

ONE 3.5"

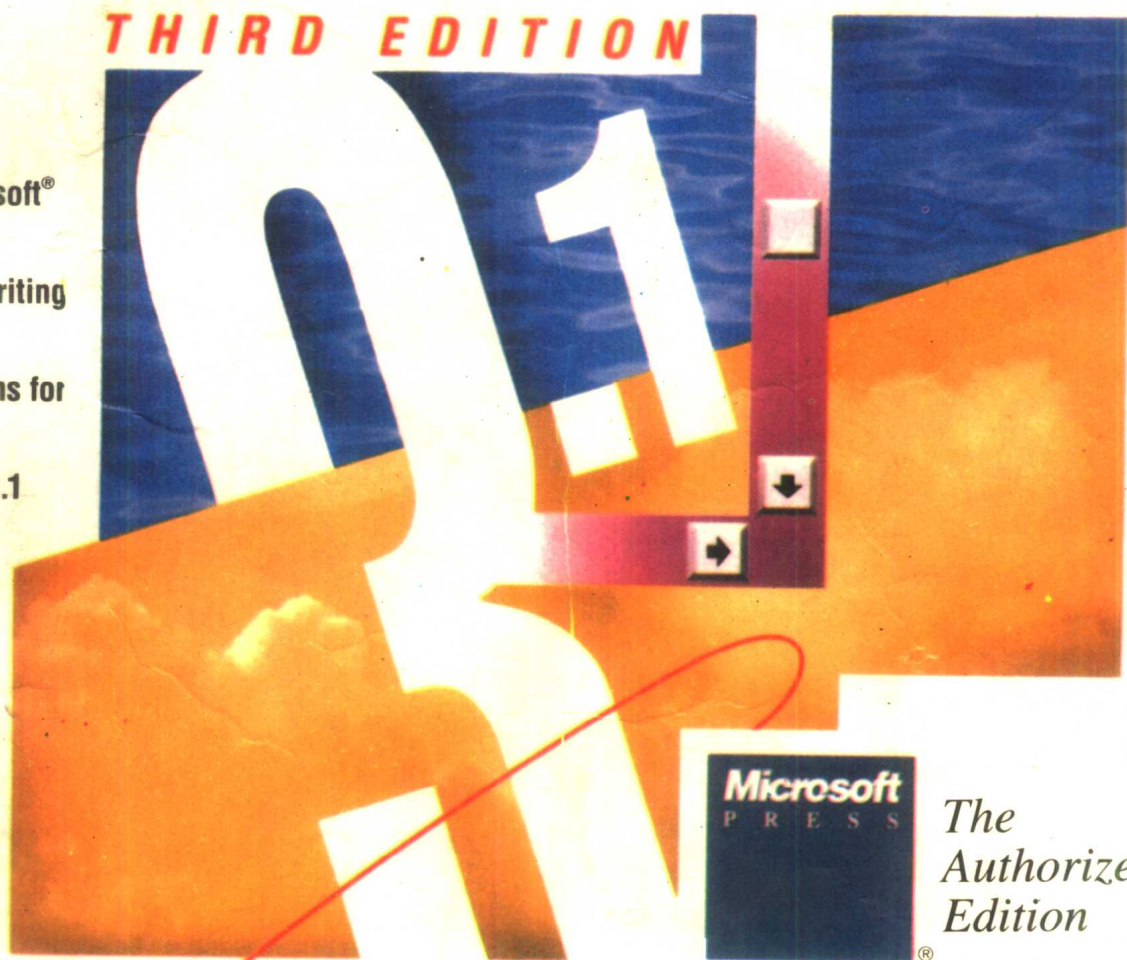
DISK INCLUDED



PROGRAMMING WINDOWS™ 3.1

THIRD EDITION

The Microsoft®
guide to writing
applications for
Windows 3.1



*The
Authorized
Edition*

程 序 设 计

北京希望电脑公司
海洋出版社

Windows 3.1 程序设计

文 都 等译
希 望 审校

海洋出版社
1993年·北京

PROGRAMMING WINDOWS 3.1

By Charles Petzold

本书英文版由 Microsoft 公司属下的 Microsoft 出版社(Microsoft Press)出版。版权归 Microsoft 公司所有。

本书中文版版权由 Microsoft Press 授予北京希望电脑公司和海洋出版社独家出版、发行。未经出版者书面许可,本书的任何部分都不得以任何形式或任何手段复制或传播。

