

Windows 95/98 操作系统

编程实例详解

周万宁 孙抗毒 编著



随书附实例软盘

全面介绍了 Windows 95/98 操作系统的编程方法

- 应用程序基本结构与开发步骤
- 图形设备接口与 Windows 绘图
- 文本输出
- Windows 资源
- Windows 标准控件
- 通用控件
- 文件操作
- 操作剪贴板
- 多线程编程
- 多文档界面
- 动态链接库
- Internet 编程
- 数据库编程

所附软盘包括书中所有实例的源程序



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

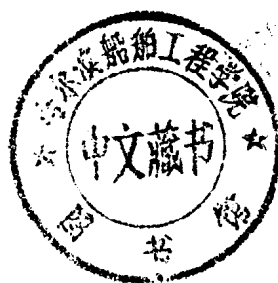
TP316.17

429221

7 +

Windows 95/98 操作系统 编程实例详解

周万宁 孙抗毒 编著



00429221



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

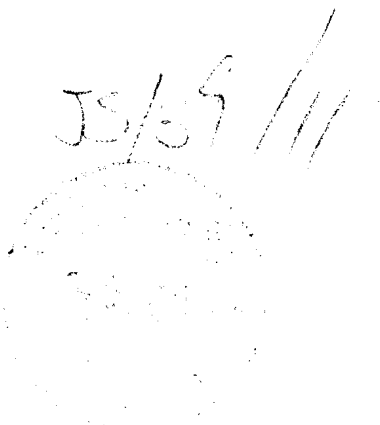
内容提要

Windows 操作系统是当前最为流行的微机操作系统,自其问世以来,受到全世界广大用户的热烈欢迎。特别是近几年来先后问世的 Windows 95/98 操作系统,由于大大增强了与 Internet 相关的功能,所以应用更加广泛。

Windows 操作系统的广泛应用迫使软件专业技术人员尽快从 DOS 平台转到 Windows 平台,因此,目前学习软件开发,必须首先了解并掌握 Windows 操作系统的编程。

本书面向急需学习 Windows 操作系统编程或有初步基础的软件专业技术人员,以讲解实例为主,说明如何编写 Windows 操作系统环境下运行的程序。

书中的实例从易到难,逐步深入,涉及 Windows 操作系统编程的多个方面。所有实例均在本书的伴随软盘中,读者可直接或稍加修改后使用。



书 名: **Windows 95/98 操作系统编程实例详解**

编 著 者: 周万宁 孙抗毒

责任编辑: 陆伯雄 寇国华

印 刷 者: 北京市天竺颖华印刷厂

出版发行: 电子工业出版社出版、发行 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话 68214070

经 销: 各地新华书店经销

开 本: 787×1092 1/16 印张: 19.75 字数: 462 千字

版 次: 1998 年 11 月第 1 版 1998 年 11 月第 1 次印刷

定 价: 40.00 元

印 数: 5000 册

书 号: ISBN 7-5053-4942-2/TP.2425

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换

版权所有·翻印必究

前 言

Microsoft 公司大力推出的 Windows 操作系统是当前最为流行的微机操作系统,自其问世以来,受到全世界广大用户的热烈欢迎。特别是近几年来先后问世的 Windows95/98 操作系统,由于大大增强了与 Internet 相关的功能,所以应用更加广泛。

为充分发挥 Windows 操作系统的强大功能,同时 Windows 操作系统的广泛应用也迫使软件专业技术人员尽快从 DOS 平台转到 Windows 平台。虽然目前有许多开发工具以及开发环境,例如 Delphi, PowerBuild 等等,但是不可否认的是,目前学习软件开发,必须首先了解并掌握 Windows 操作系统的编程。

本书面向急需学习 Windows 操作系统编程或有初步基础的软件专业技术人员,以讲解实例为主,说明如何编写 Windows 操作系统环境下运行的程序。

第 1 章 Windows 操作系统概述,说明了 Windows 操作系统发展历程及功能特点。

第 2 章开发应用程序预备知识,介绍了有关 Windows 操作系统窗口、事件驱动、消息、对象与句柄及匈牙利表示法等。

第 3 章应用程序基本结构与开发步骤,说明了应用程序组成、源程序的组成结构、WinMain 函数和窗口函数及数据类型和数据结构、API 函数、WinMain 函数、应用程序示例以及开发步骤等。

第 4 章图形设备接口与 Windows 绘图,讲解了图形刷新、图形设备接口、映射模式、绘图工具与颜色、常用绘图函数及位图等。

第 5 章文本输出,主要介绍了设置文本设备环境以及文本输出过程等。

第 6 章 Windows 资源,描述了 Windows 的光标、图标、位图、菜单、加速键资源以及对话框等资源的应用。

第 7 章 Windows 标准控件,说明了 Windows 标准控件的概念、分类、创建、与应用程序之间的消息传递,并描述了静态控件、按钮、滚动条、列表框、编辑框及组合框等控件的编程方法。

