



Windows 3.1

编程指南

[美] Tim Farrell

Runnoe Connally

李莉 晓淀 曾建 译
欣力 校



清华大学出版社



Windows 3.1 编程指南

[美] Tim Farrell, Runnoe Connally

李莉 晚浣 曾建 译
欣力 校

清华大学出版社

(京)新登字 158 号

Programming in Windows 3.1, Second Edition

Tim Farrell, Runnoe Connally

Authorized translation from the English language edition published by Que Corporation.

Copyright © 1992 by Que Corporation

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission in writing from the Publisher.

Chinese language edition published by Tsinghua University Press.

Copyright © 1994 by Tsinghua University Press.

本书英文版由 Prentice Hall 出版社属下的 Que 计算机图书出版公司于 1992 年出版。版权为 Que 所有。Que 将本书的中文版专有出版权授予清华大学出版社。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。

图书在版编目(CIP)数据

WINDOWS 3.1 编程指南/(美)法雷尔(Farrell, T.),康内利(Connally, R.)编著;李莉等译.——北京:清华大学出版社,1994

ISBN 7-302-01499-X

I. W… II. ①法… ②康… III. 窗口软件, Windows 3.1-编辑程序-程序设计-手册 IV. TP313

中国版本图书馆 CIP 数据核字(94)第 03353 号

出版者:清华大学出版社(北京清华大学校内,邮编 100084)

印刷者:清华大学印刷厂

发行者:新华书店总店北京科技发行所

开 本:787×1092 1/16 印张:31.75 字数:753 千字

版 次:1994 年 9 月第 1 版 1994 年 9 月第 1 次印刷

书 号:ISBN 7-302-01499-X/TP·604

印 数:0001 - 8000

定 价:46.00 元

引 言

Microsoft Windows 为程序员提供了极为方便的编程环境。使用 Windows 编程不用考虑与显示屏、打印机、绘图仪和定位设备等的接口,编出的应用程序可以运行在多个平台上,并可以用于 Microsoft Windows 的将来版本。

Microsoft Windows 为应用程序开发人员和用户提供了一个新的天地,随着计算机硬件与软件的发展,开发者不再需要为特定的机器开发程序,Windows 应用程序可以不加修改地运行在不同的硬件上。

总起来说,本书的目的在于缩短 Windows 编程的进程,当然也减少开发第一个 Windows 应用程序的时间。如果已经编过 Windows 应用程序,本书将解答开发 Windows 应用程序中遇到的一切问题。本书是按照这些问题为线索组织的,因此也是一部快速参考手册。

谁使用本书

如果要编写 Windows 应用程序,则本书详细介绍了什么是 Windows 应用程序和如何编写它。本书含有编写基本 Windows 应用程序的所有信息,而不必阅读 Windows 软件开发工具(SDK)手册。

如果你有 Microsoft SDK,则会发现本书对 Microsoft C Compiler 和 SDK 部分的信息作了提炼。本书按易查易用的风格组织这些信息。

对于有 Windows 应用程序开发经验的读者,本书特别注重对将来开发有用的参考信息。

无论你是管理 Windows 程序,还是购买 Windows 作为标准工作环境,都可以从本书中获得对 Microsoft Windows 的进一步理解。

使用本书的预备知识

很明显,本书不是为那些从未接触过计算机的人准备的,尽管书中清楚地解释了建立 Windows 应用程序的所有事项,但是,要有效地使用本书应有两个预备知识:

- 必须有丰富的 C 或 PASCAL 语言的编程经验,因为本书中的每一应用程序都涉及到 C。

- 熟悉 Microsoft Windows 零售软件包也很有用,因为这些使用经验有助于为 Windows 用户开发程序。

预备知识仅此而已。建立基本 Windows 应用程序的其它知识,本书中都有详尽的介绍。

本书约定

本书使用了一些约定来帮助读者迅速找到所需要的信息：

■ 本书是以开发的应用程序为线索组织章节的,因此增加到已有代码清单的代码都增亮显示(书中则以黑体表示),从第 6 章开始的代码清单中,增加或修改的行用黑体单间隔字体表示。