Apple、LaserWriter、Macintosh 和 TrueType 是 Apple Computer, Inc. 的注册商标。

Borland 和 Turbo C 是 Borland International, Inc. 的注册商标。

ColorPro 和 Hewlett-Packard 是 Hewlett-Packard Company 的注册商标。

Current 和 Graphics Assistant 是 International Business Machines Corporation 的注册商标。

Microsoft、MS-DOS、Multiplan、QuickC 和 Windows 是 Microsoft Corporation 的注册商标。

OS/2 是 Microsoft Corporation 的注册商标。

SmallTalk 和 Xerox 是 Xerox Corporation 的注册商标。

(京)新登字第 087 号

责任编辑 阎世尊

Windows 3.1 程序设计

文 都 等译

希 望 审校

*

海洋出版社出版发行(北京复兴门外大街1号)

兰空印刷厂印刷

*

开本:787×1092毫米 1/16 印张:00.00 字数:000千字

1993年4月第1版 1993年4月第1次印刷

印数:0001—10000

ISBN:7-5027-3316-7/TP·178

定价:98.00元

前 言

自 1985 年秋问世以来的近七年中,Microsoft Windows 已成为使用 Intel 微处理器的个人计算机上的工业标准图形环境。《Windows 程序设计》一书虽然不是与 Windows 同时问世的,但其第二版距今已有两年时间。随着 Windows 3.1 的推出,有必要对该书作某些修订。

本版(第三版)作了一些修改。其一,所有程序均可用 Microsoft 或 Borland 编译器进行编译。所有 make 文件均使用编译器标志、链接库等环境变量。其二,所有程序均可以 C++ 方式进行编译。尽管本书没有使用任何具体的 C++ 特征,但是,如果希望以后向代码中添加 C++ 功能,则不妨先在 C++ 方式下对程序代码进行编译。

由于 Windows 3.1 只在保护模式下运行,所以,本书的文字和程序尽量考虑了这一模式所提供的各种优点。在 Windows 中,保护模式下的内存管理通常要比实模式下的内存管理更为简单。

有关文本和字体的章节集中介绍了 Windows 3.1 的另一增强功能,这就是被称为 TrueType 的无极轮廓字体技术。有关 DDE 的章节则给出了 Windows 3.1 中 DDE 管理库(DDE Management Library)的有关信息。本书还采用了新的通用对话框库(Common Dialog Box Library)。此外,本书增加了关于设备无关位图(DIB)和定制按钮的篇幅,它们是 Windows 3.1 所特有的。

自从本书的第一版在 1988 年出版以来,已经有许多程序员反映,这本书对他们编写 Windows 环境下的应用程序帮助极大。说心里话,这种与 MS-DOS 字符方式接口截然不同的 Windows 环境,的确是需要阅读有针对性的参考书才能够掌握的。使读者受益,是本书的唯一宗旨。

在《Windows 程序设计》第一版中,作者有意给出了用 C 语言编写应用程序的基础知识。当然,象这样的一本书,不可能一开始就探究功能齐全的应用程序的复杂特性,但是,我们可以告诉读者如何处理 Windows 程序的所有不同组件。应用程序员可以将这些组件结合在一起,以形成整个程序。

Windows 使用简单,但编程却很麻烦。有志成为优秀 Windows 程序员的人们经常要走一些弯路,才能满足大量程序设计的需求。为了满足这种要求,本书给出了 50 个以上的完整程序。这是本书的一大优点。许多程序短而精,并清晰地演示了各种 Windows 程序设计技巧。有些程序稍长,用来显示将各种组件结合在一起的方法。还有些程序是极有用的工具程序,另一些则是研究 Windows 的工具。

