第二级命令处理器，处理 DOS 命令，然后退出第二级命令处理器回到第一级。
/K filespec	在显示 DOS 命令提示符之前运行指定的程序。

图 43.3 COMMAND 命令总结

试着在自己的系统上装入第二级命令解释程序：

1. 键入 COMMAND /E:1024 命令，装入命令解释程序并设置 1KB 的环境。
2. 键入 PROMPT DT$G，改变命令提示符，在命令提示符中显示时间和日期。用户可以立即看到出现新的命令提示符。
3. 键入 SET EXERCISE=Trying out a secondary command interpreter 命令。
4. 键入 SET 命令查看所设置的环境变量。
   注意所有的常规变量仍然存在，但 PROMPT 变量的值是当前的命令提示符。另外，也出现了 EXERCISE 变量。
5. 键入 EXIT 命令，返回父命令解释程序。
   命令提示符也恢复其原值。
6. 键入 SET 命令，查看环境变量。
   现在的显示和启动第二级命令解释程序之前的显示一样，特别是，EXERCISE 变量被取消了。

43.4.1 指定 COMSPEC

如果在装入第二级命令解释程序时给 COMSPEC 赋值，那么在新的环境中，除了 COMSPEC 变量外，其余都是空的，不再有 PATH、PROMT 或其他任何变量，即使这些变量曾存在于父环境中。用户会立即注意到这些变化，因为命令提示符变成了 DOS 的缺省值——也即是说，仅出现当前驱动器而不出现当前目录。一个恢复路径和提示符的比较快的方法是重新运行 C:\AUTOEXEC.BAT，但也可以第二次装入 TSR，虽然有时不理想。

下面的例子可以在装入第二级命令解释程序时指定 COMSPEC：

1. 键入 COMMAND C:\ 命令，装入第二级命令解释程序。
2. 用 SET 命令查看环境变量。
   只显示 PATH 和 COMSPEC 两个环境变量，而且 PATH 没有值。
3. 键入 EXIT 命令回到正常环境。

43.4.2　忽略 COMSPEC

如果不说明 COMSPEC，那么第二级环境将继承父环境的变量值。用户可以改变、添加或删除第二级环境中的变量而不影响父环境。当回到父命令解释程序时，环境变量亦恢复到原来的值。这是启用第二级命令解释程序的主要理由。例如，如果用户需要建立一个批处理程序，该批处理程序将改变命令提示符、搜索路径和 TEMP 目录，而用户又希望该批处理程序运行结束后，所有的环境变量又恢复原值，那么就可以按下述方法去做：

```
@ECHO OFF
REM Preserve current environment:
COMMAND
.
. (remainder of batch program)
.
REM Restore original environment:
EXIT
```

假如用户建立一个需要 2KB 环境的批处理程序，启动第二级命令解释程序不仅可以获得临时性的额外环境空间，而且在回到父命令解释程序时将自动删除多余的环境变量：

```
@ECHO OFF
REM Establish a larger environment:
COMMAND /E:2048
.
. (remainder of batch program)
.
REM Return to original environment:
EXIT
```

43.4.3　动态环境分配

当 COMMAND 建立第二级环境时，DOS 将再次初始化缺省的或所要求的环境空间。但是，如果设置变量时超过了环境空间大小而且相邻的内存又可以利用，DOS 将动态地把相邻的内存的一部分按用户所要求的大小分配给环境，并建立所要求的环境变量。用户可以利用这种方法设置几千字节的环境变量，但必须记住这样做占用了常规内存的可用部分，这将会对程序的装入造成困难。

注释： 如果没有说明/E 选项，缺省的环境是 256 字节，或者是原父环境的大小。

43.4.4　在其他命令解释程序为第一级命令解释程序时启用 COMMAND.COM

假如 NDOS 是作为第一级命令解释程序装入的，而用户现在所要运行的程序又要求 COMMAND.COM 存在，便可以键入下面的命令，把 COMMAND.COM 作为第二级命令解

释程序启动：

```
COMMAND C:\ [switches]
```

用户必须说明 COMSPEC 路径，这样 COMMAND 环境才不至于继承 NDOS 的 COMSPEC 变量，否则 DOS 在从一个错误的目录下装入 COMMAND 的暂住内存部分时会出错。因为说明了 COMSPEC 路径，因而不会从父环境继承任何变量，在继续工作之前用户有时需要输入路径和提示符。

要返回 NDOS 时，键入下面命令：

```
EXIT
```

43.5 装入新的第一级命令解释程序

假如用户想用 COMMAND.COM 取代原来的第一级命令解释程序，可以在命令中写上/P 选项，例如：

```
COMMAND C:\ [switches] /P
```

这个方法和重新启动的效果有一些相似，位于当前驱动器的根目录下的 AUTOEXEC.BAT 被执行。(如果当前驱动器不是 C 盘，也没有可执行的 AUTOEXEC.BAT 文件，则将执行 DATE 和 TIME 命令以代替执行 AUTOEXEC.BAT)。键入 EXIT 命令将不会中止解释程序。虽然不能返回先前的命令解释程序，但它仍然驻留于内存中，这有可能引起内存空间的不足。如果内存太拥挤，最好改变 CONFIG.SYS 中的 SHELL 命令，并且重新启动，以装入所希望的命令解释程序及设置相应的环境。

43.6 用 COMMAND.COM 执行一个命令

/C 选项传递一个命令到第二级命令解释程序，在执行完该命令后，又返回到父命令处理程序。用户可利用这个特点来重定向输出，即使这些输出不能被其他方式重定向。

提醒用户不能重定向批文件的输出或者 FOR 命令的管道输出。但是，可以重定向 COMMAND 的输出。下面用 HELLO.BAT 为例作一练习。

1. 键入 COMMAND /C HELLO>HELLO.TXT 命令。
   DOS 把 HELLO 命令的输出重定向到 HELLO.TXT 中去，结果是用户在屏幕上看不到任何信息。然后第二级命令处理器中止，原来的命令提示符又出现。
2. 用 TYPE 命令打印 HELLO.TXT，查看结果。
3. 删除 HELLO.TXT。

在第 7 章“在 Shell 中选择文件”中说过如何把 FOR 和 FIND 一起使用，从而在一组文件中搜索一个指定的词组。但是用户不能用管道把输出流向 MORE，因为 FOR 不能被管道化。再者，COMMAND 可用于完成重定向，例如下面的操作：

1. 转到 DOS 目录下去。

2. 键入 COMMAND /C FOR %F IN(* . TXT) DO FIND /I "network" %F |MORE 命令。

结果将一页一页地显示。(如果用户对一页一页地显示感到不满,可以按 Ctrl+Break 键结束显示)。如果用户想从 DOS. TXT 文件中找到更多有关网络的信息,现在就可以获得足够的信息以确定要读取或打印的文件。

除非用户把 TEMP 定向到 RAM 驱动器上,当 MORE 为 FOR 命令输出的所有东西建立临时文件时,该命令可能要暂停很长一段时间。

注释: 如果重定向 COMMAND 命令而不使用/C 选项,所有第二级命令解释程序的标准输出也要被重定向,直到退出该命令解释程序返回第一级为止。这其中包括命令提示符,它不再出现在监视器上,但被记录在重定向输出中。不过,即使看不见命令提示符,用户仍然可以键入命令和数据。

43.7　用 COMMAND. COM 检验一个批处理程序

在 DOS 6.2 中,如果在 COMMAND 命令的/C 和/K 参数之前说明了/Y 选项,就可以通过一行一行的执行方式检验一个批处理程序的逻辑。试以 HELLO. BAT 为例:

1. 键入 COMMAND /Y /C HELLO 命令。

DOS 装入第二级命令解释程序并通过下列提问询问用户是否打算执行 HELLO:

hello [Y/N]?

2. 按 Y 键,执行 HELLO。

DOS 便询问是否执行 HELLO 的第一行,如下所示:

```
echo off [Y/N]?
```

3. 按下 Y 键,执行 ECHO OFF 命令。

DOS 询问是否执行 HELLO 的下一行:

```
echo Hello,name [Y/N]?
```

4. 按下 N 键,跳过该行的命令。

DOS 不执行 ECHO 命令。由于它是批处理程序的最后一条命令,因此程序执行结束。第二级命令解释程序也中止,并且返回第一级命令解释程序。

43.8　自 测 练 习

本章讲了如何设置和使用环境变量,也讲了怎样控制命令解释程序的装入。如果用户想练习本章中学得的技巧,可参考下列步骤:

1. 查看系统当前的环境变量列表。
2. 设置环境变量 ONMYOWN 的值为 ITEM2。
3. 写一个批处理程序,把 ONMYOWN 变量作为命令提示符的一部分加以显示。

4. 设置 DIR 命令的缺省选项，使 DOS 按与年代顺序相反的次序显示文件，并且把系统文件和隐含文件也列出来。
5. 为 COPY，XCOPY 及 MOVE 命令设置缺省选项，以忽略 DOS 的复写保护。
6. 设置 Backup 和 Anti-Virus 的缺省选项，使它们使用目录 C:\而不是它们的常规目录。检验 Backup 中的结果。
7. 如果用户有至少 4MB 的内存，而且没有使用 RAM 磁盘来存储临时文件，那么建立一个 512KB 的 RAM 驱动器，并把 DOS 的临时文件定向到它上面去。在建立起 RAM 驱动器之前和之后都键入一个命令，如 DIR C:\ /S /B|FIND "MEM"，用户会发现在建立前后命令执行所用的时间有很大不同。
8. 确定当前的环境空间大小。如果需要更大的空间，便在 CONFIG. SYS 文件中插入（或编辑）适当的命令。
9. 执行 HELLO 程序，把输出重定向到打印机。
10. 执行 HELLO 程序，一次只执行一行。
11. 在继续操作之前，从 CONFIG. SYS 中删除 RAMDRIVE 命令（如果用户按第 7 步执行的话），然后，重新启动计算机恢复原来的环境设置。然后，删除 HELLO. BAT 文件及第 3 步中所建立的批处理程序。

第 44 章　腾出更多的内存空间

简介： 对于 386 或更高级的 PC 机，如果用户在充分利用 DOS 的内存管理时出现障碍，就可能需要使用 HIMEM、EMM386 及 MEMMAKER 的高级功能。

44.1　定制高端内存

前面已经讲述了怎样装入 HIMEM、怎样把 DOS 装入 HMA 以及怎样用 MEMMAKER 配置启动文件，以装入 EMM386，并使尽可能多的 TSR 和驱动程序适应高端内存。但是，如果 MEMMAKER 或 EMM386 不能装入用户系统，或者如果 MEMMAKER 没有把足够多的 TSR 和驱动程序移出常规内存，要解决这些问题，还要做很多事情。

本章将阐述以下几个方面：

- 在 HIMEM 命令中使用选项，以控制怎样访问扩展内存
- 在装入 EMM386 后，打开或关闭扩充内存
- 建立 DOS 经常保留的可用高端内存
- 查看程序在内存中所占用的空间
- 在自定义方式而不是快速方式下运行 MEMMAKER
- 弄清楚为什么有时不重新启动就不能卸出 TSR(常驻内存程序)

44.2　使用 HIMEM.SYS 出现的问题

如果用户发现在缺省配置时 HIMEM 不能工作——例如，HIMEM.SYS 不能装入，或者安装 HIMEM 后在启动过程中出现死机，或者在系统运行中出现系统性能的降低——这些情况都需要在 HIMEM 的设备驱动命令中加上选项。图 44.1 示出了 HIMEM 的选项。

HIMEM.SYS

管理扩展内存，包括高内存区(HMA)。

句法：

DEVICE=[path] HIMEM.SYS [/A20CONTROL:ON|OFF]

[/CPUCLOCK:ON|OFF] [/EISA] [/HMAMIN=size] [/INT15=x]

[/MACHINE:machine] [/SHADOWRAM:ON|OFF] [/VERBOSE]

参数和选项：

path　　指明 HIMEM.SYS 的路径，缺省时是启动盘的根目录下。

/A20CONTROL:ON|OFF	说明 HIMEM 是否执行 A20 地址线的控制，即使 A20 地址线已处于 on（打开）状态。如果命令形式为/A20CONTROL:OFF，则仅 A20 在 HIMEM.SYS 被装入时已处于 off(关)状态时，HIMEM 才会执行 A20 地址线控制。缺省值是/A20CONTROL:ON。
/CPUCLOCK:ON|OFF	说明 HIMEM 是否影响计算机的时钟速度。如果安装 HIMEM 时计算机的时钟速度发生改变，/CPUCLOCK:ON 可以校正这个问题，但会降低 HIMEM 的速度。缺省值是 OFF。
/EISA	指明 HIMEM.SYS 应配置所有可用的扩展内存。这个参数仅在 EISA（Extended Industry Standard Architecture）总线的计算机上且具有 16MKB 内存时才会用到。在其他计算机上，HIMEM 将配置所有可用的扩展内存。
/HMAMIN=size	说明使用 HMA（高内存区）的程序所必须具备的大小（以 KB 计）。有效值是 0 到 63，缺省值是 0。当 size 为 0 时，迫使 HIMEM 把 HMA 分配给第一个需要它的程序。该选项对在 386 增强模式下运行的 Windows 无效。
/INT15=X	为 INT15h 中断分配扩展内存空间（以 KB 计）。有效值是 64 到 65535，但不得超过系统可用的内存。缺省值是 0。如果 X 小于 64，HIMEM 以 0 计。
/MACHINE:machine	说明所用的 A20 地址处理器。
/SHADOWRAM:ON|OFF	说明是否激活影象内存。如果计算机的内存小于 2MB，缺省值是 OFF。
/VERBOSE	使 HIMEM.SYS 在被装入时，显示状态和错误信息。缺省时，HIMEM.SYS 不显示这些信息。用户可把/VERBOSE 简写为/V。

图 44.1 HIMEM.SYS 配置命令总结

44.2.1 装入正确的 HMA 驱动程序

如果 HIMEM 根本不能装入，或者用 HIMEM 启动时出现死机，最可能的问题在于/MACHINE选项。该选项使 HIMEM 装入一个驱动程序来管理指示 HMA 地址的电路（或地址线）。在处理器和 HMA 之间的连接线称为 A20 地址线（连接处理器和常规内存及高端内存的地址线编号为 A00 到 A19）。由于它管理 A20 地址线，因此该驱动程序称为 A20 处理器。

Microsoft 在 HIMEM.SYS 中包含了几个 A20 处理器。在本书后面命令参考中 HIMEM.SYS 项下列出了所有这些处理器。HIMEM.SYS 力图为用户的硬件装入正确的处理器，但对于部分计算机，它却做不到这一点。当这种情况出现的时候，用户不得不使用/MACHINE选项以装入正确的处理器。如果用户不能肯定该用哪个处理器，这对试验并不妨害。如果通过试验仍然找不到，则硬件销售商及 Microsoft 公司的产品支持组将能帮助用户解决问题。

注释： 如果在装入 HIMEM 时出现故障，便在 DEVICE 命令后面加上/VERBOSE 选项，使 HIMEM 显示状态信息。该信息也许会帮助发现错误出在什么地方以及怎样做才正确。

如果用户装入了正确的驱动程序但还不能访问 HMA，那么试使用/A20CONTROL:ON。该选项使 HIMEM 对 A20 进行控制，即使有别的程序正在使用 A20 时也可以获得 A20 的控制权。

44.2.2　控制时钟速度

在个别情形下，HIMEM 改变计算机的时钟速度，这可能影响其他软件的运行。如果这种情况出现了，可使用/CPUCLOCK:ON 选项，该选项虽然对 HIMEM 的速度有所降低，但能防止改变时钟速度。

44.2.3　在 EISA 总线的计算机中，利用 16MB 以外的内存

如果在 HIMEM 命令中不包含/EISA 选项，那么在 EISA 总线的计算机上，HIMEM 便不能访问 16MB 以外的内存。

44.2.4　控制直接访问扩展内存地址的老程序

当刚能使用扩展内存，但还没有建立任何标准的时候，一些软件使用 INT15h 机制(中断 15h)来访问扩展内存。如果用户还在使用这样的程序，就可能在内存空间上和 HIMEM 发生冲突，并有可能丢失数据。最好的解决办法就是用与 HIMEM 兼容的新版本替换原来的旧版本。假如没有新版本，则可能通过写上/INT15h 选项，调用 INT15h 来访问扩展内存。例如，如果程序要求有 128KB 的扩展内存，便可以在 HIMEM 命令中包括/INT15h=128 选项。为 INT15h 保留的扩展内存不能被其他软件访问，所以不要给 INT15h 分配超过实际需要的内存空间。

44.2.5　确定 DOS 装入了 HMA

一次只能把一个程序装入 HMA。缺省时，DOS 把第一个需要 HMA 的程序装入高端内存，而不管程序有多小。在 CONFIG.SYS 中 DOS 执行 DOS=HIGH 命令，不论这个命令写在 CONFIG.SYS 中的任何地方，另一个程序都有可能接管 HMA。如果是一个很大的程序，这当然很好，但如果程序小于 43KB，则最好把 DOS 从 HMA 中卸出。如果用户的系统中曾发生过这样的问题，那么在 HIMEM 命令中加上/HMAMIN=40 选项，这样那些小于 40KB 的程序就不会被装入 HMA。

44.2.6　影象 ROM-BIOS

ROM-BIOS 是系统最基本最重要的部分。不幸的是，访问 ROM 比访问 RAM 要慢。很多计算机为了加速 BIOS 的速度，便把 ROM 拷贝到 RAM 中去，通过访问 RAM 完成对 ROM 的访问。ROM 在 RAM 中的拷贝便称为影象 RAM(shadow RAM)。

如果内存 RAM 不足 2MB，在 HIMEM 装入时，它自己关掉系统中的影象 RAM，以便获得更多的扩展内存。关掉影象 RAM 可能严重减慢整个系统的运行速度。如果不需要额外的内存，可以在 HIMEM 命令中写上/SHADOWRAM:ON 选项，来防止 HIMEM 关闭影象 RAM。

注释： 如果内存中没有影象 RAM，那么/SHADOWRAM:ON 选项并不生成它，

它仅仅是防止 HIMEM 把影象 RAM 关掉。

注释： HIMEM 还有其他几个用得很少的选项，这些选项在本书后面的命令参考中有描述。

44.3 在高端内存中找到额外的空间

通常，MEMMAKER 能毫无问题地把 TSR 和驱动程序装入高端内存。在使用 MEMMAKER 以后，用 MEM /C 命令查看一下常规内存，就可以发现是否有应该放到高端内存的程序被装入常规内存里。如果是这样，为了要利用更多的高端内存，用户还必须做一些工作。

通过重新运行 MEMMAKER 和选择自定义安装方式而不是快速安装方式，可以很方便地完成大部分工作。表 44.1 示出了在建立自定义方式时 MEMMAKER 所提供的选项。这些选项可以帮助用户找到更多的高端内存空间。下面几段就讲述这些选项。

表 44.1 自定义建立选项

选 项	功能说明
Specify drives and TSRs (指定驱动程序和常驻内存程序)	如果不想把所有的常驻内存程序和驱动程序都装入高端内存，选择该项
Aggressive scan (强制性扫描)	如果想让 MEMMAKER 只使用从 F000 到 F7FF 的高端内存时，那么选择该项。快速安装方式便会使用这个内存范围。假如快速方式建立系统后有故障，那么关掉该选项后再试
Optimize for Windows (优化 Windows)	如果用户不在 Windows 下，基于 DOS 的应用程序，就可选此项。如果用户不使用 Windows，或者不在 Windows 下运行，基于 DOS 的程序，那么关掉此项
Use monochrome region (使用单显内存区)	如果用户没用单色或 SuperVGA 监视器，选择此项。如果 MEMMAKER 使用单色显示器内存缓冲区后出现故障，那么返回 MEMMAKER，关掉该选项
Keep Current inclusions and exclusions (保留当前包含和排除的高端内存空间)	如果用户正装入 EMM386.SYS 驱动程序，以包括或排除适当的高端内存空间，便选择此项来保留这些结果。关掉此项可以给 MEMMAKER 提供更多的自由控制权，但有可能出现一些问题
Move extended BIOS data (移动扩展 BIOS 数据)	选择此项，便把扩展的 BIOS 数据区移到高端内存。如果 MEMMAKER 使用该选项后出现了问题，那么回到 MEMMAKER 把此选项关闭

44.3.1 强制性扫描(Aggressive Scan)

EMM386 在装入时便扫描高端内存的可用空间。正常情况下，它不扫描 F000h 以后的高端内存块，因为它属于 ROM-BIOS 所有。在启动过程中，如果扫描这个区域，便会使计算机死机。但是，用户有时需要使用在这个区域内没被 ROM-BIOS 使用的内存块。如果选用 Aggressive Scan 选项，MEMMAKER 在 EMM386 命令的后面加上/HIGHSCAN 选项，这样

EMM386 便会扫描 F000h 以后的内存区。假如这样做以后系统被挂起，那么按 F8 键重新启动，以跳过装入 EMM386 的命令，从 EMM386 命令中删除/HIGHSCAN 选项以后再试。

44.3.2 使用单显内存区(Use Monochrome Region)

从 B000h 开始的高端内存区，在正常情况下由单色显示适配器、CGA 适配器以及采用单色方式的其他适配器使用。一些 SVGA 监视器在高分辨图形方式下也使用这个内存区。假如用户没有使用单色显示器或 CGA 适配器，也没有在单色方式下使用适配器，或者也没有 SVGA 监视器，则可以很安全地把这部分内存当作 UMB 来使用。

假如用户选择了这个选项而又在使用 Windows，则必须在 SYSTEM.INI 文件中插入一个命令来装入称作 MONOUMB.386 的特殊驱动程序。即把下面命令插到 SYSTEM.INI 的[386Enh]部分：

```
DEVICE=C:\DOS\MONOUMB.386
```

警告： 不正确的使用 EMM386.EXE 参数有可能使系统瘫痪。假如发生了这个事情，按 F8 键以跳过 EMM386.EXE 命令，并对 CONFIG.SYS 中的这个命令作些编辑。

44.3.3 排除高端内存区的访问

一些系统自己要使用一部分高端内存，MEMMAKER 不能对这部分内存作访问。用户可以在自定义安装方式运行 MEMMAKER 之前，在 EMM386 DEVICE 命令(见图 44.2)中加上 X 选项，以划分出一定地址范围内的内存不让 MEMMAKER 使用。而且，一定要激活选项 Keep Current EMM386 Memory Exclusions and Inclusions。例如，如果已经知道系统使用的地址范围是从 B800 到 BB00，而其余的单色显示缓冲区部分没被占用，便可在 DEVICE=EMM386 命令中加上下面的选项：

```
x=B800-BB00
```

EMM386.EXE

提供对高端内存区的访问，并把扩展内存仿真成扩充内存。

句法：

DEVICE=[path] EMM386.EXE [ON|OFF|AUTO] [expanded-memory] [MIN=size] [Mx]

[[X=mmmm1-nnnn1]...] [RAM|NOEMS] [HIGHSCAN] [/VERBOSE]

参数和选项：

path	指明 EMM386.EXE 文件的路径，缺省时是启动驱动器的根目录。
ON|OFF|AUTO	ON：装入并激活 EMM386 驱动程序；OFF：装入该驱动程序但它被禁止使用；AUTO：使驱动程序处于自动状态，只有当程序需要时才去访问扩展内存支持和高端内存块支持。该驱动程序装入后，可以用 EMM386 命令改变这个选项的值。
expanded-memory	说明要作为扩充内存使用的扩展内存大小。有效值可以是 64 到 32768(或是可用的空闲扩展内存量)。缺省值是可用的空闲扩展内存的大小。指定 NOEMS 选项，则缺省值为 0。EMM386.EXE 将此参数舍入到它最接近的那个 16 的倍数。

MIN＝size	说明 EMM386.EXE 可提供的最小扩充内存空间(以 KB 计)。其值可以是 0 到整个扩充空间。如果没有说明 NOEMS 参数,则缺省值是 256。
Mx	说明扩充内存页帧的起始地址。有效值是 1 到 14。如果计算机的内存小于 512KB,X 必须小于 10。
RAM	要求扩充内存和 UMB(高端内存块)支持,并且说明在扩充内存中用作 UMB 的段地址范围。
NOEMS	提供 UMB 支持,但不提供扩充内存支持。
HIGHSCAN	扫描高端内存的最高区域,寻找可用的内存块。
/VERBOSE	使 EMM386.EXE 在装载过程中显示状态信息和错误信息。缺省时,EMM386.EXE 仅显示错误信息。可以把/VERBOSE 缩写为/V。

图 44.2 EMM386.EXE 配置命令的总结

在运行 MEMMAKER 的时候,选择 Custom Setup,Use Monochrome Region 以及 Keep Current EMM386 Memory Exclusions and Inclusions 等项。

44.3.4 打开及关掉扩充内存

在启动过程中通过 CONFIG.SYS 装入 EMM386 以后,只要它不处于使用状态,便可以用 EMM386 命令关闭它或打开它(如图 44.3 所示)。关掉它可以获得更多的扩展内存空间给另外的应用程序使用,而当运行一个需要扩充内存空间的程序时,又可以把它打开。

注释: 虽然 EMM386.EXE 不能在 286 上工作,但可以买到别的具有相同(甚至更多)功能的适用于 286 的软件。有很多内存管理软件能够提供比 EMM386.EXE 更多的功能,即使对 386 或更高档的计算机亦是如此。

EMM386

控制扩充内存支持。

句法:

EMM386 [ON|OFF|AUTO] [W＝ON|W＝OFF]

参数和选项:

none	显示 EMM386 支持状态。
ON	激活 EMM386.EXE 驱动程序。
OFF	关闭 EMM386.EXE 驱动程序,但不卸出内存。
AUTO	让应用程序自己控制是否激活 EMM386.EXE 驱动程序。
W＝ON	打开 WeiteR 协处理器。
W＝OFF	关掉 WeiteK 协处理器,缺省值是 W＝OFF。

注释:

在使用这个命令之前,必须具备 386 或更高档的协处理器,并且必须装了 EMM386.EXE 设备驱动程序。

在 EMM386 正在提供 UMB 支持或者有别的程序正在使用扩充内存时,用户不能关掉 EMM386。

图 44.3 EMM386 命令总结

在 CONFIG.SYS 中装载 EMM386.EXE 的命令中写上 OFF 或 AUTO 参数，就可以把 EMM386.EXE 装入但让它处于 OFF(关闭)状态。然后在需要的时候再用 EMM386 命令把它打开。

注释： 一些程序，特别是 Windows 3.0，有它们自己相仿的程序，并且不能和 EMM386 兼容。

在 EMM386 正在提供高端内存块访问支持时，或者一些程序正在使用它提供的扩充内存时，假如用户试图把 EMM386 关掉，就会看见显示下列信息：

Unable to de-activate EMM386 as UMBs as UMBs are being provided and/or EMS is being used.

44.4　使用 UMB 的问题

并不是所有的程序都能使用 UMB(高端内存块)。一些程序在装入后便试图扩展它们所用的内存空间。但空间又不够；一些程序原本就假定好了自己装在内存中的什么地方。假如在启动过程中或启动后出现死机，便跳过启动文件，并查看一下是哪个程序引起了问题。

提示： DOS 6 下的所有 TSR 及设备驱动程序都能在 UMB 下毫无问题地运行。

使用下面的命令，可以查看程序装入内存(常规内存或高端内存)后所占的实际空间大小：

```
MEM /MODULE:program-name
```

例如，要查看 DOSKEY 占用了多少内存，可键入下述命令：

```
MEM /MODULE:DOSKEY
```

所显示的报告如下：

```
DOSKEY is using the following memory:
  Segment  Region        Total          Type
  -------  ------  ----------------  ----------
   0D5EB      1        4144    (4K)    Program
                     ----------
  Total Size:         4144    (4K)
```

从报告中可以看出 DOSKEY 占用了 4144 字节的内存。只有当程序位于高端内存中时才显示区号(the Region Number)。有时可以看到一个程序占用了两个或更多的内存区，可能是高端内存中的两个区。这时 Total Size 一项中所显示的是所有区的内存大小之和。

假如用户怀疑一个程序没有正确装入，而且希望它避开高端内存，那么编辑装载它的命令，把 DEVICEHIGH 改变成 DEVICE (在 CONFIG.SYS 中)，或者删除其前面的 LH 命令(在 AUTOEXEC.BAT 中)。同时也删除 MEMMAKER 加到命令上的/L 和/S 选项。

44.5　操作 TSR(驻留内存的程序)

一个 TSR(Terminate-and-Stay-Resident)程序是装入内存后便驻留内存的程序，不论它

是否启动都是如此。很多 TSR 扫描键盘输入，以待键入命令，或者给键附属一定的功能(即热键)。DOSKEY 便是一明显的例子，当用户在命令提示符下按向上的箭头键时，便重复出现上一次的命令。

如果没有安装 TSR，当用户按下一个键时，DOS 便把信号送到键盘处理器去处理。装入 TSR 后，它便扫描键盘输入，TSR 把自己插入到正常的键盘信号处理链中，以便它能首先发现键盘的输入信号，并对信号进行处理，并把不认识的信号送给 DOS 的键盘信号处理器。

假如用户装入第二个 TSR 以扫描键盘输入，它便把自己插到第一个 TSR 之后。如果用户不停地装入 TSR，则键盘信号处理链就会变得越来越长：

TSR #1—→TSR #2—→TSR #3—→DOS's keystroke hander

假如 TSR 不是位于链的末尾，则用户不能把它从链中删去。假如用户能够删除一个位于链中心的 TSR，则可能是因为链断了，这样用户的击键信息就不能到达 DOS 的键盘处理器。实际上，系统就很可能死机。

并不是所有的 TSR 都能在键盘处理链中正常的工作，有时会出现死机的现象。这种情况常常是因为两个或更多的 TSR 不相互兼容。很可能其中之一为盗版或者没有完全正确安装。假如这种情况发生在用户的计算机上，那么试试按不同的顺序安装 TSR。如果仍不能工作，一次跳过一个 TSR，直到发现是哪一个出现了问题为止。和 TSR 的提供商联系，他们会告诉你怎样安装 TSR，并且他们知道什么地方出了问题以及相应的解决办法。

第 45 章　再谈 DrvSpace 和 DblSpace

简介： 一旦用户安装了 DrvSpace 及 DblSpace 后，便可以压缩其他的磁盘（包括软盘），以及维护压缩过的驱动器，或者解除它们的压缩格式。

45.1　DrvSpace 和 DblSpace 基本功能之外的特色

如果用户已经压缩了唯一的硬盘，可能就不会再去考虑 DrvSpace 或 DblSpace 了。在每次启动计算机时，压缩软件都被装进去，并管理压缩了的驱动器，以便用户能够访问。除了能存储更多的数据之外，用户可能意识不到它的存在。

但 DrvSpace 和 DblSpace 还有其他一些特色。例如，用户还可以压缩其他的磁盘（包括软盘）。用户可以改变压缩驱动器的空间大小，在需要时对它们进行扩展或收缩。或者也能对这些磁盘解除压缩。本章将解释怎样使用这些功能，用户将学会：

- 压缩更多的硬盘
- 压缩一张软盘
- 在驱动器的空余空间之外建立一个压缩文件卷
- 查看有关压缩盘的信息
- 重新设置压缩盘的大小
- 重整一个压缩盘，使每个文件的内容放到连续的扇区内
- 改变压缩盘的压缩比
- 减小 DrvSpace 和 DblSpace 所用的内存空间
- 删除或者恢复一个压缩驱动器
- 解除一个驱动器的压缩

45.2　DrvSpace 和 DblSpace 的核心程序

在安装 DrvSpace 和 DblSpace 时，DRVSPACE. BIN 或 DBLSPACE. BIN 变成了 DOS 核心中的第三个文件，它们连接常用的 IO. SYS 和 MSDOS. SYS。如果它们位于启动驱动器的根目录下，启动时，在 CONFIG. SYS 执行之前，随同其他两个核心文件一起装入内存。进入内存后便立即抓住压缩文件卷的控制权，以便在执行 CONFIG. SYS 的时候，压缩驱动器已经处于可用状态了。这就避免了第三方的压缩软件常在启动中出现的问题，这些软件必须通过 CONFIG. SYS 装入。

DOS 最初把 DRVSPACE. BIN 或 DBLSPACE. BIN 装到常规内存的高端，随后，DRVSPACE. SYS 或 DBLSPACE. BIN 程序把它们移到更适合的地方。如果用户的系统建立成使用高端内存块的形式，而且 DRVSPACE. SYS 和 DBLSPACE. SYS 在被装入时使用

了 DEVICEHIGH 命令，那么 DBLSPACE. SYS 或 DRVSPACE. BIN 便把 DBLSPACE. BIN 或 DRVSPACE. BIN 移到高端内存(DOS 把它装到常规内存的顶端，这样做，是因为在将它移到高端内存后，不会在常规内存区内留下不能使用的间隙)。如果不能把 DBLSPACE. BIN 或 DRVSPACE. BIN 移到高端内存，DBLSPACE. SYS 和 DRVSPACE. SYS 就把它移到常规内存中位置较低的地方。在安装 DrvSpace 和 DblSpace 之后，在 CONFIG. SYS 中会出现下述命令，该命令用于装入 DRVSPACE. SYS 或 DBLSPACE. SYS，它们的功能是移动 DRVSPACE. BIN 及 DBLSPACE. BIN：

```
DEVICE=C:\DOS\DRVSPACE.SYS /MOVE
```

或者

```
DEVICE=C:\DOS\DBLSPACE.SYS /MOVE
```

如果系统建立成使用高端内存的方式，在安装 DrvSpace 以后，运行 MEMMAKER，把命令转变成一个 DEVICEHIGH 命令，该命令能把 DRVSPACE. BIN 或 DBLSPACE. BIN 装入高端内存。

45.3 DRVSPACE 或 DBLSAPCE 命令

当在 DrvSpace 及 DblSpace 程序下工作时，可以在一个有菜单和对话框的窗口中工作，或者在 DOS 命令提示符下键入命令，完成相同的功能。本章将介绍如何使用这两种方法 。这些命令的选项很多，不能在一张图中加以总结。因此，在讨论每个功能时，本章都给出使用该功能的句法。本章以 DRVSPACE 为例，如果用户使用的是 DrvSpace，仅仅把 DRVSPACE 换成 DBLSPACE 就可以了。

注释： *在本书的后面命令参考中详细列出了 DblSpace 及 DrvSpace 命令。*

45.4 压缩其他的硬盘

在 DrvSpace 或 DblSpace 已安装好并且已建立了第一个压缩驱动器后，DRVSPACE 或 DBLSPACE 命令打开如图 45.1 所示的窗口。用户的所有压缩驱动器都列在一个表中。用户使用该窗口去维护压缩驱动器，例如重设驱动器的大小，以及压缩更多的驱动器。

怎样压缩另外一个硬盘

1. 把要压缩的驱动器上的所有重要文件全部备份下来。
2. 确信要压缩的硬盘上有 1.1MB 的空闲空间。如果没有，把一些文件移到软盘或另外一个压缩硬盘上，以便腾出空间来完成压缩操作。
3. 键入 DBLSPACE 或 DRVSPACE 命令，打开 DblSpace 或 DrvSpace 窗口。
4. 选择 Compress Existing Drive 项。

   DrvSpace 或 DblSpace 列出用户所有适合于压缩的驱动器。如果没有列出任何项，便说明用户没有可压缩的磁盘。
5. 选择要压缩的磁盘，然后按屏幕上的指示去做。DrvSpace 或 DblSpace 将对所选的磁盘完成压缩。

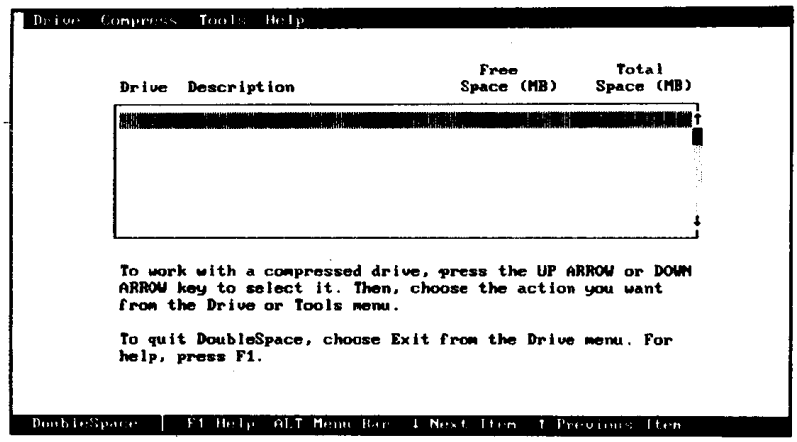

图 45.1　DrvSpace 或 DblSpace 窗口

若要压缩另外的硬盘，选择 Compress 菜单中的这项命令：Compress Existing Data or Create New Drive(压缩空闲空间)。DrvSpace 或 DblSpace 将向用户询问选择准备压缩的磁盘，然后像先前所作的那样进行压缩。压缩完成后，在 DrvSpace 或 DblSpace 主菜单的驱动器列表中，将会出现一个新的完成压缩的驱动器。

另外，用户亦可在 DOS 命令提示符下键入下面格式的命令：

DRVSPACE /COMPRESS drive1 [/NEWDRIVE=drive2] [/RESERVE=megabytes]

该命令压缩磁盘上的文件。Drive1 选项指明主驱动器，Drive2 给主驱动器取一个新的名字，只在用户不想使用 DrvSpace 或 DblSpace 的缺省值时才使用 drive2。使用/RESERVE 选项说明在主驱动器上保留多少不压缩的空间，如果省略该选项，DrvSpace 或 DblSpace 仅留下最小的空间不加以压缩。

注释： 压缩和解除压缩所用的少量时间，会妨害一些对时间比较敏感的程序的执行，例如动画程序和声音合成程序。这些应用程序不能存储在压缩盘上。

要在驱动器上的空闲空间之外建立一个压缩盘，键入下列格式的命令：

DRVSPACE /CREATE drive1 [/NEWDRIVE=drive2] [/SIZE=megabytes|/RESERVE=megabytes]

例如要在 D 盘上建立一个 10MB 空闲空间之外的压缩盘，可键入下述命令：

DRVSPACE/CREATE D/SIZE=10

如果用户想压缩另外一个硬盘，这可能要花好几个小时(并有可能失败，使得用户不得不从文件备份中恢复硬盘上的所有数据)，下面的练习可为用户提供一些指导：

1. 将准备压缩的硬盘上的所有重要文件备份下来。
2. 确保准备压缩的硬盘上至少有 1.1MB 的空闲空间。如果没有，把一些文件移到软盘上，以留下足够空闲空间用于压缩。
3. 键入 DBLSPACE 或 DRVSPACE 命令，打开 DblSpace 或 DrvSpace 窗口。
4. 选择 Compress Existing Drive 一项。

DblSpace 或 DrvSpace 列出用户的所有能够被压缩的磁盘。如果没有任何项被列出，则说明没有能够被压缩的磁盘。

5. 选择准备压缩的磁盘，然后按屏幕上显示的指示去做。

DblSpace 或 DrvSpace 将完成压缩。

45.5 压缩软盘

用户可以在软盘上建立压缩盘。假如需要把一大堆文件移到拥有 DrvSpace 或 DblSpace 的客户机上，采用压缩工具可以节约大量的磁盘空间。或者，如果需要把一项工作结果从硬盘上备份下来，而且需要在软盘上保存一年或几年，这时候使用压缩功能会节约更多的软盘。

用户可以使用 Backup 完成这些任务，但是 DrvSpace 或 DblSpace 提供了一些更现实的优点。使用 DrvSpace 或 DblSpace，用户仅仅是来回地拷贝文件，在要访问这些文件时，没必要处理软盘上的备份目录，另外，可以添加、改变和删除软盘上的文件。但使用 Backup 时，一旦制成了备份，就不能被改动。不足之处也有，例如 DrvSpace 或 DblSpace 不能提供 Backup 的警告信息，例如碰到错误代码时发出的警铃声。再者，当用户从一个压缩盘上读取数据时，DrvSpace 或 DblSpace 自动解除数据的压缩，因此不能以压缩的方式进行通信，但 Backup 能。

准备压缩的软盘必须是经过格式化且有至少 0.65MB 空闲空间。另外，DrvSpace 或 DblSpace 不能压缩 360MB 的软盘。假如 DrvSpace 或 DblSpace 驱动器列表中没有出现要压缩的软盘，便说明驱动器中的软盘不符合要求。

如果是在一张新的软盘上建立压缩盘，那么把软盘插入驱动器并选择 Compress Existing Drive 命令(在软盘上不能使用 Compress Greate New Drive 命令)。DrvSpace 或 DblSpace 显示可以压缩的驱动器列表(如图 45.2 所示)。选择软驱并按 Enter 键，DrvSpace 或 DblSpace 便把几乎整张盘转化成一张压缩驱动器。安装完毕后，便把它加到 DrvSpace 和 DblSpace 窗口中的驱动器列表里去。

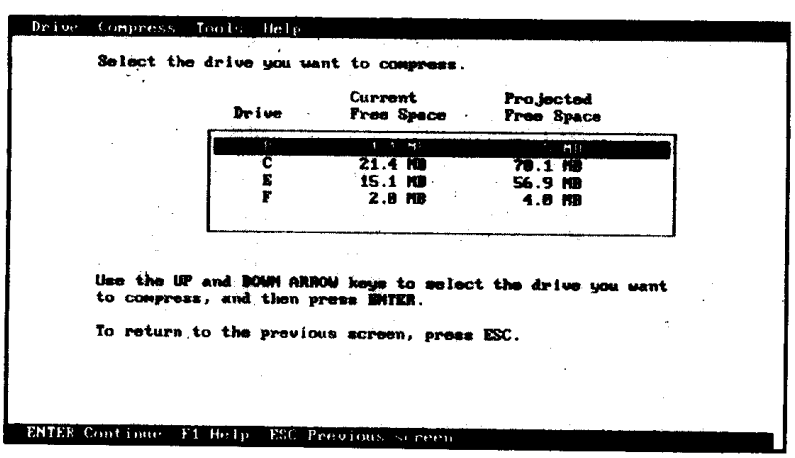

图 45.2 Compress Existing Drive 的屏幕显示

另外，也可以键入前文中说过的 DRVSPACE /COMPRESS 命令。

假如用户已经安装了 DrvSpace 或 DblSpace，并打算压缩一张软盘，可按下列步骤进行：

1. 找一张至少有 1.1MB 空闲空间的软盘(已经格式化但还没有使用过的软盘最好。)
2. 把软盘插到适当的驱动器中去。
3. 备份软盘上的重要文件。
4. 键入 DBLSPACE 或 DRVSPACE 命令。

   屏幕上便出现 DblSpace 或 DrvSpace 窗口。
5. 选择 Compress Existing Drive 命令。

   DblSpace 或 DrvSpace 显示合法驱动器列表，包括软盘在内(如果其中没有软盘，说明它没有足够的空闲空间)。
6. 从列表中选择软盘并按下 Enter 键。

   DblSpace 或 DrvSpace 提醒用户首先备份驱动器中的文件。
7. 选择 Continue(继续)项。

   DblSpace 或 DrvSpace 便显示大致要用的时间。
8. 按 C 键，确认继续进行。

   DblSpace 或 DrvSpace 便开始压缩指定的软盘。当压缩完成后，在驱动器列表中又出现一个新的压缩驱动器。

和硬盘一样，软盘接受一个新的名字，而 CVF 接纳了旧的驱动器名。压缩盘位于原来的软盘上，驱动器 A 用于访问压缩文件卷(Volume)，同时宿主驱动器称作驱动器 L。

当用户把磁盘从驱动器中拿出以后，该驱动器便自动卸下。如果下一张插入的磁盘是没有压缩的，它便接受正常的驱动器字母。但是，如果又插入另一张压缩过的软盘时，DrvSpace 或 DblSpace 又自动把它装上。

注释： 只有在 DOS 6.0 中，DblSpace 才不能自动安装上压缩软盘。在插入一张磁盘以后，必须键入 DBLSPACE /MOUNT d:命令，例如 DBLSPACE /MOUNT A:命令。

在压缩一张软盘以后，该软盘在任何安装了兼容的 DrvSpace 及 DblSpace 的系统上都能安装或阅读。所谓“安装”，是指该系统必须有至少一个硬盘是压缩盘，这样 DblSpace 或 DrvSpace 便可作为系统核心的一部分装入计算机。

如果是用 DrvSpace 压缩一张软盘，用户便可以在任何装有 DrvSpace 的系统上使用它，但不能在装 DblSpace 的系统上使用，也不能在装有其他压缩程序或根本没有装压缩程序的计算机上使用。如果软盘是用 DblSpace 压缩的，它便能在装有 DblSpace 的系统上使用，但不能在装有其他压缩程序或没有装压缩程序的系统上使用。

一个装有 DrvSpace 的系统可以读出由 DblSpace 建立的软盘上的数据，但不能写入。用户必须按前面所示的命令自己装上该磁盘。要完成写操作，必须把 DblSpace 文件卷转变成 DrvSpace 文件卷，下面就会讲如何转变过程。

怎样把 DblSpace 软盘转变为 DrvSpace 软盘

1. 键入 DRVSPACE /MOUNT A:,装上驱动器(如果磁盘在 B 驱动器中,就用 B:代替 A:)。
2. 键入 DRVSPACE 命令打开 DrvSpace 窗口。
3. 从驱动器列表中选择软驱。
4. 选择 Tools Convert DoubleSpace 项。

注释: 在 DOS 6.0 下(仅在此种情况下),DblSpace 在软盘上没有压缩的部分装有 READTHIS.TXT 文件,该文件说明软盘上有一个压缩驱动器,并说明怎样装上它。如果用列目录命令列出一个未装入的磁盘上的目录,则会看到 READTHIS.TXT 文件。

提示: 在驱动器菜单中包含有 Unmount 命令,该命令既可用于硬盘也可用于软盘。但没有多少使用的必要。在 DRVSPACE 命令中亦有一个/UNMOUNT 选项。

45.5.1 在宿主驱动器上的文件

DrvSpace 或 DblSpace 都会在宿主驱动器上留下一定数量的空间来存储没有被压缩的文件。缺省时,它在硬盘上留下 2MB 空间。但在压缩驱动器第一次建立时,用户可以说明要保留的空间大小。而且,以后还可以重新设置压缩驱动器的大小。

注释: 即使宿主驱动器在 DrvSpace 或 DblSpace 压缩数据时只有 1.2MB 的空闲空间,但在压缩完成后,会有足够的空间来为主驱动器留下 2MB 的空闲空间。

如果宿主驱动器是启动驱动器,DOS 的系统文件将被存储在没压缩的区域内,和 DRVSPACE.INI 或 DBLSPACE.INI 放在一起。所有这些文件都是隐含文件,但可以通过在 DIR 命令中写上/A 选项去查看它们,同时还可以看到 DRVSPACE.nnn 或者 DBLSPACE.nnn,它们是压缩了的卷文件。如果压缩的是数据,则 nnn 为 000,否则它是除 000 外的数字。

警告: 不能对 DRVSPACE 或 DBLSAPCE 作删除、移动或其他操作,否则将使数据和压缩驱动器之间失去联系。

如果被压缩的驱动器上有 Windows 的永久性对换文件,DrvSpace 或 DblSpace 就把对换文件放在没被压缩的空间内,因为被压缩后对换文件就不能工作。如果用户还有其他不能压缩的文件,便不得不在压缩完成后,把它们从压缩驱动器移到非压缩驱动器之上。如果用户随后发现某个程序不能在压缩驱动器上正常工作,便可以把它移到宿主驱动器中。如有必要,重新设置被压缩驱动器的大小以留给主驱动器更多的空间。有时候不得不重新配置甚至重装该程序,以适应改变了的驱动器名。

答疑: 在压缩了硬盘以后,一个应用程序显示拷贝保护信息,并且拒绝执行,怎么办?

拷贝保护程序的一个办法是记住安装时它被放置的位置,用这种方式保护的程序被压缩后有可能拒绝执行。这不是因为被压缩的缘故,而是因为它们被移动了。在

这种情况下，就不得不重装该程序。

45.6　维护压缩驱动器及它们的宿主驱动器

在 Drive 菜单及 Tools 菜单中，包含了几个为维护被压缩驱动器而设的选项。这些选项在 DRVSPACE 命令中也一样有效。

45.6.1　为压缩驱动器安装高速缓存系统

SMARTDRV 不能提高一张压缩盘的存储速度，但它能提高宿主盘的存储速度，而这也使得压缩驱动器上数据一起被缓存。如果用户有一个压缩驱动器，那么 SMARTDRV 的状态报告如下所示：

```
                Disk Caching Status
drive   read cache   write cache   buffering
------------------------------------------------
  A:      yes            no            no
  B:      yes          no          no
  C:*     yes          no          no
  H:      yes          no          no
* Compressed drive ceched via host drive.
```

在这个例子中，H 是压缩驱动器 C 的宿主盘。

警告： 不要对压缩驱动器及其宿主盘作写缓存。因为在 DOS 花时间去把那些延迟数据写到压缩盘之前，由于缓存系统的运行，会丢失大量的数据。

如果用户使用 SMARTDRV 并有一个压缩盘，便可以查看驱动器的缓存状态。亦即是键入 SMARTDRV 命令，查看宿主盘及其上的压缩驱动器是如何被缓存的。

45.6.2　查看压缩驱动器的信息

图 45.3 示出了一个对话框，这个对话框是在用户选择一个压缩驱动器并且选择 Drive Info 命令后打开的。

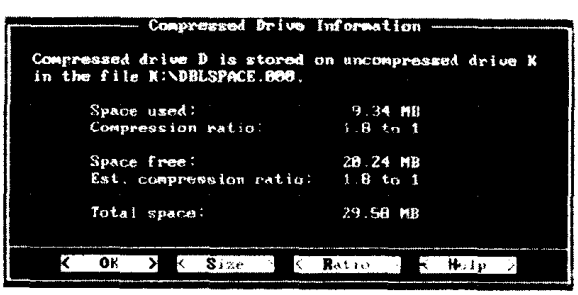

图 45.3　压缩驱动器信息对话框

用户亦可在命令提示符下键入下面的命令来查看相同的信息：

DRVSPACE [/INFO] drive

/INFO 选项是非强制性的。当用户键入 DRVSPACE 或者 DBLSPACE 命令，如果命令

中带有驱动器名字但没有选项说明特别的功能时，/INFO 便是缺省值。(如果用户也忽略了驱动器名字，则 DrvSpace 或 DblSpace 把窗口打开)。

试练习查看压缩盘的信息：

1. 键入 DRVSPACE 或 DBLSPACE 命令，打开压缩程序的窗口。
2. 选择要查看信息的驱动器。
3. 选择 Drive Info 项。

DblSpace 或 DrvSpace 便显示相应驱动器的压缩信息。

如果要查看所有非网络驱动器、压缩驱动器及其他驱动器，可键入下述命令：

```
DRVSPACE /LIST
```

该列表说明驱动器是否是本地硬盘驱动器、压缩驱动器、软盘驱动器、没有插入盘片的可移动媒体、可被 DrvSpace 或 DblSpace 使用的空闲驱动器名，等等。(RAM 虚拟盘被看作本地硬盘驱动器。)同时，如果能够的话，还显示出每个驱动器上已经使用了的空间和空闲空间大小。对于每个已经安装好的压缩驱动器，报告中显示出压缩驱动器的卷名(及其宿主驱动器名)。该功能不能在 DrvSpace 或 DblSpace 的窗口中使用。

练习查看系统中的所有驱动器：

1. 把一张软盘放进驱动器(最好是张压缩盘)。
2. 回到 DOS 命令提示符下。
3. 键入 DRVSPACE /LIST 或 DBLSPACE /LIST。

45.6.3 重新设置驱动器的大小

如果用户准备把 2MB 的 Windows 永久性对换文件建在宿主盘上，但宿主盘上却只剩下 1.3MB 的非压缩空间，如果压缩驱动器上还有空间可用，便可以通过缩小压缩驱动器，以获得更多的可用空间。

另外，如果在宿主盘上还有 5MB 的空闲空间，但压缩驱动器的空间快要用完了，就可以占用宿主盘的空闲空间以扩展压缩驱动器的大小。

在 Information 对话框中，可以按 Size 按钮，或者选择 Drive Change Size 命令，从而打开用于重设当前选定的驱动器大小的对话框(如图 45.4 所示)。

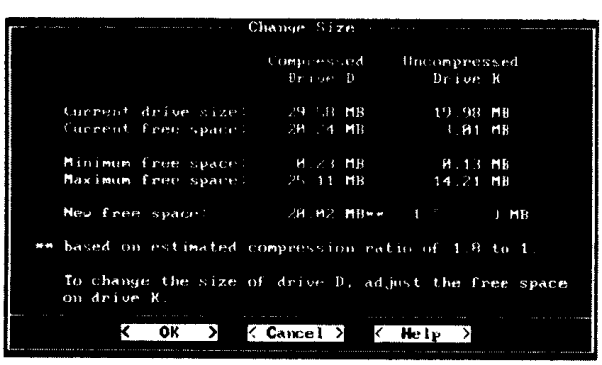

图 45.4 Chang SIze 对话框

键入下面的命令，可以在 DOS 命令提示符下改变驱动器的大小：

DRVSPACE /SIZE=megabytes drive

或者

DRVSPACE /SIZE /RESERVE=megabyte drive

例如，要把驱动器 C 的大小改变为 60MB，可键入下面命令：

DRVSPACE /SIZE=60M C:

所指明的大小是宿主盘上的 CVF 大小，而不是估计的容量。如果不对大小加以说明，如在 DBLSPACE /SIZE C:命令中，则 DblSapce 或 DrvSpace 把驱动器尽可能缩小。

如果要改变驱动器 C 的大小，而给宿主驱动器留下 20MB 的空间，可如下键入命令：

DRVSPACE /SIZE /RESERVE=20 C:

如果说明/RESERVE=0 选项，DrvSpace 或 DblSpace 尽可能地把压缩驱动器扩大。

如果用户准备把驱动器缩小(特别想缩到最小)，那么首先必须对驱动器重整，下面就讲述有关这方面的内容。

45.6.4　重整压缩驱动器

和正常情况不同，重整一个压缩驱动器并不能提高访问速度，但是，在用户想把压缩驱动器缩小时，重整能够帮助把所有有空闲空间集中在一起。要重整一个压缩驱动器，可以选择 Tools Defragment 命令或键入如下格式的命令：

DRVSPACE /DEFRAGMENT [drive]

如果省略了驱动器名字，就对当前驱动器作重整。在命令中加上/F 选项表示更全面地重整。

注释： 如果一个很大的压缩驱动器上存放有大量的数据，对它重整则要花几个小时。

如果想把驱动器缩小到最小，就必须特别严格地将驱动器重整，下面给出了重整的步骤：

如何彻底地重整压缩驱动器
1. 用 DOS 的 DEFRAG 命令重整宿主盘。
2. 使用带有/F 选项的 DRVSPACE /DEFRAGMENT 命令将压缩驱动器作全面重整。
3. 使用不带/F 选项的 DRVSPACE /DEFRAGMENT 命令作微调。

45.6.5　改变压缩比

压缩比表示通过压缩节约了多少空间。例如，压缩比 2.3:1 意思是，每一个 2.3KB 的非压缩文件，压缩后只占 1KB 的空间。一些类型的文件的压缩比比其他文件要大，例如一个典型的字处理文件，压缩比大致为 2:1，但一个位映象的图形文件，压缩比可以达 20:1。其他类型文件的压缩比仅为 1.1:1。用户压缩驱动器的总压缩比和驱动器上的所有文件都有关，一般为 2:1。

当用 DIR 或 CHKDSK 报告一个压缩驱动器上的可用空间时，它们报告的实际字节数可能没有意义。更重要的是用户能够在驱动器上存储多少数据。要对此作计算，可用 DrvSpace 或 DblSpace 根据驱动器上已存在的数据作出压缩比估计值，再乘上驱动器上的空闲空间。得出的结果可以估计能够把多少没有压缩的数据写到驱动器上，这可能不太精确，其精确程度取决于写到驱动器上的文件的性质。DIR、CHKDSK 及其他一些报告可用空间大小的程序都使用 DrvSpace 或 DblSpace 提供的估计值。

另外，DIR 命令含有一个新的选项，用于显示压缩驱动器上的压缩比。选调/C 显示一个基于 8KB 大小的簇所作出的压缩信息。如果再加上 H 选项，即/CH，它便会显示基于宿主盘上的簇的大小所作出的压缩信息，这要精确一些。DIR /CH 命令列出的结果如下：

```
Volume in drive D has no label
Volume Serial Number is 13EF-2218
Directory of D:\FRENCH

.            <DIR>          02-21-94   12:28p
..           <DIR>          02-21-94   12:28p
CHKLIST   MS         54     12-22-94    3:58     4.0 to 1.0
FLASHCRD  BAS     30896     05-05-94   10:42a    2.3 to 1.0
FLASHCRD  EXE     91084     05-05-94   10:43a    1.3 to 1.0
VOCAB     BAS     31265     05-05-94   11:42a    2.2 to 1.0
VOCAB     EXE     91242     05-05-94   11:42a    1.3 to 1.0
WORK      BAS     13054     05-04-94   11:59a    1.6 to 1.0
                  1.4 to 1.0 average compression ratio
        8 file(s)   257595   bytes
                  21102592 bytes free
```

用户可以看到所列的每个文件的压缩比，以及所列文件的平均压缩比。每个文件列出的大小是没有压缩过的大小，它告诉用户当这些文件读入内存时要占用的内存大小，以及把它们拷贝到一个非压缩驱动器上时要占用的空间大小。最后一行的空间大小估计值是由 DrvSpace 或 DblSpace 提供的。

用户亦可按压缩比大小的顺序用 DIR 列出文件。选项/OC 使 DIR 能够从压缩比最小到最大的顺序列出文件，而/O—C 的顺序则相反，按从最大到最小的顺序列出文件。

注释： 如果对一个压缩驱动器使用 SUBST 命令，DrvSpace 及 DblSpace 不再提供驱动器空间的估计值，而显示出实际的字节数。

当 DrvSpace 或 DblSpace 装载压缩驱动器时，它便更新其压缩比估计值，以适合实际的压缩比，但用户可以改变压缩比的估计值。例如，如果用户正在装入一个应用程序，该程序有 40MB 的 TIF 文件，程序及其他文件仅占 6MB，压缩驱动器的当前压缩比是 2.1:1，DblSpace 或 DrvSpace 所报告的驱动器上的剩余空间是 30MB。此时，应用程序的安装程序就会拒绝安装，因为驱动器上的空间不够。但 TIF 文件的压缩比可达 20:1，因此驱动器上还有足够的空间可以安装该应用程序。此时要完成安装工作，就得改变压缩比。

要改变压缩比，选择 Information 对话框中的 Ratio 选项，或者选择 Drive，Option，Change Ratio 等项(如图 45.5 所示)。

也可以键入下述命令：

```
DRVSPACE /RATIO[=r.r] [drive |/ALL]
```

比例取值范围从 1.0 到 16.0。如果省略比值，DrvSpace 或 DblSpace 把估计值调整为当

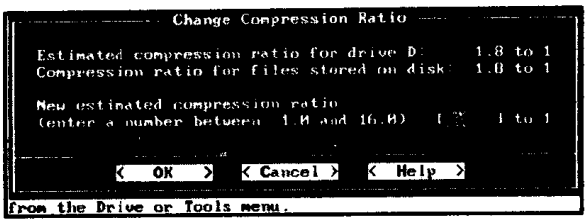

图 45.5　改变压缩比对话框

前的压缩比值。/ALL 选项把所有已装上的压缩驱动器的压缩比设定为一个特定的值。如果没有说明驱动器名或/ALL 选项，DrvSpace 或 DblSpace 便为当前驱动器设置压缩比。

如果要调整所有已装上的压缩驱动器的压缩比值，以适合它们各自的实际值，可键入下面命令：

```
DRVSPACE /RATIO /ALL
```

要完成前面例子中的安装工作，用户先键入 DRVSPACE /RATIO=16.0 C: 命令，这样 DblSpace 或者 DrvSpace 便会报告驱动器上有足够空间的信息。装好新的应用程序之后，可以键入 DRVSPACE /RATIO C:命令，计算实际的压缩比。

45.6.6　格式化压缩驱动器

如果是第一次建立压缩驱动器，就没有必要将它格式化。但可以通过格式化而删除驱动器上的所有数据。但必须使用 DrvSpace 或 DblSpace 的格式化功能，不能使用 DOS 的格式化程序。

警告： *用户不能恢复已格式化压缩驱动器上的内容。*

格式化一张压缩盘时，可以选择 Drive Format 命令或者键入下述命令：

```
DRVSPACE/FORMAT drive
```

45.6.7　扫描驱动器上的错误

在 DOS 6.2 中，可以用第 27 章“CHKDSK 和 SCANDISK”中所讲的 SCANDISK 扫描压缩驱动器。但是，在扫描压缩驱动器时，SCANDISK 坚持首先扫描宿主驱动器。选择 Yes 扫描宿主驱动器，扫描完成后，选择 Next Drive 扫描压缩驱动器。

在 DOS 6.0 下，使用 DblSpace 窗口下的 Tools Chkdsk 命令检查压缩驱动器上的目录结构，或者键入如下命令：

```
DBLSPACE /CHKDSK [/F] [drive]
```

如果希望 DblSpace 更正它所发现的错误，就在命令中使用/F 选项。

在 DOS 6.2 下，如果选用 ChkDsk 功能，便会出现一些信息，告诉用户在驱动器上运行 SCANDISK。

45.7　释 放 内 存

DrvSpace 或 DblSpace 有两个选项能帮助释放 DrvSpace 或 DblSpace 所占用的内存空

间:Last Drive(最后的驱动器)及 Removable Media Drives(可移动媒体式驱动器)。

DrvSpace 或 DblSpace 都自动地在系统中保留一定数目的驱动器名字,以便在必要时能够装上这些驱动器,但是它保留的名字一般都超过了实际需要。由于它保留的每个名字都要占用 24KB 的内存空间,因此有必要删除一些多余的名字。选择 Tools,Options 命令查看并可以改变最后的驱动器名字。例如,假如只建立了一个压缩驱动器,而且,不想再建立压缩驱动器了,另外,也没有使用任何 RAM 磁盘或其他假驱动器,这样最后的驱动器名是 C 而不是 DrvSpace 或 DblSpace 所分配的名字,此时,便可以很安全地把 DrvSpace 或 DblSpace 的最后驱动器名改为 D,从而节约 168KB 的内存空间。

Tools,Option 命令也允许用户改变所用的可移动媒体式驱动器的数目。DrvSpace 或 DblSpace 计算所有的软盘驱动器,可移动盒式驱动器和快闪内存卡(Flash memory card)驱动器,并且保留足够的内存,以使能够在每个驱动器上装上压缩驱动器。用户可以把驱动器的数目减少一些,以满足自己所需为准,这样可以节约一部分内存。例如,假设有两个软盘驱动器而没有其他可移动媒体式驱动器,并假设只在 A 驱动器上装压缩驱动器,就可以把可移动媒体式驱动器的数目减少为 1。如果不想在任何一个软盘驱动器上建立压缩驱动器,就把数目减少至 0。

45.8 删除压缩驱动器

假如想从一个宿主驱动器上删除压缩驱动器及其上的所有数据,可选用 Drive Delete 命令或键入下述格式的命令:

```
DRVSPACE /DELETE drive
```

删除压缩驱动器时也同时把它卸下来。如果是通过压缩原来盘上的数据而建立的压缩驱动器,而且被赋予了宿主驱动器的原始名字,那么删除后,宿主驱动器将恢复其原始名字。假如是从空闲空间中划分一部分出来建立的压缩盘,而且分配了一个新的驱动器名,那么删除后,这个驱动器名就可以用作其他用途。

可以用 UNDELETE 恢复被删除的 CVF,就像其他 DOS 文件可被恢复一样。恢复后,用户就不须装上它并再使用。

45.9 解除驱动器的压缩方式

可以选用 Tools Uncompress 选项或键入命令 DRVSPACE /UNCOMPRESS d:(DOS 6.0 没有此命令)来解除对一个驱动器的压缩。但首先应该备份盘上的所有重要文件。和压缩时的情形一样,有时候在解除压缩时也会失败,因此必须事先保护必要的数据。

如果 DblSpace 或 DrvSpace 认为驱动器没有足够的空间来存储所有解压缩后的文件,它便拒绝执行。用户必须把足够多的文件移到别的驱动器上,以获得更多的空间。用户也可以把这些文件删除。

当解除 PC 上最后一个压缩驱动器的压缩时,DrvSpace 或 DblSpace 把执行压缩功能的 DrvSpace 移出内存。当回答 Yes 后,压缩软件在启动时便不再装入。但是,它存储在 DOS 目录下,用户随时可以调用。如果手中还有压缩过的软盘,便不能把压缩软件从内存中移出,这

样，才能装上软盘并访问其中的数据。

解除压缩是 DOS 6.2 中的新设功能，DOS 6.0 中没有这个命令。这里将介绍如何在 DOS 6.0 中解除一张盘的压缩，并且把 DblSpace 从内存中移出。

怎样用 DOS 6.0 解除驱动器的压缩

1. 拷贝所有准备保护的文件(亦可用带压缩选项的 Backup 完成这项工作)。
2. 删除压缩文件卷。

   解除压缩后，DrvSpace 或 DblSpace 卸下文件卷，宿主盘得以恢复其原来的名字。
3. 把拷贝(或备份)下来的文件恢复到宿主驱动器上去。

45.10　自 测 练 习

如果已经安装了 DblSpace 或 DrvSpace，就可以练习一下本章涉及到的技术。下面各项中的黑体字部分要花费很长的时间(可能几个小时)，如有必要，才可花时间练习。

1. **压缩另外一个硬盘。**
2. 压缩一个驱动器上的空闲空间，建立一个新的驱动器。
3. 压缩一张软盘，然后，从驱动器中拿出软盘，以便卸下软盘驱动器上的压缩驱动器；再重新插进去，并重新装上。
4. 查看所有驱动器的列表，看是否被压缩。
5. 把软盘上的压缩驱动器缩小到最小。然后再把它扩展到最大。
6. **全面重整宿主压缩驱动器。**
7. 改变一个压缩驱动器的压缩比估计值，然后，让 DblSpace 或 DrvSpace 计算实际的压缩比。
8. 如果可能，减少 DrvSpace 或 DblSpace 所占用的内存。
9. **为一个不需要保持压缩状态的压缩驱动器解除压缩。**

第 46 章 数据操作的其他技巧

简介： 本书已经讲述过格式化磁盘以及比较文件的基本知识，但有时候这些基本知识是不够的，特别是在和版本较旧的系统打交道的时候。

46.1 基本技能以外的知识

FORMAT 命令主要用于为用户自己的系统格式化磁盘，但有时候不得不在密度不同的驱动器或 DOS 版本较低的计算机上使用自己系统格式化的软盘。FORMAT 拥有一系列选项来处理这类问题。

正如前面所说，FC 是比较两个 ASCII 文件的最好工具，但有时用户打算忽略两个被比较文件中的某些差别。例如比较两个 AUTOEXEC. BAT 的拷贝时，可能希望忽略字母的大小写或空格数目的差别。假如这两个文件中的命令完全相同，只要不考虑字母的大小写和空格，它们的功能是完全相同的，FC 包含一系列选项来处理诸如忽略大写字母和小写字母间的差别或空格上的差别。本章还会讲如何应付 FC 在使用其缺省设置进行比较时所出现的不同步现象。

用户也许想知道如何使用 DOS 的验证功能拷贝文件并且在发现文件中的错误时便停止工作。本章将阐述 DOS 的验证工作是如何进行的，以及为什么有时候也会有疏忽。

本章包括下述内容：

- 为低密驱动器格式化磁盘
- 为旧的 DOS 版本格式化磁盘
- 在比较两个 ASCII 文件时忽略字母大小写及空格
- 在比较两个 ASCII 文件时的扩大重新同步功能
- 识别 DOS 的验证功能出现的疏忽

46.2 格式化非缺省容量的磁盘

用户可能已多次用 FORMAT 命令为自己的系统准备磁盘，但当为一台具有同样尺寸但不同容量的驱动器准备磁盘时怎么办呢？例如，自己的驱动器具有 3 ½英寸 1. 44MB 的容量(高密度)，但要为 3 ½英寸 720KB(双密)的驱动器准备磁盘。如果另一台计算机的驱动器的密度比自己的要高，这当然不会有问题，按常规操作进行格式化即可，另一台驱动器也能读写格式化的磁盘。任何 1. 44MB 3 ½英寸的驱动器能够对 720KB 的磁盘读写；任何 1. 2MB 5 ½英寸的驱动器能够对 360KB 的磁盘读写。

但反过来就不同了，双密度驱动器不能读高密度磁盘。因此，如果用户手中的驱动器为高密度，要为双密度的驱动准备磁盘，则必须做一些额外的工作。

图 46.1 示出了 FORMAT 命令的另外一些选项。/F 选项最常用，它用于格式化非缺省容量的磁盘。假如准备在 1.44MB 的驱动器 A 上格式化一张 720KB 的磁盘，则在命令中必须说明 F 选项的值，即：FORMAT A：/F：720。

FORMAT

格式化磁盘。

句法：

FORMAT drive [/U] [/V[:label]] [/B] [/F:size]

参数和选项：

drive	指明要格式化的驱动器。
/U	作无条件格式化。
/V[:label]	给被格式化的磁盘指明一个卷标。卷标最多可达 11 个字符。如果省略/V 或者使用了/V 但没有指明卷标，在格式化完成后，DOS 便向用户询问是否加上卷标。但/V 选项不能和/F:160 或/F:320 一起使用。
/B	在一个新格式化的盘上为系统文件保留空间。
/F:size	指明要格式化的软盘容量。DOS 支持的容量有：
	160KB　5 ¼英寸，单面双密度，每磁道 8 个扇区，共 40 磁道。
	180KB　5 ¼英寸，单面双密度，每磁道 9 个扇区，共 40 磁道。
	320KB　5 ¼英寸，双面双密度，每磁道 8 个扇区，每面 40 磁道。
	360KB　5 ¼英寸，双面双密度，每磁道 9 个扇区，每面 40 磁道。
	1.2MB　5 ¼英寸，双面高密度，每磁道 15 个扇区，每面 80 磁道。
	720KB　3 ½英寸，双面双密度，每磁道 18 个扇区，每面 40 磁道。
	1.44MB　3 ½英寸，双面高密度，每磁道 18 个扇区，每面 80 磁道。
	2.88MB　3 ½英寸，双面四倍密度，每磁道 36 个扇区，每面 80 磁道。

图 46.1　FORMAT 命令总结

使用非缺省值容量参数并不是总是成功，有时也会出现问题。要提高成功率，则必须：

- 使用先前没被格式化过的新盘。
- 使用指定容量的磁盘。例如不要在 720KB 的磁盘作 1.44MB 的格式化。

如果指定的容量大小和驱动器不相符（如为 3 ½英寸的驱动器指定大小为/F:1.2），则会显示 Parameters not supported on drive 或者 Incompatible parameters 等信息。如果磁盘的容量不正确，FORMAT 有可能仍然会格式化（但会出现很多坏扇区），也有可能显示信息 Invalid media or track 0 bad，disk unusable，而拒绝格式化。

提示： FORMAT 的/AUTOTEST 选项使 FORMAT 在执行时没有任何提示。但是要仔细——因为在它开始工作之前，用户不能改换驱动器中的软盘。

如果用户拥有高密度的驱动器，其身边就不太可能有还没有格式化的双密度软盘，鉴于此，这里不打算提供练习。假如用户的确需要格式化双密度软盘，最好去买一些新的（或从朋友那儿去取一些）容量正确的软盘。

46.3 为旧的 DOS 版本格式化软盘

DOS 1.0 或更老的版本所使用的软盘均为 5 ¼英寸的软盘，软盘的格式是每磁道 8 个扇区，每面 40 个磁道。单面磁盘总容量为 160KB，双面磁盘总容量为 320KB。从理论上讲，可以使用带/F:160 或/F:320 的 FORMAT 命令格式化 160KB 或 320KB 的磁盘，但必须明白的是自己的驱动器有时却做不到这一点。这些旧型磁盘不能有卷标，因此在格式化时不要给它指定卷标。

本来，可以在 FORMAT 命令中包含/S 选项来制备一张启动盘。但是，当为旧的 DOS 版本准备启动盘时，FORMAT 把当前的系统文件拷贝到磁盘上就有些不适当了。如果目的计算机上使用的 DOS 是 DOS 5.0 或更高的版本，在格式化磁盘时就不需要做任何特别的事情，只需使用 SYS 命令即可建立 DOS 启动盘。但是，在 DOS 5.0 版本以前，DOS 的启动文件必须放在磁盘上开始的地方，所以，如果为这样的系统准备启动盘，要使用/B 选项，以便在磁盘上留下相应的空间，以存放以后 SYS 命令传送的系统文件。

46.4 扩展 FC 的重新同步缓冲区

FC 命令有一系列选项来控制比较工作的完成(如图 46.2 所示)，本部分将阐述如何使用这些选项。

FC

比较两个文件，显示不相同的行或字节。

句法：

FC filepce1 filespec2 [/LBlines] [/resynch] [/C] [/T] [/W]

参数和选项：

filespec1	指明第一个要比较的文件。
filespec2	指明第二个要比较的文件。
/LBlines	指出存于重新同步缓冲区的行数，缺省值是 100。
/resynch	指明在 FC 发现差异之后，文件重新同步之前必须匹配的行数，缺省值是 2。
/C	忽略字母的大小写。
/T	不将制表符(tab)视为空格，即不停止在第 8 个字符的位置上，并且忽略制表符本身。
/W	忽略空白处，即将相连的多个空格视为一个空格。

图 46.2 FC 命令总结

在发现差异之后寻找重新同步的行时，FC 使用重新同步缓冲区中的内容，按缺省值缓冲区存有 100 行。在缓冲区内如果没有发现相匹配的行，便通知用户比较失败。用户可以使用/LBlines 选项指定一个较大的缓冲区。如果匹配失败，便用/LB200 再试试看是否能正常工作。但有可能不能指定一个大小 200 行的缓冲区。

答疑： 在使用 FC 时出现下列消息 FC:Out of memory(内存不够),这是怎么回事?