■ 变量的命名采用一种称之为匈牙利表示法(Hungarian Notation)的方法,匈牙利表示法为纪念 Microsoft 程序员 Charles Simonyi 而得名。每个变量的开始一个或多个小写字符表示变量的数据类型。例如,lpsz MenuName 表示变量 MenuName 是一个指向以零结束的字符串的长指针。表 0.1 列出了一些经常使用的前缀。

表 0.1 匈牙利表示法

前 缀	数 据 类 型
b	BOOL(整型)
by	BYTE(无符号字符)
c	字符
fn	函数
g	全局
h	句柄
i	整数
n	短整数或整数(数字)
x,y	短整数(x,y 坐标)
cx,cy	短整数(除 c 表示计数外,与上项相同)
l	LONG(长整数)
w	WORD(无符号整数)
dw	DWORD(无符号长整数)
p	指针
s	字符串
sz	以 0 结束的字符串

综述

本书通过例子来讲述编程技术,详细介绍建立一个简单 Windows 应用程序 Editor 的每一步。全书按逻辑顺序组织,从软件、硬件的选择开始,然后介绍设置和安装开发工作环境。

每一章本身都有自己的主题。例如,第 13 章“Windows 中的打印”介绍了有关打印的所有知识。要学习某一特定主题,不必翻来翻去。作为本书信息组织的概观,下面列出每一章的主要内容。

提示：第一部分“开发工具及其概念”概述了创建 Windows 应用程序的基本硬件和软件工具,从硬件需求到基本 MS-DOS 开发工具,还介绍了 Windows 风格和面向对象编程技术的基础知识。

第1章“Windows 编程准备”概述了编写 Windows 应用程序的最小软件、硬件要求。本章解释了编写 Windows 应用程序时什么是必须的,什么是不必要的,另外还介绍了 Windows 软件开发工具的正确安装。

第2章“开发工具的使用”包括建立 Windows 应用程序需要的各种 MS-DOS 工具: Microsoft C 编译程序、Microsoft 汇编程序 MASM、Windows 连接程序 LINK、资源编译程序 RC 和程序维护程序 NMAKE。使用 MS-DOS 工具开发 Windows 时应注意 Windows 专用的选择项。

第3章“Windows 编程模式”介绍 Windows 应用程序的一般框架,学习面向对象的编程技术、消息处理概念、多任务、内存管理和资源的使用。学完本章后将对 Windows 应用程序设计的内部结构和概念有清楚的了解。

第4章“Windows 风格指南:窗口和菜单”包括 Windows 中的原理和方法,基本输入输出、选择与导向的一般技术和应用程序窗口与菜单。本章还详细介绍用户界面风格。学完本章,你将会看到一致的用户界面如何提高用户的学习进度。

第5章“Windows 风格指南:对话框”讨论 Windows 图形用户接口中的高级内容,本章主要着眼于组织和使用对话框来输入输出信息,辅助内容有:装载、初始化、定制和帮助。

提示:第二部分“建立 Windows 应用程序”包括两个 Windows 应用程序,一个是简单的程序编辑器,另一个是产生简单图形,建立这两个应用程序时,每一章都以前一章为基础循序进行。

第6章“建立程序 Editor:基础”介绍建立 Windows 应用程序的基本元素,所有例子都围绕着建立一个称为 Edit 的 Windows 应用程序的核心过程,它们建立 Edit 的基础。作为一个 Windows 应用程序范例,以后各章都要用到它。

第7章“生成菜单”以应用程序 Edit 为例建立并使用下拉式菜单。本章解释下拉式菜单定义在资源文件中的语法并描述管理下拉式菜单用 C 编程的注意事项。

第8章“生成对话框”说明设计、建立、初始化和管理工作对话框消息的正确方法。本章介绍了两种类型的对话框:模式对话框和无模式对话框。为学习对话框的建立,将为 Edit 程序建立一个 Save As 对话框和一个 Find 对话框。本章还定义了其它项如复选框、按钮和单选按钮,并说明了建立对话框时资源文件的正确使用。

第9章“Windows 3.1 公共对话框:文件名选择”介绍标准 File Open 对话框的建立。本章还介绍了列表框的概念和如何获得文件目录,可以使用这个对话框在你自己的应用程序中完成用户文件名的输入。