本书没有讲述如何使用 Windows。如果读者尚无使用 Windows 环境的经验,则需要略费工夫安装并熟悉之。Windows 学习起来并不困难。

本书也不致力于指导读者怎样编写 C 程序。在使用 Windows 编程之前,必须先对传统环境(如 MS-DOS)下的 C 程序设计有丰富的实践知识。否则,如果对 C 语言过于陌生,则有必要花费一些时间熟悉结构和指针等概念。熟悉 Intel 80x86 系列微处理器的分段体系结

构有助于读者理解本书的内容。如果读者了解 80x86 在实模式和保护模式下是如何寻址的,以及远程指针、函数等预备知识,这就足够了。为了方便读者,作者在适当的地方加入了一些说明。

要编译本书的程序或编写自己的 Windows 程序,需要下面两种软件包之一:

- 7.0 版 Microsoft C/C++ Professional Development System for Windows
- Borland C++ 3.1

如果读者还没有安装编译器,则应知道本书的程序只需要 small 模式下的函数库。也可以使用 Microsoft 和 Borland 之外的 C 编译器,只要这种编译器适合于编译 Windows 程序。大多数普通 C 编译器不能用来编译 Windows 程序。

要最有效地运行 Windows 和众多的开发工具,需要以下硬件支持:

- 基于 Intel 80386 微处理器的 IBM 个人机(或兼容机),配有硬盘、4MB 以上内存、使用 MS-DOS 3.30 或更高版本
- 图形显示器和视频卡,最好与 IBM VGA(视频图形阵列)或更好的显示器兼容
- 鼠标。尽管对大多数 Windows 程序来说,鼠标是可选的,但本书中的某些程序需要鼠标支持

有些读者对作者所使用的计算机配置比较感兴趣,不妨在此介绍一下。《Windows 程序设计》的第一版发行于 1987 年,当时所使用的是 IBM PC/AT Model 339(8MHz)。它配有两个 30MB 硬盘,系统板上有 512KB 内存以及一个 1.5MB 的 Intel Above Board PS/AT。该系统还包含一个 Microsoft 总线鼠标、一个 IBM 256KB 增强型图形适配器(EGA)和一个 NEC MultiSync 监视器。使用的字处理软件是 WordStar 3.3,打印工作都是在 IBM 5152 图形打印机上完成的。

对第二版的修订是在 1989 年和 1990 年完成的。当时使用的是 20MHz 的 IBM PS/2 Model 70。它配有 120MB 硬盘、6MB 内存、一个 IBM 8514/A 图形卡(尽管作者经常在 VGA 方式下运行 Windows)、一个 NEC MultiSync 4D 监视器和一个 Microsoft 鼠标。那时,作者已抛开了 WordStar,而使用 Microsoft Word 来修订本书。打印是在 NEC SilentWriter LC-890 PostScript 打印机上进行的。

现在发行的是本书的第三版。作者是在一台 Northgate Elegance 433(486-33)上运行 Windows 的,该机配置有 300MB 硬盘、8MB 内存和超级 VGA(SVGA)卡。目前所使用的还是 Windows 2.0 中的 Word 字处理软件。

象这样的一本书,如果没有其它人的帮助和鼓励,是不可能问世的。在此,我要以握手或拥抱(如果合适的话)来表示我对为本书的编写、审校、出版和发行作出过贡献的所有同仁表示衷心感谢!

编译者序

为了引进国外最新计算机技术,促进我国计算机应用与开发的水平,海洋出版社与北京希望电脑公司合作,首次向美国微软公司出版社(Microsoft Press)购买了一批计算机图书的版权,不久将与读者见面。这些图书是目前最新的计算机技术资料,涉及到 Windows、Windows for Workgroups、软件开发工具、数据库管理系统、程序设计语言、多媒体技术和标准计算机词典等领域。我们还将进一步与美国微软公司合作,推出各种软件和图书资料,为读者提供更好的服务。

本书以 Windows 程序设计为主题,针对各种不同层次的读者,由浅入深地介绍了编写 Windows 应用程序的各方面的技术。它讨论了以下五个主题:Windows 编程技术简介、读取输入、使用资源、图形设备接口、数据交换和链接。