FC 不能找到足够多的内存来作为其缓冲区,可以用/LB 选项指定一个较小的缓冲区再试一次。

用户可以使用/resynch 选项来改变比较两个文件时必须匹配的行数。/1 把该数目设置为 1,所以只有一行必须匹配;/5 要求必须匹配 5 行。除了指定的数目为 1 以外,可以在不匹配输出中看到相同数目的行数没有输出。如果某几行是匹配的,但却不够指定的行数,也视为不匹配。

假设要比较含有如下数据的两个文件:

```
MEMBERS4
Doherty,Greg,97 Poplar Drive,30038
Fernandez,Davida,95 Poplar Drive,30038
Fernandez,Judi,33 Main Street,30038
Fisher,King,42 Poplar Drive,30038
Gold,Esther,52 Oak Terrace,30033
Gold,Marv,52,Oak Terrace,30033
MacQuown,Dori,41 Spring Drive,30033
MacQuown,Frank,41 Spring Drive,30033
Tabler,Barbara,28 Elm Street,30033
Tabler,George,28 Elm Street,30033
Zoundts,Richard,12 Main Street,30038
MEMBERS6
Doherty,Greg,97 Poplar Drive,30038
Fernandez,Judi,33 Main Street,30038
Gold,Esther,52 Oak Terrace,30033
Gold,Marv,52 Oak Terrace,30033
MacQuown,Dori,41 Spring Drive,30033
MacQuown,Frank,41 Spring Drive,30033
Von Huben, Willie,18 Front Street,30033
Zoundts,Richard,12 Main Street,30038
```

如果比较时使用缺省的重新同步参数,则显示的结果如下:

```
Comparing files MEMBERS4.TXT and MEMBERS6.TXT
***** MEMBERS4.TXT
Doherty,Greg,97 Poplar Drive,30038
Fernandez,Davida,95 Poplar Drive,30038
Fernandez,Judi,33 Main Street,30038
Fisher,King,42 Poplar Drive,30038
Gold,Esther,52 Oak Terrace,30033
***** MEMBERS6.TXT
Doherty,Greg,97 Poplar Drive,30038
Fernandez,Judi,33 Main Street,30038
Gold,Esther,52 Oak Terrace,30033
*****
***** MEMBERS4.TXT
MacQuown,Frank,41 Spring Drive,30033
Tabler,Barbara,28 Elm Street,30033
Tabler,George,28 Elm Street,30033
Zoundts,Richard,12 Main Street,30038
***** MEMBERS6.TXT
MacQuown,Frank,41 Spring Drive,30033
```

```
Von Huben,Willie,18 Front Street,30033
Zoundts,Richard,12 Main Street,30038
* * * * *
```

该报告说明在"Doherty,Greg"和"Gold,Esther"之间的行不匹配,注意"Fernadez,Judi"所在的行在两个不匹配系列中都有,但它不用于重新同步,因为缺省时,FC 寻找系列中相邻的两个相匹配的行。

如果使用参数/1,则输出有所不同:

```
Comparing files MEMBERS4. TXT and MEMBERS6. TXT
* * * * * MEMBERS4. TXT
Doherty,Greg,97 Poplar Drive,30038
Fernandez,Davida,95 Poplar Drive,30038
Fernandez,Judi,33 Main Street,30038
* * * * * MEMBERS6. TXT
Doherty,Greg,97 Poplar Drive,30038
Fernandez,Judi,33 Main Street,30038
* * * * *
* * * * * MEMBERS4. TXT
Fernandez,Judi,33 Main Street,30038
Fisher,King,42 Poplar Drive,30038
Gold,Esther,52 Oak Terrace,30033
* * * * * MEMBERS6. TXT
Fernandez,Judi,33 Main Street,30038
Gold,Esther,52 Oak Terrace,30033
* * * * *
* * * * * MEMBERS4. TXT
MacQuown,Frank,41 Spring Drive,30033
Tabler,Barbara,28 Elm Street,30033
Tabler,George,28 Elm Street,30033
Zoundts,Richard,12 Main Street,30038
* * * * * MEMBERS6. TXT
MacQuown,Frank,41 Spring Drive,30033
Von Huben,Willie,18 Front Street,30033
Zoundts,Richard,12 Main Street,30038
* * * * *
```

把这个报告和前一个报告相比较,注意"Fernandez,Judi"所在的行被用于重新同步,在报告的第一部分中同时也用它作为文件重新同步的起始行。报告的第二部分显示了在"Fernandez,Judi"和"Gold,Esther"之间不匹配的行。

如果使用参数/5,则显示为:

```
Comparing files MEMBERS4. TXT and MEMBERS6. TXT
* * * * * MEMBERS4. TXT
Doherty,Greg,97 Poplar Drive,30038
Fernandez,Davida,95 Poplar Drive,30038
Fernandez,Judi,33 Main Street,30038
Fisher,King,42 Poplar Drive,30038
Gold,Esther,52 Oak Terrace,30033
Gold,Marv,52 Oak Terrace,30033
MacQuown,Dori,41 Spring Drive,30033
MacQuown,Frank,41 Spring Drive,30033
Tabler,Barbara,28 Elm Street,30033
Tabler,George,28 Elm Street,30033
```

```
Zoundts,Richard,12 Main Street,30038
***** MEMBERS6.TXT
Doherty,Greg,97 Poplar Drive,30038
Fernandez,Judi,33 Main Street,30038
Gold,Esther,52 Oak Terrace,30033
Gold,Marv,52 Oak Terrace,30033
MacQuown,Dori,41 Spring Drive,30033
MacQuown,Frank,41 Spring Drive,30033
Von Huben,Willie,18 Front Street,30033
Zoundts,Richard,12 Main Street,30038
*****
```

在这个例子中，FC 不能找到相邻的相匹配的 5 行，所以它不能将文件同步。因此，整个文件都视为不匹配。这个例子对于用户想找出两个文件中不相同的行来说是不利的，因为输出太多。不过，在比较两个程序文件时，使用像/5 这样较大的参数是比较合适的。

46.5　比较两个文件的功能

有时候用户想比较两个文件的功能，而不是严格的按一个字符一个字符地比较。例如，如果想比较两个姓名和地址文件，很可能就不关心文件中的字母是大写还是小写，或者一个文件中有空行而另一个没有，所要关心的仅仅是文件中的姓名和地址是否相同。幸运的是 FC 有些选项可用于忽略诸如字母大小写及空行等偶然差别。

假如用户想比较 CLIENTS.CRD 和 MAILING.CRD 两个文件，但想忽略字母大小写及空格，便可以键入 FC CLIENTS.CRD MAILING.CRD /W /T /C 命令。/W 选项告诉 FC 忽略空白——即空行或多余的空格。/T 选项忽略制表字符和相同数目的空格(8 个)之间的差别，而/C 忽略字母大小写。

假如用户想在比较时忽略大小写及空白，可参考下列步骤：

1. 将目录转到 C:\。
2. 打开 AUTOEXEC.BAT 进行编辑。
3. 存储文件，更名为 AUTOEXEC.TOO。
4. 改变任何命令的大小写。把命令名的大写字母改为小写，反过来亦可。
5. 在一行的前面插入制表字符，但不要插在改变了大小写字母的行的前面。
6. 在一行中插入多余的字符。
7. 在行之间插入空行。
8. 存储文件，退出编辑器。
9. 键入 FC AUTOEXEC.BAT AUTOEXEC.TOO |MORE。
   所有被改变的地方都视为不同而报告出来。
10. 所有 FC AUTOEXEC.BAT AUTOEXEC.TOO /W /T /C |MORE 命令。
    报告中就不再出现不同的地方。

46.6　了解 DOS 的验证功能

系统提供了几种级别的保护措施，以使系统在读数据和写数据的时候，不会因为硬件和

存储媒体的问题而损坏数据。

46.6.1 CRC(循环冗余校验)

正常情形下,在每次写扇区时,磁盘驱动器的控制器都会作循环冗余校验(CRC)。它计算出每个扇区中的字节数并把计算结果存放在位于每个扇区之前的扇区控制区里。然后,它重新读扇区,并再计算一遍,并把这次计算的结果和存放的结果比较,如果不同,控制器就向DOS 报告写出错。在读扇区时,CRC 又重做一遍该操作。如果计算值和存储值不匹配,就报告读出错。CRC 可以发现数据传送到磁盘控制器所发生的所有错误,但用户有时还需要它能更确切地指出错误之所在。

46.6.2 DOS 的验证功能

用户可以要求 DOS 验证它所写入的每一个扇区。当 DOS 验证时,它把所写的数据和相应内存中的数据作比较,以确保准确无误。如果在验证过程中出错,DOS 就报告写出错。

46.6.2.1 一般性验证

正常情况下,DOS 依赖于硬件的 CRC 验证功能,而不额外验证所写数据的正确性。用户可以在命令提示符下使用 VERIFY ON 命令,强迫 DOS 验证所有写入磁盘的数据。键入没有参数的 VERIFY 命令,DOS 便显示当前验证功能的状态。键入 VERIFY OFF 命令就关闭验证功能。

要弄清楚系统的验证功能的当前状态是 ON 还是 OFF(即开、关状态):

1. 键入 VERIFY 命令。
   DOS 便回答是 VERIFY ON 还是 VERIFY OFF。
2. 如果打算改变验证功能的状态,便键入 VERIFY ON 或 VERIFY OFF 命令。

一般性验证打开后将降低整个系统的速度,但并不十分明显。但用户在大多数情况下只对一些特殊情况执行验证操作。

46.6.2.2 特殊验证功能

在 COPY、XCOPY 及 DISKCOPY 命令后面使用/V 选项,便要求 DOS 只对该命令的操作进行验证。该选项不理会 VERIFY 命令的设置状态,不过在 VERIFY 是 ON 状态时便显得多余了。在 VERIFY 为 ON 时,虽然要花费更多的时间,但在有些场合下是值得的,因为它确保写数据不出错。

46.7 自 测 练 习

下面给出了一些题目,以练习本章所讲述的技巧:

1. 如果手中有几个 CONFIG. xxx 版本(例如 CONFIG. SYS 和 CONFIG. SAV),试比较它们,看看能否指出它们之间的差别。再从 AUTOEXEC. xxx 为例练习一遍。
2. 在 AUTOEXEC. BAT 中写入相应的命令,改变计算机中 VERIFY 的原有设置状态。

第 47 章　高级补救技术

简介： 不管有多小心，也不管采取了多少预防措施，有时候都会丢失一些数据。这些失误有可能是由硬件故障造成，也有可能是由病毒、或者仅仅因为输错命令等因素造成的，但不管如何，用户都不得不尽最大努力去加以恢复。

47.1　DOS 的恢复工具

DOS 6 有一系列诊断和恢复工具，能够帮助用户处理系统区和目录树中出现的错误，也能恢复意外删除的文件以及意外格式化的磁盘。DOS 的工具能满足用户大多数情况下的需要，但有时也需要用别的方法解决问题。

本书已经讲述了一些最通用的工具：

- 用 SCANDISK 和 CHKDSK 恢复丢失的簇以及交叉链接的文件
- 用 SCANDISK 从损坏的扇区中挽救数据
- 用 UNFORMAT 从被格式化的磁盘中恢复数据
- 用 UNDELETE 恢复被删除的文件
- 用 BACKUP 从备份中恢复文件

本章将阐述 CHKDSK、SCANDSK 及 UNFORMAT 的一些功能，这些功能也许很少用到。同时，本章还会讲述怎样使用 MSD 获得计算机的有用诊断信息。本章的具体内容有：

- 如何处理 SCANDISK 或 CHKDISK 报告的下述错误：分配错误(Allocation error)，无效目录项(Invalid subdirectory entry)，以及文件分配表中出现的坏扇区(Bad sectors in the FAT)。
- 当镜象文件丢失后恢复一个格式化了的磁盘。
- 使用 MSD。

47.2　用 SCANDISK 和 CHKDSK 处理结构错误

SCANDISK 和 CHKDISK 报告的错误，在大多数情况下都属于丢失分配单元及文件发生交叉链接。在第 27 章“CHKDSK 和 SCANDISK”中讲了如何处理这类问题，本章将告诉用户如何处理不常遇到的一类问题。

47.2.1　无效分配单元

在 FAT(文件分配)表中，代表无效分配单元的数值是 0 和 1，以及大于磁盘总簇数的数值。对于没有被使用的簇来说 0 也是有效值，但它们不能出现在一个文件链中。

图 47.1 示出了一个 FAT 的例子。其中，从第 46 簇开始的文件链为无效文件分配链；第

49 簇的值是 1，此处分配链便断了。在这种情况下，SCANDSK 或 CHKDISK 从目录中的文件大小一项中得知文件还有其他的簇，但它却找不到那些簇。（也许我们自己能从 FAT 表猜出来第 49 簇的值应为 50，但 SCANDISK 或 CHKDSK 不作这样的猜测。）

0-7：			3	4	4	<EOF>	<EOF>	<EOF>
8-15：	0	10	11	12	13	14	15	16
16-23：	17	18	19	<EOF>	21	22	23	24
24-31：	25	26	18	28	29	30	31	32
2-39：	33	34	35	36	37	38	39	.40
40-47：	<EOF>	0	0	0	0	<EOF>	47	48
48-50：	49	1	51					

图 47.1 FAT 表的一个例子

其他属于该文件的簇就被隔断了联系，如同孤立了一样，但用户可以从磁盘上丢失的簇中把它们找回来。

图 47.2 是 SCANDISK 报告无效分配单元的一个例子，它指出在文件的开头是正确的，但在文件分配链的其他地方出了问题。

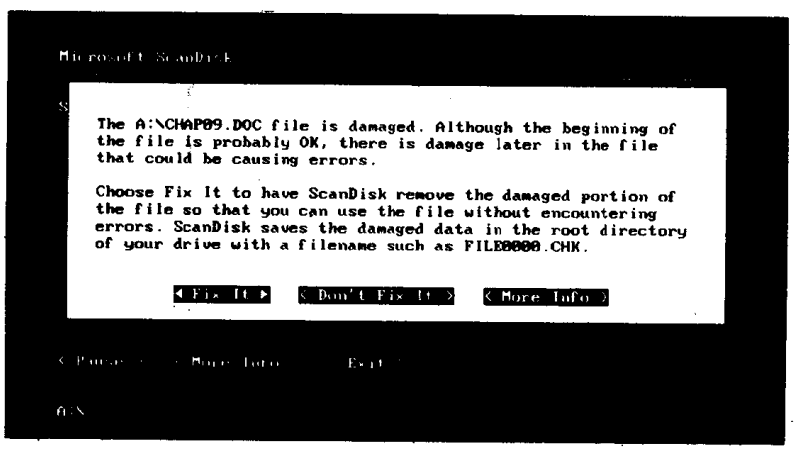

图 47.2 SCANDISK 对无效分配单元的报告实例

CHKDSK 对无效分配单元的报告如下：

```
C:\BOOK1\CHAPTER.CPY
        First allocation unit is invalid,entry truncated

C:\BOOK1\STUDY.STY
        Has invalid allocation unit,file truncated

19 lost allocation units found in 4 chains
        38912 bytes would be freed
```

如果第一个分配单元无效，那么文件的任何一部分都无法直接挽救。丢失分配单元的链中至少有一个属于这个文件。在第二个例子中，无效文件分配单元不在文件的第一个簇上，因此可以直接找回一部分。除非无效部分在文件末尾，否则至少有一个丢失的链中包含有文

件的数据。

47.2.2　SCANDISK 和 CHKDSK 怎样修复无效分配单元

对于出现丢失分配单元问题的最好解决办法是记下文件名，完成扫描后，从备份中恢复该文件。只有在手中没有备份时才让 SCANDISK 或 CHKDSK 修复丢失的链。在处理该问题上，SCANDISK 和 CHKDSK 都不能做得十分圆满。

要让 SCANDISK 修正错误，在对话框中选择 Fix it 一项。对于 CHKDSK，重新运行带有 /F 参数的 CHKDSK 命令。SCANDISK 将自动恢复丢失的分配单元中的数据，但 CHKDSK 是把所有丢失的分配单元中的数据组成几个文件，其中一个. CHK 文件中可能包含原文件中丢失的数据。

当试图修复有丢失分配单元的文件时，SCANDISK 或 CHKDSK 用 EOF 代替无效项，并调整目录中的文件大小，以适合新的簇数。现在便可以对文件进行访问并查看丢失了多少数据，然后检查. CHK 文件，看看哪个包含了自己所需要的数据。但不要运行用这种方法所恢复的程序文件。

注释： 使用 SCANDISK 时，无效分配单元也会导致出现分配表出错（下面会讨论），因为目录项中的文件大小不符合实际分配单元中所包含的簇数目。如果使用 SCANDISK 修复无效分配单元，那么同时也让它修复分配表中的错误。

用户应该使用建立该文件的应用程序检查所修复的文件。但是，很多应用程序在建立文件时都在文件的开头建立文件头，以及在文件的最后建立文件尾，当 SCANDISK 或 CHKDSK 修复文件时，它会砍掉文件尾信息，因此应用程序有可能不能打开修复好的文件。假如用户碰到了这样的问题，可以先打开一个新文件，然后把这个受损的文件插入其中。如果这样仍然不行，便可试试用文本编辑器如 EDIT 打开文件，删除其中的非 ASCII 码文本，然后把该文件以 ASCII 文本文件方式存盘。而后，再用原应用程序打开一个新文件，再把编辑过的文件插入其中。用户亦可把. CHK 文件插入到相同的文件中（可能首先也需要把它们转换为 ASCII 码文本文件）。

47.2.3　分配错误

当 SCANDISK 或 CHKDSK 探测到目录中所记录的文件大小和文件的分配单元中簇数不相符时，它们便报告分配出错。CHKDSK 显示的信息如下：

```
A:\NEWSTUDY.DOC
    Allocation error,size adjusted
```

如果不使用/F 选项，则目录项中所记录的文件大小值就不会被调整，即使给出的信息中说明已调整过了。

图 47.3 显示了是当 SCANDISK 在遇到分配出错时所显示的信息。它告诉用户有关文件大小的信息有错。

47.2.3.1　CHKDSK 怎样修复文件分配错误

用户最好从备份上把文件恢复，而不是去修复出错的部分。假如手中没有备份，才去使用 SCANDISK 对话框中的 Fix It 功能以及重新运行带有/F 选项的 CHKDSK 命令。CHKD-

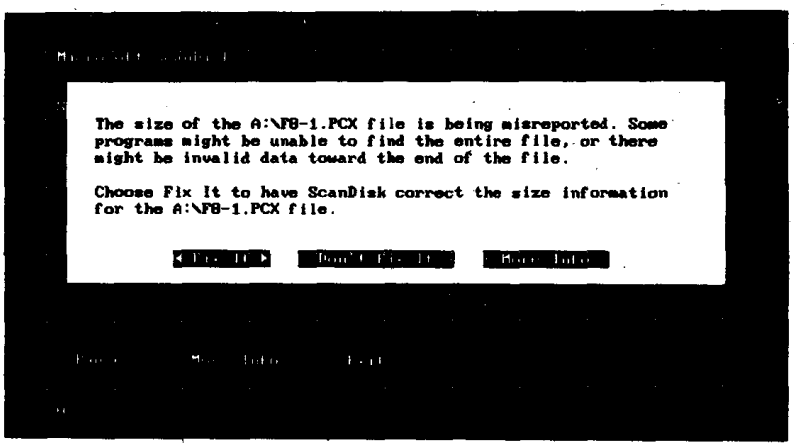

图 47.3 分配出错时 SCANDISK 的报告

SK 把所有丢失的链作为文件存储，因为在修复时可能砍掉文件的尾部。

SCANDISK 或 CHKDSK 通过调整相应目录项中的文件大小数值来修复分配错误。整个最后的一簇被视为文件的一部分，因为 SCANDISK 和 CHKDSK 都不能区别哪儿是文件的末尾，哪儿是无关紧要的部分。

如果记录的文件大小出错而 FAT 中分配单元数目是正确的，CHKDSK 在修复时可能把无关紧要的东西加到文件末尾。如果所记录的文件大小是正确的而 FAT 中所记录的文件的分配单元数少了或者太多了，则修复的文件有可能偏小或偏大，但两种情况下文件中都包含有无意义的部分，或者丢失了一些分配单元。但用户现在至少可以访问文件并了解下一步该做些什么。

47.2.3.2 处理文件分配错误

修复操作完成后，再在原建立文件的应用程序下检查该文件。（用户可能需要首先把它转换为 ASCII 码文件，并把它插入一个新的文件中）。用户可能要自己删除文件尾部的无意义部分，或者定位和插入从. CHK 中找回的丢失了的数据，这类. CHK 当然是由 SCANDISK 或 CHKDSK 所建立。

警告： 如果 SCANDISK 或 CHKDSK 发现一个程序文件中出现文件分配错误，请删除该文件并从其原始安装盘上进行重装。千万不要修复程序文件中的错误，或者运行由 CHKDSK 或 SCANDSK 所修复的程序文件。

47.2.4 无效子目录项

子目录是作为文件存储的，但有特定的目录属性。它们不能被损坏。如果. . 目录项显示出父目录的错误簇号，例如，DOS 找不到子目录和父目录之前的联系，如果子、父目录之间的联系被破坏，DOS 便不能定位子目录。

SCANDISK 可以识别几种子目录项受到损坏时所带来的相关问题，例如，除了能找出无效目录项以外，它可以找到独立的子目录项。图 47.4 显示的便是 SCANDISK 关于无效子目录的一例典型的信息显示。

当 CHKDSK 发现无效子目录项时，所显示的信息决定于问题的严重程度。下面即是一

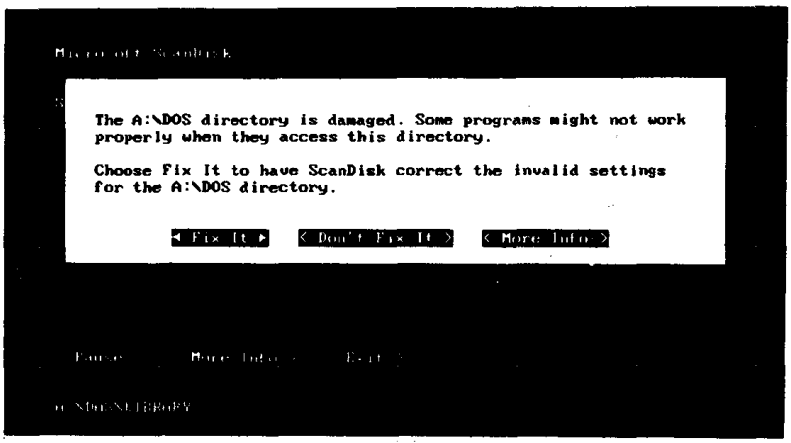

图 47.4　SCANDISK 关于无效子目录的报告

例：

```
C:\BOOK1\CHAP03.DOC
    Invalid subdirectory entry
```

在这种情形下，CHKDSK 能够找到问题之所在。例如，可能是和父目录的联系中断了，此时，CHKDSK 搜索目录树，找到父目录的位置，并将联系续上。此时，不会有文件或分配单元丢失。

在发现无效子目录项时，也可能显示如下的信息：

```
324 lost allocation units in 5 chains
    165888 bytes disk space would be freed
  Convert lost directory to file (Y/N)?
```

最后一行表明用户不能访问子目录中的任何东西，其中的所有文件都被视为丢失链的数据。同样，该子目录中的子目录及其文件也被视为失去链的数据，此时 CHKDSK 不能对子目录作任何修复工作。

47.2.4.1　SCANDISK 及 CHKDSK 怎样修复无效子目录项

如果 SCANDISK 发现磁盘上有包括目录项的丢失的链时，它便给目录分配一个通用名字(为第一个目录指定的名字是 DIR00001)并为它在根目录下建立目录项。它将修复所有它能修复的无效子目录，并且删除那些它不能修复的子目录。

CHKDSK 在修复无效目录时与 SCANDISK 有些不同。如果它不向用户询问是否将子目录转换成一个文件，说明带有/F 选项的 CHKDSK 能够找到它所需要的信息。它会将出现的问题修复并不会丢失数据。

但是，如果 CHKDSK 向用户询问是否将子目录转换为一个文件，而且用户按 Y 表示同意，则 DOS 把以前包含子目录的簇写成一个文件。由于它们不再和整个目录结构有联系，因此，子目录中所包含的所有文件的最开始的分配单元将丢失。在 CHKDSK 向用户询问时，用户一定要按 Y，以便 CHKDSK 把丢失的分配单元转变成一个文件。

47.2.4.2　处理无效子目录项

因为 SCANDISK 在恢复子目录时仍把它们看作子目录，所以用户不必修改或补救其中

的文件，但必须更正它的子目录名。

如果 CHKDSK 向用户询问是否将无效子目录变成一个文件时，为了安全起见，一定得回答 yes，即使所得到的文件没什么用处。但是，如果用户不能从备份中恢复文件时，把丢失的文件簇转变成文件就相当重要了。

在运行 CHKDSK 之后如果要对目录进行恢复，则先修改所有的子目录并检查.CHK 文件，把它们拷贝到先前的子目录下并更正它们的名字。最后，删除所有不再需要的.CHK 文件。

47.2.5 磁盘控制区中的坏扇区

SCANDISK 的表面扫描功能能够从损坏的扇区中恢复数据，但是，由于磁盘的控制信息，例如 FAT 及启动记录，必须位于磁盘上的特殊位置上，SCANDISK 不能采用把其中数据移到另外地方的方式对它们进行修复。图 47.5 示出的是当 SCANDISK 遇到这类问题时所显示的信息。用户所能做的只能是选择 OK，然后程序中止。

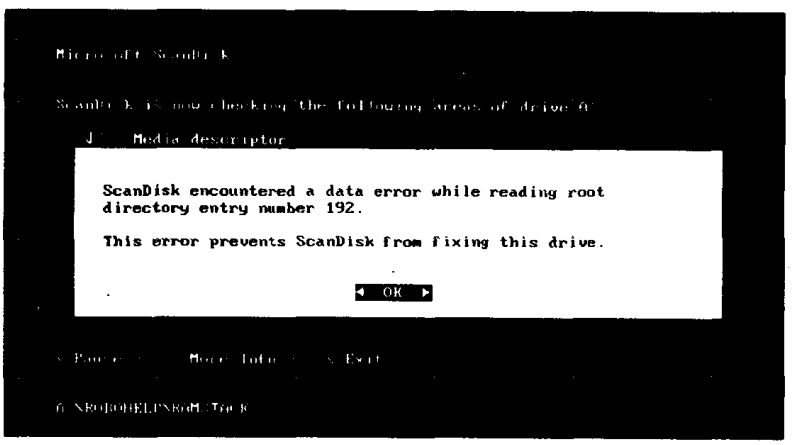

图 47.5 根目录区有坏扇区时 SCANDISK 的报告

CHKDSK 不能检查坏扇区或其他磁盘表面出现的问题，但是，如果磁盘控制区出现问题，很可能会出现下列信息：

```
Bad sector in FAT
```

或者

```
Probable nonDOS disk
```

Probable nonDOS disk 信息表明可能是由于启动记录扇区中出现个别损坏。上面两则信息都表明所出现的问题太大，CHKDSK 不能处理。

如果是软盘出现问题，而且用户手中有备份，便可以把坏盘扔掉。但是，如果问题出在硬盘上，用户也许希望能加以补救。一些别的应用软件如 Norton Disk Doctor 也许能做到这点，但一般情况下 CHKDSK 不能。即使不能修复硬盘，在撤换硬盘之前也许能够恢复其上的数据。

47.3　在没有映象信息的情况下重建磁盘

几乎所有的用户在格式化软盘时都出现过意外，错误地格式化了不该格式化的软盘。在第 17 章中阐述了如何使用 UNFORMAT 程序恢复意外地格式化了的磁盘上的数据。在使用 UNFORMAT 时，到底能够恢复多少数据取决于以下几个因素：

- 是否是无条件格式化
- 是否磁盘中含有恢复先前格式化的信息
- 在格式化后，是否又向磁盘上添加了数据
- 如果没有记录格式化信息，磁盘上是否含有分段存放的文件

在下述情况下，恢复先前格式化的没有映像信息的磁盘是必要的：

- 用户格式化的磁盘是由第三方厂商提供的格式化程序所格式化的，该程序没有存储其映象信息。
- 在使用 FORMAT 命令时，同时使用了/U 和/Q 选项。
- 由于磁盘太满，格式化程序不能存储其映像信息。
- 映象信息被损坏会丢失。

图 47.6 示出了 UNFORMAT 命令的选项，这些选项用于重新建立磁盘。

UNFORMAT

恢复用 FORMAT 命令格式化的磁盘，恢复删除了的目录，显示分区表。

句法：

UNFORMAT drive [/L] [/TEST] [/P] [/PARTN] [/J]

参数和选项：

drive	指明要恢复的驱动器。
/J	核实映像文件。
/L	显示由 UNFORMAT 命令找到的每一个文件和子目录。如果不指定该选项，只列出部分文件和子目录。
/TEST	仅显示如何在磁盘上重建磁盘信息。而不真正格式化磁盘。
/P	指定把信息从打印机输出。
/PARTN	当和/L 一起使用时，显示分区表。

注释：

当恢复硬盘时，每个扇区的字节数必须是 512，1024 或 2048。

当使用/L，/TEST 或/P 时，UNFORMAT 跳过映象信息，并搜索(很慢)子目录所在的簇。

图 47.6　UNFORMAT 命令总结

UNFORMAT 所做的第一件事情是磁盘上搜索映象信息。映象文件并不是有目录项的常规文件，FORMAT 把它放到磁盘上最后面的可用簇里面。UNFORMAT 仅仅是从最后一簇开始读取数据，从中搜索映象文件。如果在通常存放映象文件的地方找不到映象信息，便显示下面的信息：

```
Unable to find the Mirror control file. If you want to
search for the Mirror image file through the entire
hard drive,press Y,or press N to cancel the UNFORMAT command.
```

如果用户允许的话，也可以在磁盘（并不一定是硬盘）的其他部分搜索，即使是在软盘上搜索，也会要花费很长的时间。假如用户认为映象信息一定在磁盘上，那么按 Y 键，让 UNFORMAT 继续搜索，否则按 N 键，中止 FORMAT。然后，就可以重新启动 UNFORMAT，并在命令中使用让它恢复磁盘的选项。

提示： 当使用/L 或/P 选项时，UNFORMAT 可能能够恢复被删除的目录，即使磁盘没有被格式化。但这种重建目录的方法很慢，而且不一定成功。有时候它会建立一个陌生的目录，而且会损坏分段存储的文件，即使这个文件属于已存在的目录下。

恢复被删除目录的最好办法是用 UNDELETE 命令。不幸的是，DOS 中的 UNDELETE 不能恢复被删除的目录，但 Windows 中的相关命令或许可以做到。而 Norton Utilities 的 UnErase 程序也可以恢复被删除的目录。

47.3.1 UNFORMAT 命令怎样重建磁盘

当 UNFORMAT 没有使用映象信息时，它在磁盘上搜索子目录，并用子目录和父目录之间的联系确定父目录之所在。用这种方法，它可以重建整个目录树。当子目录被恢复后，其下的文件也被尽可能地恢复。但是，对于分段存储的文件，UNFORMAT 只能找到它的第一段。

当映象文件不可用时，所出现的最大问题在于根目录下的文件。由于在建立新的根目录时，它们的目录项被清除，因此它们不能被恢复。第一级子目录将在数据区找到，而 UNFORMAT 能够识别它们属于根目录，因为它们和 0 簇相联系；但是却没有办法找回它们的名字。UNDELETE 使用 SUBDIR.n 形式的通用名把它们恢复在根目录下，用户可以检查这些目录并对它们重新命名。但是不能从根目录下恢复文件，除非使用第三方厂商的工具软件，这些软件也许使用户能够检查可用的簇，并根据它们拼凑文件。

当没有使用映象文件时，UNFORMAT 在屏幕上列出它所发现的子目录的完整路径。它显示出每一个分段存储的文件，并且向用户询问该文件是否被截断或删除（如果命令中使用了/L 选项，它列出所有文件，包括分段存储的文件或其他形式的文件）。

47.3.2 启动重建过程

- 键入 UNFORMAT d: /TEST，便启动重建过程。/TEST 选项使 UNFORMAT 模拟重建过程，它将模拟所有的步骤并显示所有的信息，但并不真正改写磁盘。这些信息提醒用户决定是否真正值得重建磁盘。如果用户决定重建，便键入 UNFORMAT d:命令，于是便可

看到下列信息：

```
CAUTION ! This attempts to recover all the files lost after a
format,assuming you've not been using the Mirror command.
This method can't guarantee complete recovery of your files.
The search-phase is safe;nothing is altered on the disk.
You will be prompted again before changes are written to the disk.
Using drive A:
Are you sure you want to do this?
If so,press Y;anything else cancels.
?
```

此时，用户还可以改变主意，并且不会发生意外。如果键入 Y，UNFORMAT 便开始搜索与子目录有关的可用簇，在屏幕上显示搜索过的簇所占的百分数以及已经找到的子目录的数目。当搜索完毕后，用户可以看到在根目录区找到的文件数目和子目录数目。这些子目录中可能有以前删除了的没有用的目录，随后用户可以重新把它们删除。所显示的信息如下：

```
Searching disk...
n% searched. n subdirectories found.
Files found in the root: n
Subdirectories found in the root: n
```

随后，UNFORMAT 检查整个目录树，以寻找文件。所有被删除的第一级子目录将失去它们的名字。由 UNFORMAT 建立的名字如下：

```
Walking the directory tree to locate all files...
Path=A:\
Path=A:\SUBDIR.1\
Path=A:\SUBDIR.1\CHAPS\
Path=A:\SUBDIR.1\
etc.
Files found:n
Warning! The next step writes changes to disk.
Are you sure you want to do this?
If so,press Y;anything else cancels.
?
```

这是用户想中止的最后一次机会。如果选择继续，UNFORMAT 开始恢复子目录及文件。如果文件只有第一个存储段可用，就会显示下列信息：

```
Checking for file fragmentation....
Path=A:\
Path=A:\SUBDIR.1\
Path=A:\SUBDIR.1\CHAPS\
GETTY2.DOC        2560  9-25-94   11:32am  Only     512 bytes
are recoverable. Truncate or Delete this file?
```

用户可以决定是否让 UNFORMAT 存储它截断后的文件，或者将其删除。

47.4　使用 MSD

MSD 程序(Microsoft Diagnostics，Microsoft 诊断程序)显示一些与系统有关的重要设置

的值。图 47.7 示出了 MSD 主菜单的屏幕显示。它是在 DOS 命令提示符下键入 MSD 命令后的显示屏幕。

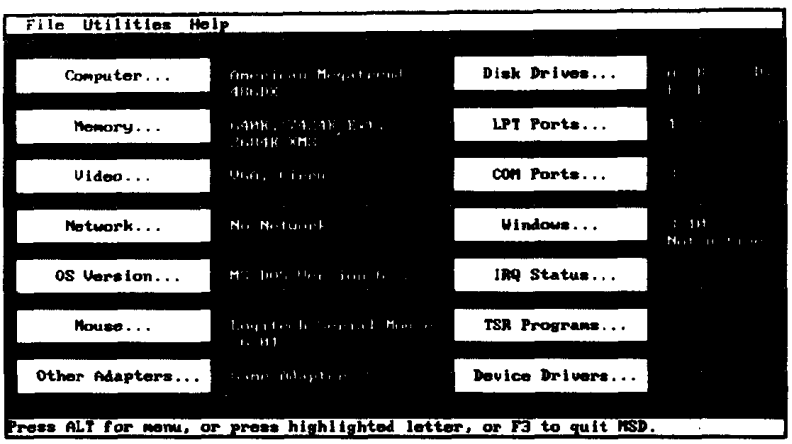

图 47.7 MSD 的主屏幕显示

注释：用户也可以在 Windows 下启动 MSD，但所显示的信息可能不相同。MSD 发现 Windows 存在后，便向用户询问这是否正常。

用户可以通过选择菜单中的某个按钮而查看有关该项的详细信息，例如，如果选择 Video 按钮，则屏幕上出现如图 47.8 所示的对话框。用户不难看到，所列的信息非常详细。虽然使用 MSD 了解系统结构显得有趣，但一般情况下不需要这种级别的信息。但是，如果 DOS 或 Windows 出现了系统级的问题，MSD 所提供的信息就至关重要。

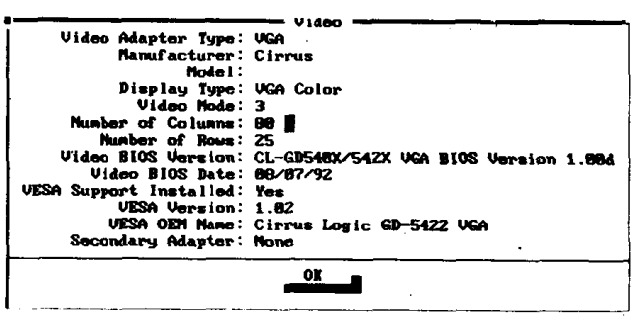

图 47.8 MSD 的 Video 对话框

MSD 特别详细的提供系统内存使用的有关信息。查看内存状况信息的步骤如下：

1. 退出所有程序，回到 DOS 的命令提示符下。
2. 键入 MSD 命令。

   MSD 显示主菜单。
3. 选择 Memory 按钮。

   MSD 显示高端内存映象。屏幕顶部的图注说明内存映象中的各符号代表的意义。滚动屏幕便可以看到所有高端内存。
4. 选择 OK，关闭内存映象。

5. 要更详细地察看内存，选择 Utilities，Memory Block Display 按钮。

   此时对话框左边是当前已装入的程序的清单，右边是内存映象。

6. 把光标在左边上移或下移，观看右边内存映象所显示的结果。

   当在程序清单内移动光标时，右边|--|标志跟着移动，停留的地方正是光标所指的程序在内存中的位置。用户可在该映象显示中看到所有的常规内存和高端内存映象。

7. 选择 Close，关闭 Memory Block Display。

注释： Utilities，Memory Block Display 显示内存(RAM)中的内容，而 Utilities，Memory Browser 显示 ROM 中的内容。

MSD 的 File 菜单使用户能够查看(但不能编辑)系统的启动文件及. INI 文件，例如 AUTOEXEC. BAT 及 SYSTEM. INI 文件(如图 47. 9 所示)。用户亦可把 MSD 所显示的报告内容在打印机上打印出来。用户选择 File，Print Reprot 项后所出现的对话框见图 47. 10。

所有被符号 X 所标记的项目将出现在报告中。用户使用 Tab 键从一列跳到另一列，并使用空格键进行选择。注意，用户也可以选择打印机端口，或者把报告作为一个文件存盘。当选定建立报告的方式后，便选择 OK 打印。

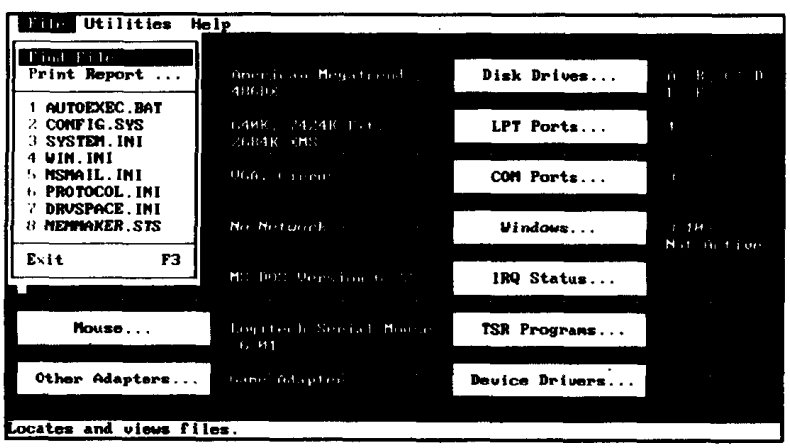

图 47. 9　MSD 的 File 菜单

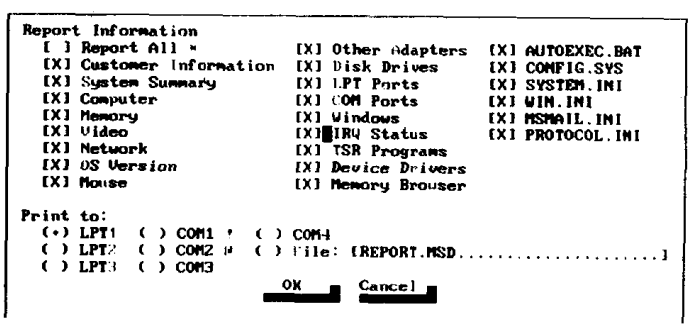

图 47. 10　MSD 的 Print Report 对话框

47.5 自 测 练 习

下面的建议能帮助读者练习本章所讲到的技术：

1. 检查硬盘驱动器，纠正所发现的错误。
2. 准备一张软盘，上面要求有几个目录和文件。(确保软盘上的数据确实没有用处，不要意外地损坏自己需用的数据)。格式化该软盘但不存储映象文件，然后再将其恢复。
3. 准备另一张软盘，上面也要求有目录和文件(并且也要求保证其上的数据没有用处)。删除其中一个目录，然后试试用 UNFORMAT 去恢复。如果计算机中装有 Windows，可练习用 Windows 的 UNDELETE 命令删除和恢复目录。
4. 用 MSD 查看自己的系统。例如，查出 COM 端口的地址、硬盘有多少柱面、以及计算机中是否有 DMA 控制器等。

第 48 章　控制台的控制

简介： 在一般情况下，DOS 命令行方式下的屏幕就像过云使用的黑板一样：平淡的背景，文本显示，每屏 24 行，没有图形；通过键盘键入的字符也只是键盘上所有的那些或稍微多一点。但是，可以使用 DOS 的一些特性让屏幕和键盘变得生动起来。

48.1　包装 DOS

通过一些 DOS 命令可以控制显示器和键盘。可以给显示屏加上颜色(一定程度上)和图形，向用户闪烁显示一些信息，可以在显示屏的顶端显示一个色彩丰富的标题，其中包含日期、时间和其他重要信息。可以重新定义键(在键盘上)打出一些国际上的字符比如 é 之类，也可以打出图形字符如☺，甚至可以宏定义键盘的键为 DIR *.* /S /B /P /A 或者 CD \NORTH\ALASKA。本章的一些技巧可以使用户的工作以及用户所建的批处理文件更有意义，效率更高。

在这一章里，将介绍以下 DOS 功能：

- 给命令行显示方式下的屏幕加上颜色
- 在命令行显示方式下控制光标的位置
- 在命令行显示方式下显示简单图形(比如一些方框)
- 设置字符显示的卷屏换行开关
- 在键盘上重新定义键
- 使用不同的控制设备
- 通过 DOS 的能源管理特性来保护便携式 PC 机的电池
- 调整字符在屏幕上的位置和大小
- 调整键盘的打键速度

48.2　DOS 和视频监视器

DOS 支持几种不同的显示卡：单色显示卡(MDA)，彩色图形卡(CGA)，扩展图形显示卡(EGA)和视频图形阵列显示卡(VGA)，其他种类的视频显示卡通常是与上述的一种或几种相兼容。如果一个显示卡或监视器拥有比上述基本种类更多的一些特性(如更多的象素或为桌面出版系统设置的双页显示)，用户可以通过其附带的软件来帮助得到这些特性。

48.2.1　CON 驱动程序

DOS 的命令行界面使用内设的 CON 设备驱动器和计算机所安装的显示卡打交道。

CON 提供简单的面向文本的服务，因此 DOS 可以在命令行方式下在屏幕上写一些信息。CON 使用视频控制器内设的一些功能来驱动硬件，这样就使 DOS 不必依赖真正的显示设备，同时让操作系统不必关心到底系统使用什么样的显示卡。不幸的是，这样造成的结果是 DOS 用户非常厌恶的呆板得像一潭死水一样单调的单色屏幕。

48.2.2 视频缓冲区

每一个显示卡都有一个存储区域，一般称作视频缓冲区，所要显示的内容首先存储在视频缓冲区内。视频被称作存储映象是因为在缓冲区里的每一个字节都对应着显示屏幕上一个特定的位置。

在文本显示的模式里，视频缓冲区对于屏幕中的每一个字符位置都相应包含两个字节。第一个字节含有所要显示字符的 ASCII 码，第二个字节含有这个字符的显示属性(颜色、闪烁等等)。图 48.1 表示一个视频缓冲区的开头部分，在这里，所要显示的内容是“Sanity!”，从屏幕的第一个字符位置开始。第一个字节表示字母“S”的 ASCII 码，接下来的字节里含有它的显示属性；07 表示 DOS 的一种常用显示方式：黑色背景下的白色显示。第三个字节表示字母“a”的 ASCII 码，第四个字节含有它的显示属性。

s		a		n		i		t		y		!	
83	07	97	07	110	07	105	07	117	07	121	07	33	07

图 48.1 文本模式的视频缓冲区

彩色显示方式下，属性字节里包含有前景色和背景色，还有前景显示的亮度和闪烁或稳定的状态。单色显示方式下，属性包括亮度、黑体、闪烁和正常显示或反相显示。上述是在文本模式下的情况。在图形显示模式下如何控制它的属性是完全不一样的；系统所要处理的是比文本模式要小得多的屏幕单元。

48.2.3 ANSI.SYS 所提供的控制

CON 驱动程序不会使用颜色，它只是把字符的 ASCII 码放入视频缓冲区的偶数地址(对应字符)而不会影响到奇数地址(对应字符的属性)。因此当前的显示属性不会被 CON 驱动程序所修改。

ANSI.SYS 设备驱动程序是一种超级的 CON 驱动程序，它能提供更多的选项，用来发挥显示器和键盘的特性。ANSI.SYS 提供一种指定屏幕显示颜色和其他显示属性的方法。同时它能控制光标的位置和清理屏幕。它可以让每屏显示 43 行或 50 行(如果监视器许可的话)，也可以让每屏显示 40 列而不是 80 列。它的另一个特性是能够操纵键盘上键的定义。

一些依赖于 ANSI.SYS 的程序能够支持扩展的视频显示和键盘设备，它们所提供的说明文件里要求用户系统必须装有 ANSI.SYS。还有其他一些程序使用 PC 机的视频 BIOS，其中包含有更有效的屏幕显示功能。即使一个不会编程的用户也可以利用 ANSI.SYS 来为他的屏幕显示，命令行提示符，以及批处理文件增加一些趣味。所有这些方法只能影响到那些直接在命令行方式下键入的命令，如：DIR、MORE 以及 CHKDSK 等。它不可能影响那些能自己提供显示方式的一些应用程序，如：EDIT、DOS Shell 以及 SCANDISK 等。

48.3　ANSI.SYS 设备驱动程序

用户可以在 CONFIG.SYS 文件里用“DEVICE”或“DEVICEHIGH”命令来安装 ANSI.SYS。图 48.2 给出了命令格式。ANSI.SYS 安装以后，它就监视所有的标准输出寻找发给自己的命令 。

ANSI.SYS

提供控制屏幕颜色、光标位置以及定义键的功能。
句法：

DEVICE[HIGH]=[path] ANSI.SYS [/X] [/K]

参数和选项：

path	指定 ANSI.SYS 的路径，缺省时指启动盘的根目录。
/X	对于 101 键盘(扩展的)而言，使 DOS 确认扩展的光标控制键、删除键、插入/覆盖键等除具有数字键同样的功能外有不同的扫描码。
/K	使 ANSI.SYS 把 101 键盘当作 84 键盘使用。

图 48.2　ANSI.SYS DEVICE 命令总结

如果想要给键盘上重复的键定义不同的字符，用/X 选项，比如对向上箭头键、Home 键、以及＋(加号)键等。如果没有使用/X 选项，重复的键就被定义为相同的字符。例如，两个＋(加号)键将具有相同的作用；相反如果用了/X 选项，就给这两个键定义不同的功能。关于键盘定义的细节将在本章的后文详细介绍。

现在装上 ANSI.SYS，这样就可以试一试本章的一些应用技术了：

1. 编辑 CONFIG.SYS，加入命令 DEVICEHIGH =C:\DOS\ANSI.SYS /X。
2. 重新启动计算机。

注释： 如果程序不能识别 101 键盘上的扩展键请用选项/K。如果 SWITCHES 命令中用了/K 选项，那么在 ANSI.SYS 中也要用/K 选项。

48.4　给 ANSI.SYS 发送命令

安装好 ANSI.SYS 后，它就驻留在内存里了，占用大概 48KB 的内存。它不停地监视所有的标准输出，寻找转义(escape)代码序列，这是它所能识别的命令。一个转义代码序列由一个 Esc 字符开始，接下来是一个左方括符，并且以一个特殊的终止字符作为结束。它们之间的代码组成了一个命令 ，关于这些特殊命令的细节以后还会讲到，这里先给出一个例子，它设置彩色的屏幕显示：红色背景下的青色字：

```
Esc[31;46m
```

代码 31 代表红色背景，46 代表青色前景。对于所有控制颜色的转义代码序列都必须使

用终止符 m;终止符的作用是告诉 ANSI. SYS 代码序列所要求的功能。要让颜色显示回到缺省状态,向标准输出发送如下指令:Esc [m(不带任何颜色的说明代码)。

注释: 终止符的使用是唯一的,如果用 M 代替 m 将不会对颜色显示有任何影响。

可以马上试一下这个命令,但首先必须学会怎样向标准输出发送一个转义代码序列。

48.4.1 键入转义代码序列

不能简单地在键盘上按下 Esc 键来开始一个转义代码序列;Esc 键的作用是取消当前的命令 ,并开始下一个命令 。如何输入 Esc 取决于到底采取何种方式向 ANSI. SYS 发送转义代码序列。产生标准输出的几个非常有用的命令是 PROMPT、TYPE 以及 ECHO。

48.4.2 使用 PROMPT 命令

在 PROMPT 命令里,可以键入普遍字符也可以键入特殊的字符,用以表示一些不可能键入的特殊字符。表 48.1 列出了 PROMPT 命令中所有特殊的字符,这些特殊字符能以大写键入,也能以小写键入。命令 PROMPT PG 的功能就是建立我们非常熟悉的提示符,即显示出当前的磁盘驱动器以及路径名,后面再跟一个大于号。

表 48.1 PROMPT 命令所用的特殊字符

$B	|(竖线或管道标记)
$D	当前日期
$E	Escape 字符
$G	>(大于号)
$H	Backspace(删除字符)
$L	<(小于号)
$N	当前磁盘驱动器
$P	当前驱动器和路径
$Q	=(等号)
$T	当前时间
$V	DOS 版本号
$$	$(美元符号)
$_	Enter 回车键

PROMPT 命令可以用来传送转义(escape)代码序列,用 $E 来表示 Esc 字符。如果有一台颜色监视器,试一试以下这些 PROMPT 命令:

1. 输入 PROMPT $E[36;41m$P$G。
   在红色背景下提示符以青色显示。
2. 输入 CLS。
   此时,整个屏幕都变成红色,并且屏幕左上角显示提示符(青色)。
3. 输入 PROMPT $E[36;41m$PGE[m。(记住:可以通过 DOSKEY 取回以前用过的命令并编辑它)。

在红色背景下设置青色的文本显示，显示标准提示符，然后关闭颜色设置以致于除提示符自身外均恢复为缺省的颜色。

4. 输入 CLS 使整个屏幕都成为背景色，此时，提示符被显示为青色并以红色为底色，但屏蔽的其余部分均是黑色。

当忙于通过 PROMPT 命令来控制颜色时，很容易就会忘记在其中包含 PG(或者其他想要的提示文本)。如果忘记了，结束时，会发现没有了提示符。不过，没关系，即使是看不见命令提示符，仍然可以键入命令。试一试下面的几个命令：

1. 输入 PROMPT $E[36;41m。
   提示符消失了。
2. 输入 CLS 使整个屏幕变成背景色。屏幕完全变成了红色。提示符照样没有出现。
3. 输入 PROMPT PG 以重新获得提示符。

用 PROMPT 向 ANSI.SYS 传送命令的一个缺陷就是它很可能失去以前拥有的提示符，除非它被非常小心地重新设置。不过，在通常情况下 PROMPT 都是传送转义代码序列的最快途径，并且可以立即看到它的作用。

48.4.3　使用文本文件

当你用 TYPE 显示一个文件时，DOS 将它送到标准输出中。任何转义代码序列都会被转送给 ANSI.SYS，这样，可以建立一个文件并在其中包含一个或多个转义代码序列，用 TYPE 显示这个文件就会使它们都发挥作用。

可以用任何文本编辑器来建立一个其中含有转义代码序列的文件，只要知道如何在其中输入 Esc 字符。为了能够正确地被 TYPE 输出，这些文件必须以 ASCII 形式存放。DOS 的 EDIT 编辑器就是以这种方式存放所有经过编辑的文件的。在 EDIT 编辑过程中，先按组合键 Ctrl+P，接着按 Esc 键，就在文件中键入了 Esc 字符。这时在屏幕上就会出现一个左箭头，TYPE(打印)这个文件时，它的作用就相当于一个 Esc 字符。

提示： 先按 Ctrl+P，再按 Esc 键，在 WordStar 中同样有效。在 Microsoft Word for DOS 里，输入 Esc 字符的过程如下：按着 Alt 键，接着在数字键盘上按 2，再按 7(此时，要求 NumLock 处于打开状态)，当松开 Alt 键时，屏幕上就会出现一个左箭头。在 WordPerfect for DOS 里，按 Ctrl+V 接着按 Esc 键，在屏幕上将出现一个 ^[。

每个 ANSI.SYS 的转义代码序列的第二个字符必须是左方括符：[，紧接着左方括符的是代码的命令本身，最后是终止符。请试建立一个名为 COLORTST 的文件，用来设定屏幕的颜色：

1. 执行 EDIT，编辑一个名为 COLORTST 的新文件。
2. 在文件中键入以下这一行字符，这里←代表 Esc 字符：

   ←[30;46m

3. 存储这个文件并退出编辑器。
4. 输入 TYPE COLORTST 命令。

下一个提示符则以青色背景黑色前景显示。

5. 输入 CLS 以扩展背景颜色至整个屏幕。最好清除红色，对吗？

48.4.4 使用批处理文件

在通常情况下，在批处理文件里包含一些 ANSI.SYS 的转义代码序列是很有用的。ECHO 命令向标准输出传送文本，由于它并不改变提示符，所以这种方法比 PROMPT 更方便（不过，你不能在批处理文件以外使用 ECHO 命令，因为它不能提供键入一个 Esc 字符的方法）。

下面的练习，以红色为背景青色为前景显示“Sorting Records”，接着恢复原来的以青色为背景以黑色为前景的显示方式：

1. 打开编辑器，建立一个新文件 COLOR.BAT。
2. 在文件里输入以下几行：

```
@ECHO OFF
ECHO ←[36;41mSorting Records←[30;46m
```

3. 存此文件并退出编辑器。
4. 键入 COLOR 执行这个批处理文件。

那条信息以红底青色显示之后，青底黑色的显示方式又被重新设定。

48.5 用于屏幕颜色控制的转义 ANSI.SYS 代码

ANSI 转义代码序列控制屏幕显示的不同属性。一类代码控制前景和背景的颜色，以及文本的显示属性。另一类代码控制光标的位置和清屏操作。还有一类代码控制图形显示的模式。

控制屏幕颜色的转义代码序列要求使用多种代码，表 48.2 给出所有这类属性和颜色的代码。每一个控制颜色的转义代码序列都要求同一种终止符：小写的 m 字符。

表 48.2 控制颜色显示的代码

	属性代码	
0	正常的文本状态	
1	高亮度的文本	
2	低亮度的文本	
4	加下划线(只用于单色显示)	
5	闪烁	
7	反相显示	
8	不可见的文本方式(前景和背景颜色一致)	
	颜色代码	
颜色	前景	背景
黑色	30	40
红色	31	41
绿色	32	42

（续表）

	颜色代码	
黄色	33	43
蓝色	34	44
紫色	35	45
青色	36	46
白色	37	47

一种颜色看起来如何将取决于所使用的监视器。例如，在许多显示屏上，黄色看起来像棕色。在每一个转义序列中可以包含用户所需要的任意多的代码（颜色或属性），只要用分号分隔它们就行了。

对于大多数系统来说，上表中的属性代码只用于前景显示。高亮度属性产生更亮的前景色彩，与正常的色彩有很大的区别；这样，实际上就有 16 种颜色可用于前景颜色的显示。另外，在某些系统中，属性代码 5 产生一个高亮度的背景而不是产生一个闪烁的前景字符。

在有些系统里，低亮度属性也产生不同的颜色（不同于正常状态），不过它在大多数系统中与高亮度属性产生的颜色相同。如果使用属性代码 8，前景颜色的代码仍被存储，但前景显示时实际上用的是背景的颜色。

提示： *在为其他用户准备批处理文件时，要尽可能检查一下他们所使用的监视器上面的颜色和属性代码。*

不同的监视器，可能有所差别。假如想要一个背景为紫色前景为闪烁的高亮度的黄色显示方式的话，把下面的命令存放在一个 ASCII 文件里，如果用 TYPE 输出到屏幕上，就会产生如上所说的结果：

```
←[1;5;33;45m
```

如果需要在一个批处理文件工作时在屏幕上显示出“Sorting Members...”，可以在这个批处理文件里输出下面这一行：

```
ECHO ←[5;33;45m Sorting Members...[m
```

48.6　屏幕和光标控制

在缺省时，一个命令结束后光标将出现在下一行的开始位置。不过，采用 ANSI 命令，用户可以把它定位在其他任何地方——可以在屏幕顶端用以显示日期的一个信息框里，或者在屏幕底端用来显示版权的信息框里。用于控制光标位置的转义代码序列要用几种不同终止符，表 48.3 说明了这一点。命令代码中可能需要包括行或列的信息用以控制光标。与通常的情况一样，终止符是唯一的。

表 48.3 屏幕和光标的控制代码序列

s	存储当前光标位置
u	恢复光标位置到所存储的状态
xxA	光标上移 xx 行
xxB	光标下移 xx 行
xxC	光标左移 xx 列
xxD	光标右移 xx 列
H	光标移到左上角(0,0)
xx;yyH	光标移到所指定的位置
xx,yyf	光标移到所指定的位置
2J	清屏,光标回到左上角
K	清屏,从光标当前位置移至最后一行
6n	显示光标的位置,←[xx;yyR

控制光标位置时,必须要小心,不要把光标位置放在已存在的文本的前面或上面以免使用户糊涂。必须仔细考虑光标位置的改变是否是受欢迎的。

48.6.1 创建一个标题

在 PROMPT 命令里利用光标位置设定的一些特性可以在屏幕的顶端创建一个标题。图 48.3 就是一个有简单标题的显示屏。用来建立这个标题的命令也显示在图中的屏幕上。提示符出现的每一次都将重新显示这个标题,因此可以说标题里的时间是始终准确的。

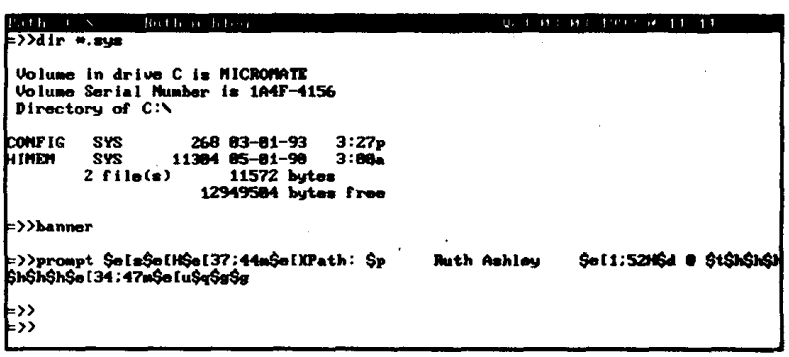

图 48.3 带简单标题的屏幕

这个 PROMPT 命令包含了迄今学到的所有 ANSI.SYS 转义代码序列,其中还有 PROMPT 的特殊字符和直接用于显示的文本。对此命令各个部分的解释如下:

$E[S	存储当前光标的位置
$E[H	把光标移至屏幕左上角
$E[37;44m	设定标题的显示为蓝底白色
$E[K	从光标位置起删除第一行的所有东西
Path:	显示指定的文本
$P	插入当前的驱动器和路径名
Ruth Ashley	显示用来提示的文本

$E[1;52H	把光标移至第一行第 52 列
$D	显示当前日期
@	显示指定的文本
$T	显示当前时期
HHHHHH	光标回退六格，删除秒和百分之秒的显示
$E[34;47m	设置屏幕的显示为白底蓝字
$E[u	恢复原来的光标位置
QG$G	显示新的提示符(=>>)

试一下这个标题的建立，不过，如果 DOSKEY 未被激活，那么先激活它；DOSKEY 可以大大简化反复键入 PROMPT 命令的工作。

注释： 在这个标题里，时间显示并非完全准确，因为它只是在命令提示符出现的时候才被刷新一次。

试建一个用户自己的标题，下面是具体步骤：

1. 键入 PROMPT $E[s$E[H$E[37;44m$E[KPath:　　$P　　your name $E[1;52H$D @ THHHHHHE[34;47m $E[u$QGG。
2. 键入 CLS。

48.6.2　显示一个信息框

用户可以用 ANSI.SYS 来建一个彩色的方框并在里面显示一条信息。这一定很有趣，它是一种吸引人们注意力的显示方法。图 48.4 显示了一个文件，其中包含用来产生一个内有黑色文本的红框的转义代码序列。这里为方便起见，分行显示了不同代码序列，当然也可以根据用户自己的爱好把它们连在一起。还可以根据自己的喜好来修改这些命令，以改变它的位置、颜色、大小和要求显示的文本内容。在批处理文件里用 TYPE filespec 来产生这个信息框。另外，也可以在批处理文件中把转义代码序列放入 ECHO 命令中去。

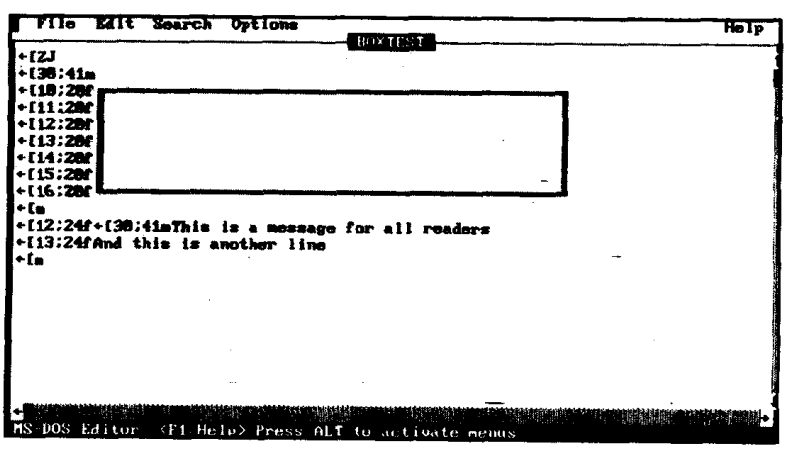

图 48.4　创建一个彩色的信息框

画方框时使用了 IBM 的扩展字符集(参看附录 B“返回码”)，这需要附带解释一下。在

视频监视器或打印机里，单个字节代表一个文本字符，其值从 0 到 255(0FFH)。0 到 127 是标准的 ASCII 码，代表字母、数字、标点符号等等。从 128 到 255 的数值并未被标准化，对于不同的硬件和不同的软件，它们可能有不同的含义。例如，DOS 系统按照 IBM 扩展字符集来解释这些数值，但 Windows 就不是这样。当用 ECHO 命令在一个批处理文件里显示文本时，可以期望看到 IBM 扩展字符集的作用。

键盘上并没有用来表示扩展字符的键，产生扩展字符的方法是按住 Alt 键同时通过数字小键盘敲入所要产生的字符的 ASCII 码。比如要产生一个双线框的左上角符号，可以敲入 Alt+201。这种方法不仅仅适用于扩展字符，对于任意一个字符都适用。

下面简单解释一下图 48.4 所示的文件中的一些转义代码序列的作用：

←[2J　　清屏并把光标置于左上角

←[30;41m　设置显示方式为红底黑字

新的颜色显示只能影响到文本或空格输入的地方，把光标定位在显示文本的每一行就可以把这种颜色的显示限制在方框里。屏幕的其余部分仍保持着以前的颜色。

←[10;20f　把光标置第 10 行第 20 列；这一行命令的其余部分画出方框的顶部

←[11;20f　把光标置于第 11 行第 20 列；这一行命令的其余部分画出方框的第二行紧接着的五个命令画出方框的两个边。

←[16;20H　把光标置于第 16 行第 20 列；这一行命令的其余部分画出方框的底部

←[m　　恢复缺省的颜色显示

←[12;24f←[30;41m　把光标定位在方框并设定方框内文本显示的颜色；这一行命令的其余部分在方框内显示信息

←[13;24f　重新定位光标并显示另外一条信息

←[m　　恢复缺省的颜色显示

在光标定位的代码序列中，终止符 f 的作用和 H 是一样的。如果在指令代码中没有行和列的值，终止符 f 将不产生作用，而 H 则缺省定为(0,0)位置。

依据下列步骤，用户可以创建一个自己的方框：

1. 用 EDIT 键建一个新文件 BOXTEST。
2. 参考前面的图 48.4，敲入文件 BOXTEST 的内容。

画一个方框所需要的一些代码如下所示：

Alt+201	方框的左上角
Alt+202	方框的左下角
Alt+187	方框的右上角
Alt+188	方框的右下角
Alt+205	水平直线段
Alt+186	竖直直线段

提示： 要画出多个水平直线段，没有必要每次都用 Alt+205。打出这个字符以后，可以按 Ctrl+Insert 键把它拷贝到剪贴板里，然后右移光标一格，再按 Shift+Insert 键就可以把它从剪贴板里拷到屏幕上了，并且可以多次按 Shift+Insert 键拷贝这个字符以满足要求。用同样的方法可以多次敲入 Alt+186 对应的字符。

3. 存储这个文件，退出 EDIT。

4. 键入 TYPE BOXTEST 显示这个方框。

48.6.3　字符回绕

在缺省情况下，当一个命令的长度达到 80 列时，超过的字符就回绕到下一行显示。在 DOS 早期的版本里，超过的字符堆积在第 80 列的位置而没有回绕功能，这样在屏幕上就只能看到最近敲入的字符。如果关掉字符回绕功能，则可以把一个长命令的最后几个字符隐藏起来。下面的命令可用于开或关字符回绕的功能：

←[=7h　　字符能回绕
←[=7l　　字符不能回绕

字符能否回绕不仅仅影响用户所敲入的命令，而且还将影响到如 FIND 和 TYPE 之类的命令，如果这些命令的输出中包含长句的话。

48.6.4　屏幕的显示模式(图形)

每个显示卡和屏幕都有缺省的模式。运行 BASIC 程序时，可以改变屏幕的显示模式来产生不同的效果。ANSI 的另一个转义代码序列的指令集也可以用来改变屏幕的显示模式。改变的结果将取决于所使用的视频监视器。表 48.4 列出了可以用来指定屏幕显示模式的一些数值，这类转义代码序列使用终止符 h，例如：

←[=modeh

如果想把屏幕显示改为单色文本模式，每行只有 40 个字符，则可以输出指令 PROMPT $E[=0h$P$G，这是用 PROMPT 命令来实现这个功能的一般方式。

表 48.4　屏幕显示模式

模式	含义
0	单色文本，40×25
1	彩色文本，40×25
2	单色文本，80×25
3	彩色文本，80×25
4	中分辨率图形显示(4 色)，300×200
5	与模式 4 一样，但禁止颜色脉冲
6	高分辨率图形显示(2 色)，640×200
13	彩色图形显示 320×200
14	彩色图形显示(16 色)，640×200
15	单色图形显示，640×350
16	彩色图形显示(16 色)，640×350
17	彩色图形显示(2 色)，640×480
18	彩色图形显示(16 色)，640×480
19	彩色图形显示(256 色)，320×200

如果使用了系统并不支持的显示模式，屏幕上就会出现奇怪的图案，甚至可能导致整个系统的瘫痪。此时，唯一的办法就是重新启动计算机。

48.7 ANSI 对于键盘的控制

ANSI 的转义代码序列可以用来控制键盘，例如，可以通过转义代码来调换键盘上两个键的定义。可以用一个键或者键的组合(如 Ctrl+D)来表示一个扩展字符，甚至可以表示一个字符串。表 48.5 列出了 ANSI.SYS 指令所能表示的一些键，圆括号里的指令并非对所有键盘都有效。可以在 ANSI.SYS 驱动程序的联机帮助中看到完整的键盘命令表。

给一个键盘新赋值也就是重映象这个键，如果给一个键赋值为一个字符中，就相当宏定义键盘。键盘的重新映象只能在 DOS 命令行方式起作用。

可以象其他 ANSI.SYS 的转义代码序列一样，通过一个 PROMPT 指令，ECHO 指令或者用 TYPE 命令输出一个文本文件来实现对键盘的重新映象。

表 48.5 键盘上的键的 ANSI 代码

键	正常	↑Shift	Ctrl	Alt
F1	0;59	0;84	0;94	0;104
F2	0;60	0;85	0;95	0;105
F3	0;61	0;86	0;96	0;106
F4	0;62	0;87	0;97	0;107
Home	0;71	55	0;119	
Up	0;72	56	(0;114)	
PageUp	0;73	57	0;132	
Left	0;75	52	0;115	
PrtSc			0;114	
Panse/Break			0;0	
BkSp	8	8	127	(0)
Enter	13		10	(0;28)
Tab	9	0;15	(0;148)	(0;165)
		(Grey beys)		
Home	(224;71)	(224;71)	(224;919)	(224;1051)
Up	(224;72)	(224;72)	(224;141)	(224;152)
PageUp	(224;73)	(224;73)	(224;132)	(224;153)
Left	(224;75)	(224;75)	(224;115)	(224;155)
		(Keypad Keys)		
Enter	13		10	(0;166)
/	47	47	(0;142)	(0;74)
*	42	(0;144)	(0;78)	
'	45	45	(0;149)	(0;164)
+	43	43	(0;150)	(0;55)
5	(0;76)	53	(0;143)	
		(Typeable Characters)		

（续表）

键	正常	↑Shift	Ctrl	Alt
a	97	65	1	0;30
b	98	66	2	0;48
z	122	90	26	0;44
1	49	33		0;120
0	48	41		0;129
'	45	95	31	0;130
=	61	43		0;131

48.7.1　重新映象一些的键

许多键的组合是没有赋值的，因此可以重新定义它们来表示一些扩展字符，也可以简单重新定义一个键，用户可以通过这些方法把一个 QWERTY 布局的键盘变成更有效的 Dvorak 布局的键盘。

按下列方式使用 ANSI 指令来重新映象键盘：

```
Esc[key;new-valuep
```

参数 Key 表示所要重新映象的键或组合键。可以用两种方式来引用这个参数：用当前所敲入的字符表示，所敲入的字符必须用双引号括起来；或者以键的 ANSI 代码（见表 48.5）来表示。参数 new-value 表示对这个键（由 key 所指定）重新映象的值，对这个参数也可以用引号里的字符来表示，也可以用从 0 到 255 的数字代码来表示。附录 B 给出了 ASCII 编码 128 到 255 的扩展字符。比如，如果想用 Ctrl+F1 来表示屏幕上打印出一个双线方框的左上角的话，可以用下面两个命令中的任何一个：

```
←[0;94;"╔"p
←[0;94;201p
```

在这两个指令里，0;94 代表组合键 Ctrl+F1。第一个指令直接用双引号括起来的方式来表示按 Alt+201 键的结果；第二个指令则使用了它的数字代码 201。

如果必须频繁的使用字母 A（Alt+142）则可以把它定义为 Alt+A：

1. 键入命令 PROMPT $E[0;30;142p$P$G。
2. 按 Alt+A。
   屏幕上就会出现字母 A。
3. 按 Esc，消除当前的命令行。
4. 回到根目录。
5. 键入 COPY AUTOEXEC.BAT HAGEN.XXX（用 Alt+A 输入 A）。
6. 在用户目录中键入 DIR 命令来查看这个新文件。
7. 键入 DEL HAGEN.XXX 删掉这个文件。

如果想要来点小玩笑的话，可以通过下述指令把“t”键和“h”相互调换了：

```
ECHO ← ["t";"h"p
ECHO ← ["h";"t"p
```

增强型键盘里有一些重复的键。如果想使这些键得到相同的处理，不必做什么特殊的工作。但是，如果想使它们得到不同的值，则必须在 DEVICE＝ANSI. SYS 命令里使用/X 选项。

安装 ANSI. SYS 时如果使用了/X 选项，就可以用数字小键盘来打印国际上的通用字母：

1. 键入命令 PROMPT $E[0;71;148p$P$G。
2. 关掉 NumLock 键，按下数字小键盘的 Home 键。
   屏幕上将出现字母 O。

只有在安装 ANSI. SYS 时使用了/X 选项，上述命令才有效。在增加型的 101 键盘上，有些键(如灰色的光标控制键等)属于扩展的键。这些键的 ANSI 代码在表 48. 5 的括号里。如果使用了/K 选项，或者所用键盘并非增强型，那么对这些扩展键的重新映象是不可能的。

48. 7. 2 为键指定字符串

可以把文本中的字符串赋给一些键。比如，把单个的键或组合键定义为一个完整的 DOS 命令(包括命令最后的回车字符)。这样就可以建立一个完全由键定义的命令宏的集合。由键盘的键来定义的宏命令与 DOSKEY 所定义的宏指令是不同的，由键盘定义的宏命令不能带有可替换的命令行参数，而 DOSKEY 定义的宏指令却可以带这类可替换参数。

字符串的定义可以用下列的命令格式：

```
Esc[key;"string"p
```

下面是一个具体例子，把公司名赋给 Shift＋F1 组合键：

1. 在命令行方式下键入 PROMPT $E[0;84;"name"p$P$G，用所需的字符串代替 name。
2. 按 Shift＋F1 键。
   屏幕上将出现 PROMPT 命令中用双引号括起来的字符串。
   还可以在字符串中包含引号和数字，用分号分隔它们。
3. 如果用 F12 键代替 CD \命令，那么采用 PROMPT $E[0;134;"CD\";13p$P$G 命令。

注释： 上面的命令中 13 将为输入的 CD\命令加上回车符。

4. 按 F12 键，DOS 将转向根目录。
5. 如果使用 DOSKEY，按 F7 键将列出 DOSKEY 存储的所有命令，对于 DOSKEY 宏指令却没有相应的这种功能，用上述的方法定义 F10 键具有这个功能，具体命令如下：

   ```
   PROMPT $E[0;68;"DOSKEY /M";13p$P$G
   ```

6. 按 F10 键，则屏幕上列出所定义的所有 DOSKEY 宏指令。

到目前为止，我们已经介绍了 ANSI. SYS 中主要的一些重要功能，继续使用它们将发现它们对于用户编程，管理和维护系统有更大的用处。

48.8　控制 CON(控制台)模式

用 MODE 命令可以改变监视器文本显示的行数和列数;这种改变只能影响命令行方式的文本显示,而其他应用程序如 EDIT、DOS Shell 以及 MSBACKUP 等不受影响。图 48.5 给出 MODE 的命令格式,MODE 命令用来控制 CON 设备,包括键盘和显示器。

48.8.1　指定文本显示的行数和列数

ANSI.SYS 必须首先要安装在用户的系统中。可以用参数 LINES=n 来改变文本显示的行数,VGA 卡支持 43 行和 50 行的显示方式,EAG 卡只能支持 43 行显示方式。

同样,还可以把屏幕显示设为 40 列或 80 列的显示方式。当设置为 40 列的显示方式时,屏幕上将显示比通常情况下更大的字符,并且它还能和标准输出模式相作用。比如 DIR/W 命令可在这种模式显示文件目录的内容:

下面是一些模式改变的例子:

1. 输入命令 MODE CON COLS=40。
   屏幕显示将改为大字符显示。
2. 输入命令 MODE CON COLS=80 LINES=43。
   屏幕显示回到正常状态时的显示模式,但每屏为 43 行。
3. 输入 DIR 命令比较两种显示模式下不同的行密度。

提示: *启动 DOS Shell,在菜单 Option 选项的 Display 设置中可以看到用户的显示器可以支持哪几种文本显示的模式。*

4. 除非想改变当前设置,否则键入 MODE CON LINES=24。

48.8.2　控制键盘的响应

PC 机的键盘有自动打键的功能,当键被按住一段时间后,键盘将重复所按的键。如果键被按住的时间超过半秒钟,则这个键将以每秒钟 20 次重复出现。用箭头键移动光标或用退格键删除文本时都可使用键盘的这个特性。但是,用这种方法时要特制小心,否则就可能使光标跳过要到达的位置或删除了过多的字符,因为对于一般的用户来说键盘重复的频率太快了。

MODE

配置控制台设备。

句法:

```
MODE [display-adapter]
MODE CON[:] [COLS=c] [LINES=n]
MODE CON[:] RATE=r DELAY=d
```

参数：	
none	显示所有设备的当前配置的状态。
display-adapter	指定显示卡模式，有以下各种情况： 40 或 80 设定每行显示的字符数。 BW40 或 BW80 选择彩色适配卡的黑白显示模式，每行显示 40 或 80 个字符。 CO40 或 CO80 选择彩色适配卡的彩色显示模式，并确定每行显示的字符数。 MONO 选择单色适配卡，每行 80 个字符。
COLS=c	指定每行显示的字符数(40 或 80)。
LINES=n	指定每屏显示的行数(24，43 或 50)。
RATE=r	指定键盘重复字符的速率，数值可取 1 到 32，对应键盘每秒重复字符次数为 2 到 30。缺省值为 20。
DELAY=d	指定键盘重复之前按键所需的时间，d 取值为 1，2，3 及 4，键盘等待的时间为 d/4 秒。缺省时取 2(即 0.5 秒)。

图 48.5 MODE 命令总结(控制台功能)，指定行数和列数

用户可以自己设定比缺省值慢一些的键盘自动重复的速率，当然也可以设置得更快些。MODE 命令还可以让用户改变键盘重复打键之前等待的时间。

下面的练习可以让用户熟悉上面的命令：

1. 按下一个键而不松开，注意键盘开始自动打键前的时间以及重复打键的频率。
2. 输入命令 MODE CON RATE=2 DELAY=4。
3. 重复第一步骤，观察各种情况的改变。
4. 输入 MODE CON RATE=20 DELAY=2 恢复键盘的正常值。

注释： 在 AUTOEXEC.BAT 中使用 MODE 命令可以使用户开机之后就进入自己喜欢的显示模式或键盘响应速度。

48.9 在标准控制台设备间切换

控制台设备是用户键入命令以及 DOS 用来显示标准输出和标准错误输出的设备。在通常情况下，控制台设备是 CON。包括键盘和显示器。用户可以改变标准控制 台设备，使它不再是 CON。例如，用户有另外的设备与 COM4 相联，可以用下面的命令来改变标准控制台设备：

```
CTTY device
```

device 可以是任何 DOS 标准设备名：从 COM1 到 COM4，LPT1 到 LPT3，PRN，AUX，PRN 以及 CON。假如想通过一个远程终端来控制 DOS，而这个远程终端与 COM4 相联，可以用下列命令：

```
CTTY COM4
```

一旦敲入上述命令，用户就不能再通过键盘输入命令了，DOS 的显示也不再通过显示

器，而是由与 COM4 相连的设备来控制命令的输入和显示 DOS 的标准输出。如果要恢复正常的标准控制设备，通过与 COM4 相连的设备输入以下命令：

```
CTTY CON
```

注释： 如果从 CON 切换到标准设备之后，无法输入命令，则必须通过键盘重新启动计算机，重新启动之后，CON 又恢复为标准控制台设备。

48.10　节省膝上型计算机电池的电能

DOS 6 的 POWER 程序可以用来节省膝上型或笔记本型计算机电池的电能。当用户实际上并没有使用计算机时，它将减少电能的消耗。使用 POWER 前必须在 CONFIG.SYS 中安装 POWER.EXE 设备驱动程序：

```
DEVICE[HIGH]=C:\DOS\POWER.EXE [ADV[:advtype] | STD|OFF] [/LOW]
```

参数 advtype 可以是 MAX，REG 或 MIN。MAX 将最大限度地节省电能，REG 在节省电能和维持系统基本耗能间进行折衷；MIN 不干涉系统服务只是节省很少的电能。安装 POWER.EXE 不带任务参数，或者使用无值的 ADV 参数，则默认为 ADV:REG。

如果用户计算机支持高效能量管理(APM)规范，则可以在上述命令中包含参数 STD，以指定计算机使用硬件的能量管理功能。如果用户的计算机并不支持 APM，使用 STD 参数将关掉能量管理系统。参数 OFF 在任何情况下都将使能量管理系统不起作用。

注释： 如果用户的计算机支持 APM 规范，这意味着计算机可以节省 25%的能量，而不是通常情况下的 5%。

使用选项/LOW 把 POWER.EXE 装入常规内存，尽管高端内存可以使用。

48.10.1　POWER 命令

装入 POWER.EXE 驱动程序以后，可以用 POWER 命令来检查或改变当前 POWER 程序的设置。直接键入 POWER 命令不带任何参数，计算机将显示当前 POWER 程序的设置情况。如果要改变设置，则键入 POWER 命令，同时包含参数 ADV，STD 或 OFF。如果 POWER 驱动程序可能干扰。用户即将运行的用户程序，可以用下面的命令避免这种干扰：

```
POWER ADV:REG
```

如果这样还不行，就用下面的命令：

```
POWER ADV:MIN
```

如果仍然不行，而用户计算机支持 APM 规范，则用下面的命令：

```
POWER STD
```

如果仍然不行，则只好关掉这个能量管理程序：

```
POWER OFF
```

提示： 如果计算机(笔记本)有它自己的能量管理程序，它就会经常关掉屏幕显示，即使在使用交流电源适配器代替电池工作时也如此。这时必须用命令 POWER OFF 使屏幕能显示。当用户使用交流电源时可以试一下这种方法。不过，用户应该养成习惯：在没有使用交流电源时输入命令：POWER ADV:MAX

48.11 自测练习

1. 设计一个你自己喜欢的命令行屏幕显示模式，在 AUTOEXEC.BAT 文件里键入这个 PROMPT 命令，使之在计算机启动就被执行。
2. 创建一个键盘的宏，按 Alt+M 时代替命令 MEM /C /P。
3. 编写一个自己的 BOXTEST 批处理程序。在方框内显示自己给定的信息，试一试用不同的颜色显示这个方框，用不同的边框格式，还可以用文本闪烁的属性来显示信息。
4. 如果用户不需要保留的话，恢复 CONFIG.SYS，即把本章使用的设备驱动程序从 CONFIG.SYS 中删掉：删掉 ANSI.SYS 和 POWER.EXE。
5. 删掉本章建立的但用户不再需要的文件：BOXTEST，BOXTEST.BAT，COLORTEST，COLOR.BAT。

第 49 章　处理另一个国家的信息

简介： DOS 兼容的计算机的内部配置都是为制造计算机的国家或买这些计算机的国家而设的。但是，用户可以在自己的计算机上临时性或永久性的设置其他国家的配置，以处理这些国家的信息。

49.1　用其他国家的语言通信

假如整个计算机系统是在美国购买的，那么键盘、监视器以及打印机可能都只能对标准美国英文字母进行操作。

数字、日期及时间都可能采用美国的传统格式。假如用户是在其他国家购买的计算机，那么其字符集、数字、日期及时间格式都可能采用这个国家的标准。

但是用户常常希望能用自己的计算机为其他国家的人们准备或编写文本，而这些人有可能倾向于使用他们自己熟悉的日期和时间格式。有时用户要为其他国家的人回函或提交报告，或者使用常用的外语字符，但这些字符并不出现在键盘上。

本章将阐述使用 DOS 处理国际通信问题的几种方法，即包括下面方面内容：

- 使用另一个国家的日期、时间和数字格式，以及排序的顺序
- 设置自己的键盘，以打印国际字符
- 使用另一个国家的键盘布局
- 使用另一个国家的字符集（字符编码页）

49.2　总　览

DOS 提供了一整套命令，以改变它在处理国际交流问题的方式。KEYB 命令可以模仿或者解释一个特定国家的键盘，COUNTRY 命令使系统采用其他国家或地区的数字格式。

在监视器和打印机中都装有字符代码页。字符代码页实际上是一张表，它告诉设备每个单字节的代码（从 0 到 255）所对应的字符。对于标准的英文代码页，代码“66”对应字符“B”；代码“123”对应字符“{”；代码“156”对应字符“£”；而代码 168 对应字符“¿”。改变字符代码页就是告诉 DOS 对于同一代码使用不同的字符集。

DOS 提供了 6 套软件代码页，这些代码页可以下装到设备内，以替换原设备中的硬件代码页。每个代码页中的前 128 个编码（从 0 到 127）所对应的字符是相同的，因为它们是标准的 ASCII 码，它们之后的编码对应的字符随代码页的不同而不同。在《DOS 用户指南》的附录中列出了每张代码页的全部内容。

用户有时可以不通过 COUNTRY 或 KEYB 命令就能使用不同的代码页，但是，如果需要经常性地改变代码页，最好还是从建立系统设置开始，以便能很方便地改变代码页。

49.3 国际格式

其他国家的日期、时间和数字格式常常和美国的格式不同，例如，有可能把年号放在最前面，或者使用不同的货币符号，在数字表示中的逗号和小数点等都可能不一样。但用户可以让 DOS 采用一个特定国家或地区的格式。(这些格式会影响诸如 DATE、TIME、DIR 以及一些应用程序如 DOS Shell 和 Backup 的执行结果，但它们不会对 Windows 产生影响)。

如果一个国家使用 é 和 ü 这样的字符，在搜索排序时，它们的位置可能和面向美国的代码页中的位置不同。假如要经常使用这些字符，那么用户会希望根据这些国家的标准排列它们的顺序。(该排列顺序影响 SORT 及 DIR 命令的执行结果，同时也会影响在 DOS 下运行的应用程序的执行结果，但对 Windows 及其应用程序不产生影响)。

表 49.1 列出了 DOS 所支持的国家的代码、它们的时间和日期格式。

表 49.1 国家及相应的信息格式

国家	代码	日期	时间
Belgium	032	05/01/1992	17:15:25,00
Brazil	055	05/01/1992	17:15:25,00
Canadian-French	002	1992-01-05	17:15:25,00
Croatia	038	1992-01-05	17:15:25,00
Czech Republic	042	1992-01-05	17:15:25,00
Denmark	045	05-01-1992	17.15.25,00
Finland	358	05.01.1992	17.15.25,00
France	033	05.01.1992	17:15:25,00
Germany	049	05.01.1992	17:15:25,00
Hungary	036	1992-01-05	17:15:25,00
Int'l Englist	061	05/01/1992	17:15:25.00
Italy	039	05/01/1992	17:15:35,00
Latin America	003	05/01/1992	17:15:25.00p
Netherlands	031	05-01-1992	17:15:25.00
Norway	047	05.01.1992	17:15:25,00
Poland	048	1992-01-05	17:15:25,00
Portugal	351	05-01-1992	17:15:25,00
Serbia/Yugoslavia	038	1992-01-05	17:15:25,00
Slovakia	042	1992-01-05	17:15:25,00
Slovenia	038	1992-01-05	17:15:25,00
Spain	034	05/01/1992	17:15:25,00
Sweden	046	1992-01-05	17:15:25,00
Switzerland	041	05.01.1992	17:15:25.00
United Kingdom	044	05/01/1992	17:15:25.00
United States	001	01/05/1992	05:15:25.00p

注释： 在特殊的 DOS 版本中，配置有阿拉伯、以色列、日本、朝鲜、中华人民共和国及台湾的文字字符，可以向 Microsoft 访问如何购买这些 DOS 版本。

COUNTRY 配置命令告诉 DOS 去使用一个特定国家的信息格式，图 49.1 示出了这个命令的句法。该命令写在 CONFIG.SYS 文件中。

COUNTRY

说明一个国家的日期和时间等信息格式、货币符号及排序顺序、标点和小数点等。

句法：

COUNTRY=country[,[page][,filespec]]

参数：

country	指定国家代码。
page	指定所要的国家信息代码页。
filespec	指明包含国家信息的文件。

图 49.1　COUNTRY 配置命令总结

在 COUNTRY 命令中必须包含国家代码(见前面表 49.1)，国家代码常常(但不总是)和该国家的国际电话区位号一致。不过命令中的代码页参数常被省略，它是以前遗留下来的，现在没有其他命令对它进行控制，本章后面将会讲到。但是，即使省略了代码页，也必须在命令中写上一个逗号。例如，要使用匈牙利格式，则必须在 CONFIG.SYS 文件中写入下述命令，但注意 COUNTRY=036 和 C:\DOS\COUNTRY.SYS 之间的逗号：

```
COUNTRY=036,,C:\DOS\COUNTRY.SYS
```