第 8 章通用控件,主要介绍了通用控件的分类及其功能、创建方法以及与应用程序间的消息传递等。具体说明了工具栏、状态栏、窗口类、资源管理类以及滚动类通用控件的编程方法。

第 9 章文件操作,介绍了有关 Windows 操作系统文件操作的特点,常用文件操作函数以及文件操作的编程方法。

第 10 章操作剪贴板,说明了 Windows 操作系统剪贴板的数据格式以及如何更为有效地通过编程操作剪贴板。

第 11 章多线程编程,讲解了 Windows 操作系统中进程与线程概念、管理、定时器与定时器消息、多线程编程方法及线程同步概念等。

第 12 章多文档界面,说明了多文档的特点、消息及其处理及保存子窗口数据等。

第 13 章动态链接库,主要介绍了动态链接库的概念、文件组成结构、加载方式以及使用

动态链接库实现数据共享等。

第 14 章 Internet 编程, 讲解了 Internet 网络基础、网络编程接口 WinsockAPI、Socket 编程模型等。

第 15 章数据库编程, 说明了与数据库编程有关的 ODBC 功能及其组成结构、结构化查询语言 SQL、关系数据库模式结构及使用 ODBCSDK 开发数据库等。

附录部分列出了常用的 WindowsAPI 函数, 供读者参考。

周万宁同志负责编写本书的第 1 章到第 10 章, 孙抗毒同志负责编写第 11 章到第 15 章。参加本书编写工作的还有胡建华、刘加明、吕梅、段来盛、卢山、杨新、韩珊以及官章全等同志。

本书所附的伴随软盘包括书中所用到的所有实例, 程序的安装与使用详见软盘中的“Readme. Txt”文件。

由于作者水平有限, 书中难免有不足之处, 还望读者批评指正。

关于本书所附软盘

本书所附的软盘包括书中所用到的所有实例的源程序, 程序的安装与使用详见软盘中的“Readme. txt”文件。

目 录

前言

第 1 章 Windows 操作系统概述..... 1

- 1.1 发展历程..... 2
- 1.2 功能特点..... 3

第 2 章 开发应用程序预备知识..... 5

- 2.1 窗口..... 5
 - 2.1.1 性质..... 5
 - 2.1.2 外观..... 5
 - 2.1.3 管理..... 6
- 2.2 事件驱动..... 6
- 2.3 消息..... 7
 - 2.3.1 组成..... 7
 - 2.3.2 产生和分类..... 8
 - 2.3.3 应用程序常用消息..... 8
 - 2.3.4 消息队列..... 10
- 2.4 对象与句柄..... 10
- 2.5 匈牙利表示法..... 11

第 3 章 应用程序基本结构与开发步骤..... 12

- 3.1 应用程序组成..... 12
- 3.2 源程序组成结构..... 12
 - 3.2.1 WinMain 函数和窗口函数..... 13
 - 3.2.2 数据类型和数据结构..... 13
 - 3.2.3 Windows API 函数..... 15
- 3.3 WinMain 函数..... 15
 - 3.3.1 函数说明..... 15
 - 3.3.2 初始化..... 16
 - 3.3.3 消息循环..... 18
- 3.4 窗口函数..... 19
- 3.5 应用程序示例..... 20
- 3.6 开发步骤..... 23

第 4 章 图形设备接口与 Windows 绘图..... 24

- 4.1 图形刷新..... 24
 - 4.1.1 刷新请求..... 24
 - 4.1.2 系统响应..... 24

- 4.1.3 无效区域与无效矩形..... 25

- 4.1.4 刷新方法..... 25

4.2 图形设备接口..... 25

- 4.2.1 设备无关性..... 25
- 4.2.2 设备环境和设备描述表..... 26
- 4.2.3 获取设备环境方法..... 27

4.3 映射模式..... 29

- 4.3.1 概念..... 29
- 4.3.2 设置..... 30
- 4.3.3 窗口与视口..... 30
- 4.3.4 定义窗口与视口..... 31

4.4 绘图工具与颜色..... 32

- 4.4.1 画笔..... 32
- 4.4.2 画刷..... 33
- 4.4.3 颜色..... 35

4.5 常用绘图函数..... 35

4.6 绘图示例..... 39

- 4.6.1 使用基本绘图工具与函数..... 39
- 4.6.2 设置与使用映射模式..... 42

4.7 位图..... 45

- 4.7.1 概念..... 45
- 4.7.2 内存设备环境..... 45
- 4.7.3 操作过程..... 46
- 4.7.4 应用程序组织方法..... 48
- 4.7.5 位图示例..... 49