第10章“Edit 类窗口的使用”介绍内部编辑过程。本章说明如何使用 Windows 内部编辑过程,要把这些概念应用到实际应用程序中,必须把 Windows 内部编辑过程插入程序 Editor。

第11章“Windows 下的文件功能”介绍 Windows 下访问磁盘文件的正确途径。本章介绍从磁盘读出信息并传到内存的方法。完成本章后,程序编辑器应用程序就可以读、编辑并保存程序了。

第 12 章“全局内存和剪贴板的使用”教你如何管理全局内存和 Windows 剪贴板。本章将对程序编辑器进行修改以便能直接从剪贴板拷贝文件或拷贝文件到剪贴板,还介绍如何将位图拷贝到剪贴板。

第 13 章“Windows 下的打印”说明用图形设备接口(GDI)向打印机输出正文的方法。本章介绍启动 Windows 假脱机队列和分页 Edit 应用程序中样本文件的技术,并修改程序 GDITEST 演示输出图形对象的方法。

第 14 章“多文本接口(MDI)”介绍 Microsoft 单个应用程序操作多个文本的标准,每一个文本拥有自己的子窗口。本章介绍 MDI 的原理并为程序 Edit 实现多文本接口。

提示:第三部分“高级内容”需要有第一部分和第二部分基础的支持。包括管理全局内存、对象链接与嵌入(OLE)、动态数据交换(DDE)和动态链接库(DLL);还将介绍图形环境下打印的概念和方法以及调试 Windows 应用程序的方法;最后还介绍应用程序的调试、Windows 3.1 的新特性、Windows NT 和第三方 Windows 软件及工具。

第 15 章“用图形设备接口(GDI)显示正文”首先介绍在屏幕上显示正文,还将说明不同字体的使用并用所有可用的字体建立一个样本。

第 16 章“显示图形”解释显示图形对象如矩形、椭圆和弧的过程。本章还将介绍映像方式,Windows 使用它将逻辑单元转换成物理单元,并产生一个简单的程序称为 GDITEST 以说明这些概念。

第 17 章“滚动屏幕内容及控制鼠标事件”介绍如何使用滚动条和鼠标。本章中将修改程序 GDITEST 使得屏幕上的内容可以移动。本章结束后,可以对 Windows 应用程序中鼠标的使用有个完整的理解。

第 18 章“动态数据交换(DDE)”解释 Microsoft Windows 的最显著特点之一——DDE,它通过 Windows 的消息使不同的应用程序可以交换数据、传递遥控命令;由于 Windows 是基于消息传递的结构,所以在应用程序之间通过消息传递来传送信息是合理的。

第 19 章“对象链接与嵌入的用户接口指向”描述插入链接与嵌入对象、观看对象、编辑对象和修改属性的方法。广义上来说,OLE 是 DDE 的超集。

第 20 章“动态链接库”讨论 DLL——使应用程序可以共享代码和资源的可执行模块,这种共享可以是在应用程序的不同模块之间,也可以在不同的应用程序之间。本章包括究竟什么是 DLL、什么时候使用它和如何建立等方面的内容,这种库在运行时链入而不是在编译时链入,它们是一个模块程序。

第 21 章“调试 Windows 应用程序”介绍 Windows 调试工具 HeapWalk 和 CodeView 的正确使用。本章说明如何用 HeapWalk 分析内存分配问题,并解释 CodeView 在设置断点、原码跟踪方面的使用,这是在阅读第二部分时可以参考的章节。

第 22 章“TrueType 字体和其它新特性”介绍 Windows 3.1 的新特性。尽管 Windows 3.1 大部分改进是针对前版本的不合理之处,Microsoft 还是在 Windows 3.1 中增加了许多新功能,在本章中介绍包括 TrueType 字体(与 Adobe 的 PostScript 字体类似)、版本库、稳定性测试、屏幕保护、Watson 博士 UAE 工具和打印。

第 23 章“Windows NT”介绍即将推出的 Windows New Technology,由于 Microsoft 有关这方面的材料超过 1500 页,所以本章只能作为一个概要介绍,这个概要包括 Microsoft 操作系统策略、32 位应用程序编程接口(API)和一些编程注意事项。将要产生的程序 Editor 也可以在 Windows NT 上运行。

