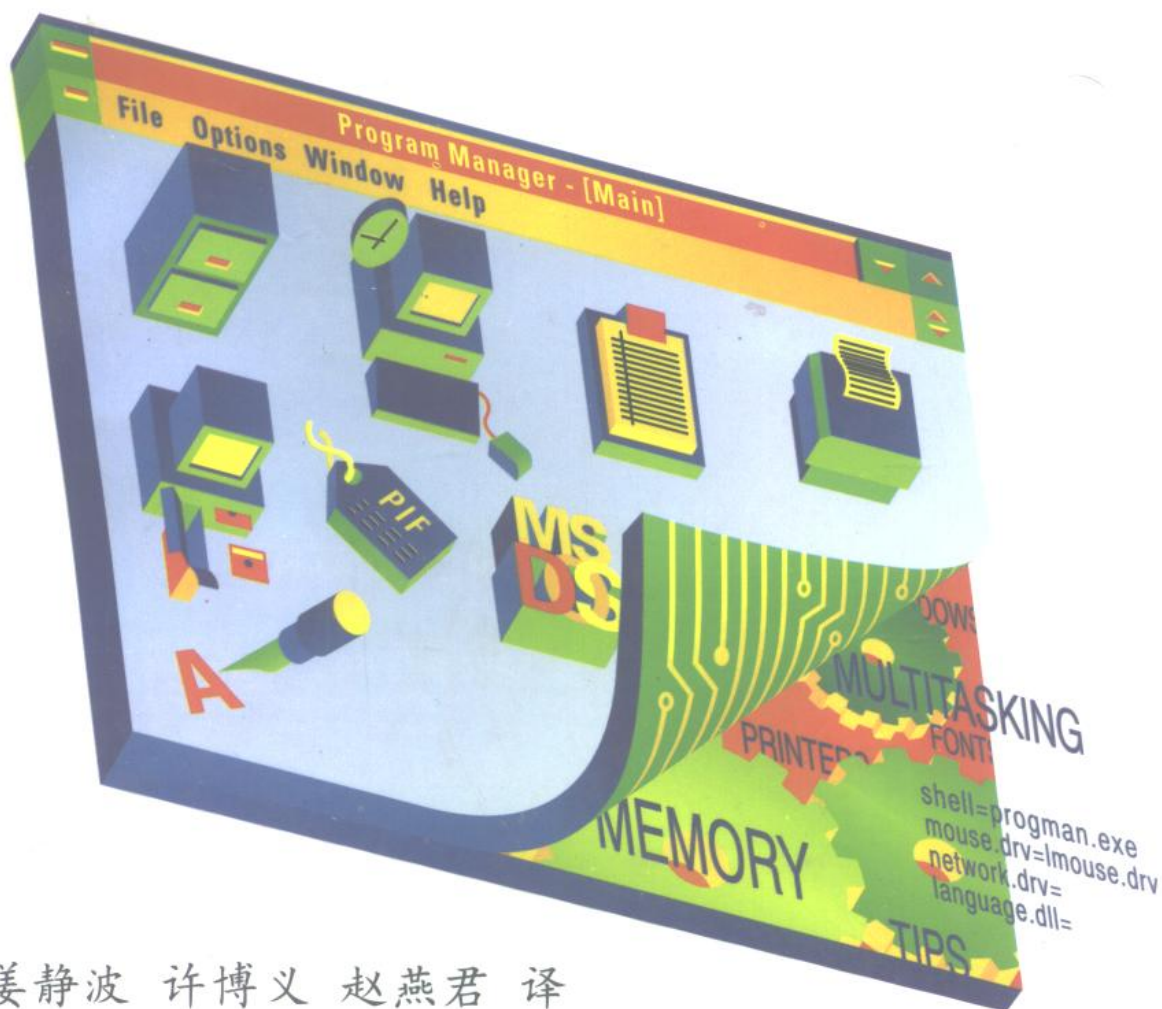


Windows 排错必读

Troubleshooting

WindowsTM

Mark Minasi



姜静波 许博义 赵燕君 译
姜静波 校

电子工业出版社

(京)新登字 055 号

内 容 提 要

本书由美国著名的 PC 机专栏作家、PC 机排错大师 MARK MINASI 所著。作者利用在 Windows 方面丰富的教学和实践经验引导读者步入 Windows 的殿堂,去探索 Windows 的奥秘,学会 Windows 排错的各种实用方法和技巧。本书内容丰富、针对性强,用通俗易懂的语言由浅入深地对 Windows 3.0 和 3.1 中多任务、内存管理、视频显示、字体和打印、批处理、连网、硬盘使用、运行 DOS 程序以及 Clipboard、DDE 和 OLE 等重要内容做了充分的介绍。



Copyright © 1993 SYBEX Inc., 2021 Challenger Drive, Alameda, CA 94501. World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