本书的主要内容包括:

- 文本、键盘、鼠标、定时器、子窗口控制
- 内存管理、对话框、菜单、加速键
- GDI、绘图、位块传输、源文件、字体
- 打印机
- 剪辑板、动态数据交换
- 多文档界面、动态链接库

本书语言精练、编排合理、便于查阅,是一本学习 Windows 编程技术的经典著作。

在本书的翻译出版过程中,自始至终得到了微软公司北京代表处的热心指导和帮助,在此谨向他们表示诚挚的谢意。

1993年4月 北京

目 录

第一章 Windows 编程要素简介

1.1	Windows 发展史简介	(3)
1.2	用户的观点	(4)
1.2.1	图形用户界面(GUI)	(4)
1.2.2	GUI 的概念及基本原理	(5)
1.2.3	一致的用户界面	(5)
1.2.4	多任务的优点	(6)
1.2.5	内存管理	(7)
1.2.6	与设备无关的图形界面	(7)
1.2.7	MS-DOS 应用程序	(8)
1.3	程序员的观点	(9)
1.3.1	Windows 与 MS-DOS	(9)
1.3.2	投入 Windows	(9)
1.3.3	函数调用	(9)
1.3.4	动态链接	(10)
1.3.5	面向对象的程序设计	(11)
1.3.6	消息驱动式体系结构	(11)
1.3.7	窗口过程	(11)
1.4	编写第一个 Windows 程序	(12)
1.4.1	传统程序的问题何在	(12)
1.4.2	HELLOWIN 的文件	(13)
1.4.3	Make 文件	(17)
1.4.4	C 源代码文件	(19)
1.4.5	Windows 函数调用	(20)
1.4.6	大写字母标识符	(20)
1.4.7	新的数据类型	(21)
1.4.8	句柄简介	(21)
1.4.9	匈牙利表示法	(22)
1.4.10	程序入口点	(23)
1.4.11	登录窗口类	(24)
1.4.12	创建窗口	(26)
1.4.13	显示窗口	(27)
1.4.14	消息循环	(27)
1.4.15	窗口过程	(29)
1.4.16	处理消息	(30)
1.4.17	WM_PAINT 消息	(30)

1.4.18	WM_DESTROY 消息	(31)
1.4.19	模块定义文件	(32)
1.5	Windows 程序设计难点	(32)
1.5.1	Windows 程序中的调用关系	(33)
1.5.2	进队消息与不进队消息	(34)
1.5.3	无优先级的多任务	(35)
1.5.4	学习策略	(35)

第二章 文本输出

2.1	绘图与刷新	(37)
2.1.1	WM_PAINT 消息	(38)
2.1.2	有效矩形与无效矩形	(38)
2.2	GDI 简介	(39)
2.2.1	设备环境	(39)
2.2.2	获取设备环境的方法之一	(40)
2.2.3	绘图信息结构	(40)
2.2.4	获取设备环境句柄的方法之二	(42)
2.2.5	细论 TextOut	(42)
2.2.6	系统字体	(43)
2.2.7	字符大小	(44)
2.2.8	细论文本尺度	(45)
2.2.9	格式化文本	(46)
2.2.10	综合使用	(47)
2.2.11	SYSMETS1.C 窗口过程	(52)
2.2.12	屏幕空间不够	(53)
2.2.13	客户区域的大小	(53)
2.3	滚动条	(54)
2.3.1	滚动条的范围和位置	(55)
2.3.2	滚动条消息	(55)
2.3.3	在 SYSMETS 中加入滚动功能	(57)
2.3.4	使程序结构化以利于绘图	(61)
2.3.5	建立更好的滚动条	(62)
2.3.6	不用鼠标怎么办	(67)

第三章 键盘

3.1	键盘基础	(71)
3.1.1	键盘驱动程序	(71)
3.1.2	忽略击键	(72)
3.1.3	焦点	(72)
3.1.4	击键和字符	(73)
3.2	击键消息	(73)