第 24 章“第三方工具和程序集”详细介绍 Microsoft 以外开发者提供的工具和程序集,这些产品中大部分要依靠 Microsoft SDK,但是,也有些工具不需要 SDK。

阅读指南

无论对 Windows 有多少了解,都应以前向后的顺序阅读本书。如果对 Windows 编程已有所了解,则可以浏览一下第一部分,确保系统配置正确。即使有了相当的 Windows 使用经验,重温一下某些基本原理也没有什么坏处。由于本书各部分的复杂程度和循序渐进的组织结构,我们强烈推荐按顺序阅读每一章节。一旦对 Windows SDK 有了基本了解,就可以确定第三部分的开发工具中哪一个最适合你。

目 录

引言	VII	本书约定	IX
谁使用本书	VII	综述	IX
使用本书的预备知识	VIII	阅读指南	XI

第一部分 开发工具及其概念

第 1 章 Windows 编程准备	1	1.7 测试与调试	16
1.1 硬件要求	1	1.7.1 为 CodeView 设置一个 调试终端	17
1.2 软件要求	2	1.7.2 在单监视器上使用 CodeView	17
1.3 Windows 概述	2	1.7.3 为调试设置一单色 监视器	17
1.3.1 Windows 与 DOS 的比较	3	1.8 本章小结	17
1.3.2 Windows 接口	4	第 2 章 开发工具的使用	19
1.3.3 Windows 库	5	2.1 用 Microsoft C 编译源代码	19
1.3.4 开发工具与实用程序	5	2.1.1 为 Windows 函数使用 Windows.h	20
1.3.5 建议	7	2.1.2 选择内存模式和浮点库	20
1.3.6 Windows 环境小结	8	2.1.3 为 Windows 使用 Pascal 调用约定	20
1.4 Windows 编程的目录结构	8	2.1.4 在 Windows C 应用程序中 使用 WinMain()	21
1.4.1 子目录 WINDOWS\WIN	9	2.1.5 回调函数	21
1.4.2 子目录 WINDOWS\ BIN	10	2.1.6 使用 Windows 的 C 运行 库函数	22
1.4.3 子目录 WINDOWS\ LIB	10	2.1.7 浮点库与 Windows	23
1.4.4 子目录 WINDOWS\ DEBUG	11	2.1.8 使用环境和命令行参数	23
1.5 安装 Windows 开发环境	11	2.2 编译 Windows C 应用程序	23
1.5.1 安装 Windows	12	2.2.1 使用 MASM 编译汇编 文件	25
1.5.2 安装 Microsoft C	12	2.3 使用 LINK 连接应用程序	27
1.5.3 安装软件开发工具 (SDK)	13	2.3.1 一个简单的模块定义 文件	27
1.6 优化 Windows 开发环境	14	2.3.2 一个复杂的模块定义 文件	29
1.6.1 减小 Windows.h 的大小 增加编译速度	14	2.3.3 使用 LINK 产生 Windows	
1.6.2 以 Windows 为开发环境 配置系统	14		
1.6.3 设置.PIF 文件	15		
1.6.4 WIN.INI 设置	15		
1.6.5 SYSTEM.INI 设置	16		