本书英文版由美国 SYBEX 公司出版,SYBEX 公司已将中文版独家版权授予北京美迪亚电子信息有限公司。未经许可,不得以任何形式和手段复制或抄袭本书内容。

Windows 排错必读

MARK MINASI 著

姜静波 许博义 赵燕君 译

姜静波 校

责任编辑 尚 平

电子工业出版社出版(北京市万寿路)

电子工业出版社总发行 各地新华书店经销

北京美迪亚电子信息有限公司排版

顺义天竺颖华印刷厂印刷

开本:787×1092 毫米 1/16 印张:18.75 字数:450 千字

1993 年 12 月第 1 版 1993 年 12 月第 1 次印刷

印数:4000 册 定价:28 元

ISBN7-5053-2210-9/TP·582

出版说明

计算机科学技术日新月异。为了引进国外最新计算机技术,提高我国计算机应用与开发的水平,中国电子工业出版社与美国万国图文有限公司合资兴办的北京美迪亚电子信息有限公司取得了美国 SYBEX 公司的独家版权代理。SYBEX 公司授权本公司通过电子工业出版社等出版机构全权负责在中国大陆出版该公司的中文版和英文版图书。现在与广大读者见面的是最近推出的第一批图书。今后我们还将陆续推出 SYBEX 公司的最新计算机图书和软件,为广大读者提供更好的服务,传递更多的信息。

美国 SYBEX 公司是世界著名的计算机图书出版商,该公司自 1976 年创办开始,其宗旨就是通过出版有效的、高质量的图书向计算机用户介绍实用技巧。我们优选翻译出版的图书是 SYBEX 公司的最新计算机图书,并采用了该公司提供的电子排版文件,从而提高质量并大大缩短了图书的出版时间,从根本上改变了以往翻译版图书要落后原版书较长的“时差”现象,这在电子技术日新月异的时代具有深远意义。

北京美迪亚电子信息有限公司

1993 年 11 月

献给那些曾经帮助我有幸成为这了不起行当中一员的人们：

Sonja Schwartzman，他是我的第一个读者；

Nancy Dehncke，她给了我第一个讲课的机会；

Pete Moulton，他给了我独自出外闯荡的勇气；

Fred Langa 和 **Stephen Levy**，他们使我开始了写作的生涯；

Dianne King，她使书中原先一些无把握的论题有了很大提高；

Gary Masters，他是 Sybex 公司的 **Maxwell Perkins**。

谢谢你们！谢谢这里没有指出名字的其他所有人。

致 谢

感谢：

Kris Ashton,她对书中关于 DDE 的章节做了大量的研究并且通读了这本书,她提出的建议使作者受益非浅；

Donna Cook,她为作者跑前跑后,做了大量的事情；

Leeta Wiley 和 Waggener/Edstrom 中的其他人,当我需要从 Microsoft 得到写作素材时,他们提供了大量的帮助(Waggener 是 Microsoft 的 PR 公司)；

Dave Stewart,他对这本书提出了大量的建议；

Gary Masters 以及为这本书付出了辛勤劳动的 Sybex 公司中的其他人。

这本书是在 Windows 3.00、3.00a 和 3.10 环境下用 Ami Pro 书写的。书中的屏幕是用 Windows 的打印屏幕特性和 Paintbrush 制作的,书中的图案是用 Micrografx Designer 3.1 绘制的。

引 言

当您第一次听说 Windows,是不是感到很激动?“突破了 640K 的界限”、“支持多任务——甚至 DOS 程序的多任务”、“提供的图形用户界面甚至连孩子也学得会”,我敢肯定当您阅读了这本书以后,就会彻底了解 Windows 并且开始迷信 Windows。确实,上面所说的那些内容都是千真万确的事实。值得一提的是最后一点——“容易使用”,它似乎是使 Windows 摆脱理论上应该是个操作系统的争论而进入流行桌面系统行列的关键。假如 Windows 是个操作系统的话,它的支持人员可就要叫苦连天了。

Windows 一开始似乎就很不错。还记得 Windows 3.0 刚启动时呈现在屏幕上的标志吗?见鬼,我竟然不知道 VGA 原来还可以产生那样朦胧的兰色调!它有着灰/暗灰兰的颜色,这种颜色竟然比幼儿园小朋友用手指画出来的我早已习以为常的色彩还要柔和。如果对此您也深有同感,那么会从下面的事实中得到更深的感受:虽然在只有 Windows 的老版本 Windows 1.0 和 2.0 的时候,我也能凑凑合合地使用它们,但当 ZSoft 公司开发的 Windows 形式的 PC Paintbrush 和相当好的 Windows Write 以及很好用的 Windows Terminal 程序出现时,它们那漂亮的界面深深地吸引了我(至于单人纸牌游戏 Solitaire,当然就更没得说了)。