3.2.1	系统与非系统击键	(74)
3.2.2	IParam 参数	(74)
3.2.3	虚拟键代码	(76)
3.2.4	变换状态	(78)
3.2.5	使用击键消息	(79)
3.3	增强系统:增加键盘接口	(79)
3.3.1	加入 WM_KEYDOWN 逻辑	(79)
3.3.2	发送消息	(81)
3.4	字符消息	(87)
3.4.1	WM_CHAR 消息	(89)
3.4.2	死字符消息	(89)
3.5	浏览键盘消息	(90)
3.6	脱字符(非光标)	(95)
3.6.1	脱字符函数	(95)
3.6.2	TYPE 程序	(96)
3.7	Windows 字符集	(103)
3.7.1	OEM 字符集	(103)
3.7.2	ANSI	(104)
3.7.3	OEM、ANSI 和字体	(105)
3.8	国际化考虑	(105)
3.8.1	字符集的处理	(106)
3.8.2	关于 MS-DOS 的讨论	(106)
3.8.3	数字小键盘的使用	(107)
第四章 鼠标		
4.1	鼠标基础知识	(109)
4.1.1	一些简单定义	(109)
4.2	客户区鼠标消息	(110)
4.2.1	简单的鼠标处理:一个例子	(111)
4.2.2	POINT、RECT 和 IParam	(115)
4.2.3	Shift 键的处理	(116)
4.2.4	双击鼠标按钮	(117)
4.3	非客户区鼠标消息	(117)
4.3.1	命中测试消息	(118)
4.3.2	消息产生消息	(119)
4.4	用户程序中的命中测试	(120)
4.4.1	一个假想的例子	(120)
4.4.2	示例程序	(121)
4.4.3	用键盘模拟鼠标	(124)
4.4.4	在 CHECKER 中加入键盘接口	(126)

4.4.5	子窗口用于命中测试	(130)
4.4.6	CHECKER 中的子窗口	(131)
5	鼠标的俘获	(135)
4.5.1	BLOWUP1 程序	(136)
4.5.2	鼠标光标形状的修改	(139)
4.5.3	StretchBit 调用	(140)
4.5.4	绘制俘获块	(141)
第五章 定时器		
5.1	定时器基础	(143)
5.1.1	SYSTEM.DRV 和 Windows 定时器	(143)
5.1.2	非异步定时器消息	(144)
5.2	定时器的使用:三种方法	(145)
5.2.1	方法一	(145)
5.2.2	方法二	(151)
5.2.3	方法三	(156)
5.3	状态报告中定时器的使用	(156)
5.3.1	图标的创造性使用	(159)
5.3.2	图标的强制使用	(160)
5.3.3	图标的保持	(161)
5.3.4	自由内存的计算	(161)
5.4	定时器在时钟中的应用	(161)
5.4.1	弹出式窗口的定位和缩放	(165)
5.4.2	日期与时间的获取	(166)
5.4.3	国际化	(166)
5.5	Windows 标准时间	(167)
第六章 子窗口控制		
6.1	按钮类	(169)
6.1.1	建立子窗口	(172)
6.1.2	子窗口向父窗口发信号	(174)
6.1.3	父窗口向子窗口发信号	(175)
6.1.4	压入按钮	(175)
6.1.5	复选框	(176)
6.1.6	单选按钮	(177)
6.1.7	分组框	(177)
6.1.8	改变按钮文本	(177)
6.1.9	可见的和启用的按钮	(178)
6.1.10	按钮和输入焦点	(178)
6.2	控制和颜色	(179)
6.2.1	系统颜色	(179)