在装入 COUNTRY.SYS 驱动程序时，DOS 必须先找到 COUNTRY.SYS 文件。如果该文件存放在启动盘的根目录下，这不会出现任何问题，但很多人都倾向于把有 DOS 文件放在\DOS 目录下，所以命令中必须包括完整的路径名。

下面的练习是使计算机采用拉丁美洲的信息格式：

1. 在 CONFIG.SYS 命令中插入下列命令：

   ```
   COUNTRY=003,,C:\DOS\COUNTRY.SYS
   ```

2. 退出编辑程序，回到 DOS 命令提示符下后，再重新启动计算机。
3. 用 DIR 命令列出根目录下的文件清单。
   注意日期格式。
4. 启动 DOS Shell，请注意文件列表中的日期。
5. 退出 DOS Shell。
6. 在设置新的日期时，必须使用拉丁美洲格式。键入 DATE 31/12/95 命令，把日期设置为 1995 年的最后一天。
7. 再恢复正确的日期设置。
8. 从 CONFIG.SYS 文件中删除 COUNTRY 命令行，重新启动计算机，从而使计算机恢复先前的配置。

49.4 打印国际字符

一个代码页包括 256 个字符，不过其中很多都不能直接输入。监视器内装的缺省代码页中的许多可用字符在键盘上都没有相对应的单键或组合键。不过，按住 Alt 键，再从数字键盘上输入一个代码，便可以打印出任何字符(但注意 NumLock 键应为 on 状态)。例如，如果当前所使用的是标准英国代码页(437)，但想打印字符 è，于是可以按住 Alt 键，再在数字键盘上输入代码 138。当松开 Alt 键后，字母 è 便出现在屏幕上了。这样便可以通过标准打印机打印这个字符。

在 DOS 命令提示符下，用户可以在命令中使用扩展字符。例如，可以用 FIND 命令从文件中搜索字符串“Bo rg”，可以把目录名定为 MAÑANA，或者给一个文件取名为 HÔPITAL。这些名字在 DOS Shell 的文件列表、Windows 的 File Manager 以及一些显示文件列表的应用程序中都能正确显示出来。

试练习输入这些名字：

1. 在 DOS 命令提示符下，先键入 B，然后输入 Alt＋148 就键入了 Ö，接着键入 rg。
2. 按下 ESC 键清除刚输入的命令行。
3. 键入 MA，接着输入 Alt＋165 以键入 Ñ，最后键入 ANA。
4. 按 ESC 键清除命令行。

一些桌面排版软件，比如文字处理程序或桌面印刷软件包，它们的字符集和计算机中配置的代码页可能不同。事实上，相同的代码可能对应不同的字符，这取决于用户所使用的印刷字体。还应当记住，Windows 所使用的字符集也不相同。

注释： Windows 使用的是 ANSI 字符集而不是 DOS 的 ASCII 字符集。ANSI 字符中不包括画直线的字符，但提供了更多的国际字符和符号。不幸的是，如果在 DOS 应用程序比如在 EDIT 下打开由 Windows 应用程序建立的文件，如果这个文件含有 Windows 的增强字符，这些增强字符就会被 DOS 应用程序按 ASCII 字符进行解释，从而会出现错误。很多 Windows 应用程序，包括 Windows 的 Microsoft Word，都允许用户使用 DOS 的文本格式存储文件，这些文件便能由 DOS 编辑器比如 EDIT 读出来(不过这些文件在以 DOS 格式存储时，它们便失去了黑体和斜体等字体形式)。

假如用户是在 DOS 命令提示符下工作，而想建立使用扩展字符的捷径，就可以装入 ANSI.SYS 驱动程序。下面的命令是把 ä 字符和 Ctrl＋A 对应，这样便可以在 DOS 命令提示符下很方便地键入 ä。

```
PROMPT $E[1;"ä"p$P$G
```

也可以使 Alt＋132 和 ä 对应，第 48 章“控制台的控制”中详细讲了 ANSI.SYS 的使用。

注释： Remapped 键只在 DOS 命令提示符屏幕状态下有效，不能在 Shell 或其他软件中使用，但输入 Alt＋number(数字)在许多地方均可用。

49.5　使用国际键盘布局

假如在工作时要使用很多非英语字符，此时就希望 DOS 使用相应的键盘布局，以便能直接输入这些字符。有时候甚至希望改变大小写的控制。KEYB 命令用于指明用户所希望的键盘布局，DOS 据此对按键分配不同的代码，并对输入的字符作解释（忽略 Keycaps 的状态）。键盘布局影响 DOS 命令提示屏幕以及在 DOS 下运行的应用程序，如 DOS Shell 及 EDIT。但不影响 Windows 及其应用程序。

表 49.2 列出了 DOS 6.22 支持的键盘布局，以及缺省代码页和可以使用的替换代码页（在下一部分会解释代码页）。ID 一列中示出了键盘的 ID 码。在《DOS 用户指南》的附录中列出了每种键盘的确切布局。

表 49.2　键盘代码和代码页

国家/区域	代码	标准	备用	ID
Belgium	BE	850	437	
Brazil	BR	850	437	
Canadian-French	CF	863	850	
Czech	CZ	852	850	
Denmark	DK	850	865	
Finland	SU	850	437	
France	FR	850	437	120,189
Germany	GR	850	437	
Hungary	HU	852	850	
International	+	437	850	
Italy	IT	850	437	141,142
Latin America	LA	850	437	
Netherlands	NL	850	437	
Norway	NO	850	865	
Poland	PL	852	850	
Portugal	PO	850	860	
Slovak	SL	852	850	
Spain	SP	850	437	
Sweden	SV	850	437	
Switzerland(French)	SF	850	437	
Switzerland(German)	SG	850	437	
United Kingdom	UK	437	850	166,168
United States	US	437	850	
Yugoslavia (Croatia,Slovakia,Slovenia)	YU	852	850	

49.5.1　KEYB 命令

图 49.2 示出了 KEYB 命令的句法。KEYB 命令既可以在 DOS 命令提示符下直接使

用，也可以通过 CONFIG. SYS 文件装入。键盘代码指定要使用的键盘布局。如果是通过 CONFIG. SYS 装入的，而 KEYBOARD. SYS 又不在启动盘的根目录下，那么在 KEYB 命令行中就必须说明全部路径。当在 DOS 命令提示符下使用该命令时，就可以根据所建立的搜索路径找到 KEYBOARD. SYS 文件的存放地点。

提示： 可以使用 LH 把 KEYB 装入 UMB(高端内存)。但是，由于 KEYB 不和 DEVICE 命令一起装入，因此不能通过 CONFIG. SYS 把 KEYB. COM 装入高端内存。

提示： 如果用户需要经常使用非标准键盘布局，那么在 CONFIG. SYS 文件或 AUTOEXEC. BAT 文件中写上适当的命令，使这种键盘布局变为缺省布局。

试练习装入拉丁美洲的键盘布局：

1. 键入 KEYB LA 命令。
2. 通过键盘输入这些字符：
   按=打印 ¿
   按+打印 i
   按;打印 ñ
3. 要查看当前所装的键盘布局，则键入没有参数的 KEYB 命令，然后 DOS 将显示如下信息：

```
Current keyboard   code:LA   code page:  850
```

因为还没有装入代码页，所以会出现下列信息：

```
Active code page not available from CON device
```

KEYB

装入 KEYB TSR，使用户能用自己的键盘仿真别的国家的键盘布局及键入其他语言的文字。

句法：

KEYB [code[,[page] [,filespec]]]

从 CONFIG. SYS 文件中装入时格式如下：

INSTALL=[path] KEYB. COM [code[,[page],[filespec]]]

参数和选项：

none	显示当前的键盘代码和代码页，以及当前监视器的代码页。
code	指明要使用的键盘布局所属的国家代码。
page	指明要使用的代码页，缺省值为当前使用的代码页。
filespec	指明键盘定义文件的名字及存放位置，缺省值是 KEYBOARD. SYS，存放在当前目录下或搜索路径中。
path	当 KEYB. COM 不在根目录下时，指示其路径。

图 49.2 KEYB 命令总结

当安装好一种键盘布局后，也许用户会发现在这种非标准键盘布局下输入熟悉的 DOS 命令时会出现一些困难。例如，如果刚装好的是位丁美洲键盘布局，那么按/键时屏幕上打印出连字号(－)。如果要输入字符/，就得按 Shift＋7 组合键。但至少会发现，在想输入 DOS 命令时，能够很容易切换回到原来的美国键盘布局，而在准备打印一个国际文本时，又容易切换到另一种键盘布局。也就是说能够在所装的键盘布局和缺省的(美国)键盘布局间来回切换，按 Ctrl＋Alt＋F1 键，便回到硬件中装的键盘驱动程序，而按 Ctrl＋Alt＋F2 键，又转到新装的驱动程序的控制之下。

1. 按 Ctrl＋Alt＋F1 键，激活缺省的键盘布局。
2. 试着键入＝，＋及；。
   这些键都是标准字符。
3. 按 Ctrl＋Alt＋F2 键，回到拉丁美洲键盘布局。
4. 再按＝，＋及；等键。
   现在按这些键，屏幕上便出现拉丁字符而不是原来的标准字符。

注释： DOS 清楚地说明 850 代码页不能使用，为什么屏幕上还会出现拉丁字符呢？这是因为这些字符存在标准的 437 代码页中。事实上 437 代码页中只有部分字符和 850 代码页不同。其他代码页之间的差别也很小。

注释： 一些国家使用标准的打字机键盘布局，这种布局和计算机键盘有所不同，可以通过按 Ctrl＋Alt＋F7 转到这些打字机键盘布局中。

49.6　使用代码页

键盘布局和代码页之间有何关系?键盘布局决定了每个按键所产生的代码，而代码页决定了按键产生的代码会在显示器上出现什么字符。

每种键盘布局都对应着两种代码页——一个是标准的，另一个用作替换(见表 49.2 所示)。它们都能显示从键盘输入的字符。当然，最好是装入准备使用的键盘布局的标准代码页。如果用户要在几种键盘布局之间来回切换，则最好装入 850 代码页(它是多语种代码页)，对于每种键盘布局，它既可以用作标准代码页，也可以用作替代码页。在一些键盘布局中亦可使用 437 代码页(英国代码页)，它很可能就是用户计算机中硬件设置的缺省代码页，因此不需要做别的什么事就可以直接使用。

如果所使用的代码页不是硬件设置的，在使用它之前就得先将其装入。如果在键入 KEYB 命令之前该代码页已经装入，则 KEYB 自动切换到所需的代码页。例如，如果已经装入了代码页 850 和 852，下面的命令切换到波兰键盘布局，并使用监视器的 852 代码页，它是波兰键盘布局的缺省代码页：

```
KEYB PL
```

下面的命令切换到波兰键盘布局，并使用 850 代码页：

```
KEYB PL,850
```

49.6.1 怎样装入代码页

在使用代码页(指非硬件代码页)之前必须几个命令联合作用。如果 KEYB 或 COUNTRY 需要一个软件代码页,那么在该代码页生效之前应把它装入内存。同时 CONFIG.SYS 文件必须为监视器安装一个设备驱动程序(DISPLAY.SYS),这样才能使用装入的代码页。

当所需的环境建立好后,又需要几个命令来处理装入的代码页。如果用户每次都要使用一个代码页,则可以把这些命令写到 AUTOEXEC.BAT 文件中去。或者一个一个地在 DOS 提示符下把它们键入计算机。NLSFUNC 命令建立本国语言支持(National Language Support),这使得在切换代码页时要容易些。MODE 命令装入特定的软件代码页。别的 MODE 命令可以切换代码或显示当前状态,CHCP 命令用于改变国家代码。

49.6.2 监视器代码页

标准的单色和 CGA 监视器不能使用监视器代码页。但是,所有 EGA、VGA、LCD 以及更高级的型号则可以。在监视器上使用软件代码页之前,必须装入驱动程序 DISPLAY.SYS (如图 49.3 所示)。

和其他设备驱动程序一样,如果系统允许的话,可以用 DEVICEHIGH 把 DISPLAY.SYS 装入高端内存。下面的命令为 LCD 监视器装入 DISPLAY.SYS:

```
DEVICE=C:\DOS\DISPLAY.SYS CON=(LCD)
```

下面的命令装入 DISPLAY.SYS,并让 DOS 判断监视器的类型,它指定了缺省代码页,所以可以切换回去,并能使用两个软件代码页:

```
DEVICEHIGH=C:\DOS\DISPLAY.SYS CON=(,437,2)
```

DISPLAY

支持屏幕和键盘代码页的改变。

句法:

DEVICE [HIGH]=[path] DISPLAY.SYS CON=([type] [,codepage] [,max])

参数:

path	指明 DISPLAY.SYS 文件的路径,缺省值指启动盘的根目录。
type	指明所用的显示适配器,有效值为 EGA 和 LCD,而 EGA 包括 EGA 和 VGA 两种适配器。如果用户没有说明适配器类型,则 DISPLAY.SYS 将对硬件进行检查,以确定适配器的类型。
codepage	指明硬件中所装的代码页。
max	说明要装入的代码页数目,但硬件中的代码页除外。对于 EGA,有效值为 0 到 6,对于 LCD,有效值为 0 到 1。

图 49.3 DISPLAY.SYS DEVICE 命令总结

如果在命令行中省略硬件代码页,在装入别的代码页后,就不能再重新回到硬件代码页去。不过,如果用户只使用软件代码页,能否切换到硬件代码页就无关紧要了。代码页的数

目多少决定为切换代码页所保留的内存空间的大小，其缺省值为 1。如果用户希望能够在硬件代码页(437)及多语种代码页(850)之间来回切换，则只需储存一个软件代码页即可。但是，如果要在 850、852 及 860 代码页间来回切换，就必须在内存中存储三个代码页。

现可以练习装入 DISPLAY.SYS：

1. 在 CONFIG.SYS 文件中插入下述命令：

```
DEVICEHIGH=C:\DOS\DISPLAY.SYS CON=(,437,1)
```

2. 退出所有的程序，回到 DOS 命令提示符下，然后重新启动计算机。

在下一部分中就可以装入一个软件代码页了。

49.6.3　装入软件代码页

在软件代码页投入使用之前，必须做好一切准备。这个准备过程即是把代码页装入内存并使它处于可用状态。这样 COUNTRY 和 KEYB 命令就可以使用它了。图 49.4 给出了怎样使用 MODE 命令管理代码页。

MODE

建立并控制代码页的切换，显示打印机及监视器所装的代码页数目。

句法：

```
MODE CON CP PREP =((page [...]) filespec)
MODE CON CP SEL = page
MODE CON CP REF
MODE CON CP [/STA]
```

参数和选项：

PREPARE	为指定设备准备代码页，缩写为 PREP。
page	说明要使用的代码页。
filespec	指明代码页信息(.CPI)文件的路径及文件名称。
SELECT	装入一个代码页以供使用，缩写为 SEL。
REFRESH	若已启动的代码页因硬件故障或其他错误而丢失，该命令重新启动原已准备好的代码页。缩写为 REF。
/STATUS	显示当前已准备好的代码页的数目，缩写为/STA。

注释：

使用不带 PREP、SEL、REF 或/STA 参数的 MODE 命令，和带/STA 参数时的结果是一样的。

DOS 提供的代码页信息文件如下：

EGA.CPI	适用于 EGA 或 VGA，或 IBM PS/2。
LCD.CPI	适用于 IBM PC 液晶显示器。

图 49.4　MODE 命令总结(设置代码页)

每个命令都说明一种设备，随后是一个字 CODEPAGE，再后面便是要进行的操作。PREP 命令安装代码页，如果要为屏幕安装多语种代码页，则命令如下：

```
MODE CON CODEPAGE PREP=((850) C:\DOS\EGA.CPI)
```

如果在设备驱动程序配置命令中说明了硬件代码页，则可以在软件代码页和硬件代码页之间来回切换。如果通过一个或多个 MODE 命令将装入几个代码页，也可以在这几个代码页之间切换。

现在练习装入多语种代码页：

1. 如果监视适配器是 EGA 或 VGA，则键入 MODE CON CODEPAGE PREPARE=((850) C:\DOS\EGA.CPI)命令，如果是液晶监视器，则键入 MODE CON CODEPAGE PREPARE=((850) C:\DOS\LCD.CPI)命令。

49.6.4 切换代码页

当从一个代码页切换到另一个不同的代码页时，监视器就使用新的代码表，来把单字节的代码转换成屏幕上显示的字符。一些已经存在于屏幕上的字符也改变为新的字符集。

如果在程序运行过程中改变代码页，则有可能引起内部冲突，为了避免出现这个问题，在改变代码页之前，应退出所有的程序。

警告： *如果正在使用任务对换程序，当正在执行任务时，千万别改变代码页，否则会引起严重的问题。*

MODE 命令的 CODEPAGE SELECT 方式启动已准备好的代码页。下面的命令用于启动多语种代码页：

```
MODE CON CODEPAGE SEL=850
```

要启动硬件代码页，则使用如下命令：

```
MODE CON CODEPAGE SEL=437
```

如果在设备驱动程序命令中没有说明硬件代码页，则在选择不同的代码页后，就不能返回硬件代码页了。

练习装入多语种代码页：

1. 键入 MODE CON CODEPAGE SEL=850 命令。
2. 键入 KEYB LA 命令，装入拉丁美洲键盘布局。
3. 键入无参数的 KEYB 命令，查看装入的键盘布局及代码页。
   将看到出现下列信息：

```
Current keyboard code: LA code page:850
Current CON code page: 850
```

4. 现在键入一些属于多语种代码页的特殊字符：按 Alt+157 以输入 ¢；按 Alt+189 以输入 ¢；按 Alt+226 以输入 Ô。
5. 键入 MODE CON CODEPAGE SELECT=437 命令，切换回到硬件代码页。
6. 再按 Alt+157、Alt+189 及 Alt+226 等键，将会发现这次出现的字符与上次不同。

49.6.5 一个完整的代码页装载实例

假定用户有时需要使用葡萄牙键盘及一些常用格式，就可以使用替代的葡萄牙代码页

(860)。因为用户是用英语及硬件代码页(437)工作的,所以希望能在两种代码页之间切换。

首先在 CONFIG. SYS 文件中插入下述命令:

```
COUNTRY=351,860,C:\DOS\COUNTRY.SYS
DEVICEHIGH=C:\DOS\DISPLAY.SYS CON=(,437)
```

COUNTRY 命令建立葡萄牙(国家代码 351)的数字及其他格式,同时也说明了交替代码页。DEVICEHIGH 命令为一个软件代码页保留出内存空间,同时它也定义 437 代码页为硬件代码页,这样在需要时就可以切换回去。

因为不能肯定将先用哪种键盘布局及代码页,所以把一些命令留下来,以后在 DOS 提示符下直接键入。把下述命令写到 AUTOEXEC. BAT 中去:

```
MODE CON CODEPAGE PREPARE=((860) C:\DOS\EGA.CPI)
```

MODE 命令为指定的设备装上软件代码页。

在 AUTOEXEC. BAT 执行完毕后,便能够对键盘布局及代码页进行切换了。也可以在 DOS 命令提示符下键入下列命令,以管理键盘布局及监视器代码页。

```
MODE CON CODEPAGE SELECT=860    把监视器切换到代码页 860
MODE CON CODEPAGE SELECT=437    把监视器切换到代码页 437
[LH] KEYB PO                    切换到葡萄牙键盘布局
KEYB US                         切换到美国英文键盘布局
```

假如是在键盘上输入这些命令或类似的命令,就可以建立起 DOSKEY 宏命令,使操作简便些。下面的批文件包含两个 DOSKEY 宏定义:

```
DOSKEY C8=MODE CON CODEPAGE SELECT=860
DOSKEY C4=MODE CON CODEPAGE SELECT=437
```

也可以设置组合键来表示这些命令。例如,下面的命令设置组合键 Shift+F1,使它可以完成到代码页 860 的切换,并恢复常用的命令提示符:

```
PROMPT $E[0;84;"MODE CON CODEPAGE SELECT=860"p$P$G
```

用户可以通过按 Ctrl+Alt+F2 键访问驻留在内存中的代码页(葡萄牙代码页),并通过按 Ctrl+Alt+F1 键回到硬件代码页。

49.6.6　本国语言支持(National Language Support)

DOS 的 National Language Support 功能使得代码页切换更加容易,而不论是内部切换还是外部切换。虽然用户仍使用 MODE 命令装入代码页,但不一定要使用 MODE 命令来改变代码页。要使用 National Language Support 功能,必须先把 NLSFUNC 作为 TSR 装入。图 49.5 示出了该命令的句法。如果 NLSFUNC 已经加入到 CONFIG. SYS 文件的 COUNTRY 命令中,就不必对其文件名和路径作进一步说明。缺省时 COUNTRY. SYS 在启动盘的根目录下。

可以在 DOS 提示符下直接装入 NLSFUNC 程序,亦可通过 CONFIG. SYS INSTALL 命令装入。使用 INSTALL 命令时,TSR 比 COMMAND. COM 要先装入内存,这有可能克服语言支持功能不工作的问题。如果使用 INSTALL,随后必须用 DEVICE 命令装入 DISPLAY. SYS(使用 INSTALL 时,不能把 TSR 装入高端内存块)。

NLSFUNC

启动为 NLS 装入国家代码信息的程序。

句法：

NLSFUNC [filespec]

参数和选项：

none	使用 COUNTRY=这个命令中所指定的驱动器和文件名(缺省值)。
filespec	指明含有国家代码信息的文件。

图 49.5 NLSFUNC 命令总结

装好 National Language Support 之后，就可以用 CHCP 改变代码页了(见图 49.6)。没有带参数的 CHCP 命令显示活动代码页的数目，即使 NLSFUNC 没被装入。

CHCP

改变所有设备的代码页，或显示当前代码页的数目：

句法：

CHCP [page]

参数：

none	显示当前代码页的数目。
page	改变到指定的代码页。

图 49.6 CHCP 命令总结

49.6.7 使用 NLSFUNC 和 CHCP 的另一个代码页装载实例

假如用户希望使用 National Language Support 功能，并且使用多语种代码页及拉丁美洲键盘布局，则在 CONFIG.SYS 文件中需要写入下述命令：

```
COUNTRY=003,,C:\DOS\COUNTRY.SYS
DEVICEHIGH=C:\DOS\DISPLAY.SYS CON=(,437,1)
```

并且仍然需要用下述命令准备好代码页：

```
MODE CON CODEPAGE PREPARE=((850) C:\DOS\EGA.CPI)
```

现在可以装入 NLSFUNC 并切换代码页了：

1. 键入 LH NLSFUNC 命令，装入语言支持。
2. 键入 CHCP 850 命令，选择多语种代码页。
3. 键入 KEYB LA 命令，装入拉丁美洲键盘布局。
4. 键入 KEYB 命令，查看一下哪个键盘布局和代码页被装入了内存。

49.6.8 熟练使用代码页

有时，DOS 可能丢失代码页，在这种情况下，就用 MODE 命令把代码页重新装入。下面

的命令为打印机重新装入活动代码页：

```
MODE CON CP REF
```

如果监视器被锁死，用户便不得不重新启动计算机。在这种情形下，CONFIG. SYS 及 AUTOEXEC. BAT 中的命令将被重新执行一遍。然后用户可以在键盘上用 MODE 命令改变代码页，或者使用 CHCP(如果 NLSFUNC 已装入的话)。

49.6.9　显示代码页信息

有几个命令能为用户提供系统的状态信息，这些命令都可以在 DOS 提示符下使用，如需要的话亦可通过 DOS Shell 使用。

无选项的 KEYB 命令，显示当剪的键盘信息，信息格式如下：

```
Current keyboard code: BE code page: 850
Current CON code page: 850
```

该信息给出了当前的键盘布局及正在起作用的代码页。如果以前没有使用过 KEYB 命令，则显示的信息如下：

```
KEYB has not been installed
Active code page not available from CON device
```

没带参数的 CHCP 命令只显示当前的代码页信息：

```
Active code page: 850
```

即使 NLSFUNC 没被装入内存，不带参数的 CHCP 命令仍然会给出上述报告。

不带参数的 MODE 命令显示计算机所连接的所有设备的信息及它们的当前代码页，它提供的信息量大，所以要使 MODE|MORE 命令或将输出重新定向，这样才能看清所有的报告内容。

带/STATUS 选项的 MODE 命令显示指定设备的信息，例如，命令 MODE CON CODEPAGE /STATUS 给出下列信息：

```
Active code page for device CON in 850
Hardware code pages:
  code page 437
Prepared code pages:
  code page 850
MODE status code page function completed
```

在大多数情况下，/STATUS 是缺省值，因此在命令中不管它是否存在，都会给出相同的信息报告。

49.7　自 测 练 习

本章讲述了如何使计算机处理国际信息，用户可以作一些练习，把自己的计算机设置成法国加拿大信息格式及键盘布局，并使用合适的代码页。

当完成该练习后，把计算机恢复成原来的正常设置(除非自己想采用刚设置好的国际格式)。

第50章 DOS的打印功能

简介： 用户在打印文件时，大多数情况都使用别的应用程序。但是，当把目录列表重定向到打印机、当使用屏幕打印功能打印CHKDSK信息以及在打印DOS帮助信息时，都要用到DOS的打印功能。

50.1 DOS最基本的打印服务

当使用应用程序例如字处理程序打印文件时，这些应用程序很可能有它们自己的打印驱动程序。比较高级的面向打印的应用程序都在它们的打印驱动程序上费了不少功夫，以最大限度地利用流行的打印机模块。

DOS不是面向打印的，但它提供了一些原始的打印服务，这样用户可以通过按PrintScreen键进行屏幕打印，或者打印ASCII文本文件(如README文件等)，以及将标准输出重定向到打印机，或者让监视器处于模仿打印状态。DOS假定打印机接在LPT1端口上，并且是最基本的打印机，但用户也可以使用其他类型的打印机，甚至可以是接在串行口上的激光打印机。

本章将讨论在DOS下打印文件的几种技术，主要包括：

- 使打印机打印出现在屏幕上的每一行内容
- 在文本或图形方式下，拷贝屏幕
- 在DOS下打印时，定义每英寸要打印的行数及列数
- 配置PRINT命令，管理打印队列

50.2 打印机类型

DOS只使用最基本的打印机功能。鉴于此，它可以在任何PC兼容的打印机上完成打印。很多现代化的打印机能够模仿一些类型的基本打印机，因此，如果DOS不能完成同打印机的通信，可查看它的说明书，查出怎样使用该打印机模仿IBM打印机、Hewlett-Packard打印机、Epson打印机或Diablo打印机。DOS能很好地操作这些类型的打印机。

DOS不能直接同PostScript打印机通信，但大多数PostScript打印机能够模拟Epson等这些打印机的一种，只是用户必须在打印机上设置一个开关。不过，在使用有PostScript打印驱动程序的软件包之前，一定不要忘记把打印机的开关设置恢复正常。

DOS经常操作并行打印机，并把LPT1作为缺省打印机端口；设备PRN也指向同一个端口。当使用打印或屏幕打印功能时，DOS把输出送到PRN上去。但可以使用MODE命令把并行端口输出重定向到串行端口，本章后面将阐述这方面的内容。

50.2.1　回显打印功能

前面已经讲述了如何把单个命令的输出重定向到打印机，但有时也需要把一系列命令的输出都在打印机上打印出来。例如，有可能想打印一系列目录列表或诊断报告，或将工作会议存档。

回显打印功能将屏幕上出现的所有内容在缺省打印机上备份下来：提示符、命令、错误信息以及命令输出。它只对从命令提示符屏幕上滚动过的命令及信息产生响应。如果在回显打印正在执行期间启动了别的程序，在这些程序中止后，回显打印又重新恢复。Ctrl+PrintScreen 控制回显打印的启动和关闭。在 PC 机上试练习：

1. 确定打印机所处的工作状态。
2. 按 Ctrl+PrintScreen 键。
3. 键入 DIR 命令。
   DOS 把 DIR 命令及其输出以及下一个 DOS 命令提示符，都在打印机上打印出来。
4. 键入 DIR \DOS 命令。
   DOS 继续打印屏幕上出现的任何内容。
5. 如要从打印机弹出最后一页，便按 Ctrl+L 键并按 Enter 键。
   虽然出现一错误信息，但仍将最后一页弹出来。
6. 再按 Ctrl+PrintScreen 键，关掉回显打印。

如果没有打印机接到或重定向到 LPT1 上，系统就将会挂起来。因此需要装好打印机并重新启动。在回显打印生效后，可按 Ctrl+PrintScreen 或者 Ctrl+P 键将其关闭。

注释： *不能在一个批处理程序及 DOSKEY 中控制回显打印的开启和关闭。Ctrl+P 或 Ctrl+PrintScreen 必须在命令提示符下键入。*

50.2.2　打印文本屏幕

如果用户想捕获从屏幕上滚动过去的文本信息，在此之前必须启动回显打印。如果文本已经显示在屏幕上了，便可以用屏幕打印把文本拷贝到打印机上去。仅仅按一下 PrintScreen 键(有的键盘上按 Shift+PrintScreen 键)，屏幕上的所有文本内容就在缺省打印机上打印出来。如果要打印图形方式的屏幕，例如 DOS Shell 的缺省屏幕，就要求先运行 GRAPHICS 命令，本章后面会讲到这个命令。

试练习打印屏幕：

1. 按一下 PrintScreen 键或 Shift+PrintScreen 键。
   在打印机上会打印出当前屏幕的内容。
2. 键入 ECHO ^L >PRN 命令(^L 代表 Ctrl+L)，弹出这一页。

答疑： *按 PrintScreen 键时，若系统死机，是怎么回事？*

如果在 LPT1 上没有可用的打印机时按下了 PrintScreen 键，系统将会发出蜂鸣声或挂起来，以等待准备好打印机。如果系统挂起来了，重新启动或准备好打印机都可使其恢复。

50.3 配置打印机

每台打印机在垂直方向都设有间隔尺寸，以每英寸多少行(lpi)计算，在水平方向的间隔尺寸则以每行多少字符计算。很多打印机的缺省值是 6 lpi 及每行 80 个字符。用户可以用 MODE 命令对这个尺寸作改动(如图 50.1 所示)。对于并行打印机，垂直方向可设为 6 lpi 或 8 lpi，水平方向可设为每行 80 个字符或 132 个字符。(该命令不能用于串行打印机及任何激光打印机和喷墨打印机)。

MODE

配置并行打印机，使其使用特定的垂直方向和水平方向的间隔尺寸，并重新动作。

句法：

MODE LPTn [COLS=c] [LINES=1] [RETRY=r]

参数：

LPTn	指明并行口：从 LPT1 到 LPT3。可以用 PRN 代替 LPT1。
c	指明每行字符数，可以为 80 或 132。
l	指明每英寸的行数，可为 6 或 8。
r	当打印机发生超时错时，不断测试与打印机连通。有效值有：
E	返回一错误。
B	返回忙信号“busy”。
P	继续测试，直到打印机接收输出为止。
R	返回就绪信号“ready”。
N	无测试返回值(缺省)。

图 50.1 MODE 命令总结(配置并行打印机)

试设置打印机为最大打印密度：

1. 键入 MODE LPT1 COLS=132 LTNES=8 命令。
2. 键入 DIR >PRN 命令，查看上述 MODE 命令是否生效。

如果 DOS 知道有打印机，但打印机还没有准备了接收信号时，RETRY 参数会告诉 DOS 如何行动。缺省时，它仅仅作放弃处理。参数 P 告诉 DOS 继续测试直到打印机接收输出为止。如果用户想打印东西而打印机已经关掉时，P 参数会使计算机挂起来。同样，如果出现纸用完或其他错误时，也会产生相同的结果。要跳出这一循环，可以准备好打印机或按 Ctrl+C 键。

在用 MODE 命令配置好打印机后，对以后的打印都会产生影响，包括使用 DOS 打印服务的所有应用程序。很多应用程序都使用它们自己的打印驱动程序，但当这些应用程序中断后，系统会恢复所配置的打印状态。

50.4　使用串行打印机

如果要在 DOS 下直接使用串行打印机，首先必须告诉 DOS 该设备的通信特征。如果用户只有串行打印机，或者打算把串行打印机作为 DOS 的缺省打印设备使用，就必须把 LPT1 重定向到该打印机上。

50.4.1　配置串行口

一个串行口(作为通信端口之一)可以连接到一台打印机上，也可以和鼠标、调制解调器或其他设备相连接。要在 DOS 下使用串行口，必须用 MODE 命令定义其通信特征(如图 50.2 所示)。本部分阐述怎样用 MODE 命令为打印机配置一串行口。这些参数对任何串行设备都适用。

用户不能对通信特征作猜测，必须使用所连设备的正确特征值。在大多数情况下，设备说明书都会告诉用户怎样使用 MODE 命令。下面的例子便给打印机配置下列参数：9 600 波特率但无奇偶校验，8 个数据位，一个停止位，并且继续测试直到打印机被接受为止：

```
MODE COM1 BAUD=96 PARITY=N DATA=8 STOP=1 RETRY=P
```

该配置一直作用于这个 COM 端口，直到其他 MODE 命令将其改变为止。

MODE

为一串行设备如串行打印机配置串行端口(COM)。

句法：

```
MODE COMn BAUD= baud [PARITY=parity] [DATA=data]
[STOP=stop] [RETRY=retry]
```

参数：

COMn	指明串行口，n 可以是 1 到 4。
baud	指明数据传送的波特率。(见下面有关取值的注意事项)。
parity	指明系统怎样用奇偶校验位检查数据传送中的错误(见下面的注意事项)。
data	指明每个字符的数据位数，其值可以是 5 到 8，缺省值是 7。
stop	说明表征一个字符传送结束的停止位数目。其值可以是 1，1.5 或 2。如果波特率是 100，其缺省值为 2；其他情况下缺省值为 1。
retry	串行通信发生超时错时，系统应采取的行动(见注意事项)。

注释：

波特率的取值如下：

11 或 110	110 波特
15 或 150	150 波特
30 或 300	300 波特
60 或 600	600 波特
12 或 1 200	1 200 波特

24 或 2 400	2 400 波特
48 或 4 800	4 800 波特
96 或 9 600	9 600 波特
19 或 19 200	19 200 波特

parity 的值可以是 N、E、O、M 或 S。N 代表不设校验，E 代表偶校验，O 代表奇校验，M 代表标志，S 代表空白。缺省值为 E。

retry 的取值如下：

E	若端口为忙时，返回一错误。
B	若端口忙时，返回 busy。
P	继续测试，直到端口接受输出。
R	若端口忙时，返回 ready。
N	不作任何继续测试即放弃(缺省值)。

如果在网络上使用 MODE 命令时，不能使用 retry 的任何取值。

MODE 命令的有些有效值并不是所有的计算机都支持，这些值有：

BAUD=19；PARITY=S；PARITY=M；DATA=5；DATA=6；以及 STOP=1.5。

除了 baud 外，如果省略其他所有参数，MODE 就采用在这个端口上最近配置的值。如果在此之前没有使用过 MODE 命令，MODE 便采用所有选项的缺省值。由于 baud 选项没有缺省值，因此必须加以说明。

图 50.2 MODE 命令总结(配置串行口)

用户不能对通信特征值作猜测，而必须使用所连设备的正确特征值。在大多数情况下，设备说明书中都告诉了用户如何写 MODE 命令。下面的命令是如下配置一台打印机：9600 波特，无奇偶校验，8 个数据位，1 个停止位并继续测试：

```
MODE COM1 BAUD=96 PARITY=N DATA=88 STOP=1 RETRY=P
```

该配置在 COM 端口上一直生效，直到其他 MODE 命令改变它为止。

注释： 一些 MODE 功能要求装入一个小的 TSR(驻留内存程序，0.5KB)。配置串行口便是这类要求驻留内存的功能之一。无论是哪个 MODE 命令，在装入 TSR 后都会显示下列信息：

```
Resident portion of MODE loaded
```

TSR 只需装入一次，然后所有 MODE 的要求驻留内存的功能都能使用。该 TSR 不能被装入高端内存块。当对 MODE 命令使用 LOADHIGH 命令时，DOS 会将其忽略。

50.4.2 重定向串行打印机

如果想让 DOS 把一台串行打印机当作 PRN 使用(比如用于打印屏幕，回显打印等等)，就必须使用如下的命令，把 LPT1 重定向到这台打印机：

```
MODE LPTn=COMn
```

例如，MODE LPT1=COM1 命令告诉 DOS，把所有定向到 LPT1 的数据送到 COM1 去输出。当然，可以在命令中说明任何有效的串行口(COM)和并行口(LPT)，但是，如果要把串行打印机当作缺省打印机使用，就必须对 LPT1 重定向。

注释： 该 MODE 命令格式只把并行口重定向到串行口，它不能用于重定向串行口，也不能用于把并行口重定向到其他任何类型的端口。

很多应用程序都依赖于 DOS 的打印机服务功能，因此，即使用户不作屏幕打印或回显打印操作，也可能需要重定向 LPT1。设置 COM 端口特征值的 MODE 命令必须在重定向命令之前执行，否则 DOS 就不能使用该 COM 端口。

如果要取消重定向，那么键入 MODE LPTn 命令(其中不再有=COMn)，然后将看到下面的信息：

```
LPT1:not rerouted.
```

总之，如果用户手中的唯一打印机是串行打印机，并且连接到 COM2 口上，就应当在 AUTOEXEC.BAT 文件中插入下述命令，这样才能使用 DOS 的打印功能：

```
MODE COM2 BAUD=96 PARITY=N DATA=8 STOP=1 RETRY=P
MODE LPT1=COM2
```

50.4.3　查看重定向状态

不带参数的 MODE 命令将显示它所控制的所有设备的状态信息。另外，当 MODE 后面紧跟一设备名而没有其他参数时，它便显示指定设备的状态信息，请看下面的例子：

MODE CON	显示监视器的状态信息
MODE COM1	显示有关第一串行口的状态信息

但是，下述命令不会显示指定的并行打印机的状态信息，它将取消到打印机的重定向：

```
MODE LPT1
```

要查看并行打印机的状态信息，必须在命令中写上/STATUS 选项(可以缩写为/STA)，例如：

```
C:\>MODE LPT1 /STA
Status for device LPT1:
------
LPT1: rerouted to COM1:
Retry=NONE
Code page operation not supported on this device
```

/STATUS 选项也可与其他设备名一起使用，但不是必需的。

50.5　PRINT 命令

我们已经讲述过如何用 PRINT 命令打印 ASCII 文本文件。回忆一下在启动过程中装

入 PRINT TSR 以后怎样执行 PRINT 命令，怎样建立打印队列及指明打印设备。

50.5.1 配置 PRINT

如果要通过 PRINT 命令执行大量的打印任务，则可能希望在初始化命令中配置其他的打印功能(如图 50.3 所示)。用户可以控制这些特性：打印缓冲区大小、打印队列大小以及打印处理(在后台)和其他工作(在前台)之间的平衡等。

注释： 只能在初始化命令中对 PRINT 作配置，如果不重新启动计算机，便不能更改其配置。

选项/B 设置打印缓冲区的大小，缓冲区越大，打印速度越快，但会减小可用内存空间。不要忘记可以用 LH 命令把 PRINT、打印缓冲区及打印队列装入高端内存。

提示： 如果高端内存太拥挤，可把 PRINT 命令——包括拟定的缓冲区大小及打印队列大小等参数——插到 AUTOEXEC.BAT 文件中，并且运行 MEMMAKER 程序，为 PRINT 在高端内存内寻找一最佳位置。如果 MEMMAKER 不能为它在高端内存区挤出空间，那么将缓冲区及打印队列缩小后再试一次。

其余的配置选项用于控制当 PRINT 在后台运行时，DOS 如何分割时间。一个时钟周期约为 1/18 秒。正常情况下，如果在一个时钟周期内打印机不处于可使用状态，DOS 则会退出 PRINT 命令，但如果用户想增加一些时间，可以使用/U 选项。/M 选项说明在打印一个字符时允许用掉的时钟周期数。如果用户的打印机非常慢而且 PRINT 经常放弃执行并显示错误信息，就可以用该选项指定更大的时钟周期数。/S 选项说明在 DOS 执行前台任务之前必须完成的后台打印工作量。如果用户设置的值大于缺省值 8，则打印速度加快，但前台工作速度减慢。

PRINT

管理后台打印。

句法：

PRINT [/D:device] [/B:buffer] [/U:ticks1] [/M:ticks2][/S:ticks3] [/Q:qsize] [/T] [filespec...] [/C] [/P]

参数和选项：

none	显示当前打印队列或装入 TSR。
/D:device	说明打印机所接的端口。有效值是 LPT1、LPT2、LPT3、COM1、COM2、COM3 及 COM4。LPT1 和 PRN 相同。如果使用该参数，/D 必须放在命令行中的 filespec 之前。
/B:buffer	设置打印缓冲区的大小(以字节数计)。其值可以是从 512 到 16384，缺省值是 512。
/U:ticks1	说明等待打印机准备好的最大时钟周期数。其值可以是从 1 到 255，缺省值为 1。如果在 ticks1 个时钟周期内打印机仍不可用，则将放弃打印工作(每秒约 18 个时钟周期)。
/M:ticks2	说明在打印机上打印一个字符要用的最大时钟周期数，可以是从 1 到 55，缺省值是 2。如果打印一个字符所用的时间长于 ticks2 个时钟周期，DOS 将显示错误信息。
/S:ticks3	说明在回到前台工作之前，DOS 分配给后台打印的时钟周期数。其值可以是 1 到 255，缺省值是 8。

/Q:qsize	指明打印队列中的最大文件数目，可以是从 4 到 32，缺省值是 10。
/T	清除打印队列中的所有文件。即停止打印。
filespec	指定要放到打印队列中的文本文件的路径和名称，最长可为 64 个字符。
/C	从打印队列中清除文件。
/P	把文件加到打印队列中。

注释：

/T 参数将清除整个打印队列。如果/T 前面有文件说明，则该文件便从队列中清除。如果/T 后面跟有文件说明，则该文件到底是被加到队列中去还是被清除，将取决于是否对它使用了/P 或/C 参数。

图 50.3　PRINT 命令总结

50.5.2　控制打印队列

在正常情况下，一个打印队列最多可有 10 个文件。每个新增加的文件都被放到队列的最后，如果原队列是空的，则该文件立即被打印。可以在队列中写上几个由空格隔开的文件名，或者用通配符在同一时间给几个文件排队。当队列满后，DOS 会给出警告信息。

/Q 选项属于初始化命令，它设置打印队列的大小。如果用户需要经常对多于 10 个文件排队，就可能希望队列更大些。另一方面，如果需要尽可能地节约常规内存，而队列中的每一项要用去 64 字节，这样，如果使队列大小设为最小值 4，就可以节约 384 字节的内存。

键入不带参数的 PRINT 命令，可以查看当前的打印队列。使用/T 选项，但不带任何文件名，能够立即中止打印队列，而且正在打印的文件也会停止打印。

试练习使用最小的打印队列(如果是通过 AUTOEXEC.BAT 装入 PRINT 的，便不要做下面的练习)：

1. 如果已经装入了 Print，退出所有程序并重新启动，以清除内存中的 PRINT。
2. 键入 LH PRINT /Q:4 命令，装入 PRINT，设置打印队列大小为 4 个文件。
3. 键入打印设备的名字(如果是 PRN，仅按 Enter 即可)。
4. 转到 DOS 目录下。
5. 键入 PRINT *.TXT 命令。

   前面的四个 *.TXT 文件便被装入打印队列中，同时显示 PRINT quene is full 信息。
6. 键入 PRINT 命令，查看 PRINT 队列的状态。
7. 键入 PRINT/T 命令，清除打印队列。

答疑： 在键入 PRINT 命令后，并没有文件被打印。

试键入不带参数的 PRINT 命令以查看打印队列的状态。在状态报告中，PRINT 也会给出错误信息。

注释： 当键入/T 后，PRINT 停止向打印机传送数据，但打印机继续打印，直到它自己的缓冲区打印完为止。如果打印机自带的缓冲区很大，在停止打印之前有可能会打印出好几页。不过可以关掉打印机，以使它停止并清除其缓冲区中的内容。

要从打印队列中删除一个文件，可以键入带参数/C 的 PRINT 命令，要删除的文件的名

字放在/C 的前面和后面均可。所有在/C 后面的文件都将从队列中删除，刚好在/C 前边的文件也会被删除。要在同一命令中把文件加到队列中去，就使用/P 参数。和/C 一样，/P 对位于它之后的所有文件和刚好在它前面的一个文件起作用。如果 PRINT 命令中的文件名既不带/P 也不带/C 参数，则这些文件被加到打印队列中去。

用户不能直接改变队列中的文件顺序。如果要使排在后面的文件先被打印，就必须清除在它之前的所有文件，然后再把这些文件重新加到队列中去。

试练习清除队列中的一个文件：

1. 键入 PRINT *.TXT 命令，放四个文件到打印队列中去。
2. 键入 PRINT 命令，查看队列中的文件次序。
   注意 README.TXT 不是处在队列中第一的位置上。
3. 键入 PRINT /T README.TXT *.TXT 命令，将 README.TXT 移到打印队列的最前面(这个命令先用/T 清除打印队列，再把 README.TXT 加到队列中去，然后再把其他 .TXT 文件加到队列中)。
4. 键入 PRINT 命令，查看当前打印队列的状态。
   现在可以看到 README.TXT 正在被打印。
5. 键入 PRINT /T 命令，清除打印队列。

50.6 打印图形屏幕

在正常情况下，DOS 不能打印 CGA、EGA 和 VGA 监视器的图形屏幕，它可能会把这些图形当作文本字符来处理。用户可以用其他一些的软件来处理、打印甚至改动图形屏幕，但有时也希望能用 PrintScreen 键打印 CGA、EGA 或 VGA 监视器屏幕上的图形。GRAPHICS 命令能完成这项任务。DOS 不能处理 BIOS 不支持的视频模式，例如，DOS 不能处理很多 SuperVGA 图形。一些应用程序把 VGA 显示设为别的模式，这些模式不能在 DOS 下打印。也不能通过 DOS 打印 Windows 屏幕。

注释： *在 Windows 下，PrintScreen 键把当前屏幕的拷贝存储在 Windows 剪贴板上，用户可以把这些内容传递给和剪贴板及图形图象兼容的任何应用程序。*

50.6.1 GRAPHICS 是如何工作的

GRAPHICS 装入一个 TSR，以使 DOS 能在几种打印机上打印它所支持的图形屏幕。当装入 GRAPHICS 以后，按 PrintScreen 键所产生的请求信号将被 DOS 分析。如果屏幕是文本方式，则该请求被传送给 BIOS 的标准文本屏幕打印。如果屏幕处于图形方式，就由 GRAPHICS 处理。它扫描屏幕上的每个象素，并把它们转换给打印机，将它们打印出来。象素的数目(屏幕分辨率)和每英寸打印点的数目(打印机分辨率)都影响打印结果。为获得最好的效果，一些图形显示按列进行打印。

在 DOS 目录下有一个名字为 GRAPHICS.PRO 的文件，该文件中包含了 DOS 所支持的打印机类型的大致情况，以及有关转换不同显示方式的细节。这是个 ASCII 文件，可以用 EDIT 对它作检查或用 PRINT 命令打印出来。

50.6.2　GRAPHICS 命令

图 50.4 示出了 GRAPHICS 命令。不带参数的 GRAPHICS 命令和 GRAPHICS HPDEFAULT 相同；运行该命令后，DOS 便准备在认识 PCL 命令(page control language 页面控制语言)的打印机上打印图形屏幕。PCL 命令被大多数 Hewlett-Packard(惠普)打印机所使用。表 50.1 列出了一个完整的可用打印机类型参数的清单。

表 50.1　GRAPHICS 打印机类型

类型	包括的打印机
COLOR1	黑色带的 IBM PC 彩色打印机
COLOR4	RGB 色带的 IBM PC 彩色打印机
COLOR8	CMY 色带(青兰、紫红、黄色)的 IBM PC 彩色打印机
DESKJET	Hewlett-Packard DeskJet 打印机
HPDEFAULT	任何 PCL HP 打印机(缺省值)
GRAPHICS	IBM 个人图形打印机、Proprinter 或 Quietwriter 打印机，也包括大多数 Epson 点阵打印机
GRAPHICSWIDE	任何盒式图形打印机
LASERJET	HP LaserJet 打印机
LASEJET Ⅱ	HP LaserJet Ⅱ 打印机
PAINTJET	HP PaintJet 打印机
QUIETJET	HP QuietJet 打印机
QUIETJETPLUS	HP QuietJet plus 打印机
RUGGEDWRITER	HP RuggedWriter 打印机
RUGGEDWRITERWIDE	HP Ruggedwriter wide 打印机
THERMAL	IBM PC 可转换型 Thermal 打印机
THINKJET	HP ThinkJet 打印机

如果 GRAPHICS.PRO 不和 GRAPHICS.COM 位于同一目录下而且也不位于当前目录中，则必须使用 filespec 参数，告诉 DOS 它的存放地点。

试练习用 DOS Shell 在打印机上打印图形屏幕。(如果读者的监视器不能显示处于图形方式下的 Shell 屏幕，那么跳过该练习。)

1. 从表 50.1 中找出最接近于自己打印机的打印机类型。
2. 键入 GRAPHICS type /R 命令。
3. 启动 DOS Shell。
4. 按 Print Screen 键或 Shift+PrintScreen 键。
   DOS 将打印 DOS Shell 的屏幕。如果不打印或打印一部分便中止了，则图形方式可能和打印机不兼容。

50.6.3　GRAPHICS 选项

在正常情况下，屏幕均以白色背景被打印，这对打印机来说是容易做到的，因为仅仅忽略背景就行。但典型的 DOS 屏幕都是以黑色为背景显示的。如果想采用黑色背景打印屏幕，

可以使用/R 选项。这个打印结果和屏幕显示更为相似(但很多打印机在黑色背景上不能很好地完成打印工作)。

GRAPHICS

装入图形 TSR,并使 DOS 能够打印图形屏幕。

句法:

GRAPHICS [type] [filespec] [/R] [/B] [/LCD] [/PRINTBOX:size]

参数和选项:

none	装入 Graphics TSR,将使用缺省值。
type	说明打印机类型。缺省值是 HPDEFAULT。
filespec	指定打印机的信息文件,缺省值是 GRAPHICS. PRO。
/R	说明打印出来的图形黑白部分和屏幕相同(正片,正图象)。缺省值是反白(或负图象,反相)。
/B	对 COLOR4 或 COLOR8 打印机适用,指示要打印出的背景颜色。
/LCD	用 LCD 象素比例的方式打印图象,该选项与/PRINTBOX:LCD 的效果相同。
/RPINTBOX:size	选择打印长宽。可以把/PRINTBOX 缩写为/PB。size 可以是 STD 或 LCD,它必须和 GRAPHICS. PRO 文件中的 PRINTBOX 语句的第一个操作数相匹配。

注释:

不能通过 DOS 在 PostScript 打印机上打印图形屏幕。

在装入 GRAPHICS 以后,可以按 Shift+PrintScreen 键或 PrintScreen 键(视键盘而言)打印屏幕。灰度阴影的数目取决于监视器和打印机。如果监视器的分辨率为 640×200 或更高,则屏幕将被打在打印机的边上。

图 50.4 GRAPHICS 命令总结

注释: 一些应用程序,例如 DOS Shell,在显示时是白底黑字(或彩色字),但打印时正好相反。如果要打印的结果和屏幕一致,则必须加上/R 选项。

当使用彩色打印机打印图形时,GRAPHICS 把白色背景作为缺省值。/B 参数告诉 DOS 使用彩色背景,但这只对 COLOR4 和 COLOR8 两种打印机有效。如果用户检查一下 GRAPHICS. PRO 文件,就会明白每种显示的色彩是如何被打印的。

50.6.4 长宽比

长宽比例是垂直和水平二维方向上的比例。LCD 监视器的长宽比例和 VGA 及其他类型监视器有所不同。如果在打印屏幕时使用了错误的长宽比,图形就会被扭曲,例如圆变成了椭圆。

在正常情况下,可以让 GRAPHICS 自动选择长宽比。在 GRAPHICS. PRO 文件中对每种打印机都给出了长宽比,亦即是 PRINTBOX 的值。如果发现实际所用的长度比不适合自己的打印机,可检查 GRAPHICS. PRO 文件,看看自己的系统是否对 LCD 和 STD 都能操作。/LCD 或/PB:LCD 说明 LCD 长宽比,而/PB:STD 说明标准长宽比。

如果要替换 GRAPHICS.PRO 文件，就必须重新启动计算机，假如新的 GRAPHICS.PRO 文件比原来的占用内存要少，它就可以覆盖在原来已装入的 GRAPHICS.RPO 之上。如果不是这样，DOS 便显示错误的信息并且必须重新启动。

提示： 在为自己的系统确定了正确的 GRAPHICS 命令以后，可通过 AUTOEXEC.BAT 将其装入，这样该命令会一直生效。假如不能把它装入高端内存；而且也不能让它总驻留于常规内存中，可以写一个批文件或 DOSKEY 宏命令，在需要时才将 GRAPHICS 装入内存。

50.7　自 测 练 习

本章讲了如何使用 DOS 完成打印工作。下面给出一些练习以帮助读者熟悉这些技术：

1. 启动回显打印功能，并键入几个 DOS 命令。再关闭回显打印功能。
2. 打印 DOS 命令提示屏幕。
3. 如果自己的打印机和 GRAPHICS 命令兼容，那么练习打印 DOS Shell 屏幕（或其他 GUI 屏幕）。
4. 改变屏幕方式，使每屏显示的行数为最大值。
5. 改变屏幕方式，使每行显示 40 个字符。
6. 装入 PRINT 命令，并为它建立一很大的缓冲区及 20 个文件的打印队列。练习打印几个文本文件，然后清除整个打印队列。

第 51 章 配置磁盘和网络

简介： 与 CMOS 和 ROM-BIOS 一起工作，DOS 能够非常完美地完成磁盘操作。但是，如果出现了一些问题，就必须给 DOS 一些特殊的帮助。DRIVPARM 命令和 DRIVER.SYS 设备驱动程序使用户能够重定义磁盘驱动器。在网络上运行 DOS 也需要作一些特殊的考虑。

51.1 为 DOS 定义驱动器

在 PC 机的 CMOS 集成块里保存着有关系统驱动器的物理描述，在系统启动过程中，首先就得访问 CMOS 中的内容。在 ROM-BIOS 中存储的是有关驱动器读写的低级软件(同时也有控制其他设备的程序)。当 DOS 启动时，它利用 CMOS 和 ROM-BIOS 提供的这些功能访问系统的驱动器。

在极少的情况下，PC 机有一个在 CMOS 中不能正确定义的驱动器，或者是不被 ROM-BIOS 所支持的驱动器，此时，用户就不得不用 DRIVPARM 或 DRIVER.SYS 驱动程序为 DOS 定义这个驱动器了。用户也可以利用 DRIVER.SYS 为软盘驱动器建立额外的逻辑驱动器名字，以使得拷贝和移动文件更加方便。本章将讲述在什么时候以及如何使用这些配置命令。

要把 PC 机接到网络上去，要求硬件和软件都要和网络相适应。网络操作系统能够满足大多数软件的需要，但是，当在网络上运行 DOS 时，用户必须考虑 DOS 的一些特点。本章亦将阐述这方面的内容。

本章将讲述下列内容：

- 使用 DRIVPARM 定义块设备，如磁盘驱动器或磁带驱动器
- 用 DRIVER.SYS 配置一个逻辑软盘驱动器
- 为软盘驱动器分配另一个名字
- 弄清楚在一个网络上安装 DOS 6.22 时，什么时候需要额外的信息
- 明确不能在网络上使用的 DOS 命令
- 装入 SHARE 以保护网络中的文件

51.2 配置物理磁盘

DRIVPARM 命令应加入在 CONFIG.SYS 文件中。当 DOS 在没有额外的帮助信息就不能正确地访问一个设备时，使用 DRIVPARM 来定义块设备(如图 51.1 所示)。所谓块设备，是指在同一次读和写都超过一个字符的设备，比如软盘驱动器、硬盘驱动器和磁带机。

DRIVPARM

为一个已经存在的物理驱动器说明新的参数。

句法：

DRIVPARM /D:drive [switches]

选项：

/D:drive　　指明物理驱动器编号：0=A；1=B；2=C 等等。该选项是必需的。

/F:type　　指明驱动器类型。有效值如下：

0　360KB 或更小

1　1.2MB

2　720KB(缺省值)

5　硬盘

6　磁带机

7　1.44MB

8　读/写光盘

9　2.88MB

/H:heads　　指明磁头数。

/S:sectors　　指明每磁道扇区数。

/T:tracks　　指明磁道数。

/C　　指定驱动器有换线(change-line)支持(对可移动媒体而言——译者注)。

/N　　指定不可移动驱动器。

/I　　指明系统不支持的 3 ½软盘驱动器。

图 51.1　DRIVPARM 配置命令总结

一般情况下，如果一个块设备需要用 DRIVPARM 进行定义，则制造商会提供详细的信息。用户所要做的就是把它加到 CONFIG.SYS 文件中去。

必须用 DRIVPARM 重定义块设备的一个原因是该设备没有使用缺省的磁头数、扇区数和磁道数。表 51.1 示出了 DOS 支持的磁盘类型的缺省参数。如果用户所安装的驱动器没有采用这些缺省值，则要在设备档案中指出正确的参数值。如果用户所指出的值是错误的或者不全相容，DOS 也可能从该驱动器上读数据，但在写磁盘时就会出现不稳定或写上错误的格式。

表 51.1　标准软盘参数

容量	磁头数	扇区数	磁道数
160KB	1	8	40
180KB	1	9	40
320KB	2	8	40
360KB	2	9	40
720KB	2	9	80
1.2MB	2	15	80
1.44MB	2	18	80
2.88MB	2	36	80

警告： 如果在 DRIVPARM 命令中对启动加以说明，而且所给的参数是错误的，则有可能使系统不能启动，或者可能像平常一样完成启动但当它在写磁盘时会将根目录破坏。因此，在重定义启动盘之前必须确保手中有备份磁盘。也可以按 F8 键跳过 DRIVPARM 而完成启动，但如果已经破坏了根目录，就必须从备份中恢复后才能继续下去。

假如用户要把一个 1.44MB 的磁盘驱动器作为 A 驱动器安装，该驱动器有 2 个磁头，每磁道 20 扇区，共 72 个磁道，并且有 change-line 支持，则必须在 CONFIG.SYS 文件中插入下述命令：

```
DRIVPARM /D:0 /F:7 /C /S:20 /T:72
```

假如要把一个 2.88MB 的驱动器当作 B 驱动器安装，而 DOS 却不认识这个新的驱动器，则需要在 CONFIG.SYS 这件中插入下述命令：

```
DRIVPARM /D:1 /F:9 /C
```

该命令为驱动器 1(B 驱动器)设置参数以使用驱动器类型 9(2.88MB)的缺省值，而且说明该驱动器有换线(change-line)支持。change-line 支持将在下一部分给予解释。

51.2.1 change-line(换线)支持

change-line(换线)是让 DOS 检查一个驱动器的门是否开启，当替换磁盘时给出一个信号。很多 360KB 或容量更低的磁盘驱动器不具有此功能。

如果 DOS 认为一个驱动器具有 change-line 支持，那么在它需要访问一个还没打开的驱动器时，它会直接去缓存区内读取目录而不是从磁盘上读目录，这样便节约了时间，因为目录被存于高速缓存区内。如果驱动器已经打开了，DOS 便会再读驱动器的目录结构并把它放到缓存区去。

功能差的 change-line 装置会威胁软盘上数据的完整性。如果 DOS 用另一种软盘的目录结构去访问这一张软盘，则有可能从错误的簇中读取数据或将数据写到错误的簇中去。如果用户不希望发生这种情况，可以修复或替换该驱动器，或者用 DRIVPARM 告诉 DOS 该驱动器没有 change-line 支持。

51.3 配置逻辑软盘

在一个物理驱动器上，DRIVER.SYS 设备驱动程序可以建立一个逻辑软盘。它将给该逻辑软盘分配一个新的驱动器字母，该字母是原驱动器名序列的下一个。DRIVER.SYS 的两种主要用途如下：

- 强迫 DOS 为一外部驱动器分配一逻辑驱动器名，并能识别该逻辑驱动器。
- 为一磁盘驱动器分配一个可替代的逻辑驱动器名字。

注释： 要为一个硬盘建立逻辑驱动器名，可使用 SUBST 命令，该命令在第 39 章“目录管理”中作了解释。

图 51.2 示出了和 DRIVER.SYS 驱动程序一起使用的选项。注意既可以用 DEVICE 也可以用 DEVICEHIGH 将 DRIVER.SYS 装入内存。亦可用多个 DRIVER.SYS 命令来定义多个逻辑驱动器。每个装入内存的驱动程序将占用 240 字节的内存空间。这个命令的最基本的用途是为驱动器分配替代名字，有关解释可参见下一节。

DRIVER.SYS

在一物理的磁盘驱动器上建立新的逻辑驱动器。

句法：

DEVICE[HIGH]=[path] DRIVER.SYS /D:drive [switches]

参数和选项：

path	如果 DRIVER.SYS 文件不是处于启动驱动器的根目录下，用此选项说明 DRIVER.SYS 的路径。
/D:drive	指定物理软盘驱动器的序号：0＝第一个驱动器；1＝第二个驱动器；最大可为 127。
/F:type	指定驱动器类型。
0	360KB 或更小
1	1.2MB
2	720KB(缺省值)
7	1.44MB
9	2.88MB
/H:heads	指定磁头数。
/S:sectors	指定每道的扇区数。
/T:tracks	指定磁道数。
/C	指定该驱动器具有 change-line(换线)支持。

图 51.2　DRIVER.SYS 设备驱动程序总结

51.3.1　建立一个替代驱动器名字

现在很多计算机都配有两种不同的软盘驱动器，例如 1.2MB 和 1.44MB。用户不能从一个磁盘上直接将文件移到或拷贝到另一个相同类型的磁盘上，因为 DOS 拒绝执行下述命令：

```
XCOPY B:\*.* B:
```

但可以使用 DRIVER.SYS 为一个物理驱动器再分配第二个逻辑驱动器名，这样用户就可以使用 MORE、COPY 和 XCOPY 命令在同一个驱动器中将一个磁盘上的文件移动或拷贝到另外一个磁盘上去。下面这个命令为驱动器 B(1.44MB 容量)建立第二个驱动器字母代号：

```
DEVICE=C:\DOS\DRIVER.SYS /D:1 /F:7
```

一定要注意在启动过程中 DOS 分配给该驱动器的字母代号是什么。当驱动程序被装入时，会出现诸如以下的信息：

```
loaded External Driver for Drive E
```

该信息告诉用户驱动器 B 的替代名称是 E。然后用户就可以用下述命令，使用刚建立的逻辑驱动器拷贝文件了：

```
COPY B:\*.* E:
```

DOS 将提示用户一次插入一张磁盘(首先插入目标盘，以便 DOS 进行检验以确认无误)。屏幕上将显示如下的信息：

```
C:\>XCOPY B:*.* D:
Insert diskette for drive D:and press any key when ready
Insert diskette for drive B:and press any key when ready
Reading source file(s)...
Insert disk for drive D: and press any key when ready
B:CHAP02.CHP
...
B:APPP.CHP
Insert diskette for drive B: and press any key when ready
        7 File(s) copied
Insert disk for drive D: and press any key when ready
```

提示： 当拷贝的文件不止一个时，最好使用 XCOPY。因为在写文件之前，XCOPY 先把文件读到内存中去，直到内存装满为止，这样就会减少来回换盘的动作；与此相反，COPY 是单个地处理文件。

51.3.2 建立两个替代驱动器名

假如计算机配有两个软盘驱动器，容量分别为 1.2MB 和 1.44MB，同时也配有两个硬盘驱动器，则可以在 CONFIG.SYS 文件中写入下述命令！

```
DEVICE=C:\DOS\DRIVER.SYS /D:0 /F:1
DEVICE=C:\DOS\DRIVER.SYS /D:1 /F:7
```

当系统启动时，屏幕上将出现下列信息：

```
Loaded External Disk Driver for Drive E
Loaded External Disk Driver for Drive F
```

在 DOS 工作过程中，如果需要，用户可以随意使用驱动器 E 和 F，这和使用 A 和 B 驱动器的结果完全一样。最重要的是，用户可以把文件从驱动器 A 拷贝到 E 上(E 和 A 是同一物理驱动器)，以及从驱动器 B 拷到驱动器 F 上。

注释： DOS 保留最后使用的逻辑驱动器名。例如，如果 E 是当前使用的驱动器，而键入 DIR A:命令，则会显示下列信息：

```
,Insert diskette for drive A:and press any key when ready
```

用户亦可以随意切换驱动器名。例如，如果驱动器 E 是当前驱动器，则可键入 A:，把当前驱动器切换到 A 上。

51.3.3 增加一个外部软盘驱动器

如果用户要安装一个 DOS 不能自动识别的驱动器，则可以用 DRIVPARM 或者 DRIVER.SYS 告诉 DOS 在 PC 机上安装了这个新的驱动器。例如，下面的命令告诉 DOS 安装了一个 1.2MB 的外部软盘驱动器，该驱动器是第三软驱：

```
DEVICE=C:\DOS\DRIVER.SYS /D:2 /F:1 /C
```

如果用户希望能在该驱动器上进行文件拷贝，则需要再写入一个相同的 DEVICE 命令，以建立第二个逻辑驱动器。

51.3.4　选择使用 DRIVPARM 或 DRIVER.SYS

如果使用 DRIVER.SYS 去改变一物理驱动器的参数，它会分配给该驱动器一个新的逻辑驱动器名。相比之下，使用 DRIVPARM 重定义驱动器要方便些，它使用户能继续使用该驱动器原来的名字。同时，DRIVPARM 所占用的内存空间也比 DRIVER.SYS 少。

答疑： 当安装好一个新的(或原来的)软盘驱动器后，DOS 不能识别，其原因是什么？

如果用户增加了一个 BIOS 不支持的软盘驱动器，则为它所提供的驱动程序可能不能工作。例如，假如用户有一个 2.88MB 的驱动器要作为驱动器 A 安装，但计算机的 BIOS 不支持这个驱动器(很多旧版本的 BIOS 都不支持)，作为第一步，须要在 CONFIG.SYS 文件中加上下述 DRIVPARM 命令：

```
DRIVPARM /D:0 /F:9 /C
```

如果仍然不能访问该驱动器，那么试着用 DRIVER.SYS 对它进行定义，然后有两个逻辑驱动器名对应着同一个软盘驱动器。使用的配置命令如下：

```
DEVICE=C:\DOS\DRIVER.SYS /D:0 /F:9 /C
```

因此，有可能同时需要 DRIVPARM 及 DRIVER.SYS 两个命令才能完成安装。

51.4　在网络上运行 DOS

网络是指两台或多台计算机通过硬件和软件彼此连接起来的集合，连接的方法随网络不同而有根本的区别。但是，硬件总是包括位于每台计算机中的网络适配器卡(简称网卡)及连接计算机的缆线。软件包括网络操作系统和文件服务器，文件服务器管理对文件的访问。网络软件也可能包括控制打印机及通信的其他服务器。

网络中的每台计算机称为工作站点或网络节点。工作站点的范围很广，可以是没有其他本地资源(如磁盘或打印机)而只有键盘和监视器的 PC 机，也可以是装备很完整的 PC 机，除了连在网络上外，它还有丰富的局域资源(亦称本地资源)。一些 PC 网络还能操作 Macintosh 计算机。一个工作站点能够把信息和文件传送到另一个工作站点，并能访问网络驱动器上的文件(比如某部门的数据库)以及访问至少一台共享打印机。

个人用户被赋予访问网络上共享资源的部分权力。例如，Kevin 可能被允许读网络驱动器但不能写，而 Donna 既能读又能写，但不能重新格式化网络驱动器。

很多工作站点能够独立于网络工作。在它们自己的驱动器上使用大多数 DOS 命令。(当然，对于没有磁盘驱动器的工作站点来说，它们只能访问网络驱动器)。当 DOS 程序和数据存储在工作站点自己的驱动器上时，DOS 运行过程及所有命令的执行和不在网络上是一样的。但是，在传输数据时，时间就变成了一个重要的因素，特别是在几个用户同时工作时尤其如此。

当打开计算机时，特殊的网络命令就建立起网络系统。网络操作系统有它自己的启动文件，这些文件有些像 AUTOEXEC.BAT，当启动网络时启动文件被自动执行。一些 DOS 命令，比如 JOIN 和 SUBST，可以用来为网络建立必要的访问路径。虽然这些命令是有效的，但除了在建立网络时必须使用外，不能在网络上随便使用。否则它们会给 DOS 运行带来不必要的障碍。

51.4.1 把网络操作系统升级为 DOS 6.22

如果用户所使用的 DOS 版本比 DOS 5.0 高，而且在该版本下网络能够正常工作，那么，把操作系统更新为 DOS 6.22，这不会影响网络。如果用户当前所使用的 DOS 版本较旧，则在升级过程中会出现问题，问题的大小取决于网络软件版本旧的程度。

DOS 6.22 升级软件包中有一个名为 NETWORKS.TXT 的文件，该文件详细地说明了几种网络系统。如果系统已经连网，在升级至 DOS 6.22 之前一定要检查一下这个文件。对于很多系统，该文件都包含了在装入新版本 DOS 之前必须执行的命令。用户可以从软盘上显示或读出（甚至打印出）NETWORKS.TXT 文件，而不是一定要装上 DOS 6.22 后才对它进行访问。

51.4.2 DOS 命令的局限

大多数 DOS 命令在网络上的工作情况和在独立计算机上的工作情况是一致的。但是，一些 DOS 命令不能在网络上使用，另一些命令不能在网络驱动器上使用。表 51.2 列出了这些命令。

表 51.2 DOS 命令的网络限制

不能在网络驱动器上使用下述命令：
CHKDSK
DEFRAG
DISKCOMP
FDISK
RECOVER
SYS
不能在网络上使用下述命令：
FASTOPEN
FORMAT
LABEL
UNFORMAT

不能在网络驱动器上使用的命令仍然可以在连网计算机自己的驱动器上使用。任何不能在网络上使用的命令，只能在计算机自己的驱动器上使用。

如果用户的系统上有令牌环（Token-ring）网络，并且在运行 MEMMAKER 时出现问题，便可以在 MEMMAKER 命令后面加上/T 选项，以解除对 Token-ring 网卡的检测。

51.4.3　共享和锁定文件

在网络上工作时，很可能出现两个用户同时访问同一个文件的情况。他们甚至同时对同一个文件作一些改动。例如，几个用户都试图更新同一个部门数据库。网络软件应该防止出现这类情况下的冲突。一些网络操作系统使用 DOS 的 SHARE 命令来解决这类冲突。图 51.3 示出了 SHARE 命令的句法。

Windows OLE 2.0 应用程序也可能需要 SHARE 支持，因为有些时候会有不止一个应用程序同时通过 OLE 2.0 访问同一个文件。一些 Windows 应用程序也需要 SHARE，以便能打开应用程序的多个实例。例如，只要装入了 SHARE，就可以在多个窗口中打开 Word for Windows。

SHARE 命令指定为存储文件说明保留的空间大小，以及一次能被锁定的文件的最大数目。如果网络需要 SHARE，在安装网络时，可以把 SHARE 放到 AUTOEXEC.BAT 文件中。

注释： Windows for Workgroups 自己拥有 SHARE 命令，因此不需要用户自己装入。

SHARE

在硬盘上安装文件锁定和共享功能，通常在网络上或多任务环境中使用。在安装 SHARE 以后，它使得所有读写要求生效。

句法：

从命令提示符下键入时：

SHARE [switches]

从 CONFIG.SYS 文件中装入时：

INSTALL=[path] SHARE.EXE [Switches]

参数和选项：

path	如果 SHARE.EXE 不在启动驱动器的根目录下，该选项指明其路径。
/F:space	指定为了共享文件所需的内存空间大小(以字节数计)，以记录为实现文件共享必须保存的信息。缺省值是 2 048 字节。
/L:locks	指定一次可以锁定的文件数目。缺省值为 20。

图 51.3　SHARE 命令总结

51.5　自 测 练 习

不必对本章所讲的技巧都要求做练习。例如，如果需要重定义一个驱动器，便在启动文件中插入一相应的命令。但是，如果用户有一个 3 ½英寸或 5 ¼英寸的软盘驱动器，便可以练习为该驱动器分配第二个驱动器名，并在这个驱动器上实现两张软盘间的拷贝和移动操作。

第 52 章 多媒体 PC 机

简介： *多媒体——90 年代成人的玩具和工具。但是什么是多媒体以及怎样利用当前流行的 CD 软件提供的财富呢？*

52.1 多媒体的新概念

回想 70 年代(PC 标准化以前)，笔者曾采用多媒体方式授课。每堂课都从打印出来的内容介绍开始，然后是 10 分钟电视，接下来是一系列打印出来的授课内容。一些课程还包括 10 分钟的听，内容当然是几堂课之一。这就是多媒体。多媒体包含至少两种甚至三种通信方法。

相比之下，今天的在 PC 机上实现的多媒体给使用者一个完全不同的感受。在今天的多媒体中，有用于教育的文字、声音、图象、照片、电视以及其他生动的内容，也有信息、对人的规劝和娱乐。使用者可以对这些内容随意控制。例如，Microsoft 的 Cinemania，它建立了几千部电影及其制片人的档案。用户可以从查寻脚本开始，查看有关的评论，看一看它所得的荣誉和奖品，欣赏电影剪集(带声音)，通过选择 Dustin Hoffman 的名字而看他的传记，详细阅读一部黑色喜剧，再看看 Hoffman 制作的电影胶片，观看一部电影等等。

本章将介绍多媒体 PC 机以及为了和多媒体的硬件和软件交互所需用的 DOS 命令。这些命令不多，因为大多数多媒体程序都在 Windows 下而不是在 DOS 下运行。本章的内容有：

- 在多媒体 PC 机中通用的硬件是什么
- 需要为多媒体 PC 机安装哪些软件
- 怎样装入 DOS 的 MSCDEX CD-ROM 驱动程序
- 怎样使用 DOS 的 MSD(Microsoft Diagnostic，诊断程序)程序为多媒体硬件设置中断(IRQ)

52.2 多媒体硬件

用户可以凑合着使用 386SX 及廉价(单速)的 CD-ROM 驱动器，甚至可以不使用声音卡，而装入名为 SPEAK.EXE 的免费程序，它能使计算机的扬声器发出声响。但这样的效果如同观看一部早期的电影——美国口语中称 flickers——一样，不如人意。

The Multimedia PC Marketing Council,Inc，已经为 MPC(多媒体 PC 机)定义了工业标准，已升级的标准称为 MPC2，其中包括：

- 486SX 25MHz 处理器
- 160MB 硬盘

- 4MB RAM(内存)
- 双速 CD-ROM 驱动器
- Super VGA 640×480 显示卡,带 64KB 色彩
- 1.44MB 软盘
- 游戏杆
- 16 位声音板,带 8 声阶合成器和 MIDI 回放功能(也需要扬声器)

凡有 MPC2 标志的软件都假定用户的 PC 有这些设备,但事实上这并不够。要从 MPC 软件的视频部分得到最佳效果,则需要更快的处理器(至少是 486DX),一个三倍速度的 CD-ROM 驱动器,以及一台高速度、高分辨率的显示器。要得到最好的听觉效果,则需要使用比声音卡所带音箱更高档的音箱。

假如用户自己设计多媒体系统,就会发现需要更多的硬件。声音卡能让用户录下自己的声音,但需麦克风及很大的硬盘空间。平均 10 秒钟的声音信号就要占用 880KB 的磁盘空间。

如果要通过录象机或 VCR 录下图象,首先需要一个视频记录卡,同时,视频记录需要更大的硬盘空间。一个经压缩过的 10 秒钟的录象,也需要 5.5MB 的磁盘空间。而且如果要正常显示这些图象,磁盘驱动器的速度必须很快,这意味着自图象数据很可能不能采用压缩格式。压缩程序如 DriveSpace,在每次用户访问一个文件时都要对该文件进行解压缩。当用户在用字处理程序处理档案一类的文件时,不会注意到访问时间的轻微增加,但如果是显示视频图象,情况就不同了。如果用户打算制作动画,也需要速度很快的硬盘驱动器(同时也需要高质量的,有色彩扫描器的压敏数字化仪,因为用鼠标很难画光滑的曲线)。

注释: *定期地重整硬盘(即消除硬盘上的文件碎片)能够保证硬盘以尽可能快的速度运行。在第 33 章中讲述了 DOS 的 DEFRAG 程序。*

52.3 软　件

需要用驱动程序来管理 CD-ROM 驱动器、声音卡及其他 MPC 硬件。这些软件都是和它们相应的硬件一起销售的,而且附带一张安装盘,该盘负责把驱动程序装到 CONFIG.SYS 文件中。如果制造商没有提供安装程序,可以仿照本节中所讲的步骤自己安装驱动程序。

怎样安装一个新的设备驱动程序

1. 阅读硬件说明书。驱动程序总要求在 DEVICEHIGH 命令中设置一些选项。要明确自己的 PC 需要设置哪些选项(如果说明书未讲清楚,可以向制造商询问或者做一些试验,以确定适当的设置)。
2. 在一个 ASCII 文本编辑器如 EDIT 中打开 CONFIG.SYS 文件以准备编辑。
3. 在 EMM386.EXE 和 DOS=UMB 命令之后,为驱动程序插入一条 DEVICEHIGH 命令。例如下述命令:

```
DEVICEHIGH=path\filespec parameters and switches
```

4. 对于这次要装入的所有驱动程序，都重复第 3 步，写入相应的命令。
5. 存储这个文件并退出编辑器。
6. 重新启动计算机以执行修改过的 CONFIG. SYS。
7. 如果 DOS 不能完成启动过程(当装入了一个新的设备驱动程序后，这种情况是可能发生的)，则重新启动并按 F8 键，使得一次执行 CONFIG. SYS 中的一行命令。跳过新加入的 DEVICEHIGH 命令以完成启动过程，然后回到第 1 步，再重新开始。
8. 如果 DOS 启动很成功，那么试试新的设备是否工作。假如它不工作，再回到第 1 步重新尝试。
9. 如果新的设备能够正常工作，可以运行 MEMMAKER 程序，以优化高端内存(如果 MEM /C 显示设备驱动程序已被成功地装入了高端内存，就不必再运行 MEMMAKER)。

用户可能需要大量的软件，这些软件有时被称为创作软件。因为有的用户希望录制自己的声音和视频文件；制作动画；制作很复杂的图形；欣赏、处理照片；有时甚至需要把这些溶入一个游戏中或教育软件包中，也可能将这些商品化。DOS 没有提供创作这些多媒体的软件，也不提供播放这些多媒体 CD 的软件。不过 Windows 中包含了一些基本的功能：

- 用于播放音频 CD 的 Media player。
- 用来记录和播放声音(这些声音存于. WAV 文件中)的 Sound Recorder。

Windows 还提供了几个管理 MPC 的实用程序，例如可以用 Drivers 装入硬件驱动程序，用 Sound 把声音文件和正在 Windows 下运行的应用程序联系起来。例如在启动 Windows 时，计算机发出："嗒、嗒"声，而在退出时发出和谐的钟声，但用户可以随心所欲地改变这些声音(甚至可以模仿 BBS(电子公告牌系统)和联机设备发出的声音)。

多媒体程序，如 Microsoft Cinemania 或者 Compton 的 Multimedia Encylopedia，都有它们自己的播放软件。用户所要做的仅仅是把 CD 装入 CD-ROM 驱动器，然后键入正确的命令以装入(或启动)这些多媒体程序。

52.3.1 MSCDEX 程序

DOS 有一个重要的多媒体应用程序，名叫 MSCDEX。该程序使 DOS 能够访问和管理 CD-ROM 驱动器(如图 52.1 所示)。MSCDEX 必须从 DOS 命令提示符或 AUTOEXEC. BAT 文件中装入。事实上，这个用于在 CONFIG. SYS 中安装 CD-ROM 驱动程序的程序，也可用于在 AUTOEXEC. BAT 中装载正确的 MSCDEX 命令。假如不行，用户就得自己往 AUTOEXEC. BAT 中写命令了，但首先应查看一下 CONFIG. SYS 文件中该设备驱动程序的/D选项。用户必须在 MSCDEX 命令中使用同样的/D 选项，这样 DOS 才能知道是为哪个驱动器装入 MSCDEX。(如果不止一个 CD-ROM 驱动器，那么为每个驱动器都写上 MSCDEX 命令。)假定一设备驱动程序是用下面的命令装入的：

```
DEVICEHIGH=\DEV\MTMCDS.SYS /D:MSCD001 /P:300 /A:0
```

则 MSCDEX 命令如下：

```
LH C:\DOS\MSCDEX /D:MSCD001
```

MSCDEX

使 CD-ROM 驱动器能被 DOS 所用。

句法：

MSCDEX /D:signature [...] [/E] [/K] [/S] [/V] [/L:drivename] [/M:buffers]

参数和选项：

/D:signature	指定 CD-ROM 设备驱动程序的标号。
/E	让设备驱动的程序使用扩充内存。
/K	识别 Kanji(汉字)编码的 CD-ROM 磁盘。
/S	允许 MS-NET 及 Windows for Workgroups 服务器共享 CD-ROM 驱动器。
/V	当 MSCDEX 启动时显示内存状态报告。
/L:drivename	指定分配给 CD-ROM 驱动器的驱动器名字。如果命令中不止一个驱动器，只指明第一个驱动器的名字，其他的驱动器按顺序接受后续的名字。
/M:buffers	指明分配给驱动器的缓冲区数目。