第 5 章 文本输出..... 52

5.1 设置文本设备环境..... 52

- 5.1.1 选用系统字体..... 52
- 5.1.2 创建自定义字体..... 53
- 5.1.3 设置字体和背景颜色..... 53

5.2 文本输出过程..... 54

- 5.2.1 获取字体信息..... 54
- 5.2.2 格式化文本..... 55
- 5.2.3 调用函数输出文本..... 56

5.3 文本输出示例..... 56

- 5.3.1 程序名称..... 56
- 5.3.2 示例目的..... 57

5.3.3 程序清单	57	7.3.3 滚动条与应用程序间的消息传递 ...	117
5.3.4 运行结果	59	7.3.4 常用滚动条操作函数	118
第 6 章 Windows 资源	60	7.3.5 滚动条消息处理模板	119
6.1 光标、图标和位图资源	60	7.3.6 滚动条控件示例	120
6.1.1 光标资源	60	7.4 静态控件	124
6.1.2 图标资源	61	7.4.1 功能与特点	124
6.1.3 位图资源	62	7.4.2 创建	124
6.2 菜单与加速键资源	62	7.4.3 发送静态控件消息	125
6.2.1 菜单外观及组成部分	62	7.4.4 静态控件示例	125
6.2.2 创建菜单过程	63	7.5 列表框	128
6.2.3 操作菜单项	66	7.5.1 功能与特点	129
6.2.4 动态创建菜单	69	7.5.2 创建	129
6.2.5 加速键资源	70	7.5.3 列表框和应用程序之间消息传递 ...	130
6.2.6 菜单示例	72	7.5.4 列表框示例	132
6.3 对话框资源	77	7.6 编辑框	135
6.3.1 功能	77	7.6.1 功能特点	135
6.3.2 特点	78	7.6.2 创建	136
6.3.3 控件	79	7.6.3 编辑框与应用程序间的消息传递 ...	136
6.3.4 分类	79	7.6.4 编辑框示例	137
6.3.5 模态对话框编程方法	79	7.7 组合框	143
6.3.6 模态对话框示例	83	7.7.1 功能与特点	143
6.3.7 非模态对话框编程方法	88	7.7.2 创建	144
6.3.8 非模态对话框示例	90	7.7.3 组合框与应用程序间消息传递	145
6.3.9 通用对话框	94	7.7.4 组合框示例	146
6.3.10 通用对话框示例	96	第 8 章 通用控件	149
第 7 章 Windows 标准控件	102	8.1 概述	149
7.1 概述	102	8.1.1 分类及其功能	149
7.1.1 分类	102	8.1.2 创建	150
7.1.2 创建	102	8.1.3 通用控件与应用程序间消息传递 ...	150
7.1.3 控件与应用程序之间的消息传递 ...	104	8.2 窗口控制类通用控件	152
7.1.4 常用子窗口操作函数	105	8.2.1 工具栏	152
7.1.5 输入焦点	107	8.2.2 工具栏提示	154
7.2 按钮控件	107	8.2.3 状态栏	156
7.2.1 按钮控件的类型及功能特点	107	8.2.4 窗口类通用控件示例	157
7.2.2 创建按钮控件	109	8.3 资源管理类通用控件	164
7.2.3 按钮控件与应用程序的消息传递 ...	110	8.3.1 树形视图	165
7.2.4 按钮控件示例	111	8.3.2 列表视图	170
7.3 滚动条控件	116	8.4 滚动类通用控件	175
7.3.1 功能特点与分类	116	8.4.1 跟踪条控件	176
7.3.2 创建滚动条	116	8.4.2 微调按钮控件	178
		8.4.3 滚动类通用控件示例	180

第 9 章 文件操作.....	184	第 13 章 动态链接库.....	253
9.1 特点.....	184	13.1 概述.....	253
9.2 常用文件操作函数.....	184	13.1.1 与静态链接库的区别.....	253
9.2.1 CreateFile 函数.....	185	13.1.2 运行机制.....	253
9.2.2 ReadFile 函数.....	187	13.1.3 典型应用.....	254
9.2.3 WriteFile 函数.....	188	13.2 文件组成结构.....	254
9.2.4 GetFileInformationByHandle 函数.....	189	13.3 加载方式.....	256
9.2.5 SearchPath 函数.....	190	13.4 使用动态链接库实现数据共享.....	257
9.2.6 SetFilePionter 函数.....	191	第 14 章 Internet 编程.....	260
9.3 文件操作示例.....	192	14.1 Internet 网络基础.....	260
第 10 章 操作剪贴板.....	204	14.1.1 Internet 协议.....	260
10.1 剪贴板数据格式.....	204	14.1.2 Internet 地址.....	261
10.2 操作文本剪贴板.....	204	14.1.3 Internet 域名系统.....	261
10.2.1 向剪贴板发送文本.....	204	14.1.4 协议端口.....	261
10.2.2 获取剪贴板文本.....	207	14.2 网络编程接口 Winsock API.....	262
10.3 操作位图剪贴板.....	208	14.3 Socket 编程模型.....	263
10.4 剪贴板示例.....	209	14.3.1 建立 Socket.....	263
第 11 章 多线程编程.....	221	14.3.2 配置 Socket.....	264
11.1 进程与线程概念.....	221	14.3.3 网络 I/O.....	265
11.2 进程与线程管理.....	221	14.3.4 关闭 Socket.....	266
11.3 定时器与定时器消息.....	222	14.4 编程示例.....	266
11.3.1 功能与特点.....	222	第 15 章 数据库编程.....	269
11.3.2 使用方法.....	222	15.1 ODBC 功能及其组成结构.....	269
11.3.3 定时器消息.....	223	15.2 结构化查询语言 SQL.....	270
11.4 多线程编程方法.....	223	15.2.1 关系数据库模式结构.....	270
11.4.1 基本操作.....	223	15.2.2 SQL 语法简介.....	271
11.4.2 线程之间通信.....	225	15.3 使用 ODBC SDK 开发数据库.....	274
11.4.3 多线程编程示例.....	226	15.3.1 开发步骤.....	274
11.5 线程同步概念.....	234	15.3.2 ODBC 应用程序示例.....	276
11.5.1 问题提出.....	234	附录 常用 Windows API 函数.....	278
11.5.2 临界区法.....	234	1. 窗口操作类.....	278
第 12 章 多文档界面.....	237	CloseWindow 函数.....	278
12.1 特点.....	237	CreateWindow 函数.....	278
12.2 组成及层次关系.....	238	DestroyWindow 函数.....	279
12.3 创建.....	238	EnableWindow 函数.....	279
12.4 消息及其处理.....	240	GetClientRect 函数.....	279
12.4.1 多文档消息.....	240	GetParent 函数.....	279
12.4.2 处理消息.....	243	GetWindowLong 函数.....	279
12.5 保存子窗口数据.....	244	GetWindowText 函数.....	280
12.6 多文档界面示例.....	244	GetWindowTextLength 函数.....	280