应用程序	30	4.4.3 窗口的组成	79
2.3.4 使用语句 ICON	33	4.4.4 基本的窗口操作	83
2.3.5 用 STRINGTABLE 定义 字符串表	33	4.5 菜单	87
2.3.6 使用语句 DIALOG	33	4.5.1 下拉式菜单	87
2.3.7 使用语句 MENU	34	4.5.2 弹出式菜单	87
2.3.8 运行资源编译器	36	4.5.3 级联式菜单	88
2.4 用 NMAKE 维护应用程序	37	4.5.4 菜单的组成	89
2.5 本章小结	38	4.5.5 菜单操作	91
第3章 Windows 编程模式	39	4.5.6 标准菜单	92
3.1 对象和对象编程	39	4.5.7 常用的可选菜单	97
3.2 句柄	40	4.6 本章小结	99
3.2.1 实例	41	第5章 Windows 风格指南:对话框	100
3.3 消息与事件	43	5.1 对话框中的控制	100
3.4 多任务	46	5.1.1 按钮	100
3.4.1 MS-DOS 与 Windows 3.1 的合作多任务	47	5.1.2 复选框	102
3.4.2 Windows NT	48	5.1.3 列表框	103
3.5 内存管理——Windows 风格	48	5.1.4 编辑控制	105
3.6 资源	50	5.1.5 只读弹出式文字域	107
3.7 图形设备接口(GDI)	52	5.1.6 滑动条	108
3.7.1 显示描述表(DC)	52	5.1.7 静态文字域	108
3.7.2 位图	53	5.1.8 分组框	108
3.7.3 裁剪	54	5.1.9 控制标签	108
3.7.4 映射	55	5.2 使用对话框	109
3.8 本章小结	56	5.2.1 对话框的类型	109
第4章 Windows 风格指南:窗口和菜单	57	5.2.2 对话框中的按钮	112
4.1 规则与方法	57	5.2.3 对话框中的字体	114
4.2 基本输入元素	58	5.3 标准对话框	114
4.2.1 鼠标器输入	58	5.3.1 文件操作	114
4.2.2 键盘输入	59	5.3.2 打印	118
4.3 图形用户接口技术概要	63	5.3.3 文字搜索和替换	120
4.3.1 选择	63	5.3.4 拼写	122
4.3.2 控制焦点	67	5.3.5 字符特性	123
4.3.3 移动	68	5.3.6 边界对话框	125
4.3.4 拖放	69	5.3.7 Page Setup 对话框	126
4.3.5 给用户反馈	70	5.3.8 About 对话框	126
4.3.6 编辑控制框	73	5.4 其它问题	127
4.3.7 绘图	74	5.4.1 装载和初始化	127
4.4 窗口	76	5.4.2 用户等级和定制	127
4.4.1 应用程序窗口	76	5.4.3 帮助	128
4.4.2 文本窗口(多文本窗口)	77	5.4.4 国际化考虑	128
		5.5 本章小结	129

第二部分 建立 Windows 应用程序

第 6 章 建立程序 Editor:基础	130
6.1 用 NMAKE 编译应用程序	131
6.1.1 生成资源文件 EDIT.RC	132
6.1.2 生成头文件 edit.h	132
6.1.3 生成模块定义文件 EDIT.DEF	133
6.1.4 生成图标文件 ICON.ICO	134
6.1.5 变量命名约定	135
6.1.6 生成 C 源文件 EDIT.C	136
6.2 从 WinMain()开始执行	139
6.2.1 使用 ghInstance 和 ghPrev- Instance 参数	140
6.2.2 登记窗口类	142
6.2.3 生成一个覆盖窗口	143
6.2.4 用 ShowWindow()显示 一个窗口	144
6.2.5 生成主消息处理循环	145
6.2.6 回顾一下 WinMain()	145
6.3 用 EditWndProc()处理消息	146
6.3.1 使用 Windows 的 Def WindowProc()函数	147
6.3.2 处理 WM_DESTROY 消息	147
6.3.3 处理 WM_PAINT 消息	147
6.4 本章小结	148
第 7 章 生成菜单	149
7.1 给程序 Editor 加入下拉式菜单 ...	149
7.1.1 生成菜单消息处理程序 doCommand()	151
7.1.2 使用 Windows 的 Mess- ageBox()函数	155
7.2 改变菜单项的外观	156
7.2.1 检查菜单项	158
7.2.2 使菜单项无效	158
7.2.3 加亮顶层下拉菜单	159
7.2.4 拷贝菜单标号	159
7.2.5 删除和增加下拉菜单项 ...	160
7.2.6 获取下拉菜单的句柄	160
7.3 生成随机菜单	160
7.4 给菜单加上加速键	165
7.5 本章小结	166
第 8 章 生成对话框	168
8.1 模式对话框	168
8.2 非模式对话框	170
8.3 使用 Dialog Editor (DLGEDIT.EXE)	171
8.4 分析模式对话框的源代码	172
8.4.1 修改 C 源文件增加模式 对话框	175
8.4.2 建立对话框消息处理源 文件 DLGCMD.C	177
8.4.3 处理消息 WM_INITDIALOG	180
8.4.4 处理消息 WM_COMMAND	181
8.4.5 复习模式对话框	182
8.5 分析非模式对话框的源代码	182
8.5.1 修改 C 源文件增加非模 式对话框	184
8.5.2 非模式对话框消息处理 过程 FindBox()	187
8.5.3 FindBox()中消息 WM_ INITDIALOG 的处理	189
8.5.4 FindBox()中消息 WM_ COMMAND 的处理	190
8.5.5 FindBox()中消息 WM_SYSCOMMAND 的处理	190
8.6 本章小结	191
第 9 章 Windows 3.1 公共对话框: 文件名选择	192
9.1 其它公共对话框	192
9.2 使用 Filename 公共对话框	193
9.3 建立文件打开对话框	193
9.4 驱动 Filename 对话框的 C 源代码	198