6.2.2	按钮颜色	(181)
6.2.3	WM_CTLCOLOR 消息	(181)
6.2.4	自绘按钮	(183)
6.3	静态类	(190)
6.4	滚动条类	(191)
6.4.1	COLORS1 程序	(192)
6.4.2	自动键盘接口	(197)
6.4.3	窗口分组	(198)
6.4.4	给背景着色	(199)
6.4.5	给滚动条着色	(200)
6.4.6	处理多个实例	(201)
6.4.7	COLORS1 作为图标	(201)
6.5	编辑类	(201)
6.5.1	编辑类风格	(204)
6.5.2	编辑控制通知	(205)
6.5.3	使用编辑控制	(205)
6.5.4	发送给编辑控制的消息	(206)
6.6	列表框类	(206)
6.6.1	列表框风格	(207)
6.6.2	将串放入列表框	(208)
6.6.3	选择和提取条目	(209)
6.6.4	接收来自列表框的消息	(210)
6.6.5	一个简单的列表框应用	(210)
6.6.6	文件列表	(214)
6.6.7	Windows 的 head 程序	(215)
6.6.8	被浪费的 8KB 空间	(219)

第七章 内存管理

7.1	微处理器与内存	(224)
7.1.1	近地址和远地址	(225)
7.1.2	保护模式	(226)
7.2	Windows 中的内存组织	(227)
7.2.1	固定段和可移动段	(228)
7.2.2	可抛弃内存	(228)
7.2.3	全程内存布局	(229)
7.2.4	局部内存	(230)
7.3	代码段和数据段	(231)
7.3.1	小、中、压缩、大和巨内存模式	(231)
7.3.2	多个代码段	(232)
7.3.3	关于压缩模式和大模式	(233)

7.3.4	避免移动问题	(234)
7.3.5	程序段属性	(234)
7.4	在程序内分配内存	(236)
7.4.1	锁定用户块	(236)
7.4.2	快速示例	(237)
7.4.3	全程内存函数	(238)
7.4.4	其它全程内存函数	(240)
7.4.5	使用可抛弃的全程内存	(241)
7.4.6	巨型全程内存块	(242)
7.4.7	分配局部内存	(243)
7.4.8	其它局部内存函数	(245)
7.4.9	锁定用户自己的数据段	(245)
7.4.10	内存分配的简单方法	(246)
7.4.11	C语言内存分配函数的使用	(247)
第八章 图标、光标、位图和字符串		
8.1	编译资源	(248)
8.2	图标和光标	(250)
8.2.1	图象编辑器	(254)
8.2.2	获取图标句柄	(256)
8.2.3	图标的使用	(257)
8.2.4	备用光标的使用	(258)
8.3	资源和内存	(258)
8.3.1	位图:象素组成的图象	(259)
8.3.2	位图和刷子的使用	(259)
8.4	字符串	(263)
8.4.1	字符串资源的应用	(264)
8.4.2	消息框和字符串	(264)
8.4.3	字符串资源和内存空间	(265)
8.5	用户定义的资源	(266)
第九章 菜单与加速键		
9.1	菜单	(275)
9.1.1	菜单结构	(275)
9.1.2	菜单模板	(276)
9.1.3	在程序中访问菜单	(279)
9.1.4	菜单和消息	(280)
9.1.5	程序示例	(282)
9.1.6	菜单格式	(288)
9.1.7	定义菜单的严格方法	(288)
9.1.8	定义菜单的第三种方法	(289)

9.1.9	浮动弹出式菜单	(290)
9.1.10	使用系统菜单	(295)
9.1.11	修改菜单	(298)
9.1.12	其它菜单命令	(298)
9.1.13	创建菜单的其它方法	(299)
9.2	位图菜单	(304)
9.2.1	创建菜单位图的方法	(312)
9.2.2	内存设备环境	(312)
9.2.3	文本位图	(313)
9.2.4	位图的比例缩放	(314)
9.2.5	连接菜单	(315)
9.2.6	加入键盘接口	(316)
9.3	加速键	(317)
9.3.1	加速键的使用	(317)
9.3.2	安排加速键的几条规则	(317)
9.3.3	加速键表	(318)
9.3.4	加速键表的装入	(319)
9.3.5	键盘代码转换	(320)
9.3.6	接收加速键消息	(321)
9.3.7	菜单与加速键应用程序 POPPAD	(321)
9.3.8	启用菜单项	(328)
9.3.9	处理菜单选项	(328)
第十章 对话框		
10.1	模态对话框	(331)
10.1.1	创建一个“About”对话框	(331)
10.1.2	对话框模板	(335)
10.1.3	对话框过程	(337)
10.1.4	激活对话框	(338)
10.1.5	对话框风格的补充说明	(339)
10.1.6	定义控制的补充说明	(341)
10.1.7	一个更复杂的对话框	(342)
10.1.8	使用对话框控制	(348)
10.1.9	OK 和 Cancel 按钮	(351)
10.1.10	Tab 和组	(352)
10.1.11	在对话框上画图	(354)
10.1.12	使用对话框的其它函数	(354)
10.1.13	定义自己的控制	(355)
10.2	消息框	(361)
10.2.1	弹出信息	(362)