但是另一方面,开始使用 Windows 之后我遇到了“无法恢复的应用程序错误”、“无法解释的系统死机”以及“与某些程序不兼容”的现象。最糟糕的是在启动了 Windows 之后,我那台以前在运行常用 DOS 应用程序如 WordPerfect、1-2-3 和 dBase III + 时速度很快的 16MHz 386SX PC 机的性能真的明显地下降了。

如果您也遇到过类似的情况,那您算是找对书了(所以去买一本吧!要知道,如果您只是在书店里“蹭”书看,那我可就得不到版税了)。

本书旨在填补“Windows 速成”与“Windows 编程”书之间的空白,它是一本为那些虽然不是 C 程序员但又必须为商务文件的准备提供 Windows 支持的人(哪怕这种商务活动只由一个雇员负责),或者为那些为等待文档打印和电子数据表格图形更新而浪费了太多时间而又特别注重效率的用户——那些追求尽善尽美、总是在说“一定还会有更好方法”的人编写的书。

“Windows 速成”是一些充斥 Windows 诀窍、技巧(对于专业人员来说,这些诀窍和技巧是人见人爱的)以及所谓“未公开”信息的书,但这种书大多不太严谨。我曾经翻阅过许多这种书,我认为它们讲述的内容都只有部分正确,也就是它们只适用于 Windows 的老版本。事实证明,所有“未公开”的信息都是任何人只要通过打印 Windows 的“read me”(自述)文件、购买 Microsoft 公司的“Windows Resource Kit”(Windows 资源工具集)或者阅读 Microsoft 公司的联机问题及解答数据库就可以得到的信息。最让我迷惑不解的是这些 Windows 速成书从来不向读者介绍重要的内容,就好象读者都笨得无法理解 Windows 的组织结构一样。

Windows 编程书虽然介绍重要的内容,但使用的却是 C 语言。如果读者不是个程序员,

或者不是个有相当水平的程序员,那么这些书就毫无用处。虽然它们包含了大量的信息,但读者需要的是一本能够读懂的书。

这就是这本书的妙处,它只提供那些有实用价值的诀窍和技巧。本书并不打算向您介绍如何得到“Windows 启动屏幕奥秘”之类没多少意义的东西,只想介绍 Windows 是如何工作的以及 Windows 为什么有的时候能够工作、有的时候无法工作。头脑中有了这些知识,就可以解决哪怕最棘手的 Windows 问题。这并不是说本书读起来会令人很烦,作者的意图其实是尽量不要浪费读者的宝贵时间,并在这一点上费了大量的心血。本书旨在让读者从头到尾地通读。如同软件也有生态的概念一样,您会发现 Windows 的各个部分之间存在着千丝万缕的联系。例如,我就遇到过由于显示控制板出了问题,结果使 Windows 显示出“系统内存出现一般错误”这一信息的事情。

在第一章中,我们将深入 Windows 去探讨 Windows 的组织结构。您会了解到 Windows 是由一些分别称之为 GDI(图形设备接口)、USER. EXE(Windows 的核心程序)和多任务调度程序 WIN386 等重要部分组成的。您还会了解到什么是 INI 文件,INI 文件是怎样组织的以及怎样通过修改 INI 文件来解决一些在使用 Windows 时会遇到的问题。

第二章我们探讨 Windows 实现 Windows 程序和 DOS 程序多任务功能的方法。在试图使 Windows 程序和 DOS 程序在一块儿运行时,Windows 对二者的处理是不相同的。在这一章中,您将学到调节 Windows 性能的方法。对 Windows 的性能进行调节有助于 Windows 更好地进行程序对换,使 Windows 可以最大限度地发挥所用计算机的效率。

第三章的篇幅非常大,它涉及了一个非常大的论题——内存管理。首先,我们向读者介绍理解不同种类的内存所需要的各种知识。如果您不打算了解什么是扩展内存、什么是扩充内存和什么是常规内存,那么您只要翻翻这章就行了。1987 年以来,在我举办的讨论班中,我已经向 20000 多人讲解了这些特别容易混淆的概念。因此,在内存概念的讲述方面,我认为我已经摸索出了一种大家容易理解的方法。DOS 的内存管理程序是读者必须要了解的内容,因为在 Windows 3.1 环境下,大部分的内存管理工作都是由 DOS 的内存管理程序完成的。因此,如果 DOS 内存管理程序初始化得不正确,就会引起大量的 Windows 故障。因为内存管理程序给出的信息有时会显得高深莫测,令人不着边际,所以当遇到了这种情况时,就需要通过阅读这一章来找出可以帮助您解决这类问题的诀窍和技巧。最后,我们还针对目前市场上流行的另外两个内存管理程序 QEMM-386 和 386Max,提供了一些使用方面的建议。