9.4.1 使用例程		修改 DOCMD.C	241
OpenBoxFN()	201	12.3.1 处理消息	
9.4.2 确认文件名	201	MN_FILETOCLIP	247
9.4.3 测试字符合法性	202	12.3.2 ReadToClip()例程	248
9.5 本章小结	202	12.3.3 处理消息	
第 10 章 Edit 类窗口的使用	203	MN_CLIPTOFILE	250
10.1 Edit 类窗口的概念	203	12.3.4 ClipToFile()例程	251
10.2 用 Edit 类窗口作为子窗口	205	12.3.5 处理消息	
10.3 给程序 Editor 增加一个 Edit		MN_WINTOCLIP	252
类窗口	205	12.4 本章小结	254
10.3.1 修改 EDIT.C 产生		第 13 章 Windows 下的打印	255
一个 Edit 类窗口	210	13.1 打印背后的概念	255
10.3.2 改变 DOCMD.C 使用		13.2 修改程序 Editor 增加	
Edit 类窗口	212	打印功能	257
10.4 本章小结	217	13.3 编辑程序 Editor 中的	
第 11 章 Windows 下的文件功能	218	打印处理	267
11.1 Windows 文件功能的原则	218	13.4 本章小结	270
11.2 文件操作的用户接口	219	第 14 章 多文本接口	271
11.3 修改程序 Editor 增加		14.1 MDI 应用程序结构	271
文件操作	222	14.2 MDI 应用程序初始化	272
11.3.1 处理消息 MN_		14.2.1 窗口类的注册	272
NEW	227	14.2.2 窗口创建	284
11.3.2 处理消息 MN_		14.3 主消息环	285
OPEN	228	14.4 边框窗口函数	285
11.3.3 处理消息 MN_		14.5 子窗口函数	286
SAVE	229	14.6 用子窗口协作数据	286
11.3.4 处理消息 MN_		14.6.1 保存数据	287
SAVEAS	230	14.6.2 使用属性	287
11.3.5 SaveTheData()例程	231	14.6.3 子窗口控制	287
11.3.6 ReadTheData()例程	232	14.6.4 创建子窗口	288
11.4 本章小结	234	14.6.5 撤消子窗口	288
第 12 章 全局内存和剪贴板的使用	235	14.6.6 子窗口的激活	288
12.1 使用剪贴板和全局内存连接	235	14.6.7 子窗口在屏幕上的	
12.2 增加全局内存操作和剪贴板		位置	295
功能	238	14.7 本章小结	295
12.3 为全局内存和剪贴板功能			

第三部分 高级内容

第 15 章 用图形设备接口 (GDI)		15.3 列出 Windows 中的各种字体	302
显示正文	296	15.3.1 修改 EDIT.DEF 和	
15.1 Font 是什么?	296	DOCMD.C	303
15.2 第一次显示正文	297	15.3.2 calcFont 过程	305