10.3	非模态对话框	(363)
10.3.1	模态对话框与非模态对话框的区别	(363)
10.3.2	新 COLORS 程序	(365)
10.3.3	HEXCALC: 窗口还是对话框	(370)
10.3.4	创造性地使用控制 ID	(376)
10.4	对文件的操作	(377)
10.4.1	推荐的方法	(377)
10.4.2	Windows 文件 I/O 函数	(378)
10.4.3	在 POPPAD 中加入文件处理	(380)
10.4.4	查找与替换	(399)
10.5	只有一个函数调用的 Windows 程序	(400)
第十一章 GDI 简介		
11.1	GDI 基本原理	(405)
11.2	设备环境	(406)
11.2.1	获取设备环境的句柄	(406)
11.2.2	获取设备环境信息	(408)
11.2.3	DEVCAPS1 程序	(408)
11.2.4	设备的大小	(419)
11.2.5	获取颜色信息	(421)
11.2.6	设备环境属性	(422)
11.2.7	保存设备环境	(423)
11.3	映射方式	(424)
11.3.1	设备坐标和逻辑坐标	(426)
11.3.2	设备坐标系统	(426)
11.3.3	视口和窗口	(426)
11.3.4	WM_TEXT 映射方式	(428)
11.3.5	“度量”映射方式	(430)
11.3.6	“自作主张的”映射方式	(433)
11.3.7	WHATSIZ 程序	(438)
第十二章 绘制图形		
12.1	画点	(443)
12.2	画线	(443)
12.2.1	使用后援画笔	(445)
12.2.2	画笔的创建、选择和删除	(446)
12.2.3	避免与设备相关	(449)
12.2.4	填充空隙	(450)
12.2.5	绘图方式	(450)
12.2.6	ROP2LOOK 程序	(451)
12.2.7	ROP2 与颜色	(455)

12.3	画填充区域.....	(456)
12.3.1	边界框.....	(457)
12.3.2	ARCS 程序	(461)
12.3.3	饼状图的三角学.....	(466)
12.3.4	Polygon 函数和多边形填充方式	(468)
12.3.5	用刷子填充内里.....	(468)
12.3.6	刷子和位图.....	(471)
12.3.7	位图刷子的创建与使用.....	(472)
12.3.8	刷子的对齐.....	(475)
12.4	矩形、区域和剪取	(477)
12.4.1	矩形函数.....	(477)
12.4.2	区域的创建和绘制.....	(479)
12.4.3	矩形与区域的剪取.....	(480)
12.4.4	CLOVER 程序	(481)
12.5	其它 GDI 函数	(485)
12.6	不停地画图的程序.....	(490)
第十三章 位、块传输和元文件		
13.1	颜色和位图.....	(497)
13.2	与设备无关的位图(DIB)	(497)
13.2.1	DIB 文件	(498)
13.2.2	压缩 DIB 内存格式	(499)
13.2.3	显示 DIB	(500)
13.2.4	创建 DIB	(507)
13.3	GDI 位图对象.....	(507)
13.3.1	在程序中创建位图.....	(508)
13.3.2	单色位图格式.....	(509)
13.3.3	彩色位图格式.....	(510)
13.3.4	位图的尺寸.....	(511)
13.4	内存设备环境.....	(511)
13.5	强大的 BLT	(512)
13.5.1	PatBlt 函数	(512)
13.5.2	Blit 坐标	(515)
13.5.3	用 BitBlt 传输位	(516)
13.5.4	DrawBitmap 函数	(517)
13.5.5	使用不同的 ROP 码	(518)
13.5.6	再论内存设备环境.....	(520)
13.5.7	颜色转换.....	(523)
13.5.8	映射方式转换.....	(524)
13.5.9	用 StretchBlt 伸展位图	(524)