第四章和第五章侧重于显示器和磁盘系统与 Windows 的配合使用。在这两章中,您会学会如何为 Windows 选择最合适的硬件,如何使现有的硬件达到最高的速度;您还会明白怎样通过更合适的显示驱动程序来加快 Windows 的屏幕显示速度;最后,您会学会磁盘高速缓存 SMARTDrive 的初始化方法,避免 SMARTDrive 最新版本一个令人遗憾的副作用——“吞吃”磁盘数据。

第六章侧重于网络与 Windows。在这一章中,您会学会如何对网络进行最佳地配置,在什么情况下把 Windows 装在工作站上最好,在什么情况下把 Windows 装在服务器上最好。这一章还将介绍作者以国家一些最大公司的顾问身份,在安装 Windows 的过程中摸索到的一些诀窍。对于目前最流行的每个 LAN 操作系统,这一章都提供了单独的章节,可供对网络/Windows 排错方面感兴趣的读者参考。如果您是 LAN 上 Windows 的支持人员,我认为这一

章对您值得一读。

第七章讨论字体、打印机和 Windows 的打印功能。在这一章中,您也许可以搞清楚 True Type 系统的究竟。(实际上,True Type 是个非常好的系统。我这么说丝毫没有相反的意思)。如果您的打印机在 DOS 下打印正常,但在 Windows 下却乱打乱印;或者您不十分清楚 Palatino 与 Times Roman 字体之间有什么不同,可以参考这一章的内容。

第八章向您介绍如何使 Windows 的处理过程自动化。在 DOS 下,我们很多人都习惯于编写批文件。因为批文件能对某类任务进行自动处理,从而为您或者您的用户减少一些复杂的操作步骤。在 DOS 下,为了解决这一问题只需编写一个批文件即可。但是在 Windows 下,不存在相应的批处理系统,至少 Windows 没有提供这种批处理系统。因此我将向您介绍一些虽然并不昂贵,但却可以为 Windows 支持人员提供一些十分必要支持的系统。

第九章介绍在 Windows 程序之间进行数据剪贴和数据连接的方法。您一定知道 DDE、OLE 和剪贴板这些功能,至少您会听说过它们的名字。使多个程序相互协同地一道工作可以有多种方法,这一章先介绍比较简单的数据剪贴功能,然后再介绍相对比较难学的 OLE 功能,让您学会对象的连接和嵌入。

Windows 是个相当不错的环境,但不管怎样,一般我们大家还是要返回到 DOS 环境下。第十章就介绍这方面的内容。如果您曾经有过在做到 PIF 编辑程序的 Advanced 屏幕时,突然觉得“呀,好象不是这样!”而不得不选择 Cancel 的经历,那么这一章对您一定会有所帮助,因为它包括了与在 Windows 环境下运行 DOS 程序有关的各种复杂问题。不仅如此,这一章还将介绍如何使 DOS 程序在 Windows 环境下运行得更加快捷可靠。作为本书的最后一章——第十一章,我们将给出一些其它的诀窍。这一章将介绍如何在可能的情况下处理和避免讨厌的“Unrecoverable Application Error”(无法恢复的应用程序错误)错误,这一错误在 Windows 3.1 环境下被改称为“GP Fault”(GP 故障)。

术语和约定

除非特别指定,本书所说的“386”可以代表 386 系列中的任何机型。在编写本书之时,这一系列包括 80386DX(最初的 386)、80386SX(廉价较慢的 386,许多笔记本型微机用的就是这种 386)、80486DX(最初的 486)、80386SL(膝上型微机使用的功率较低、具有电源管理功能的 386 型号)、80486SX(486DX 的“经济”型号,它没有提供协处理器)和 80486DX2(486DX 的加速型号,其时钟频率增加了一倍)。

当提到存储器,1K 或一千字节指的是 1024 个字节,一个字节刚好可以存储一个字符所能包含的有效信息;1MB 或一兆字节代表 1024K,共 1,048,576 个字节。因此,配有 4 兆内存的计算机的存储空间不是 4,000,000 个字节,而是 4,194,304 个字节,它比 4 兆略多一点。RAM(随机访问存储器)是一种读写速度非常快的存储介质,计算机用它来暂时存储当前正被处理的程序和数据。RAM 是非永久性的存储器,亦即 PC 掉电之后它存储的内容会随之丢失。微机配备的 RAM 并不很多,大多数运行 Windows 的 PC 只配有 2 到 8 兆的 RAM。与 RAM 相比,硬盘存储的内容是永久性的。关掉 PC 之后,硬盘仍然具有数据记忆功能,这不会有任何问题。硬盘也叫做固定盘、温切斯特盘或者 DASD(直接访问存储设备)。硬盘的存储空间一般都比较大,大多数 Windows 用户的硬盘空间都在 60 到 210MB 之间。但同