InvalidateRect 函数	280	EnableMenuItem 函数	290
MoveWindow 函数	280	EndDialog 函数	291
RegisterClass 函数	281	GetDlgItem 函数	291
SetFocus 函数	281	GetDlgItemInt 函数	291
SetMenu 函数	281	GetDlgItemText 函数	291
SetWindowLong 函数	281	GetMenu 函数	292
SetWindowText 函数	282	GetScrollPos 函数	292
ShowWindow 函数	282	GetScrollRange 函数	292
UpdateWindow 函数	282	GetStockObject 函数	292
2. 图形用户接口(GDI)类	282	InsertMenu 函数	293
Arc 函数	282	LoadAccelerators 函数	293
BeginPaint 函数	283	LoadBitmap 函数	293
BitBlt 函数	283	LoadCursor 函数	293
CreateCompatibleBitmap 函数	283	LoadIcon 函数	294
CreateCompatibleDC 函数	283	LoadMenu 函数	294
DeleteDC 函数	284	ModifyMenu 函数	294
Ellipse 函数	284	SelectObject 函数	294
EndPaint 函数	284	SetDlgItemInt 函数	295
GetBkColor 函数	284	SetDlgItemText 函数	295
GetDC 函数	285	SetScrollPos 函数	295
GetMapMode 函数	285	SetScrollRange 函数	296
GetObject 函数	285	4. 文件操作类	296
GetTextColor 函数	285	CreateFile 函数	296
GetTextExtentPoint32 函数	285	GetFileInformationByHandle 函数	296
GetTextMetrics 函数	286	ReadFile 函数	297
LineToEx 函数	286	SearchPath 函数	297
MoveToEx 函数	286	SetFilePointer 函数	297
Pie 函数	286	WriteFile 函数	298
Polygon 函数	287	5. 剪切板类	298
Polyline 函数	287	CloseClipboard 函数	298
Rectangle 函数	287	EmptyClipboard 函数	298
ReleaseDC 函数	288	GetClipboardData 函数	298
RoundRect 函数	288	OpenClipboard 函数	299
SetBkColor 函数	288	SetClipboardData 函数	299
SetMapMode 函数	288	6. 消息处理类	299
SetTextColor 函数	289	DispatchMessage 函数	299
TextOut 函数	289	GetMessage 函数	299
3. 资源和控件类	289	IsDialogMessage 函数	300
AppendMenu 函数	289	PeekMessage 函数	300
CreateDialog 函数	289	PostQuitMessage 函数	300
CheckMenuItem 函数	290	PostThreadMessage 函数	300
CreateMenu 函数	290	SendDlgItemMessage 函数	301
DialogBox 函数	290	SendMessage 函数	301

TranslateAccelerator 函数	301	ResumeThread 函数	303
TranslateMessage 函数	302	SuspendThread 函数	303
7. 定时器类	302	TerminateThread 函数	303
KillTimer 函数	302	9. 其他	304
SetTimer 函数	302	CloseHandle 函数	304
8. 线程操作类	302	MessageBeep 函数	304
CreateThread 函数	302	MessageBox 函数	304
ExitThread 函数	303		