13.5.10 动画	(525)
13.6 元文件.....	(529)
13.6.1 内存元文件的简单应用.....	(529)
13.6.2 将先文件存放在磁盘上.....	(531)
13.6.3 使用预先存在的元文件.....	(532)
13.6.4 将元文件用作资源.....	(534)
13.6.5 细论元文件.....	(538)
13.6.6 元文件能做的与不能做的.....	(539)

第十四章 文本和字体

14.1 简单的文本输出.....	(541)
14.1.1 文本输出函数.....	(541)
14.1.2 文本的设备环境属性.....	(543)
14.1.3 使用后援字体.....	(544)
14.1.4 输出灰色字符串.....	(545)
14.1.5 GrayString 的简单用法	(547)
14.1.6 不用 GrayString 显示灰色串	(547)
14.2 字体的基础知识.....	(547)
14.2.1 字体的类型.....	(547)
14.2.2 主题之一:系列与字型	(550)
14.2.3 字体资源文件.....	(552)
14.2.4 主题之二:获取点	(554)
14.2.5 为什么使用逻辑英寸.....	(555)
14.2.6 主题之三:行空和间距	(555)
14.2.7 “逻辑 twips”映射方式	(556)
14.3 逻辑字体的创建、选择和删除	(557)
14.3.1 PICKFONT 程序	(559)
14.3.2 逻辑字体结构.....	(568)
14.3.3 字体映射算法.....	(571)
14.3.4 查明字体信息.....	(572)
14.4 枚举字体.....	(574)
14.5 格式化文本.....	(585)
14.5.1 单行文本的对齐.....	(586)
14.5.2 处理段落.....	(587)
14.6 字体的 POPPAD	(595)

第十五章 使用打印机

15.1 打印、缓冲和转义	(599)
15.2 打印机设备环境.....	(602)
15.2.1 获取 CreateDC 参数	(602)
15.2.2 修改后的 DEVCAPS 程序	(604)

15.2.3	DeviceMode 调用	(611)
15.2.4	检查 BitBlt 能力	(612)
15.3	打印原理	(613)
15.3.1	Escape 函数	(613)
15.3.2	FORMFEED 程序	(614)
15.4	打印图形和文本	(616)
15.4.1	Bare-Bones 打印	(619)
15.4.2	设置终止过程	(621)
15.4.3	Windows 如何使用 AbortProc	(622)
15.4.4	实现终止过程	(623)
15.4.5	增加一个打印对话框	(626)
15.4.6	增加 POPPAD 的打印功能	(630)
15.4.7	处理错误码	(635)
15.5	打印分块技术	(637)
15.5.1	形成打印块	(638)
15.5.2	终止过程的不同应用	(640)
15.6	打印机和字体	(645)
第十六章 剪辑板		
16.1	剪辑板的简单使用	(649)
16.1.1	标准剪辑板数据格式	(649)
16.1.2	把文本传送到剪辑板	(650)
16.1.3	从剪辑板上获取文本	(651)
16.1.4	剪辑板的功能	(652)
16.1.5	打开和关闭剪辑板	(653)
16.1.6	使用位图剪辑板	(653)
16.1.7	修订后的 BLOWUP 程序	(653)
16.1.8	元文件和元文件图片	(659)
16.2	复杂的剪辑板应用	(663)
16.2.1	利用多个数据项	(664)
16.2.2	延迟再生	(665)
16.2.3	私有数据格式	(666)
16.3	实现剪辑板浏览器	(668)
16.3.1	剪辑板浏览器链	(668)
16.3.2	剪辑板浏览器函数和消息	(668)
16.3.3	一个简单的剪辑板浏览器	(671)
第十七章 动态数据交换 (DDE)		
17.1	基本概念	(675)
17.1.1	应用程序、主题和项	(676)
17.1.2	交谈的类型	(676)