RAM 相比, 硬盘的速度要慢得多。目前速度最快的硬盘一秒钟传输的数据量也不会超过 1.5MB, 而即使普通的 RAM 也可以达到每秒 100MB 的数据传输速度。相比之下, 硬盘难免相形见绌。所以如果您搞不清楚这两种存储介质之间的区别, 请记住: RAM 的速度很快但容量较小, 至少同硬盘相比是这样; 硬盘的速度较慢, 但容量很大; 硬盘在掉电之后仍有记忆功能, 而 RAM 则会忘得一干二净。

在本书中, 作者将使用一些最适当的术语来定义上述内容。作者使用这些术语的目的不是为了迷惑读者或者有意加深读者的印象, 只是为了减小篇幅简化叙述。

Windows 目录和 System 目录

我们介绍的第一个概念是 Windows 目录和 System 目录。Windows 总是要被安装到某个目录中, 如果您没有改变 Windows 的缺省目录, 那么 Windows 程序将被安装到子目录 C:\WINDOWS 中。由于作者无法预料 Windows 会被安装到哪个目录中, 所以在本书中我们把这一目录统称为 Windows 目录。但当需要用到具体的例子时, 我们还是使用 C:\WINDOWS 作为 Windows 的目录名。因此假如读者打算采用本书给出的例子, 而您的 Windows 目录又与此不同, 就应该对这一目录做相应的修改。在 Windows 目录下面通常还有一个叫做 C:\WINDOWS\SYSTEM 的子目录, Windows 利用这个目录保存各种驱动程序和一些基础性的程序等。然而如果在网络服务器上安装 Windows, 就没有这一目录。这时, 所有的 Windows 文件都被安装在 Windows 目录下。因此, 再考虑到 Windows 目录的任意性这一因素, 我们不把这一目录叫做 C:\WINDOWS\SYSTEM, 而采用与 Windows 目录相同的方法, 把它统称为 System 目录。

INI 文件和节

Windows 程序的行为与扩展名为 .INI 的 ASCII 正文文件有关。INI 文件规定了 Windows 程序的启动方式, 给出了 Windows 程序的配置信息。许多应用程序都有自己的 INI 文件, 例如流行的 Excel 电子表格程序就使用一个名为 EXCEL.INI 的文件来保存相应的配置信息。同样, Windows 在启动时也需要一些 INI 文件。正如您也许知道的那样, Windows 两个最重要的 INI 文件是 WIN.INI 和 SYSTEM.INI。它们都是 ASCII 文件, 所以用户可以使用任何正文编辑程序, 例如 DOS 5.0 的 EDIT 或者 Windows 的 Notepad 编辑程序, 对其进行编辑。INI 文件由节组成, 节由逻辑上相关的一组命令组成。节的开头是单独占一行的节名, 节名被括在方括号内。例如, 下面是 SYSTEM.INI 的一些可能内容:

```
[nonwindowsapp]
localtsrs = dosedit, ced
[386enh]
display = * vddvga
ega80woa.fon = ega80woa.fon
ega40woa.fon = ega40woa.fon
```

这一片断首先说明了一个节[nonwindowsapp],它只有一行;然后又说明了一个节[386enh],它有三行。

我们在例子中使用节[386enh]有一个很好的理由。关于这一点,相信读者在本书中会多次地领悟到。事实上,Windows的许多奇妙之处都产生于WIN.INI文件的[windows]节、SYSTEM.INI文件的[boot]节和[386]节。

顺便透露一点,任何INI文件的长度都不允许超过64K。如果INI文件超过64K就坏事了,这时或者系统会被锁住,或者用户会丢失某类数据。例如,如果SYSTEM.INI大于64K,Windows就会无法启动。在本书的正文部分,我们将对此给出更详细的介绍。

我们在第一章中还要详细地介绍INI文件。这里之所以预先提到这一文件是为了避免读者产生混淆,因为INI文件实在是太重要了。

驱动程序和 VxD

下面要介绍术语是“驱动程序 VxD”。早在操作系统问世的初期,操作系统的设计者就清醒地认识到操作系统中那些直接控制和处理打印机、显示器和磁盘驱动器等外部设备的部分,不应该被包括在操作系统的核心中,而应该把它们作为独立的驱动程序放到操作系统的外围。通过把操作系统的设备驱动功能划分成独立的设备驱动程序,在操作系统发行之后操作系统的设计者可以很容易地为新的设备增加驱动程序。

在这一点上,Windows与早期的操作系统完全相同。如果不包括显示器、鼠标器、键盘的驱动程序,以及在网络环境下不包括网络驱动程序,我们就无法建立一个最小的Windows配置。Windows实际上有两种驱动程序:用来控制Windows程序的驱动程序以及用来控制DOS程序的驱动程序。本书所说的“驱动程序”一般指的是“一种Windows可以用来控制系统硬件的程序”。