图 52.1　MSCDEX 命令总结

当 DOS 装入 CD-ROM 的驱动程序时，它便给 CD-ROM 驱动器分配一个名字。查看启动过程中给出的信息就能看见这个驱动器名字。假如错过而没看见，可以启动 DOS Shell 查看一下驱动器列表；CD-ROM 驱动器有它们自己标志，所以在驱动器列表中很容易指出哪个是 CD-ROM 驱动器。

52.3.2　选择 IRQ(中断)

DOS 通过中断和 PC 机的设备发生联系。中断很像设备打给 DOS 的电话。当设备有所请求时(例如在键盘上按下了某个键)它便发出一中断信号，DOS 得到中断信号后，便对该信号作出回答并处理这个中断请求。

PC 机有 16 条硬件中断线，编号为从 IRQ0 到 IRQ15。这意味着 DOS 可以同时处理最多 16 台硬件设备。(不过有例外，一些设备可以共享同一条中断线而不至于发生冲突)。当用户装上了一个新的设备驱动程序后，它可使用尚未占用的中断线(如果幸运的话)，也可能得由用户告诉它使用哪条中断线(这就不那么幸运了)。用户可以使用 DOS 的 MSD(Microsoft Diagnostics，诊断程序)程序定位那些尚未占用的中断线。

启动没有参数的 MSD 程序，图 52.2 示出了 MSD 的初始屏幕，它是 MSD 的主菜单。选择 IRQ Status 项，就可以看见图 52.3 所示的屏幕，它给出了 16 个 IRQ 的状态。

在这个例子中，计时器使用 IRQ0，键盘通讯使用 IRQ1，等等。

注意 IRQ5 是为 LPT2 保留的，但这个系统的 LPT2 口上没有设备，于是可以把这个 IRQ 挪给多媒体设备使用。但是，假如这样做，就不能同时在 LPT2 口上接上另一台与多媒体驱动器同时运作的设备。例如，假定用户把 CD-ROM 分配给了 IRQ5，在启动过程中，CD-ROM 设备驱动程序就会抓住 IRQ5 的控制权，这时，用户就不能通过 INTERLNK 使用 LPT2 口把计算机和另一台计算机相连接，除非用户重新启动计算机并跳过 CD-ROM 设备驱动程序的装入命令，然后才可能完成 INTERLNK 任务。但在一些特殊的系统中，不能挪

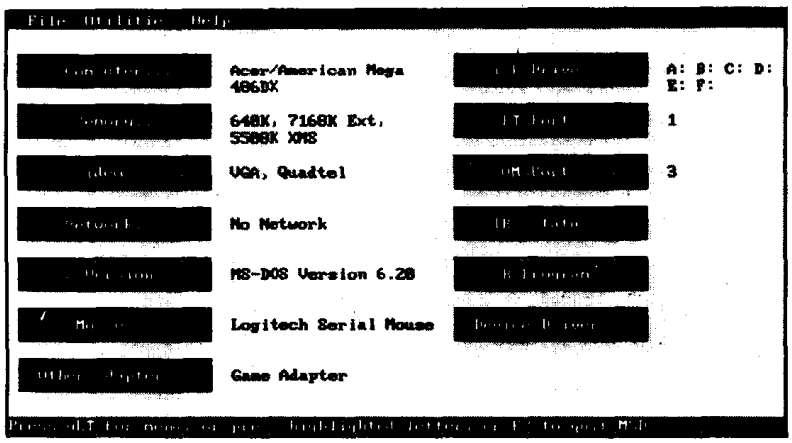

图 52.2 MSD 主屏幕(主菜单)

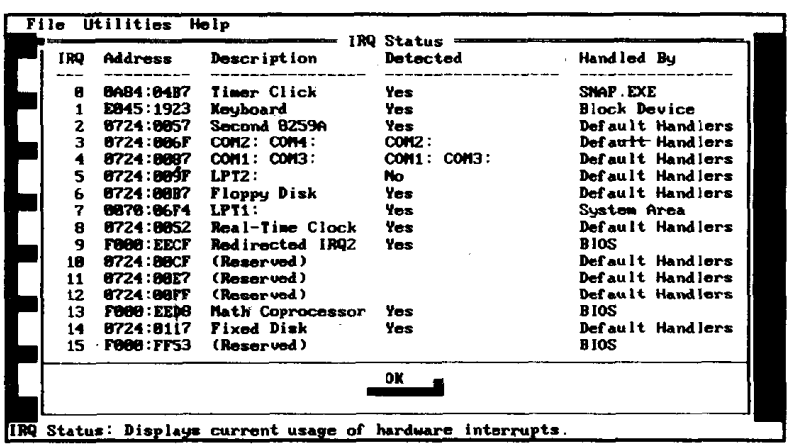

图 52.3 IRQ 状态窗口

用 IRQ5,不过可使用 IRQ10 到 IRQ12,以及 IRQ15。

怎样告诉设备使用哪条 IRQ 呢?可以查阅说明书。在安装过程中,安装程序可能会向用户询问使用哪条 IRQ,或者依照制造商所给的指南设置一跳线或双掷开关。

52.3.3 优化 CONFIG.SYS 和 AUTOEXEC.BAT

如果使用制造商提供的安装程序安装一台设备,在安装完成后,应查看一下自己系统的 CONFIG.SYS 和 AUTOEXEC.BAT 文件。明确以下几个方面:

- 设备驱动程序被装入了高端内存。如果没有,便自己改写 DEVICE 命令,或者运行 MEMMAKER 程序。
- 所有不用的驱动器程序都已被清除。
- 如果是使用多重配置,确保在所有适当的配置块中都装入了该设备驱动程序。
- PATH 命令指定了系统最佳的搜索路径。(一些安装程序把自己的目录插到搜索路径的最前面,这样做有些鲁莽)。
- 任何不用的目录路径都已从搜索路径中删除。

第 53 章　使用 QBASIC

简介： 如果用户在 PC 机上使用 BASIC 设计程序，可以使用 DOS 与推出的称为 QBASIC 的新语言环境。QBASIC 比早期使用的 BASIC 或 GW-BASIC 具有更多的功能。

53.1　QBASIC 的特色和功能

对那些设计应用程序的人们来说，BASIC 和 PC 机简直是密不可分。QBASIC 是比早期的任何版本的 BASIC（包括 GW-BASIC 和 BASICA）都要高级的语言。BASICA 和 GWBASIC 是由早期版本的 DOS 所提供的。使用 QBASIC，用户可以设计结构化的程序，这样，不仅使程序运行更为有效。而且更容易编码和调试。

QBASIC 是 Microsoft QuickBASIC 的可解释版本，它不象完整的 QuickBASIC，它没有编译器。但是，任何在 QBASIC 下设计的程序，都能在 QuickBASIC 下编译。同样，任何在 QuickBASIC 下编写的程序的源代码都能够在 QBASIC 下运行。

早期使用的 BASIC 和 QBASIC 之间存在着很多差别，主要的差别在于用户界面、数据类型、行号及一些新的语句，这些新语句让用户能够控制程序的结构化。本章的主要目的不是讲解如何使用 QBASIC 设计程序，而是向 BASIC 程序设计者介绍 QBASIC，并讨论 QBASIC 和没有结构化的 BASIC 之间的主要区别。本章的内容有：

- 使用 QBASIC 编辑器
- 把 BASIC 程序转换为 QBASIC 的存储格式
- 使用下述 QBASIC 特色：
  数组下标范围
  RECORD 数据类型
  IF 块
  CASE 结构
  DO 循环
  命名常数
  长整型数
  固定长度的字符串变量
  新的字符串功能：UCASE$，LCASE$，LTRIM$，RTRIM$
- 使用 QBASIC 编辑器编写子程序和函数
- 在子程序和函数中使用通配符和局部变量
- 使用 QBASIC 编辑器测试和调试 QBASIC 程序

本章不讲述如何设计 BASIC 程序，如果读者不是熟练的 BASIC 程序设计人员，便跳过

本章,从下一章继续学习。

53.2 QBASIC 界面

QBASIC 的程序设计环境是由 DOS 的 EDIT 命令提供的,它是全屏幕 ASCII 文本编辑器。但是,EDIT 是以文档方式启动编辑器,而 QBASIC 是以它的程序设计方式启动编辑器(如图 53.1 所示),两者之间有很大不同。用户可以在 QBASIC 命令后面加上/EDITOR 选项,使 QBASIC 也以文档方式启动编辑器。

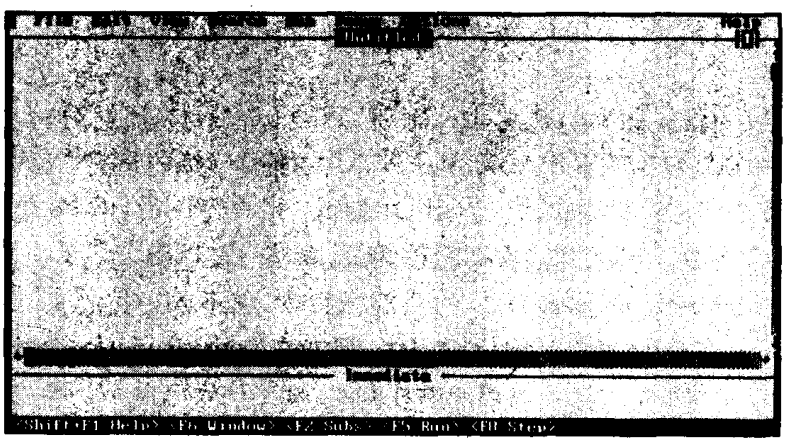

图 53.1 QBASIC 编辑器屏幕

在 PC 机上启动编辑器:

1. 在 DOS 命令提示符下,键入 QBASIC 命令。

   编辑器启动时,显示 Welcome 对话框,注意这和 EDIT 打开的窗口有些象,但不相同。

2. 按 ESC 键,清除 Welcome 对话框。

53.2.1 QBASIC 编辑器

QBASIC 主菜单支持用户建立、测试和调试 QBASIC 程序。File 菜单让用户打开和存储文件,同时也让用户退出 QBASIC。Edit 菜单中包括剪裁和拷贝文本的命令,以及建立子程序和函数。View 菜单让用户浏览和编辑当前程序的任何一部分,并可以把屏幕分割,以使用户能同时看到同一个程序的两个不同部分,并能立即切换到输出屏幕。Search 菜单的命令能寻找并替换文本字符串,Run 菜单的命令能执行整个程序或程序的一部分。Debug 菜单提供了几种工具,比如程序跟踪和分步执行等。Option 菜单包含改变显示方式的命令以及设置语法检查(Syntax Checking)功能状态的命令。

编辑屏幕有两个窗口。较大的窗口是标准的程序编辑区域,它的标题栏中所显示的是 Untitled,当然,如果所装入的程序已经有了名字,该栏中便显示被编辑的程序名。较小的窗口,称为 Immediate 窗口,每次只显示程序的一行,按 Enter 键时这行就被执行。用户可以利用该窗口测试自己所编写的程序的某些部分。

试在 Immediate 窗口中输入一些 BASIC 语句:

1. 选择 Immediate 窗口或按 F6 键，以启动该窗口。

   Immediate 窗口中出现光标。

注释： 注意状态栏中的提示<F6=Windows>，该提示告诉用户 F6 键用于切换窗口。

2. 键入 C=5 * 25。

   QBASIC 计算出 C 的结果，但不显示出来。

3. 输入 PRINT C。

   QBASIC 切换到输出屏幕，即切换到 DOS 命令提示符屏幕，并显示 125。在屏幕的底部告诉用户按任意键继续。

4. 按任意键返回 QBASIC 窗口。

   用户所输入的两条语句仍然出现在 Immediate 窗口中，但是，在上面较大的窗口中没有程序语句。

状态行显示光标的位置以及用户可以使用的一些功能键。一旦窗口中有语句，按 F1 键，就可以得到光标所在 QBASIC 单词的上下文帮助信息。表 53.1 列出了当编辑器窗口处于编辑状态时，用户能用的所有热键。如果不是处于编辑状态，则可使用的热键有所不同。

试显示有关 PRINT 命令的帮助信息：

1. 在 Immediate 窗口中，按光标移到单词 PRINT 所有的位置。
2. 按 F1 键。

   帮助信息窗口开启并显示 PRINT、LPRINT 语句标题。此时屏幕上出现了三个窗口。

3. 按 F6 键若干次，观察光标在三个窗口中的移动情况。
4. 按 ESC 键，关闭帮助窗口。

表 53.1　QBASIC 编辑器的热键

键	Unshifted 键	Shift+键	Ctrl+键
F1	Help	Using Help	
F2	View SUBs	Next Procedure	Previous Procedure
F3	Repeat Find		
F4	Output Screen		
F5	Continue	Run Start	
F6	Next Window	Previous Window	
F7	Execute to cursor		
F8	Step		
F9	Breakpoint		
F10	Procedure Step		Zoom
Del	Clear	Cut	
Ins	Overtype/Insert	Paste	Copy

53.2.2　改进编辑器效果

当用户输入程序时，编辑器便开始在程序行上工作。当用户按了 Enter 键或者把光标移

动到另一行上后，编辑器把所有 QBASIC 单词都转换为大写，并把空格插入等号和运算符两侧。如果 Options Syntax Checking 是 ON 状态，用户可以立即得到任何语法错误信息。按 F1 键可以查看当前语句的正确格式。编辑器会自动保持所有变量名字大、小写形式的一致性。如果为不同的过程设定了引用操作，则编辑器把所要求的 DECLARE 语句插到程序的开头。如果主程序中有 DEFtype 语句，编辑器也自动地把它插到其他过程的开头。

试用编辑器输入一个小程序：

1. 切换到编辑窗口。
2. 用小写字母输入语句 input "enter the first multiplier>",a
   按 Enter 键以后，QBASIC 立即把单词 input 转换为大写形式。
3. 用小写字母输入语句 input "enter the second multiplier>",b
   QBASIC 把单词 input 转换为大写形式。
4. 用大写字母输入语句 C=A * B。
   按 Enter 键以后，QBASIC 把空格插入到语句中，并且把前两句中的 a 和 b 变成大写字母，以保持和第三个语句中的 A 和 B 一致。
5. 输入 PRINT C。

用户可以利用 Edit 菜单或热键选出一些程序行并对它们进行拷贝或删除。被拷贝或删除的程序行存储在剪贴板中，直到有其他拷贝或删除代替它们为止。用户可以经常把剪贴板中的内容进行粘贴，甚至可以把其中的程序行拷贝或移动到其他程序或过程中，因为除非有其他的删除或拷贝操作，或退出 QBASIC 外，否则剪贴板中的内容不会受到任何影响。

试拷贝正在编写的程序中的程序行：

1. 选择所有的四行程序行。
2. 选择 Edit，Copy 菜单。
3. 把光标移到第一个空白行位置。
4. 选择 Edit，Paste。
5. 编辑这些新的程序行，使它们变成：

```
INPUT "Enter the third multiplier>",D
INPUT "Enter the fourth multiplier>",E
F=D * E
PRINT F
```

6. 按 F5 键。
   输出屏幕显示信息 Enter the first multiplier>。
7. 输入一个数。
   又出现信息 Enter the second multiplier>。
8. 再输入一个数。
   QBASIC 显示这两个数的积，并显示 Enter the third multiplier>。
9. 重复第 2 和第 3 步，输入第三和第四个乘数。
   显示第二个乘积后，QBASIC 显示信息 Press any key to continue。
10. 按任意键，返回 QBASIC 窗口。

53.3　将 GW-BASIC 或 BASICA 程序转换成 QBASIC 格式

QBASIC 以 ASCII 码格式存储文件，所以，如果用户编写好了的程序是用 GW-BASIC 或 BASICA 的缺省二进制形式存储的，就不得不对它们进行转换。启动其他 BASIC 编辑器，调入要转换的程序，然后把它们以 ASCII 码格式存储在有 BAS 扩展名的文件中。例如，如果要把一个 GW-BASIC 文件以 ASCII 码形式存储在 OLDPAY.BAS 中，可以遵照下列步骤：

- 启动 GW-BASIC。
- 装入文件。
- 输入命令 SAVE "OLDPAY.BAS",A。
- 输入 SYSTEM 命令，退出 GW-BASIC。

现在就可以启动 QBASIC，用 File,Open 命令打开这个程序了，并且可以在 QBASIC 编辑器中运行这个程序。

53.3.1　运行程序

在 QBASIC 中运行一个装入程序的最简便的办法是按 F5 键。另外，也可以选用 Run Start 命令。一个从别的 BASIC 格式转换而来的程序在 QBASIC 中运行时一般不会出现问题，除非它使用了一些 QBASIC 所不支持的语句功能(见表 53.2)。如果用户的程序使用了这些功能，那么必须对它作改动后才能在 QBASIC 中运行。如果出现语法错，屏幕上会显示信息提示用户按 F1 键。

表 53.2　QBASIC 和其他 BASIC 不兼容的语句

QBASIC 不支持的语句：
DEF USR
EXTERR(n)
LIST
LLIST
MERGE
MOTOR
USR
不支持的主要功能：
CALL (offvar,var1...)
CHAIN (ALL,MERGE,DELETE,linenumber)
CLEAR (,n)
DRAW (var=;)
PLAY (var=;)
VARPTR (#filename)
其他可能出现的问题：
QBASIC 有另外一些保留字
数据段中的 PEEK 和 POKE 可能有不同的结果。

53.3.2 行号

QBASIC 不要求一定要有行号，但它们可以存在。如果用户使用 GOTO 语句来控制跳转时，他们仍然可以使用行号，使用方法和其他 BASIC 完全一样。如果用户愿意，也可以使用标号。标号是由一个符号，后面跟一个冒号组成。例如，语句 GOTO ERRORLINE，是把控制传递到以 ERRORLINE：开头的行，并从那儿继续运行。任何含有最多可达 40 个字符的字符串或数字，只要不是 QBASIC 的保留字，并且在程序中也没有用作其他目的，就可以当作标号使用。冒号告诉 QBASIC 该字符串是一个标号。

53.4 新的常数和变量类型

在 QBASIC 中，常数和变量的类型比 GW-BASIC 或 BASICA 多。变量名最长可达 40 个字符，可以是大写字母和小写字母、数字和句点的组合。但是，如果使用了相同的字符串，即使大小写字母不同，QBASIC 也把它们看作同一个变量名。编辑器把所有字母相同的字符串转换为格式相同的大小写形式，以保持变量名的一致性。如果用户使用了变量名 FIRSTNAME $，然后又用了 Firstname $，则所有的 FIRSTNAME $ 都变转换为 Firstname $。

和早斯 BASIC 版本一样，可以用 DEFtype 语句把变量指定为字符串变量(DEFSTR)、整型变量(DEFINT)、单精度变量(DEFSNG)或者双精度变量(DEFDBL)。主程序中的所有 DEFtype 语句都将被自动地拷贝到后面编写的过程中。但是，如果在写好过程之后才在主程序中加上 DEFtype 语句，则过程不会被改动。变量名的数据类型后缀将使 DEFtype 语句无效。

53.4.1 常数

用户可以使用 CONST 语句设立命名常数，该语句有些象赋值语句：

```
CONST ConstantName=Value
```

Constantname 和 Value 必须类型相同，例如，可以使用语句 CONST Logo $ ="IBM"或者 CONST TotalLines% =400。变量的值中不能有变量名或函数，但如果需要的话，可以包含表达式或其他常数变量名。

QBASIC 不允许改变一个常数的值。假如已经定义 CONST Rate=0.0935，则不能再说明 Rate=Rate+0.01。当然，可以使用任何类型的字符常数，这和其他版本的 BASIC 一样。

53.4.2 数字变量

除了标准的数值变量类型外，QBASIC 还提供了长整型数，其值可以是从－2 147 483 648 到 2 147 483 647(标准整型数的值是从－32 768 到 32 767)。长整型数存储时占用 4 字节内存，而标准整型数只占用 2 字节。即使长整型数占用的空间和单精度数一样多，它仍然采用 2 的补码形式存储，并允许程序更有效地执行(与单精度数相比)。

用户可以用 DEFLNG 语句定义长整型数变量，或者在变量名的后面加上类型字符 & 表示长整型数，一旦在语句中定义后，长整型数就和其他整数一样使用。

53.4.3　字符串变量

字符串变量可以与在其他 BASIC 中一样被定义和使用。在 QBASIC 中，用户还可以用 DIM 语句定义固定长度的字符串变量。即：

```
* DIM StringName AS STRING * n
```

下面的语句定义长为 30 个字符，名为 Fullname 的字符串变量：

```
DIM Fullname AS STRING * 30
```

该语句优先于 DEFSTR 语句，另外，不能在变量名中使用 $ 类型字符。在很多文件及数据库应用中，固定长度的字符串非常有用。如果想设一个固定长度字符串数组，那么在上述语句的 StringName 后面加上数组的维数说明。下面的语句定义一个有 50 个元素的字符串数组，每个元素的字符串长度为 30：

```
DIM Fullname(50) AS STRING * 30
```

当数据被放入固定长度字符串变量时，它向左对齐，右边不足时用空格补上。如果数据太长，便会按给定的长度截断，而且不会有任何警告信息。

53.4.4　其他字符串函数

QBASIC 除了支持标准的字符串函数外，还支持一些新的函数。UCASE $ 和 LCASE $ 函数把字符串中的字母全部变成大写或小写。下面的语句把 Fullname(字符串变量)中的字符全部变成大写，然后把它赋给变量 Fixname $ (一个可变长度字符串变量)：

```
Fixname $ =UCASE $ (Fullname)
```

此时，Fixname $ 值的长度 Fullname 一样长。

LTRIM $ 和 RIRIM $ 函数删除一个字符串中左边或右边的空格或制表符。下面的语句删除字符串 Fullname 右边的所有空格，并把结果赋给变量 Fixname $ ：

```
Fixname $ =RTRIM $ (Fullname)
```

Fixname $ 的值不再有尾部空格，因此它可能比 Fullname 的值要短。

53.4.5　数组变量

QBASIC 也能定义和使用数组。另外，还可以用 To 词组定义下标的范围。下面语句设置一个 12 元素的数组，该数组有确切的下标：

```
DIM MONTH(1 TO 12)
```

下面的语句设置一个 61 元素的数组，下标范围是从 1940 到 2000：

```
DIM YEAR(1940 TO 2000)
```

也可以利用记录数据类型定义另外一类数组，它的每个元素都包括几种不同类型的数据。下面便讲述记录数据类型。

53.4.6 记录数据类型

QBASIC 支持用户自定义数据类型，称为记录数据类型。记录数据由 TYPE 块定义并生成一个记录。记录中可包含任何数据类型的组合，因此有几个不同的值。在定义每个元素中含有几种不同类型数据的数组或定义由混合类型数据组成的记录时，记录数据类型就十分重要了。

```
TYPE usertype
    element1 AS typename1
    element2 AS typename1
    ...<CE>END TYPE
```

usertype 是一个标准的 QBASIC 名字，它在过程中不能用于其他任何目的。在记录中，每个元素是一个数据项，能够被程序单独访问，typename 说明每个元素的类型和长度，它可以是整型(INTEGER)、长整型(LONG)、单精度型(SINGLE)、双精度型(DOUBLE)、字符串(STRING) * n、以及其他已经定义过的数据类型。例如，下面的 TYPE 块定义用户数据类型 GPA，其中包含三个不同的元素：

```
TYPE GPA
    StudentNumber AS STRING * 6
    StudentHours AS INTEGER
    StudentPoints AS SINGLE
END TYPE
```

usertype 定义一个数据类型，而不是一个变量名。所以它不能在语句中直接使用。它只能在自己的定义域中作为一类型名使用，或者在其他 TYPE 块中作为一个元素使用。要使用所定义格式中的数据，首先要把一个变量说明成这种数据类型。例如，定义好上述数据类型后，可以使用下面的语句，来把 Student 定义成一个 12 字节的变量，其中 6 字节存放字符串，两字节存放整数，而 4 字节存放一个单精度数：

```
DIM Student AS GPA
```

要引用记录的一部分，用句点把变量名和元素名连接起来，如下所示：

```
Student.StudentHours=Student.StudentHours+NewHours
```

53.4.6.1 数组中的记录

在数组中可以使用用户定义的数据类型变量。下面的语句设置一个有 180 个元素的数组，每个元素有 12 字节长度：

```
DIM Student (180) AS GPA
```

Student(25).StudentNumber 代表数组第 25 个元素中 6 字符长的字符串。

下面的例子定义数据类型 Employee，并把它用于数组变量 Worker：

```
TYPE Employee
    EmpNumber AS STRING * 9
    EmpSalary AS SINGLE
    EmpDependents AS INTEGER
END TYPE
```

```
DIM Worker(240) AS Employee
```

用一句点把变量名和记录的元素名连接起来，就可以访问记录中的某个元素。例如，Worker(Next%).EmpSalary，便指数组第 Next%个元素中记录的 EmpSalary 字段的值。

53.4.6.2　文件中的记录

在 QBASIC 程序中，记录数据类型可用于代替 FIELD 变量。它能使用户更加容易地建立和改动文件中的数据，并克服 FIELD 变量的局限性。当然，为了兼容起见，QBASIC 也支持 FIELD 语句。

53.5　新的程序流程控制功能

在早期的 BASIC 版本中，一个程序只能有基本的线性分枝。单行的 IF 语句处理选择，GOTO 语句及其中的行号能建立有条件循环。WHILE... WEND 语句为循环控制。QBASIC 中有更多的技巧及语句，以结构化和更有效的方式来控制程序的流程。

53.5.1　扩展的 IF 语句

QBASIC 支持早期的所有 IF 格式，包括单行的 IF... THEN 及 IF... THEN... ELSE 语句。阅读这些 IF 语句是比较困难的，而且更难调试。因此在 QBASIC 中设置了扩展的多行 IF 块，它符合结构化程序设计原理。下面是它的格式：

```
IF condition THEN
        statements to be executed if condition is true
[ELSE
        statements to be executed if condition is false]
END IF
```

IF 块从以 IF 开头的行开始。它必须包括条件表达式和单词 THEN。如果条件表达式的值为真，则其以下到 ELSE 为止的程序语句行都将被执行。如果没有 ELSE 语句，则 END IF 以前的语句都要被执行。如果条件表达式的值为假，则执行 ELSE 语句和 END IF 之间的所有语句。如果没有 ELSE 语句，程序便从 END IF 之后的语句开始执行。下面的例子中，如果条件是真，则执行程序中的几个命令：

```
IF TotalBill! >=CreditLimit! THEN
        PRINT OverLimitMessage $
        OverFlag $ ="Y"
        CALL warningflash
END IF
```

在下面的例子中，根据条件的值，程序只执行一组命令：

```
IF ApplicantAge% < 21 THEN
        PRINT MinorMessage $
        AgeProblem $ ="Y"
ELSE
        PRINT NoProblemMessage $
        AgeProblem $ ="N"
END IF
```

QBASIC 还支持复杂的多个 IF 块，即可以使用 ELSEIF 语句。如下所示：

```
IF condition THEN
    statements to be executed if condition is true
[ELSEIF condition-n THEN
    statements to be executed if condition-n is true]...
[ELSE
    statements to be executed if all conditions are false]
END IF
```

在下面的例子中，ELSEIF 语句中包含了第二个条件表达式，程序只允许执行三种可能中的一种：

```
IF ApplicantAge ! < 21 THEN
    PRINT MinorMessage $
    AgeProblem $ ="Y"
ELSEIF ApplicantAge% > 65
    PRINT SeniorMessage $
    AgeProblem $ ="Y"
ELSE
    PRINT NoProblemMessage $
    AgeProblem $ ="N"
END IF
```

如果需要，可以使用多个 ELSEIF 语句，以实现分枝。

提示： 可以优化 IF 和 ELSE 中的条件表达式，这样做可以缩短运行时间。

53.5.2 使用 Case 结构

也许用户对 ON...GOTO 或 ON...GOSUB 分枝语句很熟悉。这种情形也可以用 IF 或 ELSEIF 开语句处理。QBASIC 提供了一种特殊的语句块来处理这种情况。SELECT CASE 语句让用户命名一个变量，然后根据这个变量的值，执行指定的语句块。整个块构成一个 Case 结构，其最简单的格式如下：

```
SELECT CASE variable
    [CASE valuelist
        statements to execute if variable=any value in valuelist]...
    CASE ELSE
        statements to execute if variable <> any value in valuelist
END SELECT
```

在执行 SELECT CASE 块时，CASE 变量只能有一个值。只要有一个 CASE 块或 CASE ELSE 块被执行，控制就跳到 END SELECT 语句，然后往下继续执行。用户可以在每个 Valuelist 中写上很多值(每个值用逗号隔开)，并能使用尽可能多的 CASE 块。如果 CASE 变量的值和所指定的值都不相等，程序则转去执行 CASE ELSE 块中的语句。

例如，假如用户要显示一个菜单或实现多选择问题，所要输入的值是一个字符，并假定是从 A 到 D。对于每种情况，都执行一个不同的子程序，则所写的 CASE 块如下：

```
SELECT CASE MenuChoice $
    CASE "A","a"
        CALL DoSubA
```

```
        CASE "B","b"
            CALL DoSubB
        CASE "C","c"
            CALL DoSubC
        CASE "D","d"
            CALL DoSubD
        CASE DoBadChoice
    END SELECT
```

该例子让用户可以输入大写字母，也可以输入小写字母。如果所输入的字符不在 A 到 D 以内，子程序 DoBadChoice 将告诉用户重试一遍。如果需要，就可以在每个 CASE 语句后写上一组语句。

事实上，SELECT CASE 并不限制用户在每个 CASE 中只能写一个变量和一个变量值表。完整的格式如下：

```
SELECT CASE variable
    [CASE expressionlist
      statements to execute if any expression is true]...
    CASE ELSE
      statements to execute if no expressions are true
END SELECT
```

expressionlist 让用户可以使用多个变量或变量值表，以决定该执行的语句块。如果要在 expressionlist 中包括条件表达式，就在假定的变量名后面写上带关系运算符的 IS 单词 。例如可以使用 CASE IS ＞65 或者 CASE IS ＜＝20。如果要说明一个取值范围，就使用关键词 TO，如 CASE 1 TO 5，或 CASE "X" TO "Z"。也可以使用混合表达式，如 CASE 0，18 TO 20，IS ＞65。但要确保每一个值或表达式都是有效的。各值或表达式之间用逗号隔开。

53.5.3　循环结构

在早期版本的 BASIC 中，循环是由 FOR... NEXT 或 GOTO 语句构成，或者由 WHILE... WEND 构成。这些循环表达式在 QBASIC 中仍然有效。但 QBASIC 提供了比 WHILE... WEND 更简单更易结构化的 DO 循环块。DO 块从有 DO 的语句开始，到 Loop 语句结束。在循环过程中，所有在 DO 和 Loop 之间的语句都会被执行。该块包括了 WHILE 或 UNTIL 的基本元素，并把它们加到 DO 或 Loop 语句中。一个 DO WHILE... Loop 块和 WHILE... WEND 块是等效的。

WHILE 和 UNTIL 包含一个条件表达式。如果是写在 DO 语句中，则该表达式在循环体的开头被测试，如果是写在 LOOP 语句中，则在循环体尾部被测试。在使用了 WHILE 时，只要条件是真，循环就继续下去。如果使用了 UNTIL，当它后面的条件为真时，循环便结束。如果希望在第一次进入循环时，在循环的开头测试条件，就使用下面所示的 DO 循环格式：

```
DO [WHILE|UNTIL] condition
      statements to be executed
LOOP
```

如果希望循环块至少被执行一次，而不考虑条件的真和假，则使用下面的格式：

```
DO
```

```
        statements to be executed
LOOP [WHILE|UNTIL] condition
```

53.5.4 循环的内容

在 DO...Loop 块中可以使用任何语句。可以使用 IF 语句,FOR...NEXT、WHILE...WEND、嵌套 DO 块,甚至调用子程序。嵌套的 DO 循环可以是任意类型,内循环的结构不一定要和外循环一致。仅有的要求是内循环(包括 FOR...NEXT)必须在外循环结束之前结束。

如果必须在条件变成假之前结束循环,可以使用 EXIT DO 语句。当执行 EXIT DO 语句时,控制立即转移而去执行 LOOP 后面的语句,循环因此结束。

53.6 模块化程序设计

早期版本的 BASIC 程序设计都是线性化的,只要程序中不出现分枝,所有的语句都按顺序依次执行。

QBASIC 使用户能够以模块的方式工作,程序执行的开始和结束都在主模块内进行。程序中可以有尽可能多的过程、子程序和函数。每一个过程都是独立的有名字的整体,它们被主模块所调用。

在建立子程序时,用户应该给它一个名字。在主程序中,可以把子程序名作为一个语句或者写在 CALL 语句中,以启动子程序。当子程序中的语句执行完毕后,控制又返回主程序中调用子程序的语句的下一行语句,这和 GOSUB 过程很相似。

用户自定义的函数也需要有名字。用户可以象使用内部函数一样使用自己定义的函数。定义 QBASIC 函数和 DEF FN 语句的结果是一样的,只不过它允许多个程序行。

注释: *为了和其他 BASIC 版本兼容,QBASIC 也支持 GOSUB 和 DEF FN 语句。但可以在 GOSUB 语句中使用标号代替行号。*

子程序和函数是独立的过程,所以主模块(或其他过程)中的变量不会对其他过程产生影响,除非把这些变量说明成全局变量(COMMON SHARED)。不过,可以把指定的值传递给自变量表中的子程序或函数(全局变量及值传递将在后面讲述)。

53.6.1 检查模块程序

一个模块程序有主模块,一些函数和子程序。在打开程序文件时,可以在编辑窗口中检查主模块。如果按下 F2 键或选择 View SUB 功能,就可以在窗口中浏览和编辑另外的过程。按 Shift+F2 键直接显示下一个过程,而按 Ctrl+F2 键显示前一个过程。

53.6.2 建立过程

如果要建立子程序或函数,那么选择 Edit New SUB 或 Edit New Function 功能。如果光标停留在文字上,用户就可以看见如图 53.2 所示的对话框中的文字。用户可以接受这个子程序名字或者重新输入一个,然后选择 OK。下面所显示的窗口便是过程的编辑窗口。QBASIC 把用户提供的名字插入第一行(SUB 或 FUNCTION),在名字的后面加上一个空的括

号，同时也把这个名字加到最后一行(END SUB 或 END FUNCTION)中去。用户可以在这两行之间加上所需的任何语句。如果用户在主程序中使用了 DEFtype 语句，那么在建立过程时，这些语句会被自动拷贝到每个过程的前面。

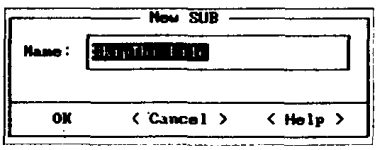

图 53.2　新过程的说明对话框

提示： 如果在程序中键入 SUB ProcedureName 再按 Enter 键，则会立即转到新过程编辑窗口。

当已经给过程取了名字并已将它存储后，就可以使用 View SUB 功能或按 F2 键查看或对它进行编辑。图 53.3 示出了对话框之一，该对话框让用户选择要编辑的过程。如果必要，也可以将过程删除。当存储程序时，编辑器就把有关新过程的 DECLARE 语句加到主模块的前面。

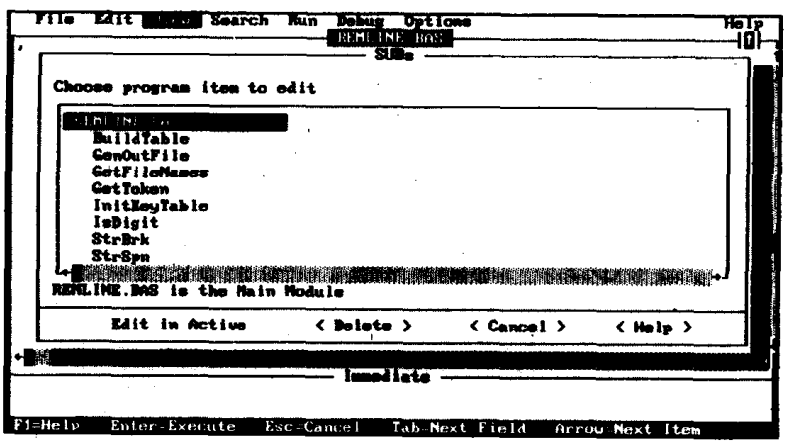

图 53.3　查看 SUBs 对话框

53.6.3　全局变量

缺省时，所有的变量都是局部变量，它们只能在自己出现的过程中使用。如果不是作了特殊安排，主程序中名为 TaxRate 的变量不会对子程序或函数中同样叫 TaxRate 的变量产生影响。用户可以告诉 QBASIC 使一些变量对所有过程都有效，这些变量称为全局变量。

全局变量必须在主程序中用 COMMON SHARED 语句说明，例如下面的格式：

```
COMMON SHARED VariableList
```

如果需要，语句中可以包括尽可能多的变量名，彼此间用逗号隔开。也可以使用多个 COMMON SHARED 语句。

提示： 对于比较简单的程序来说，可以用一个 COMMON SHARED 语句把所有

用到的全局变量一次性加以说明。

假定主程序中使用了如下语句：

```
COMMON SHARED PrimeRate !,CreditLimit!
```

如果在一个过程中设置了 PrimeRate! 的值，那么在程序使用的每个过程中，其值都相同。假如这个变量是局部变量，那么它的值只能在设置它的这个子程序中使用。

53.6.4 子程序

子程序是处于 SUB 和 END SUB 语句之间的代码块。当选用 Edit，New SUB 功能时，这两行便自动包含在程序中了。如果用户把 SUB 语句作为主模块的一部分输入，编辑器则认为用户准备建立新的子程序，并立即切换到子程序编辑窗口。这样做的结果和选用 Edit，New SUB 功能是一致的。一经定义，子程序就能被主模块调用，或者被其他子程序调用了。当执行到 END SUB 语句时，控制便返回调用子程序的语句的下一行。

子程序的通用格式如下：

```
SUB subprogramname [(parameters)]
        [local variable or constant declarations]
        statements to be executed
END SUB
```

子程序可用于处理很多复杂的操作。用户可使用子程序为报告的每一页打印题头，而调用另一子程序打印脚注。

下面的子程序演奏乐曲“生日快乐”：

```
SUB birthday
' Play a tune
PLAY "L4 C8C8DCFE2 C8C8DCGF2"
PLAY "C8C8>C<AFED2 B-8B-8AFGF2"
END SUB
```

主程序可其他过程通过 CALL 语句或直接把其名字当作语句来调用子程序。如果子程序没有参数，在调用时使用 CALL 或不使用 CALL 都行。例如可用下面的语句之一调用上面所述的子程序：

```
CALL birthday
birthday
```

每次调用子程序时，它的局部变量都重置到缺省值。但全局变量保持其当前值。如果要让子程序在两次执行期间保持其局部变量的值，就应在第一行中加上 STATIC 单词。下面的语句使得子程序 taxwithholding 在运行结束后保持它的局部变量的值，这些值可被下次执行子程序时使用：

```
SUB taxwithholding STATIC
```

如果子程序要计算程序运行的进程，则必须使用静态变量。例如，用户可能使用一静态子程序为一个报告的第一页打印脚注或页码。子程序每执行一次，页码数就增加 1。

在程序中，DECLARE 语句要放到所有执行语句的前面。主程序就通过 DECLARE 语

句找到子程序。当建立子程序时，QBASIC 就自动产生一个 DECLARE 语句，其中包含了 SUB 语句中可供主程序使用的信息：

```
DECLARE SUB subprogramname (parameters)
```

如果没有参数，则使用一个空括号。如果用户对子程序中的参数表作了改动，也一定要在 DECLARE 语句中作相同的改动。

53.6.5　变量传值

有时候，子程序必须使用从调用程序传递而来的指定值。如果是用 CALL 语句调用，则要传递的值必须都写在一个括号内，如果省略了 CALL，也要将值表的括号去掉。

可以用命名常数的形式传递一个常数，也可传递字符串、数字值或表达式。也可以用自变量的形式传递局部变量或表达式。每个被传递的局部变量在子程序中其值可以改变，在返回调用程序时，这个被改动过的值仍然有效。如果要防止传递的变量的值被改动，可以在调用语句中，把这个变量写在一个括号内，以构成一个表达式。

在下面的例子中，使 TaxRate 在子程序返回时不被改动：

```
CALL taxwithholding (Gross,Dependents,(Taxrate))
```

对表达式中的变量的值所作的改动，在子程序返回时无效。

子程序通过参数表接收传递过来的自变量，每个参数都用一个变量名代表。自变量是按位置关系而不是按名字传递给各个参数的。也即是说，第一个自变量传递给第一个参数，第二个自变量传递给第二个参数，诸如此类。由于是按位置关系传递的，因此，如果要跳过某个参数，就必须使用逗号。相互对应的项必须具有相同的数据类型，即字符串自变量传递给字符型参数，整型自变量传递给整型参数等。下面是一些有效传值的例子。

调用语句：

```
MakeMailingLabel Name $ ,Address $ ,City $ ,State $ ,Zip $
```

子程序：

```
SUB MakeMailingLabel (Line1 $ ,Line2 $ ,City $ ,State $ ,Zip $ )
```

调用语句：

```
CALL FigureTax (Total # ,TaxRate! ,TypeTax $ )
```

子程序：

```
SUB FigureTax (BaseAmount # ,CurrentRate! ,TaxType $ )
```

FigureTax 是对 BaseAmount #、CurrentRate! 和 TaxType $ 变量进行操作而不是对 Total #、TaxRate! 或 TypeTax $ 进行操作。但是，当控制返回主模块时，Total #、TaxRate! 和 TypeTaz $ 中的值，是控制退出子程序 FigureTax 时，它们所对应的变量的值。

试建立一个名为 DOUBLER 的子程序：

1. 在程序的下一行空行上，输入语句 SUB DOUBLER。

一旦按了 Enter 键后，QBASIC 便打开子程序编辑窗口，其中已经写好了第一行和最

后一行。

2. 输入下列子程序：

```
SUB DOUBLER (INVAR,OUTVAR)
OUTVAR=INVAR * 2
END SUB
```

3. 按 F2 键。

QBASIC 显示主程序(现仍叫做 Untitled)和子程序清单。

4. 选择 Untitled。

又返回主程序编辑窗口。

5. 在主程序的最后输入下列两条语句：

```
CALL DOUBLER (F,G)
PRINT G
```

6. 按 F5 键，测试所设计的程序。输入要求的变量值。

在 QBASIC 显示第二个积后，它把积乘以 2 再显示乘得的结果。

7. 按任意键返回 QBASIC 编辑器。

53.6.6 函数

函数计算并只返回一个值，它很象 GW-BASIC 和 BASICA 的 DEF FN 语句。两者之间的差别是 QBASIC 的函数名不需要以 FN 开头，以及函数过程可以使用尽可能多的语句。

和子程序一样，每个函数都必须用 DECLARE 语句定义。当通过 QBASIC 建立函数时，QBASIC 自动地生成一个适当的 DECLARE 语句，并把它插到主程序中。假如用户随后又改变了函数中的参数表，则也应改动相对应的 DECLARE 语句中的对数表，使两者完全一致。

53.6.7 建立函数

要建立一个新的函数，可以选用 Edit，New FUNCTION 功能或者输入 FUNCTION 语句并按下 Enter 键。当 QBASIC 获得了函数名以后，就打开函数编辑窗口，用户便可以编辑新的函数了。

函数的格式如下：

```
FUNCTION functionname [(parameters)]
        [local variable or constant declarations]
        statements to be executed
        functionname=expression
END FUNCTION
```

下面是一个函数，它计算一个介于 1 和整型自变量之间的随机整数：

```
FUNCTION RANDINT%(N%)
        RANDOMIZE TIMER
        RANDINT%=INT(RND * N%)+1
END FUNCTION
```

函数可被如下所示的语句调用：

```
CurrentValue=RANDINT%(GUESS%)
```

53.7 调试工具

QBASIC 提供了几种帮助用户调试程序的功能。当程序因为出错而中止时，在编辑器的窗口中，用户可以看到由光标指出的出错的程序行，同时也可以看到该行之上和之下的语句。调试功能甚至可以给出一些信息以确定错误所在。

53.7.1 Immediate 窗口

Immediate 窗口不仅仅使用户能在程序输入过程中检查出语法错误，而且在程序因出错而停止运行时，它还允许用户检查变量。用户可以在 Immediate 窗口中输入 PRINT 语句，以查看关键变量的当前值。

53.7.2 跟踪程序

在 Debug 菜单中有三个控制跟踪的命令。其中 Trace ON 用于启动或关闭跟踪功能。在命令后面有一小点表明跟踪处于 ON 状态。当跟踪生效时，程序运行很慢，光标指出当前正在执行的语句。通过屏幕，用户可以查看各条语句是否按照自己所预期的顺序执行。按 Ctrl+Break 键可以中断跟踪及暂停程序。Immediate 窗口可用于检查变量的值，按 F4 键还可以查看屏幕输出。按 F5 键又恢复跟踪。

Debug，Step 用于设置单步执行程序，即每执行一条语句后都暂停一下。选用 Debug，Step 或按 F8 键，执行下一条语句。如果所执行的语句有屏幕输出，可以按 F4 查看输出的结果。按任意键又返回程序窗口。在任何时候用户都可以使用 RUN，Start 来执行程序余下部分的语句。

试用单步方式执行所写的程序：

1. 选择 Debug，Step。

   QBASIC 把光标指在程序的第一条语句上。

2. 按 F8 键，执行这条语句。

   QBASIC 显示信息：Enter the first multiplier>。

3. 输入一个数。

   QBASIC 把光标指向第二条语句。

4. 继续按 F8 键，并输入乘数，直到程序运行完毕。

5. 随意按一个键，返回编辑器。

Debug Procedure Step 命令(F10)与此相似，但它把每个函数或子程序当作一条命令来执行。如果所有的函数和子程序已经调试完毕，则用该命令可以加快调试过程。

53.7.3 断点

如果希望程序运行到某处时就暂停，就可以在那里设置断点。用户可以在准备检查的地

方插入断点，以便于对程序的检查。例如，可以插入一个特殊的 PRINT 命令，以作为断点的标志。当在运行程序时，就会打印出 PRINT 命令中的内容，并且程序暂停。

如果要设置断点，把光标移动到选定的程序行，然后按 F9 键或选择 Debug Toggle Breakpoint 功能。光标所在的行会用红色显示或反相显示，这取决于用户使用的监视 器。用户可以随意设置断点。如果要删除一个断点，把光标移到断点所在的程序行，按 F9 键或选择 Debug Toggle Breakpoint。如果调试完成后要删除所有的断点，就选择 Debug Clear All Breakpoints 功能。

53.8 在 DOS 提示符下直接运行 QBASIC 程序

在 QBASIC 命令中写上/RUN filename 选项，就能在 DOS 提示符下直接运行 QBASIC 程序，而不需打开编辑器窗口。试着在命令提示符下运行前面所编写的 QBASIC 程序：

1. 选择 File，Save 功能，存储文件，给它取名为 PRACTICE. BAS。
2. 退出 QBASIC。
3. 在 DOS 命令提示符下，键入 QBASIC /RUN PRACTICE. BAS 命令。
   程序立即开始运行。

提示： 用户可以重定向 QBASIC /RUN filename 命令的输出。例如使用命令：QBASIC /RUN PRACTICE. BAS>PRN。

可以在批文件中写上一条命令，以便在批文件中执行 QBASIC 程序。当程序结束时，又回到批处理程序中。这样可以通过批处理程序把几个 QBASIC 程序连起来执行。使用 DOS 环境，可以在批处理程序和 QBASIC 程序及连接的 QBASIC 程序之间传递变量。ENVIRON 函数用于设置环境变量，例如命令 ENVIRON ALTDIR $ =C:\ALTERED $；而 ENVIRON $ 函数返回一个环境变量，例如命令 OPEN ENVIRON $ (ALTDIR $)\NAMELIST. TXT $ AS 1。

53.9 自 测 练 习

本章讨论了 QBASIC 和 DOS 软件包中早期版本的 BASIC 之间的差别。读者可按下列步骤作一些练习，以熟悉这些 QBASIC 的新功能：

1. 改动本章内所编写的 QBASIC 程序，使其仅能接受介于 2 和 30 之间的数。
2. 建立一个子程序，计算子程序被调用的次数，并显示出来。
3. 把一些旧版本的 BASIC 程序转换为 QBASIC 存储格式。

第 54 章 FDISK

简介： 在物理格式化之后和逻辑格式化之前，必须对硬盘进行分区，即使将整个硬盘分成一个区也不例外。FDISK 是 DOS 的分区程序。

54.1 用 FDISK 给硬盘分区

在 DOS 4 以前，硬盘驱动器限制在 32MB 以内，所以都要将大容量的硬盘分成几个逻辑驱动器。现在大多数用户已不再会遇到硬盘太大的问题，但是，仍然使用 FDISK 来建立多个分区以及建立多个逻辑驱动器。FDISK 还允许用户查看硬盘的分区信息。

FDISK 可用于建立非 DOS 分区。例如，假定用户除了使用 DOS 外，还希望能使用 Unix，在物理格式化硬盘后，就可以用 FDISK 来建立 DOS 分区和非 DOS 分区。然后把 Unix 装入非 DOS 分区中。当从 DOS 分区启动时，就装入 DOS；当从 UNIX 分区启动时，就装入 UNIX。

本章将讲解怎样使用 FDISK，即：

- 建立主 DOS 分区
- 建立第二 DOS 分区
- 在分区上设立逻辑驱动器
- 确定活动(启动)分区
- 删除一个分区
- 查看分区信息

警告： 如果硬盘上有数据，就不能改变硬盘的分区结构，否则就会丢失数据。在已经格式化并且已经开始使用硬盘以后，使用 FDISK 只能去查看分区信息。如果的确需要对硬盘重新分区，首先必须备份所有的数据，在分区完成并格式化后，再把它们恢复。

54.2 启动 FDISK

键入不带参数的 FDISK 命令，以启动 FDISK。假如用户仅有的硬盘没有分区，就必须从有 FDISK.EXE 文件的软盘上启动 FDISK。图 54.1 示出了 FDISK 启动后的主菜单屏幕。

```
MS-DOS Version 6
Fixed Disk Setup Program
(C)Copyright Microsoft Corp. 1983-1993

FDISK Options

Current fixed disk drive:1

Choose one of the following:

1. Create DOS partition or Logical DOS Drive
2. Set active partition
3. Delete partition or Logical DOS Drive
4. Display partition information

Enter choice:[1]


Press Ese to exit FDISK
```

图 54.1 FDISK 主菜单屏幕

注释： 如果用户的系统有两个或更多的硬盘，则将显示第 5 个选择项，让用户选择在哪个盘上工作。

54.3 建立 DOS 分析

选择项 1 可在硬盘上建立 DOS 分区。下一屏幕让用户选择是建立主 DOS 分区还是建立扩展 DOS 分区(如图 54.2 所示)。在同一个硬盘上，只能建立一个主分区和一个扩展分区。主 DOS 分区只能包含一个逻辑驱动器，即驱动器 C。扩展分区可以有多个逻辑驱动器。在建立扩展分区之前，必须先建立好扩展分区以后，并且，只有在建立好扩展分区以后，才能在它上面设置逻辑驱动器。

```
Create DOS Partition or Logical DOS Drive

Current fixed disk drive:1

Choose one of the following:

1. Create Primary DOS Partition
2. Create Extended DOS Partition
3. Create Logical DOS Drive(s) in the Extended DOS Partition


Enter choice:[1]


Press Esc to return to FDISK Options
```

图 54.2 建立 DOS 分区和 DOS 逻辑驱动器的菜单

54.3.1　建立主 DOS 分区

当选择建立主 DOS 分区时，FDISK 向用户询问是否要将整个硬盘都划分到该分区下，是否让它成为启动驱动器。如果回答 Yes，便开始对硬盘分区。这便建立了驱动器 C，并且可以退出 FDISK 并对 C 进行格式化了。

提示： 如果用户使用磁带备份系统，那么最好把分区的空间大小限制为不超过磁带机备份空间的最大值，这样做比较合理。例如，如果磁带的最大容量是 250MB，则分区大小也设为 250MB 为宜。如果用户准备压缩驱动器或压缩磁带上的备份数据，也必须相应地调整空间大小。例如，如果用户准备压缩磁带上的备份数据而不压缩硬盘驱动器，而且压缩比为 2 比 1，则应该给分区划分 500MB 的硬盘空间。

注释： 只有硬盘驱动器上没有主 DOS 分区时，才能用 FDISK 建立。

如果用户打算在硬盘上建立更多的逻辑驱动器，或者为非 DOS 分区保留一部分空间，则必须回答 No。然后 FDISK 向用户询问给主分区划分多大的空间。因为这个分区只能有一个逻辑驱动器，所以当完成分区后，FDISK 自动地建立驱动器。

注释： 如果在硬盘上为另一个操作系统比如 UNIX 保留了空间，则必须用它的操作系统在这部分空间上建立分区。

在用 FDISK 完成分区后，就必须用 FORMAT（或其他应用软件，比如 Norton Utilities 的 Safe Format）对主 DOS 分区进行格式化。在 FORMAT 命令中使用/S 选项，以完成系统文件的传输，这样该分区才能成为启动分区。

54.3.2　建立扩展 DOS 分区

如果为扩展分区保留了硬盘空间，那么选择主菜单中的第 1 项。这次选择建立扩展分区。同样，FDISK 将询问是否全部占用余下的磁盘空间。由于只能建立一个扩展分区，因此，只要不是为非 DOS 操作系统保留空间，就应该选择 Yes。

在建立扩展分区以后，可以在扩展分区上设置一个或多个逻辑驱动器。用户必须指明为每个逻辑驱动器分配的空间大小。每个逻辑驱动器都接受一个驱动器名字（D、E 等等）。完成分区后，就可以用 FORMAT 命令（或其他软件）分别格式化这些逻辑驱动器。在格式化扩展分区上的逻辑驱动器时，就不必使用/S 选项了。

54.4　设置活动分区

如果为另一个操作系统建立了非 DOS 分区，用户就必须指明从哪个分区启动。在 FDISK 的主菜单中，选择第 2 项即 Set Active Partition 项。FDISK 显示可作为合法启动驱动器的所有分区，并让用户选择把哪一个设置为活动分区。

假定用户是把 DOS 分区设置成了活动分区，那么，怎样才能访问另一个分区中的操作系统呢？只要系统是在 DOS 控制之下，另一个分区便找不到，DOS 不能访问那个分区的任何部分。如果用户想启动其他操作系统，则启动 FDISK，把非 DOS 分区设置成活动分区。这

样，在重新启动时，另一个操作系统就控制计算机。同样，它也不能访问 DOS 分区。

如果要回到 DOS 系统，就必须使用另一个操作系统的分区程序，把 DOS 分区设置成活动分区。

54.5 删除一个分区

如果用户打算重新对硬盘进行分区，就必须先删除当前的分区，以便给新的分区留出空间。删除一个分区并不删除其中的数据，FDISK 仅仅是删除分区表中有关分区的登记项。如果新建立的分区和原来的分区完全一样，则用户可以重新访问以前的数据（只要用户没有格式化）。

但是，如果新建立的分区使用了以前旧分区空间的一部分或全部，但不和旧分区完全一样，则用户就不能访问原分区中的数据了。当往新的分区中存储数据时，这些数据就会冲掉原来分区中的数据。

在 FDISK 主菜单中，选择第 3 项 Delete Partition or Logical Disk Drive 可以删除一个分区（如图 54.3 所示）。在删除扩展分区之前，必须删除它上面的所有逻辑驱动器。如果删除了主分区，同时也删除了 DOS 启动驱动器，就不能再从硬盘上启动了。假如已经把非 DOS 分区设置成了活动分区，则此时仍然能够从非 DOS 分区上启动。

```
                Delete DOS Partition or Logical DOS Drive

Current fixed disk drive: 1

Choose one of the following:

1. Delete Primary DOS Partition
2. Delete Extended DOS Partition
3. Delete Logical DOS Drive(s) in the Extended DOS Partition
4. Delete Non-DOS Partition


Enter choice: [ ]


Press Esc to return to FDISK Options
```

图 54.3 删除 DOS 分区或逻辑驱动器的菜单

注意，也能够从菜单中选择删除非 DOS 分区。假如用户打算删除属于另一个操作系统的分区，也可以使用 FDISK。用户可以把那部分空间加到自己的 DOS 分区中，但也得先删除那个驱动器，然后建立一个新的、更大的分区，这也使用户失去了对那个分区中数据的访问。

54.6 查看分区信息

如果要查看分区的状况，可以选择 FDISK 主菜单中的第 4 项 Display Partition Informa-

tion。图 54.4 是一个例子，它显示硬盘中有一个主 DOS 分区和一个扩展分区。主 DOS 分区是驱动器 C，扩展分区上的驱动器没有显示出来。

```
                    Display Partition Inforamtion

Current fixed disk drive:1

Partition  Status    Type  Volume Label  Mbytes  System  Usage
C:1        A      PRI DOS                 50     FAT16    45%
  2               EXT DOS                 62              55%

Total disk space is 112 Mbytes (1 Mbyte=1048576 bytes)

The Extended DOS Partition contains Logical DOS Drives.
Do you want to display the logical drive information (Y/N).....? [Y]


Press Esc to return to FDISK Options
```

图 54.4　显示分区信息的屏幕

状态栏中指出哪一个分区是活动分区。在本例中，驱动器 C 是启动驱动器。Usage 栏中显示的是该驱动器空间被数据所占用的百分数。System 栏显示驱动器是怎样格式化的。FAT16 是指该驱动器是用 DOS 格式化的，其 FAT 项为 16 位；FAT12 意思是由 DOS 格式化，FAT 项为 12 位。因为这些驱动器较小，其上的簇也少，因此 FAT 项只需 12 位就可以了。比 32MB 还大的驱动器，以及用其他非 DOS 软件分区和格式化的驱动器在 System 栏中的值有所不同。

如果磁盘上有扩展分区，FDISK 会询问是否打算查看逻辑驱动器的信息。如果回答 Y，则会出现类似图 54.5 所示的屏幕显示。对于扩展分区上的所有逻辑驱动器，FDISK 都会给出他们的大小、系统和使用状况等信息。

```
          Delete Logical DOS Drive(s) in the Extended DOS Partition

Drv  Volume  Label  Mbytes  System  Usage
D:                   20      FAT12   32%
E:                   42      FAT16   68%




Total Extended DOS Partition size is 62 Mbytes (1 MByte=1048576 bytes)


Press Esc to return to FDISK Options
```

图 54.5　显示 DOS 逻辑驱动器的有关信息

试查看硬盘的分区信息：

1. 键入 FDISK 命令，启动 FDISK。

FDISK 显示主菜单。

2. 选择第 4 项,Display Partition Information。

屏幕上便显示出分区信息。读者应该能指出硬盘上有几个分区,哪一个是活动分区。

3. 按 ESC 键,返回主菜单。

4. 按 ESC 键,退出 FDISK。

54.7 退出 FDISK

只要改动了分区表,在退出 FDISK 时需要重新启动计算机,这样新的分区表才能生效。在完成启动后,下一步就是对新的驱动器进行格式化。

54.8 自 测 练 习

本章讲述了如何使用 FDISK 查看和管理硬盘上的区分,不希望用户做这方面的练习,除非用户的硬盘确实需要分区。

第 55 章　由 DOS 跨上 Internet 网

简介： 世界各地的人们都可以通过他们的 PC 机彼此交谈和共享信息，这一切不是通过统一的联机服务来实现的，而是通过众所周知的全球互联网络 Internet 来实现的，它提供了计算机间广泛的(通常不是严格组织的)网络连接。

55.1　Internet 是什么？

确切地描述 Internet 是什么并非易事，但说明 Internet 不是什么却不是难事：Internet 不是有规划的服务，也不是由专业机构组织和管理的网络服务，没有任何人去建立 Internet 网络，然而它却存在了，除了 Internet 的用户没有去运行它。Internet 始于两个主要来源：大学和政府部门，它们都需要大量共享信息。随着调制解调器和远程通信时代的来临，大学教授，学生，以及政府官员都开始拨号进入彼此的计算机中，从而实现数据共享。随着数据共享的深入，人们开始互相发送电子信报，于是 Internet 诞生了。

今天，Internet 提供了全球范围内的计算机互联网络，这个网络连接了各种类型和型号的计算机。Internet 现在不再只是大学和政府机构的计算机互联网络，它提供了各种各样的服务，如：新闻服务，遍布全球的电子公告牌系统(BBS，Bulletin Board System)，购物服务，公司，图书馆等等，诸如此类。最新估计，约有 300 多万台计算机通过 Internet 实现了互联，当你从自己的计算机进入 Internet，在一个小时左右你就可以(假定你知道如何在网络上畅游)：

- 从 Glasgow 的一所大学里下装有关遗传工程学的一份报告
- 阅读《纽约时报》的首页内容
- 阅读来自世界各地朋友的来信，然后给他们回信
- 祝贺你在 Hanover 市的堂兄生日快乐
- 核对最近的股票市场报价
- 为旅行预定机票
- 购买一台新的打印机
- 下装个人信息管理共享软件的最新版本
- 参加 Little League 上的会议
- 把一份完整的报告发给 300 英里之外的客户
- 把一封电子信件发给总统，把复本发给议员和代表

以及很多很多其他工作！

55.2　Internet 资源

因为 Internet 每天都在发展变化，所以为完整的资源表分类是办不到的。不过，我们可

以讲述五个主要的 Internet 服务：数据库，新闻论坛(newsgroups)，电子邮件(e-mail)，网络闲谈(chatting)，以及其他服务。

注释： 有关 Internet 资源的使用指南，有联机的电子书籍，也有正式出版物。

55.2.1 数据库

你想在早上 3 点看宪法的副本吗，你想查找什么政府部门开发更好的捕鼠器？你也许想知道谁赢得了 1927 年的网球系列公开赛冠军，或者想知道如何治疗水痘。所有这些信息在 Internet 数据库里都有。

没有人知道 Internet 提供了多少数据库，但是它们的数量以千计，为了帮助人们以某种合理的方式搜索他们需要的信息，在 Internet 中有几种数据库搜索工具：Gopher，WAIS 和 Archie 用得最广泛。诸如 Gopher 这类搜索工具通过一系列菜单和子菜单帮助用户访问数据库。而 WAIS(广域信息服务程序，Wide Area Information Server 的缩写，其发音为“ways”)这类搜索工具允许用户用关键词来搜索。例如，可以用关键词“chicken pox”或“mousetrap.”搜索数据库。另一些搜索工具允许用户以电子邮件的形式返回查询结果(而不是以联机方式返回查询结果)，如 Archie。

55.2.2 新闻论坛(Newsgroups)

对于许多用户来说，新闻论坛(newsgroups)就是所谓的 Internet。新闻论坛是一组想共享有关特殊专题信息的人们，这些专题覆盖各种方面，如：刺绣，所喜爱的曲棍球队，有机园艺技术，诗歌，使用 Lotus 1-2-3，以及家谱。新闻论坛的成员把电子信报放到其他论坛成员可以阅读和回答的电子公告牌中。当用户进入 Internet 后，往往要查看新闻论坛是否有新的电子信报。一些新闻论坛是温和的，它们以适当的方式回答论坛成员关心的问题。许多新闻论坛并不温和，正如他们所说，任何问题都可讨论。然而，非温和论坛内的不适当行为经常使解答者感到“脸红”，而解答者常常是本论坛内的其他成员。

55.2.3 电子邮件(E-mail)

现在，人们一周要写数十封信件给朋友们，同事们，有时还要写信给相当陌生的人。电子邮件通过 Internet 传送，速度非常快以至于一天可以交换二至三次电子信报。由邮电局投递的老式邮件与电子邮件形成了鲜明对比，人们戏称之为“蜗牛邮件”。

当有人给你发送电子邮件时，电子邮件服务器就为你存储该邮件，直到你取走它。作为欢迎信息的一部分，你将收到一个通知，它告诉你有你的电子邮件。你可以取走电子邮件或者联机阅读之，这取决于你愿意花费多少联机时间。在取走或阅读完电子邮件之后，就可以从服务器中删除之。要发送你自己的电子邮件，你可以脱机生成之，然后把它发出去；也可以联机生成之，然后发送每一封写好的信件。

55.2.4 邮件论坛(Mailing Lists)

邮件论坛(Mailing list)在些象新闻论坛(newsgroup)，只是它的成员用电子邮件进行通信，而不是把电子信报放到电子公告牌中来实现信息共享。邮件论坛通常集中于诸如铁路旅行这类主题的讨论，大多类似于新闻论坛。邮件论坛不是由用户把电子邮件逐个发送给每个

论坛成员，而是先把电子邮件发送服务器，再由邮件服务器把它分发给论坛的所有成员。

用户若属于一个邮件论坛，则会收到比自己所需的多得多的电子邮件。每当用户登入网络系统时，就会发现有数十封发给你的电子邮件。当然，与私人的电子邮件不同，不是每封信都需要回信，不过一封封阅读它们也是令人厌烦的。

55.2.5　网络闲谈(Chatting)

网络闲谈(Chatting)代表了 Internet 通信的最快速形式。在这种形式的通信中，两个以上的人同时联机通信，并通过他们的键盘来回交换电子信息。在白天或晚上的任何时刻，在 Internet 上都会发现有网络用户在闲谈，他们可能位于世界的各个角落。Internet 提供了许多命令用于查询谁处于联机状态，某个特定人是否在上机，建立两个闲谈或召开一组人的网络会议(这些会议和闲谈可以是公开的，也可以是非公开的)。但是，大多数人通过他们的联机服务进行闲谈。

游戏屋(game room)就是一种流行的闲谈会议，参加者玩的游戏有 trivia，Dungeons，Dragons 以及其他一些联机游戏。科幻闲谈会议也是很流行的，参加者可以到 Merry Olde England 去旅行，也可以前往 Starship Enterprise 飞船。一些闲谈会议有一个规章制度，如俱乐部或专业团体一样。网络闲谈可能是迷人的，一些人沉溺于其中。许多献身于 Internet 事业的用户已经破产了，这并非耸人听闻，一天内花费几小时同世界各地的新朋友闲谈，高额的电话费和联机费就随之而来。

55.2.6　其他服务

新闻论坛、电子邮件、网络闲谈，以及数据库只是 Internet 上可用的几个资源。许多公司和代理提供范围广泛的各种特定服务，从旅行预定机票到股票市场报价，从购物服务到产品技术支持，从 Commerce Business Daily 日报到 United Press International。用户可以联机存取自己喜爱的杂志；核对自己的信用率；下装各种照片，声音片段，视频片段以及软件；查看最近的电影评论；甚至可以从网络上攻读大学学位。现在，Internet 提供的各式服务几乎是无限的，将来更是如此。

55.3　访问 Internet 网

会使用计算机和调制解调器的人都可以使用 Internet 网。但是，还需要有控制调制解调器的远程通信软件，以及某种进入 Internet 的网关。对于国内用户来说，网关通常搜索电子公告牌的一种服务形式或者其他类型的联机服务；但是，用户若是访问自家公司或大学的计算机，就不需要外部的网络服务。

55.3.1　有关调制解调器

调制解调器将更快更好。数据传输率的单位是波特，一波特表示一秒传送一个比特。五年以前，调制解调器的传输率一般为 300 到 1 200 波特，这种传输率还太慢而使 Internet 不能适用于大多数国内用户。今天，波特率为 2 400 到 9 600 的调制解调器是很常见的。市场中的调制解调器往往具有 14 400 波特的数据传输率，与此同时 28 800 波特的调制解调器也

开始出现了。

无论你的调制解调器何时与另一个调制解调器相连，波特率取决于相连两个调制解调器的最高速率。如果你的调制解调器具有 14 400 波特的数据传输率，但与之相连的另一个调制解调器的最高速率只有 9 600 波特，那么你的调制解调器也必须采用 9 600 波特的速率。(所有的调制解调器都可以采用比它们最高传输率低的传输率)。许多联机服务目前尚未升级它们的调制解调器，实际上，用户会发现自己的调制解调器的数据传输率被限制在 2 400 波特(因为大多数 Internet 提供的联机服务的波特率限制为 2 400)。

波特率会产生很大的不同吗？对，确实如此。以 9 600 波特下装一个文件如果需要 5 分钟，那么以 2 400 波特下装这个文件就需要 20 分钟。当用户要为使用的联机服务付帐时，再加上占用长途电话线路的费用，多出那些的通信时间就得多付许多钱。

如果你要为访问 Internet 的服务付钱，我建议你购买最快的调制解调器(如果其价格你可以接受的话)。也许你不能马上使用高速调制解调器的最高速率，但是你的调制解调器可以用上几年而不过时，而且，在这段时间内你使用的服务也会不断地提高它的设备。

55.3.2 远程通信软件

远程通信软件用于控制用户的调制解调器，并为用户拨号和与另一台计算机通信提供界面。如果用户通过政府部门或学校的计算机访问 Internet，就不必提心远程通信软件包，因为可以使用安装在这些系统上的通信软件。如果用户在家中通过拨号来访问政府部门或学校的计算机，就需要在用户家中的计算机上安装远程通信程序。用户的调制解调器可能配备有通信程序，但是它不是访问 Internet 的最佳选择。为传真/调制解调器配备的许多程序更适用于发传真而不是联机通信，这些程序只含有经过缩简的功能集。

注释： 一些美国国内的联机服务(如 America Online 和 CompuServe)为用户提供了他们自己的远程通信软件，这些程序是专为帮助非技术人员访问这些服务而精心设计的。

在寻找合适的通信软件时，要考虑以下几点：

- 它是在 DOS 下运行还是在 Windows 下运行？(你喜欢哪一种呢？许多 DOS 界面的通信程序不能使用鼠标，因此用户必须自己输入命令。)
- 它易于安装吗？易于对用户的调制解调器进行设置吗？远程通信软件的设置可以非常困难。据我所知，有些远程通信软件要花费通信技术人员几个小时才能安装完。而有些通信软件可以完成所有安装和设置工作，如搜寻用户的调制解调器，再自动根据它的特性进行配置，完整地安装好只需几分钟。如果用户精明的话，最好在选用通信软件之前通过朋友了解一下情况。
- 它支持什么协议？协议决定了两台计算机间传输文件的方法。常见的协议有 Kermit，XModem，YModem 及 ZModem。其中最快的协议是 ZModem，如果通信软件支持这种协议就使用它。
- 如果你将在 Internet 上使用网络闲谈，候选的通信软件包括便于进行网络闲谈的各种功能吗？(如分割屏幕，向前卷屏等)，分割屏幕可使网络用户在接收信息的同时在屏幕的另一个块区域输入要发出去的信息。若没分割屏幕功能，到达的闲谈信息就

会和用户要发出去的信息混合在一起而无法阅读。向前卷屏功能使网络用户可以重读前面的信息。

- 它可以脱机编制电子邮件吗？联机编制电子邮件的费用很高。

可以带来方便的其他功能：

- 个人地址簿，只需从中选择电子邮件的地址而无需手工输入它们。Internet 地址不够友好是众所周知的。例如：coralcee@lamer. colo. edu 和 30900512. 3899210@dca. com，采用地址簿就可以节省用户的时间，也可以减少错误。
- 宏，可以记住用户经常重复输入的命令，使用户不必每次输入它们。
- 自动处理电子邮件的子程序，它可以自动接收和投递电子邮件，根据需要自动脱网和上网（指 Internet 网）。
- 在完成数据传输之后自动脱网。因为在 Internet 上传输数据要花费许多分钟，如果通信软件可以设置成在文件传输结束后自动脱网，那么用户在发出传送文件的命令后就可以离开计算机去干别的事情。

55.3.3　连通 Internet 的网关

远程通信软件为用户提供了拨号上网功能，但是在网络另一端的软件必须提供连通 Internet 的网关。而正是由网关决定哪些 Internet 资源可供联网用户使用。

一些网关完全开放 Internet，用户可以访问自己所知的任何 Internet 资源。另一些网关只提供了有限的存取范围，它们只提供有限的一组新闻论坛、邮件论坛、Gopher 及 WAIS 数据库等服务，而没有提供 Archi、局域网络闲谈和 Internet 网络闲谈（IRC）等服务。对于许多人来说，有限的服务功能已足够用了，而且这类服务往往更易于使用。

如果用户是通过学校或其他部门的计算机联入 Internet，就不用考虑网关了，因为那些计算机都已配备了。不过，用户还需掌握一些 UNIX 的命令，如基于 UNIX 的网关命令（而非 DOS 命令）。

如果用户通过自己的计算机访问 Internet，那么用户必须是电子公告牌系统（BBS）或联机服务的用户，因为只有通过它们用户才能联入 Internet。许多本地的 BBS 都提供至少几种访问 Internet 的服务，另外，美国国内的一些联机服务，如 CompuServe，America Online，及 Delphi 提供部分或全部的 Internet 服务。

注释： 许多电子公告牌系统（BBS）提供类似于 Internet 的各种资源，但它们并不真正访问 Internet。大多数 BBS 都提供电子邮件和新闻论坛服务，许多 BBS 还提供邮件论坛服务，而某些 BBS 甚至提供网络闲谈及其他服务。在某些情况下，它们链接到其他 BBS 网络，甚至全球网络，因此网络用户可以接识数十万的网络爱好者而无需进入 Internet。用户也许会发现本地的 BBS 提供了自己需要的所有资源。本章只介绍了在 Internet 上广泛使用的一些服务。

在选择网关时，要考虑的因素如下：

- 网关有本地电话号码吗？用户当然不想为本地联机服务付长途通信的费用，是计划使用网络闲谈服务时这一点尤为关键。
- 网关的使用费用如何？从理论上说，Internet 是免费的，但是用户必须自己支付联机

服务费用。大多数情况下，用户按月付占用线路的费用。

- 有无付加费？一些联机服务在高峰时间要增收付加费(如白天增收附加费)。
- 必须先拨入诸如 Sprint 或 Tymnet 这类电话网吗？这样是不是要另行收费？在许多地区，这是必然的，但是在非高峰时间内拨号入网是不收费的。
- 网关提供的服务是你需要的吗？
- 如果网关包括网络闲谈服务，那么它提供 Internet 上的网络闲谈服务还只是提供本地网络闲谈服务？你也许只喜欢同属于本地电子公告牌上的人们闲谈，也许只喜欢一种服务，如 America Online，它不提供 Internet 上的网络闲谈服务，不过它已拥有一百多万用户，而且每时每刻都有数以千计的人们在联机闲谈。当然，你也许要求有国际网络闲谈服务，以便可以和世界各地的朋友和家人交谈。

55.4 Internet 的一些警告

Internet 是有趣的，也是有益的，人们可以从中受到教育，进行娱乐。但是，它也有存在着难以避免的危险。下面列出 Internet 存在的一些问题，以提醒人们的注意：

- 计算机病毒可能在 Internet 上传播。如果从 Internet 下装数据——特别是软件，那么应该经常查一查病毒。
- 绝不要把你的口令告诉任何人！如果有人通过电子邮件或其他形式的电子消息询问你的口令，千万不要告诉他。保存这些消息并立即通知你的联机服务提供者。如果你把口令告诉他，他也许会在你知道前将银行帐号里的存款搜刮干净。
- 要采用不易猜测的口令。不要使用自己配偶的名字，狗的名字，自己的生日或社会保险号，不要使用任何易于猜出的口令。
- 一些人在网络上行骗——男人冒充女人，小孩冒充成人，个人冒充社团，职员冒充百万富翁，或者反过来。在网络上同陌生人交谈时你一定要谨慎，直到你确信他们说的是实话。
- Internet 上的人们来自各行各业，他们有不同的背景和生活情趣。你将会收到大量你认为受到侵犯的电子邮件和信息，最好的做法是不予理睬。如果你发现受到某种干扰，请向你的联机服务提供者报告。

55.5 怎样从 DOS 访问 Internet？

DOS 中没有提供访问 Internet 的命令。用户可以在 DOS 或 Windows 下启动自己的远程通信软件，由这类软件通过网关访问 Internet。

第六部分 命令用法参考

命令用法参考

约定： 在本章的命令参考中使用如下印刷约定：

全部字母大写　必须原样输入的文字，但可以用大写或小写字母。

[ ]　表示括号内的内容可有可无；所有方括号外的内容是必需的。

...　省略号代表其前面的参数可重复任意多次。比如，filespec...代表用户可输入一个或几个文件名。

|　竖线代表两个互斥的选项。比如，/B | /C 说明可能输入/B 或/C，但不能同时输入。

本参考中，命令名旁出现的图标分别代表命令的类型：

① 内部命令（在 COMMAND.COM 中）

Ⓔ 外部命令（有单独的执行文件）

Ⓑ 批处理命令（只能用于批处理程序文件中）

Ⓒ CONFIG.SYS 命令（只用于 CONFIG.SYS 中）

Ⓔ APPEND

允许程序访问指定目录中的文件，就像这些文件位于当前目录中一样。

句法 1

APPEND [path[;path...]] [/X[:ON|:OFF]] [/PATH:ON|:OFF]

附加 path 目录到当前目录中。

参数与选项

无参数　显示已被附加的所有目录。

path　将被附加的目录。

/X　指示 DOS 是否到被附加的目录中寻找可执行文件。缺省的设置是 OFF。/X的作用与/X:ON 相同。

/PATH　指示当数据文件名本身带有路径时 DOS 是否到附加目录中去寻找。缺省设置是 ON。

注释

假如 path 中不包含驱动器名，DOS 则认为目录在执行 APPEND 时的当前驱动器上。

当/X 设成 ON 时，DOS 将首先在当前目录中查找程序，然后在附加的目录中查找，最后在指定的路径中查找。

如果用户准备在系统使用期间的某一时刻用到/X:ON 选项，那么必须在首次使用这一句法的 APPEND 命令时指定/X:ON。于是，在这以后的任何时候都可以使用/X:OFF 和/X:ON 了。

每一次使用这种句法的 APPEND 命令都会取消这一部分前一次设置的结果。若要附加多个目录到当前目录中，应该在一个 APPEND 命令中列出这些目录。在更改/X 或/PATH 选项的设置时，要确保在命令中包含应被附加的所有目录。

如果在使用过程中同时用到 APPEND 和 ASSIGN 命令，那么即使两个命令不涉及到同一驱动器，AP-

PEND 也要在使用 ASSIGN 命令之前使用。

DIR 命令作用于当前目录时并不会列出附加目录中的文件。

警告： 许多应用程序在需要更新文件时生成一个新文件而不是覆盖旧文件。即使原始文件在附加的目录中，新文件仍会存在当前目录中——就是说没有替换掉原来的文件。仅仅因为这一原因，使用 APPEND 可能带来危险。许多专家建议不要使用 APPEND。在想用 APPEND 命令时，更明智的选择是使用路径名和 PATH 命令。

不要在运行 Windows 或 Windows 的 SETUP 命令时使用 APPEND 命令。

应用举例

```
APPEND D:\MYDIR\TOKEN
```

附加指定的目录到当前目录。

```
APPEND\NEWDIR /X
```

附加 NEWDIR 目录(在当前驱动器上)并打开/X 选项，这样当程序不在当前目录中时 DOS 也会到 NEWDIR 中去寻找。

```
APPEND \NEWDIR;A:\BDIR /X:OFF /PATH:OFF
```

附加当前驱动器上的 NEWDIR 目录和 A 驱动器上的 BDIR 目录到当前目录。关掉/X 选项以使程序不在当前目录中时 DOS 到指定的查找路径中搜寻。若关掉/PATH 选项，而且当数据文件全名中含有路径时，DOS 只查找指定目录。

句法 2

```
APPEND /E
```

将被附加的目录表赋给名为 APPEND 的环境变量。

注意

如果使用这一命令，它必须作为启动系统后的第一个 APPEND 命令。注意在这一句法中不带任何路径。

句法 3

```
APPEND;
```

取消当前的附加目录表。

Ⓔ ATTRIB

见第 16 章

显示、改变文件和目录的属性或清除所有属性。

句法 1

显示或改变文件和目录的属性。

```
ATTRIB [+R | -R] [+A | -A] [+H | -H] [+S | -S] [filespec] [/S]
```

参数和选项

无参数	显示当前目录中所有文件的属性。
+R	打开只读属性。
-R	关闭只读属性。
+A	打开归档属性。
-A	关闭归档属性。
+H	打开隐含属性。

−H　关闭隐含属性。

+S　打开系统属性。

−S　关闭系统属性。

filespec　指定需做处理的文件或目录。用通配符可指定文件但不能指定目录。当使用了 filespec 但没有加属性选项时，ATTRIB 则显示这些文件或目录的属性。

/S　处理当前目录中所有子目录中的文件。

注意

只读属性可防止文件被改写或删除。隐含属性可使很多命令不能访问该文件。系统属性组合了只读和隐含的特征。归档属性说明文件没有做备份。

如果文件具有隐含或系统属性，那么只有在先前执行过的命令或本次命令执行中关掉隐含或系统属性后才能改变为其他属性。

在创建或改写了一个文件后，DOS 自动为其打开归档属性。很多备份程序根据归档属性选择文件，并在备份之后关掉归档属性。

若同时使用了/S 和文件名，ATTRIB 忽略/S 选项而只处理指定的文件或目录。

应用举例

```
ATTRIB +H \D1
```

打开目录 D1 的隐含属性(除非 D1 是系统目录)。

```
ATTRIB −H −S −R TESTFILE. *
```

关闭当前目录中名字符合的所有文件的隐含、系统和只读属性。

```
ATTRIB +A /S
```

打开当前目录及其子目录中所有文件的归档属性(除了系统文件和隐含文件)。

```
ATTRIB *.EXE
```

显示当前目录中所有名字符合的文件的属性。

句法 2

关闭文件的所有属性。

```
ATTRIB, [filespec] [/S]
```

参数和选项

无参数　关闭当前目录中所有文件中的属性。

filespec　表示需要处理的文件。可以在文件名中使用通配符。

/S　处理当前目录的所有子目录中的文件。

应用举例

```
ATTRIB,
```

关闭当前目录中所有文件的属性。

```
ATTRIB,/S
```

关闭当前目录及其子目录中所有文件的属性。

```
ATTRIB,MYSYS.EXE
```

关闭当前目录中文件 MYSYS.EXE 的所有属性。

Ⓘ **BREAK**

© 见第 21 章

控制扩展的中断检查。

句法

BREAK[=][ON|OFF]

参数

无参数 显示扩展中断检查的当前状态。

ON 打开扩展中断检查。

OFF 关闭扩展中断检查。

注意

当 BREAK 设成关闭时，DOS 只有在读取键盘或向屏幕和打印机输出时才检查中断键(Ctrl+Break 或 Ctrl+C)。如果 BREAK 设为打开，DOS 在其他操作，如读写硬盘时也会检查中断键。增加中断检查可能会使这些操作变慢。

缺省情况下 BREAK 被关掉。

可以在命令提示符下或 CONFIG.SYS 中使用带一个参数的 BREAK 命令。

应用举例

BREAK ON

打开扩展中断检查。

BREAK=OFF

关闭扩展中断检查。

BREAK

显示中断检查的当前状态。这一形式不能用于 CONFIG.SYS 中。

© BUFFERS

见第 21 章

指定磁盘缓冲区的数目。

句法

BUFFERS=read-write[,look-ahead]

参数

read-write 指定读写缓冲区的数目。范围从 1 到 99。如果不用 BUFFERS 命令，缺省值取决于系统设置。

look-ahead 指定预读缓冲区的数目。范围从 0 到 8。缺省值是 0。如果 read-write 或 look-ahead 中任一个数值出界，look-ahead 则被设置为 0。

注意

DOS 在读写磁盘时利用读写缓冲区保存数据。在实际读或写磁盘时，DOS 首先查看所需数据是否在缓冲区中。缓冲区越多，DOS 越有可能减少访问磁盘的次数。

如果分配了预读缓冲区，DOS 在读盘时会多读一些扇区，这就增加了后面要读的数据已进入缓冲区的可能性。

最合适的缓冲区数目取决于系统配置。合理的数目会提高整机性能。使用过多的缓冲区不仅浪费内存而且会降低读写速度，因为在这样多的缓冲区中寻找数据可能比直接访问硬盘还要慢。

如果使用了 SmartDrive，那么分配大约 15 个读写缓冲区并且不需要预读缓冲区。

每个缓冲区大约占用 500 字节的内存。如果 DOS 被装入 HMA(高内存区，见 DOS 命令)，如有足够的空间，那么全部读写缓冲区随 DOS 一并装入 HMA，否则它们位于常规内存。BUFFERS=10 可使 HMA 有足够的空间装下 DOS、DoubleSpace 和缓冲区。要注意有些安装程序会增加 BUFFERS 设置以利于执行它

的应用程序。

应用举例

BUFFERS 30,5

分配 30 个读写缓冲区和 5 个预读缓冲区。

BUFFERS 15

分配 15 个读写缓冲区，没有预读缓冲区。

Ⓑ CALL

见第 37 章

调用第二个批处理程序而不中止第一个批处理程序。

句法

CALL batch-command

参数

batch-command　用来执行被调用批处理程序的命令。

注意

batch-command 与在命令提示符下执行批处理程序的命令完全相同，并带有必要的参数。

被调用程序的执行结束后，DOS 返回原来的程序中继续执行 CALL 下面的命令。

若想在第二个程序结束后不返回原来的程序，只要在原来程序中使用 batch-command 而不加 CALL，就像在 DOS 提示符下输入命令一样。

批处理程序也可调用自身。

应用举例

CALL NEWDAY

启动名为 NEWDAY.BAT 的批处理程序。NEWDAY 执行结束以后，DOS 返回到原来程序中 CALL 命令的下面一行。

CALL NEWDAY JAN 1999 2

启动名为 NEWDAY.BAT 的批处理程序，并传递命令行参数以替换%1(JAN)、%2(1999)和%3(2)。NEWDAY 结束后，DOS 继续执行原来程序中 CALL 之后的命令。

NEWDAY

启动名为 NEWDAY.BAT 的批处理程序。因为没有使用 CALL，NEWDAY 执行结束后 DOS 不返回原来的程序。

Ⓘ CD

见第 14 章

改变或显示缺省目录。CHDIR 是本命令的另一种形式。

句法

CD [drive:] [path]

参数

无参数	显示当前驱动器上的缺省目录(即当前目录)。
drive	显示指定驱动器上的缺省目录。驱动器必须以冒号结尾。
path	指定新的缺省路径名。path 必须是一个已存在的路径。

注意

DOS 为系统中的每个驱动器建立一个缺省目录。要改变一个非当前驱动器上的缺省目录，在 path 中包含驱动器名。

path 中的两个点代表表示目录树中下一个更高层的目录。例如，..代表当前目录的父目录；..\..代表父目录的父目录。

应用举例

CHDIR

显示当前目录名。

CD E:

显示 E 盘的缺省目录。

CD \

使当前驱动器的根目录作为当前目录。

下面的例子假定当前驱动器是 C:，而当前目录是\OFFICE\MYDIR。(这些例子相互独立，且无先后次序。)

CD NEWDIR

改变当前目录为 C:\OFFICE\MYDIR\NEWDIR。

CD ..