第1章 Windows 操作系统概述

多年来,微软(Microsoft)公司的 Windows 操作系统独领风骚,占据了全世界 80%以上的市场,用户总量数以亿计。究其原因,这固然与 Windows 良好的性能和功能有关,但更重要的是微软公司采用了开放软件开发技术的策略,鼓励支持软件企业和用户开发应用软件。这一策略造就了大批 Windows 程序设计专业人才,进而推动了 Windows 应用软件的大发展。在今天,Windows 程序设计技术成为广大软件开发人员的基本技能,掌握该项技术的软件开发人员能够在各个应用开发领域充当主角。

事实上,伴随操作系统的发展,微软公司也在同步发展相应的软件开发平台,在 Windows 不断增强功能和翻新版本的同时,相应的开发工具和开发支持系统也在迅速发展。早期,伴随 MS-DOS 操作系统发展的是汇编、BASIC、C 语言系统以及 DOS 系统调用接口等等,它们共同组成了传统意义的开发平台。从技术上看这一平台并无新意与过人之处,但却有着无可比拟的潜在优势,这就是:开发技术完全公开并且可以随意使用。当 Windows 推向市场时,微软公司不失时机地推出一系列的开发工具和公开系统软件的“秘密”,包括以 WIN16 和 WIN32 为代表的 Windows 应用开发接口(API)、软件开发工具包(SDK)、多媒体开发工具包(MDK)以及相应的 Windows C(Microsoft Visual C)、Windows BASIC(Microsoft Visual Basic)、Windows 数据库(Microsoft Visual FoxPro)等等。这些开放的技术和开发工具组成了先进的 Windows 软件开发平台,使应用软件开发能够充分利用成千上万的 Windows 内部功能,在遵循 Windows 开发技术规范的基础上发挥自己的创造才能,在十分方便的 Windows 开发平台上编写出丰富多彩的应用软件。

Windows C 是一种全新意义的程序设计语言,是描述 Windows 软件技术细节的主要语言,当然也是开发 Windows 应用软件的主要工具之一,它甚至于是其他 Windows 语言的“元语言”。Windows 的系统功能模块大多是 C 语言的模块,而 C 模块总能够在其他语言中调用,比如在 VB 和 FoxPro 程序中能够调用 Windows 系统函数。Windows C 又是某些语言的“平台语言”,例如 Oracle 数据库采用 C 函数形式提供功能模块,以 C 程序作为宿主程序通过调用自己的函数实现数据库管理。Windows C 从传统 C 发展而来,用传统 C 的观点看 Windows C 可能会不得要领。不过了解 Windows 程序设计可以结合了解 Windows C,而了解 Windows C 的最佳途径也莫过于从了解 Windows 程序设计起步;事实上如果掌握了 Windows 程序设计方法,Windows C 的方法将不攻自破。今天,Windows 开发高手往往也是 Windows C 的行家。

目前的 Windows 应用软件开发平台大多是“可视(Visual)”的,它往往是一个集成了下列系统可用资源和开发工具的综合性开发平台:

- Windows 语言的源程序编辑器和编译器;
- 程序调试工具,包括源程序语法检查、可执行程序修改和运行监视等等;
- 系统函数库和系统函数开发工具;
- 资源管理器,包括图形化窗口及组成元素的多种对象的编辑器;
- 可选择并构成具体语句或源程序结构的例程库及 Help;

- 应用程序 Help 和 Setup 开发工具包;
- 其他功能。

Windows 的程序设计语言,包括 C、Basic、FoxPro 等等,都被称之为“面向对象”的语言。所谓“对象”其实是指 Windows 的规范部件,包括各种窗口、菜单、按钮、对话框及程序模块等等。这些多样化的“对象”能够充分满足构成应用软件操作界面的需要,因此编写 Windows 程序相当一部分工作是在创建对象和为对象属性赋值。Windows 对象在具有规范形态的基础上还具有规范的操作模式,能够对鼠标或键盘的规范操作产生规范的程序分支响应。用户可以采用传统的源程序代码编写方法编写程序,也可以采用 Windows 特色的交互式操作方法构造程序。采用交互式方法,可视的开发平台给出许多可选用的对象,程序员选择所需要对象并为对象的属性确定参数值,由此搭建起应用程序的“大模样”,在这个“大模样”中,程序员进一步编写必要的细节代码段,最后构成完整的应用程序。

为了编写 Windows 应用程序,程序员总是要掌握和了解系统有哪些对象和如何使用对象,当然也需要会编写程序。因此,Windows 程序设计总是需要有较多的概念知识以及相应的应用知识。比较传统程序设计方法,需要掌握的知识可能多了些,但这是必要的,是支持用户高效率地开发高水平的应用程序的基础。

在深入介绍 Windows 程序设计方法之前,首先介绍有关 MS-DOS 及不同 Windows 版本的发展历程,这些内容对于初学 Windows 应用程序开发的读者了解基础开发知识是必要的。本章后面的内容从以下两个方面概述 Windows 操作系统:

1. Windows 操作系统的发展历程;
2. Windows 操作系统的主要特点。

1.1 发展历程

微软(Microsoft)公司的 Windows 系统是从 MS-DOS 磁盘操作系统发展出的图形界面操作系统,其典型的版本是 Windows 3.X。源于 MS-DOS 的 Windows 3.X 依赖 MD-DOS,适合于当时流行的 Intel 386/486 的 PC 机硬件环境。而后,在 Windows 3.X 的基础上,微软公司推出了适合于新一代“奔腾”处理器的 Windows 95,以及今年 6 月推出的 Windows 98。此外,在发展普通个人型 PC 机的操作系统之外,微软公司一直在发展企业 Windows 系统,即 NT 系列。伴随 Windows 系统的发展,Windows 程序设计系统也有相应的发展。

1985 年,在 CPU 和图形显示技术新发展的基础上,微软公司针对 MS-DOS 的字符显示和单任务处理的不足,推出了图形用户界面(GUI)的 Windows 操作系统 Windows 1.1 版。1990 年,推出了走向成熟的图形窗口与“DOS 窗口”并存的多任务系统 Windows 3.X,该版本奠定了视窗型操作系统的基础,其主要的技术规范一直沿用到今天。其后,微软公司在 Windows 3.X 上逐步扩充了 32 位访问和 Microsoft 网络功能,其代表是 Windows For Workgroups,这一版本使 Windows 3.X 进一步成为适应多方需求的流行操作系统。

1995 年 8 月,微软公司再次迎合 PC 机硬件新发展,综合多年成就,推出了全新意义的 Windows 95。Windows 95 摆脱了 Windows 对 DOS 的依赖,将 DOS 作为 Windows 的局部模块纳入系统,全面实现 32 位内存访问,将 Microsoft 网络并入到 Windows 系统中,同时还全面更新了操作显示界面,引入“即插即用”功能等许多先进技术,达到了相当高的技术水平。

可以说 Windows 95 是 PC 机操作系统发展史上的一个里程碑,代表着一种质的飞跃。新近推出的 Windows 98,是继 Windows 95 之后的又一重要产品,其主要特点是将 Internet 的应用软件纳入系统,使 Windows 向着公用网络操作系统的方向向前发展了一大步。

在所有的发展中,Windows 程序设计的语言、工具、接口以及开发方式和资源环境也发生了很大的变化,可用于开发的系统功能越来越多,标准化水平越来越高。不过,Windows 程序设计的基本功能特点一直保持不变。

1.2 功能特点

在 Windows 版本系列中,下列特点是始终坚持并不断发展的:

1. 窗口化的图形界面。
2. 多任务方式的运行环境。
3. 虚拟化的设备接口,如图形设备接口(GDI),即与设备无关的图形化显示模式,使多样化的图形硬件和软件设备都能够运行 Windows。
4. 以虚拟内存为核心的内存管理。
5. 网络功能及应用程序,包括 Microsoft 网络、通用基础网络协议和 Internet 网络等。
6. 多媒体功能及应用程序,包括图形、图像、声音、动画、视像处理和开发工具等。
7. 功能丰富的用户管理工具和实用软件。

Windows 95/98 在上述特点的基础上更上一层楼,实现了质的飞跃,其进步大致包括以下几个方面:

1. 先进的抢先式多任务处理功能

Windows 95 采取抢先式的多任务方式,这种方式使系统中所有的进程都能共享 CPU,从而避免了 Windows 3. X 版本中由于采用协作式的多任务方式使某个进程独占 CPU 而导致死机的情况。

2. 良好的兼容性

Windows 95 本身是 32 位平台的操作系统,但同时也支持 16 位的应用程序和 DOS 程序,具有良好的兼容性。

3. 高效的内存管理

Windows 95 采用全新的 32 位线性寻址方式。而且通过采用虚拟内存技术,使一个独立的 32 位应用程序可访问高达 4GB 的存储器地址空间,为应用程序的运行提供了近乎于无限的存储空间。

4. 其他特点

- (1) 系统的健壮性增强,易于维护;
- (2) 即插即用(PNP)技术;
- (3) 更加完善的对象连接嵌入(OLE)技术;

(4) 长文件名(最多 255 个字符);

(5) 内嵌的网络系统, 在 Windows 95 中嵌入了 Microsoft 网络, 在 Windows 98 中嵌入了与 Internet 网络有关的功能;

(6) 内嵌的功能更强的多媒体功能, 如 CD 音乐和 MPEG 视频播放。

在今年 6 月推出的 Windows 98 系统中, 微软的网络浏览器 Internet Explorer 正式纳入系统, 使 Windows 系统正式具备了 Internet 特色。长期以来, Windows 系统的网络功能总是局限于传统网络范围之内, 包括点对点通信、电话网数据通信、Microsoft 局域网及电子邮件等等。在发展 Windows 95 之后, 微软公司果断调整方向, 开始向 Internet 大举进发, 在新一代产品的改进与扩充中, 增加 Internet 的 WWW(World Wide Web) 网络功能是其最重要的内容, 其核心就是浏览器 Explorer, 包括 Internet 的电子邮件、电子新闻、电子会议等在内的完备的 Internet 功能。包含 Explorer 的 Windows 95 和 Windows 98 还支持一种语法格式类似于 Windows Visual Basic 语言的 VBScript, 它是微软的 JavaScript, 支持 Internet 程序的运行与开发。