Windows在386增强模式下要对设备进行控制,必须配有能够使Windows把相应设备处理为“虚拟”设备的程序。下面我们以显示屏幕为例来说明这个问题。使屏幕成为虚拟设备,这意味着应该允许15个DOS程序可以按照各自的意思同时修改显示器屏幕上面的内容,而事实上这些程序修改的又都不是实际的显示器屏幕。为了产生这种魔术般的效果,Windows提供了虚拟显示驱动程序即VDD,例如虚拟鼠标器驱动程序VMD和虚拟键盘驱动程序VKD等。由于这些驱动程序的名字都以V开头、以D结尾,只有它们之间的内容可变,所以人们给它们起了个一般的名字“VxD”。SYSTEM.INI文件[386enh]节中的“device=Line”行用来装入VxD,所有的VxD都在这里装入。

例如,假定我们有个用来解决网络问题的VxD—VNETD.386。为了安装这个驱动程序,需要进行下列步骤:

1. 将VNETD.386拷贝到System目录。(还记得吧,我们在前面已经用过“System目录”。我们在以后还要经常用到这一术语。)

2. 用ASCII正文编辑程序如DOS EDIT或者Windows Notepad编辑SYSTEM.INI文件。

3. 在SYSTEM.INI中找到[386enh]节,它的第一行是[386enh]。

4. 在[386enh]这一行的下面建立一个空行,然后从键盘输入以下内容:

```
device=vnetd.386
```

5. 保存 SYSTEM. INI 文件。

就是这么简单,顺便应该提到的是,对 SYSTEM. INI 和 WIN. INI 文件的修改只有在退出 Windows 然后再次进入之后才能生效。

文件扩展

有时由于在用 Windows SETUP 程序安装 Windows 时没有安装某个文件,或者安装在硬盘上的文件被损坏了,我们需要从 Windows 的安装盘上再次装入某个文件。比起单单把文件拷贝到硬盘上,这时我们需要做更多的事情。为了能够用 6 张而不是 10 张 1.44MB 的软盘片发行 Windows,Microsoft 采用数据压缩技术对 Windows 安装盘上的文件进行了压缩。这就是说,如果我们想把 Windows 安装盘上的某个文件弄到硬盘上,单单把相应的文件拷贝到硬盘上是不够的。

如果把任何一张 Windows 安装盘的盘片插入软盘驱动器然后列一下目录,我们就会发现盘上大多数文件的扩展名都以下横线(一个“-”字符)结尾。这个下横线的意思是相应文件是以压缩形式存放的,所以只有在被复原之后才能使用。例如,您可以在一张磁盘上找到一个名为 TARTAN. BM _ 的文件,它是 Windows 提供的位图背景图案文件 TARTAN. BMP 的压缩形式。我曾经见过有人只是把软盘上的文件拷贝到硬盘上,然后再给它们换个名字就想使用,比如把 TARTAN. BM _ 改成 TARTAN. BMP(虽然我们愿是以这个位图文件为例,但实际上这同样适用于名字末尾带下横线的任何文件)。这样当然没法使用。

要使用压缩形式的文件,我们应该利用 Windows 安装盘提供的 EXPAND. EXE 程序。EXPAND. EXE 程序能自动地先将一个驱动器中压缩形式的文件复原,然后再将扩展之后的内容送到另一个驱动器中。EXPAND 的用法如下:

```
expand <源文件名> <目标文件名或者驱动器名:\子目录>
```

<源文件名>可以包括驱动器字母以及目录名,但遗憾的是不能有通配字符——比如 expand a: * . * c: 是不允许的。对于压缩形式的文件 TARTAN. BM _ ,我们可以使用以下的形式进行复原:

```
expand a:tartan. bm _ c:\windows\tartan. bmp
```

或许,在您系统的 Windows 目录中已经有了 EXPAND. EXE 。否则的话,您可以在 Windows 的安装盘上找到这个程序。对于 1.44MB 或者 1.2MB 的 Windows 安装盘,该程序在第 3 号盘上。

感谢您加入 Windows 的行列。现在让我们进入 Windows 的内部,去探索 Windows 的奥秘。

目 录

引言	1
第一章 Windows 的组织结构	1
启动 Windows; WIN.COM	1
一个软件的联合体	1
Windows 的模式:实模式、标准模式和 386 模式	1
Windows 的组成部分	4
DOS 在 Windows 中的作用	4
DOS 扩展程序	6
Windows 的核心程序	6
图形设备接口	7
USER.EXE	8
Windows 系统资源的保管	9
Windows 的用户命令解释程序	9
驱动程序与 SYSTEM.INI	10
INI 文件与 Windows 的各个组成部分	11
Windows 的资源	15
如何修改资源	16
小结	18
第二章 控制 Windows 的多任务	20
什么是多任务	20
并发和上下文切换的适用场合	22
如何实现 Windows 程序的多任务	23
为什么多任务有时会适得其反	27
Windows 程序的优先级	28
结论:应该选择应用程序	29
如何在 386 模式下实现非 Windows 程序的多任务	30
启动 DOS 的多任务	31
应用程序的互斥运行	32
386 方式下多任务的内部实现机制	32
加强多任务效果的调节方法	37
小结	40
第三章 Windows 的内存管理	41