改变当前目录为 C:\OFFICE。

CD ..\YOURDIR

改变当前目录为 C:\OFFICE\YOURDIR。

CD \NEWDIR

改变当前目录为 C:\NEWDIR。

CD E:\XDIR

改变 E 驱动器的缺省目录为 XDIR。当前驱动器和当前目录不变。

Ⓔ CHCP

见第 49 章

更改所有设备的代码页，或显示当前代码页的号码。

句法

CHCP [page]

参数

无参数　显示当前代码页的号码。

page　改变到指定的代码页。

注意

代码页是键盘或显示器的字符集。DOS 以软件形式提供 6 个不同的代码页，这样用户就可以输入或显示非英语字符。

美国的缺省代码页号是 437。这也是在美国出售的大部分硬件内建的代码页。美国的另一个可选代码页是 850，其中含有更多的国际通用字符。

如果在 CONFIG.SYS 中使用了 COUNTRY 命令，DOS 则按照指定的国家建立缺少的和可选的代码页。

在用 CHCP 改变代码页之前必须完成以下操作：

- 在 CONFIG. SYS 中装入适当的驱动程序使显示器可以使用软件代码页。
- 要装入 NLSFUNC 才能用 CHCP 改变代码页。
- 用 MODE 命令装入显示器代码页。

CHCP 命令立即改变所有可用设备的代码页，但无论已装入多少代码页，只能在当前国家的两个代码页中选择其一。用 MODE 命令可改变到其他代码页或者一次改变一个设备的代码页。

应用举例

CHCP

显示当前代码页号。

CHCP 850

改变当前代码页为 850 页。

CHDIR

见 CD。

Ⓔ CHKDSK

见第 27 章和 47 章

报告磁盘状态，报告和更正在 FAT 和目录结构中存在的错误，并报告文件碎片情况。如果用的是 DOS 6.2 或更高版本，可以用功能更强的 SCANDISK 代替本命令。

句法 1

CHKDSK [drive] [/F] [/V]

报告磁盘状态并报告和修改磁盘结构中的错误。

参数和选项

无参数	检查当前驱动器
drive	检查指定驱动器。
/F	如果可能，修改驱动器的 FAT 和目录结构中的错误。
/V	检测时列出所有文件。

注意

CHKDSK 命令的这一形式自动分析和报告磁盘空间的使用情况以及 FAT 和目录结构中的错误。若未发现错误，则只报告磁盘空间的使用状况。

CHKDSK 能发现 FAT 的目录结构中的逻辑错误，比如丢失的分配单元或文件的交叉链接。但它不能发现物理错误，如坏扇区(见 SCANDISK)。

如果 CHKDSK /F 找到丢失的分配单元(即被标记为含有数据的簇实际上却不属于任何文件)，它会询问用户是否将其转换成文件。若按 Y 键(或回车键)，CHKDSK 将每个簇链转换成一个名为 FILEnnnn. CHK 的文件。若按 N 键，CHKDSK 立即将丢失的簇变为可用的簇。

在使用了 ASSIGN 或 SUBST 命令的驱动器或网络驱动器上不要运行 CHKDSK 命令。

有文件被打开时不要使用 CHKDSK /F 命令。不能在 Microsoft Windows 或 DOS Shell 运行时或者在其他程序(如字处理器)中启动 CHKDSK /F。使用 CHKDSK /F 之前可能需要重新启动计算机以消除内存驻留程序。

在 DOS 6.2 及更高版本中，SCANDISK 程序比 CHKDSK 更受欢迎，因为它可以发现并修改更多的错误。

应用举例

CHKDSK

检查当前驱动器，报告 FAT 和目录结构错误以及磁盘空间的使用情况。

CHKDSK E: /F

检查驱动器 E，报告磁场空间的使用情况及 FAT 和目录结构错误，并尽量修改错误。

CHKDSK /V

检查当前驱动器，报告 FAT 和目录结构错误，报告磁盘空间的使用情况并列出磁盘上的每个文件。

句法 2

CHKDSK filespec

报告当前目录的文件碎块情况。

参数

filespec 检查指定文件碎块情况。

注意

含有碎块的文件位于磁盘上的几个不同区域。过多的文件碎块会影响系统性能。可以用 DEFRAG 消除文件碎块。

应用举例

CHKDSK *.*

列出当前目录中任何有碎块的文件。

Ⓑ CHOICE

见第 37 章

使用户可以在批处理程序运行中做出选择。

句法

CHOICE [/C:list] [/N] [/S] [/T:char,sec] [text]

参数和选项

/C:list 定义一串字符，使用户可以通过敲入字符进行选择。冒号是可选的。

/N 使 CHOICE 不显示任何提示。

/S 使 CHOICE 区分字母大小写。否则忽略字母大小写。

/T:char,sec 使 CHDICE 在等待 sec 秒后将 char 作为缺省选择。冒号是可选的。char 必须是 list 中的一个字符。sec 的范围是 0 到 99。这一参数中不要加任何空格。

text 指定在出现提示符前显示的文字。

注意

CHOICE 将 list 中的字符用逗号分开，并用方括号括起来，其后紧跟一个问号以提示用户输入，比如 [A,B]?。若不加/S，列表全部用大写字母。

若省略/C，CHOICE 用 Y/N 作为缺省列表。

用/N 选项使 CHOICE 显示 text 而不显示提示符。

如果 text 中含有正斜线(/)，必须用引号将 text 括起来。

加上/T 选项，CHOICE 在 sec 秒后将 char 作为选择。否则，它会永远等待直到用户输入表中的一个字符(或按 Ctrl+C 键或 Ctrl+Break 键)。对于用户的其他输入，CHOICE 响铃然后继续等待。

CHOICE 根据用户输入设置 ERRORLEVEL 参数的值：1 对应表中第一个字符，2 对应第二个，如此等等。如果用户按了 Ctrl+C 或 Ctrl+Break 键，CHOICE 将 ERRORLEVEL 设置为 0。如果发生错误，比如/

T 选项中的参数非法，CHOICE 就把 ERRORLEVEL 设置为 255。

用 IF 命令可以检查 ERRORLEVEL 的值，然后根据结果决定下一步的工作。

应用举例

CHOICE /C:ABC /S Which do you want ?

显示 Which do you want [A,B,C]? 并等待输入，合法输入有 A,B,C,Ctrl+C 或 Ctrl+Break 键。

CHOICE /C:ABC Which do you want?

显示 Which do you want[A,B,C]?并等待输入，合法输入是 A,B,C,a,b,c,Ctrl+C 或 Ctrl+Break 键。

CHOICE /T:y,3 Are you done

显示 Are you done[Y,N]? 并等待输入。合法输入是 Y,y,N,n,Ctrl+C 或 Ctrl+Break。若 3 秒钟内用户无输入，CHOICE 将 y 作为缺省输入，然后程序继续执行。

Ⓘ CLS

见第 20 章

清除屏幕显示。

句法

CLS

注意

CLS 将整个屏幕设置成当前背景色，命令提示符和光标位于屏幕左上角。

Ⓔ COMMAND

见第 43 章

启动 DOS 命令解释器 COMMAND.COM 的新版本。

句法

COMMAND [path] [ctty] [/E:n] [[/Y]/C command|/K command]

参数和选项

path	指定命令处理器的位置。
ctty	指定命令的输入输出设备。缺省是当前设备。
/E:n	以字节为单位设定新命令解释器的环境变量占用空间的大小。范围从 160 到 32 768。
/Y	有条件地执行由/C command 或/K command 指明的程序。
/C command	装入第二个命令解释器，执行 command，然后退出第二个命令解释器。
/K command	装入第二个命令解释器，执行 command，然后显示命令提示符。

注意

系统启动时，DOS 装入的缺省命令解释器是 COMMAND.COM，它位于启动盘的根目录下。关于用其他选项启动系统的更详细内容，如设置不同的路径或不同的命令处理器，参见 SHELL 命令。

COMMAND 命令把 COMMAND.COM 的新版本作为当前命令处理器的子进程进行，这叫做二级命令处理器。二级命令处理器用完以后，可以用 EXIT 命令结束之并返回前一级处理器。

关于如果改变主控设备，参见 CTTY 命令。

若指定了/C command，DOS 加载二级命令处理器，完成所指定的命令，然后返回前一级命令处理器。若指定/K command，DOS 加载二级命令处理器，完成指定的命令，然后提示输入新命令。除非输入 EXIT 命令，否则不退出二级命令处理器。无论/C 或/K 都可以执行批处理程序、内部命令和外部命令。

如果在/C 或/K 前加入/Y 选项，DOS 则显示程序名或 DOS 命令并附以[Y,N]?。按 Y 键则执行相应程序或命令，按 N 键则跳过它们。对于批处理程序，DOS 为文件的每一行显示类似的提示符，而由用户决

定是否执行该行命令。这一功能可用来查找批处理文件中的错误，每次执行一条命令直到发现错误之处。

命令解释器的每一个实例都有它们自己的一套环境变量。当 DOS 加载二级命令处理器时，完全拷贝父进程的环境。但是可以改变子进程的环境而不影响父进程。

如果没有/E 选项，DOS 为环境空间分配 256 字节内存。若所要求的环境空闲(显式使用/E 或用其缺省值)太小而装不下父进程的全部环境变量，DOS 则会增加环境空间的大小直到装下为止。

path 用于给环境变量 COMSPEC 赋值，由它决定需要重装 COMMAND. COM 的暂驻部分时 DOS 到何处寻找该文件。若忽略 path 参数，DOS 使用父进程环境的 COMSPEC。假如 path 是非法路径所指定的目录中不存在 COMMAND. COM，DOS 则显示出错信息并将 COMSPEC 置为\COMMAND. COM。

用 path 参数设置环境变量 COMSPEC 之后，子进程不再继承父进程的环境变量。

用 path 参数设置 COMSPEC 的唯一用处是重装 COMMAND. COM 的暂驻部分。它并不指示 DOS 在执行命令 COMMAND 时到任何去寻找文件 COMMAND. COM。若 COMMAND. COM 不在当前目录和查找路径中，就需要在命令名前指定路径名，如 D:\OLDDOS\COMMAND。

COMMAND 命令常与 SHELL 命令联用，参见本章。

应用举例

```
COMMAND C:\DOS /E:1024
```

从当前目录或查找路径中加载 COMMAND. COM 的第二副本，设置 COMSPEC 为 C:\DOS\COMMAND. COM 并为环境下 1 024 字节的空间。不继承父进程的环境变量。

```
COMMAND /C FOR %X IN (C:\ D:\) DO DIR %X *. BAT>PRN
```

将 FOR 命令的输出重定向到打印机。注意从"FOR"到"BAT"的文本是选项/C 中的命令。

```
E:\NEWDOS\COMMAND C:\NEWDOS COM1 /K C:\STARTUP. COM
```

加载 C:\NEWDOS\COMMAND. COM，设置 COMSPEC 为 C:\NEWDOS，将输入、输出设备改为 COM1，运行 STARTUP. COM，然后等待从 COM1 输入命令。二级命令处理器不继承父进程的环境变量。

```
COMMAND /Y /C C:\BATCHES\FILESAVE. BAT
```

加载 COMMAND. COM 的第二副本，执行 C:\BATCHES\FILESAVE. BAT，该文件中的每一行命令都由用户确认后才执行，最后退出二级命令处理器。

Ⓔ COMP

比较两个或两组文件。

注释： DOS 6 中没有提供这一命令，但有一个更好的文件比较工具，见 FC。

句法

```
COMP [filespec1] [filespec2] [/D|/A] [/L] [/N=n] [/C]
```

参数和选项

无参数	提示用户输入文件名和选项。
filespec1	要比较的第一个文件(组)。
filespec2	要比较的第二个文件(组)。
/D	以十进制形式报告文件的不同之处。
/A	以 ASCII 字符形式报告文件的不同之处。
/L	报告不同之处时用行号而不是偏移量。
/N=n	只比较文件的前 n 行。
/C	忽略字母大小写。

注意

COMP 逐字节地比较文件。如果两文件完全相同，COMP 报告文件比较结果正确。

用通配符比较多个文件时，COMP 将每个 filespec1 与具有同样通配字符的 filespec2 进行比较。例如，COMP A*.* B:*.*命令将 AFILE.DAT 与 BFILE.DAT 比较，将 ATYPE.TXT 与 BTYPE.TXT 比较，如此等等。

除非指定/N 选项，否则 COMP 不比较大小不同的文件。

在发现 10 处不同后，COMP 停止比较两个文件。

COMP 报告每一个不相同的字节。缺省情况下，COMP 以十六进制形式显示该字节的偏移量以及它在每个文件中的值。

如果用了/L 或/N 选项，COMP 报告错误时用行号而不是偏移量。这对于行末带有回车换行符的文本文件有用，但是对于没有行末标记的二进制文件，如图形或可执行文件并没有多少用处。

用/D 选项使 COMP 不用十六进制而是十进制报告字节的值。用/A 使 COMP 用 ASCII 字符报告字节值(不可打印字符以空格显示)。

Microsoft 正逐步以 FC 命令代替 COMP，FC 更新颖也更灵活。但如果想快速查看一下两文件是否相同，COMP 仍不失为一个选择。

应用举例

```
COMP AFILE.BAT BFILE.BAT
```

将 AFILE.BAT 与 BFILE.BAT(假定两文件大小相同)相比较并报告最多 10 处不同，对于不同之处以十六进制显示偏移量和两个不同的值。

```
COMP AFILE.BAT BFILE.BAT /D
```

同前例一样比较 AFILE.BAT 和 BFILE.BAT，但用十进制显示不同的字节。

```
COMP AFILE.BAT BFILE.BAT /A /N=19
```

比较 AFILE.BAT 和 BFILE.BAT，以 ASCII 形式报告不同之处；文件长度可以不同并最多比较 19 行。如果一个或两个文件少于 19 行，COMP 读完较短的文件后停止比较。

```
COMP AFILE.* BFILE.*
```

将每个名为 AFILE 的文件与具有相同扩展名的 BFILE 相比较。例如，COMP 将 AFILE.BAT 与 BFILE.BAT(如果存在)比较，AFILE.D1 与 BFILE.D1(如果存在)比较。

---

① COPY

见第 15 和第 40 章

拷贝一个或多个文件；拼接文件。

句法 1

```
COPY source [/A] [/B] [destination] [/A] [/B] [/V] [/Y|/-Y]
```

拷贝文件。

参数和选项

source	被拷贝的文件。
destination	拷贝目标的位置或文件名。
/A	将文件看作 ASCII 文本文件，其大小由文件结束符标记。
/B	将文件看作二进制文件，其大小记录在目录表项中。
/V	验证拷贝后的文件。
/Y	不加警告覆盖目的文件。
/-Y	在覆盖目的文件前要求用户确认。

注意

COPY 的这一形式产生与源文件日期、时间记录相同的目的文件，但为每个目的文件打开归档属性。

关于如何将新拷贝文件的时间、日期设置成当前时间和日期，参见句法 2。

destination 可以包含目录和新文件名。若不包含目录，COPY 使用当前目录。若不包含文件名，COPY 为新拷贝的文件取与原来文件相同的名字。如果用户省略 destination 参数，COPY 在当前目录中置入一个与原文件名字相同的新文件。

如果目的路径与源路径相同，destination 参数必须为新拷贝的文件赋予一个新名字（文件不能拷贝到其自身）。

在 DOS 6.2 之前的版本中，如果 destination 中指定的文件已存在，COPY 将其覆盖。这在效果上相当于删除了 destination 文件并用同样名字生成一个新文件（可以使用第三方软件商提供的文件恢复实用程序刚刚删除的文件）。

在 DOS 6.2 及更高版本中，如果 destination 指明的文件已存在，COPY 的行为与几个因素有关。缺省情况下，如果是从命令提示符下输入的命令，COPY 会询问用户是否删除目的文件，但如果是在批处理程序中执行本命令，COPY 则不加警告覆盖目的文件。然而，如果环境变量 COPYCMD 的值是/Y，COPY 不加提示覆盖目的文件。如果 COPYCMD 的值是/－Y，COPY 询问用户是否覆盖文件。在命令中加入/Y 或/－Y 选项可使 COPY 的缺省行为和环境变量 COPYCMD 的作用失败。

在 DOS 6.2 之前的版本中没有/Y 和/－Y 选项。

当 DOS 询问用户是否覆盖掉已存在的目的文件时，按 Y 键则覆盖文件，按 N 键则保留文件（不执行拷贝），按 A 键则覆盖这一文件及本次 COPY 命令随后涉及到的所有目的文件。换句话说，A 键终止对本次 COPY 命令中剩余文件的保护。

如果 source 指明了多个文件（利用通配符）而 destination 只提供了一个文件名，DOS 则将 source 中的文件连接（拼接）后放在 destination 中。关于连接文件的更多内容参见句法 2。

要分别拷贝多个文件并且每个文件用一个新名字，必须在 destination 中也使用通配符。比如，源文件是 *.BAT，那么目的文件可以是 *.OLD。

source 和 destination 可以是设备名。若 source 是 CON，COPY 从键盘接收输入直到用户输入文件结束符（可以按 Ctrl＋Z 或 F6 键输入这一字符）。若 destination 是打印端口，如已连接了打印机的 LPT1，DOS 就会打印源文件。

在这一句法中，若 source 后面有/A 或/B 选项，该选项也对 destination 起作用，除非 destination 后有另一个选项。

COPY 从目录表项处获知文件有多少字节，当对 source 使用了/A 选项时，如果 COPY 命令在达到整个文件结尾前遇到文件结束标记，则停止拷贝，文件结束标记不拷贝。若对 source 使用/B 选项，COPY 将拷贝完整个文件。拷贝文件时的缺省选择是/B；而当连接文件或源文件是 CON 时/A 是缺省选项。

二进制文件，如 EXE 或 COM 文件中可能会含有作为数据的文件结束符。拷贝时用二进制方式很重要，这可以防止截断文件。

如果对 destination 使用/A 选项，COPY 在文件尾加入文件结束符；用/B 选项不加。

应用举例

COPY MYFILE.DAT YOURFILE.DAT

拷贝 MYFILE.DAT 的内容到 YOURFILE.DAT 中，生成新文件或覆盖已存在的文件。两个文件都在当前目录中。在目录表项的文件 MYFILE.DAT 的大小决定了 COPY 要拷贝多少个字节。除非 MYFILE.DAT 结尾是文件结束符，否则 YOURFILE.DAT 不会以文件结束符结尾。

COPY MYFILE.* YOURFILE.*

将每个与源文件名相匹配的文件拷贝到一个有相同扩展名的名为 YOURFILE 的文件中。例如，MYFILE.A1 拷贝到 YOURFILE.A1。

SET COPYCMD=/Y

COPY /－Y MYFILE. * YOURFILE. *

第一个命令为COPYCMD 赋值/Y，关掉了所有随后的 COPY、XCOPY 和 MOVE 命令的文件保护功能。这些命令在覆盖文件时将不再发出警告。第二个命令把当前目录中所有匹配 MYFILE. * 的文件拷贝到当前目录，并将新文件命名为 YOURFILE. *。该命令使 COPYCMD 的作用失效，要求保护所有已存在的 YOURFILE. *，因此 COPY 在覆盖文件之前会请用户做出选择。

句法 2

COPY source1＋ [source2] [＋...] [/A] [/B] [destination] [/A] [/B] [/V] [/Y|/－Y]

连接文件。

参数和选项

source1	第一个或第一组被连接的文件。
source2	另一个或另一组被连接的文件。
destination	连接后的数据置于该文件或文件组中。
/A	将文件视为 ASCII 文本文件。
/B	将文件视为二进制文件。
/V	连接后对文件进行验证。
/Y	不加警告覆盖目的文件。
/－Y	覆盖目的文件时要求用户确认。

注意

如果需要，可以重复＋source2 部分。源文件名中可包含通配符。

因为目的文件不同于任何源文件，COPY 为其赋予当前日期和时间。可以用一个假连接为文件指定当前日期和时间(见应用举例)。

如果忽略 destination 参数，COPY 将所有源文件连接到第一个实际存在的源文件中。例如，如果 source1 不存在，但 source2 存在，COPY 将连接后的数据存于 source2 中；如果 source1 和 source2 都不存在，但 source3 存在，则由 source3 接收数据。

在这一句法中，如果在两个或更多的文件名中使用了通配符，COPY 把具有相同通配符的文件连接起来。例如，COPY AFILE. * ＋BFILE. * CFILE. * 连接 AFILE. DAT 和 BFILE. DAT，产生或替换 CFILE. DAT，并连接 AFILE. NEW 和 BFILE. NEW，产生或替换 CFILE. NEW。COPY AFILE. * ＋BFILE. * 将 AFILE. BAT 和 BFILE. BAT 连接起来存入 AFILE. BAT(如果该文件已存在)。

/A 是本句法的缺省选项。/A 或/B 选项对其前面的和所有后续文件名起作用，直到 COPY 遇到一个后面跟有/A 或/B 选项的文件名为止。

与句法 1 一样，若对源文件使用/A 选项，COPY 遇到文件结束符即停止拷贝，且不拷贝结束符。如果没有文件结束符，或是使用了/B 选项，则由该文件在目录中的表项决定文件的大小。如对目的文件使用了/A 选项，COPY 在文件结尾处加上文件结束符；若是/B 选项，那么除了从源文件中拷贝来的文件结束符之外，COPY 不再另加文件结束符。

当把数据连接到一个源文件中时，可能会出现信息“Content of destination lost before copy.”。这是正常的信息，并不意味着将要丢失数据。这说明 COPY 发现目的文件同时也可以作为源文件，但因为已作为目的文件，COPY 将不把它用作源文件。

应用举例

COPY C:\DS1\MYFILE. * NEWFILE. CAT

在当前目录中产生(或覆盖)文件 NEWFILE. CAT 并将指定的源文件连接后存放其中。对于每个源文件，COPY 在遇到文件结束符时停止拷贝。在文件 NEWFILE. CAT 的结尾会加上一个文件结束符。

COPY MYFILE. DAT＋YOURFILE. DAT OURFILE. DAT /V /－Y

连接文件 MYFILE.DAT 和 YOURFILE.DAT，覆盖 OURFILE.DAT 并验证拷贝结果。COPY 将这些文件视为 ASCII 文本文件，在新文件结尾处加上文件结束符。如果 OURFILE.DAT 文件已存在，COPY 会询问是否将其覆盖。

COPY MYFILE.DAT+YOURFILE.DAT+OURFILE.DAT

如果文件 MYFILE.DAT 存在，连接 MYFILE.DAT，YOURFILE.DAT 和 OURFILE.DAT 到 MYFILE.DAT 中。若 MYFILE.DAT 不存在，则连接 YOURFILE.DAT 和 OURFILE.DAT 到 YOURFILE.DAT 中。若 MYFILE.DAT 和 YOURFILE.DAT 都不存在，就按连接的方法将 OURFILE.DAT 拷贝到自身。COPY 视所有文件为 ASCII 文本文件，并在新文件结尾加上文件结束符。

COPY MYFILE.DAT/B+YOURFILE.DAT OURFILE.DAT

连接文件 MYFILE.DAT 和 YOURFILE.DAT，产生或覆盖 OURFILE.DAT。所有文件被视为二进制文件，拷贝源文件中的所有字符，新文件末尾不添加文件结束符。

COPY MYFILE.DAT+YOURFILE.DAT /B OURFILE.DAT

连接文件 MYFILE.DAT 和 YOURFILE.DAT 生成或覆盖 OURFILE.DAT。COPY 将 MYFILE.DAT 作为 ASCII 文本文件，YOURFILE.DAT 和 OURFILE.DAT 作为二进制文件。

COPY MYFILE.DAT+,, /B NEWFILE.DAT

文件 MYFILE.DAT 不与任何文件连接(由两个逗号表示)，产生或覆盖 NEWFILE.DAT。结果是将 MYFILE.DAT 拷贝到 NEWFILE.DAT，并将当前时间和日期赋予新文件。两个文件都作为二进制文件。两个逗号表示源文件列表结束，否则 COPY 会把 NEWFILE.DAT 也当成源文件。

COPY MYFILE.DAT +,,/B

文件 MYFILE.DAT 不与其他文件连接，覆盖 MYFILE.DAT。结果是更新了文件的时间和日期记录。/B 保证了 COPY 拷贝整个文件，包括文件结束符，但并不在文件结尾加文件结束符。+,,是必需的，指示这是一个连接而非拷贝。

© COUNTRY

见第 49 章

为日期和时间显示、货币符号、排序、标点、数字分隔符和代码指定某个国家的格式。

句法

COUNTRY=country[,[page] [,filespec]]

参数

country	相应国家的代号。
page	指定一个代码页(即字符集)。
filespec	含有国家信息的文件。

注意

缺省国家(如果 CONFIG.SYS 中没有 COUNTRY 命令)是美国。

国家代号、日期和时间的格式及正确的代码页已写入 COUNTRY 命令的屏幕帮助信息之中。

与每个国家相关的有两个可能的代码页。一个是缺省代码页，另一个是候选代码页。如果有省略 page 参数，DOS 使用缺省代码页。

诸如对日期时间格式的改变在 CONFIG.SYS 中的 COUNTRY 命令执行后立即生效，但必须在执行完 CHCP 命令中提及的各步骤之后才可以使用软件代码页。除非加载驱动程序并装入代码页，否则 DOS 仍使用缺省的硬件代码页。

如果省略 filespec 参数，DOS 在启动盘根目录下的 COUNTRY.SYS 中查找有关国家设置的信息。

应用举例

COUNTRY=39

将国家改为意大利;使用缺省代码页 850(采取必要步骤之后);在启动盘的\COUNTRY.SYS 中查找关于国家的信息。

COUNTRY=351,860,C:\DOS\COUNTRY.SYS

国家设置成葡萄牙;使用候选代码页 860(采取必要步骤之后);在 C:\DOS\COUNTRY.SYS 中查找关于国家的信息。

COUNTRY=047,C:\OLD\SPECIAL.SYS

国家设置为挪威;使用挪威的缺省代码页 850(采取必要步骤之后);在文件 C:\OLD\SPECIAL.SYS 中查找关于国家的信息。

---

① CTTY

见第 48 章

改变控制台设备。

句法

CTTY device

参数

device　　指定新的控制台设备。

注意

控制台设备接收命令的输入并显示命令的输出。正常情况下,这一设备是 CON,它代表键盘和显示器。

设备可以是 PRN,LPT1,LPT2,LPT3,CON,AUX,COM1,COM2,COM3 或 COM4。LPT1 到 LPT3 指的是并行端口。PRN 与 LPT1 相同。COM1 到 COM4 指的是串行端口。AUX 指辅助端口。

如果指定一个 COM 端口,必须在使用 CTTY 之前用 MODE 命令建立端口通讯参数。

若控制台设备不是 CON,DOS 无法响应键盘输入。用户可能需要重新启动以将控制归还键盘和显示器。

另一种改变控制台设备的办法参见 COMMAND 命令。

本命令并不转移所有的输入输出,而只是转移 DOS 命令的标准输入输出。例如,用户仍可使用键盘热启动。

应用举例

CTTY AUX

用一个与辅助端口相连的远程终端作为命令的输入输出设备。

CTTY CON

将输入输出功能转移回键盘和显示器(从当前控制台设备输入)。

---

① DATE

见第 12 章

显示并允许修改系统日期。

句法

DATE [date]

参数

无参数　　以标准格式显示日期并询问新日期。

date 日期改成指定的日期。

注意

系统日期会影响到目录项并被用于 MSBACKUP、XCOPY 和许多其他应用程序。

DOS 询问新日期时，按回车键保持当前日期。

COUNTRY 命令决定日期格式。缺省情况下，输入日期的格式按照美国标准：mm/dd/yy，mm-dd-yy 和 mm.dd.yy。在每种格式中，yy 可以是从 80 到 99 的两位数(解释为 1980)或是从 1980 到 2099 的四位数。

DOS 依据系统日期计算星期，如 Sun(星期日)或 Mon(星期一)；不要试图输入星期。

应用举例

```
DATE
```

显示系统日期并询问新日期。

```
DATE 12-15-97
```

改变系统日期为 1997 年 12 月 15 日。

```
DATE 12/15/2003
```

改变系统日期为 2003 年 12 月 15 日。

Ⓔ DBLSPACE

参见第 34 和第 45 章

压缩硬盘或软盘；管理压缩盘。

句法 1

```
DBLSPACE
```

启动 DoubleSpace 系统设置程序，由其加载 DoubleSpace 并压缩和装入一个驱动器。

警告： 在 DOS 6.22 中，DriveSpace 已经替代了 DoubleSpace。如果已经使用了 DoubleSpace 处理压缩文件，可以继续使用 DBLSPACE 命令，也可以转换到 DriveSpace。如果准备使用压缩文件，那么用 DriveSpace 代替 DoubleSpace(参见命令 DRVSPACE)。

注意

第一次使用 DBLSPACE 必须用本句法。DBLSPACE 为压缩文件设置系统，在启动盘的根目录下放置 DBLSPACE 的核心程序 DBLSPACE.BIN 和其他文件。它还把命令 DEVICEHIGH=DBLSPACE.SYS 加入 CONFIG.SYS 中，这样，如果条件允许，DBLSPACE.BIN 可被移到高端内存中；否则，它被移到常规内存的底部。DBLSPACE 还要压缩至少一个硬盘，否则不能安装 DBLSPACE。

当再次使用句法 1 时，加载 DoubleSpace 程序并显示一个带有下拉菜单的对话框，从中可以选择压缩其他驱动器或管理压缩盘。

安装成功后，可以使用 DBLSPACE 命令的其他句法完成 DoubleSpace 的功能而无需打开对话框。

DoubleSpace 在压缩已存在的数据时，会生成一个名叫 DBLSPACE.000 的大文件，其中含有压缩过的数据(该文件具有系统、隐含和只读属性)。原来的驱动器上现在有 DBLSPACE.000、其他不适合压缩的文件(如 Windows 磁盘交换文件)和一些剩余空间。压缩后的数据文件称为压缩卷文件(CVF)并通过原来的盘符来访问它们。原来的驱动器被称为宿主驱动器，它通过 DoubleSpace 指派的另一个盘符来访问。(如果不知道 DoubleSpace 为宿主驱动器指定了哪个盘符，用 DBLSPACE /INFO 或/LIST 句法 9 和句法 10)。

为压缩盘和宿主盘指派盘符和建立访问方式称为“装入”(mount)压缩盘。当启动系统时，DoubleSpace 自动装入所有压缩硬盘。对于 DOS 6.2，同时自动装入软盘。若是 DOS 6.0，必须在每次插入软盘时装入它(见句法 11)。如果压缩盘未被装入，就无法对其进行访问，因为没有指派盘符。

DoubleSpace 不能压缩一个无剩余空间的磁盘。启动盘上至少要有 1.2MB 的剩余空间;其他驱动器,包括软盘,至少 1.1MB。

如果原来的盘上有一个 SENTRY 目录,那么压缩后在压缩盘和宿主盘上都有 SENTRY 目录。可是,在进行压缩之前删除的文件将无法恢复。压缩之后删除的文件仍可像平常一样利用删除卫士(Delete Sentry)恢复。但如果要保护未被压缩的宿主驱动器,应在被删除卫士(Delete Sentry)保护驱动器中加入新的驱动器符。

如果驱动器上原来有一个 PCTRACKR.DEL 文件,它会被保留在宿主驱动器上,但压缩之前删除的任何文件都可能无法恢复,因为它们占用的簇已被 CVF 文件重新使用。压缩后删除的文件仍可像平常一样利用删除跟踪(Deletion-tracking)恢复。若要保护未被压缩的宿主盘,应在被删除跟踪(Deletion-tracking)保护的驱动器中加入新的驱动器符。

如果在 SMARTDRV 命令中列出一个压缩盘,则会有一条消息告诉用户必须使用缓冲加速宿主盘。无需更改这一命令,DoubleSpace 和 SmartDrive 会相互合作通过宿主盘对压缩盘进行缓冲。

警告: *不要对压缩盘进行写缓冲。如果系统在写盘完成之前崩溃,那么延迟写入会造成压缩盘上的数据丢失。若在升级到 DOS 6.2 或更高版本之前就已安装了 SmartDrive,那么很可能已经为压缩盘提供了缓冲。改变 SmartDrive 设置使之不对压缩盘进行写缓冲。*

以下的命令格式在命令提示符下管理 DoubleSpace,所有这些功能也可通过 DoubleSpace 对话框完成。输入不带参数的 DBLSPACE 命令即可打开对话框(句法 1)。

句法 2

DBLSPACE /CHKDSK [/F] [drive]

检查压缩的目录结构(只用于 DOS 6.0)。

参数和选项

/CHKDSK	使用 CHKDSK 功能,其缩写形式为/CHK。
/F	修改压缩盘上发现的错误。
drive	指定要检查的压缩盘。缺省盘是当前盘。

注意

DoubleSpace 的/CHKDSK 功能已被 DOS 6.2 以上的 SCANDISK 程序代替。

在 DOS 6.0 中,本命令检查压缩卷文件的内部结构。用 DOS 的 CHKDSK 命令检查压缩盘的文件分配表。

应用举例

DBLSPACE /CHK

如果当前盘是压缩盘,检查其内部结构。报告错误但不修改错误。

DBLSPACE /CHK /F D

检查压缩盘 D 的内部结构,如果可能,修改错误。

句法 3

DBLSPACE /COMPRESS drive1 [/NEWDRIVE=drive 2] [/RESERVE=size] [/F]

压缩一个已存在的硬盘,软件或其他可移动媒体上的文件并装入(mount)驱动器。

参数和选项

/COMPRESS	使用 COMPRESS 功能,其缩写形式为/COM。
drive1	指定一个将压缩的现存驱动器。
/NEWDRIVE=drive2	指定旧宿主驱动器使用的盘符。其缩写形式为/NEW。可以用句法 10 查出还有哪个字母可用。若不用/NEW 选项,则由 DoubleSpace 为新盘选择一个盘符。

/RESERVE=size 以兆字节为单位指定要在宿主驱动器上保留的不压缩的空间。其缩写形式为/RES。省略/RES时，如果可能，DoubleSpace在硬盘上留下2MB空间。

/F 略去最后的报告自动返回命令提示符。

注意

DoubleSpace在压缩软盘时，在宿主驱动器上生成一个名为READTHIS.TXT的文件。该文件向接收者说明磁盘中有压缩数据以及如何访问这类数据。在安装了DoubleSpace的DOS 6.2以上的系统中不会出现READTHIS.TXT文件，因为压缩盘会被自动装入(mount)。它只出现于使用DOS 6.0及更低版本和没有安装DoubleSpace的系统中。

无论是否在此处指定了/RES选项，都可在以后增加或减少未压缩空间。

关于压缩磁盘时状况的更多信息，参考句法1。

应用举例

DBLSPACE /COM A

压缩驱动器A中软盘。压缩后为宿主驱动器指派一个可用驱动器符。

DBLSPACE /COM D /RES=2.5 /NEW=L

压缩D盘，保留2.5MB的剩余空间(如果可能)。为宿主驱动器指派盘符L。

句法4

DBLSPACE /UNCOMPRESS drive

对驱动器进行解压缩。

参数与选项

/UNCOMPRESS 使用UNCOMPRESS功能。

drive 指定要解压缩的驱动器。

注意

UNCOMPRESS功能解压缩CVF中的数据，并存放到宿主驱动器上。如果宿主驱动器上没有足够的空间存放解压缩后的数据，用户可能需要删除一些压缩文件或把部分文件移到其他盘上。解压缩后，压缩盘被卸出并删除，宿主驱动器的原名得以恢复。

最后一个压缩盘被解压缩后，DBLSPACE.BIN从内存中删除，但并不从根目录中删除。

句法5

DBLSPACE /CREATE drive1 [/NEWDRIVE=drive2][/SIZE=size1|/RESERVE=size2]

利用非压缩盘上的空闲空间生成一个新的空压缩盘。

参数和选项

/CREATE 使用CREATE功能，其缩写形式为/CR。

drive1 指定宿主驱动器。

/NEWDRIVE=drive2 为新的CVF指定驱动器符。缩写为/NEW。若不加/NEW选项，DBLSPACE则为新的CVF指定驱动器符。

/SIZE=size1 为压缩卷文件分配的磁盘空间，单位为兆字节数(以压缩前为准)，其缩写形式为/SI。

/RESERVE=size2 为非压缩盘留下多少兆字节的空闲空间，其缩写形式为/RES。

注意

不能对软盘使用/CREATE选项。

当从空闲空间而不是已存在的数据中生成一个CVF时，DoubleSpace为新卷指派另一个盘符，宿主盘保留原来的盘符。

可以在一个已有压缩卷文件驱动器上生成一个新的空卷。原来的 CVF 称为 DBLSPACE.000 并使用原来的盘符。新的 CVF(假定是盘上的第二个 CVF)叫做 DBLSPACE.001 并使用新盘符。剩下的未压缩空间在宿主驱动器上,宿主驱动器仍使用 DBLSPACE.000 生成后指派的盘符。

如不加/SIZE 或/RES 选项,DoubleSpace 只留下 1MB 的空闲空间。

应用举例

DBLSPACE /CR E

在宿主驱动器 E 上生成一个空的压缩卷文件,留下 1MB 空闲空间。为新的压缩卷文件指派一个可用的盘符。

DBLSPACE /CR E /NEW=L /RES=1.5

在宿主驱动器 E 上生成一个空的 CVF。如果可能,在 E 盘上留 1.5MB 的空闲空间,为新的 CVF 指派盘符 L。

句法 6

DBLSPACE /DEFRAGMENT [drive] [/F]

消除压缩盘上的存储碎块。

参数和选项

/DEFRAGMENT	使用 DEFRAGMENT 功能,其缩写形式为/DEF。
drive	要整理的压缩盘。缺省盘是当前盘。
/F	进行更全面的存储碎块清除操作(只能用于 DOS 6.2 以上)。

注意

为了提高压缩盘的性能而整理存储碎块是不必要的,而且它能合并盘上的空闲空间。若要减少压缩盘,应首先使用 DBLSPACE /DEF。

如果在一个 DoubleSpace 驱动器上运行 DOS 的 DEFRAG 命令,运行结束后 DOS 立即自动运行 DBLSPACE /DEF。

应用举例

DBLSPACE /DEF /F

消除当前(压缩)驱动器的存储碎块。

句法 7

DBLSPACE /DELETE drive

删除一个压缩盘及上面的所有文件。

参数与选项

/DELETE	使用 DELETE 功能,其缩写形式为/DEL。
drive	指定要删除的压缩盘。

注意

删除一个 DoubleSpace 盘的同时也将其卸出(unmount)。如果删除 DBLSPACE.000,宿主驱动器恢复以前的盘符。如果删除任何其他的 CVF,被删除卷的盘符可移作它用。

如果在删除了 DoubleSpace 盘后改变主意,使用 DOS 的 UNDELETE 命令也许能将其恢复。它在宿主驱动器上寻找一个已被删除的名为 DBLSPACE.nnn 的文件。

用 UNDELETE 成功恢复文件后,必须将其装入(mount)才能读写数据。见句法 11。

应用举例

DBLSPACE /DEL F

删除压缩盘 F。

句法 8

DBLSPACE /FORMAT drive

格式化压缩盘。

参数和选项

/FORMAT 使用 FORMAT 功能，其缩写形式为/FOR。

drive 指定要格式化的压缩盘。

注意

格式化压缩盘会删除其中的所有文件。即使改变主意也无法恢复压缩盘上被格式化的数据。

应用举例

DBLSPACE /FOR G

格式化压缩盘 G，删除所有数据。

句法 9

DBLSPACE [/INFO] drive

显示关于压缩盘的信息。

参数与选项

/INFO 使用 INFO 功能。当不指明其他 DBLSPACE 功能时，这是缺省值。

drive 显示该驱动器的信息。

注意

显示的信息包括宿主驱动器，CVF 的文件名，总空间，已用空间，CVF 和宿主驱动器上的空闲空间，实际平均压缩率，及压缩估算比率。CVF 上的空闲空间是使用压缩估算比率对 CVF 中剩余空间所能容纳的未压缩字节数的估计值。

应用举例

DBLSPACE D

显示压缩盘 D 的信息。

DBLSPACE /INFO

显示当前驱动器的信息。

句法 10

DBLSPACE /LIST

列出并描述计算机上所有非网络驱动器，包括 DoubleSpace 盘。

参数和选项

/LIST 使用 LIST 功能，其缩写形式为/L。

注意

本句法列出每个可用的驱动器符并描述其状态，如 Local hard drive(本地硬盘)，Compressed hard drive (压缩硬盘)，Floppy drive (软盘)，Removable-media with No disk in drive(未放入磁盘的可移动媒体)，Available for DoubleSpace(可用于 DoubleSpace)，等等。(RAM 盘被描述为本地硬盘。)如果合适，显示每个驱动器上已用空间和空闲空间。对于每个已安装(mount)的压缩盘，显示压缩卷名和宿主驱动器。

句法 11

DBLSPACE /MOUNT[=nnn] drive1 [/NEWDRIVE=drive2]

装入(mount)一个压缩卷。

参数与选项

/MOUNT 使用 MOUNT 功能。其缩写形式为/MO。

nnn	被装入(mount)卷的扩展名。缺省是 000。
drive1	指定一个驱动器,该驱动器上含有要被装入的压缩卷文件。
/NEWDRIVE=drive2	指定将被指派的驱动器符,其缩写形式为/NEW。若没有/NEW 选项,DoubleSpace 则自动指派一个可用驱动器符。

注意

在装入(mount)文件 DBLSPACE.000 时,宿主驱动器符被指派给压缩盘,另一个新字母赋予宿主驱动器。而装入任何其他 DBLSPACE.nnn 文件时,系统会为压缩盘指派新的字母。

DBLSPACE 生成压缩盘后会自动将其装入。在压缩盘上执行某些特定功能时会自动卸出再装入该盘。系统启动时会自动装入压缩的硬盘,只有在用户卸出或删除并恢复压缩硬盘的情况下才需要重新装入硬盘(见句法 14 和 17)。

DOS 6.2(或更高版本)还会自动装入软盘。但如果用的是 DOS 6.0,需要在每次插入一个软盘到驱动器中时进行装入。

应用举例

```
DBLSPACE /MO A
```

装入驱动器 A 中的磁盘,将字母 A 指派给 DBLSPACE.000,将另一个可用字母指派给磁盘的未压缩部分。

```
DBLSPACE /MO=001 E /NEW=L
```

装入一个压缩盘,盘符 L 指派给 E:\DBLSPACE.001,字母 E 作为宿主驱动器的盘符。

句法 12

```
DBLSPACE /RATIO[=r.r] [drive|All]
```

改变一个所有压缩盘的估计压缩率。

参数与选项

/RATIO=r.r	指定新的压缩率,从 1.0 到 16.0。缺省值是驱动器的综合压缩率,其缩写形式为/RA。
drive	指定一个驱动器。
/ALL	用此选项改变所有当前已装入的压缩盘的估计压缩比。

注意

DoubleSpace 利用估计压缩率估计在压缩时能有多少字节的未压缩数据被压缩进 CVF。这将影响诸如 DIR 和 DOS Shell 等的报告结果。

如果未指定 drive 也没有使用/ALL 选项,DoubleSpace 就会改变当前驱动器的压缩率。

用句法 9(/INFO)查看驱动器估计压缩率和实际压缩率。

应用举例

```
DBLSPACE /RA /ALL
```

把当前已装入的所有压缩盘的估计压缩率改成各自的平均压缩率。

```
DBLSPACE /RA=3 D
```

将压缩盘 D 的估计压缩率改成 3.0 比 1。

句法 13

```
DBLSPACE /SIZE[=size1 | /RESERVE=size2] drive
```

改变压缩卷文件的大小。

参数与选项

/SIZE	使用 SIZE 功能,其缩写形式为/SI。
size1	以兆字节为单位的 CVF 的新长度(压缩前)。不能与/RESERVE 同时使用。

/RESERVE=size2 以兆字节为单位指定要为宿主驱动器保留多大的空闲空间，其缩写形式为/RES。不要与 size1 同时使用。如果 size2 是 0，压缩盘的磁盘空间为可能情况下的最大值。

drive 指定需要调整大小的压缩盘。

注意

如果既不使用 size1 也不使用/RES，DoubleSpace 就将压缩盘减到尽可能小。

应用举例

DBLSPACE /SI=30 D

在压缩前将压缩盘 D 的大小改为 30MB(如果可能)。

DBLSPACE /SI /RES=10 D

改变压缩盘的大小，为宿主驱动器留下 10MB 的空闲空间(如果可能)。在此之前应消除压缩盘上的碎块。

DBLSPACE /SI /RES=0 D

使压缩盘 D 尽可能大。

DBLSPACE /SI D

使压缩盘 D 尽可能小。在此之前应首先消除压缩盘上的碎块。

句法 14

DBLSPACE /UNMOUNT [drive]

卸出(unmount)一个驱动器。

参数和选项

/UNMOUNT 使用 UNMOUNT 功能，其缩写形式为/U。

drive 指定要卸出的驱动器。缺省时为当前驱动器。

注意

卸出的驱动器在重新装入之前不能访问。如果宿主驱动器符(也可简称盘符)被交换给了 CVF，那么宿主驱动器将恢复原来的驱动器符，现在的驱动器符可作它用。若没有交换驱动器符，那么 CVF 的驱动器符可作它用。

从驱动器中取出软盘前不需要将其从系统中卸出。

应用举例

DBLSPACE /U D

卸出压缩盘 D。如果 D 已被指派给 DBLSPACE.000，则它现在代表宿主驱动器。否则，宿主驱动器符不变而 D 成为可用的驱动器符。

句法 15

DBLSPACE [/AUTOMOUNT=0 |1 |X] [/DOUBLEGUARD=0|1] [drive1 /HOST=drive2] [/LASTDRIVE = x] [/MAXFILEFRAGMENTS = n] [/MAXREMOVABLE-DRIVES=n] [/ROMSERVER=0|1] [/SWITCHES=F|N]

改变 DBLSPACE.INI 中的缺省设置。

参数和选项

/AUTOMOUNT 表示 DoubleSpace 是否自动装入软盘。

/DOUBLEGUARD 允许或禁止 DoubleGuard 功能。DoubleGuard 的作用是不断检查 DoubleSpace 的内存缓冲区是否被破坏。

/HOST 指定压缩盘(drive1)的缩主盘名(drive2)。

/LASTDRIVE	DoubleSpace 装入(mount)驱动器时可以使用的最后一个驱动器名。
/MAXFILEFRAGMENTS	指定一个 CVF 在其宿主驱动器上可被分成的最大碎块数。
/MAXREMOVABLEDRIVES	指定在启动期间 DoubleSpace 应另外再为多少个压缩盘预留空间。
/ROMSERVER	允许或禁止在 ROM BIOS 中检查是否存在 Microsoft 实时压缩界面(MRCI)的服务程序。
/SWITCHES	控制是否允许用户在启动期间跳过 DoubleSpace。

注意

DBLSPACE. INI 中含有 DoubleSpace 的缺省设置。因为它是一个隐含、系统、只读文件,无法直接编辑(去掉隐含、系统和只读属性会很危险)。用这里提到的命令来改变 DBLSPACE. INI 的设置。

虽然不能编辑,但仍能用 DOS 的 EDIT 查看 DBLSPACE. INI 文件,这样就可以查出这些参数的当前设置。

在 DOS 6.2 以上,DoubleSpace 缺省情况下自动装入可移动驱动器(如软盘)。DoubleSpace 必须为每个可能被自动装入的驱动器保留内存空闲。命令 DBLSPACE /AUTOMOUNT=0 设置 DBLSPACE. INI 以使可移动驱动器不被自动装入,这样就节省了内存空间。DBLSPACE /AUTOMOUNT=1 恢复缺省设置。它可以使用 DBLSPACE /AUTOMOUNT=A,DBLSPACE /AUTOMOUNT=B 等命令专门指定某个可移动驱动器被自动装入。(装入一个未被自动装入的压缩盘的命令,参见句法 11)。

在 DOS 6.2 或更高版本中,DoubleGuard 不时地检查 DoubleSpace 的内存缓冲区,如果发现其被侵占,就停止系统的运行以防止压缩盘被破坏。DoubleGuard 会略微降低一些系统速度。可以用命令 DBLSPACE /DOUBLEGUARD=0 禁止使用 DoubleGuard。命令 DBLSPACE /DOUBLEGUARD=1 允许 DoubleGuard 工作。

警告: 禁止 DoubleGuard 可能会导致压缩数据受损。

如果用户看到信息 Compressed drive x is currently too fragmented to mount,说明作为驱动器 x 的 CVF 被分成的碎块数超过了 DBLSPACE. INI 中所指定的 MAXFRAGMENTS 值。除非解决这个问题,否则无法装入或使用压缩盘。可以增加允许的碎块数,方法是输入命令 DBLSPACE /MAXFRAGMENTS=n,其中 n 是新的碎块数。另一个办法是去掉 CVF 的文件属性,消除宿主驱动器的碎块,然后恢复 CVF 的属性(非常重要)。

参数 MAXREMOVABLEDRIVES 决定了启动之后 DoubleSpace 可额外处理的压缩盘数(无论是否为可移动驱动器)。这不仅控制了能从可移动驱动器上装入(mount)多少个压缩卷,还决定在不重新启动计算机的情况下最多能生成多少个压缩卷。缺省的最大值是 2。降低这个值可以节省内存空间,但会影响生成新压缩盘的能力。

MRCI 检测功能在 ROM BIOS 中查找 Microsoft 实时压缩界面的服务程序。某些系统使用 MRCI 作为 DoubleSpace 压缩功能的底层。如果系统的 ROM BIOS 中含有 MRCI,要将这一选项设为 1。但这种检测可能会妨碍没有 MRCI 的 ROM BIOS,所以,除了确实需要,不要使用 MRCI 检测功能。命令 DBLSPACE /ROMSERVER=1 允许 MRCI 检测功能。用 DBLSPACE /ROMSERVER=0 禁止检测功能。

在缺省情况下,如果用户在启动期间按了 Ctrl+F5 键,DOS 显示 CONFIG. SYS 和 AUTOEXEC. BAT 的每一行命令并询问是否执行,同时还会询问是否装入 DoubleSpace。如果启动期间按下 Ctrl+F8 键,DOS 则跳过 CONFIG. SYS、AUTOEXEC. BAT 和 DBLSPACE. BIN 而不询问。可以用命令 DBLSPACE /SWITCHES=N 禁止跳过 DoubleSpace 部分。在这种情况下,按 Ctrl+F5 键仍有条件地执行 CONFIG. SYS 和 AUTOEXEC. BAT,而 Ctrl+F8 键仍然跳过 CONFIG. SYS 和 AUTOEXEC. BAT。

命令 DBLSPACE /SWITCHES=F 缩短 DOS 等待用户按跳过键的时间,使启动速度略微提高。

一旦输入了/SWITCHES=N 或/SWITCHES=F,就无法再用命令把它们从 DBLSPACE. INI 中去掉。必须直接编辑文件去掉 SWITCHES 参数,使设置回到缺省状态。

关于能禁止用户跳过 CONFIG.SYS 和 AUTOEXEC.BAT 的命令参见 SWITCHES。

Ⓔ DEBUG

启动一个程序，利用它可以对可执行文件进行测试和查错。

句法

DEBUG [filespec [parameters]]

参数

无参数	启动 DEBUG 但不加载程序。
filespec	加载一个程序以进行调试。filespec 中可包含路径。必须含有扩展名(通常是.EXE 或.COM)。
parameters	提供给被加载程序的所有参数。

注意

DEBUG 启动后，显示自己的命令提示符：一个连字符(-)。在连字符后输入 DEBUG 命令。命令 Q(其意为退出)可退出 DEBUG。

其他 DEBUG 命令参见 DEBUG 的屏上帮助。

应用举例

DEBUG

启动 DEBUG，不加载程序。

DEBUG MYCAL.COM JAN

启动 DEBUG，加载当前目录中名为 MYCAL.COM 的程序，并将参数 JAN 传递给 MYCAL.COM。

Ⓔ DEFRAG

见第 33 章

重新组织驱动器上的文件以提高文件访问性能，对目录项排序。

句法

DEFRAG [drive] [/F|/Q|/U] [/Sorder] [/B] [/H] [/SKIPHIGH] [/LCD|/BW|G0]

参数和选项

无参数	打开 DEFRAG 对话框使用户可以选择驱动器和其他选项。
drive	需要优化的驱动器。
/F	消除文件和空闲空间的碎块。
/Q	只优化空闲空间。
/U	只消除文件碎块。
/Sorder	对目录项排序。order 指定排序方式(见注释)。
/B	DEFRAG 结束后重新启动计算机。
/H	允许 DEFRAG 移动隐含文件。
/SKIPHIGH	在常规内存中加载 DEFRAG 程序。
/LCD	用 LCD 配色方案显示对话框。
/BW	用黑白颜色显示对话框。
/G0	不显示对话框中的图形字符。

注意

过多的文件碎块会使系统速度减慢。DEFRAG 把文件所有分散的簇集中到一起(若空间允许的话)。/F 选项还把所有未用空间移到磁盘的最后，减少了将来出现碎块的可能性。

/Q 选项把所有未用空间移到磁盘的最后，但不消除文件碎块。这是优化空闲空间的最快方法。这是一个未公开的选项，可能不会出现在所有的版本中。不能将/F、/U 或/S 与/Q 一起使用。

DEFRAG 不移动系统文件，因为这些文件很可能必须留在现在的位置上。如果不用/H 选项，隐含文件也不会被移动。不能移动的文件可以引起其他文件出现碎块，尤其是用了/F 选项。

在启动 DEFRAG 之前，应完成下列工作。

- 删除驱动器上所有不再需要的文件。
- 删除所有记录删除操作的文件，如 PCTRACKR. DEL。（运行了 DEFRAG 后这些文件中的记录不再正确）。
- 从 SENTRY（或其他删除保护工具）中清除所有无用文件。
- 退出所有程序，包括 Microsoft Windows。（不要在另一个程序加载的命令提示符下使用 DEFRAG）。
- 运行 CHKDSK /F（DOS 6.0）或 SCANDISK（DOS 6.2 以上）清理丢失的分配单元。
- 停止所有可能在 DEFRAG 运行过程中写盘的程序，比如定时更新一个文件的程序。
- 去掉将消除碎块和移动的文件的系统属性。如果不使用/H 选项，按同样方法处理隐含文件。（要记住许多具有系统属性和隐含属性的文件不能移动）。

如果在命令行中指定了 drive 和/F、/Q 或/U 选项，DEFRAG 就会马上开始消除磁盘碎片。否则 DEFRAG 打开对话框，用户在对话框中可以选择驱动器、碎片消除方式和排序方式。其他功能，如/H，/B 或/SKIPHIGH 只能在命令行中指定，而无法在对话框中选择。

/S 选项对驱动器上每个目录的目录项进行排序，但不移动文件。order 可用的值如下：

N	按名字排序（字母顺序）
N—	按名字排序（字母逆序）
E	按扩展名排序（字母顺序）
E—	按扩展名排序（字母逆序）
D	按日期和时间排序，时间早的文件在前
D—	按日期和时间排序，时间晚的在前
S	按大小排序，小文件在前
S—	按大小排序，大文件在前

如果可能，DEFRAG 装入高端内存，除非使用了/SKIPHIGH 选项。

执行 DEFRAG 后应重新启动计算机以清除旧目录信息的高速缓冲区和普通缓冲区。如果正在使用 FASTOPEN，必须重新启动计算机。

不能使用 DEFRAG 优化网络驱动器或由 INTERLNK 生成的驱动器。

如果指定的驱动器是被压缩的，DEFRAG 会被压缩软件打断。压缩程序会用自己的碎块消除功能代替它。

DEFRAG 的返回值如下：

0	DEFRAG 执行成功。
1	遇到内部错误。
2	磁盘上没有足够的空闲空间。DEFRAG 需要至少一个簇的空闲空间。
3	用户按下 Ctrl+C 或 Ctrl+Break 键中断 DEFRAG。
4	遇到一般性错误。
5	出现读盘错误。
6	出现写盘错误。
7	出现空间分配错误。可以运行 CHKDSK /F（DOS 6.0）或 SCANDISK（DOS 6.2 以上）来更正这一错误。

应用举例

DEFRAG

打开 DEFRAG 对话框。

DEFRAG /B

打开 DEFRAG 对话框。DEFRAG 结束后重新启动计算机。

DEFRAG D:

打开 DEFRAG 对话框使用户可以选择碎块消除方式。

DEFRAG C: /F /B

消除 C 盘上的文件碎块并消除文件间的空闲空间。DEFRAG 结束后重新启动计算机。

DEFRAG C: /U /SD -

消除 C 盘的文件碎块但不消除文件间的空闲空间。按日期和时间顺序整理目录项,时间晚的文件在前。

---

① DEL

见第 16 章

删除一个或一组文件。ERASE 可与 DEL 互换使用。

句法

DEL filespec [/P]

参数和选项

filespec	将被删除的文件。
/P	每次删除前要求用户确认。

注意

DEL 不能删除带有系统、隐含或只读属性的文件。

将 *.* 或不带文件名的目录作为 filespec 可以删除目录中的所有文件(系统、隐含和只读文件除外)。不加/P 选项,DEL 会询问是否确实要删除所有文件。

恢复一个已删除文件的相关内容参见 UNDELETE。

应用举例

DEL OLDFILE.EXE

从当前目录中删除文件 OLDFILE.EXE,除非它具有系统、隐含或只读属性。

DEL C:\OFFICE\OLDFILE.* /P

删除目录 C:\OFFICE 中所有文件名为 OLDFILE 而扩展名任意的文件,但具有系统、隐含或只读属性的文件除外。DEL 显示每个文件名并询问是否删除。

DEL *.*

询问是否删除当前目录中的所有文件。若回答 Y,DEL 删除当前目录中除系统、隐含、只读文件外的所有文件。

DEL C:\OFFICE

询问是否删除该目录中所有文件。如果回答 Y,DEL 删除目录 C:\OFFICE 中除具有系统、隐含、只读属性的所有文件。

DEL C:\OFFICE *.* /P

显示 C:\OFFICE 中的每个文件名(具有系统和隐含属性的文件除外)并询问是否删除。如果对只读文件回答 Y,DEL 显示 Access denied(访问失败)。

Ⓔ DELOLDOS

见第 4 章

从驱动器上删除 DOS 的旧版本。

句法

DELOLDOS [/B]

参数和选项

无参数　显示彩色警告信息。

/B　显示黑白警告信息。

注意

安装 DOS 6 时,SETUP 把在启动盘上找到的以前任何的 DOS 版本保存在一个名为 OLD_DOS.n 的目录中。如果由于某种原因多次安装 DOS,可能会同时存在几个这样的目录,n=1 是第一个,n=2 是第二个,等等。

当确信 DOS 6 已经正常运行后,可以运行 DELOLDOS 命令立即删除所有这些 OLD_DOS.n 目录。

启动 DELOLDOS 后会出现一个警告屏。如果不想继续,按除 Y 以外的任何键。

警告: 如果用以前版本的 DOS 做过备份,在使用 DELOLDOS 前一定要保存旧版本的 RESTORE 或 MSBACKUP。关于如何保存和使用这些旧版本的详细内容参见第 24 章。

Ⓔ DELTREE

见第 14 章

删除一个目录以及其中的所有文件和子目录;删除目录中的文件。

句法

DELTREE [/Y] path|filespec [...]

参数和选项

/Y　启动 DELTREE 后不出现最开始的提示确认。

path　指定将被删除的目录树的最高一级目录。

filespec　指定要删除的文件和目录。

注意

不能使用 DELTREE 删除当前目录,但它可以删除当前目录中的文件和所有子目录。

DELTREE 删除文件和子目录而不管它们的属性。DELTREE 在删除目录时,将删除以该目录开始的整个目录分枝而不考虑分枝内的文件名或属性。

可以在命令中使用通配符。DELTREE 删除所有名字与之相匹配的文件和子目录(连同其中所有文件和子目录)。

DELTREE 在删除文件或目录分枝之前总要提示确认信息(在提示中它将文件称为子目录)。

DELTREE 执行成功后,返回码为 0

应用举例

DELTREE C:\TEST1

删除 C 盘上的 TEST1 目录,包括它的所有文件和子目录以及子目录中的所有文件。删除目录分枝之前会出现提示(若 TEST1 是个文件,则删除文件)。

DELTREE C:\HOLD*.1

如果 C:\HOLD 中有文件名匹配的文件，将其删除，并在删除每个文件前显示提示信息。如果 C:\HOLD 中有名字匹配的目录，连同其中所有文件和子目录一并删除。删除每个目录分枝前显示提示信息。

DELTREE /Y D:\TEST D:\TEMP D:\PRACTICE

如果 D:\TEST，D:\TEMP 和 D:\PRACTICE 是子目录名，删除从它们开始的三个目录分枝。如果这三个名字是文件，则分别删除它们。

© DEVICE

见第 21 章

在常规内存中装入一个设备驱动程序。

句法

DEVICE=filespec [parameters]

参数

filespec	指明要装入的设备驱动程序。
parameters	设备驱动程序所需的所有命令行参数。

注意

决不能用 DEVICE 命令装入 COUNTRY. SYS 和 KEYBOARD. SYS。用 COUNTRY 命令装入 COUNTRY. SYS，KEYB 命令装入 KEYBOARD. SYS。用 DEVICE 命令装入其中任何一个都会使系统停机，必须跳过 CONFIG. SYS 中的这几条命令才能重新启动。

附录 A“设备驱动程序”，指导用户如何安装 DOS 的设备驱动程序。对于第三方软件商提供的设备驱动程序，参考制作者的提供的说明。

将设备驱动程序装入高端内存而不是常规内存的方法，参见 DEVICEHIGH。

应用举例

DEVICE=C:\DOS\ANSI. SYS

将设备驱动程序 ANSI. SYS 装入常规内存。

DEVICE=C:\DOS\DISPLAY. SYS CON=(EGA,437,3)

将设备驱动程序 DISPLAY. SYS 装入常规内存。要注释 CON=(EGA,437,3)代表 DISPLAY. SYS 运行所需的参数。

© DEVICEHIGH

见第 30 章

如果可能，装入设备驱动程序至高端内存。

句法

DEVICEHIGH [[/L:region[,min] [;region[,min]...]] [/S]]filespec [parameters]

参数与选项

filespec	要装入的设备驱动程序。
/L	装入设备驱动程序到高端内存的指定位置。
region	标记高端内存的一段区域。
min	指定设备驱动程序运行时在区域中所需的最小空间。
/S	装入设备驱动程序时将 UMB 收缩至最小。
parameters	设备驱动程序所需的任何命令行参数。

注意

关于装入设备驱动程序的信息和注意事项参见 DEVICE 命令。

如果使用的处理器是80386或更新产品，安装了EMM386.EXE并已请求支持使用UMB，那么可能会有一些高端内存块(UMB)可用。DEVICEHIGH将设备驱动程序装入可用的UMB中。否则，DEVICEHIGH将尽可能把驱动程序装入常规内存。

由于本命令在所有PATH命令之前执行，所以要在filespec中包含完整的路径，除非驱动程序位于启动盘的根目录下：

MEMMAKER是一个将DEVICE转换成DEVICEHIGH的命令。/L和/S选项由MEMMAKER使用。除非清楚这些选项的作用，否则不要使用它们。

DOS 5的DEVICEHIGH命令的句法与本处不同，但在DOS 6下仍可正常工作。

应用举例

```
DEVICEHIGH=C:\DOS\ANSI.SYS
```

如果可能，将驱动程序ANSI.SYS装入高端内存。

---

① DIR

见第14章

列表显示某个目录中的文件和子目录。

句法

```
DIR [filespec] [/P] [/W] [/A[attrib]] [/O[order]] [/S] [/B] [/L] [/C[H]]
```

参数和选项

无参数	以缺省格式列出当前磁盘的完整目录。
filespec	指定要列出的目录和文件。
/P	每次显示一页。
/W	用宽行格式只显示名字，每行最多可有5个名字。
/A[attrib]	显示符合属性判断的目录项。
/O[order]	以order指定的顺序显示目录项。
/S	同样也列出子目录的内容。
/B	不显示开头信息和最后的汇总信息，只列出文件和目录全称。
/L	用小写字母列目录。
/C	显示压缩卷上的文件压缩率。
/CH	更精确地显示存储在DBLSPACE或DRVSPACE卷上文件的压缩率。

注意

环境变量DIRCMD为DIR命令建立缺省选项(见SET命令)。以下注释说明未设置DIRCMD时DOS的缺省选项。

若使用/W选项，列表中不显示文件大小、日期和时间。在这种格式中，DIR用方括号把子目录名括起来，如[DOS]。

如不使用/A选项，DIR不列出系统或隐含的文件和目录。

使用/A选项后，DIR只列出具有指定属性的文件和子目录。不加属性的/A选项列出所有文件而无论其属性如何。attrib的允许值是：

H	列出隐含文件或子目录。
−H	列出非隐含文件或子目录。
S	列出系统文件或目录。
−S	列出非系统文件或目录。
D	列出子目录而不列文件。

-D 列出文件而不列子目录。

A 列出有归档属性的文件或子目录。

-A 列出没有归档属性的文件或子目录。

R 列出只读文件或子目录。

-R 列出非只读文件或子目录。

可以使用多个属性。不要在 attrib 中的多个属性间加空格。/AHS-R 列出所有隐含、系统但非只读的文件和子目录。

缺省情况下,DIR 按目录项在目录中的顺序列出目录项。不带 order 的/O 首先按名字与扩展名顺序列出目录,再按名字和扩展名顺序列出文件。order 的值如下:

A 按名字的字母顺序列目录。

-A 按名字的字母逆序列目录。

E 按扩展名的字母顺序列目录。

-E 按扩展名的字母逆序列目录。

D 按日期和时间列目录,时间早的在前。

-D 按日期和时间列目录,时间晚的在前。

S 按文件大小列目录,小文件在前。

-S 按文件大小列目录,大文件在前。

G 所有子目录列在所有文件之前。

-G 所有文件列在所有子目录之前。

C 按压缩率列目录,低的在前。

-C 按压缩率列目录,高的在前。

可以指定多个排序标准。多个 order 代码之前不要加空格。DIR 首先按第一个值排序,在这之中按第二值排序,等等。例如,用选项/OES 使 DIR 按扩展名排序时,每个扩展名中又按照文件大小(小文件在前)排序。

/S 选项为目录分枝中的每个子目录列表(隐含和系统目录除外)。组合/S 和/B 选项可以精简列表。

组合/S 和/A 选项可以在父目录中列出隐含和系统子目录的目录项,但不列出该子目录的内容。列出隐含或系统子目录中内容的唯一途径是在 filespec 中指定子目录名。

可以通过在选项字母前加减号废除 DIRCMD 的设置对单个 DIR 命令的作用。例如,如果 DIRCMD 使/W 成为缺省选项,可以在命令行中使用/W 选项为单个 DIR 命令关掉/-W 选项。

应用举例

以下例子假定没有为 DIRCMD 赋值。

```
DIR /P
```

列出当前目录内容(除系统、隐含文件和子目录外),每显示一整屏幕后暂停直到用户按下任意键。

```
DIR E:\WIN\*.BAT /S
```

列出 E:\WIN 及其所有子目录中扩展名为 BAT 的所有文件和子目录(除系统、隐含的文件和子目录外)。

```
DIR MYFIEL.* /OE /B > PRN
```

不带开头信息及汇总信息,打印当前目录中所有名为 MYFILE 且扩展名任意的文件和子目录(具有系统、隐含属性的文件和目录除外)。DIR 按扩展名的字母顺序排列各项。

```
DIR /A
```

列出当前目录中所有文件和子目录,包括带有系统或隐含属性的文件和子目录。

DIR /A－RA /W

以宽行格式列出当前目录中所有带归档属性但不带只读属性的文件和子目录，系统或隐含属性的文件和子目录包括在内。

DIR /O－GND

列出当前目录中除具有系统和隐含属性之外的所有文件和子目录。DIR 首先显示文件，其次是子目录。每组中 DIR 按名字的字母顺序排序，同样的名字按日期和时间排序。

DIR MYFILE.BAT /B /S

列出当前目录分枝中所有名为 MYFILE.BAT 的文件，但具有系统或隐含属性及位于系统或隐含子目录中的文件除外。目录或任何子目录的列表中没有开头信息和汇总信息，但会显示每个文件和子目录的完整路径。这是在驱动器上查找丢失文件的好方法。

Ⓔ DISKCOMP

见第 18 章

比较两个软盘。

句法

DISKCOMP [drive1] [drive2] [/1] [/8]

参数与选项

drive1	装有待比较磁盘的第一个驱动器。
drive2	装有待比较磁盘的第二个驱动器。
/1	即使是双面磁盘，也只比较第一面。
/8	只比较每磁道的前 8 个扇区，即使盘上每道不只 8 个扇区。

注意

DISKCOMP 只用于软盘。

如果省略 drive2，DISKCOMP 把当前驱动器作为 drive2。若同时省略 drive1 和 drive2，DISKCOMP 则将当前驱动器即作为 drive1 又作为 drive2。drive1 与 drive2 相同时，DISKCOMP 在需要时会提示用户更换磁盘。

两张磁盘格式必须相同。但/1 和/8 选项可以忽略第二面或每磁道多出的扇区，从而使某些磁盘看起来格式相同。

DISKCOMP 逐个磁道而不是逐个文件比较磁盘。若是用 COPY 或 XCOPY 命令把一个磁盘上的所有文件拷贝到另一个盘，两个盘上的数据位置可能不同，这样 DISKCOMP 不会认为两张磁盘相同。

DISKCOMP 忽略卷系列号的差别。

如果刚刚用 DISKCOPY 拷贝了一张盘但 DISKCOMP 却发现拷贝盘与原来盘不同，问题可能是存在坏扇区、驱动器故障或内存芯片损坏。

DISKCOMP 设置返回值如下：

0	磁盘相同。
1	发现不同之处。
2	用户按下 Ctrl+C 或 Ctrl+Break 中止了 DISKCOMP。
3	出现严重错误。
4	遇到初始化错误。

批处理程序中，可以在 DISKCOMP 后加入一个 IF 命令检测 ERRORLEVEL 以决定下一步的工作。

应用举例

DISKCOMP A: B:

将驱动器 A 中的磁盘与驱动器 B 中的磁盘相比较。

DISKCOMP A: A:

将驱动器 A 中的磁盘与另一个将被换入驱动器的磁盘进行比较。DISKCOMP 在需要换盘时会通知用户。

DISKCOMP

将当前驱动器中的磁盘与另一个将被换入驱动器的磁盘进行比较。

Ⓔ DISKCOPY

见第 18 章

将一张盘拷贝到另一张。

句法

DISKCOPY [drive1] [drive2] [/1] [/M] [/V]

参数与选项

drive1	装有源盘的驱动器。
drive2	装有目的盘的驱动器。
/1	只拷贝源盘的第一面。
/M	拷贝时只使用常规内存。
/V	拷贝后验证数据的正确。

注意

DISKCOPY 只用于软盘。如果目的盘未格式化,DISKCOPY 将其格式化以匹配源盘。

若省略 drive2,DISKCOPY 用当前盘作 drive2。若同时省略 drive1 和 drive2,DISKCOPY 用当前盘作为 drive1 和 drive2。drive1 和 drive2 相同时,DISKCOPY 会在需要时提示用户换盘。

DISKCOPY 为目的盘产生新系列号并在拷贝结束时显示该系列号。

DISKCOPY 完全按原样拷贝源盘。源盘上有碎块的文件在目的盘上也有碎块。DISKCOPY 运行结束后,目的盘的 FAT 与源盘相同。源盘上的坏扇区标记也被复制到目的盘(使目的盘略有损失),但是目的盘上原有的坏扇区标记则被覆盖(通常会引起 DISKCOPY 失败)。

当在 DOS 6.2(或以上版本)下不加/M 选项做单驱动器拷贝时,如果由环境变量 TEMP 所指的驱动器上有足够的空间,DISKCOPY 将整个源盘拷贝到临时文件中。然后要求用户更换磁盘并从临时文件中产生目的盘。它提供选择使用户可以在删除临时文件前以相同的数据拷贝多个目的盘。这一过程较之 DISKCOPY 的以前版本快了很多。

DISKCOPY 的返回码如下:

0	拷贝成功。
1	出现非致命读写错。
2	用户按下 Ctrl+C 或 Ctrl+Break 中断了 DISKCOPY。
3	出现严重错误。
4	出现初始化错误。

批处理程序中,可以在 DISKCOPY 后用 IF 命令检测 ERRORLEVEL 的值以决定下一步工作。

应用举例

DISKCOPY A: B:

将驱动器 A 中软盘的内容拷贝到驱动器 B 中的盘上。

DISKCOPY A: A:

将驱动器 A 中的盘拷贝到另一个将被换入驱动器 A 的盘。

DISKCOPY /V

将当前驱动器中的磁盘拷贝到另一张将被换入当前驱动器的磁盘，拷贝后检查数据的正确性。

© DOS

见第 21 章

将 DOS 装入高内存块(HMA)并指定 DOS 管理高端内存块(UMB)。

句法

DOS=[HIGH|LOW] [,] [UMB|NOUMB]

参数

HIGH|LOW	是否将 DOS 装入 HMA。缺省值是 LOW(不装入)。
UMB|NOUMB	是否由 DOS 管理高端内存块(UMB)。缺省是 NOUMB(不管理)。

注意

可以只用一个参数或两个都用。若用两个参数，任何一个都可以在前，但必须用逗号分开。

必须在安装了扩展内存(Extended Memory)管理器，如 HIMEM. SYS 后才能使用 HIGH 或 UMB。使用 UMB 前还必须安装能提供高端内存块的程序。如果处理器是 80386 或更高级的处理器，可以用 EMM386. EXE 提供对高端内存块的访问。

使用了 HIGH 参数时，DOS 试图将自己的一部分装入 HMA 从而节约常规内存。如果 DOS 无法装入 HMA，它就使用缺省参数 LOW。

DOS 装入 HMA 后，有些程序在装入常规内存时可能会出现问题。详细内容参见 LOADFIX 命令。

应用举例

DOS=HIGH

将 DOS 装入 HMA。

DOS=UMB

DOS 在加载程序和设备驱动程序时可以使用高端内存块。

DOS=HIGH,UMB

DOS 将本身装入 HMA 并可以将程序或设备驱动程序装入高端内存块。

Ⓔ DOSKEY

见第 13 章

保存并可再次使用在 DOS 命令提示符下输入的命令和宏。

句法

DOSKEY [/BUFSIZE = size] [/INSERT | /OVERSTRIKE] [/HISTORY] [/MACROS] [mname=text] [/REINSTALL]

参数和选项

无参数	装入 DOSKEY 内存驻留程序，使用 BUFSIZE 和/INSERT|/OVERSTRIKE 的缺省值。立即可以在 DOSKEY 的缓冲区中记录命令。
/BUFSIZE=size	指定 DOSKEY 缓冲区大小。只有在装入或重新装入 DOSKEY 时本参数才起作用。
/INSERT	缺省输入状态为插入状态。
/OVERSTRIKE	缺省输入状态为覆盖状态(不指定/INSERT 或/OVERSTRIKE 时此为缺省参数)。
/HISTORY	列出 DOSKEY 缓冲区中的所有命令。缩写为/H。
/MACROS	列出当前所有宏定义。缩写为/M。

mname=text 定义一个名为 mname 的宏。text 是当运行 mname 时执行的一个或一组命令。

/REINSTALL 装入 DOSKEY 的新副本。

注意

DOSKEY 是一个内存驻留程序，它保存随后在 DOS 提示符下输入的每个命令。缓冲区装满后，每个新命令会覆盖缓冲区中最旧的命令。DOSKEY 缓冲区最小是 256 字节，缺省是 512 字节。