15.3.3 列出字体部分小结	308	光标形状	352
15.4 维护显示窗口	308	17.6.2 修改 GdiWndProc(), 处理鼠标消息	352
15.4.1 处理 AW_PAINT 消息	312	17.6.3 处理 WM_LBUTTONDOWN DOWN 消息	355
15.4.2 对过程 displayFont 的 理解	314	17.7 处理 WM_MOUSEMOVE 以及 WM_LBUTTONUP 消息	356
15.4.3 有关重画的提示	314	17.8 鼠标处理部分小结	356
15.5 本章小结	315	17.9 增加打印功能	357
第 16 章 显示图形	316	17.10 本章小结	361
16.1 逻辑与物理坐标	316	第 18 章 动态数据交换(DDE)	361
16.1.1 MM_TEXT 映象模式 的使用	316	18.1 数据交换	361
16.1.2 使用 MM_ISOTROPIC 映象模式	318	18.1.1 利用剪贴板进行传输 ..	361
16.1.3 使用 MM_ANISO- TROPIC 映象模式	320	18.1.2 动态连接库	362
16.1.4 坐标系统部分小结	321	18.1.3 动态数据交换	362
16.2 创建 GDITEST 程序	322	18.1.4 DDE 的用途	362
16.2.1 GDITEST 中使用的 全局变量	333	18.1.5 一个最终用户使用 DDE	363
16.2.2 处理 Shapes 菜单 消息	333	18.2 DDE 的有关概念	363
16.2.3 响应消息 WM_PAINT	335	18.2.1 客户和服务端对话	363
16.2.4 处理 Map Mode 菜单 消息	336	18.2.2 应用程序名称,课题 名称以及项目名称	364
16.3 本章小结	338	18.2.3 热的与温的数据链路 ..	364
第 17 章 滚动屏幕内容和控制鼠标事件	339	18.3 DDE 消息	364
17.1 滚动的理论	339	18.4 DDE 的消息流	365
17.2 为 GDITEST 增加滚动条	341	18.4.1 开始一个对话	365
17.3 修改 WinMain()函数增加滚动条 处理功能	342	18.4.2 单个项目的传输	367
17.4 修改 SetMap()移动逻辑坐标 原点	342	18.4.3 建立一个永久性数据 链路	370
17.4.1 修改程序 doCommand() 显示和取消滚动条	343	18.4.4 远程应用程序命令	373
17.4.2 修改 GdiWndProc()处理 滚动条消息	346	18.4.5 对话结束	376
17.5 鼠标移动的理论	349	18.4.6 动态数据交换小结	377
17.6 在 GDITEST 应用程序中增加 鼠标控制功能	351	18.5 DDE 管理库	377
17.6.1 修改 GdiInit(),改变		18.5.1 DDEML 概述	377
		18.5.2 DDE 管理库概念	378
		18.5.3 串管理	379
		18.5.4 服务器名称服务	379
		18.5.5 对话管理	380
		18.5.6 DDE、DDEML 以及 OLE	381
		18.6 本章小结	382
		第 19 章 对象链接与嵌入的用户接口	

指南	383	20.1.2 应用程序和 DLL 模块	404
19.1 基本术语和概念	383	20.1.3 任务与 DLL	404
19.2 OLE 的好处	384	20.1.4 栈和 DLL	405
19.3 最终用户角度的 OLE	385	20.1.5 Windows 如何找到 DLL	405
19.3.1 插入嵌入对象	385	20.2 编写自己 DLL 的理由	406
19.3.2 察看对象	388	20.2.1 应用程序中的代码和 资源共享	406
19.3.3 编辑对象	388	20.2.2 不同的市场体制	407
19.3.4 从非 MDI 服务器中 更新对象	389	20.2.3 定制控制	407
19.3.5 从 MDI 源中更新 对象	389	20.2.4 系统范围消息过滤	412
19.3.6 包含未完成 OLE 对象 的客户中的操作	390	20.2.5 设备驱动程序	413
19.3.7 访问对象动词:通过 编辑激活	390	20.2.6 项目控制和开发	413
19.3.8 Links 对话框	393	20.3 DLL 的产生	414
19.3.9 用于链更新的其它 对话框	395	20.3.1 源文件产生	414
19.4 对象链接与嵌入的实现	396	20.3.2 建立 DEF 或模块 定义文件	417
19.4.1 OLE 数据传输	396	20.3.3 建立 NMake 文件	418
19.4.2 OLE 库的通信	396	20.4 应用程序访问 DLL 代码	420
19.4.3 剪贴板约定	397	20.4.1 建立库函数原型	420
19.4.4 使用包	398	20.4.2 输入库函数	420
19.4.5 使用服务器的用户 接口	399	20.5 Windows 对象所属规定	422
19.5 选择 DDEML 或 OLE 库指南	400	20.6 本章小结	423
19.5.1 DDE、OLE 消息协议 以及库之间的比较	400	第 21 章 Windows 应用程序调试	424
19.5.2 什么时候适于选择 OLE 库,而不是选择 DDEML	400	21.1 CodeView	424
19.5.3 什么时候选择 DDEML, 而不是 OLE 库	401	21.1.1 CodeView 设置	424
19.5.4 什么时候选择 OLE 和 DDEML	402	21.1.2 CodeView 窗口	425
19.5.5 多样化的 OLE 与 DDEML 版本	402	21.1.3 在 CodeView 中执行 程序	425
19.6 本章小结	402	21.1.4 断点	426
第 20 章 动态链接库	403	21.1.5 变量检查	427
20.1 什么是 DLL	403	21.2 Heap Walker	428
20.1.1 DLL 和输入 (Import)库	403	21.3 Spy	428
		21.4 本章小结	429
		第 22 章 TrueType 字体和其它新特点	430
		22.1 TrueType 介绍	430
		22.2 有关字体背景	430
		22.2.1 字体到底是什么?	430
		22.2.2 Windows 字体	432
		22.2.3 可移植性问题	434
		22.2.4 TrueType 的好处	435