内存的种类	42
设计计算机的内存:划分内存的第一兆地址空间	42
常规内存	44
TSR 与设备驱动程序	45
命令解释程序	46
用户程序	46
显示 RAM	46
系统保留区	49
扩展内存	51
EMS、LIM、调页式内存和扩充内存	56
DOS 扩展程序与 Windows	60
内存管理程序的使用	62
内存管理程序的功能	62
第一步:在开始内存管理之前,先检查内存的使用情况	63
高部内存区 HMA	65
理解 A20 处理程序的作用	69
建立并且使用高端内存块 UMB	71
建立并使用 UMB 的第一步:映射出高端内存的分布情况	78
第二步:建立 UMB	83
第三步:把程序装入 UMB 中	89
使用内存管理程序来模拟 LIM	93
使用内存管理程序建立 UMB 和 LIM	95
Shadow RAM	95
内存的排序和回填	97
DOS 内存管理小结	98
Windows 的内存管理	98
Windows 为 DOS 对话提供的 Limulation 支持	98
Windows 的 API 传送缓冲区	101
LIM 模拟与 API 传送缓冲区	104
API 传送缓冲区与标准模式	105
使用内存管理程序的诀窍与技巧	106
通过占用 ROM 的地址空间获得更多的 UMB	107
处理那些愚蠢的 TSR 程序	107
确定程序在 UMB 中的次序	108
问题程序与内存管理程序	108
所有的 TSR 都能被装入到内存的高端吗?	109
Quarterdeck 的扩充内存管理程序(QEMM—386)	109
QEMM 与 DOS 5.0 之间的相同之处	109
QEMM 比 DOS 5.0 多的功能	110
象 DOS 5.0 那样配置 QEMM	110

使用 QEMM 的特殊功能	114
QEMM 小结	119
Qualitas 的 386 to the Max(386Max)	120
Max 对那些愚蠢的 TSR 程序的处理	121
Maximize;Max 的配置程序	121
386Max 小结	122
通过内存管理程序排除 Windows 的错误	122
不同模式下 Windows 所使用的内存	123
Windows 使用的内存类型	124
标准模式下 Windows 的内存保护	125
386 模式下 Windows 的内存保护	126
Windows 的虚拟内存	126
实模式与标准模式下的 VM	126
386 增强模式下的 VM	127
小结	131
第四章 显示设备的改进和配置	133
VGA 与 Windows	133
为什么要使用分辨率更高的显示方式呢	134
显示方式的选择	135
提高 Windows 的显示速度	136
第一种方法:使用快速的显示驱动程序	136
显示协处理器和显示加速器	136
协处理器与加速器	137
显示协处理器的内部结构	137
IBM 的显示协处理器:PGA、8514 和 XGA	138
显示协处理器的前途、问题以及需要面对的现实	141
价格比较便宜的显示加速器	144
配有局部总线的显示器	145
Windows 彩色模型的理解	145
Windows 实现 16 色驱动程序的方法	147
Windows 实现 256 色显示方式的方法:一个附注	149
安装不同显示方式的驱动程序	149
谁来控制显示驱动程序	150
使用非标准的字体	151
改变 Windows 的 System 字体	152
选择系统主显示器的驱动程序:控制 SETUP.INF	153
通过修改 SETUP.INF 来实现驱动程序的自动选择	153
建立安装大字体用的 OEMSETUP.INF	156
在 Windows 环境下排除显示方面的错误	156
在 DOS 环境下对显示情况进行检查	156