DOSKEY 允许用户在提示符下用下列编辑键修改命令：

左箭头	光标左移一个字符。
右箭头	光标右移一个字符。
Ctrl+左箭头	光标左移一个单词。
Ctrl+右箭头	光标右移一个单词。
Home 键	光标移到命令行开头。
End 键	光标移到命令行末尾。
Esc 键	消除命令行。
Insert 键	切换插入/覆盖方式。
Backspace 键	删除光标左边的一个字符。
Delete 键	删除光标所在处的字符。
Ctrl+Home 键	删除光标到命令行开头所有字符。
Ctrl+End 键	删除光标到行末所有的字符。

若打开插入方式，输入字符则插入在光标处。当打开覆盖方式时，输入字符则覆盖光标处的字符。用 Insert 键可在两种方式间切换，但每次按回车键后都恢复到缺省方式。

如果装入或重新装入 DOSKEY 时未指定输入方式，/OVERSTRIKE 则是缺省选项。

带有/INSERT 或/OVERSTRIKE 选项的 DOSKEY 命令可以改变缺省值(不重新装入 DOSKEY)。

使用下列键可让 DOSKEY 从缓冲区中召回一个命令：

上箭头	召回缓冲区中的前一个命令。
下箭头	召回缓冲区中的下一个命令。
上卷页(PgUp)键	召回缓冲区中最旧的命令。
下卷页(PgDn)键	召回缓冲区中最新的命令。
F7 键	编号显示存储的命令。
Alt+F7 键	删除缓冲区中的所有命令。
F8	召回缓冲区中下一个以一个或几个字母开头的命令。
F9 键	提示输入命令号，然后召回该条命令。

DOSKEY 在 DOS 提示符后显示召回的命令。可对其进行编辑和输入，如同用户刚刚敲入的一样。

要想不搜寻整个列表就召回某条命令，敲入命令的前几个字母然后按 F8 键。不断地按 F8 键直到 DOSKEY 找到所需的命令。

若要以编号召回命令，按 F7 键查阅编了号的列表，找到所需命令，然后按 F9 键就可以输入命令对应的编号。

DOSKEY 允许一个命令行中包含两个或多个命令。分隔命令使用 Ctrl+T 键，它显示为 。这一功能可以产生效率更高的命令列表。例如，命令 DIR A: DIR B:[PP]DIR C:用于显示驱动器 A、B 和 C 上的缺省目录，它可以作为一条命令被召回并输入。

宏由一条或多条命令组成，输入宏名即可执行宏。宏定义中可含有下列特殊字符。

$G	重定向输出，与输出重定向符号(>)作用相同。如果使用这个重定向符号会导致重定向 DOSKEY 命令的输出，而不是被作为宏定义的一部分来对待。

GG	将输出添加到一个文件的末尾。与添加重定向符号(>>)作用相同。
$L	输入重定向。与输入重定向符号(<)作用相同。
$B	输出送到下一个命令。与管道符号(|)作用相同。
$T	分隔命令。与 DOSKEY 的命令分隔符(Ctrl+T)作用相同。
$$	美元符。
$n	代表运行宏时所给的一个参数。n 可以 1 到 9。这与批处理文件中的%1 到%9 类似。
$*	代表所有参数。$* 会被命令行中宏名后的任何内容代替。

宏可与 DOS 命令同名。要运行宏,在 DOS 提示符下敲入宏名。如果要运行的是 DOS 命令,在提示符和命令之间插入一个或几个空格。因为某种原因需要更改 DOS 命令,可以使用这一功能。

DOSKEY 的宏和命令记录共用一个缓冲区。新定义的宏可能会冲掉缓冲区的命令。但新宏或命令都不会覆盖已有的宏。DOSKEY 限制了宏的数目,为命令记录保留大约一半的缓冲区。

虽然在批处理文件中可以用 DOSKEY 命令生成宏,但却不能执行宏。

一个宏的长度不能超过 120 个字符。

/REINSTALL 选项装入 DOSKEY 的新副本并带有一个空的 DOSKEY 缓冲区。以前的 DOSKEY 副本仍留在内存中,但已失去作用,只是浪费内存空间。

应用举例

```
DOSKEY
```

如果这是计算机启动后第一个 DOSKEY 命令,则装入 DOSKEY 内存驻留程序,带有 512 字节的缓冲区并以覆盖作为缺省输入方式。否则这一命令不起作用。

```
DOSKEY  /BUFSIZE=1024  /INSERT
```

如果是计算机启动后的第一个 DOSKEY 命令,则装入 DOSKEY 内存驻留程序,带有 1204 字节的缓冲区,缺省输入方式是插入状态。否则这一命令只把缺省输入方式改为插入状态,而/BUFSIZE 选项不起作用。

```
DOSKEY /H /M
```

显示所有命令记录和宏定义。

```
DOSKEY /H > TEMP.BAT
```

将命令记录拷贝到文件 TEMP.BAT 中。

```
DOSKEY ACOPY=COPY $1 $2 $T DIR $2 $G PRN
```

定义一个宏 ACOPY,其作用是拷贝一个文件然后打印新文件的目录列表。

```
ACOPY JUNEDAT.NEW JUNEDAT.OLD
```

执行宏 ACOPY,拷贝 JUNEDAT.NEW 到 JUNEDAT.OLD,打印 JUNEDAT.OLD 的文件列表。

Ⓔ DOSSHELL

见第 5 章

启动 DOS Shell——DOS 图形界面。

句法

```
DOSSHELL [/T[:resolution[n]] [|/G[:resolution[n]]] [/B]
```

参数与选项

/T	以文本方式启动 DOS Shell。
/G	以图形方式启动 DOS Shell。
/B	以黑白方式启动 DOS Shell。

resolution 指定屏幕分辨率类别。

n 当一类中仍有多个选择时具体指定屏幕分辨率。

注意

DOS 6.2 或更新版本中不再包含 DOS Shell。若未从 DOS 旧版本中获得一份 DOS Shell，可向 Microsoft 索取(详情参见《用户指南》)。

DOS Shell 至少要求 384 KB 常规内存。

如果要同时运行 DOS Shell 和 Microsoft Windows，从 Windows 内启动 DOS Shell，而不要反过来。

若文件 DOSSHELL.INI 与 DOSSHELL.EXE 不在同一目录中，在 DOS Shell 启动前应设置环境变量 DOS Shell，使之指向 DOSSHELL.INI 的目录(参见 SET 命令)。

Resolution 可以是 L、M 或 H，分别代表低、中、高分辨率。缺省值取决于硬件。n 可能的值和缺省值也同样取决于硬件。

如果启动 DOS Shell 时未用/T 或/G 选项，因而无法读出屏幕内容，那么按 F3 键回到命令提示符下，然后试试文本方式的/T 选项。DOS Shell 运行后，可以使用 Options、Display 命令调整显示方式(T 或 G)和屏幕分辨率。若用 Options、Display 设定新值，DOS Shell 会把它们保存在 DOSSHELL.INI 中并使它们成为缺省值。命令行参数和选项会覆盖当前的缺省设置但并不会将它们存为新的缺省设置。

应用举例

```
DOSSHELL /T:L
```

以低分辨率文本方式(每屏 25 行)启动 DOS Shell。

```
DOSSHELL /G /B
```

以缺省分辨率的黑白图形方式启动 DOS Shell。

```
DOSSHELL
```

以 DOSSHELL.INI 中的缺省设置启动 DOS Shell。

© DRIVPARM

见第 51 章

重新定义物理驱动器的参数。

句法

```
DRIVPARM=/D:n [/C] [/F:type] [/H:heads] [/I] [/N] [/S:sectors] [/T:tracks]
```

参数和选项

/D:n	需要重定义的驱动器。
/C	说明驱动器有换线(change-line)支持。只用于可移动驱动器。
/F:type	指定驱动器类型。
/H:heads	指定磁头数。
/I	表示电子兼容的(electronically compatible)3 ½英寸驱动器。
/N	表示固定驱动器。
/S:sectors	指定每个磁盘的扇区数。
/T:tracks	指定磁道总数。

注意

正常情况下，DOS 在启动时会查出软驱、硬盘和磁带驱动器等块设备，但偶尔也会出错。如果 DOS 在访问某一块设备，尤其是进行格式化时出现问题，在 CONFIG.SYS 中加入 DRIVPARM 使 DOS 每次都正确访问驱动器。

在选项/D 中，n 为 0 表示驱动器 A，1 表示驱动器 B，2 表示驱动器 C，等等。

换线支持由/C 表示，说明该磁盘驱动器可以检测它的门是否开着。若忽略/C 选项。DOS 将驱动器作为没有换线支持来对待，每次访问驱动器时都要检查磁盘的系列号。

警告： 如果磁盘驱动器没有换线功能，千万不要指定它有。这样做会使 DOS 覆盖重要数据，因为它不知道磁盘已换过。

在/F 选项中，type 可有以下值：

0	360KB 或更少
1	1.2MB
2	720KB
3	8 英寸单密度
4	8 英寸倍密度
5	硬盘
6	磁带驱动器
7	1.44MB
8	可读写光盘
9	2.88MB

如果不用/F 选项，磁盘缺省类型是 2(720KB)。

下表显示当/F 选项指示一个软盘(类型 0、1，7 或 9)时磁头数、扇区数和磁道数的缺省值。若要指定其他值，可根据需要用/H，/T 和/S 选项。

类型	容量	磁头数	扇区数	磁道数
0	160KB	1	8	40
0	180KB	1	9	40
0	320KB	2	8	40
0	360KB	2	9	40
2	720KB	2	9	80
1	1.2MB	2	15	80
7	1.44MB	2	18	80
9	2.88MB	2	36	80

如果驱动器是 3 ½英寸的软盘驱动器并使用已有的磁盘控制器，但机器的 ROM BIOS 却不支持 3 ½英寸的软盘，可以用/I 选项。

错误或不一致的值会导致对驱动器的不稳定或错误的格式化。如果必须重新定义驱动器，应从驱动器的说明书或生产厂家得到正确的参数。

如果错误地重新定义了启动盘，系统可能无法启动或在启动后破坏根目录。重新定义启动盘之前一定要做好备份。

应用举例

```
DRIVPARM /D:1 /C
```

定义 B 驱动器为带换线支持的 720KB (缺省值)驱动器。

```
DRIVPARM /D:4 /F:6 /H:1 /S:40 /T:28
```

定义第 5 个驱动器为磁带驱动器，有 1 个磁头、28 个磁道、每道 40 个扇区。

Ⓔ DRVSPACE

见第 34 章和第 45 章

压缩硬盘或软盘;管理压缩盘。

注释: 只有在 DOS 6.22 中才能使用 DRVSPACE。DOS 6.0 的压缩程序参见 DBLSPACE。除 DRVSPACE 命令不提供/CHKDSK 功能外,DRVSPACE 与 DBLSPACE 命令相同。

句法 1

DRVSPACE

启动 DRVSPACE 设置程序,安装 DriveSpace 并压缩和装入驱动器。如果 DriveSpace 已安装,则打开 DriveSpace 窗口。

注意

第一次使用 DRVSPACE 时必须用本句法。DRVSPACE 为用户建立文件压缩设置系统,将 DRVSPACE 的核心程序(DRVSPACE.BIN)和其他文件放在启动盘的根目录下。它还将命令 DEVICEHIGH=DRVSPACE.SYS 放在 CONFIG.SYS 文件中,以便在可能情况下使 DRVSPACE.BIN 移入高端内存,否则移到常规内存的底部。DRVSPACE 还要至少压缩一个硬盘,不压缩硬盘就不能安装 DriveSpace。

再次使用句法 1 时,装入 DriveSpace 程序并显示带下拉菜单的对话框,可以从中选择压缩其他驱动器或管理已压缩驱动器。

安装成功后,可以使用 DRVSPACE 命令的其他格式完成 DriveSpace 功能而无需打开对话框。

当 DriveSpace 压缩已有数据时,会生成一个名为 DRVSPACE.000 的大文件来包含这些数据(该文件具有系统、隐含、只读属性)。原来的驱动器上现在有文件 DRVSPACE.000,其他不适于压缩的文件(如 Windows 的交换文件)和一些空闲空间。压缩后的数据文件称为压缩卷文件(CVF),通过原来的驱动器符访问。原来的驱动器称为宿主驱动器,通过由 DriveSpace 指派的新字母来访问(即指定了新的盘符)。(如果想知道 DriveSpace 将哪一个字母指派给了缩主驱动器,用命令 DRVSPACE/INFO 或/LIST,参见句法 8 和 9)。

为压缩盘和宿主盘指派驱动器符称为装入(mount)压缩盘。系统启动时 DriveSpace 自动装入所有压缩的硬盘驱动器。除非关掉自动装入参数,否则它还会自动装入软盘驱动器(见句法 14)。

DriveSpace 不能压缩空间已满的驱动器。启动盘至少要有 1.2MB 的空闲空间,其他驱动器,包括软盘,至少有 1.1MB 空闲空间。

如果驱动器上原有的一个 SENTRY 目录,压缩后在压缩盘和宿主盘上都会有 SENTRY 目录。但在压缩前删除的文件已无法恢复。压缩后删除的文件可通过 Delete Sentry(删除卫士)按通常方式恢复。但如果要保护未压缩的宿主盘,应将新的驱动器符加入被 Delete Sentry 保护的驱动器中。

若驱动器上原有一个 PCTRACKR.DEL 文件,它仍留在宿主盘上。可是压缩前的文件可能无法恢复,因为它们的簇已被 CVF 重新利用。压缩后删除的文件仍可通过删除跟踪(deletion-tracking)按通常方式恢复。若要保护未压缩的宿主驱动器,应将新驱动器符加入到被删除跟踪保护的驱动器中。

如果在 SMARTDRV 命令中提到一个压缩盘,则会出现一条信息告知用户必须缓冲宿主盘。不需要改动该命令,DriveSpace 会与 SmartDrive 相互合作通过宿主盘对压缩盘进行缓冲。

警告: 不要对压缩盘进行写缓冲。假如系统在完成写入前崩溃了,延迟写入会使压缩盘上的数据受损。如果在升级到 DOS 6.22 之前就已安装了 SmartDrive,那么很可能已对压缩盘设置了写缓冲。更改 SmartDrive 设置撤销压缩盘写缓冲。

本命令的以下句法通过命令行管理 DriveSpace 盘。所有这些功能同样可在 DriveSpace 对话框中完成。该对话框中输入不带参数的 DRVSPACE 命令(本句法)后被打开。

句法 2

DRVSPACE /COMPRESS drive1 [/NEWDRIVE=drive2] [/RESERVE=size] [/F]

压缩一个已存在的硬盘、软盘或其他可移动媒体上的文件,并装入(mount)驱动器。

参数和选项

/COMPRESS	使用压缩功能，其缩写形式为/COM。
drive1	需要压缩的驱动器。
/NEWDRIVE=drive2	指定将由宿主驱动器使用的驱动器符。其缩写形式为/NEW(使用句法 10 可查出哪个驱动器符可用)。不加/NEW 选项时，DriveSpace 为新盘选定驱动器符。
/RESERVE=size	以兆字节为单位指定要在宿主驱动器上保留多少空间不被压缩。该选项的缩写形式为/RES。若不加/RES 参数，DriveSpace 在可能情况下为硬盘留下 2MB 的空间。
/F	略过结束时的状态报告，直接返回命令提示符。

注意

当 DriveSpace 压缩软盘时，会在宿主驱动器上加进一个名为 READTHIS. TXT 的文件。该文件向磁盘的接受者说明其中含有压缩数据及装入(mount)和读写的方法。如果接受者用的是已安装了 DriveSpace 的 DOS 6.22 系统，文件 READTHIS. TXT 就不会出现，因为压缩过的软盘会被自动装入(mount)。若未安装 DriveSpace 则该文件会出现。

无论是否此时指定/RES 选项，都可以在以后增加或减小未压缩的空间(见句法 13)。

关于压缩磁盘时的详细情况参见句法 1。

应用举例

```
DRVSPACE /COM A
```

压缩 A 驱动器中的软盘。压缩后为宿主驱动器指派一个可用的驱动器符。

```
DRVSPACE /COM D /RES=2.5 /NEW=L
```

压缩 D 驱动器，保留 2.5MB 的空闲空间(如果可能)。将 L 指派给宿主驱动器。

句法 3

```
DRVSPACE /UNCOMPRESS drive
```

将驱动器解压缩。

参数和选项

/UNCOMPRESS	使用 UNCOMPRESS 功能。
drive	指定将被解压缩的压缩盘。

注意

UNCOMPRESS 功能将 CVF 中的数据解压缩后存放在宿主驱动器上。如果宿主驱动器没有足够的空间装下解压缩后的数据，可能需要事先由用户删除一些压缩文件或移到其他盘上。解压缩后，压缩盘被卸出(unmount)并删除，宿主盘恢复原名。

最后一个压缩盘被解压缩后，DRVSPACE. BIN 退出内存但并不从根目录下删除。以后还可以运行 DRVSPACE 来压缩驱动器。

句法 4

```
DRVSPACE /CREATE drive1 [/NEWDRIVE=drive2] [/SIZE=size1 | /RESERVE=size2]
```

从未压缩驱动器的空闲空间中生成一个新的空压缩盘。

参数和选项

/CREATE	使用 CREATE 功能，其缩写形式为/CR。
drive1	指定宿主驱动器。

/NEWDRIVE=drive2 为新 CVF 指定驱动器符，其缩写形式为/NEW。若无/NEW 参数，DriveSpace 为

新 CVF 选择驱动器符。

/SIZE=size1 以兆字节为单位指定压缩卷文件的总空间。该选项的缩写形式为/SI。

/RESERVE=size2 指定在未压缩盘上留下多少兆字节的空闲空间。其缩写形式为/RES。

注意

不能在软盘上使用/CREATE。

若从空闲空间而不是已有数据中生成一个 CVF,DriveSpace 为新卷指定新的驱动器符,而宿主驱动器保留原名。可以从已存在压缩卷文件的驱动上生成另一个新的空卷。原来的 CVF 名为 DRVSPACE.000 并拥有原驱动器符。新的 CVF 名为 DRVSPACE.001(假定它是驱动器的第二个 CVF),使用新的驱动器符。余下的未压缩空间位于宿主盘上,宿主盘保留 DRVSPACE.000 生成时赋予的驱动器符。

若未指定/SIZE 和/RES 选项,DriveSpace 会利用除 1MB 空闲空间外的全部空间。

应用举例

DRVSPACE /CR E

在宿主盘 E 上生成一个新的空压缩卷文件,留下 1MB 的空闲空间。为新压缩卷文件指派一个可用驱动器符。

DRVSPACE /CR E /NEW=L /RES=1.5

在宿主盘 E 上生成一个新的空压缩卷文件,为 E 盘保留(如果可能)1.5MB 的空闲空间,新 CVF 的驱动器符为 L。

句法 5

DRVSPACE /DEFRAGMENT [drive] [/F]

消除压缩盘的存储碎块。

参数和选项

/DEFARGMENT 使用 DEFRAGMENT 功能,其缩写形式为/DEF。

drive 需要整理的压缩盘。缺省时是当前驱动器。

/F 更全面地进行碎块清除。

注意

为提高压缩盘性能而消除碎块不是必需的,但它可以集中盘上的空闲空间。如果要减小压缩盘的容量(见句法 14),应首先执行 DRVSPACE /DEF。

如果在 DriveSpace 盘上运行 DOS 的 DEFRAG 命令,DOS 在 DEFRAG 结束后会自动运行 DRVSPACE /DEF。

应用举例

DRVSPACE /DEF /F

消除当前(压缩)盘的碎块。

句法 6

DRVSPACE /DELETE drive

删除压缩盘以及其中的所有文件。

参数和选项

/DELETE 使用 DELETE 功能,其缩写形式为/DEL。

drive 指定要删除的压缩盘。

注意

删除 DriveSpace 的同时也将其卸出(unmount)。若删除 DRVSPACE.000,宿主盘将恢复原来的驱动器符。如果删除任何其他 CVF,其对应卷的驱动器符就可重新使用。

如果删除 DriveSpace 盘后改变主意，可以利用 DOS 的 UNDELETE 命令进行恢复。在宿主盘上查找一个名为 DRVSPACE. nnn 的删除文件。

若已使用 UNDELETE 成功地恢复了文件，在将其装入前仍无法访问数据(见句法 11)。

应用举例

DRVSPACE /DEL F

删除压缩盘 F。

句法 7

DRVSPACE /FORMAT drive

格式化压缩盘。

参数和选项

/FORMAT	使用 FORMAT 功能。其缩写形式为/FOR。
drive	要格式化的压缩盘。

注意

格式化压缩盘会删除其中的所有文件。无法恢复已格式化的压缩盘上的文件(unformat)。

应用举例

DRVSPACE /FOR G

格式化压缩盘 G，删除所有数据。

句法 8

DRVSPACE [/INFO] drive

显示关于压缩盘的信息。

参数和选项

/INFO	使用 INFO 功能。未指定其他 DRVSPACE 功能时此为缺省参数。
drive	显示该驱动器的信息。

注意

显示的信息中包括宿主驱动器，CVF 的文件名，总空间，已用空间，CVF 和宿主驱动器的空闲空间，实际平均压缩率和压缩估算比率。CVF 的空闲空间是基于压缩估算比率对能够压缩进 CVF 的剩余空间中的字节数的估计。

应用举例

DRVSPACE D

显示压缩盘 D 的信息。

DRVSPACE /INFO

显示当前驱动器的有关信息。

句法 9

DRVSPACE /LIST

列出并描述计算机上所有非网络驱动器，包括 DriveSpace 卷。

参数和选项

/LIST	使用 LIST 功能，其缩写形式为/L。

注意

本句法列出每个可用驱动器符并描述其状态，如 Local hard drive(本地驱动器)，Compressed hard drive(压缩硬盘)，Floppy drive(软盘)，Removable-media with No disk in drive(未插入盘的可移动媒体驱动器)，

Available for DriveSpace(可用于 DriveSpace)等等。RAM 驱动器被描述为本地驱动器。对于每个驱动器在适当情况下显示已用空间和未用空间。对于已装入的压缩盘,报告中显示压缩卷名和宿主驱动器。

句法 10

DRVSPACE /MOUNT[=nnn] drive1 [/NEWDRIVE=drive2]

装入(mount)一个压缩卷。

参数和选项

/MOUNT	使用 MOUNT 功能,其缩写形式为/MO。
nnn	指定要装入的卷的扩展名。缺省是 000。
drive	指定含有要装入的压缩文件的驱动器。
/NEWDRIVE=drive2	指派一个驱动器符,其缩写形式为/NEW。若不加/NEW 选项,DriveSpace 则指派一个可用驱动器符。

注意

当装入文件 DRVSPACE.000 时,宿主驱动器的驱动器符被指派给宿主驱动器。装入任何其他 DRVSPACE.nnn 文件时,新字母被指派给压缩盘。

DRVSPACE 在压缩或生成驱动器时自动将其装入。它在驱动器上执行功能时会对该盘进行卸出和装入操作。系统启动时它自动装入压缩硬盘。唯一需要用户进行装入操作的情况是执行了卸出操作之后(见句法 13)。除非改变了 DRVSPACE.INI 中的 AutoMount 参数,否则 DriveSpace 还自动装入软盘(见句法 14)。如果 AutoMount=0,用户要在每次插入软盘后进入装入操作。

应用举例

DRVSPACE /MO A

装入驱动器 A 中的磁盘,指派驱动器符 A 给 DRVSPACE.000,另一个可用驱动器符给磁盘的未压缩部分。

DRVSPACE /MO=001 E /NEW=L

装入一个压缩盘,驱动器符 L 指派给 E:\DRVSPACE.001,保留 E 为宿主驱动器符。

句法 11

DRVSPACE /RATIO[=r.r] [drive|/ALL]

改变一个或所有压缩盘的压缩估算比率。

参数和选项

/RATIO=r.r	指定新的压缩估算比率,从 1.0 到 16.0。缺省值是驱动器当前的综合压缩率。该选项的缩写形式为/RA。
drive	指定驱动器。
/ALL	改变当前已装入的所有压缩盘的压缩估算比率。

注意

DriveSpace 利用压缩估算比率估计在压缩时,可以有多少字节的数据被压缩进 CVF。这会影响一些功能如 DIR 和 DOS Shell 报告的结果。

若未指定驱动器或/ALL,DriveSpace 则改变当前驱动器的比率。

利用句法 8 可以查出驱动器的压缩估算比率和实际压缩率。

应用举例

DRVSPACE /RA /ALL

将当前已装入的所有压缩盘的压缩估算比率改成各自的平均压缩率。

DRVSPACE /RA=3 D

将压缩盘 D 的压缩估算比率改成 3.0 比 1。

句法 12

DRVSPACE /SIZE[=size1|/RESERVE=size2] drive

改变压缩盘的大小。

参数和选项

/SIZE	使用 SIZE 功能，其缩写形式为/SI。
size	以兆字节为单位指定 CVF 文件的新长度。不能与/RESERVE 选项一同使用。
/RESERVE=size2	以兆字节为单位指定 DriveSpace 应为宿主盘留下的空闲空间总量。该选项的形式为/RES。它不与 Size1 同时使用。若 Size2 为 0，压缩盘尽可能增加到最大。
drive	指定要改变大小的压缩盘。

注意

如果没有指定 size1 和/RES，DriveSpace 则使压缩尽可能小。

应用举例

DRVSPACE /SI=30 D

压缩盘 D 的大小改为 30MB(如果可能)。

DRVSPACE /SI /RES=10 D

改变压缩盘 D 的大小，为宿主盘留下(如果可能)10MB 的空闲空间。在此之前应首先消除压缩盘上的存储碎块。

DRVSPACE /SI /RES=0 D

使压缩盘 D 的空间尽可能大。

DRVSPACE /SI D

使压缩盘 D 尽可能小。在此之前应首先消除压缩盘上的存储碎块。

句法 13

DRVSPACE /UNMOUNT [drive]

卸出 (unmount)一个驱动器。

参数和选项

/UNMOUNT	使用 UNMOUNT 功能，其缩写形式为/U。
drive	指定要卸出的驱动器。缺省驱动器为当前驱动器。

注意

已被卸出的驱动器再次装入之前不能使用。若宿主驱动器符以前被交换给了 CVF，那么它现在就返还给宿主驱动器，宿主驱动器当前使用的驱动器可作它用。

从驱动器中取出软盘之前不需要执行卸出操作。

应用举例

DRVSPACE /U D

卸出压缩盘 D。如果 D 被指派给了 DRVSPACE.000，它现在就代表宿主驱动器。否则，宿主驱动器符保持不变而 D 可作它用。

句法 14

DRVSPACE [/AUTOMOUNT=0|1|X][/DOUBLEGUARD=0|1][drive1/HOST=drive2][/LASTDRIVE=x] [MAXFILEFRAGMENTS=n] [/MAXREMOVABLEDRIVES=n] [/ROMSERVER=0|1] [/SWITCHES=F|N]

设置 DRVSPACE.INI 中的缺省值。

参数和选项

/AUTOMOUNT	指定 DriveSpace 是否自动装入软盘。
/DOUBLEGUARD	允许或禁止 DoubleGuard 功能。该功能不断检查 DriveSpace 的内存缓冲区是否被破坏。
/HOST	指定压缩盘(drive1)的宿主盘(drive2)。
/LASTDRIVE	指定 DriveSpace 在装入驱动器时可以使用的最高驱动器名。
/MAXFILEFRAGMENTS	指定一个 CVF 在其宿主驱动器上可被分成的最大碎块数。
/MAXREMOVABLEDRIVES	指定 DriveSpace 在启动时应另外再为多少压缩盘保留空间。
/ROMSERVER	允许或禁止检查 ROM BIOS 中的 Microsoft 实时压缩接口(MRCI)服务程序。
/SWITCHES	控制在启动期间是否允许用户跳过 DriveSpace。

注意

文件 DRVSPACE. INI 中含有 DriveSpace 的缺省设置。因为它具有隐含、系统和只读属性,所以无法直接编辑(去掉它的隐含、系统和只读属性很危险。)用这里提到的命令改变 DRVSPACE. INI 中的设置。

用 DOS 的 EDIT 可以查看 DRVSPACE. INI,但不能编辑。通过查看可以找出参数的当前设置。

缺省情况下 DriveSpace 自动装入可移动驱动器(如软驱),DriveSpace 必须为每个需要自动装入的驱动器预留空间。命令 DRVSPACE /AUTOMOUNT=0 设置 DRVSPACE. INI,以免可移动驱动器被自动装入,从而节省了内存。DRVSPACE /AUTOMOUNT=1 可恢复缺省设置。也可以使用 DRVSPACE /AUTOMOUNT=A,DRVSPACE /AUTOMOUNT=B 等命令专门指定某个驱动器应自动装入。(关于手工装入未自动装入的压缩盘的命令参见句法 10)。

DoubleGuard 不断检查 DriveSpace 的内存缓冲区是否被破坏,一旦发现问题则停止系统运行以免损坏压缩盘。DoubleGuard 会略微降低系统速度。可以使用命令 DRVSPACE /DOUBLEGUARD=0 禁止 DoubleGuard 功能。而命令 DRVSPACE /DOUBLEGUARD 又可重新恢复这一功能。

警告: 禁止 DoubleGuard 可能会导致压缩数据被破坏。

如果见到信息 Compressed drive x is currently too fragmented to mount,说明作为驱动器 X 的 CVF 被分成的碎块数超出了 DRVSPACE. INI 中的 MAXFRAGMENTS 设置值。问题得到解决前无法装入或使用该压缩盘。可以增加允许的碎块数,输入命令 DRVSPACE /MAXFRAGMENTS=n,其中 n 是新的碎块数。也可以去掉 CVF 的属性,消除宿主盘上的碎块,再恢复属性(这一操作至关重要)。

参数 MAXREMOVABLEDRIVES 决定了在启动后 DriveSpace 能够另外处理的压缩盘的最大数目(无论磁盘是否可移动的)。这不仅控制了最多能从可移动驱动器中装入多少个压缩卷,还控制了在不重新启动计算机的情况下最多能生成多少新的压缩卷。缺省的最大值是 2。减小这个值可以节省内存空间但限制了生成新压缩盘的能力。

MRCI 检测会妨碍不使用 MRCI 的 ROM BIOS,所以,除非确知系统需要,否则不要进行这一检测。命令 DRVSPACE /ROMSERVER=1 允许 MRCI 检测。DRVSPACE /ROMSERVER=0 将其禁止。

在缺省情况下,假如用户在启动计算机期间按了 Ctrl+F5 键,DOS 则显示 CONFIG. SYS 和 AUTOEXEC. BAT 的每一行命令并询问是否执行。它还会询问是否装入 DriveSpace。如果用户在启动期间按了 Ctrl+F8 键,DOS 则不加询问跳过 CONFIG. SYS、AUTOEXEC. BAT 和 DRIVESPACE. BIN。使用命令 DRVSPACE /SWITCHES=N 可以禁止跳过 DriveSpace 部分。当跳过操作被禁止时,Ctrl+F5 键仍有条件执行 CONFIG. SYS 和 AUTOEXEC. BAT,而按 Ctrl+F8 键仍然跳过 CONFIG. SYS 和 AUTOEXEC. BAT。

命令 DRVSPACE /SWITCHES=F 缩短了 DOS 等待用户按跳过键的时间,使启动速度略有提高。

一旦输入了/SWITCHES=N 或/SWITCHES=F,没有命令可以把它们从 DRVSPACE. INI 文件中去

掉。必须编辑该文件去掉 SWITCHES 参数并置为缺省设置。

关于禁止用户跳过 AUTOEXEC.BAT 和 CONFIG.SYS 的命令，参见 SWITCHES。

Ⓘ ECHO

Ⓑ 见第 20 章

打开或关闭命令的回显功能或显示一条信息。

句法

ECHO [message|ON|OFF]

参数和选项

无参数	显示 ECHO 的当前设置。
ON|OFF	打开或关闭命令回显功能。
message	在屏幕上显示 message。
ECHO.	这个特殊的 ECHO 命令显示一个回车符(一个空行)。在 ECHO 和句点间一定不要有空格。

注意

· 如果打开命令回显功能，DOS 在准备好接收和处理命令时会显示命令提示符。运行批处理程序时，DOS 为程序中每一条命令都显示命令提示行(包括提示符和命令)并跟随一个空行。缺省情况下命令回显是打开的。当它是关闭状态时可用命令 EOHO ON 打开。

关掉命令回显功能则不显示命令提示符，但仍可以输入命令。执行批处理程序时，不显示命令行和正常时紧随其后的空行。只有命令的输出被显示。命令 ECHO OFF 关掉命令回显功能。

带有 message 的 ECHO 命令将 message 作为 ECHO 的命令输出来显示，所以即使关掉命令回显功能仍会显示 message。

由批处理程序改变的命令回显状态只维持到批处理程序结束。如果批处理程序开始执行时命令回显是打开的，那么在程序结束时它将被自动打开。如果程序开始时它是关闭的，那么程序结束时它将被自动关闭。

ECHO OFF 命令在 DOS 处理下一条命令时才会发生作用。若要禁止回显 ECHO OFF 命令本身，或批处理程序中任何单独一行，将@放在行首。

应用举例

```
@ ECHO OFF
ECHO.
ECHO This batch program formats disks in drive A.
ECHO You will need new 5.25 inch disks.
ECHO.
ECHO ON
```

在批处理程序中的这一系列命令的作用是将回显功能关掉(不显示 ECHO 命令)，然后显示一个空行、两行信息和另一空行。最后又将回显功能打开。

```
ECHO . >PRN
ECHO Beginning of Weekly Reprot > PRN
ECHO . > PRN
```

打印一个空行、一行信息和另一空行。

Ⓔ EDIT

见第 19 章

启动 DOS Editor，它用于创建或修改 ASCII 文本文件。

句法

EDIT [filespec] [/B] [/G] [/H] [/NOHI]

参数和选项

无参数	以缺省的屏幕式样启动 Editor，不载入文件。
filespce	打开一个 ASCII 文本文件。若文件不存在则生成一个。若已存在，则显示其内容。
/B	采用黑白显示方式。
/G	对 CGA 显示器使用快速屏幕刷新。
/H	显示出显示器的最大行数。
/NOHI	允许使用 8 包显示器。

注意

如果文件 QBASIC.EXE 不在当前目录、程序搜寻路径、也不在与 EDIT.COM 相同的目录中，EDIT 则不能工作。若为了节省空间而删除了 QBASIC.EXE，就不能使用 DOS Editor 了。

如果 DOS Editor 在显示器上不能正确显示，或屏幕最底行不显示快捷键(shortcut keys)，试试用选项/B 或/NOHI(或两个都用)。

应用举例

EDIT ANYFILE.BAT /B

创建或载入 ANYFILE.BAT，用黑白方式显示。

EDIT

以缺省方式启动 Editor，启动时不载入文件。

Ⓔ EMM386

见第 30 章和第 44 章

控制扩充内存(expanded memory)。

句法

EMM386 [ON|OFF|AUTO] [W=ON|W=OFF]

参数和选项

无参数	显示 EMM386 状态。
ON	激活 EMM386.EXE 驱动程序。
OFF	关闭但不把 EMM386.EXE 从内存中卸出。
AUTO	由程序控制激活或关闭 EMM386.EXE 驱动程序。
W=ON	允许支持 Weitek 协处理器。
W=OF	禁止支持 Weitek 协处理器。缺省是 W=OFF。

注意

必须具备 386 或更高档的处理器且已安装了 EMM386.EXE 设备驱动程序才能使用本命令。该驱动程序的说明见附录 A 的 EMM386.EXE。

当 EMM386 正在提供 UMB 或有程序在使用扩充内存(expanded memory)时不能关掉 EMM386。

某些程序，包括以标准模式(Standard mode)运行的 Microsoft Windows 3.0，要求关掉 EMM386。

不能在同一个 EMM386 命令中同时使用 OFF 和 W=ON 选项。

Weitek 协处理器要求有 HMA。如果用了 DOS=HIGH，则可能无法支持 Weitek 协处理器(见命令 DOS)。

应用举例

EMM386 OFF

关闭扩充内存功能但不卸下 EMM386 驱动程序。

EMM386 AUTO

由程序根据需要允许或禁止扩充内存功能。

ERASE

见 DEL。

---

① EXIT

见第 43 章

退出二级命令处理器。

句法

EXIT

注意

在不带/P 选项的 COMMAND 命令启动的二级命令处理器下，EXIT 关掉二级 COMMAND 命令副本回到其父进程。如果是由其他程序，如 Microsoft Windows 或 DOS Shell 启动的二级命令处理器，EXIT 则关掉二级命令处理器返回启动它的程序。在其他情况下，EXIT 不起作用。

有关父、子命令处理器的更多内容参见 COMMAND 命令。

Ⓔ EXPAND

见第 4 章

展开 DOS 安装盘上的文件。

句法

EXPAND [filespec...[destination]]

参数

无参数	提示输入 filespec 和 destination。
filespec	DOS 安装盘上需要展开的压缩文件。
destination	指定展开后文件的位置和名字。

注意

DOS 6.22 安装盘上提供的文件大部分是压缩的。在磁盘上可以认出这些文件，因为文件名字的最后一个字母是下划线(_)。DOS 6.22 的 SETUP 程序用于展开这些文件。如果需要其中某个文件的新拷贝，查看第一张盘上的 PACKING.LST 文件可以找出所需的那个文件在哪张盘上，它的压缩文件名和展开后应取的文件名。然后将相应的安装盘插入驱动器，再使用 EXPAND 命令将文件拷贝到 DOS 目录。

EXPAND 命令不展开由其他磁盘压缩程序压缩的文件，如 DoubleSpace、DriveSpace 或 PKZIP。

可以列出多个 filespec，但不能使用通配符。命令行中有多个 filespec 时，EXPAND 认为最后一个是 destination。如果展开多个文件，destination 必须是目录，而不能是文件名。

对于单个的 filespec，destination 可以是目录，也可以是文件名。若忽略 destination，EXPAND 会提示用户输入之。

如果 destination 是目录，展开的文件与压缩文件同名。若文件名以下划线结尾，则作为 DOS 6 文件使用前要改成正确的名字。

应用举例

EXPAND A:\SORT.EX_ C:\DOS\SORT.EXE

从驱动器 A 中的磁盘上展开 SORT.EX_。展开文件名为 SORT.EXE，放在 C:\DOS 目录中。

EXPAND A:\SORT.EX_

从驱动器 A 中的磁盘上展开 SORT.EX_，并提示用户输入目的文件。

EXPAND CGA.IN_ CGA.VI_ C:\DOS

展开当前目录中的文件 CGA.IN_和 CGA.VI_，展开后的文件放在 C:\DOS 中。因为目的是一个目录，所以展开文件与原来文件同名。在使用它们之前应分别改名 CGA.INI 和 CGA.VID。

Ⓔ FASTHELP

见第 12 章

提供 DOS 命令的屏上(on-screen)帮助信息或列出所有命令。

句法

FASTHELP [command]

参数

无参数	列出所有命令并带有简要说明。
command	显示命令的句法概要。

注意

FASTHELP command 命令产生一个简要报告，说明 command 命令的用途和句法，以及命令参数和选项的说明。command/? 产生与 FASTHELP command 同样的报告。

FASTHELP 所提供的信息、注释和例子比命令 HELP 提供的要少。

可以使用任何 ASCII 文本编辑器修改 FASTHELP.HLP 文件以适应用户需要。在其中加入 DOS 或非 DOS 命令的信息就可以通过 FASTHELP 显示出来。加入的信息只影响在执行未指定命令名的 FASTHELP 命令时产生的列表。

应用举例

FASTHELP

分页显示所有命令和每个命令的简要说明。

FASTHELP COPY > PRN

打印 COPY 命令的简要说明及其参数和选项。可以输入 COPY /? >PRN 得到同样的结果。

Ⓔ FASTOPEN

启动 FASTOPEN，使打开常用文件的速度更快。

句法

FASTOPEN drive[=n][...][/x]

参数和选项

drive	指定由 FASTOPEN 跟踪的驱动器。
n	该驱动器上被跟踪的文件数目，从 10 到 999，缺省值是 48。
/X	将驱动器信息存于扩充内存而不是常规内存中。

注意

每次打开文件时，FASTOPEN 记录文件的名字和位置。如果重新打开该文件，FASTOPEN 从缓冲区

而不是磁盘目录结构中取出该文件的位置，从而节省了时间。

FASTOPEN 只能在硬盘本地上使用，但不能用于网络驱动器。

所有驱动器上被跟踪的文件总数不能超过 999。对于每个文件，FASTOPEN 需要大约 48 个字节的空间。

必须重新启动计算机才能改变 FASTOPEN 参数。在系统一次运行期间不能执行两次 FASTOPEN。

可以通过命令行或 CONFIG.SYS(见 INSTALL 命令)装入 FASTOPEN。

如果在 FASTOPEN 运行期间使用了 DEFRAG 命令，那么 DEFRAG 结束后必须重新启动，这样 FASTOPEN 就不会对已移动的文件使用原来的位置记录。

警告： 一些专家认为 FASTOPEN 不可靠，而有些软盘在 FASTOPEN 存在时拒绝运行。在 DOS Shell 中使用 FASTOPEN 命令会锁死系统。

应用举例

```
FASTOPEN C:=100
```

装入 FASTOPEN，跟踪驱动器 C 上的 100 个文件。

```
FASTOPEN C:=100 D: E:=300  /X
```

装入 FASTOPEN，跟踪 C 盘上的 100 个文件，D 盘上 48 个(缺省值)文件和 E 盘上 300 个文件。将 FASTOPEN 的记录保存在扩充内存(expanded memory)中。

Ⓔ FC

见第 16 章

比较两个文件并显示不匹配的行或字节。

句法

```
FC [/B] [/L] [/A] [/C] [/LBlines] [/N] [/T] [/W] [/resynch]filespec1 filespec2
```

参数和选项

filespec1	要比较的第一个文件。
filespec2	要比较的第二个文件。
/B	以二进制方式逐字节比较文件，发现不同之处后不重新同步文件。对于具有扩展名.EXE、.COM、.SYS、.OBJ、.LIB 或.BIN 的文件此为缺省方式。二进制方式下不要使用其他选项，所有其他选项属于 ASCII 方式。
/L	以 ASCII 方式逐行比较文件，发现不匹配后试图重新同步文件。对于扩展名不是.EXE、.COM、.SYS、.OBJ、.LIB 或.BIN 的文件此为缺省方式。
/A	缩短输出，即对于一系列不匹配的行只显示第一行或最后一行。缺省时显示所有不匹配的行。
/C	忽略大小写。
/N	在 ASCII 方式下，显示不匹配的行时带行号。
/LBlines	指定重新同步缓冲区中的行数。缺省值是 100。
/T	不扩展制表符为空格。缺省是将制表符扩展到第 8 个字符的位置上。
/W	忽略空格(white space)。
/resynch	指定文件在重新同步前必须匹配的行数。缺省值是 2。

注意

在任何一个文件全名中都可使用通配符。如果只在 filespec1 中使用通配符，FC 将指定的所有文件与名为 filespec2 的文件相比较。如果两个文件全名中都使用了通配符，FC 则比较文件名中有相同通配符的文件(见应用举例)。

在 ASCII 方式中,FC 发现一处不匹配后会试图找到一处继续比较。它从每个文件读出一些行送到重新同步缓冲区,然后在第一个文件和缓冲区之中寻找匹配的行。如果找到一处匹配,则文件已重新同步,比较从这一点开始继续。否则,FC 放弃比较(要注意它可能不搜寻文件的其余部分)。参数/resynch 决定了要重新同步每列中必须匹配多少行。参数/LB 决定重新同步缓冲区的大小。增加 lines 或减少 resynch 会提高重新同步成功的机会。

由于重新同步的原因,一个文件中多出的行不会引起 FC 认为余下的行都不匹配。

对于 ASCII 比较方式,FC 报告不同之处时显示第一个文件名,这两个文件中匹配的最后一行,以及第一个文件中的不匹配行,然后是两文件中第一个匹配行,对第二个文件也列出类似的信息。

如果 resynch 大于 1,某些匹配行可能被计数并在不匹配行的列表中显示出来,直到指定数目的连续行匹配,否则文件不被重新同步。

若使用/W 选项,FC 则忽略行首的制表符和空格。任何一处连续的制表符或空格(白空格)计作一个空格。换句话说,一行中多余的空格或制表符不会造成不匹配。

在二进制模式下,FC 报告不匹配的内容时以十六进制显示相对于文件头的地址和不匹配的字节;filespec1 中的在前,filespec2 中的在后。

如果两个文件的缺省比较方式不同且未指定/B 或/L,FC 则使用 filespec1 的缺省比较方式。若对 filespec1 的扩展名使用了通配符,FC 就使用缺省的 ASCII 方式。

应用举例

```
FC /LB200 /W JANUARY.DOC NEWYEAR.DOC
```

将 JANUARY.DOC 和 NEWYEAR.DOC 进行 ASCII 比较。若连续 200 行不匹配则停止比较。忽略空格。重新同步要求有两行匹配。

```
FC /B *.OLD EXTRA.SAV
```

将当前目录中每个具有扩展名.OLD 的文件与 EXTRA.SAV 进行二进制比较。

```
FC OLD.EXE E:\*.*
```

将当前目录中的 OLD.EXE 与 E:\下的 OLD.EXE 进行二进制比较。

```
FC /C /1 *.OLD *.SAV
```

将当前目录中每个扩展名为.OLD 的文件与文件名相同但扩展名为.SAV 的文件进行 ASCII 比较,忽略字母大小写。重新同步只要求一行。

© FCBS

见第 21 章

分配文件控制块(FCB)。

句法

```
FCBS=n
```

参数

n 指定要分配的 FCB 数,其值可从 1 到 255。缺省值是 4。

注意

早期的 DOS 版本处理文件时使用文件控制块(FCB)。某些旧程序仍使用这种方式。这样的程序要求在 CONFIG.SYS 中使用命令 FCBS。

较新的文件处理方法参见 FILES 命令。

应用举例

```
FCBS=8
```

指定 DOS 最多可以同时打开 8 个文件控制块。

Ⓔ FDISK

见第 54 章

启动 FDISK 程序，为使用 DOS 的硬盘建立分区。显示硬盘状态。

句法

FDISK [/STATUS] [/MBR]

参数和选项

无参数　　启动 FDISK 程序。

/STATUS　　显示硬盘的分区信息。不启动 FDISK 程序。

/MBR　　重建主引导记录。不启动 FDISK。

注意

不要用分区操作做试验。如果偶然删除或改变了一个盘或分区，可能会丢失其中的数据。

启动后，FDISK 提示用户输入想要完成的任务。

在用 ASSIGN、JOIN 或 SUBST 组织的磁盘上，网络驱动器上或 INTERLNK 盘上无法使用 FDISK。

如果怀疑主引导记录中附有病毒而又没有其他方法去除之，可以使用 FDISK /MBR。选项/MBR 可去掉病毒，但不能修复任何已对主引导记录造成的破坏。使用 FDISK /MBR 后立即重新启动。

应用举例

FDISK /STATUS

显示硬盘的分区信息。

FDISK /MBR

重建硬盘主引导记录。

FDISK

启动 FDISK 程序。

Ⓒ FILES

见第 21 章

设置 DOS 在使用文件句柄访问的文件时可访问的文件数。

句法

FILES=n

参数

n　　指定文件句柄数，可以 8 到 255。缺省值是 8。

注意

DOS 使用文件句柄标记已打开的文件。许多程序需要文件句柄多于 8 个，因而就需要用户在 CONFIG. SYS 中指定 FILES。

除原有 8 个文件句柄之外，每个新分配的文件句柄会使 DOS 体积增大 39 个字节。

应用举例

FILES=20

允许 DOS 同时打开最多 20 个文件。

Ⓔ FIND

见第 36 章

在一个或一组文件中搜寻指定的文本字符串并显示包含该串的行。

句法

FIND [/V] [/C] [/N] [/I] "string" [filespec ...]

参数和选项

"string"	指定要搜寻的一组字符。引号是必需的。
filespec	指定将被搜寻的文件。
/V	显示不包含指定字符串的所有行。
/C	只显示包含指定字符串的行数。
/N	显示中包含行号。
/I	忽略字母大小写。

注意

若忽略 filespec，FIND 从 DOS 标准输入中接收输入，通常是键盘或管道。

不能在 filespec 中使用通配符。但可以列出多个 filespec，中间用空格分开。也可以在 FOR 命令中使用 FIND 来搜寻多个文件(见 FOR 命令)。

如果查找字符串中包含引号，使用两个引号来查找一个。也就是说，"David""Daffy""Jones"查找 David "Daffy" Jones。

同时使用选项/C 和/V 可显示不包含指定字符串的行数。若同时使用/C 和/N，FIND 则忽略/N。

FIND 仅仅是按照字母串中字节的值进行匹配。它不考虑制表符、回车符、连字符以及其他常由字处理器和应用程序插入的符号。尽量查找不会在行尾被截断，也不会被生成文件的应用程序格式化过的字符串。

应用举例

FIND /I /N "TENTH" TEST1.BAT TEST2.BAT

显示文件 TEST1.BAT 和 TEST2.BAT 中包含单词"tenth"的每一行。搜寻中忽略字母大小写，所以也会找到"TENTH"、"tenth"或"Tenth"。每一行都显示行号。

DIR *.* | FIND /V -93

显示目录列表中不包含-93 的每一行。这可以产生一个当前目录中所有未在 1993 年中生成或修改过的文件列表。

FOR %F IN (*.BAT) DO FIND /I "FORMAT" %F

搜寻当前目录中所有的.BAT 文件，列出所有包含单词"FORMAT"的行，忽略字母大小写。

---

① FOR

Ⓑ 见第 37 章

对集合中的每一项重复执行一条命令。

句法

FOR %x | %%x IN (set) DO command

参数和选项

x	标志一个用于 command 中的可替换变量。可以是任何非数字字符。%x 用在命令提示符下，而%%x 用于批处理程序中。
(set)	用来替换 x 的集合，括号是必需的。
command	需要重复执行的命令，包括参数和选项。

注意

FOR 对于 set 中的每一项重复执行 command。在 command 中使用可替换变量(%x 或%%x)指示何处

需由 set 中的项来替换。

用空格分隔 set 中的各项。

当 set 中包含一个带通配符的文件名时，FOR 对于每个与该文件名匹配的文件重复 command 命令。

command 不能是另一个 FOR 命令。

可以使用＞或＞＞重定向 command 的输出，但不能使用管道。

应用举例

FOR %R IN (A B C D) DO DIR %R:\ > PRN

用在命令提示符下，打印驱动器 A、B、C、D 的根目录列表。

FOX %%R IN (A B C D) DO DIR %% R:\> PRN

用于批处理程序中，打印驱动器 A、B、C、D 的根目录列表。

FOR %F IN (*.SAV *.OLD) DO FIND ECHO %F

对当前目录中每个扩展名为.SAV 或.OLD 的文件，显示含有 ECHO 的行。用于命令提示符下。

FOR %%F IN (*.SAV *.OLD) DO FIND ECHO %%F

对当前目录中每个扩展名为.SAV 或.OLD 的文件，显示含有 ECHO 的行。用于批处理程序中。

Ⓔ FORMAT

见第 17 章

使磁盘能够使用或重新使用。

句法

FORMAT drive [/Q][/U][/V[:label]][/S|/B][/F:size][/T:tracks /N:sectors][/1][/4][/8][/C][/AUTOTEST][/BACKUP]

参数和选项

drive	需要格式化的驱动器。
/Q	快速格式化。
/U	进行无条件格式化。
/V[:label]	指定格式化后磁盘的卷标。label 最多可有 11 个字母。若省略/V 选项，或使用/V 但省略 label，DOS 则在格式化完成后会提示用户输入卷标。它不能与/8 选项同时使用。
/S	使磁盘成为可启动盘。
/B	在新格式化的磁盘上为系统文件保留空间。
/F:size	覆盖驱动器的缺省容量。以字节为单位指定 size，如 360KB 或 1.2MB。此选项不与/T、/N、/1、/4 或/8 选项同时使用。
/T:tracks	指定盘上磁道数，必须与/N 联用。不与/F 同时使用。
/N:sectors	指定每个磁道的扇区数，必须与/T 联用。不与/F 同时使用。
/1	格式化单面磁盘。不与/F 同时使用。
/4	在 1.2MB 驱动器中格式化 360KB 的磁盘。若与/1 联用，则在 1.2MB 驱动器中格式化 180KB 的磁盘。不与/F 同时使用。
/8	每道 8 扇区格式化 5 ¼英寸磁盘(用 DOS 2.0 或更早版本)。不与/F 或/V 同时使用。
/C	格式化时重新测试坏簇。
/AUTOTEST	格式化时不带任何提示信息，也不产生磁盘空间的报告。
/BACKUP	格式化时除输入卷标外不作任何提示。产生磁盘空间报告。

注意

如果指定选项/U(但未用/Q)，或者改变了磁盘容量，或者磁盘未被格式化过，那么 DOS 进行无条件

格式化。划分新扇区、磁盘表面测试以寻找坏区、装入引导扇区并生成根目录和FAT。磁盘上已有的任何数据丢失。(对于未格式化过的磁盘,使用/U 选项比让 DOS 去发现磁盘未格式化过要快一些。)

如果对于已格式化过的磁盘未指定/U 或/Q 选项,也未改变磁盘容量,DOS 就使用安装格式化。它将FAT 清零,清除根目录,检查坏簇,并为 UNFORMAT 命令保存信息。已有数据不被毁掉。这比无条件格式化快,并且如有必要,可用 UNFORMAT 恢复已有数据。

在检查坏簇时,FORMAT 认为已被标记的坏簇仍为坏簇。除非指定/C 选项,否则不重新检验。用 DISKCOPY 拷贝的盘中可能会有从源盘中拷贝来的坏簇标记。格式化时用/C 选项可以更正错误的标记,但会降低格式化速度,尤其是有很多簇标记为坏簇时。在 DOS 6.2 之前,FORMAT 总是重新检查坏簇。

如果对于已格式化过的磁盘指定/Q 选项,并且未改变磁盘容量,DOS 就执行快速格式化。除不检测坏簇外,它与安装格式化相同,但比安全格式化更快,并且如有必要,可用 UNFORMAT 来恢复原有数据。

同时使用/Q 和/U 选项是最快的格式化操作。DOS 在可能情况下使用快速格式化,但不保存 UNFORMAT 信息。因而不能使用 UNFORMAT 从盘中恢复原有数据。

如果磁盘在读写时发生未找到扇区(sector not found)错误,用/U 选项格式化以更新扇区。注意所有数据将被永久擦除。

不要用比出厂容量更高的数值格式化磁盘。例如,不要把 360KB 的软盘格式化成 1.2MB(但允许把 1.2MB 磁盘格式化成 360KB)。某些 360KB 驱动器不能正确读写由 1.2MB 驱动器格式化的 360KB 磁盘。

当不使用任何容量选项(/F、/T、/N、/1、/4 或/8)时,无条件格式化使用驱动器容量。安装或快速格式化时使用磁盘当前容量。

size 的允许值及意义如下:

160 或 160K 或 160KB	160KB、单面、倍密度、5 ¼英寸盘。
180 或 180K 或 180KB	180KB、单面、倍密度、5 ¼英寸盘。
320 或 320K 或 320KB	320KB、双面、倍密度、5 ¼英寸盘。
360 或 360K 或 360KB	360KB、双面、倍密度、5 ¼英寸盘。
720 或 720K 或 720KB	720KB、双面、倍密度、3 ½英寸盘。
1 200 或 1 200K 或 1 200KB 或 1.2 或 1.2M 或 1.2MB	1.2MB、双面、高密度、5 ¼英寸盘。
1 440K 或 1 440K 或 1 440KB 或 1.4 或 1.4M 或 1.4MB	1.44MB、双面、高密度、3 ½英寸盘。
2 880 或 2 880K 或 2 880KB 或 2.8 或 2.8M 或 2.8MB	2.88MB、双面、超高密度、3 ½英寸盘。

DOS 5.0 以前的版本要求两个 DOS 系统文件在磁盘上相邻。如果格式化磁盘之后要拷贝 DOS 5.0 之前版本的系统文件,则需要/B 选项为这两个系统文件保留一块足够大的空间(见 SYS 命令)。如果用了/S 选项或永远不需向磁盘中装入 DOS 5.0 之前版的系统文件,则不需要/B 选项。

FORMAT 格式化了一张磁盘后,会询问是否还要格式化另一张。若回答是,FORMAT 则用当前参数格式化下一张磁盘,而没有改变参数的机会。

如果用了/AUTOTEST 或/BACKUP 选项,FORMAT 则不提示插入磁盘也不询问是否格式化另一张。/AUTOTEST 还不提示用户输入卷标,也不产生磁盘空间列表。这两个选项中的任何一个都使 FORMAT 在批处理程序中更易于使用,而在命令提示符下则更快。但是,如果没有在正确的驱动器中插入正确的磁盘,这样的用法就很危险。

当 drive 代表硬盘时,FORMAT 会警告用户将丢失所有数据并提供一个停止格式化操作的机会。不能对由 SUBST 组织的盘使用 FORMAT。也不能在网络驱动器或 INTERLNK 驱动器上使用 FORMAT。

若卷标中包含有空格,用引号括号起来,如/V:"Don's Disk"。也可以在括号中使用空值禁止卷标,如/

V:" "。此方法可以阻止 FORMAT 在格式化过程中询问卷标。

FORMAT 返回码如下：

0	格式化成功。
3	用户按下 Ctrl+C 或 Ctrl+Break 键中止格式化过程。
4	出现致命错误(除 0、3、5 外的任何错误)。
5	在出现警告信息后用户选择了不继续执行。

应用举例

FORMAT A:

用缺省容量格式化驱动器 A 中的磁盘。若磁盘已格式化过则使用安全格式化(使用户可能恢复格式前内容)，否则，使用无条件格式化。

FORMAT B: /F:360K /S /V:DATA1993 /U

把 B 驱动器中的磁盘格式化成 360KB 的启动盘，使用无条件格式化(无法再恢复磁盘以前的内容)。同时赋予卷标 DATA1993。

Ⓑ GOTO

见第 37 章

用于批处理程序中，使 DOS 转到有指定标号的行。

句法

GOTO label

参数

label	标识批处理程序应跳转到的行。

注意

正常情况下 DOS 逐行处理批处理程序。GOTO 命令改变处理顺序，使 DOS 转到由一个冒号后面跟随 label 组成的行。DOS 继续执行该行后面一行开始的命令。

label 中可以包含空格，但不能有其他分隔符，如分号或等号。GOTO 只注意 label 的前 8 个字符，所以：NEXTCOMMAND 和:NEXTCOMMANDS 都与:NEXTCOMM 等效。

应用举例

GOTO:ENDPROG

使 DOS 转到由:ENDPROG 组成的行中并从这一处继续处理。

GOTO %CONFIG%

将 DOS 转到标号与环境变量 CONFIG 匹配的一行。

Ⓔ GRAPHICS

见第 50 章

装入 GRAPHICS 内存驻留程序(TSR)使 DOS 能够打印屏幕上的图形。

句法

GRAPHICS [type] [filespec] [/R] [/B] [LCD] [/PRINTBOX:size]

参数和选项

无参数	以缺省值装入 GRAPHICS TSR。
type	指定打印机类型。可能值见注释。缺省值是 HPDEFAULT。
filespec	指定打印机概要(profile)文件。缺省是 GRAPHICS. PRO。

/R	指定图象应打印成与屏幕上一样(正片)。缺省情况下打印反相图象,或负片,这样打印出的图象在某些情况下会好一些(见注释)。
/B	对于 COLOR4 或 COLOR8 打印机用彩色打印背景。
/LCD	用 LCD 而不是 CGA 宽高比打印图象。此选项效果与 /PRINTBOX:LCD 相同。
/PRINTBOX:size	指定打印框(printbox)的大小。可以把/PRINTBOX 缩写成/PB。size 必须是 LCD 或 STD。它必须与打印机概要(profile)文件中的 PRINTBOX 语句的第一个操作数匹配。

注意

合法的打印机类型如下:

COLOR1	IBM PC Color Printer,黑色带。
COLOR4	IBM PC Color Printer,RGB 色带。
COLOR8	IBM PC Color Printer,CMY 色带。
DESKJET	Hewlett-Packard DeskJet 打印机。
HPDEFAULT	任何型号的 Hewlett-Packard PCL 打印机。
GRAPHICS	IBM Personal Graphics Printer Proprinter 或 Quietwriter,以及大部分 Epson 点阵打印机。
GRAPHICSWIDE	任何型号的 GRAPHICS 宽行打印机。
LASERJET	Hewlett-Packard LaserJet 打印机。
LASERJETII	Hewlett-Packard LaserJet II 打印机。
PAINTJET	Hewlett-Packard PaintJet 打印机。
QUIETJET	Hewlett-Packard QuietJet 打印机。
QUIETJETPLUS	Hewlett-Packard QuietJet Plus 打印机。
RUGGEDWRITER	Hewlett-Packard RuggedWriter 打印机。
RUGGEDWRITERWIDE	Hewlett-Packard RuggedWriter Wide 打印机。
THERMAL	IBM PC 两用 Thermal 打印机。
THINKJET	Hewlett-Packard ThinkJet 打印机。

选择最接近用户打印机的类型。但 GRAPHICS 不会使用 PostScript 打印机,除非它能模拟上面列出的打印机。

如果省略 filespec,GRAPHICS 在与 GRAPHICS.COM 相同的目录或当前目录中寻找 GRAPHICS.PRO。

一旦 GRAPHICS 装入,就可以按 PrintScreen 键打印图形。

一般说来,对于黑白图形在白色背景上打印黑色字母看起来效果最好,打印也最快。为得到这一效果,对于某些图形程序如 DOS Shell 应使用/R 选项。

正常情况下可以信赖由 GRAPHICS 为打印机指定正确的宽高比(LCD 或 STD)。如果发现比例不很正确(如把圆画成了椭圆),检查打印机概要(profile)参数以查看用户的显示器是否对 LCD 和 STD 都支持。可以用/LCD 或/PB:LCD 指定 LCD 宽高比。用/PB:STD 指定标准宽高比。

如果新的概要(profile)参数占用的总空间比原来的大,则需要重启动以改变这些参数。

应用举例

GRAPHICS

装入 GRAPHICS TSR,使用缺省值:打印机类型 HPDEFAULT,概要(profile)文件 GRAPHICS.PRO,反相屏幕图象,由 GRAPHICS.PRO 决定 PB 的大小。

GRAPHICS GRAPHICS

如果还没有装入 GRAPHICS TSR，用 GRAPHICS 打印机类型将其装入，打印机概要、反相屏幕图象和 PB 大小使用缺省值。如果 TSR 已装入，则转换到 GRAPHICS 打印机类型。

GRAPHICS /R

如果尚未装入 GRAPHICS TSR，则以打印机类型、打印机概要和 PB 大小的缺省值将其装入，但使用屏幕图象正片(即非反相图象)。若 GRAPHICS TSR 已装入，则转换到屏幕图象正片拷贝。

Ⓔ HELP

见第 12 章

启动 DOS 命令帮助程序。

句法

HELP [topic] [/B] [/G] [/H] [/NOHI]

参数

无参数	显示可实现帮助的主题列表。
topic	显示一段文字描述 topic 的句法。Topic 可以是 HELP 内容列表中的任何主题，包括所有 DOS 命令。
/B	指定黑白(单色)显示。
/G	快速更新 CGA 显示屏(但有可能出现屏幕雪花点)。
/H	在显示屏上显示尽量多的行。
/NOHI	表示显示器不支持高亮度。

注意

HELP 命令提供的信息比 FASTHELP 多得多，但是，如想要命令句法的快速、简捷的信息，使用 FASTHELP。

指定 HELP command--NOTES 或 HELP command-—EXAMPLES 可以直接看到命令的注释或应用举例，如 HELP PATH--NOTES 或 HELP PATH-—EXAMPLES。

应用举例

HELP

加载 HELP，显示内容列表。

HELP COPY

加载 HELP 并显示 COPY 命令的第一段文字(句法)。

Ⓑ IF

见第 37 章

在批处理程序中实现条件处理。

句法

IF [NOT] expression command

参数和选项

NOT	只在 expression 为假时执行 command。否则 DOS 只有在 expression 为真时执行 command。
expression	应该是 ERRORLEVEL、EXIST 或 IF 等可以计算其真假值的条件等式。表达式的三种形式是：

ERRORLEVEL n
string1==string2
EXIST filename

command 指定一个由 DOS 执行的命令。

注意

表达式 ERRORLEVEL n 在前面程序的返回码大于等于 n 时为真。否则其值为假。例如，ERRORLEVEL 1 在返回码是 0 时为假，返回码是 1 或更大值为真。

表达式 EXIST filespec 在 filespec 表示已存在文件时为真，否则为假。

表达式 string1==string2 在 string1 与 string2 相同时为真。检测要区分字母大小写。例如，语句 x==X 为假。

字符串可以是可替换参数，如%1==Y 或%FACTOR%==X。如果 string1 和 string2 中任何一个所代表的可替换参数可能为空值，那么用引号将两个都括起来，如"%1"="Y"。IF 命令中的空字符串会引起句法错。

应用举例

```
FORMAT A:
IF ERRORLEVEL 6 GOTO UNKNOWN
IF ERRORLEVEL 5 GOTO UENDED
IF ERRORLEVEL 4 GOTO FATAL
IF ERRORLEVEL 3 GOTO INTERRUPT
IF ERRORLEVEL 1 GOTO UNKNOWN
ECHO * * * * FORMAT SUCCESSFUL
GOTO ENDING
:UNKNOWN
ECHO * * * * FORMAT ENDED WITH UNKNOWN ERROR CODE
GOTO ENDING
:UENDED
ECHO * * * * USER DECIDED NOT TO PROCEED WITH FORMAT
GOTO ENDING
:FATAL
ECHO * * * * FATAL ERROR WHILE FORMATTING
GOTO ENDING
:INTERRUPT
ECHO * * * * USER INTERRUPTED FORMAT
:ENDING
```

IF 这个批处理程序使用了几个 IF 命令，它根据 FORMAT 提供的返回码显示不同的信息。

```
IF NOT EXIST MAKEBAT.DAT NEWPROG.BAT
ECHO FILE ALREADY EXISTS:NEWPROG CAN'T RUN
```

如果名为 MAKEBAT.DAT 的文件不存在，则运行批处理程序 NEWPROG.BAT，DOS 不会再返回处理批处理程序中的下一行。若文件存在则不执行 NEWPROG，DOS 在下一行显示一条信息。

```
IF %1==T TYPE  %2
IF %1==P PRINT  %2
```

此程序显示或打印用户指定的文件。采取的动作根据用户通过%1 传递的参数决定。注意，如果用户输入错误的参数，如 t、p、x 或没有参数，则不采取任何动作。

© INCLUDE

见第 38 章

用于 CONFIG.SYS 中，在此处包含一个系统设置语句块的内容。

句法

INCLUDE=blockname

参数

blockname　　代表一个系统设置块。

注意

只有 CONFIG.SYS 中包含多重设置时才能使用 INCLUDE。关于在 CONFIG.SYS 中为多重设置建立使用菜单的信息请参见 MENUITEM、MENUDEFAULT、MENUCOLOR 和 SUBMENU 命令。

系统设置块中包含一套 CONFIG.SYS 命令。用方括号括起来的块名(blockname)表示系统设置块的开始。每块以下一个块名或文件尾为结束。

DOS 可能要在用户的几种系统设置中执行相同的命令。把这些命令定义在单独的块中并使用 INCLUDE=blockname，在其他系统设置块中执行这些命令。

应用举例

INCLUDE=BASIC

处理[BASIC]块中的命令，然后继续当前系统设置块的余下部分。

© INSTALL

见第 49 章

在 CONFIG.SYS 中装入 TSR。

句法

INSTALL=filespec [parameters]

参数

filespec　　要装入的内存驻留程序。

parameters　　程序需要的所有参数和选项。

注意

用 INSTALL 装入程序占用的内存比在批处理程序中或命令提示符下装入同样的程序略少一些，因为它在内存中没有自己的环境空间。

警告： *某些程序用 INSTALL 装入后可能不正常运行。不要装入使用环境变量、热键或要求 COMMAND.COM 存在以处理严重错误的程序。*

DOS 中能够使用 INSTALL 安全装入的 TSR 有：FASTOPEN.EXE、KEYB.COM、NLSFUNC.EXE 和 SHARE.EXE。

DOS 在处理完所有 DEVICE 命令之后和装入命令解释器之前处理全部 INSTALL 命令，而不考虑命令在 CONFIG.SYS 中的顺序。

应用举例

INSTALL=C:\DOS\SHARE.EXE /L=25

装入 Share TSR 并传递参数 /L=25

Ⓔ INTERLNK

见第 41 章

完成客户机和服务器计算机间的 INTERLNK 连接。重定向服务器驱动器。

注释： *关于 INTERLNK 设备驱动程序的说明参见附录 A 的 INTERLNK.EXE。*

句法

INTERLNK [drive1=[drive2]] [...]

参数

无参数	必要时完成连接。显示状态报告,包括当前驱动器和打印机的设定。
drive1	赋予 drive2 的客户机驱动器名。
drive2	指定一个由名字 drive1 访问的服务器驱动器。

注意

客户计算机必须从 CONFIG.SYS 中装入 INTERLNK.EXE 设备驱动程序,服务器计算机上要启动 INTERSVR,并以电缆连接两台计算机,然后才能使用本命令。关于连接两台计算机的更多信息参见 INTERSVR。

使用本命令可以完成两台计算机的连接,显示连接状态,或重新指派客户驱动器名。

原来在 INTERLNK 设备驱动程序装入时指派的驱动器名可以由 INTERLNK 命令重新指派。两个驱动器必须都在由 INTERLNK 设备驱动程序连接的驱动器中。如果服务器的驱动器 A、B、C 被分别指派给客户驱动器符 D、E、F,INTERLNK 可以重新安排这些指派,但服务器即使有 D 驱动器,也不能将其指派给一个客户驱动器符。

若忽略 drive2,客户机驱动器指派被删除。

驱动器中的冒号是可选的。

应用举例

INTERLNK

完成客户机和服务器计算机的连接。

INTERLNK F:=D:

重新指派客户机驱动器名 F 到服务器驱动器 D,假定两个驱动器在 INTERLNK 装入时都是已被指派的。

INTERLNK F=

删除对客户机驱动器名 F 的指派。

Ⓔ INTERSVR

见第 41 章

装入 INTERLNK 服务器软件并重定向驱动器和打印机端口。从一台计算机向另一台拷贝 INTERLNK 文件。

句法 1

INTERSVR [drive...] [/X=drive...] [/LPT[n]|/LPT:[address]|/COM [n]|/COM: [address]] [BAUD:rate] [/B] [/V]

装入 INTERLNK 服务器软件,重定向驱动器和打印机端口。

参数和选项

无参数	装入服务器程序,显示服务器屏幕。
drive	需要重定向的服务器驱动器。缺省情况下,INTERSVR 重定向所有服务器驱动器。
/X=drive	不需要重定向的服务器驱动器。
/LPT	说明电缆接在并行口上。
/COM	说明电缆接在串行口上。
n	标明电缆所接的端口号。
address	连接电缆的端口地址。

/BAUD:rate　设置数据传输的最大波特率。rate 的值可以是 9 600、19 200、39 400、57 600 和 115 200。缺省是 115 200。

/B　以黑白方式显示 INTERLNK 服务器屏幕。

/V　防止与服务器计算机的定时器发生冲突。

注意

要让两台计算机共享驱动器和打印端口，用电缆连接两个串行号或两个并行口。然后在一台计算机（称为服务器计算机）上启动 INTERSVR，在另一台计算机（称为客户计算机）上从 CONFIG. SYS 中装入 INTERLNK 设备驱动程序。如果在客户计算机上装入设备驱动程序后才启动 INTERSVR，需要用 INTERLNK 命令完成连接（参见 INTERLNK）。

当装入 INTERLNK 设备驱动程序时，它将客户计算机上未用的驱动器符指派给服务器驱动器。例如，客户计算机已有驱动器 A、B、C，服务器计算机上也有三个驱动器，则客户计算机使用字母 D、E、F 访问服务器计算机的驱动器。类似地，客户机上未用的并行端口名（如 LPT2）被指派给服务器的并行口。

命令 INTERSVR 可以改变驱动器指派的顺序，也可以使某些驱动器不被客户计算机访问。例如，INTERSVR C：B：A：，使 INTERLNK 指派驱动器符 D 到服务器驱动器 C，E 到服务器驱动器 B，F 到服务器驱动器 A。而命令 INTERSVR /X＝A：/X＝B：阻止 INTERLNK 指派驱动器符到服务器的软驱。

INTERLNK 不重定向网络驱动器、CD-ROM 驱动器、或其他任何使用重定向接口的驱动器。这些驱动器不能直接被客户机访问。

选项/LPT 或/COM 告诉 INTERSVR 客户计算机连接在服务器的哪一个端口上。（INTERLNK 有一个类似的选项指定客户机端口）。

如果/LPT 和/COM 都未指定，INTERSVR 扫描所有的并行口和串行口以寻找与客户机的连接。

如果用了/LPT 但不带 n 或 address，INTERLNK 则使用它找到的第一个并行口连接客户机。

如果用了/COM 但不带 n 或 address，INTERLNK 则使用它找到的第一个串行口连接客户机。

若从正使用鼠标的 Windows 下启动 INTERSVR，应使用/LPT 或/COM 选项指示客户机所连接的端口以避免 INTERSVR 扫描鼠标器端口，这会使鼠标驱动程序失去作用。

如果两台计算机是串行连接的，而在 INTERLNK 访问驱动器或打印机端口时其中一台出现停机，则应使用/V 选项。

以下命令不能用于通过 INTERSVR 连接的驱动器上：CHKDSK、DEFRAG、DISKCOMP、DISKCOPY、FDISK、FORMAT、MIRROR、SYS、UNDELETE 和 UNFORMAT。

INTERSVR 接管服务器计算机而不允许任何其他程序在该机上运行（多任务和任务切换被禁止）。在服务器上按 Alt＋F4 可以中断连接。

驱动器名的冒号是可选的。

应用举例

```
INTERSVR /X＝A：/COM2
```

从已连接了 COM2 的计算机上启动 INTERSVR，允许客户机访问除 A 之外的所有驱动器。

```
INTERSVR C D /LPT
```

从已连接了某个并行端口的计算机上启动 INTERSVR。只允许客户机访问驱动器 C 和 D。

句法 2

```
INTERSVR /RCOPY
```

从一台计算机拷贝 INTERLNK 程序文件到另一台。

注意

本命令用于在没有 INTERLNK 的计算机上安装 INTERLNK。两台计算机必须用一根空调制解调的七线电缆连接串行口。可以在以前的 DOS 版本下使用 INTERLNK 命令，但是该版本中必须含有 MODE

命令。

如果目录计算机上使用的端口是COM1，要保证计算机上没有运行SHARE程序。如果有，重新启动时不加载SHARE。

INTERLNK文件被拷贝到目标计算机的当前目录下。

在资源计算机的命令提示符下输入INTERSVR /RCOPY命令。INTERSVR显示详细信息指导用户下一步的操作。

Ⓔ KEYB

见第49章

装入KEYB TSR，让用户可以使用非标准的键盘布局。

句法

KEYB [code [,[page] [,filespec]]] [/E] [/ID:id]

参数和选项

无参数	显示当前键盘代码和代码页，以及当前主控代码页。
code	表示使用此国家的键盘。
page	要用的代码页(字符集)。缺省是当前代码页。
filespec	指示键盘定义文件的位置和文件名。缺省值是当前目录或程序查找路径中的KEYBOARD.SYS。
/E	表示已在8086计算机上安装了增强型键盘。
/ID:id	对于键盘布局不止一种的国家，指定在使用的布局。

注意

KEYB命令指定键盘布局，以及是否确实安装了这种键盘。例如，如果用户有一个美国标准键盘，而使用KEYB模拟拉丁美洲键盘，则标有分号的键会产生字符ñ。

用KEYB可以改变键盘而无需COUNTRY命令。如果还要改变其他国家相关的信息，则需使用COUNTRY命令。

若未指定page，使用该国家的缺省代码页。然而无论是否指定page，对于大于127的ASCII码仍显示当前主控代码页中的字符。(关于改变主控设备和打印机的代码页的更多信息，参见CHCP和MODE命令)。

如果要同时改变键盘布局和代码页，那么应在改变键盘布局之前改变代码页(参阅MODE CP PREP命令)。

假如键盘文件是KEYBOARD.SYS并位于程序查找路径中，则不需要filespec参数。

按Ctrl+Alt+F1可以从当前KEYB设置切换到缺省键盘设置(US代表购于美国的大部分键盘)。按Ctrl+Alt+F2键切换回KEYB设置。按Ctrl+Alt+F7切换到打字机方式。

参数code、page和id的正确值如下：

国家	代码	页	Id
比利时	BE	850	437
巴西	BR	850	437
法语加拿大	CF	850	863
捷克	CZ	852	850
斯洛伐克	SL	852	850
丹麦	DK	850	865
芬兰	SU	850	865

法国	FR	850	437
		120	189
德国	GR	850	437
匈牙利	HU	852	850
意大利	IT	850	437
		141	142
拉丁美洲	LA	850	437
荷兰	NL	850	437
挪威	NO	850	865
波兰	PL	852	850
葡萄牙	PO	850	860
西班牙	SP	850	437
瑞典	SV	850	437
瑞士(法语)	SF	850	437
瑞士(德语)	SG	850	437
英国	UK	850	437
		166	168
美国	US	850	437
南斯拉夫	YU	852	850

KEYB 设置的返回码如下：

0　　键盘定义文件已自动装入。

1　　无效键盘码、代码页或句法错。

2　　键盘定义文件损坏或不存在。

4　　与设备 CON 的通信发生错误。

5　　没有所要求的代码页。

除在命令提示符下使用 KEYB 命令外，还可以从 CONFIG. SYS 中装入 KEYB. COM(参见 INSTALL 命令)。

应用举例

```
KEYB
```

显示键盘的当前国家、代码页和当前 CON 代码页。

```
KEYB PO,850
```

启动 KEYB，模拟葡萄牙键盘，使用代码页 850。

```
KEYB FR,, C:\COUNTRY\KEYBOARD. SYS /ID:189
```

启动 KEYB，模拟 ID 为 189 的法国键盘，使用当前代码页。使用 C:\COUNTRY 下的键盘定义文件。

Ⓔ LABEL

见第 18 章

显示和改变磁盘卷标。

句法

```
LABEL [drive] [label]
```

参数

无参数　显示当前磁盘的卷标并允许用户将其修改或删除。

drive　指定要显示或修改其卷标的盘。缺省盘是当前盘。

label　为磁盘指定新卷标。label 可以含有最多 11 个字符(限制见注释)。

注意

若未指定 label,DOS 显示磁盘的当前卷标和系列号(如存在),并提示输入新卷标。按回车键则删除当前卷标。

卷标中可含有空格,但不能有制表符。连续的空格作为一个空格。卷标中不能使用以下字符:

```
* ? /\ |. , ; : + = [] () & ^ <> "
```

命令 LABEL 不用于由 ASSING、JOIN 或 SUBST 产生的驱动器上。

应用举例

```
LABEL
```

显示当前驱动器中磁盘的卷标和系列号(若存在)并提示用户输入新卷标。

```
LABEL B: TAX 1993
```

把驱动器 B 中磁盘的卷标改成 TAX1993。

© LASTDRIVE

见第 39 章

指定最高的驱动器符,也就指定了系统所能访问的最多驱动器数。

句法

```
LASTDRIVE=x
```

参数

x　指定一个从 A 到 Z 的驱动器符。最小值是系统所使用的最大驱动器符。

注意

缺省的驱动器数是 5(A 到 E)或实际已有的数目,选其中较大者。

压缩盘,RAM 驱动器、网络驱动器以及其他类型额外的驱动器(比如由 DRIVER.SYS 生成的从属驱动器)都需要驱动器符。可能需要使用 LASTDRIVE 命令以使系统有更多的驱动器符可用。

为分配更多的驱动器,可以在 CONFIG.SYS 中使用 LASTDRIVE 命令带上所需的最大驱动器符。DOS 为每个驱动器分配大约 100 字节的内存,所以 LASTDRIVE 使用超出所需的值只会浪费内存。

应用举例

```
LASTDRIVE=K
```

使计算机可以访问最多 11 个逻辑驱动器。

Ⓘ LH

见第 30 章

把程序装入高端内存。LOADHIGH 可与 LH 互换使用。

句法

```
LH [[/L:region[,min] [;region] [,min]]...] [/S]] filespec [parameters]
```

参数和选项

filespec　要装入的程序。

/L:region　将程序装入高端内存的指定区域。

min	程序运行时在一个区域内所需的最小空间。
/S	装入程序时把 UMB 空间收缩到最小。此选项通常由 MEMMAKER 使用。
parameters	程序所需的所有命令行参数。

注意

如果所用处理器是 80386 或更高档处理器，装入了 EMM386.EXE，并已申请 UMB 支持(参见命令 DOS)，那么将可以使用高端内存块(UMB)。若空间足够，LH 把程序装入高端内存，否则 LH 尽量将其装入常规内存。

在缺省情况下，LH 把程序装入到最大的空闲 UMB 中并使其他所有 UMB 可由程序使用。选项/L 和/S 可以覆盖这些缺省动作，但通常只由 MEMMAKER 来使用。