可以说, Windows 95/98 为程序设计人员提供了一个很“现代化”的开发平台和一个包括 Internet 网络在内的宽阔的开发环境。

最后需要说明, 发展中的 Windows 应用范围越来越大, 功能模块包罗万象, 但程序设计的核心依然是针对窗口对象的编程方式, 它自 Windows 诞生之日至今一直未变。绝大部分情况下, 开发 Windows 应用软件仍然要从编辑与操作窗口开始, 万变不离其宗。

第2章 开发应用程序预备知识

Windows 应用程序与 DOS 应用程序有很大区别,掌握编程方法首先必须了解以下内容:

1. 窗口的概念;
2. 事件驱动的概念;
3. 消息及其在编程中的使用;
4. 对象与句柄;
5. 匈牙利表示法。

2.1 窗口

学习 Windows 编程的第一步是掌握窗口知识。窗口知识中包含着 Windows 程序设计的“精髓”,全面地了解这方面的知识是掌握 Windows 程序设计的入门捷径。

2.1.1 性质

窗口是 Windows 应用程序基本的操作单元,是应用程序与用户之间交互的接口环境,也是系统管理应用程序的基本单位。

编写一个 Windows 应用程序首先应创建一个或多个窗口,随后应用程序的运行过程即是窗口内部、窗口与窗口之间、窗口与系统之间数据处理与数据交换的过程。可见,窗口的概念不仅包括其外观,还包括其内部的结构与功能。

2.1.2 外观

图 2-1 是一个标准的 Windows 95 窗口。

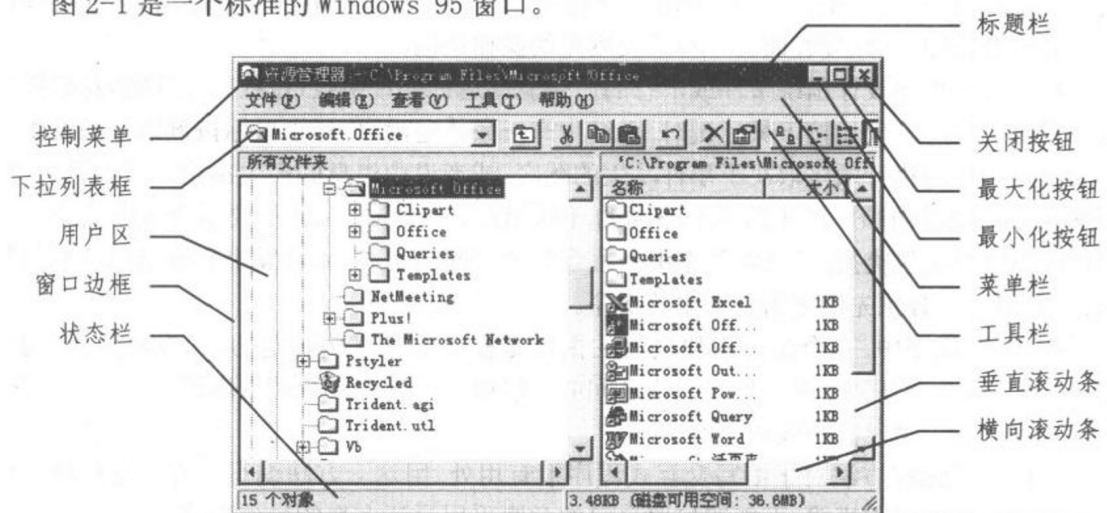


图 2-1 Windows 95 窗口

Windows 窗口的显示形态总为矩形, 内部包含多种类型的组成元素; 窗口中的组成元素共同决定了窗口的外观特征。

程序员在创建窗口时应根据需要在设计窗口的外观特征, 正确地编写能够响应元素动作的窗口处理函数。设计时必须分清各组成元素在窗口中的作用, 切不可张冠李戴。

2.1.3 管理

窗口由应用程序创建, 由应用程序和 Windows 系统协同实施管理。这一点是理解窗口概念的关键。

Windows 管理窗口, 包括显示窗口的位置与方式, 处理窗口的标准特性参数, 以及完成多种由用户初始化程序指定的设置。这些设置直接产生控制窗口的功能。应用程序管理窗口的其他操作工作, 特别是负责管理窗口内用户区的操作, 即窗口边框以内容纳用户数据的区域操作。

Windows 系统通过解释窗口描述语句参与应用程序的窗口管理。在下一章给出的窗口示例程序中, 只包含窗口特性的描述, 而没有如何生成窗口和操作窗口的内容, 但 Windows 系统能够通过解释窗口特性的描述生成窗口并提供基本的窗口操作功能, 如关闭, 最大化, 最小化等。