在 DOS 环境下显示情况正常,但在 Windows 环境下显示情况却不正常	157
问题只出现在 386 增强模式下吗	158
我们只能看见屏幕的一部分吗	158
小结	158
第五章 Windows 磁盘的有效使用	159
为什么磁盘驱动器对于 Windows 的性能如此重要	159
理解磁盘高速缓存的概念	159
局部化原理	159
读操作的高速缓冲	160
高速缓存程序与磁盘读操作	161
高速缓存程序及磁盘写操作	163
SMARTDrive——Windows 的磁盘高速缓存程序	164
高速缓存的刷新	164
写操作的高速缓冲可能带来的危险	164
SMARTDrive 进行高速缓存刷新的时机	165
控制 SMARTDrive 缓冲区的大小	166
SMARTDrive 与 Windows 3.1	166
配有总线主控器的磁盘与双缓冲	167
总线主控技术之前的磁盘数据传输技术:DMA	167
把磁盘高速缓存程序装入到高端内存	167
DMA 传输只能针对物理内存进行	168
总线主控器带来的问题	168
其它的高速缓存程序	169
磁道缓冲区与高速缓存	169
硬件高速缓存与软件高速缓存	170
在 Windows 环境下使用第三方提供的设备驱动程序	170
“不兼容”磁盘的处理	171
禁止 Virtualhdirq	171
理解 Virtualhdirq	171
处理非 WD1003 控制器的方法	172
“兼容”磁盘的处理:32 位磁盘访问	172
从 16 位访问向上升档	172
32 位访问与有可能产生问题的硬盘	173
32 位访问增加了 DOS 的空间	174
在 Windows 环境下使用 SCSI 驱动器	174
解决软盘的问题	175
第六章 保证网络在 Windows 环境下的正常运行	176
选择 Windows 的网络安装形式	176
在工作站上安装 Windows	176
在服务器上安装 Windows	177

在服务器上安装 Windows	178
在 LAN 上建立 Windows 的工作站	178
改进 Windows 的配置过程	179
改善网络的性能	180
排除网络方面的错误	181
一般的网络排错建议	181
Novell 网与 Windows	183
解决 LAN Manager 和 Pathworks 的问题	186
Banyan 与网络	187
Windows 环境下通讯端口的使用	188
增加通讯缓冲区的大小	188
在增强模式下使用 COM3 和 COM4	189
改进 Windows 的 COM 驱动程序	189
只有 25 美元的“Windows 通讯协处理器”的奥秘	190
按照 COM 端口的排列顺序来使用这些通讯端口	191
第七章 Windows 下的字体与打印机	192
Windows 内部对字体的处理	192
为什么 WYSIWYG 只能是相对而言	193
另一个例子: Notepad 与 Courier 字体	194
Windows 使用的字体种类	194
位图字体在 Windows 环境下的使用	196
打印字体的安装	197
屏幕字体的安装	200
字盒在 Windows 环境下的使用	201
轮廓线字体的使用	202
TrueType, Windows 的轮廓线字体系统	203
字体的选择	203
小字型的 TrueType 字体	204
解决 TrueType 字体的问题	204
通过图形下载方式打印 True type 文本	206
排除 Windows 打印程序的错误	207
排除打印错误的开始步骤	207
为什么在 DOS 环境下能打印而在 Windows 环境下却不能打印	208
为什么要禁止使用“Fast Printing...”	209
取消“Fast Printing...”之后可能出现的问题	209
排除 Windows 环境下打印共享设备的错误	210
抛开 Print Manager	210
尽量得到最新版本的驱动程序	210
第八章 Windows 处理过程的自动化	212
从此不再需要进行记录: Windows Recorder	212

Recorder 的用途	213
宏的自动启动	213
Windows 批处理程序的一个例子: PubTech	213
十六进制转换程序的样本输出	216
以我们喜欢的方式启动 Windows 的应用程序	216
Visual Basic: 一个肯定可以弥补 C 语言不足的语言	220
高级的 C 语言给我们带来的烦恼	220
Visual Basic: 一个成功的故事	220
使用 Visual Basic 为程序编写图形界面	221
小结	221
第九章 进程间的相互通讯	222
简单的 IPC: 剪贴板	223
剪贴板中的数据类型	223
动态数据交换入门	228
什么是 DDE?	229
简单的 DDE: Paste Link	229
在没有提供 Paste Link 选项时如何进行粘贴连接	232
为什么 Paste Link 这个选项并不总是可用?	232
多久进行一次 Paste Link?	233
热连接、冷连接和温连接	234
DDE 的内部结构: 应用程序、主题和数据项	235
超级 DDE: 对象的连接和嵌入	236
逐步地建立 OLE	237
小结: Paste、DDE 以及 OLE 之间的差别	239
OLE/DDE 连接的维护	241
解决连接中断的问题	241
OLE 的注册数据库	244
排除注册数据库的错误	247
使用 OLE 的代价	248
DDE/OLE 的前途以及需要面对的现实	249
小结: IPC 故障的排除	251
第十章 Windows 环境下 DOS 应用程序的运行	253
Windows 支持 DOS 应用程序的方法	253
建立 PIF 文件: 在 Windows 环境下运行 DOS 应用程序的第一步	254
PIF 的基本设置	254
PIF 的高级选项	258
编写以 DOS 提示符结尾的批文件	262
在 Program Manager 中增加 DOS 应用程序	263
常见的错误以及解决这些错误的技巧	265
DOS 需要更多的环境空间	266