命令 MEMMAKER 把 AUTOEXEC.BAT 中的适当命令自动转换到 LH，并为优化内存设置而生成所需的/L 和/S 选项。

应用举例

```
LH C:\TEST1\NEWDATA
```

若空间允许将 NEWDATA 程序装入高端内存。否则尽可能装入常规内存。

Ⓔ LOADFIX

见第 29 章

在第一个 64KB 以上的空间装入运行程序。

句法

```
LOADFIX command
```

参数

command	启动程序所需的命令，包括命令所需的参数和选项。

注意

如果装入 DOS 或设备驱动程序到 HMA(高内存区)，则其他程序就可能被装入到本来由 DOS 和设备驱动程序占据的内存低端。某些程序装入内存的第一个 64KB 时不能正常工作。如果启动程序时出现信息 Packed file corrupt，试用 LOADFIX 将其强制装入到 64KB 之上。

应用举例

```
LOADFIX DREPORT JAN /C
```

装入名为 DREPORT 的程序到 64KB 之上。JAN /C 是传递给 DREPORT 的命令行参数。

LOADHIGH

见 LH。

Ⓘ MD

见第 14 章

创建新目录。可与 MKDIR 互换使用。

句法

```
MD [path]dirname
```

参数

path	指定新目录的父目录。
dirname	指定新目录名。

注意

path 中可包含驱动器名，如果没有，DOS 就在当前驱动器上创建 dirname 目录。

若省略 path，DOS 在当前目录下创建名为 dirname 的子目录。

dirname 的完整路径从根目录直到 dirname，包括反斜线，不能超过 63 个字符。

父目录内的 dirname 必须是唯一的。它不能与父目录中的目录或文件同名。

应用举例

```
MD NEWSTUFF
```

创建目录 NEWSTUFF 作为当前目录的子目录。

```
MD E:\NEWSTUFF
```

创建目录 NEWSTUFF 作为 E 盘根目录的子目录。

```
MD \NEWSTUFF
```

创建目录 NEWSTUFF 作为当前驱动器根目录的子目录。

```
MD ..\NEWSTUFF
```

创建目录 NEWSTUFF 作为当前目录父目录的子目录。

Ⓔ MEM

见第 29 章

显示当前内存使用情况。

句法

```
MEM [/CLASSIFY|/DEBUG|/FREE|/MODULE program] [/PAGE]
```

选项

无选项	显示系统已用和未用内存的概要信息。
/CLASSIFY	除概要信息外还列出当前已装入的程序和它们的内存使用情况。缺省为/C。
/DEBUG	列出已装入内存的程序和驱动程序。为每个模块显示大小、段地址和模块类型。也显示概要信息，该选项的缩写形式为/D。
/FREE	显示内存的空闲区域。显示常规内存中每个空闲区域和高端内存每个区域中最大的 UMB 的地址和大小。也显示概要信息。该选项的缩写形式为/F。
/MODULE	显示某个程序当前使用内存的情况。列出程序占用的内存区域并列出每个区域的地址和大小。该选项的缩写形式为/M。
program	指定一个已装入内存的程序。
/PAGE	满屏后暂停。该选项的缩写形式为/P。

注意

MEM 命令显示扩展内存(extended memory)的状态(若已安装)。

只有当系统具备扩充内存(expanded memory)的状态(若已安装)。

只有当某个 UMB 提供者，如 EMM386 已经安装并且在 CONFIG.SYS 中含有 DOS=UMB，MEM 才显示高端内存的状态。若在 Windows 3.0 下运行，MEM 则不显示高端内存的状态。

应用举例

```
MEM /C /P
```

列出当前已装入的程序和每个程序的内存使用情况。显示一整屏后暂停。

```
MEM /M DOSKEY
```

显示 DOSKEY 所用内存的每一区域的段地址、大小和模块类型。如果 DOSKEY 的任何一部分使用了

高端内存，则显示每个 UMB 中被 DOSKEY 占用的区域。

Ⓔ MEMMAKER

见第 30 章

将设备驱动程序和 TSR 移到高端内存以优化内存的使用。

句法

MEMMAKER[/B][/BATCH][/SWAP:drive][/T][/UNDO][/W:size1,size2][/SESSION]

参数和选项

无参数	启动 MEMMAKER。
/B	用黑白方式运行 MEMMAKER。
/BATCH	用批处理方式运行 MEMMAKER，对所有输入使用缺省值。如果发生错误，MEMMAKER 恢复 CONFIG. SYS 和 AUTOEXEC. BAT，必要时还要恢复 Windows 的 SYSTEM. INI 文件。所有状态信息送到 MEMMAKER. STS 文件中。
/SWAP:drive	指定启动驱动器，见注释。
/T	禁止检测 IBM 令牌环(Token-Ring)网络。当计算机使用了这样的网络并且在运行 MEMMAKER 时出现问题，使用此选项。
/UNDO	取消 MEMMAKER 最近一次的改变，恢复原来的 CONFIG. SYS、AUTOEXEC. BAT，及必要时恢复 Windows 的 SYSTEM. INI 文件。
/W:size1,size2	以 K 字节为单位指定要为 Windows 的转换缓冲区(translation buffers)保留多少高端内存。(Windows 至少使用两块内存作转换缓冲区)。缺省值是 12,12。
/SESSION	选项 SESSION 由 MEMMAKER 本身加到 MEMMAKER 上。用户不要使用此选项。

注意

运行 MEMMAKER 需要 386 或更高档的处理器。

如果只用 Windows 应用程序则不需要运行 MEMMAKER。

不能从其他程序，如 Microsoft Windows 和 DOS Shell 中运行 MEMMAKER。启动 MEMMAKER 之前退出所有正在运行的程序。另外，除正常需要的硬件驱动程序和 TSR 外，要保证在 CONFIG. SYS 和 AUTOEXEC. BAT 中不启动其他不必要的程序。

如果未用/BATCH 选项，MEMMAKER 的对话框会提示用户输入选择项，如使用快速(express)或用户选择(custom)优化。用户选择优化要求用户输入很多选择。

如果使用了磁盘压缩程序，启动盘的驱动器符在系统启动后可能会改变。必须用/SWAP 选项告诉 MEMMAKER 到哪里寻找 CONFIG. SYS 和 AUTOEXEC. BAT 文件(必要时还要寻找 Windows 的 SYSTEM. INI 文件)。对于 DoubleSpace、DriveSpace 和 Stacker 2. 0或更高版本无需此选项。

若不使用 Windows，指定/W:0,0 可以节省高端内存。

如果 MEMMAKER 运行后系统工作不正常，或用户不喜欢新的内存设置，可以带/UNDO 选项启动 MEMMAKER 回到原来的设置。

使用/UNDO 选项时，MEMMAKER 利用保存的备份恢复 CONFIG. SYS、AUTOEXEC. BAT 和 SYSTEM. INI 等文件。如果运行 MEMMAKER 之后对这些文件做过改动，则改动丢失。

应用举例

MEMMAKER

启动 MEMMAKER，显示出欢迎屏幕，并开始提示用户输入选择。

MEMMAKER /SWAP:D

启动 MEMMAKER 并指定现在的 D 盘是原来的启动盘。

© MENUCOLOR

见第 38 章

设定启动菜单的颜色。

句法

MENUCOLOR text [,background]

参数

text 指定文本颜色。

background 指定背景颜色。缺省是 0(黑色)。

注意

只能在 CONFIG.SYS 的主菜单块或多重设置的某个子菜单块中使用 MENUCOLOR 命令。

若逗号后有空格,MENUCOLOR 则忽略 background。

参数 text 和 background 的值如下:

0	黑
1	蓝
2	绿
3	青
4	红
5	品红
6	棕
7	白
8	灰
9	亮蓝
10	亮绿
11	亮青
12	亮红
13	亮品红
14	黄
15	亮白

background 使用高于 7 的值,其效果取决于不同的显示器和显示卡。例如,当 background=12 时,背景可能是闪烁的红色而不是亮红色。

应用举例

MENUCOLOR 9

在黑色背景上生成亮蓝色文本的菜单。

MENUCOLOR 10,7

在白色背景上生成亮绿色文本的菜单。

© MENUDEFAULT

见第 38 章

为 CONFIG.SYS 启动菜单指定缺省菜单项并设定超时值。

句法

MENUDEFAULT=blockname[,time]

参数

blockname　指定缺省菜单项。

time　指定 DOS 等待用户选择菜单的秒数,其值为从 0 到 90。

注意

只能在 CONFIG.SYS 的主菜单块或多重设置的某个子菜单块中使用 MENUDEFAULT。

当使用 MENUDEFAULT 时,DOS 将菜单的缺省项加亮显示并将对应于 blockname 的数字插入在输入提示后面。

若在主菜单块中使用定时器,DOS 则显示从 time 到 0 倒计时。计时到 0,DOS 处理 blockname。在子菜单块中,DOS 目前忽略 time。

当不用 MENUDEFAULT 时,菜单屏上第一项被加亮显示,1 被插入到选择提示后面,没有倒计时,并且 DOS 一直等到用户按回车键才继续执行。

把 time 设成 0 实际上取消了用户的选择权。系统启动时启动菜单直接去执行 blockname。

应用举例

MENUDEFAULT WINDOWS,10

若 10 秒钟内用户不作出选择,则执行设置语句块[WINDOWS]。

© MENUITEM

见第 38 章

在 CONFIG.SYS 的启动菜单中定义一个菜单项。

句法

MENUITEM blockname[,text]

参数

blockname　指定与本菜单项相联系的设置块名。

text　为本菜单显示的文本;缺省是块名。

注意

只能在 CONFIG.SYS 中的主菜单块或多重设置的子菜单块中使用 MENUITEM。

每个菜单块最多包含几个菜单项。DOS 用它们在菜单块中出现的顺序编号。如果对于一个或几个菜单项需要更多或更详细的选择,可以用 SUBMENU(参见 SUBMENU 命令)替换 MENUITEM。

在 CONFIG.SYS 中的其他地方必须定义一个名为 blockname 的块。若 DOS 找不到 blockname,这一项则不包含在菜单中。

blockname 中最多能有 70 个字符,可以包含除空格、反斜线(\)、正斜线(/)、逗号(,)、分号(;)、等号(=)和方括号([)以外的任何可打印字符。

text 最长 70 个字符并可包含任何可打印字符。

应用举例

```
[MENU]
MENUITEM BASIC,BASIC STARTUP
MENUITEM WINDOWS,GO DIRECTLY TO WINDOWS
MENUITEM NETWORK
```

此菜单块产生以下启动菜单。

```
1. BASIC STARTUP
2. GO DIRECTLY TO WINDOWS
```

3. NETWORK

MKDIR

见 MD。

Ⓔ MODE

见第 48、49 和 50 章

配置系统设备。

句法 1

```
MODE LPTn[COLS=cols] [LINES=lines] [RETRY=retry]
MODE LPTn [cols],[lines],[retry]
```

设置接在并行打印口上的打印机的参数。

参数

LPTn	指定并行口。n 从 1 到 3。
cols	指定每行的字符(或列)数。可以是 80 或 132。缺省值是 80。
lines	指定每英寸的行数,可以是 6 或 8。缺省值是 6。
retry	DOS 在试图向并行打印机输出时,若出现超时错所应采取的动作。见注释。

注意

不带其他任何参数的 MODE LPTn 命令的用法参见句法 6。

可以用 PRN 代替 LPT1。

如果使用本句法的第一个,较完全的格式,参数顺序则可以任意。

如使用第二种缩写的格式,则必须按顺序加带逗号使用参数。如果空出一个参数,一定要用适应的逗号指明其位置(尾随的逗号可以省略)。

retry 的值如下:

E	若端口忙则返回一个错误。
P	不断重试直到打印机接受输出。
R	若端口忙则返回"准备好"信息。
N	不重试。

如果在网络上使用 MODE 命令,则不要使用 retry 的任何值。

若忽略某个参数,MODE 使用最近一次使用的值作为缺省值。如果前面的 MODE 都没有指定其值,MODE 则将 cols 置为 80,lines 置为 6,retry 置为 N。

应用举例

```
MODE LPT1 132,8,P
```

设置 LPT1 为 132 列和每英寸 8 行。指示 MODE 在访问打印机时,若端口忙则不断重试。

```
MODE PRN,,E
```

用缺省的列、行值设置 LPT1。在访问打印机时若端口忙则返回一个错误。

句法 2

```
MODE COMn BAUD=baud [PARITY=parity] [DATA=data] [STOP=stop]
[RETRY=retry]
MODE COMn baud,[parity],[data],[stop],[retry]
```

设置串行口参数。

参数

COMn	指定一个串行口,n 可从 1 到 4。
baud	指定传输速率(波特率)(其值见注释)。
parity	指定系统使用奇偶校验的方式(其值见注释)。
data	每个字符的数据位数。其值可从 5 到 8。缺省为 7。
stop	标志字符结束的停止位数。其值可以是 1、1.5 或 2。若波特率是 100,缺省值为 2,否则缺省值为 1。
retry	DOS 在试图向端口输出时若出现超时错所应采取的动作(见注释)。

注意

如果使用本句法第一种形式——较完整的形式,那么参数顺序,任意。

如果使用本句法第二种形式——缩略的形式,必须按顺序加带逗号使用参数。若空出一个参数,一定要使用适当的逗号指明其位置,尾随的逗号可以省略(见应用举例)。

baud 的值如下:

11 或 110	110 波特
15 或 150	150 波特
30 或 300	300 波特
60 或 600	600 波特
12 或 1 200	1 200 波特
24 或 2 400	2 400 波特
48 或 4 800	4 800 波特
96 或 9 600	9 600 波特
19 或 1 920	19 200 波特

parity 的值是 N 代表无校验,E 代表偶校验,O 代表奇校验,M 代表传号脉冲校验,S 代表空号脉冲校验。E 是缺省值。

retry 的值如下:

E	若端口忙则返回一个错误。
B	若端口忙则返回"忙"。
P	不断重试直到端口接受输入。
R	若端口忙则返回"准备好"。
N	不重试(缺省值)。

如果在网络上使用 MODE,则不要使用 retry 的任何值。

某些对于 MODE 命令来说合法的值不是所有的计算机都支持。其中包括:BAUD=19、PARITY=S、PARITY=M、DATA=5、DATA=6 和 STOP=1.5。

若忽略除 baud 外的任何参数,MODE 将最近使用的值作为此端口的缺省值。如果前面的 MODE 命令都没有指定其值,MODE 使用上面列出的缺省值。baud 没有缺省值,必须指定这一参数。

应用举例

```
MODE COM1 24,0,8,2,P
```

设置 COM1 的参数为 2 400 波特,奇校验,每字符 8 个数据位,2 个停止位。若端口在被访问时忙则不断重试。

```
MODE COM2 96,,,,E
```

设 COM2 参数为 9 600 波特,使用奇偶校验、数据位和停止位采用缺省值。若端口忙则返回一个错误。

句法 3

MODE [display] [,shift[,T]]
MODE [display] [,lines]
MODE CON [COLS=cols] [LINES=lines]

选择和设置活动的显示适配器。

参数

无参数	见句法 6。
display	指定一种显示状态,包括颜色特性和每行字符数(见注释)。
shift	向左或右移动屏幕。合法值是 L 和 R。
T	提供测试图案以利于正确对齐屏幕。
lines	指定每屏幕的行数,可以是 25、43 或 50。并不是所有的显示器都支持 50。设置行数之间必须安装 ANSI.SYS。
cols	指定每屏的列数(即每行字符数)。可以是 40 或 80。

注意

不是所有的显示器都支持左右移动或测试图案。

参数 display 的有效值如下:

40 或 80	每行字符数。
BW40 或 BW80	CGA 适配器不显示彩色,每行 40 或 80 字符。
CO40 或 CO80	彩色显示器使用彩色,每行 40 或 80 字符。
MONO	单色适配器,每行 80 个字符。

命令 MODE 的本句法控制 DOS 命令的提示屏和其他一些屏幕的显示。但很多应用程序,如 DOS Shell,使用自己的设置。

应用举例

MODE 40,43

设置显示器为每行 40 字符,每屏 43 行。

MODE BW80,25

设置显示器为黑白显示方式,每行 80 字符,每屏 25 行。

MODE,43

颜色和每行字符数不变,设置显示器为每屏 43 行。

句法 4

MODE CON RATE=rate DELAY=delay

设置键盘重复速率。

参数

rate	按住一个键不放时该字符的重复速率,可从 1 到 32。大约分配对应于每秒 2 到 30 个字符。对于 IBM AT 兼容键盘,缺省值是 20。对于 IBM PS/2 兼容键盘,缺省值是 21。
delay	指定按住一个键多长时间后开始重复。其值可以是 1、2、3、4,分别代表 0.25、0.50、0.75、1 秒。缺省值是 2。

注意

使用本句法必须同时指定 rate 和 delay 参数。

应用举例

MODE CON RATE=32 DELAY=3

按住一个键 0.75 秒之后，按大约每秒 30 次的速率开始重复，直到放开这个键。

句法 5

MODE LPTn[=COMm]

将送到并行口的输出重定向到串行口。

参数

LPTn　　指定一个并行口。n 可从 1 到 3。

COMm　　指定一个串行口。m 可从 1 到 4。

注意

重定向 LPTn 后，任何送到 LPTn 的输出转到 COMm。可能有一个串行打印机接在 COMm 上。

要结束重定向，不带=COMm 使用 MODE LPTn 命令。

如果 n=1，COM 端口成为系统的缺省打印机端口——PRN。

使用本命令前一定要用句法 2 设置 COMm 参数。

应用举例

```
MODE COM1 96,N,8,1
MODE LPT1=COM1
```

第一个命令为 COM1 建立通讯参数。然后第二个命令把送到 LPT1 的所有输出重定向到 COM1。

```
MODE LPT1
```

取消对 LPT1 的任何重定向。此后，所有定向到 LPT1 的输出被送到 LPT1。

句法 6

MODE [device] [/STATUS]

显示安装在系统上的一个或所有设备的状态，或者结束重定向。

参数和选项

无参数　　显示安装在系统上的所有设备的状态。

device　　指定要显示其状态的设备。设备可以是 CON、LPTn 或 COMm。

/STATUS　　要求设备的状态，其缩写形成为/STA。

注意

不带选项/STA 时，如果 device 是已被重定向的并行打印机(见句法 5)，重定向则被取消。若有/STA 选项，显示重定向打印机的状态并保留重定向。

如果 device 不是重定向的并行打印机，无论带或不带/STA 选项都显示设备状态。

若省略 device，显示所有设备的状态并且不影响任何重定向。

应用举例

```
MODE CON
```

显示控制台设备的状态。

```
MODE LPT1
```

如果 LPT1 已被重定向到串行口，则结束重定向。

```
MODE LPT1 /STA
```

显示 LPT1 的状态而不取消它的重定向。

句法 7

```
MODE device CODEPAGE PREPARE=((page [...]) filespec)
MODE device CODEPAGE SELECT=page
MODE device CODEPAGE REFRESH
```

MODE device CODEPAGE [/STATUS]

管理代码页。

参数和选项

device	需要应用本命令的设备。
CODEPAGE	说明本命令为代码页命令，其缩写形式为 CP。
PREPARE page	为设备准备代码页，其缩写形式为 PREP。指定一个代码页，可以是 437(美国)、850(多种语言或拉丁 I)、852(南斯拉夫或拉丁 II)、860(葡萄牙)、863(法语加拿大)或 865(北欧)。
filespec	device 的代码页信息文件的名字和位置(见注释)。
SELECT	装入代码页。该页必须已由 MODE CP PREP 命令准备好。该选项的缩写形式为 SEL。
REFRESH	如果因为硬件或其他错误丢失已准备好的代码页，就将其恢复。该参数的缩写形式为 REF。
/STATUS	显示当前为 device 准备好或选中的代码页的数字。该选项的缩写形式为/STA。

注意

不带 PREPARE、SELECT、REFRESH 或/STATUS 使用本句法与使用/STATUS 选项效果相同。

在 MODE 命令能够准备或选择代码页之前，必须装入设备驱动程序 DISPLAY.SYS(对于控制台设备而言)或 PRINTER.SYS(对于打印机而言)来保留缓冲区。为设备准备的代码页数不能比装入设备驱动程序时所指定的多。

使用 COUNTRY、NLSFUNC 和 CHCP 的组合可以立即为控制设备和打印机选择同一个代码页。

应用举例

MODE CON PREP=((850 437) C:\DOS\EGA.CPI)

为控制台设备准备代码页 850 和 437，使用在 C:\DOS\EGA.CPI 中找到的信息。

MODE CP

显示当前代码页状态。

Ⓔ MORE

见第 36 章

显示文件或程序输出，每显示一屏后暂停。

句法

MORE <filespec
command | MORE

参数

filespec	指定一个 ASCII 文本文件，每次一屏显示其内容。
command	指定一个命令(包括必要的选项和参数)，每次一屏显示其输出。

注意

使用管道(|)之前，应设好 TEMP 环境变量(见 SET)。

可以把一些命令，如 SORT 和 TYPE，或其他以 ASCII 文本作为标准输出的程序，用管道把它们的输出传给 MORE，但不能是启动批处理程序的命令。

应用程序

MORE <TESTA.BAT

显示文件 TESTA.BAT 的内容，满屏后暂停。

TREE | MORE

显示当前驱动器的目录结构，满屏后暂停。

TYPE TESTA.BAT|SORT|MORE

排序并显示文件 TESTA.BAT 的内容，满屏后暂停。

Ⓔ MOVE

见第 15 章

移动一个或多个文件到其他位置。改变文件或目录名。

句法 1

MOVE [/Y|/－Y] filespec[...] destination

移动文件并改变文件名。

参数

filespec	需要移动或换名的文件。
destination	将文件移到该文件或目录，或者是该文件的新名。
/Y	不加警告覆盖目的文件。
/－Y	覆盖目的文件前要求用户确认。

注意

命令 MOVE 假定命令行中最后一个参数是 destination。任何其他参数都被认为是 filespec。filespec 中可以包含通配符。

如果移动多个文件，则 destination 必须是目录而不能是文件名。destination 中不能有通配符。

如果只移动一个文件，destination 可以是(或包括)一个新文件名。在这种情况下，文件被移动并换名。在 DOS 6.0 中，如果由 destination 所指定的文件已存在，MOVE 则不带警告将其覆盖。

在 DOS 6.2 或更高版本中，如果由 destination 所指定的文件已存在，则 MOVE 的操作与几个因素有关。在缺省情况下，若命令是在命令提示符下输入的，MOVE 会询问用户是否覆盖目的文件，但如果命令是从批处理程序中输入的，MOVE 则不加警告覆盖目的文件。假如环境变量 COPYCMD 的值是/Y，MOVE 不带警告覆盖文件。如果环境变量 COPYCMD 的值是/－Y，MOVE 则在覆盖文件前询问用户。可以在命令中使用/Y 或/－Y 选项而阻止缺省操作或环境变量 COPYCMD 对本次命令执行的影响。

选项/Y 和/－Y 不能用于 DOS 6.2 之前的版本。

当 DOS 询问是否覆盖已存在的目的文件时，按 Y 键覆盖文件，按 N 键保留文件(不进行移动)，或按 A 键覆盖后续所有被本次 MOVE 命令影响的文件。换句话说，A 键取消对本次 MOVE 命令中其余文件的保护。

命令 MOVE 不处理具有系统和隐含属性的文件。可以移动只读文件，但不能覆盖 destination 中的只读文件。源文件的只读和归档属性被复制到目的文件。

当在同一个驱动器上移动文件时，MOVE 只移动目录项，这比拷贝并删除文件要快得多。当文件移至不同的驱动器时，MOVE 当然只能拷贝文件并删除原来的文件。

如果因为 destination 是只读文件或者因为目的驱动器已满而使 MOVE 命令未能成功执行，MOVE 则不删除原来文件。移动不成功时 MOVE 会通知用户。

应用举例

MOVE AFILE.* *.BAT C:\ACCOUNT

从当前目录中移动所有名为 AFILE 而扩展名任意以及所有扩展名为.BAT 的文件到 C 盘的 ACCOUNT 目录下。不移动具有系统或隐含属性的文件。

MOVE AFILE.NEW AFILE.OLD

在当前目录中，把文件 AFILE.NEW 改名为 AFILE.OLD。

句法 2

MOVE directory newname

更改目录名。

参数

directory	要改名的目录。
newname	指定新名字。

注意

不能改变当前目录的名字。

参数 newname 必须为新目录提供一个唯一的名字。不能与其父目录中的某一文件或目录重名。

参数 newname 不能为目录指定不同的路径。

应用举例

MOVE C:\T1 C:\ACCOUNT

若驱动器 C 上的 T1 是目录名，将其改名为 ACCOUNT(如果 T1 是文件名，见句法 1)。

MOVE T1 TESTING

如果 T1 是当前目录的子目录，将其改名为 TESTING(若 T1 是文件名，见句法 1)。

---

① MSAV

见第 25 章

在内存和磁盘上搜索病毒，发现时将其去除。

句法

MSAV [drive... |path|/A|/L] [/S|/C] [/R] [/N] [/P] [/F] [/VIDEO] [/videomouse...]

参数和选项

无参数	打开 MSAV 对话框使用户可以选择驱动器和选项。
drive	指定要搜索的驱动器。
path	指定目录分枝顶端。
/A	搜索除 A、B 外的所有驱动器。
/L	搜索除 A、B 和网络驱动器以外的所有驱动器。
/S	搜索但不清除病毒。
/C	搜索并清除发现的任何病毒。
/R	生成报告文件(MSAV.RPT)，其中列出检测的文件数、发现的病毒数、清除的病毒数和怀疑含有病毒的文件数。
/N	关闭搜索时的信息显示。
/P	以非图形的，即命令行方式运行 MSAV。
/F	不显示被搜索的文件名。只能与/N 或/P 选项一同使用。
/VIDEO	显示视频和鼠标选项的列表。本选项不进行病毒检测或装入对话框。要检测病毒必须在看到列表后再次输入 MSAV 命令。
/videomouse	使用 videomouse 选项(其值见注释)。

注意

MSAV 可以认出 1 000 多种已知病毒并可以检测未知病毒。它在每个被检测的目录中维护一个 CHK-

LIST.MS 文件以便根据目录中文件的改变确认未知病毒。它不能清除未知病毒，而且有些未知病毒也检测不到。

启动 MSAV 后的首次检测还会搜索内存以查找病毒。

从 MSAV 对话框中可以选择命令行中没有的选项，如检测"秘密"(stealth)病毒或关掉警告铃。另外，从对话框中可以看到已知病毒的列表及其中相关的信息。

若指定了驱动器但没有/N 或/P 选项，则显示对话框，用户可根据需要进行选择，甚至可以选择搜索不同的驱动器。搜索完一个驱动器后，用户还有机会选择驱动器和选项再搜索一次。

如果指定 path 但没有/N 或/P 选项，那么出现对话框后立即开始搜索病毒。搜索结束后，对话框关闭。用户没有机会选择其他选项或者继续搜索其他路径或驱动器。

参数 videomouse 的值如下：

25	设置屏幕为 25 行。此为缺省值。
28	设置屏幕为 28 行。本选项只用于 VGA 显示适配器。
43	设置屏幕为 43 行。本选项用于 EGA 或 VGA 显示适配器。
50	设置屏幕为 50 行。本选项只用于 VGA 显示适配器。
60	设置屏幕为 60 行。本选项只用于 Video 7 显示适配器。
IN	即使未检测到彩色显示适配器也使用彩色方案。
BW	使用黑白方式。
MONO	使用单色方式。
LCD	使用 LCD 颜色方式。
FF	对于 CGA 显示适配器使用最快速的屏幕刷新。
BF	使用计算机的 BIOS 进行显示。若显示的效果不好可使用此选项。
NF	禁止使用替换字库。
BT	允许使用 Windows 中的图形鼠标。
NGM	使用缺省鼠标字符而不是图形鼠标。若屏幕上出现雪花可以试用此选项。
LE	交换鼠标左右键的功能。
IM	禁止鼠标。
PS2	若光标消失或锁住则重置鼠标。

选项/VIDEO 覆盖其他任何参数和选项并且只显示视频/鼠标选项列表。

每次使用 MSAV 都会显示一个简要报告，而不论是否使用选项/R 生成 MSAV.RPT。文件 MSAV.RPT 的内容更详细，因为它包含被怀疑含有未知病毒的文件名，简要报告中没有这类信息。有时简要报告在屏幕上显示太快而无法阅读，如果遇到这样的问题，使用/R 选项得到可读的报告。

文件 MSAV.RPT 保存在被搜索驱动器的根目录下。若搜索了多个驱动器，为每个驱动器产生一个 MSAV.RPT 文件。

获得当前最新防毒软件的做法参见 DOS 的《用户指南》。

若使用选项/N 运行，MSAV 在发现病毒的情况下设置返回码为 86，否则返回码为 0。

应用举例

```
MSAV /A
```

启动 MSAV，搜索内存和除 A、B 外的所有驱动器。

```
MSAV C: D: /BW /43
```

以每屏 43 行的黑白显示方式启动 MSAV，搜索内存和驱动器 C、D。

```
MSAV C:\DOS /C /R
```

搜索内存和以 C:\DOS 开始的目录分枝中的文件，清除所发现的病毒。执行报告存在 C:\MSAV.RPT

中。

Ⓔ MSBACKUP

见第 23 章和 24 章

备份和恢复计算机上的数据。

句法

MSBACKUP [setup] [/BW|/LCD|/MDA]

参数和选项

无参数	打开 MSBACKUP 主对话框并装入 DEFAULT.SET。
setup	指定在打开 MSBACKUP 对话框时装入的设置文件。文件扩展名必须是 .SET。缺省文件是 DEFAULT.SET。
/BW	在 MSBACKUP 对话框中使用黑白显示方式。
/LCD	使用与膝上型计算机兼容的显示方式。
/MDA	使用单色显示适配器。

注意

使用 MSBACKUP 的对话框可以生成一个设置文件。

即使指定了启动设置,仍可在开始备份之前通过 MSBACKUP 对话框改变设置和选择文件。打开对话框后也可以完成系统设置、比较和恢复功能。

MSBACKUP 不能恢复由 6.0 版本以前的 DOS 所备份的文件(见 RESTORE 命令)。

在 DOS 6.22 或更高版本中,MSBACKUP 使用与 DOS 6.0 和 6.2 不同的方法压缩文件。因而,DOS 6.22 不能恢复由 DOS 6.0 或 6.2 压缩备份的文件。同样,DOS 6.0 或 6.2 的 MSBACKUP 也不能读取由 DOS 6.22 压缩备份的文件。从以前的 DOS 版本中恢复文件的技术见第 24 章。

应用举例

MSBACKUP

打开 MSBACKUP 对话框并装入 DEFAULT.SET。

MSBACKUP DAILY

打开 MSBACKUP 对话框并装入 DAILY.SET。

Ⓔ MSCDEX

使 DOS 可以使用 CD-ROM 驱动器。

句法

MSCDEX /D:signature [...] [/E] [/K] [/S] [/V] [/L:drivename][/M:buffers]

参数和选项

/D:signature	指定 CD-ROM 设备驱动程序的标记。
/E	让设备驱动程序使用扩充内存(expanded memory)。
/K	承认用日文汉字编码的 CD-ROM 盘。
/S	允许 MS-NET 和 Windows for Workgroups 服务器共享 CD-ROM 驱动器。
/V	启动 MSCDEX 时显示内存状况报告。
/L:drivename	指定赋予 CD-ROM 驱动器的驱动器名。如果命令中包含多个驱动器,为第一个指定名字,其他驱动器使用其后连续的名字。
/M:buffers	指定为驱动器分配的缓冲区数。

注意

在用命令 MSCDEX 激活驱动器前，必须在 CONFIG.SYS 中为每个 CD-ROM 驱动器装入一个设备驱动程序。

在 CONFIG.SYS 中，装入 CD-ROM 设备驱动程序的命令必须包含/D:signature 参数以标记 CD-ROM 设备。在本命令中使用同样的/D 参数将 MSCDEX 命令和设备驱动程序联系起来。

如果有多个 CD-ROM 设备，则每个设备必须有自己唯一的标记。

如果使用 Windows，那么在启动 Windows 前输入 MSCDEX 命令，多数用户把 MSCDEX 命令包含在 AUTOEXEC.BAT 中。

若不指定驱动器名，MSCDEX 把下一个可用驱动器名指派给 CD-ROM 设备。可能需要用 LASTDRIVE 命令增加可用的驱动器名(见 LASTDRIVE)。

DOS 6.2 或以后的 SmartDrive 可以对 CD-ROM 驱动器进行缓冲而明显提高其速度。如果用户有 DOS 6.2 或更高版本且正在使用 SmartDrive，一定要在 AUTOEXEC.BAT 文件中把 MSCDEX 命令放在 SMARTDRV 之前，这样 SmartDrive 就可以识别出 CD-ROM 驱动器并为其设置缓冲区(见 SMARTDRV 命令)。

应用举例

```
DEVICEHIGH=C:\CDROM\MTMCDS.SYS /D:MSCD0000(在 CONFIG.SYS 中)
MSCDEX /D:MSCD0000 (在 AUTOEXEC.BAT 中)
```

第一个命令装入 MTMCDS.SYS 设备驱动程序并为 CD-ROM 驱动器指派标记 MSCD0000(本命令取决于随 CD-ROM 驱动器附带的设备驱动程序，用户所使用的设备驱动程序可能有很大不同)。第二个命令为 MSCD0000 设备装入 MSCDEX。

```
DEVICEHIGH=C:\CDROM\MTMCDS.SYS /D:MSCD0000(在 CONFIG.SYS 中)
DEVICEHIGH=C:\CDROM\MTMCDS.SYS /D:MSCD0001(在 CONFIG.SYS 中)
MSCDEX /D:MSCD0000 /D:MSCD0001 /V /K(在 AUTOEXEC.BAT 中)
```

前两个命令为两个 CD-ROM 驱动器装入设备驱动程序，以 MSCD0000 和 MSCD0001 为标记。第三个命令为两个设备装入 MSCDEX，显示内存状况报告，并指定用日文汉字编码的 CD-ROM 卷被承认。

Ⓔ MSD

显示或报告关于计算机的技术信息。

句法

```
MSD [/F filespec|/P filespec|/S [filespec]|[/I][/B]]
```

参数和选项

filespec	指定一个文件用于保存 MSD 的报告。
/F	提示输入身份信息，然后向 filespec 写入一个完整的 MSD 报告。不与其他选项同时使用。
/P	不需要任何提示，向 filespec 写一个完整的 MSD 报告。不与其他选项同时使用。
/S	写出简要的 MSD 报告。若存在 filespec 参数，将报告写到 filespec 中。若不存在 filespec 参数，将报告输出到显示器上。不与其他任何选项一同使用。
/I	指定 MSD 在启动时不检测硬件。如果启动 MSD 出现问题或不能正确运行，可使用此选项。除/B 外不与其他任何选项一同使用。
/B	用黑白方式运行 MSD。除/I 外不与其任何选项一同使用。

注意

命令 MSD 提供关于计算机类型、处理器、内存、显示类型、DOS 版本、鼠标、其他适配器、磁盘驱动器、LPT 端口、COM 端口、IRQ 状态、TSR 和设备驱动程序的信息。

若忽略选项/F、/P 和/S，MSD 则打开对话框，用户可在其中选择想要看的信息。

应用举例

MSD

打开 MSD 的对话框使用户能够读出关于计算机的数据。

MSD /F MYSYSTEM. TXT

提示用户输入姓名、单位、地址和其他身份信息。在文件 MYSYSTEM. TXT 中生成报告，其中包括身份数据和关于计算机的技术信息。

Ⓔ NLSFUNC

见第 49 章

启动程序，装入国家相关的信息以使支持某种语言(NLS)。

句法

NLSFUNC [filespec]

参数

无参数　　使用国家相关信息的缺省文件(见注释)。

filespec　　指定一个包含国家相关信息的文件。

注意

若 CONFIG. SYS 中包含 COUNTRY 命令，则它已定义了缺省文件。否则，缺省文件是启动盘根目录下的 COUNTRY. SYS。

即使国家相关信息文件不存在也不会出现错误信息，因为 NLSFUNC 实际上并不访问该文件。在以后当命令 CHCP 试图使用文件中的信息时才会出错。

可以在 CONFIG. SYS 中装入 NLSFUNC(见命令 INSTALL)。

应用举例

NLSFUNC

启动支持语种的 TSR，使用国家相关信息的缺省文件。

NLSFUNC C:\OLDDOS\COUNTRY. SYS

启动支持语种的 TSR，使用 C:\OLDDOS 下的 COUNTRY. SYS 作为国家相关信息文件。

Ⓒ NUMLOCK

指定数字键盘的数字锁(Numlock)的初始状态是打开或关闭。

句法

NUMLOCK=ON|OFF

参数

ON　　打开 NumLock。

OFF　　关闭 NumLock。

注意

只能在 CONFIG. SYS 的菜单块中使用 NUMLOCK。

当 NumLock 打开时，数字副键盘产生数字。(在增强型键盘中这可能也会影响同名键。如果出现这样的问题，试在 CONFIG. SYS 中装入带/X 选项的 ANSI. SYS)。当 NumLock 关闭时，数字副键盘产生另一组功能(移动光标、插入和删除)。

若没有 NUMLOCK 命令，大多数计算机启动时关闭 NumLock。无论哪种情况，按 NumLock 键都可以在 NumLock 的开、关状态间来回切换。

应用举例

NUMLOCK=ON

启动时打开 NumLock，这样敲数字键盘就可以产生数字。

Ⓘ PATH

见第 12 章

为外部可执行文件(包括批文件)定义查找路径。

句法

PATH [path[;path]...]
PATH;

参数

无参数	显示当前查找路径。
path	指定一个目录，在其中可以查找程序文件。
;	作为唯一的参数时，清除查找路径。

注意

当命令名中包含路径时(如 C:\DOS\FORMAT A:)，DOS 忽略查找路径而只在指定的目录中寻找程序。

当命令名中未包含路径时，DOS 首先在当前目录中寻找程序，然后是查找路径(若存在)中的目录。搜寻目录按照查找路径中出现的顺序进行。DOS 在发现指定程序后就停止搜寻，即使在查找路径中的另一个目录下存在另一个同名程序。

如果在当前目录或查找路径的各目录中都未找到程序，DOS 显示错误信息"Bad command or file name"(即"错误的命令或文件名")。

DOS 不查找内部命令。内部命令作为 COMMAND.COM 的一部分或 DOSKEY 的宏而存储在内存中。

要保证每个 path 指定完整的、绝对路径名，包括驱动器名。

如果 path 设置中有错，命令 PATH 并不显示错误信息。DOS 在查找时如果遇到非法路径名则会显示信息"Invalid directory"(即"无效目录")。

由 PATH 命令生成的查找路径被赋值给名为 PATH 的环境变量。

每次用 PATH 命令定义查找路径时，都会完全代替以前的查找路径。如果想把另一个目录加到当前查找路径中，一定要在命令中完整的包括当前查找路径。

命令 PATH 的长度不能超过 127 个字符，这就把查找路径本身限制为 122 个字符。

应用举例

PATH C:\;C:\DOS;D:\BOOK\APPS

设置查找路径使 DOS 查找程序时首先找当前目录，其次是 C 盘根目录，然后是 C:\DOS 目录，最后是 D:\BOOK\APPS。

PATH;

清除查找路径。此后 DOS 只在当前目录中查找程序。

PATH &PATH%;C:\WORKOUT

把名为 C:\WORKOUT 的目录加到当前查找路径的结尾处。注意此命令使用了环境变量 PATH，因而只能用于批处理程序中。

Ⓑ PAUSE

见第 20 章

暂时挂起批处理程序的执行，提示用户按任意键继续。

句法

PAUSE

注意

用户响应 PAUSE 命令时按下的键如果不是中断键(Ctrl+C 或 Ctrl+Break)，则使程序继续执行。若要使用户可作有含义的输入，参见命令 CHOICE。

应用举例

```
ECHO Load Form 301/B in the printer
PAUSE
```

在命令 ECHO 显示信息后，程序暂停直到用户按下某个键，以给用户足够的时间装入表格纸。

Ⓔ POWER

见第 48 章

在膝上型计算机中，当应用程序和设备空闲时减少电源消耗。

句法

POWER [ADV:option |STD|OFF]

参数和选项

无参数	显示当前电源设置。
ADV:option	当应用程序和硬件驱动程序空闲时减少电源供应。option 可以是 MAX、REG 或 MIN(见注释)。
STD	只使用计算机硬件的电源管理功能(如果存在)。
OFF	关掉电源管理功能。

注意

使用 POWER 命令之前必须在 CONFIG.SYS 中安装 POWER.EXE 设备驱动程序。

电源管理功能与 APM(先进电源管理)规范兼容。

如果用户的计算机不支持 APM 规范，STD 会关掉电源管理功能。

使用 ADV:MAX 最大限度地节省电源。使用 ADV:REG 同时照顾到节省电源及应用程序和设备的性能。如果用 MAX 或 REG 后性能不令人满意，试用 ADV:MIN。

应用举例

```
POWER ADV:MIN
```

设置电源节省为最小值以提高应用程序性能。

```
POWER OFF
```

关闭节省电源功能。

Ⓔ PRINT

见第 16 章

管理后台打印。

句法

```
PRINT [/D:device] [/B:buffer] [/U:ticks1] [/M:ticks2] [/S:ticks3][/Q:qsize][/T]
[filespec...] [/C] [/P]
```

参数和选项

无参数	显示打印队列的内容或装入 TSR。

/D:device	打印输出送到该打印机端口。合法值是 LPT1、LPT2、LPT3、COM1、COM2、COM3 和 COM4。LPT1 与 PRN 相同。如果同时使用/D 和 filespec，/D 应在 filespec 之前。
/B:buffer	以字节为单位设置缓冲区大小。缓冲区用来打印前存储数据。其值可从 512 到 16384。512 是缺省值。
/U:ticks1	指定在打印机可用前等待的最大时钟信号数。可从 1 到 255。缺省值是 1。如果在 ticks1 个时钟信号中仍不能使用打印机，则任务不被打印。（每秒大约有 18 个时钟信号。）
/M:ticks2	指定在打印机上打印一个字符的最多时钟信号数。可从 1 到 55。缺省值是 2。如果一个字符的打印时间超过 ticks2 个时钟信号，DOS 显示出错信息。
/S:ticks3	指定 DOS 为后台打印分配的、在返回前台工作前所使用时间的时钟信号数。可从 1 到 255。缺省值是 8。
/Q:qsize	指定打印队列中最多的文件数。可以 4 到 32。缺省值是 10。
/T	从打印队列中去掉所有文件。
filespec	指定一个要放在打印队列中的文本文件。参数最长为 64 个字符。
/C	从打印队列中去掉指定文件。
/P	将指定文件加到打印队列中。

注意

启动计算机后第一次使用 PRINT 命令时，DOS 装入 PRINT 程序的内存常驻部分以保存打印参数。选项/D、/B、/U、/M、/S 和/Q 只能用于第一次使用 PRINT 命令时。要改变这些选项中的任何一个，只能重新启动并使用新的 PRINT 命令。

如果装入内存驻留程序时未指定选项/D，PRINT 会提示用户输入打印机端口名。

实际的打印发生在后台，就是说，DOS 在继续处理其他应用程序时利用小块时间（由选项/S 定义）打印队列中的内容。

增加缓冲区会减少用于其他工作的内存，但会提高打印速度。增加任何一个时钟信号值会加快打印但可能会降低其他工作的速度。

选项/P 作用于其前的 filespec（若存在）和所有后续的 filespec，直到遇见一个由/C 选项控制的 filespec（若存在）。选项/C 作用于其前的 filespec（若存在）和所有后续的 filespec，直到遇见一个由/P 选项控制的 filespec（若存在）。

如果既没有/C 也没有/P 选项作用于 filespec，就假定为有/P 选项。

选项/T 使队列变成空。如果选项/T 前有 filespec，它们被加到队列中，然后又在处理/T 选项时被清除。如果有 filespec 跟在/T 选项之后，那么它们被加到队列中。

应用举例

```
PRINT /D:LPT2 /B:2048
```

如果这是第一个 PRINT 命令，则装入 PRINT TSR，并带有 2048 个字节的缓冲区。所有后续的 PRINT 命令将文件送到 LPT2 队列。

```
PRINT LETTER1.TXT
```

把文件 LETTER1.TXT 加到打印队列中。如果尚未装入 PRINT TSR，以缺省值装入 PRINT，但提示用户输入打印机端口名。

```
PRINT /C LETTER1.TXT LETTER2.TXT LETTER3.TXT /P
```

从打印队列中去掉 LETTER1.TXT 和 LETTER2.TXT，并加入 LETTER3.TXT。

```
PRINT /T
```

从打印队列中去掉所有文件。

PRINT /T REP1.TXT REP2.TXT

从打印队列中去掉所有文件，然后将 REP1.TXT 和 REP2.TXT 放在打印队列中。

① PROMPT

见第 12 章

改变命令提示符。

句法

PROMPT [text]

参数

无参数	重置提示符为当前驱动器符后跟一个>。
text	指定作为提示符显示的文本和信息。

注意

可在提示符下使用的特别代码如下：

$Q	=(等号)
$$	$(美元符号)
$T	当前时间
$D	当前日期
$P	当前驱动器和路径
$V	DOS 版本号
$N	当前驱动器
$G	>(大于号)
$L	<(小于号)
$B	|(管道)
$_	回车(换行)
$E	ASCII 换码符(值为 27)
$H	退格(删除已写在提示符中的一个字符)

缺省的提示符相当于 PG。

$_使用户可以定义多行提示符。例如，$T$_$P$G 在一行中显示时间，然后到下一行开头显示当前驱动器和路径后跟一个>。

ASCII 换码符 $E 使用户可以把 ANSI.SYS 换码序列作为提示符的一部分。例如，$E[2J 清屏并将光标放在屏幕左上角(没有提示文字)。必须在 CONFIG.SYS 中装入 ANSI.SYS 后才能处理 ANSI.SYS 换码序列。否则，换码序列与其他文本同样对待。有几百个 ANSI.SYS 换码序列控制如屏幕背景和文本的颜色、按下 Ctrl+Shift+X 时要做的事情等等的操作。它们被编入 DOS 6 的屏上帮助系统，有关信息可查找 ANSI.SYS 相关的主题。

可以用 $H，即退格字符，删除 $T 或 $D 中不需要的字符。例如，$THH$H，按 hh:mm:ss 格式而不是通常的 hh:mm:ss.nn 显示时间。

应用举例

PROMPT USING VG

定义提示符，显示 USING 后面跟随 DOS 版本号和>。

PROMPT T_TIME$Q$$

定义一个两行的提示符。第一行显示时间,第二行显示 TIME= $ 。(光标跟在 $ 后面。)

```
PROMPT
```

将提示符缩成简单形式:当前驱动器后面跟一个>。

```
PROMPT $P$G
```

设置提示符的缺省值,当前驱动器和目录后跟>。

Ⓔ QBASIC

见第 53 章

启动 QBASIC 编程环境,允许编写、测试和运行 QBASIC 程序。

句法

```
QBASIC [/B] [/EDITOR] [/G] [/H] [/MBF] [/NOHI] [[/RUN] filespec]
```

参数和选项

无参数	装入 QBASIC 环境。
/B	用黑白方式显示 QBASIC。
/EDITOR	启动 DOS 的文本编辑器而不是 QBASIC 程序编辑器。
/G	对于 CGA 显示器提供最快速的屏幕更新。
/H	显示屏幕上可显示的最大行数。
/MBF	将内部函数 MKS $、MKD $、CVS 和 CVD 分别转换成 MKSMBF $、MKDMBF $、CVSMBF 和 CVDMBF。本选项提供了与某些早期 BASIC 版本的数据文件的兼容性。
/NOHI	允许使用不支持高亮度的显示器。不要在 Compaq 膝上型计算机上使用本选项。
/RUN	显示 filespec 前先运行。不能没有 filespec 而单独使用该选项。
filespec	指定一个装入的 QBASIC 程序。若没有/RUN,则显示程序。有/RUN 选项时,则在显示之前运行程序。

注意

使用/EDITOR 选项的效果与用 EDIT 命令启动 DOS 文本编辑器一样。

运行 QBASIC 时,如果显示器底部的行不显示快捷键,则使用选项/B(针对 CGA 显示器)或/NOHI(针对不支持粗体或高亮度字符的系统)。

应用举例

```
QBASIC TESTPROG
```

启动 QBASIC 环境,装入并显示文件 TESTPROG.BAS。

```
QBASIC /B
```

用黑白方式启动 QBASIC 环境,不装入文件。

```
QBASIC /RUN TESTPROG
```

启动 QBASIC 环境,装入并运行 TESTPROG.BAS。程序结束后显示回到 QBASIC 编辑器中。

Ⓡ RD

见第 14 章

删除一个目录。RMDIR 与 RD 可互换使用。

句法

```
RD path
```

参数

path	要删除的目录。

注意

在用 RD 命令删除一个目录之前，必须删除目录中的所有文件和子目录。除. 和.. 项外，该目录必须为空。

如果 DIR 显示目录是空的但仍无法将其删除，用 DIR/A(或 DOS Shell)查看是否存在隐含的文件和子目录。

命令 RD 不能删除当前目录，必须将其他目录作为当前目录，然后再输入 RD 命令。

不能删除根目录。

要想用一个命令删除一个目录、连同其中的文件和子目录，以及子目录中的文件和子目录，应该使用 DELTREE 命令。

应用举例

```
RD \DDIR
```

删除目录 DDIR，它是当前驱动器根目录下的一个子目录。

```
RD DDIR
```

删除 DDIR，它是当前目录的一个子目录。

Ⓒ REM

Ⓑ 见第 20 章

在批处理文件或 CONFIG.SYS 中加入注释。

句法

```
REM [comment]
```

参数

comment　　要在批处理文件或 CONFIG.SYS 中加入不影响其运行的文本。

注意

可以在批处理文件或 CONFIG.SYS 中使用 REM 命令提供文档或注释。有时在暂时不用的命令前加上 REM 是很方便的。这样就可以通过去掉行中 REM 再次恢复该行命令。

应用举例

```
REM Next run the daily backup.
```

表明下面几行命令目的的一条注释。

```
REM DEL TEMP. *
```

使 DEL 命令失去作用。批处理程序运行时此命令不会删除 TEMP 文件。

Ⓘ REN

见第 16 章

改变一个或多个文件名。可与 RENAME 互换使用。

句法

```
REN filespec newname
```

参数

filespec　　要改名的文件。

newname　　为文件指定新名。

注意

参数 filespec 中可以包含驱动器或路径名，但 newname 中不能。

如果被改名的文件所在的目录中已存在名为 newname 的文件或子目录，则 REN 无法执行。（关于能替换已存在的 newname 文件的命令，参见 MOVE。）

如果在两个名字中都使用了通配符，原来文件名中的通配符替换 newname 中的通配符。例如，在 REN A*.* B*.* 中，AFILE.DAT 成为 BFILE.DAT，ACCOUNT.EXE 成为 BCCOUNT.EXE，等等。

参数 filespec 不能是目录。给目录改名的命令参见 MOVE。

应用举例

```
REN JET.NEW JET.SAV
```

把当前目录中的 JET.NEW 改名为 JET.SAV。

```
REN D:\YEARDATA\NEWDATE.* OLDDATA.*
```

把 D:\YEARDATA 中每个名为 NEWDATA（扩展名任意）的文件改名为 OLDDATA（扩展名与以前相同）。

RENAME

见 REN。

Ⓔ REPLACE

见第 40 章

从一个目录向另一个目录中替换或加入文件。

句法 1

```
REPLACE filespec path[/P] [/R] [/W] [/S] [/U]
```

用源目录中的文件替换目的目录中相匹配的文件。

参数和选项

filespec	源文件。
path	目的目录。
/P	替换文件前提示用户确认。
/R	替换目的目录中的只读和非只读文件。
/W	开始查找源文件前等待用户按任意键。
/S	在以 path 开始的整个目录分枝中查找要被替换的文件。
/U	检查源文件和目的文件的时间标记，只有当目的文件比要替换它的源文件旧时才进行替换（更新）。

注意

命令 REPLACE 忽略源目录和目的目录中的隐含文件和系统文件。

不带/R 选项时，试图替换只读文件会产生错误信息并中止 REPLACE 命令。

如果在 filespec 中使用了通配符，则每个匹配的文件替换目的目录或其分枝中的同名文件。

加上/U 选项可以保证不会覆盖最新的版本。例如，若想保证 JOHN 和 MARY 都有 TILE.DAT 的最新版本，使用命令 REPLACE JOHN \TILE.DAT MARY/U，再用 REPLACE MARY\TILE.DAT JOHN /U。

REPLACE 设置的返回码如下：

0	执行成功
1	DOS 版本与 REPLACE 不兼容
2	未找到源文件

3	源或目的路径未找到
5	无法访问目的文件(只读)
8	内存不够
11	语法或命令行有错

应用举例

以下例子中,假定源文件和目的文件都不是系统文件和隐含文件。

```
REPLACE NEWDATA.921 \DIR92
```

如果文件\DIR92\NEWDATA.921 存在且非只读,则用当前目录中的 NEWDATA.921 代替之。

```
REPLACE DONNA\NEWDATA.* JUDI /R
```

如果两个目录中都有文件 NEWDATA.BAT,将其从 DONNA 中拷贝到 JUDI;如果两个目录中都有文件 NEWDATA.COM,将其从 DONNA 拷贝到 JUDI;如果两个目录都有 NEWDATA.EXE,将其从 DONNA 拷贝到 JUDI;等等。即使 JUDI 中的文件有只读属性也进行替换。

```
REPLACE A:\BUDGET.LST C:\ /S /R /W
```

开始前等待用户按任意键,然后在 C:\的每个目录中寻找名为 BUDGET.LST 的文件。找到的每个文件(即使是只读的)都用 A 盘根目录下的 BUDGET.LST 替换。

句法 2

```
REPLACE filespec path /A [/P] [/W]
```

从源目录中拷贝目的目录中不存在的文件。

参数和选项

filespec	源文件。
path	目的目录。
/A	指定要在目的目录中加入不存在的文件而不是覆盖已有文件。
/P	加入文件前提示用户确认。
/W	查找源文件前等待用户按任意键。

注意

使用/A 选项,REPLACE 将每个匹配的源文件加入到目的目录中,除非该目录中已存在同名文件。

只能指定一个目的目录。不能使用/S 选项(处理整个目录分枝)。

应用举例

```
REPLACE \DONNA\NEWDATA.* \JUDI /A /P
```

对于 DONNA 中的每个 NEWDATA 文件(扩展名任意),如果在 JUDI 中不存在同名文件,则将其拷贝到 JUDI 中。加入每个文件前提示用户进行确认。

Ⓔ RESTORE

见第 24 章

恢复由以前的 DOS 版本(DOS 2.0 到 5.0)的 BACKUP 命令备份的文件。

句法

```
RESTORE drive1 destination [/S] [/P] [/B:date] [A:date] [/E:time] [/L:time] [/M]
[/N] [/D]
```

参数和选项

drive1	指定包含备份文件的驱动器。
destination	指定要恢复的文件和恢复到的目录。destination 中必须包含驱动器名。可以包含路径和

文件来指定文件。

/S 将文件恢复到所有目录中。

/P 恢复只读文件或上次备份后已被改变过的文件前提示确认信息。

/B:date 只恢复最后一次修改日期是 date 或以前的文件。

/A:date 只恢复最后一次修改日期是 date 或以后的文件。

/E:time 只恢复最后一次修改时间是 time 或更早的文件。

/L:time 只恢复最后一次修改时间是 time 或更晚的文件。

/M 只恢复在上次备份后被修改过的文件。

/N 只恢复已存在的目的文件。

/D 只显示将被恢复的文件列表而不执行实际的恢复操作。使用本选项时,即使不恢复文件也需要指定 destination。

注意

RESTORE 不恢复系统文件 IO.SYS 和 MSDOS.SYS。RESTORE 不能在已被 ASSIGN 或 JOIN 命令重定向的驱动器上工作。

RESTORE 会提示用户把一套备份盘按照生成时的顺序插入驱动器。它在每张盘上查找匹配的文件。

如果用户知道哪张盘上有所需文件,可以略过一些盘然后在 RESTORE 询问是否略过它们时进行确认。

在 RESTORE 读每张盘之前可以按 Ctrl+C 键取消其操作。

一个文件可能跨越两张或更多的盘。当这样的文件只恢复了一部分时不要取消 RESTORE 的执行。

文件被恢复到备份时所在的目录中,但不一定要在同一驱动器上。例如,可以把从 D 盘备份下来的文件恢复到 E 盘上。RESTORE 会在目的盘上生成任何所需目录以使文件被恢复到备份时的目录中。

当使用日期或时间选项时,RESTORE 只注意目的文件而不是源文件的日期和时间标记。假定要恢复所有在 93 年 4 月 15 日后被改动过的 BANDLIST.* 文件,使用 RESTORE A: C:\BANDLIST.* /A:4/16/93。

警告: 使用时间选项时要小心。时间选项对每天都有作用,而不只是在日期选项中指定的一天。/A:4/16/93 /L:3:00P 恢复 4/16/93 的 3:00p.m. 之后,4/17/93 的 3:00p.m. 之后,4/18/93 的 3:00p.m. 之后等等被改动过的文件。

选项/M 根据目的盘上的归档属性恢复自从上次备份以来被改动过的文件。不重新生成在备份后被删除的文件。不恢复备份后未改动的文件。不恢复未改动过的文件可以减少恢复时间。

RESTORE 设置返回码如下:

0 执行成功。

1 找不到要恢复的文件。

3 用户按 Ctrl+C 或 Ctrl+Break 键中断本操作。

4 RESTORE 因出错(如磁盘已满)而停止。

应用举例

```
RESTORE A: C:\*.BAT
```

将所有原来位于被备份盘根目录下的扩展名为.BAT 的所有文件恢复到 C 盘根目录下。

```
RESTORE A: C:\*.BAT /S
```

恢复所有扩展名为.BAT 的文件到 C 盘,包括从子目录中备份下来的文件。每个文件所恢复到的目录与备份时的路径名相同,必要时生成目录以复制正确的路径名。

```
RESTORE A: C:\PROWORD\*.DOC /M
```

寻找从目录\PROWORD 中备份的所有扩展名为.DOC 的文件。如果 RESTORE 在备份盘和 C:\PROWORD 中同时发现一个这样的文件,并且目的文件在备份后已改动过,则用备份版本代替改动版本(注意在备份后创建或删除的文件不受影响)。

RMDIR

见 RD。

SCANDISK

见第 27 章

检测和修正驱动器中存在的问题。

句法 1

SCANDISK [drive... |/ALL] [/CHECKONLY|/CUSTOM|/AUTOFIX[/NOSAVE]] [/SURFACE] [/MONO] [/NOSUMMARY]

检查和修理一个或多个驱动器。

参数和选项

无参数	打开 SCANDISK 对话框,检查当前驱动器。发现错误后显示信息以便用户决定如何处理。还询问用户是否进行表面测试。
drive	要检查的驱动器。
/ALL	检查所有本地驱动器。
/CHECKONLY	检查指定的驱动器并报告错误,但不试图修理。
/CUSTOM	用文件 SCANDISK.INI 内[Custom]部分的参数控制扫描。
/AUTOFIX	发现错误后进行修理而不征求用户意见。若不指定/NOSAVE 选项,丢失的簇被保存。
/NOSAVE	当使用了/AUTOFIX 时,指定删除丢失的簇。
/SURFACE	不加提示进行表面测试。
/MONO	用黑白方式显示 SCANDISK 对话框。
/NOSUMMARY	扫描后略过概要报告。SCANDISK 修理磁盘中的问题也不再要求一个取消(Undo)盘。

注意

SCANDISK 可以检查驱动器的逻辑结构,也可以对它们进行表面测试。逻辑测试寻找驱动器的 FAT 和目录结构中的错误和逻辑异常,就像 CHKDSK 一样(但比 CHKDSK 做得好)。表面测试检查每个扇区是否存在物理缺陷。如果发现物理错误,SCANDISK 可以进行修理,方法是将尽量多的数据移到盘上一个安全的地方,然后标记该扇区为坏区。(某些数据可能已损坏而无法挽回)。

SCANDISK 能检查硬盘、RAM 盘、软驱、内存卡、和 DriveSpace 或 DoubleSpace 压缩盘。它不能检查 CD-ROM 驱动器,网络驱动器或由 INTERLNK、ASSIGN、SUBST、JOIN 生成的驱动器。

如果 SCANDISK 发现丢失的簇(标记为含有数据的簇却不属于任何文件),会询问是保存还是删除。若用户选择保存,SCANDISK 就把每个丢失的簇链转换成一个名为 FILEnnnn.CHK 的文件。若选择删除,SCANDISK 立即使这些簇可在将来使用。

若指定/AUTOFIX,SCANDISK 则自动保存丢失的簇,除非还使用了/NOSAVE 选项。

当 SCANDISK 修正交叉链接文件(两个或更多的文件共用相同的簇)时,为每个文件生成一个新拷贝,再将共用的簇分别拷贝给每个文件。然后删除交叉链接的文件。新文件中可能有一个是正确的,其他文件至少包含一些非法数据而且可能丢失了一些数据。如果损坏的文件中包含的是文本数据,一定要保存所有检测到的族簇,因为从中可能会恢复一些丢失的数据。

不要在 Windows(或任何其他程序)下运行 SCANDISK。已打开的文件会错误引导 SCANDISK,错误地修正数据会导致严重的数据丢失。如果 SCANDISK 发现是在其他程序下运行,则在没有/CHECKONLY 选项的情况下会拒绝继续运行(即使采用/CHECKONLY 选项也会导致不正确的报告)。

SCANDISK 打开对话框使用户可以了解进展情况并报告结果。使用/MONO 选项来设置在黑白方式下显示对话框。

如未指定/SURFACE 选项,SCANDISK 进行表面测试前会提示用户。

如果在未用/AUTOFIX 的情况下发现错误,SCANDISK 会询问用户是否进行修正。

若选择修正错误,SCANDISK 会提示用户保存信息,这样在必要时可以恢复到修正错误前的状态。如果要保存修正错误前的信息,必须提供一张软盘(不能在被扫描的驱动器内保存信息)。使用修正错误前保存的信息取消修正操作的命令参见句法 3。(事先保存的信息中包括损坏的 FAT 和目录结构。)

SCANDISK 只在第一次进行修复时提示选择恢复(Undo)盘。进行过一次修改后,FAT 和目录结构已被改变,改变后的版本已无助于将磁盘恢复到原来的、未修复前的状态。

可以像扫描其他硬盘一样扫描已装入的 DriveSpace 或 DoubleSpace 的压缩盘。但还可以通过扫描未装入(unmount)的 DriveSpace 或 DoubleSpace 压缩卷文件(CVF)来进行更多的检查。当扫描未装入 CVF 时,SCANDISK 检查卷头(volume header)、文件结构、压缩结构和卷标记(signature)。将一个卷卸出(unmount)并指定卷名,如 DRVSPACE.000 或 DBLSPACE.000,以替代装入时的驱动器名。不要使用/ALL 或/SURFACE 选项。

SCANDISK.INI 文件包含 SCANDISK 的配置信息,可以使用诸如 EDIT 之类的 ASCII 文字编辑器编辑这个文件,该文件还包含变量及如何修改变量的说明。

SCANDISK 设置的返回码如下:

0	未检测到错误。
1	SCANDISK 命令中有句法错。
2	内存不够或遇到某种内部错误。
3	用户取消 SCANDISK(但不是在表面测试过程中)。
4	用户选择进行表面测试但在开始后将其取消。
254	发现错误并已修正。
255	未纠正所有错误。

应用举例

所有例子都假定命令是在初级命令提示符下输入的。

```
SCANDISK A:
```

在驱动器 A 中的磁盘上进行逻辑扫描。若发现错误,纠正前提示用户。如果要纠正错误,则提示用户要求一张恢复(Undo)盘。逻辑扫描结束后,询问用户是否进行表面测试。最后显示概要报告。

```
SCANDISK /CHECKONLY /SURFACE
```

扫描当前驱动器但不纠正错误。表面测试前不询问用户。最后显示概要报告。

```
SCANDISK /ALL /AUTOFIX /NOSAVE /NOSUMMARY
```

对所有本地驱动器进行完全自动的逻辑扫描,自动修正错误,丢失的簇被删除,不要求恢复(Undo)盘,不提示进行表面测试,不显示概要报告。

```
DRVSPACE D:/UNMOUNT
SCANDISK D:\DRVSPACE.000 /CUSTOM
DRVSPACE D:/MOUNT
```

第一个命令卸出(unmount)作为驱动器 D 的 DriveSpace 卷。第二个命令用 SCANDISK.INI 中的缺省参数和选项扫描此卷。(驱动器名 D 现在代表宿主驱动器。)第三个命令重新将卷装入(mount)。

句法 2

SCANDISK filespec /FRAGMENT

报告文件碎块情况。

参数和选项

filespec	要检查的文件。
/FRAGMENT	报告碎块情况而不是进行正常的磁盘扫描。

注意

有碎块的文件位于磁盘上几个不同的区域中。过多的文件碎块会降低整机的性能。可以使用 DEFRAG 命令消除碎块。

当使用/FRAGMENT 选项时,SCANDISK 对话框不出现,也不进行表面检测。

参数 filespec 中可包含通配符。

每个 SCANDISK /FRAGMENT 命令只检查一个目录。

应用举例

SCANDISK D:*.*

显示 D 盘根目录下所有文件的碎块情况。

句法 3

SCANDISK /UNDO [undo-drive] [/MONO]

将一个驱动器恢复到修正错误前的状态。

参数和选项

/UNDO	执行 SCANDISK 的 UNDO(恢复)功能。
undo-drive	指定含有恢复信息的软盘。
/MONO	用黑白方式显示 SCANDISK 对话框。

注意

UNDO 功能利用存在恢复盘上的信息恢复驱动器 FAT 和目录结构,从而使驱动器回到未修正错误时的状态。

警告: 如果在修正错误后改动过驱动器中的内容,就不要用 SCANDISK 执行恢复(UNDO)操作了。如果执行 SCANDISK 后增加、修改或删除过文件,恢复以前的 FAT 和目录结构会导致数据的丢失。

不要指定要被恢复的驱动器名,因为恢复盘上含有这些信息。用户只需指定恢复盘的位置。

应用举例

SCANDISK /UNDO A:

利用驱动器 A 中的恢复盘恢复至驱动器执行修正操作前的状态。

---

① SET

见第 43 章

设置或改变环境变量的值,或显示当前值。

句法

SET [variable=[string]]

参数和选项

无参数	显示所有环境变量的当前值。
variable	要设置的环境变量。

string　　为环境变量指定新值。若缺省，则当前值被消除。

注意

使用 SET 命令时，可能会出现超出环境空间的错误信息。能够增加环境空间的命令参见 SHELL 和 COMMAND 命令。

DOS 设置某些环境变量时不需要 SET 命令。例如，当用户在启动文件的多重系统配置中选择了一项时，DOS 设置 CONFIG 标记该选项；它将 PATH 初始化为 C:\DOS；设置 PROMPT 为 PG；根据所找到的 COMMAND.COM 的位置设置 COMSPEC 为 C:\或 C:\DOS。除 CONFIG 外，可分别用 PATH、PROMPT 和 SHELL 命令覆盖缺省值。

DOS 也使用某些由用户利用 SET 命令生成的环境变量。其中包括 TEMP，指定一个保存临时文件的目录；DIRCMD，为 DIR 命令提供一套缺省选项；COPYCMD，为 MOVE、COPY 和 XCOPY 提供缺省选项；以及 MSDOSDATA，为 MSBACKUP 和 MSAV 提供缺省目录。

在批处理程序中可以使用环境变量，但在 DOSKEY 宏中却不行。用两个百分号括住变量名。例如，%CONFIG%代表 CONFIG 的当前值，它是在用多重设置的 CONFIG.SYS 启动时由 DOS 设置的。在批处理程序中使用像 GOTO %CONFIG%这样的命令使程序转到与 CONFIG 当前值匹配的标号处。这使得用户可以编制 AUTOEXEC.BAT(或其他批文件)，以便根据启动 DOS 时作出的选择而完成不同的功能。

当在一个 IF 条件等式中使用环境变量时，考虑到可能出现空值，应将整个变量，包括百分号在内，用引号引起来，如 IF "%CONFIG%"=="SHORT"。(但在其他情况下不要用引号括住环境变量)。

可以在 SET 命令中使用环境变量已有的值。例如，命令 SET PATH=%PATH%;C:\PROGS 将目录 C:\PROGS 加到当前查找路径的末尾。

应用举例

```
SET
```

显示所有环境变量的当前值。

```
SET TEMP=F:\TEMP
```

设置 TEMP 的值为 F:\TEMP。DOS 命令、Windows 和一些应用程序均使用 TEMP 指定的目录存储临时文件。

```
SET DNT=C:\OLD_DOS
```

生成(如果不存在)一个名为 DNT 的环境变量并赋值 C:\OLD_DOS。命令执行后，每次在批文件中遇到%DNT%，它都被换成 C:\OLD_DOS。

```
SET TEMP=
```

清除环境变量 TEMP。(TEMP 被清除后，DOS 在当前驱动器的根目录下存储临时文件)。

Ⓔ SETUP

见第 4 章

从软盘安装 DOS。

句法

```
SETUP [/B] [/E] [/F] [/G] [/H] [I] [/M] [/Q] [/U]
```

参数和选项

无参数　　在硬盘上安装全部的 DOS 系统，允许用户选择非必需程序。

/B　　用单色而不是彩色方式显示 Setup 屏幕。

/E　　只允许安装 Windows 和 DOS 的可选程序。

/F　　往软盘上安装 DOS(安装最小系统)。

/G	指定 Setup 不生成撤销安装过程的(uninstall)盘,也不提示用户更新网络。
/H	使用缺省的 Setup 选项。
/I	关掉视频硬件检测。
/M	安装最小的 DOS 系统。
/Q	拷贝 DOS 文件到硬盘。
/U	即使 SETUP 发现与 DOS 不兼容的磁盘分区也进行安装。

注意

SETUP 位于 DOS 安装盘上。安装 DOS 时不把它放到硬盘上。

第 4 章详细描述了 SETUP 正常运行时的情况。前面提到的选项中,除/E 外都很少用到。

当使用/E 选项运行 SETUP 时,它直接跳过一个屏幕,允许用户选择 DOS 和(或)WINDOWS 版本的 MSBACKUP、UNDELETE 和防病毒程序。例如,原来已安装了 DOS 版本的 MSBACKUP,可以再增加它的 Windows 版本。将安装盘的第一张插入驱动器,运行软盘上的 SETUP /E,然后选择 WINDOWS 版本。DOS 版本的相应程序并不会被去掉。

应用举例

```
SETUP
```

在硬盘上安装 DOS,允许选择 DOS 和 WINDOWS 版本的备份、恢复和防病毒程序。

```
SETUP /E
```

允许用户选择并安装备份、恢复(undelete)和防病毒程序而无需完全重装 DOS。

Ⓔ SETVER

见第 21 章

设置 DOS 报告给程序或设备驱动程序的版本号。显示版本表格。

句法

```
SETVER [path] [filename.ext version] [/DELETE[QUIET]]
```

参数和选项

无参数	由当前目录或查找路径中的 SETVER.EXE 来显示版本表。
path	指定 SETVER.EXE 的位置。
filename.ext	指定一个程序文件。
version	报告给指定程序的 DOS 版本号。
/DELETE	从版本表中删除 filename.ext。该选项的缩写形式为/D。
/QUIET	从版本表中删除程序时,不显示常见的警告消息。

注意

有些程序在启动时检查 DOS 版本号。它们可能被设计成只在指定的版本号下运行,如 DOS 5.0。当在 DOS 6.0 或更高版本中运行时,会出现一条信息说明 DOS 版本不对。某些这样的程序可以在 DOS 6 下正确运行,只需要报告给它一个不同的版本号。另外一些则不能正常运行,利用 SETVER 欺骗它们可能会导致出错或数据丢失。如果把一个程序加到 SETVER,在第一次运行这个程序前一定要备份硬盘。

版本表中含有程序和 DOS 版本号的列表,它被保存在 SETVER.EXE 中。启动一个列于表中的程序时,如果 SETVER 版本表已被加载,SETVER 就会告诉程序它正在表中为其指定的版本号下运行。

关于装入版本表的 CONFIG.SYS 命令参见附录 A 中的 SETVER.EXE。如果不装入版本表,可以用 SETVER 命令查看和更新版本表,但 SETVER 不向程序报告版本号。

DOS 6 的 SETUP 程序通常在 CONFIG.SYS 中插入一条命令以装入由 Microsoft 提供的版本表。如果要从 CONFIG.SYS 中删除这一命令,首先用 SETVER(无参数)命令查看一下表中的程序是否正在使用或

者是否支持用户正在使用的其他程序。如果不能确定，不要删除装入 SETVER 版本表中的命令。

某些包含于以前的 DOS 版本中但不在 DOS 6 中的程序(如 ASSIGN、EDLIN 和 JOIN)被列于 Microsoft 提供的版本表中，这样在使用时就不会出现错误信息。

把 DOS 6 的命令解释器(COMMAND.COM)加到版本表中会使系统无法启动。

当使用 SETVER 增加或删除版本表中的项时，改动不会立即生效，要等到再次启动计算机后从 CONFIG.SYS 中加载 SETVER 时才会将修正过的表调入内存。(如果 CONFIG.SYS 中没有包含装入 SETVER 的命令，则不会装入版本表)。

参数 path 并不影响执行命令所需加载的 SETVER 程序的位置，它仍按 DOS 通常的查找过程定位，但它会影响所要显示或更新的 SETVER 版本表的位置。通常若是更新 SETVER 的唯一版本则不需要 path 参数。但如果有两份版本表，例如，一个在 C:\DOS，一个在 C:\ALTDOS，要查看或更新 C:\ALTDOS 中的版本表则需使用 path 参数。

SETVER 设置返回码如下：

0	成功完成。
1	非法命令选项。
2	无效文件名。
3	内存不够。
4	非法的版本号格式。
5	文件名不在版本表中。
6	未找到 SETVER.EXE 文件。
7	(参数 path 中的)驱动器非法。
8	太多的命令行参数。
9	缺少命令行参数。
10	读 SETVER.EXE 文件出错。
11	SETVER.EXE 文件被破坏。
12	指定的 SETVER.EXE 文件不支持版本表。
13	版本表已满。
14	写 SETVER.EXE 文件出错。

应用举例

```
SETVER MYPROG.EXE 5.0
```

在版本表中加入一项，使 MYPROG.EXE 被告知在 DOS 5.0 下运行。

```
SETVER MYPROG.EXE /DELETE
```

从版本表中去掉 MYPROG.EXE。

```
SETVER E:\ |SORT|MORE
```

显示在 E 盘根目录下的 SETVER.EXE 文件中的版本表，将各项排序并每显示一页后暂停。

Ⓔ SHARE

见第 51 章

加载 SHARE TSR，其作用是为磁盘或网络驱动器安装文件共享和锁定功能。

句法

```
SHARE [/F:space] [/L:locks]
```

参数和选项

无参数　　以缺省值加载 SHARE。

/F:space 以字节为单位,为记录文件共享信息分配空间。缺省值是 2048。

/L:locks 设定可以同时锁定的文件数。缺省是 20。

注意

SHARE 在网络或程序共享文件的多任务环境中支持文件共享和锁定。SHARE 验证程序所有读写请求的合法性。例如,两个程序不能同时写同一个文件。

每个打开的文件要求足够的空间以容纳绝对路径和文件名。路径名和文件名的平均长度是 20 个字符。

可以不使用本命令而在 CONFIG.SYS 中用 INSTALL 命令装入 SHARE.EXE。对于在启动期间遇到共享问题的系统,这种作法使启动时能够更早地利用 SHARE 提供的共享功能。

应用举例

```
SHARE
```

以缺省值装入 SHARE。

```
SHARE /F:4096 /L:300
```

装入 SHARE,为文件信息分配 4096 字节空间,最多锁定 300 个文件。

Ⓒ SHELL

见第 43 章

为 DOS 指定命令解释器。

句法

```
SHELL=filespec [parameters]
```

参数

filespec 指定命令解释器。

parameters 为命令解释器指定命令行参数和选项。

注意

不用 SHELL 命令时,缺省的命令解释器是 COMMAND.COM。首先 DOS 在启动盘的根目录下寻找它,然后是\DOS。如果要使用另一个或另一目录中的命令解释器,必须在 CONFIG.SYS 中使用 SHELL 命令。

可用于 COMMAND.COM 的参数和选项见 COMMAND 命令。如果使用其他命令解释器,参数和选项的用法参见其相应文献。

有两个选项可用于 SHELL 命令中的 COMMAND.COM,但不能用命令提示符下的 COMMAND 命令:

/P 使 COMMAND.COM 的新副本成为永久命令解释器。(EXIT 命令不能去掉它)。若指定 /P 选项,加载命令解释器时会执行 AUTOEXEC.BAT。若没有 AUTOEXEC.BAT,则执行 DATE 和 TIME 命令。

/MSG 将 COMMAND.COM 的信息装入内存。如果是从软盘加载 COMMNAD.COM 而又不希望每次 DOS 需要显示信息时都读软盘,可用此选项。本选项只能与/P 联用。

应用举例

```
SHELL=C:\SYS\COMMAND.COM C:\SYS /E:2048 /P
```

从 C:\SYS 中加载 COMMAND.COM 作为初级命令处理器并指定 2048 字节的环境空间。

Ⓑ SHIFT

见第 37 章

将传递给批文件的命令行参数左移。

句法

SHIFT

注意

命令 SHIFT 从%0 到%9 移动命令行参数，使参数%1 成为%0、参数%2 成为%1，等等。

应用举例

```
:LOOPER
IF "%1"== "" GOTO ENDING
GOTO ENDING
DEL %1
SHIFT
GOTO LOOPER
:ENDING
```

这段程序删除若干个文件，由用户在批处理命令后提供所有文件名。每次执行 SHIFT 后，命令行中的一个新名字替换%1。当命令行中没有名字时，%1 成为空值，由 IF 命令将程序转到:ENDING。

```
:COMPARE
IF "%2"==""
GOTO ENDING
FC %1 %2
SHIFT
SHIFT
GOTO COMPARE
:ENDING
```

本程序比较若干对文件，文件名由命令行提供。需要两个 SHIFT 命令，这样每次循环时%1 和%2 同时被新值替换。

Ⓔ SMARTDRV

见第 31 章

启动或设置 SmartDrive，利用扩展内存(extended memory)生成磁盘缓冲。写入被缓冲的数据。

句法 1

SMARTDRV [drive[+|-]...] [/E:element] [InitCache[WinCache]] [/B:buffer] [/R] [/L] [/Q] [/V] [/S] [/X] [/U] [/F|/N]

启动或设置 SmartDrive。

参数和选项

无参数	若尚未装入 SmartDrive，用缺省值装入程序的 TSR 部分。显示驱动器的缓冲状态。
drive	需要控制其缓冲状态的磁盘驱动器(驱动器符后不加冒号)。
+|-	+允许 drive 的读写缓冲，-禁止 drive 的读写缓冲。缺省值见注释。
/E:element	指定 SmartDrive 一次读写的字节数。可以是 1024、2048、4096 或 8192，缺省是 8192。
InitCache	以千字节为单位指定启动 SmartDrive 并且不运行 Windows 时缓冲区的大小。缺省值取决于系统内存(见注释)。
WinCache	以千字节为单位指定当 Windows 启动时 SmartDrive 可将缓冲区缩减到的最小数量。缺省值取决于系统内存(见注释。)
/B:buffer	指定预读缓冲区的大小。可以是 element 的任何倍数。缺省是 16KB。
/R	清除缓冲区中的内容。
/L	防止 SmartDrive 自动装入高端内存，即使有可用的 UMB(高端内存块)。

/Q 防止 SmartDrive 在启动时显示状态信息。

/V 启动 SmartDrive 时显示状态信息。

/S 显示关于 SmartDrive 状态的额外信息。

以下选项可用于 DOS 6.2 或更高版本：

/F 每个命令结束时被缓冲的数据写盘(对于 DOS 6.2 此项为缺省值)。

/N 每个命令结束时被缓冲的数据不写盘。

/U 不装入 CD-ROM 缓冲模块。

/X 禁止所有驱动器的写缓冲。

注意

SmartDrive 在扩展内存(extended memory)中保存缓冲内容。在使用 SMARTDRV 命令前，必须装入扩展内存管理器，如 HIMEM.SYS。

不要在 Windows 启动后使用 SMARTDRV 命令，Windows 有自己的缓冲管理。

缺省情况下，SmartDrive 按以下方式为驱动器提供缓冲：

软驱	提供读缓冲但不提供写缓冲
INTERLNK 盘	提供读缓冲但不提供写缓冲
CD-ROM 驱动器	提供读缓冲但不提供写缓冲
硬盘	即提供读缓冲又提供写缓冲

其他类型的驱动器(RAM、网络、内存卡)不能被缓冲。压缩盘本身不被缓冲，但任何为宿主盘提供的缓冲也缓冲了压缩盘。

对于 DOS 6.0，CD-ROM 驱动器不被缓冲。对 DOS 6.2 或更高版本，缺省情况下，只有在装入 SmartDrive 之前装入 MSCDEX 时，CD-ROM 才被缓冲。

在装入 SmartDrive 时或这以后的任何时刻，都可以用 drive[+|-]选项覆盖缺省的缓冲状态。

选项/E、/B、/L、InitCache 和 WinCache 只能在安装 SmartDrive 时使用。第一次执行本命令时使用/C 或/S 选项防止装入 SmartDrive 和可能产生的帮助信息(选项/C 见句法 2)。