Windows 系统的基本风格是通过窗口描述而使系统生成窗口, 就是说, Windows 提供包括窗口在内的许多通用元素, 程序员用简单的描述语句调用这些元素以代替编写生成和操作这些元素的复杂程序, 从而将主要精力集中于专业化程序功能的开发。

2.2 事件驱动

编写传统的 MS-DOS 应用程序, 大多采用以函数(过程)为主程序或子模块的结构, 每个函数(过程)都有明确的开始和结束位置。主程序通过语句调用子模块实现程序描述的功能, 程序的运行、调用顺序以至整个执行顺序在程序员编程时已经确定。因此说 DOS 应用程序是一种顺序的, 结构化的, 面向过程的程序设计方法。对于这样的应用程序, 相应的操作过程也必须分次序、分层次, 而且不得违反规定的操作规范。

Windows 程序设计围绕事件或消息的产生驱动运行处理函数(过程)。所谓消息是描述事件发生的信息。例如按下鼠标键时, 系统生产一条特定消息, 标识鼠标按键事件的发生。Windows 程序的执行顺序取决于事件发生的顺序, 或者说由消息的产生决定。可以认为, 程序的执行顺序是由顺序产生的消息驱动的, 但是消息的产生往往并不要求有次序之分。程序员可以区分消息类型, 并针对类型编写程序处理接收的消息, 或者发出其他消息以驱动其他程序, 但是不必预先确定消息产生的次序。

在 DOS 程序中, 多个处理模块之间的衔接是程序本身直接确定的, 孰先孰后, 泾渭分明。在 Windows 程序中, 多个处理模块之间的衔接要经过操作系统用发送消息的方式联系, 运行次序取决于消息次序或事件次序。

事件驱动编程方法对于编写交互式程序很有用处, 用这一方法编写的程序使程序用户无须恪守死板的操作模式, 而能够按照自己的意愿采用灵活多变的操作形式。

例如, 一个统计某商场第一季度平均销售量的程序。图 2-2 是面向过程的程序流程。

按照流程,用户必须按次序输入一至三月份的销售量。如果输入三月份销量后,发现一月份的数据有误,则必须从一月份开始重新输入,这给用户操作带来不便。

采用图 2-3 所示的事件驱动程序结构,用户可在多个窗口之间切换,分别处理每个月的销售量,并可在各个窗口中增加或修改数据而不影响其他窗口。显然,采用这种结构的应用程序为用户提供了方便灵活的操作方式。

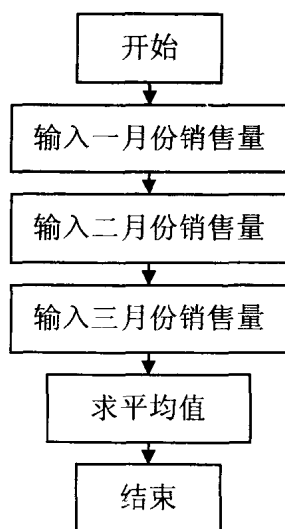


图 2-2 面向过程流程图

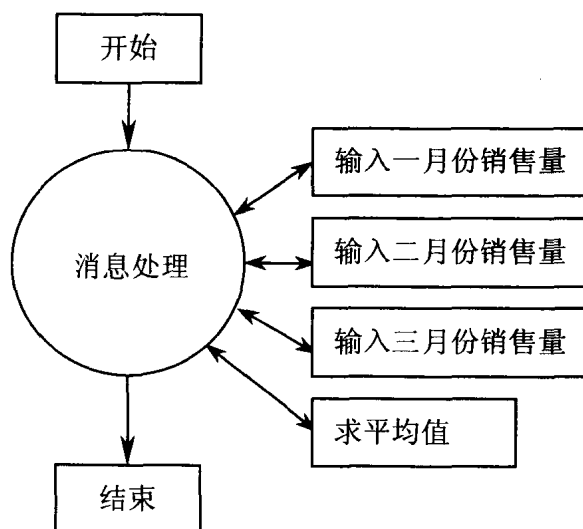


图 2-3 事件驱动流程图

2.3 消息

Windows 应用程序通过发送和接收统一格式的消息与其他应用程序和 Windows 系统进行信息交换。例如,在应用程序窗口内,当用户单击时,Windows 系统向应用程序发送一条“单击”消息,其中包含产生消息的相关信息。应用程序接收消息后,识别其中包含的信息,确定用户请求,再调用相应的过程响应该请求。

在 Windows 系统中,每一特定事件的发生都对应的一条特定的消息。例如,对单击事件产生一个名为 WM_LBUTTONDOWN 的消息。

2.3.1 组成

一条 Windows 消息由三部分组成,即:消息号,字参数和长参数。

1. 消息号

消息号由事先定义的消息名标识,如上述的 WM_LBUTTONDOWN。消息名通常可以直观地说明消息的含义和来源。

Windows 系统定义了一系列的消息号及消息名,程序员可以使用这些系统消息,也可以使用自定义的消息。