若使用/E 选项，还要指定 InitCache，并在命令行中将/E 放在 InitCache 之前。否则，缓冲的容量会被减小。

缓冲区和读写单元(element)越大，访问磁盘次数越少，系统效率越高。

装入 SmartDrive 时，缓冲区的大小是 InitCache。Windows 启动时，将缓冲区减少到 WinCache，以便为 Windows 操作保留更多空间。Windows 结束后，缓冲区大小恢复到 InitCache。

如果在 SMARTDRV 命令行中只包含一个缓冲区大小，它被同时用于 InitCache 和 WinCache。也就是说，Windows 不能减少缓冲区。

若不指定缓冲区大小，SMARTDRV 则使用下表所示的缺省值：

扩展内存	InitCache	WinCache
1MB	所有扩展内存	0
2MB	1MB	256KB
4MB	1MB	512KB
6MB	2MB	1MB
6MB 或更多	2MB	2MB

如果有高端内存可用，SmartDrive 会自动装入高端内存(除非使用/L 选项)。对于 SMARTDRV 不需使用 LH 命令。

某些系统需要装入双缓冲(double-buffering)版本的 SmartDrive。关于何时需要及如何装入双缓冲的信息参见附录 A 中的 SMARTDRV.EXE。

应用举例

SMARTDRV

若尚未装入 SmartDrive，将其装入，使用缓冲、读写单元和缓冲区大小的缺省值和所有驱动器的缺省状态。如果已经装入 SmartDrive，显示状态报告。

SMARTDRV A- B- C 2048 512 /E:2048

若尚未装入 SmartDrive，将其装入并拥有 2048K 的缓冲区。Windows 启动时允许缓冲区减到 512KB。使用 buffer 的缺省值(16KB)。指定 SmartDrive 以 2048 字节为单位读写数据。关掉对驱动器 A 和 B 的缓冲。对驱动器 C 只允许读缓冲。(其他驱动器按缺省设置进行缓冲)。

若已装入 SmartDrive，则忽略 InitCache、WinCache 和 element。改变驱动器 A、B、C 的缓冲状态。其他驱动器的缓冲状态不变。

SMARTDRV /S

如果已装入 SmartDrive，显示范围更广的状态报告。

句法 2

SMARTDRV /C

将所有写缓冲区的信息从内存写到磁盘上。

注意

如果在 DOS 6.0 下对任何驱动器提供写缓冲，最好在关掉系统或按复位(reset)键前将写缓冲区中的数据写到磁盘上。但在按 Ctrl+Alt+Del 键前不需要，在这种情况下，SmartDrive 自动把写缓冲中的数据写回磁盘。

如果在 DOS 6.0 下使用 DriveSpace 或 DoubleSpace，那么在关掉或复位系统前使用 SMARTDRV /C 尤其重要，否则会使压缩盘的数据丢失。

在缺省情况下，DOS 6.2 的 SmartDrive(及更高版本)只要一回到命令提示符就把写缓冲区中的数据存盘，这样就不需要使用 SMARTDRV /C 命令。然而，如果装入 SmartDrive 时用了 /N 选项，仍需按前面所说的方法处理缓冲区中的内容。

Ⓔ SORT

见第 36 章

排列 ASCII 数据的顺序并显示结果。

句法

SORT [/R] [/+n]

参数和选项

/R　　反向排序。

/+n　　从第 n 列起排序。缺省是第一列。

注意

命令 SORT 不区分待排序数据中的大小写字母。

SORT 能处理的文件最大为 64KB。

在正常情况下，字符按照 ASCII 码的顺序排列。数字 0 到 9 排在字母 A 到 Z 之前。特殊字符或非打印字符的位置取决于它的 ASCII 码。

如果在 CONFIG.SYS 中有 COUNTRY 命令，SORT 就使用对应于国家代码和代码页设置的顺序对照表。

在缺省情况下，SORT 从标准输入设备中接收输入并将排序结果写到标准输出设备中。可以重定向输

入使其来自于文件，或者使用前一个命令的管道输出，重定向其输出到文件或设备，或者用管道命令将其输出传输到另一个命令。

应用举例

SORT

从键盘接收输入，直到用户按下 Ctrl+Z 键(F6 键作用相同)，再将输入排序，然后将排序后的数据显示在屏幕上。(注意排序后的数据不存放在文件中，显示后即丢失)。

SORT /+2 <ADDRESS.LST >LPT1

从每行的第 2 列字符开始对文件 ADDRESS.LST 中的数据排序，输出送到打印机。(注意排序后的数据不存放在文件中)。

TYPE ADDRESS.LST | SORT /+2 > LPT1

与前例作用相同。

SETVER|SORT|MORE

按字母顺序列出 SETVER 版本表的内容，每次一页。

© STACKS

见第 21 章

支持使用数据堆栈来处理硬件中断。

句法

STACKS=number,size

参数

number 指定堆栈个数。有效值是 0 或从 8 到 64。对于 IBM PC、IBM PC/XT 和 IBM PC 便携机，缺省值是 0。对于其他计算机，缺省值是 9。

size 以字节为单位指定每个堆栈的大小。有效值是 0 或从 32 到 512。对于 IBM PC、IBM PC/XT 和 IBM PC 便携机，缺省值是 0。对于其他计算机，缺省值是 128。

注意

只能在 CONFIG.SYS 中使用 STACKS 命令。

DOS 在接到一个硬件中断时分配一个堆栈。但如果对 number 和 size 都指定为 0，DOS 则不分配堆栈，每个程序必须用自己的堆栈空间满足计算机的硬件驱动程序。很多程序在 number 和 size 设成 0 时可以正确运行。

如果把 number 和 size 设成 0 后计算机操作变得不稳定，那么恢复缺省设置。

如果 number 和 size 的值不是 0，并且出现 Stack Overflow 或 Exception error 12 信息，就需增加 number 或 size 的值。

应用举例

STACKS=12,512

为处理硬件中断分配 12 个堆栈，每个堆栈 512 个字节。

© SUBMENU

见第 38 章

在启动菜单中定义一项，选择该项时出现另一套选项。

句法

SUBMENU=blockname[,text]

参数

blockname　指定定义了子菜单的环境设置块名。

text　该菜单项显示的文本。缺省值是 blockname。

注意

只能在含有多重设置的 CONFIG.SYS 中的主菜单块或子菜单块中使用 SUBMENU 命令。对于菜单块的描述参见 MENUITEM 命令。

在 CONFIG.SYS 其他地方必须定义一个名为[blockname]的语句块。如果 DOS 找不到[blockname]，菜单中就不含这一项。

参数 blockname 最长为 70 个字符，并可包含除空格、反斜线、斜线、逗号、分号、等号和方括号外的一切可打印字符。

参数 text 最长为 70 个字符，并可包含任何可打印字符。

应用举例

```
[MENU]
SUBMENU BASIC,BASIC STARTUP
MENUITEM WINDOWS
[BASIC]
MENUITEM JOE
MENUITEM MARY
...
```

这个启动菜单块产生如下启动菜单：

```
1. BASIC STARTUP
2. WINDOWS
```

当用户选择 1 时，出现另一个菜单：

```
1. JOE
2. MARY
```

Ⓔ SUBST

见第 39 章

重新指派一个驱动器名使其代表一个路径。

句法 1

SUBST [drive:path]

使以后对 drive 的访问变成对 path 的访问。显示当前替代状态。

参数

无参数　显示当前起作用的替代。

drive　要被重新指派的驱动器名。

path　指定由驱动器代表的路径。

注意

参数 drive 不必是已存在的驱动器，但它必须在系统可能的驱动器名范围内(见 LASTDRIVE 命令)。drive 不能是当前驱动器。

参数 path 必须是现存驱动器上的现有目录。path 中可包括驱动器名，如果没有，DOS 则认为 path 是在执行 SUBST 命令时的当前目录上。

一旦建立替代，对 drive 的读写被重定向到 path。如果 drive 实际上存在，那么在取消替换前不能对其进行访问。(如果 DOS 目录在 drive 上，将无法执行任何 DOS 外部命令，包括 SUBST，只有重新启动才能清

除替换)。

不要在已被替代的驱动器名上使用下列命令:ASSIGN、MSBACKUP、CHKDSK、DEFRAG、DISKCOMP、DISKCOPY、FDISK、FORMAT、LABEL、MIRROR、RECOVER、RESTORE、SYS 和 UNDELETE /S。

应用举例

SUBST F: C:\DATADIR

使以后对驱动器 F 的访问变成对 C 盘 DATADIR 目录的访问。

SUBST A:MYDIR\D1

使以后对驱动器 A 的访问变成对当前驱动器的 MYDIR 目录的 D1 子目录的访问。

SUBST A: B:\

使以后对驱动器 A 的访问变成对驱动器 B 的根目录的访问。

SUBST

列出当前所有替代。

句法 2

SUBST drive: /D

取消对 drive 的当前替代。

应用举例

SUBST A: /D

取消对驱动器 A 的任何替代。以后涉及驱动器 A 时仍访问驱动器 A。

© SWITCHES

见第 38 章

为 DOS 的几种特征提供特殊选项。

句法

SWITCHES=[/W] [/K] [/N] [/F]

选项

/W 指定 WINA20.386 所在的目录并非启动盘的根目录。

/K 强制增强型键盘(101 个键)执行传统键盘(84 个键)的功能。

/N 防止用户按 F5 或 F8 键跳过 CONFIG.SYS 和 AUTOEXEC.BAT,或者有条件地执行 CONFIG.SYS 和 AUTOEXEC.BAT。

/F 启动期间显示 Starting MS-DOS...信息时跳过 2 秒钟的延迟。

注意

只能在 CONFIG.SYS 中使用 SWITCHES 命令。

如果正在使用 Windows 3.0 增强模式,并且将 WINA20.386 从启动盘的根目录移到其他目录,应使用 /W 选项。同时还要在 SYSTEM.INI 文件中的[386Enh]标题下加入 Device 命令,指定 WINA20.386 的新位置。若不使用 Microsoft Windows,则不要用/W 选项。

如果某个程序不能正确识别增强型键盘的输入,使用/K 选项。如果装入了 ANSI.SYS 设备驱动程序,在装入 ANSI.SYS 的 DEVICE 命令中也要用/K 选项。

选项/N 不能阻止用户在启动期间按 Ctrl+F5 或 Ctrl+F8 键跳过 DriveSpace 或 DoubleSpace。要想禁止这一功能,输入 DRVSPACE/SWITCHES 或 DBLSPACE/SWITCHES 命令,它们会在 DRVSPACE.INI 或 DBLSPACE.INI 中加入 SWITCHES /N 命令。

应用举例

SWITCHES=/F /N

强制 DOS 在显示 Starting MS-DOS... 信息后跳过 2 秒钟延迟并阻止用户跳过启动命令。

© SYS

拷贝 DOS 系统文件和命令解释器到一个已格式化过的磁盘，使其成为启动盘。

句法

SYS [path] drive

参数

path	系统文件的位置，缺省时为当前驱动器的根目录。
drive	指定将系统文件拷贝到哪个驱动器。

注意

命令 SYS 从指定目录(或缺省目录)中拷贝 IO.SYS 和 MSDOS.SYS 文件并把它们标记为隐含、系统和只读文件。如果 SYS 在 path(或缺省目录)中找到 COMMAND.COM，就拷贝该文件，否则显示信息说明无法拷贝 COMMNAD.COM。如果 SYS 在查找路径中发现 DRVSPACE.BIN 或 DBLSPACE.BIN，也拷贝该文件并标记为系统、隐含和只读文件。若 SYS 找不到 DRVSPACE.BIN 或 DBLSPACE.BIN，也不显示警告信息。

所有上述四个文件大概占用 193K。

SYS 总是把这些系统文件拷贝到 drive 的根目录下。

不需要重新格式化目标盘以便为系统文件保留空间。SYS 根据需要移动文件而将两个.SYS 文件放在磁盘开始处它们所需的地方。

命令 SYS 不能作用于已被 ASSIGN、JOIN 或 SUBST 重定向的驱动器，也不能作用于网络驱动器或被 INTERLNK 重定向的驱动器。

应用举例

SYS A:

从当前驱动器的根目录拷贝系统文件和命令解释器到驱动器 A 中磁盘的根目录。

---

① TIME

见第 12 章

显示并允许修改系统时间。

句法

TIME [time]

参数

无参数	用标准格式显示系统时间并询问新时间。
time	把系统时间改变成 time 参数指定的时间。

注意

系统时间影响目录项、MSBACKUP、RESTORE 和许多应用程序。

显示时间后，DOS 询问新时间。要保持当前时间，按回车键。要改变时间，输入新时间并按回车。也可以将新时间作为参数 time 输入。

在缺省情况下，时间的标准显示格式是美国标准 hh:mm:ss.nnx，其中：

- hh 按 12 小时制格式代表时间
- mm 代表分钟

- ss 代表秒
- nn 代表百分之一秒
- x 是 A(上午)或 P(下午)

输入时间时,可以使用 24 小时制格式而不用指定 A 或 P,小时可从 0(午夜)到 23(下午 11 点)。或者用 12 小时制格式带 A 或 P。指定 A 或 P 时忽略字母大小写。不要输入百分之一秒数。

要把时间设成整小时,可以省略分钟和秒中的零,类似的时间设成整数小时和分钟时,不需要输入秒数。

若要改成其他时间格式,如 hh. mm. ss. nn,在 CONFIG. SYS 中使用 COUNTRY 命令指定一个国家,其日期、时间格式被用作标准格式。

应用举例

TIME

显示系统时间并要求用户输入新时间。

TIME 3:15P

把系统时间改为 3:15:00.00P。

TIME 15:15:30

把系统时间改为 3:15:30.00P。

TIME 2:12

把系统时间改为 2:12:00.00A。

TIME 0

把系统时间改为 12:00:00.00A(午夜)。

TIME 12

把系统时间改为 12:00:00.00P(正午)。

Ⓔ TREE

见第 14 章

以图形方式显示目录结构。

句法

TREE [drive|path] [/F] [/A]

参数和选项

无参数	显示从当前目录开始的目录结构。
drive	要显示目录结构的驱动器。
path	显示该目录分枝的结构。
/F	显示中包括每个目录中的文件名。
/A	指定 TREE 命令用文本字符而不是图形字符显示连接子目录的线。在使用不支持图形字符的代码页或将输出送到不能正确解释图形字符的打印机时可使用此选项。

应用举例

TREE

显示当前目录开始的目录结构。不列出文件名。连接目录的线用图形字符。

TREE A:/F|MORE

显示驱动器 A 中磁盘的目录结构,列出所有目录中的所有文件,每显示一页后暂停。

TREE C:\BOOK /A > LPT1

打印 C:\BOOK 及其子目录的目录结构，连线使用文本字符而不是图形字符。

Ⓘ TRUENAME

见第 39 章

显示目录或文件的真正位置。

句法

TRUENAME [drive|path|filespec]

参数

无参数	显示当前驱动器和路径的真名。
drive	显示 drive 的真名。
path	显示指定路径的真名。
filespec	显示指定文件的真正全名。

注意

当目录结构被 ASSIGN 或 SUBST 屏蔽时可使用 TRUENAME 命令。

命令 TRUENAME 并不检查文件是否存在，而只是用正确的路径名代替被重定向的路径名。

Ⓘ TYPE

见第 16 章

显示文件内容。

句法

TYPE filespec

参数

filespec	要显示的文件。

注意

如果 filespec 不是文本文件，可能无法阅读显示结果。显示完整个文件前可能会响铃几次或停止显示。

应用举例

TYPE REPORT.TXT

在屏幕上显示文件 REPORT.TXT 的内容。

TYPE AFILE.BAT|SORT > LPT1

打印文件 AFILE.BAT 排序后的内容。

TYPE MYLIST|MORE

显示文件 MYLIST 的内容，每显示一页后暂停。

Ⓔ UNDELETE

见第 26 章

允许删除保护或恢复被删除的文件。

句法 1

UNDELETE /LOAD|/UNLOAD|/S[drive]...|/T[drive[-entries]]...|/STATUS

允许或禁止删除保护，报告保护状态。

参数和选项

/LOAD	以缺省保护方式装入 UNDELETE TSR。
/UNLOAD	去掉 UNDELETE TSR，其缩写形式为/U。
/S[drive]	装入 UNDELETE TSR，将删除保护的当前和缺省方式设置成删除卫士(Delete Sentry)。把 drive 加入到被删除卫士保护的驱动器列表中。
/T[drive]	装入 UNDELETE TSR，将删除保护的当前和缺省方式设置成删除跟踪(deletion-tracking)。把 drive 加入到被删除跟踪保护的驱动器列表中。
entries	指定 drive 的删除跟踪文件中最多的项数，可从 1 到 999。缺省值取决于驱动器的容量。
/STATUS	显示被当前保护方式所保护的驱动器。

注意

命令 UNDELETE 提供两级删除保护。较高的一级：删除卫士(Delete Sentry)，和较低的一级：删除跟踪。只有在删除文件时已装入 UNDELETE TSR 的情况下才能奏效。如果两种方法都未使用，仍有可能利用 DOS 的目录信息恢复文件(见句法 2)。

文件 UNDELETE.INI(通常与 UNDELETE.INI 在同一目录中)包含了一些缺省值，如文件被保留多长时间(通常 7 天)，保护哪些文件(经常是 *.TMP、*.VM?，其他文件不被保护)，哪个驱动器用哪种方法保护，以及缺省保护方法是什么。可以使用文本编辑器，如 EDIT，查看和修改 UNDELETE.INI 文件。

若使用删除卫士时删除一个文件，UNDELETE 将其移到一个特别的隐含目录(SENTRY)中。以后 UNDELETE 可以容易而又精确地恢复该文件，因为文件的簇未占用。每个被删除卫士保护的驱动器有自己的 SENTRY 目录。

在经过 UNDELETE.INI 文件中指定的一段时间后，或者 SENTRY 目录占用了驱动器上 20%以上的空间，或者有必要为新数据在驱动器上腾出空间，删除卫士会从 SENTRY 目录中清除文件。首先去掉最旧的文件若未装入删除卫士，则除非使用/PURGE 选项，否则无法从 SENTRY 目录中清除文件(见句法 2)。如果 SENTRY 目录相当大，在禁止删除卫士时很容易出现 Disk full(磁盘空间已满)的信息。

使用删除跟踪方法删除一个文件时，UNDELETE 把文件的目录项和 FAT 链存储在名为 PCTRACKR.DEL 的隐含文件中。如果文件的簇未被重新使用，UNDELETE 就可以利用这些信息恢复文件。在每个被删除跟踪保护的驱动器的根目录下都有自己的 PCTRACKR.DEL 文件。

文件 PCTRACKR.DEL 的容量是固定的，由用户要求保存的项数决定。当文件已装满时，新删除的项取代文件中最旧的项。

对于删除跟踪，参数 entries 的缺省值取决于磁盘容量，如下表所示：

驱动器	Entries
360KB	25
720KB	50
1.2MB	75
1.44MB	75
20MB	101
32MB	202
超过 32MB	303

不要在被 JOIN 或 SUBST 重定向的驱动器上使用删除跟踪。如果想要用 ASSIGN，应在安装删除跟踪前完成。

应用举例

```
UNDELETE /SC /SD
```

装入 UNDELETE TSR，使用删除卫士这种保护方法。把驱动器 C 和 D 加入此方法保护的驱动器列表中(以前被此方法保护的驱动器仍被保护)。

UNDELETE /U

去掉 UNDELETE TSR。

句法 2

UNDELETE [filespec] [/DT|/DS|/DOS] [/LIST|/ALL] [/PURGE[drive]]

恢复被删除的文件。列出可恢复的文件。清除一个 SENTRY 目录。

参数和选项

无参数	使用驱动器上最高级的方法恢复当前目录中所有被删除的文件。
filespec	要恢复的文件。缺省是当前目录中所有被删除的文件。
/DT	只恢复列于删除跟踪文件 PCTRACKR.DEL 中的文件，对每个将恢复的文件要求用户确认。
/DS	只恢复列于删除卫士目录 SENTRY 中的文件，对每个将恢复的文件要求用户确认。
/DOS	只恢复在目录中找到的文件，每个将恢复的文件均要求用户确认。
/LIST	列出可恢复的文件，但并不恢复。列出的文件取决于 filespec 和恢复方法。
/ALL	不加提示恢复文件。被恢复的文件取决于 filespec 和恢复方法。
/PURGE[drive]	清空 drive 上的 SENTRY 目录。缺省是当前驱动器。

注意

使用恢复功能不需要装入 UNDELETE TSR。

不指定恢复方法时，UNDELETE 使用该驱动器上可以使用的最高级方法。例如，如果驱动器上有 SENTRY 目录，UNDELETE 使用删除卫士而忽略 PCTRACKR.DEL 和 DOS 的目录信息。

无论使用哪种恢复方法，UNDELETE 显示每种方法可恢复的文件数，包括使用 DOS 目录的方法，但只恢复所选用方法可以恢复的文件。

当 DOS 删除文件时，改写文件名的第一个字母。如果是用 DOS 目录的方法恢复文件，UNDELETE 提示用户输入文件名的第一个字母。但是选项/ALL 免去了提示，所以 UNDELETE 提供第一个字母(#)。如果第一次试图提供的首字母会产生不唯一的文件名，UNDELETE 就换用另一个字母(%)，等等。选用字母按如下顺序：

#%&0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ。

应用举例

UNDELETE C:\ /DT

恢复在驱动器 C 的根目录下被删除的、被删除跟踪方法保护的文件。恢复每个文件前出现提示。

UNDELETE C:\WORDS*.TXT /DS /ALL

恢复从目录 C:\WORDS 下删除的、被删除卫士方法保护的.TXT 文件。恢复时不带提示。

UNDELETE D:\ACCOUNT /LIST

如果 D 盘是被删除卫士方法保护的，则列出从目录 D:\ACCOUNT 下被删除的、可用删除卫士方法恢复的文件。如果 D 盘是被删除跟踪保护的，列出从目录 D:\ACCOUNT 下删除的，可由删除跟踪恢复的文件。如果 D 未被删除卫士或删除跟踪保护，列出目录 D:\ACCOUNT 中标记已被删除的文件。

Ⓔ UNFORMAT

见第 17 章

恢复被 FORMAT 命令重新格式化的磁盘。

句法

UNFORMAT drive [/L] [/TEST] [/P]

参数和选项

drive	指定要恢复内容(unformat)的驱动器。
/L	略过镜象文件并列出 UNFORMAT 找到的每个文件和子目录。
/TEST	略过镜象文件并显示 UNFORMAT 将如何重建磁盘,但实际上并不重建磁盘。
/P	略过镜象文件并将输出信息送到 LPT1。

注意

如果所做的是无条件格式化,UNFORAMT 不能恢复磁盘。

如果重新格式化后已向磁盘上加入了一些数据,可以恢复至格式化前的状态,但某些旧文件会被新数据破坏。

对硬盘执行 UNFORMAT 命令时,扇区必须是 512、1024 或 2048 字节。

安全或快速的格式化将旧根目录和 FAT 的拷贝保存在一个镜象文件中,该文件位于磁盘上,即使磁盘重新格式化后,UNFORMAT 仍能找到它。不带选项使用本命令时,UNFORMAT 使用文件中的信息重建以前的 FAT 和根目录,因而重新连接簇中未损坏的数据。

如果 UNFORMAT 在磁盘上找不到镜象文件,或用户使用了/L、/TEST 或/P 选项,UNFORMAT 会通过子目录搜寻簇的方法重建磁盘的目录结构(非常慢)。但它无法断定第一级子目录(根目录的子目录)的原名,所以为它们指派了普通的名字——SUBDIR. 1、SUBDIR. 2 等等。然后尽量恢复每个子目录中的文件,但对于有碎块的文件只能恢复第一块。根目录下的文件不能恢复。

在 DOS 6.0 之前,UNFORMAT 命令包含另外三个选项:/U、/J 和/PARTN。这些选项仍然有效,可以在 UNFORMAT 的快速帮助中见到它们,但帮助文件里没有这几项。在 DOS 6.0 或更高版本中它们没有实际作用,因为它们依赖于现已不用的 MIRROR 程序保存的镜象文件和分区表文件。

应用举例

UNFORMAT A:

若能找到 FORMAT 命令生成的恢复信息,利用它恢复驱动器 A 中的磁盘。若找不到镜象文件,就重建磁盘。

UNFORMAT B: /TEST

模拟重建驱动器 B 中的磁盘,搜寻子目录和文件。

---

① VER

见第 12 章

显示 DOS 版本号。

句法

VER

---

① VERIFY

打开或关闭写验证。

句法

VERIFY [ON|OFF]

参数

无参数	显示写验证的状态。
ON	打开写验证,这样 DOS 会验证文件是否被正确写入磁盘。

OFF　　关闭写验证。

注意

即可在 CONFIG.SYS 中也可在命令提示符下使用的 VERIFY 命令。

当打开写验证时，DOS 验证任何写入磁盘的数据是否与内存中的数据匹配。这可以使用户确信在写入过程中未引入新的错误，但它并不验证内存中的数据是否正确。它还会略微降低写盘的速度。

应用举例

VERIFY

显示写验证的状态。

VERIFY ON

为以后的所有命令打开写验证。

---

① VOL

见第 18 章

显示磁盘卷标和系列号(若存在的话)。

句法

VOL [drive]

参数

无参数　　如果当前驱动器中的磁盘有卷标和系列号，则显示它们。

drive　　指定一个驱动器，显示其卷标和系列号。

应用举例

VOL A:

显示驱动器 A 中磁盘的卷标和系列号。

Ⓔ VSAFE

见第 25 章

装入或去除 VSAFE TSR，其作用是持续不断监视计算机，防止可能由病毒引起的操作。

句法

VSAFE [/option+|/option-] [...] [/NE] [/NX] [/Akey|/Ckey] [/N] [/D] [/U]

参数和选项

/option+|-　　指定 VSAFE 如何监测病毒。在选项(option)后用+或-将其打开或关闭。其选项见注释。

/NE　　防止 VSAFE 装入扩充内存(expanded memory)。

/NX　　防止 VSAFE 装入扩展内存(extended memory)。

/Akey　　将显示的 VSAFE 选项对话框的热键设为 Alt+key 键，缺省时设为 Alt+V 键。

/Ckey　　将显示的 VSAFE 选项对话框的热键设为 Ctrl+key 键。

/N　　允许 VSAFE 监视网络驱动器。

/D　　关闭和校验。

/U　　从内存中去掉 VSAFE。不与其他任何选项联用。

注意

要想同时使用 VSAFE 和 Windows，应在文件 WIN.INI 中以 LOAD=开头的行中加入命令 C:\DOS\MWAVTSR.EXE，它使 VSAFE 能在 Windows 中弹出病毒警报。

不要在 VSAFE 运行时安装 DOS、DriveSpace 或 DoubleSpace、Windows。运行安装程序时关闭 VSAFE，

然后再激活 VSAFE。

可以用 LH 命令将 VSAFE 装入高内存区。

选项 options 有效值如下：

1 对可以彻底删除硬盘数据的低级格式化提出警告。缺省是 ON。

2 对程序试图驻留内存发出警告。缺省是 OFF。

3 防止程序写盘，缺省是 OFF。

4 DOS 加载可执行文件时对其进行扫描以防病毒。缺省是 ON。

5 检查所有磁盘以防引导区病毒。缺省是 ON。

6 警告试图写硬盘的分区表或引导区。缺省是 ON。

7 警告试图写软盘的引导区。缺省是 OFF。

8 试图修改可执行文件时向用户示警。缺省是 ON。

可以用热键访问 VSAFE 选项表，显示并允许用户修改每个选项的状态，并可以去掉 VSAFE。

应用举例

VSAFE /7+ /4− /AX

装入 VSAFE，打开选项对试图写软盘引导区的操作进行警告，关掉扫描可执行文件的选项。所有其他选项采用缺省状态。热键设置为 Alt+X 键。

Ⓔ XCOPY

见第 15 章和 40 章

拷贝文件和子目录。

句法

XCOPY source [destination] [/A|/M] [/D:date] [/P] [/S[/E]] [/V] [/W] [/Y|/−Y]

参数和选项

source	要拷贝的文件。
destination	文件拷贝到的位置。缺省时是当前目录。
/A	只拷贝带有归档属性的文件，但不改变归档属性。
/M	只拷贝带有归档属性的文件，然后关掉归档属性。
/D:date	只拷贝在日期 date 后生成或修改的文件。
/P	提示用户确认要拷贝的每个文件。
/S	将命令扩展为拷贝以 source 为整个目录分枝(见注释)。
/E	在拷贝一个目录分枝时拷贝其中的空目录(见注释)。在 DOS 6.2 之前，使用/E 选项时必须用/S 选项。在 DOS 6.2 或更高版本中，使用/E 选项时可以没有/S 选项。
/V	验证每次拷贝。
/W	开始拷贝文件前显示信息并等待用户按任意键。
/Y	覆盖目的文件而不给出警告信息。
/−Y	覆盖文件前要求用户确认。

注意

拷贝多个文件时，XCOPY 要比 COPY 快得多。

命令 XCOPY 拷贝空文件(与 COPY 命令不同)但不拷贝系统或隐含文件。

命令 XCOPY 打开每个文件拷贝的归档属性，同时拷贝源文件的日期、时间标记。

当参数 source 中包括文件名时，destination 中也可包含文件名，并且两个文件可在同一目录中。当 source 的文件名含有通配符并且 destination 是一个目录时，XCOPY 为每个名字匹配的文件制作一份拷贝。

例如，在 XCOPY CHAP*.DOC A:命令中，XCOPY 将 CHAP1.DOC 拷贝到 A:CHAP1.DOC。

如果参数 source 和 destination 中的文件名都有通配符，XCOPY 会生成或替换文件名中有同样通配符的目的文件。例如，在 XCOPY CHAP*.DOC A:CHAP*.BAK 命令中，XCOPY 将 CHAP1.DOC 拷贝到 A:CHAP1.BAK。

如果 source 中有通配符而 destination 只是一个文件，XCOPY 按顺序把每个文件拷贝到 destination 文件中，结束时 destination 是最后拷贝的文件。

如果 destination 指定的不是目录也不以\结尾，XCOPY 则询问用户 destination 是目录还是文件（若以\结尾，XCOPY 就知道 destination 是目录）。如果输入 D 以表明它是目录，XCOPY 就创建该目录并拷贝文件到其中。

带/S 选项时，XCOPY 从以 source 目录开始的整个目录分枝中选择文件，然后把它们拷贝到以 destination 目录开始的目录分枝中，需要时创建目的子目录。

使用/S 而没有/E 选项时，XCOPY 只在目的子目录中含有文件拷贝时才创建它。若有 /E 选项，XCOPY 复制源目录分枝中所有子目录到目的目录分枝，即使在某些新子目录中并不需要拷贝文件。

在 DOS 6.0 及以前版本中，不能没有/S 选项而只使用/E 选项。在 DOS 6.2 或更高版本中，可单独使用/E 选项。

如果是在批文件或 DOSKEY 宏中使用 XCOPY 在软盘上来复制拷贝文件，则可以用 /W 选项在 XCOPY 开始处理文件之前提供一段时间让用户在驱动器上插入磁盘。

在 DOS 6.2 之前的版本中，如果 destination 所指的文件已存在，XCOPY 会将其覆盖而不提示任何警告信息。这实际上意味着删除 destination 并生成一个同名文件。（用户可以使用第三方软件商提供的恢复被删除文件的实用程序恢复刚刚删除的文件）。

在 DOS 6.2 及更高版本中，如果 destination 所指文件已存在，XCOPY 的行为就取决于几种因素。在缺省情况下，如果该命令是从命令提示符下输入的，XCOPY 会询问用户是否覆盖目的文件，但如果是在批处理程序中执行这个命令，XCOPY 就不加任何警告而直接覆盖目的文件。另外，若环境变量 COPYCMD 被设成/Y，XCOPY 覆盖文件时也不加询问。若环境变量 COPYCMD 的值是/－Y，XCOPY 在覆盖文件前询问用户。可以通过在命令中包含/Y 或/－Y 来取消特定命令的缺省行为或环境变量 COPYCMD 的作用。

当 XCOPY 询问是否覆盖已存在的目的文件时，输入 Y 则覆盖文件，输入 N 则保留文件（不进行拷贝），输入 A 则覆盖本次及当前 XCOPY 命令涉及的后续文件。换句话说，输入 A 结束本次命令剩余部分的文件保护。

选项/Y 和/－Y 不能用于 DOS 6.2 以前的 DOS 版本。

XCOPY 设置的返回码如下：

0	成功完成拷贝操作。
1	未找到源文件。
2	用户按 Ctrl+C 键中断了 XCOPY。
4	初始化错误。内存或磁盘空间不够，输入了非法驱动器名，或命令行句法错误。
5	出现磁盘写错误。

应用举例

```
XCOPY J*.* A:\NEWDIR\K*.*
```

从当前目录中拷贝所有以 J 开头的文件（隐含、系统文件除外）到 A:\NEWDIR。每个目的文件名与源文件名除第一个字母换成 K 外均相同。若驱动器 A 上不存在 NEWDIR 目录，XCOPY 则创建该目录。

```
XCOPY D:\ACCOUNT E:\ /S
```

如果 D:\ACCOUNT 是子目录，从 D:\ACCOUNT 开始的目录分枝被拷贝到 E 盘根目录开始的目录分枝，需要时在 E:\下创建子目录。拷贝除系统和隐含文件外的所有文件。不在新的目录分枝内创建空子

目录。

如果 D:\ACCOUNT 是文件，那么 D 盘上每个名为 ACCOUNT 的文件(隐含和系统文件除外)均被拷贝到 E 盘上命名相同的目录中，需要时创建新的子目录(以容纳文件)。

```
XCOPY *.COM A:\ /E /W /Y
```

等待用户按任意键(使用户有时间在驱动器 A 中插入磁盘)。按键后，当前目录开始的目录分枝中的所有 COM 文件(系统和隐含文件除外)都被拷贝到 A 盘上命名相同的目录中，需要时在 A 盘上创建子目录。对于不含.COM 文件的源目录，如果 A 盘上不存在相应的目录，那么将创建这个。替换 A 盘上对应命名的文件而不给出任何警告信息(注意 DOS 6.2 之前的版本要求/S 选项与/E 选项同时使用并且不能用/Y 选项)。

```
XCOPY *.DAT \SAVE /A /S
```

当前目录开始的目录分枝中，每个有归档属性的.DAT 文件(隐含和系统文件除外)被拷贝(用原来的文件名)到以\SAVE 目录开始的目录分枝中命名相同的子目录中，需要时创建新的子目录。源文件的归档属性不变。

附录 A　设备驱动程序

ANSI. SYS

见第 48 章“控制台的控制”

提供控制屏幕颜色、光标位置和键盘定义的功能。

句法

DEVICE[HIGH]=[path] ANSI. SYS [/X|/K] [/R]

参数和选项

path　指定文件 ANSI. SYS 的位置，缺省时是启动盘的根目录。

/X　对于 101 键(扩展)键盘，强制 DOS 识别扩展光标控制键(灰色)、删除键和插入/覆盖键。在数字键盘也含有与它们功能相同的键，但具有不同的扫描码。

/K　使 ANSI. SYS 将 101 个键的键盘作为 84 个键的键盘来使用。如果有些程序不能识别 101 键盘上的扩展键时可用此选项。如果在 SWITCHES 命令中用了/K 选项，则在 ANSI. SYS 中也要使用/K 选项。

/R　稍微降低卷行速度使阅读容易些，这种效果几乎察觉不到。

注译

装入 ANSI. SYS 后，任何程序都可使用 ANSI. SYS 命令对应的换码序列来调用它们的功能(如果某个程序需要事先装入 ANSI. SYS，那么在该程序的文档中会有说明)。

还有其他输入 ANSI. SYS 换码序列的方法，可以包含在 PROMPT 命令中，放在 ASCII 文本文件中并用 TYPE 命令显示文件，或者插在批处理文件的 ECHO 命令中然后运行批处理程序。

所有 ANSI. SYS 换码序列都以换码符开始。Esc 键可输入换码符。但由于该键通常用于结束命令或程序，因此不能直接在命令提示符或文件中输入。然而，可以在 PROMPT 命令中敲入 $E 生成一个换码符。在 DOS 的 EDIT 和 WordStar 中，可通过按 Ctrl+P 再按 Esc 键输入一个换码符，在屏幕上显示成左箭头。在 Microsoft Word 中，可以在按住 Alt 键的同时在数字键盘上敲入 27 来输入换码符。其他编辑器和字处理器可能使用其他技术，有些则无法做到。

有几百个 ANSI. SYS 换码序列用于控制诸如屏幕文本和背景色，按 Ctrl+Shift+X 时所发生的动作等等。它们已被编入 DOS 6 的屏上帮助系统，有关内容可查找 ANSI. SYS 标题。

应用举例

```
DEVICEHIGH=C:\DOS\ANSI. SYS
```

把在 C:\DOS 下找到的 ANSI. SYS 设备驱动程序装入高端内存，若无足够的高端内存，则尽量装入常规内存。

```
PROMPT $E[34;57m$P$G
```

将命令提示符屏幕设置成在白色背景上显示的蓝色字符，并显示正常的命令提示符。换码序列 $E[34;57m 用于定义屏幕颜色。

CHKSTATE. SYS

由 MEMMAKER 命令用于控制优化过程。

注释

MEMMAKER 将 CHKSTATE. SYS 命令暂时加到 CONFIG. SYS 文件中。优化结束时，MEMMAKER 将其删除。

DBLSPACE.SYS

见第 34 章“DriveSpace 和 DoubleSpace”和第 45 章“再谈 DrvSpace 和 DblSpace”

将 DBLSPACE.BIN 移到内存中合适的位置。

句法

DEVICE[HIGH]=[path] DBLSPACE.SYS /MOVE[NOHMA]

参数和选项

path 指定 DBLSPACE.SYS 文件的位置。缺省时是启动盘的根目录。

NOHMA 防止 DBLSPACE 与 DOS 共享 HMA(高内存区)。

注释

DoubleSpace 将本命令加到 CONFIG.SYS 中。若系统设置成可访问 UMB(高端内存块),则用 DEVICEHIGH 替换 DEVICE 命令就可以将驱动程序装入高端内存。

DBLSPACE.BIN 负责管理压缩盘。DOS 启动期间,在 CONFIG.SYS 中的任何命令执行前将 DBLSPACE.BIN 装入常规内存最高处。这可能会与其他要求使用该区域的程序发生冲突。DBLSPACE.SYS 驱动程序将 DBLSPACE.BIN 移到内存中另一处以避免冲突。

在 DOS 6.2 及更高版本中,DBLSPAC 的一部分被装入 HMA,因而减少了对高端和低端内存的需求。这使得 HMA 中只剩下 10 个缓冲区的空间,如果使用 10 个以上的缓冲区,它们将全部被装入常规内存。如果正在使用磁盘缓冲(如 SmartDrive),那么将缓冲区减到 10 个。否则用/NOHMA 选项防止 DBLSPACE 与 DOS 共享 HMA。

应用举例

DEVICE=C:\DOS\DBLSPACE.SYS /MOVE

移动 DBLSPACE.BIN。

© DISPLAY.SYS

见第 49 章“处理另一个国家的信息”

支持屏幕和键盘代码页的切换。

句法

DEVICE [HIGH]=[path] DISPLAY.SYS CON=([type] [,codepage] [,max|,(max,sub)])

参数

path 指定 DISPLAY.SYS 文件的位置,缺省时是启动盘的根目录。

type 指定所使用的显示适配器。合法值是 EGA 和 LCD。EGA 包括 EGA 和 VGA 适配器。若不指定 type,DISPLAY.SYS 检查硬件以确定所使用的显示适配器。

codepage 指定硬件内建的代码页。

max 指定除硬件内建代码页外硬件可支持的代码页数。合法值对于 EGA 是 0 到 6,对 LCD 是 0 到 1。

sub 指定每个代码页硬件支持的辅助字库个数。

注释

关于代码页的更多信息参见 CHCP 和 MODE 命令。

虽然参数 type 接收值 CGA 和 MONO,但是这些值对 DISPLAY.SYS 不起作用,因为 CGA 和 MONO 适配器不支持多字符集。

DOS 支持的代码页值如下:

437 美国

850 多种语言(拉丁 I)

852　斯拉夫语(拉丁 II)
860　葡萄牙
863　加拿大法语
865　北欧

在美国出售的大部分硬件使用代码页 437。

如果在 CONFIG.SYS 中同时安装 DISPLAY.SYS 和第三方厂商提供的控制台驱动程序，那么首先应该安装第三方的控制台驱动程序。否则第三方的控制台驱动程序可能会被 DISPLAY.SYS 禁止。

应用举例

DEVICE=C:\DOS\DISPLAY.SYS CON=(EGA,437,(2,1))

为 EGA(或 VGA)控制台设备装入 DISPLAY.SYS，硬件代码页为 437，允许两个额外的代码页，每个带一个辅助字库。将 DISPLAY.SYS 装入常规内存。

DEVICEHIGH=C:\DOS\DISPLAY.SYS CON=(LCD,437,1)

为 LCD 控制台装入 DISPLAY.SYS，硬件代码页为 437，允许一个额外代码页。将 DISPLAY.SYS 装入高端内存，若无足够的高端内存则装入常规内存。

© DRIVER.SYS

见第 51 章“配置磁盘和网络”

生成一个逻辑驱动器，将其指派到软盘驱动器。

句法

DEVICE[HIGH]=[path] DRIVER.SYS /D:physical[/C] [/F:type] [/H:heads] [/S:sectors] [/T:tracks]

参数和选项

path	指定 DRIVER.SYS 文件的位置。缺省时是启动盘的根目录。
/D:physical	指定软盘驱动器编号。合法值是 0 到 127。
/C	说明驱动器带有换线(change-line)支持(关于换线支持的详细信息见 DRIVEPARM 命令)。
/F:type	指定驱动器类型(合法值见注释)。
/H:heads	指定磁盘驱动器的磁头数。合法值从 1 到 99，缺省值是 2。
/S:sectors	每个磁盘的扇区数。合法值从 1 到 99(缺省值见注释)。
/T:tracks	每面磁道数。合法值从 1 到 999(缺省值见注释)。

注释

驱动器名 A 总被指定给物理驱动器 0。如果系统中有第二个驱动器，它就是物理驱动器 1，并被指定驱动器 B。否则，驱动器名 B 也被指定给物理驱动器 0。

如果系统中安装了一个软盘驱动器而 DOS 不能识别它，那么使用 DRIVER.SYS 为其指定一个驱动器名。如果这是系统中第三个软盘驱动器，则编号为 2，如果是第四个，则编号为 3，等等。

若想为同一个驱动器指定两个驱动器名，使用 DRIVER.SYS 指定第二个驱动器名。这样做的原因之一是可以用 COPY 或 XCOPY 命令在同一驱动器中从一张磁盘向另一张拷贝文件。

DOS 将下一个可用的驱动器名指派给 DRIVER.SYS 定义的驱动器。如果在定义了 RAM 驱动器的 CONFIG.SYS 中加入 DEVICE=DRIVER.SYS 命令，DOS 可能为 RAM 驱动器指定与从前不同的驱动器名。在 CONFIG.SYS 中加入 DRIVER.SYS 还有可能改变由 DoubleSpace 指定的驱动器名。

扇区和磁道数的缺省值取决于驱动器类型(type)。合法的类型及每种类型的缺省值如下：

类型	驱动器	扇区数	磁道数
0	160KB/180KB 或 320KB/360KB	9	40
1	1.2MB	15	80
2	720KB 或其他(缺省)	9	80
7	1.44MB	18	80
9	2.88MB	36	80

若指定选项/H、/S 和/T,可以忽略/F 选项。通常,指定/F 时可以忽略/H、/S 和/T,但要查阅磁盘驱动器文档以保证这些选项的缺省值适合于用户的驱动器。

如果用一个物理驱动器生成两个逻辑驱动器,要保证它们有相同的参数和选项。

应用举例

```
DEVICEHIGH=C:\DOS\DRIVER.SYS /D:1 /C
```

若有可能,将 DRIVER.SYS 装入高端内存,否则装入常规内存。把下一个可用驱动器名指派给第二个软盘驱动器,使用适合于 3 ½英寸 720KB 磁盘驱动器的(缺省)值,并指定驱动器具有换线支持。如果在启动期间 DOS 已经为第二个软驱指定了名字,那么本命令为驱动器额外指定一个名字。

```
DEVICE=C:\DOS\DRIVER.SYS /D:2 /F:7
DEVICE=C:\DOS\DRIVER.SYS /D:2 /F:7
```

上面这两个命令都装入 DRIVER.SYS 文件并为第三个软驱指派下一个可用驱动器名,使用适合于 3 ½英寸 1.44MB 软驱的(缺省)值。其效果是为驱动器指派两个驱动器名。

DRVSPACE.SYS

见第 34 章和第 45 章

把 DRVSPACE.BIN 移到内存中合适的位置。

句法

```
DEVICE[HIGH]=[path] DRVSPACE.SYS /MOVE [NOHMA]
```

参数和选项

path 指定文件 DRVSPACE.SYS 的位置,缺省时是启动盘的根目录。

NOHMA 防止 DRVSPACE 与 DOS 共享 HMA。

注释

在安装 DriveSpace 时它会将本命令加到 CONFIG.SYS 中。如果系统设置成可访问 UMB,可以用 DEVICEHIGH 替换 DEVICE 命令而把驱动程序装入高端内存。

DRVSPACE.BIN 负责管理压缩盘。启动时,DOS 在处理 CONFIG.SYS 中的任何命令前将 DRVSPACE.BIN 装入常规内存最高端。这可能会与其他要求使用此区域的程序发生冲突。DRVSPACE.SYS 驱动程序将 DRVSPACE.BIN 移到内存中其他地方以避免冲突。

正常情况下,DRVSPACE 的一部分被装入 HMA 以减少对高端和低端内存的需求。这使得 HMA 中只剩下 10 个缓冲区的空间。若使用 10 个以上的缓冲区,它们会被全部装入常规内存。如果正在使用磁盘缓冲程序如 SmartDrive,那么将缓冲区数量减到 10。否则,使用/NOHMA 选项防止 DRVSPACE 与 DOS 共享 HMA。

应用举例

```
DEVICE=C:\DOS\DRVSPACE.SYS /MOVE
```

移动 DRVSPACE.BIN。

EGA.SYS

见第 21 章"系统配置"

在 EGA 显示器上使用 DOS Shell 的任务交换程序时用于保存和恢复显示。

句法

DEVICE[HIGH]=[path] EGA.SYS

参数

path　　指定文件 EGA.SYS 的位置。缺省时是启动盘的根目录。

注释

如果用户使用的是 EGA 显示器，在使用 DOS Shell 的任务交换程序之前必须安装 EGA.SYS。Microsoft Windows 或其他任何切换程序也可能要求有 EGA.SYS。

如果在有 EGA 显示器的系统上使用鼠标，在安装鼠标驱动程序之前安装 EGA.SYS 可以节省内存。

© EMM386.EXE

见第 30 章“为 386、486 和奔腾机增加的内存工具”和第 44 章“腾出更多的内存空间”

提供对高端内存区的访问，用扩展内存(extended memory)到模拟扩充内存(expanded memory)。

句法

DEVICE=[path] EMM386.EXE[ON|OFF|AUTO] [expanded-memory] [MIN=size] [W=ON|W=OFF] [Mx|FRAME=address1|/Paddress1] [Pn=address2] [[X=mmmm1-nnnn1]...] [[I=mmmm2-nnnn2]...] [B=address3] [L=minXMS] [A=altregs] [H=handles] [D=nnn] [RAM[=mmmm3-nnnn3]] [NOEMS] [NOVCPI] [HIGHSCAN] [VERBOSE] [[WIN=mmmm4-nnnn4]...] [NOHI] [ROM=mmmm5-nnnn5] [NOMOVEXBDAT] [ALTBOOT]

参数和选项

path	指定 EMM386.EXE 文件的位置。缺省时是启动盘的根目录。
ON|OFF|AUTO	参数 ON 装入并激活 EMM386 驱动程序。OFF 装入驱动程序，但使其无效。AUTO 把驱动程序置为自动模式，只有在程序要求时才激活对扩充内存(expanded memory)和高端内存的支持。驱动程序装入后可以使用 EMM386 命令来改变这一值。
expanded-memory	以千字节为单位，指定 EMM386 用于提供扩充/虚拟控制程序界面(EMS/VCPI)内存的扩展内存(extended memory)的最大数量。其值从 64 到 32768(或空闲的扩展内存总量)。除非指定 NOEMS，缺省值是空闲扩展内存总量，EMM386 将这个值舍入为最邻近的 16 的倍数。
MIN=size	以千字节为单位指定 EMM386 提供的最小 EMS/VCPI 内存数量。其值从 0 到 expanded-memory。缺省值是 256，除非指定 NOEMS。
W=ON|W=OFF	允许或禁止支持 Weitek 协处理器。缺省是 W=OFF。
Mx	指定扩充内存页面的基地址。合法值从 1 到 14。如果计算机的内存少于 512KB，x 必须小于 10。与每个 x 值对应的地址值见注释。
FRAME=address1	直接指定页面基地址。合法值从 8000H 到 9000H 和 C000H 到 E000H，每步增加 400H。可以指定 FRAME=NONE，但这可能导致某些程序不能正常运行。
/Paddress1	直接指定页面基地址。合法值从 8000H 到 9000H 和 C000H 到 E000H，每步增加 400H。
Pn=address2	指定页面 n 的基地址。n 的合法值从 0 到 255，地址的合法值是从 8000H 到 9C00H 和 C000H 到 EC00H，每步增加 400H。为与 LIM EMS 版本 3.2 兼容，页面 0 到 3 的地址必须连续。如果使用了 Mx、FRAME 或/Paddress1，就不能用/Pn 选项指定页面 0 到 3 的地址。
X=mmmm1-nnnn1	防止 EMM386.EXE 为 EMS 页面或 UMB 使用特定范围的段地址。mmmm1

和 nnnn1 的合法值范围是从 A000H 到 FFFFH，并被舍入到最接近的 4KB 边界。如果参数范围与 I 参数重叠，本参数优先。

I=mmmm2-nnnn2 指定用于 EMS 页与 UMB 的段地址范围。mmmm2 和 nnnn2 的合法值范围是 A000H 到 FFFFH，并被舍入到最接近的 4KB 边界。如果与参数 X 范围重叠，参数 X 优先。

B=address3 指定用于 EMS 存储切换(bank-switching)的最低段地址。合法值是 1000H 到 4000H，缺省值是 4000H。

L=minXMS 保证在装入 EMM386.EXE 之后仍有 min 千字节的扩展内存可用。缺省值是 0。

A=altregs 指定为 EMM386.EXE 分配多少快速交换寄存器组(用于多任务)。合法值从 0 到 254。缺省值是 7。每个交换寄存器组占用 EMM386.EXE 大约 200 个字节的内存空间。

H=handles 指定 EMM386.EXE 可以使用多少个句柄。合法值从 2 到 255，缺省值是 64。

D=nnn 指定为带缓冲的直接内存访问(DMA)保留多少千字节的内存。不考虑软盘 DMA，此值应反映 EMM386 活动时的最大 DMA 传输量。nnn 的合法值是 16 到 256，缺省值是 16。

RAM[=mmmm3-nnnn3] 同时要求扩充内存和 UMB 支持，并在扩展内存中指定一个段地址范围用于 UMB。若不指定范围，EMM386.EXE 自己选择地址。

NOEMS 提供 UMB 支持，但不提供扩充内存。

NOVCPI 禁止支持 VCPI(虚拟控制程序界面)应用程序。本选项只与 NOEMS 选项同时使用。当同时使用这两个选项时，EMM386.EXE 忽略 expanded-memory 和 MIN 参数。

HIGHSCAN 在高端内存中进行额外的扫描以寻找可用内存。在某些计算机上，本选项没有作用，甚至造成停机。

VERBOSE 使 EMM386.EXE 在装入时显示状态和错误信息。缺省情况下，EMM386 不显示这些信息。可将/VERBOSE 缩写为/V。

WIN=mmmm4-nnnn4 为 Windows(而不是 EMM386.EXE)保留某一范围的段地址。mmmm4 和 nnnn4 的合法值从 A000H 到 FFFFH，并被舍入到最接近的 4KB 边界。若与参数 X 发生重叠，参数 X 优先，而参数 WIN 优先于 RAM、ROM 和 I 参数。

NOHI 防止 EMM386.EXE 装入高端内存。正常情况下，EMM386.EXE 的一部分装入高端内存。使用本选项减少了可用的常规内存而增加了可用于 UMB 的高端内存。

ROM=mmmm5-nnnn5 指定 EMM386.EXE 用于影象(shadow)RAM 的段地址范围。mmmm5 和 nnnn5 的合法值从 A000H 到 FFFFH，并被舍入到最接近的 4KB 边界。如果尚未使用影象 RAM，本选项可使系统加速。(对影象 RAM 的解释参见 HIMEM.SYS)。

NOMOVEXBDAT 防止 EMM386.EXE 从常规内存移动扩展 BIOS 数据到高端内存。

ALTBOOT 通知 EMM386.EXE 在用户按 CTRL+ALT+DEL 键时使用不同的方法重启动。如果装入 EMM386.EXE 后在重启动时出现问题，使用本选项。

注释

必须在 CONFIG.SYS 中使用 DEVICE(不能用 DEVICEHIGH)命令装入此设置驱动程序。只能在装有 386 或更高档的处理器的计算机上才能使用此设备驱动程序。

装入 EMM386.SYS 之前必须安装 HIMEM.SYS。在 CONFIG.SYS 中装入 EMM386 的命令必须在所

有 DEVICEHIGH 命令之前。

EMM386 可以通过用扩展内存(extended memory)到模拟扩充内存(expanded memory)来为程序提供扩充内存。为此目的使用 EMM386 时,通常无需指定任何内存参数,EMM386 在正常情况下可以用缺省参数运行。为此目的保留的扩展内存(由 MIN 参数指定或缺省值)不能再作为扩展内存使用。如果 EMM386 使用了额外的扩展内存,最多到 expanded-memory,这部分内存在不需要再作为扩充内存时恢复成扩展内存。

要提供对高端内存的访问,必须使用 RAM 或 NOEMS 选项。(而且必须在 CONFIG. SYS 中包括 DOS=UMB 命令)。选项 NOEMS 提供对高端内存的访问,同时取消了对扩充内存的支持。选项 RAM 同时提供对高端内存和扩充内存的支持。

当同时使用 EMM386. EXE 和 Windows 3. 1 时,参数 I、X、NOEMS、Mx、Paddress1 和 FRAME 优先于 Windows 的 SYSTEM. INI 文件中的 EMMINCLUDE、EMMEXCLUDE 和 EMMPAGEFRAME 设置。

如果系统中有小型计算机系统接口(SCSI)或增强型系统设备接口(ESDI)的硬盘或其他设置,可能需要在使用 EMM386 时,使用 SmartDrive 双缓冲(见 SMARTDRV. EXE 命令)。

参数 Mx 的值和相关的基地址如下:

x	基地址
1	C000H
2	C400H
3	C800H
4	CC00H
5	D000H
6	D400H
7	D800H
8	DC00H
9	E000H
10	8000H
11	8400H
12	8800H
13	8C00H
14	9000H

应用举例

```
DEVICE=C:\DOS\HIMEM. SYS
DEVICE=C:\DOS\EMM386. EXE OFF
```

第一个命令使用所有缺省值装入 HIMEM. SYS 扩展内存管理器。这是第二个命令的先决条件。第二个命令装入 EMM386. EXE 设备驱动程序,借用扩展内存(extended memory)提供最少 256KB 的扩充内存(expanded memory)。额外的扩充内存在程序要求时分配,最多可达扩展内存总量。所有其他参数和页面地址、交换寄存器和句柄等使用缺省值。驱动程序在装入后被禁止,所以直到一个 EMM386 命令激活它或置为自动模式之前均不提供扩充内存。

```
DEVICE=C:\DOS\HIMEM. SYS
DEVICE=C:\DOS\EMM386. EXE NOEMS
DOS=UMB
```

第一个命令使用所有缺省值装入 HIMEM. SYS 扩展内存管理器。这是第二个命令的先决条件。第二个命令装入 EMM386. EXE 设备驱动程序并且只提供高端内存。(用于 UMB 的内存借用扩展内存)。为使 DOS 可以使用 UMB,第三个命令也是必需的。

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE MIN=1024 2048 RAM
DOS=UMB
```

第一个命令使用所有缺省值 HIMEM.SYS 扩展内存管理器。这是第二个命令的先决条件。第二个命令装入 EMM386.EXE 设备驱动程序提供对扩充内存和高端内存支持。至少从扩展内存中借用 1MB 的扩充内存 。可应个别程序要求扩充至 2MB。所有其他扩充参数使用缺省值。第三个命令使 DOS 可以使用 UMB。

© HIMEM.SYS

见第 29 章"内存管理"和第 44 章

管理扩展内存,包括高内存区(HMA)。

句法

```
DEVICE=[path]HIMEM.SYS[/A20CONTROL:ON|OFF] [/CPUCLOCK:ON|OFF]
[/EISA][/HMAMIN=size][/INT15=x][/NUMHANDLES=n][/MACHINE:machine]
[/SHADOWRAM:ON|OFF][/TESTMEM:[ON|OFF][/VERBOSE]
```

参数和选项

path	指定文件 HIMEM.SYS 的位置。缺省时是启动盘的根目录。
/A20CONTROL:ON|OFF	指定 HIMEM 是否控制 A20 地址线,即使 A20 已被使用。若指定/A20CONTROL:OFF,HIMEM 只有在它装入时 A20 地址线未被使用的情况下才控制 A20。缺省时是/A20CONTROL:ON。
/CPUCLOCK:ON|OFF	指定 HIMEM 是否可以影响计算机的时钟速度。如果在装入 HIMEM 时时钟速度改变,/CPUCLOCK:ON 可以改正这个问题但会减慢 HIMEM。缺省设置是/CPUCLOCK:OFF。
/EISA	指定 HIMEM.SYS 应分配所有可用的扩展内存。只有在带有超过 16MB 内存的 EISA(扩展工业标准结构)计算机上才有必要用此选项。在其他计算机上,HIMEM 自动分配所有可用内存。
/HMAMIN=size	以千字节为单位指定程序必须使用 HMA 至少有多大。合法值从 0 到 63。缺省是 0,它强制 HIMEM 把 HMA 分配给第一个发出请求的程序。当 Windows 运行在 386 增强模式时本选项丢失作用。
/INT15=x	以千字节为单位分配扩展内存给中断 15h 接口。合法值从 64 到 65535,但不能超过系统可用内存的数量。缺省值是 0。若 x 小于 64,HIMEM 使用 0。
/NUMHANDLES=n	指定可同时使用的扩展内存块句柄的最大值。合法值从 1 到 128。缺省值是 32。每个句柄要求 6 字节内存。当 Windows 运行在 386 增强模式下时本选项不起作用。
/MACHINE:machine	使用哪个 A20 控制器。合法值见注释。
/SHADOWRAM:ON|OFF	是否允许影象(shadow)RAM。如果计算机的内存少于 2MB,缺省设置是 OFF(禁止)。
/TESTMEM:ON|OFF	启动时是否检查高内存区。缺省设置是 ON。
/VERBOSE	使 HIMEM.SYS 在装入时显示状态和错误信息。在缺省情况下,HIMEM.SYS 不显示这些信息。该选项可缩写为/V。

注释

必须在 CONFIG.SYS 中使用 DEVICE(而不是 DEVICEHIGH)命令装入该设备驱动程序。这个 DEVICE 命令必须用于任何启动时需要扩展内存的应用程序或设备驱动程序的命令之前。例如,必须在装入

EMM386.EXE 前装入 HIMEM.SYS。

大部分情况下，不需要指定命令行选项。HIMEM.SYS 的缺省值可以适应大部分硬件。

在 286 或更新的机器中，第 21 根地址线（称为 A20）使 DOS 可以访问高内存区（HMA）。HMA 是扩展内存的第一个 64KB。HMA 可以像常规内存一样用于执行程序，但必须装入特别的软件来管理 A20 地址线。

通常，HIMEM 能够成功地检测出计算机类型并装入正确的 A20 控制程序，但也有几种计算机 HIMEM 检测不出。在这样的系统中，HIMEM 使用缺省的 A20 控制程序（#1，用于 IBM AT 或兼容机）。如果 HIMEM 不能正常工作，可能需要使用/MACHINE 选项指定 A20 控制程序。需要此选项的系统包括 Acer 1100，Wyse 和 IBM 7552。

使用/MACHINE 选项时，要么指定机器代码，如/MACHINE＝WYSE，要么指定 A20 控制程序号，如/MACHINE＝8。

可以使用的机器代码和对应的 A20 控制程序号如下：

机器代码	A20 控制程序号	计算机类型
AT	1	IBM AT or 100%兼容
PS2	2	IBM PS/2
PTLCASCADE	3	Phoenix Cascade BIOS
HPVECTRA	4	HP Vectra(A & A+)
ATT6300PLUS	5	AT&T6300 Plus
ACER1100	6	Acer 1100
TOSHIBA	7	Toshiba 1600 & 1200XE
WYSE	8	Wyse 12.5 MHz 286
TULIP	9	Tulip SX
ZENITH	10	Zenith ZBIOS
AT1	11	IBM PC/AT(alternative delay)
AT2	12	IBM PC/AT(alternative delay)
CSS	12	CSS Labs
AT3	13	IBM PC/AT (alternative delay)
PHILIPS	13	Philips
FASTHP	14	HP Vectra
IBM7552	15	IBM 7552 工业计算机
BULLMICRAL	16	Bull Micral 60
DELL	17	Dell XBIOS

某些计算机在启动时通过将 ROM 信息拷贝到 RAM 中来生成 ROM 的影象（shadow）。这可使 ROM 代码运行得快一些，但会占用扩展内存。如果计算机的内存少于 2MB，HIMEM 通常会试图禁止影象内存而恢复这部分扩展内存作其他用处。（HIMEM 只在某些系统上能够禁止影象内存）。在 HIMEM 禁止影象内存后，计算机可能会比以前运行得略微慢一些。（在本身不提供影象内存的计算机上建立影象的方法参见 EMM386.EXE。）

某些旧的应用程序分配扩展内存时使用中断 15H 接口，而不是 HIMEM 提供的 XMS（扩展内存规范）方法。如果使用这样的应用程序，可以将参数 x 设成比程序所需大 64KB 的内存，以保证程序有足够的内存可用。

正常情况下，HIMEM 在 CONFIG.SYS 启动时测试扩展内存（extended memory）。为了节省时间，可使用/TESTMEM:OFF 防止这一检测。如果在 HIMEM 装入时关掉了内存检测，仍然可以在 CONFIG.SYS

启动时按住 ALT 键进行这一检测。

应用举例

DEVICE=C:\DOS\HIMEM.SYS

使用所有缺省值装入 HIMEM.SYS。

DEVICE=C:\DOS\HIMEM.SYS /MACHINE=6 /INT15=64

装入 HIMEM.SYS 驱动程序,使用 Acer 1100 的 A20 控制程序,为旧式扩展内存请求保留 64KB 内存。

© INTERLNK.EXE

见第 41 章"INTERLNK"

连接客户计算机到服务器计算机并重定向驱动器和打印机端口名。

句法

DEVICE[HIGH]=[path] INTERLNK.EXE [/DRIVES:n] [/NOPRINTER]
[/COM[n|address]] [/LPT[n|address]] [/AUTO] [/NOSCAN] [/LOW] [BAUD:rate] [/V]

参数和选项

path	指定文件 INTERLNK.EXE 的位置。缺省时是启动盘的根目录。
/DRIVES:n	指定被重定向到客户计算机的服务器一方的驱动器数。缺省值是 3。若 n=0,INTERLNK 只重定向打印机。
/NOPRINTER	禁止重定向打印机。在缺省情况下,INTERLNK 重定向所有可用的并行打印端口。
/COM[n|address]	指定客户计算机用于连接服务器计算机的端口。若 n 和 address 都未指定,INTERLNK 为建立连接扫描所有串行端口。
/LPT[n|address]	指定客户计算机用于连接服务器计算机的端口。若 n 和 address 都未指定,INTERLNK 为建立连接扫描所有并行端口。
/AUTO	只有立即与服务器计算机建立连接时才安装 INTERLNK.EXE 到内存。在缺省情况下,即使尚不能完成连接也安装 INTERLNK.EXE。
/NOSCAN	把 INTERLNK.EXE 装入内存,但防止它建立连接。在缺省情况下,只要安装 INTERLNK,它就试图与服务器计算机建立连接。
/LOW	即使有可用的高端内存,也把 INTERLNK.EXE 装入常规内存。在缺省情况下,INTERLNK.EXE 在可能时装入高端内存。
/BAUD:rate	为串行通讯设置最大的波特率。合法值是 9 600、19 200、38 400、57 600 和 115 200。缺省值 115 200。
/V	防止与计算机的定时器发生冲突。如果在计算机间使用串行连接,而在用 INTERLNK 访问驱动器或打印端口时其中一台停止运行,可用此选项。

注释

如果两台计算机已被电缆连接并且在服务器计算机上已装入 INTERSVR 程序,用 INTERLNK 可以建立连接并重定向服务器的驱动器和打印端口。否则,需要在达到这些要求后使用 INTERLNK 命令。

若未指定/LPT 和/COM,客户计算机扫描所有串行端口和并行端口。

警告: 扫描鼠标所接的串行端口会禁止鼠标驱动程序。如果正在使用串行鼠标的话,那么指定/LPT(而不指定/COM),或者指定/COMn,其中 COMn 不应该是鼠标使用的端口。

装入 INTERLNK.EXE 的 DEVICE 命令的位置会影响以后的驱动器名的指定。为保证在 CONFIG.SYS 中加入 DEVICE 命令后所有驱动器被指定为与以前相同的驱动器名,一定要把装入 INTERLNK.EXE 的 DEVICE 命令作为 CONFIG.SYS 中的最后一个命令。

可以使用选项只装入 INTERLNK.EXE 中需要的部分来节省内存空间。若指定/NOPRINTER，INTERLNK.EXE 则不装入处理打印的代码。若指定/LPT，该程序则不装入支持串行端口的代码。若指定/COM，该程序则不装入支持并行端口的代码。

如果重定向了 LPT1 或 LPT2 并在 Microsoft Windows 中打印，使用 Control Panel 将打印机赋予 LPT1.DOS 或 LPT2.DOS。

如果在服务器和客户机之间运行不同的 DOS 版本，客户计算机可能无法使用 DOS 的某些功能。例如，如果在服务器计算机上有大分区而在客户机运行的是 DOS 3.0，则客户机无法访问这些大分区，因为 DOS 3.0 不支持大分区。

如果用 INTERLNK 运行位于服务器计算机上的应用程序，要保证应用程序的参数是为客户机设置的。

以下命令不能用于 INTERLNK.EXE 设备驱动程序：CHKDSK、DEFRAG、DISKCOMP、DISKCOPY、FDISK、FORMAT、SYS、UNDELETE 和 UNFORMAT（以及其他任何跳过正常的 DOS 文件处理过程而直接读写系统区、磁道或扇区的程序）。

应用举例

```
DEVICE=C:\DOS\INTERLNK.EXE /COM2
```

为客户计算机装入 INTERLNK.EXE，使用 COM2 端口。若能建立连接，重定向服务器的所有并行打印端口和三个驱动器。

```
DEVICE=C:\DOS\INTERLNK.EXE /DRIVES:5 /NOPRINTER
```

装入 INTERLNK.EXE 设备驱动程序，为建立连接扫描客户计算机的所有串行端口和并行端口。若能建立连接，重定向五个驱动器，不包括打印机。

© POWER.EXE

见第 48 章

在膝上型计算机中，当应用程序和设备空闲时减少电源消耗。

句法

```
DEVICE[HIGH]=[path] POWER.EXE [ADV[:advtype]|STD|OFF] [/LOW]
```

参数和选项

path	指定文件 POWER.EXE 的位置。缺省时是启动盘的根目录。
ADV[:advtype]	当应用程序和硬件设备空闲时保存电源。advtype 可以是 MAX，代表最大程度地节省电源，REG 代表在应用程序及设备通用性和电源节省之间作出平衡，而 MIN 代表最小程度地节省电源。如果使用 REG 或 MAX 时系统性能不能令人满意，可以使用 MIN 选项。缺省设置是 ADV:REG。
STD	如果用户的计算机支持先进电源管理（APM）规范，STD 只通过使用计算机的电源管理功能来节省电源。如果计算机不支持 APM 规范，则 STD 关闭电源管理功能。
OFF	关闭电源管理功能。

注释

如果可能，POWER.EXE 装入高端内存。否则装入常规内存。

电源管理程序符合 APM 规范。

关于电源管理的详细信息参见 POWER 命令。

应用举例

```
DEVICE=C:\DOS\POWER.EXE
```

将 C:\DOS 下的 POWER.EXE 装入高端内存，若无足够的高端内存，则装入常规内存。使用缺省设置

ADV:REG。

© RAMDRIVE.SYS

见第 32 章“RAM 驱动器”

使用计算机的一部分随机存储器(RAM)模拟硬盘驱动器。

句法

DEVICE[HIGH]=[path] RAMDRIVE.SYS [DiskSize[SectorSize[NumEntries]]] [/E|/A]

参数和选项

path	指定文件 RAMDRIVE.SYS 的位置。缺省时是启动盘的根目录。
DiskSize	以千字节为单位指定用于 RAM 驱动器的内存数量。合法值是 4 到 32767,或较小的可用内存量。缺省值是 64。
SectorSize	以字节为单位指定磁盘扇区大小。可以是 128、256 或 512(缺省值)。Microsoft 建议使用 512。如果使用 SectorSize 参数,必须用 DiskSize 参数。
NumEntries	指定可以在 RAM 盘根目录下生成的文件和目录数。可从 2 到 1024。用户指定的限制被增加到最接近的扇区容量边界。缺省值是 64。如果包括 NumEntries 参数,同时还要指定 SectorSize 和 DiskSize 参数。
/E	在扩展内存(extended memory)中创建 RAM 盘。
/A	在扩充内存(expanded memory)中创建 RAM 盘。

注释

DOS 为 RAM 盘分配下一个可用的驱动器符。

RAM 盘比硬盘快得多,软驱更是不能相比,因为计算机访问内存的速度要比访问磁盘快。但是当关掉电源或重新启动计算机时会丢掉 RAM 盘中的所有数据。RAM 盘尤其适于存放临时文件。很多人把环境变量 TEMP 的值设为 RAM 盘上的某个目录,一是为了加快速度,二是为了避免临时数据可能装满硬盘。

如果可能,最好在扩展内存中创建 RAM 盘。

要使用/A 选项,系统中要有扩充内存,必须在创建 RAM 盘之前装入扩充管理器,如 EMM386。

若忽略/E 和/A 选项,RAM 盘则使用常规内存,只是在没有扩展、扩充内存,也没有硬盘的系统中才推荐这种用法。

如果使用 Windows 并将 TEMP 设置成 RAM 盘上的某个目录,要保证 DiskSize 至少有 2048KB(2MB)以使 Windows 有足够的空间存储临时打印文件。

如果没有足够的空间用于创建指定的 RAM 驱动器,RAMDRIVE.SYS 会试图在根目录下创建只能有 16 个目录项的 RAM 驱动器。

新 RAM 盘的名字在启动信息中显示。

应用举例

```
DEVICEHIGH=C:\DOS\RAMDRIVE.SYS 2048 /E
```

将 RAMDRIVE.SYS 装入高端内存,若无足够的高端内存,则装入常规内存。在扩展内存中创建一个 2048KB(2MB)的 RAM 盘,每扇区 512 字节,根目录下最多可有 64 个目录项。

```
DEVICE=C:\DOS\RAMDRIVE.SYS 2048 512 128 /A
```

装入 RAMDRIVE.SYS 文件并在扩充内存中创建一个 2048KB(2MB)的 RAM 盘,每扇区 512 字节,根目录下最多可有 128 个目录项。

© SETVER.EXE

见第 21 章

将 DOS 版本表装入内存。

句法

DEVICE[HIGH]=[path] SETVER.EXE

参数

path　　指定文件 SETVER.EXE 的位置。缺省时是启动盘的根目录。

注释

SETVER.EXE 设备驱动程序使用版本表向请求版本号信息的程序报告适当的版本号。命令 SETVER 允许显示和修改版本表。关于版本表更详细的信息参见 SETVER 命令。

如果要用 SETVER 向某个设备驱动程序报告不同的版本号，装入 SETVER 的 DEVICE(或 DEVICEHIGH)命令在 CONFIG.SYS 中必须出现在装入那个设备驱动程序的 DEVICE(或 DEVICEHIGH)命令之前。

应用举例

DEVICEHIGH=C:\DOS\SETVER.EXE

将 C:\DOS 下的 SETVER.EXE 文件装入高端内存，若无足够高端内存，则装入常规内存。

SIZER.SYS

由 MEMMAKER 用于确定内存中的设备驱动程序和 TSR 的大小。

注释

MEMMAKER 在 CONFIG.SYS 和 AUTOEXEC.BAT 中临时加入 SIZER.SYS。优化结束时，MEMMAKER 删除 SIZER.SYS 命令。

© SMARTDRV.EXE

见第 31 章“驱动器缓存”

为不能在由 EMM386.EXE 或运行在 386 增强模式的 Windows 所提供的内存上工作的硬盘控制器提供兼容性。

句法

DEVICE=[path] SMARTDRV.EXE /DOUBLE_BUFFER

参数

path　　指定文件 SMARTDRV.EXE 的位置，缺省时是启动盘的根目录。

注释

若未使用 EMM386.EXE，也不在 386 增强模式下运行 Windows，则不需要双缓冲。

当系统中有 SCSI(小型计算机系统接口)设备，ESDI(增强型系统接口)设备或 MCA(微通道结构)设备时，如果使用 EMM386.EXE 或在 386 增强模式下运行 Windows，可能需要使用双缓冲。

当为了双缓冲装入这个设备驱动程序时，它大概占用 2KB 常规内存。

驱动程序 SMARTDRV.EXE 不提供磁盘缓冲，进行缓冲应使用 SMARTDRV 命令。按照以下步骤确定用户的系统是否需要双缓冲(系统应设置成可以使用高端内存块)。

1. 装入 SMARTDRV.EXE 设备驱动程序用于双缓冲。
2. 如果 AUTOEXEC.BAT 文件中没有 SMARTDRV 命令，则加入一个。在使用 SMARTDRV 后，如果系统运行变慢，在 SMARTDRV 命令中加入/L 选项。
3. 运行 MEMMAKER。
4. 在命令提示符下输入 SMARTDRV，显示关于系统的信息。
5. 查看题目为 Buffering 的列，若此列中任一行为 yes，则说明需要双缓冲。
6. 若 Buffering 列中每一行都是 no，则不需要双缓冲，可以将这个命令从 CONFIG.SYS 中去掉。
7. 若 Buffering 中某一行是－，说明 SmartDrive 无法测出是否需要双缓冲。

附录B 返回码

简介： 很多DOS命令在结束时设置返回码。只有当该命令是作为批处理程序的一部分运行时才可以访问返回码。大部分情况下，返回码为0表示命令正确使用，操作成功完成。任何其他返回码表示操作没有正确完成。

命令IF ERRORLEVEL使用户可以检查返回码并根据它采取行动。第37章"批处理程序中的编程逻辑"解释了如何在批处理程序中使用返回码。

命令	返回码	含义
DEFRAG	0	消除碎块成功。
	1	内部错误。
	2	没有空闲簇(DEFRAG要求一个空闲簇)。
	3	用户按Ctrl+C键停止。
	4	普通错误。
	5	读簇错误。
	6	写簇错误。
	7	分配错误(使用CHKDSK /F)。
	8	内存错误。
	9	内存不够。
DISKCOMP	0	磁盘相同。正常退出。
	1	磁盘不同。
	2	用户按Ctrl+C键中止命令。
	3	严重错误。
	4	初始化错误。
DISKCOPY	0	磁盘拷贝成功，正常返回。
	1	执行不成功，拷贝时出现非致命读写错误。
	2	用户按Ctrl+C键中止命令。
	3	硬件致命错。
	4	初始化错误。
FIND	0	查找成功，至少找到一处匹配。
	1	成功完成，但未找到匹配。
	2	没有成功完成，FIND无法报告是否找到匹配。
FORMAT	0	磁盘格式化成功，正常退出。
	3	用户按Ctrl+C键中止命令。
	4	出错，程序中止。
	5	用户响应Proceed with Format (Y/N)时，按了N键。
KEYB	0	装入键盘文件，正常退出。
	1	错误的键盘代码、代码页值或句法。

（续表）

命令	返回码	含义
	2	键盘定义文件错误或未找到。
	4	CON 设备错误。
	5	未准备好要求的代码页。
MSAV	86	未使用图形界面，并且检测到病毒。
REPLACE	0	替换或增加指定的文件，正常退出。
	1	DOS 版本与 REPLACE 不兼容。
	2	未进行文件操作，找不到源文件。
	3	未进行文件操作，找不到源或目的路径。
	5	拒绝读/写访问，至少一个文件未被替换。使用 ATTRIB 改变权限。
	8	命令未执行，没有足够的内存运行命令。
	11	命令未执行，参数错误或格式非法。
RESTORE	0	文件被恢复，正常退出。
	1	找不到要恢复的文件。
	3	用户按 Ctrl+C 键中止命令。
	4	出现错误，命令中断。
SETVER	0	任务完成，正常退出。
	1	未执行命令，选项非法。
	2	未执行命令，文件名非法。
	3	未执行命令，内存不够。
	4	未执行命令，版本号格式错误。
	5	未执行命令，版本表中的表项找不到。
	6	未执行命令，找不到 SETVER.EXE 文件。
	7	未执行命令，驱动器非法。
	8	未执行命令，参数太多。
	9	未执行命令，缺少参数。
	10	未执行命令，读 SETVER.EXE 文件出错。
	11	未执行命令，文件 SETVER.EXE 中的版本表已损坏。
	12	未执行命令，SETVER.EXE 不支持版本表。
	13	未执行命令，版本表中没有剩余空间。
	14	未执行命令，写版本表出错。
XCOPY	0	拷贝指定文件，正常返回。
	1	找不到要拷贝的文件。
	2	用户按 Ctrl+C 键中止命令。
	4	内存或磁盘空间不够，句法错，或初始化错误。
	5	磁盘写保护或其他写错误。

附录C IBM扩展字符集

128 Ç	160 á	192 └	224 α
129 ü	161 í	193 ┴	225 β
130 é	162 ó	194 ┬	226 Γ
131 â	163 ú	195 ├	227 π
132 ä	164 ñ	196 ─	228 Σ
133 à	165 Ñ	197 ┼	229 σ
134 å	166 ª	198 ╞	230 μ
135 ç	167 º	199 ╟	231 τ
136 ê	168 ¿	200 ╚	232 Φ
137 ë	169 ⌐	201 ╔	233 Θ
138 è	170 ¬	202 ╩	234 Ω
139 ï	171 ½	203 ╦	235 δ
140 î	172 ¼	204 ╠	236 ω
141 ì	173 ¡	205 ═	237 φ
142 Ä	174 «	206 ╬	238 ∈
143 Å	175 »	207 ╧	239 ∩
144 É	176 ▒	208 ╨	240 ≡
145 æ	177 ▒	209 ╤	241 ±
146 Æ	178 ▒	210 ╥	242 ≥
147 ô	179 │	211 ╙	243 ≤
148 ö	180 ┤	212 ╘	244 ⌠
149 ò	181 ╡	213 ╒	245 ⌡
150 û	182 ╢	214 ╓	246 ÷
151 ù	183 ╖	215 ╫	247 ≈
152 ÿ	184 ╕	216 ╪	248 °
153 Ö	185 ╣	217 ┘	249 ·
154 Ü	186 ║	218 ┌	250 ·
155 ¢	187 ╗	219 ▒	251 √
156 £	188 ╝	220 ▄	252 η
157 ¥	189 ╜	221 ▌	253 ²
158 ₧	190 ╛	222 ▐	254 ■
159 ƒ	191 ┐	223 ▀	255

海淀走读 0028000

030351

Document generated by Anna’s Archive around 2023-2024 as part of the DuXiu collection (https://annas-blog.org/duxiu-exclusive.html).

Images have been losslessly embedded. Information about the original file can be found in PDF attachments. Some stats (more in the PDF attachments):

{
  "filename": "44CKRE9TLTYuMjLlrp7nlKjlpKflhajvvIjnrKzlha3niYjvvInjgItfMTAyNzcyNjMuemlw",
  "filename_decoded": "\u300aDOS-6.22\u5b9e\u7528\u5927\u5168\uff08\u7b2c\u516d\u7248\uff09\u300b_10277263.zip",
  "filesize": 60483298,
  "md5": "119c4a7085bcf51df1510bc1855d2c64",
  "header_md5": "e8a651786a3a793aedcaa10b0a781cf1",
  "sha1": "5a7945b029332b39f97b06f8df021e355ac06164",
  "sha256": "4405982ce325f47319508aed78eb1138cc182e3c833f65e586616b77372342d5",
  "crc32": 187582051,
  "zip_password": "",
  "uncompressed_size": 61430074,
  "pdg_dir_name": "\u00ed\u2562DOS-6.22\u2569\u2561\u2559\u251c\u2524\u2264\u255a\u00bd\u00fa\u00bf\u2561\u250c\u2534\u2219\u2591\u00b5\u00fa\u2310\u00ed\u2556_10277263",
  "pdg_main_pages_found": 740,
  "pdg_main_pages_max": 740,
  "total_pages": 759,
  "total_pixels": 5329024076,
  "pdf_generation_missing_pages": false
}