22.3	轮廓字体实际如何工作	435	23.2	操作系统战略	444
22.3.1	概述	435	23.2.1	ACE 建议(initiative) ...	446
22.3.2	字符间隔	435	23.2.2	基本原理	446
22.3.3	字体枚举	436	23.2.3	系统组成	447
22.4	字体量化问题	436	23.2.4	Executive 特点	448
22.4.1	可读性与正确性	436	23.2.5	总结;兼容性	451
22.4.2	量化;视觉和非线性	436	23.3	Win32 概述	451
22.4.3	ABC 宽度必须正确 使用	437	23.3.1	Windows 结构	452
22.4.4	属性可影响字体高度 ...	437	23.3.2	Windows Win32 API	452
22.4.5	TrueType 使用问题	437	23.3.3	基础操作系统核心	453
22.5	移植层次	438	23.3.4	GDI 新项;路径、转换 和独立设备色彩	453
22.5.1	获得 WYSIWYG	438	23.3.5	系统类与窗口系统	453
22.5.2	打印机可移植性	438	23.3.6	网络扩展	454
22.5.3	文本可移植性	438	23.3.7	16 位和 32 位 API 间的 兼容性	454
22.5.4	字体可移植性	439	23.3.8	总结;Win32 API 未来	454
22.6	Windows 3.1 中各种新特点	439	23.4	Win32 的编程根据	455
22.6.1	版本 Stamping 库	439	23.4.1	二进制兼容性	455
22.6.2	应用程序的重点测试 ...	439	23.4.2	32 位 Windows 一瞥 ...	457
22.6.3	全局锁定(Locking)	440	23.4.3	在实模式中的指针 处理	460
22.6.4	改进的 File Manager	440	23.5	本章小结	476
22.6.5	Start-Up 实用程序	440	第 24 章	其它工具和实用程序	478
22.6.6	屏幕保护程序与 Dr. Watson	440	24.1	工具分类	478
22.6.7	Windows SDK 一瞥	440	24.2	其它制造商介绍	479
22.6.8	用 Escape 打印	441	附录 A	资源文件语法的速查手册	491
22.6.9	拖动与下拉	441	A.1	术语定义	491
22.6.10	压缩实用程序	441	A.2	STRINGTABLE 语句	491
22.7	本章小结	442	A.3	ACCELERATOR 语句	492
第 23 章	Windows NT	443	A.3.1	MENU 语句	492
23.1	Windows NT 概述	443	A.3.2	DIALOG 语句	493
23.1.1	设计目标	443			
23.1.2	Windows 3.x API 扩展	444			

开发工具及其概念

第 1 章 Windows 编程准备

确定开发应用程序所需要的硬件和软件可能会很困难。选择哪类设备可以解决特定问题也不总是一件简单的事情。你也许有过这样的挫折或经历——寻找合适的软件版本和确认所购软件的完全兼容性并不容易。

本章介绍建立好的工作环境所需的硬件和软件。由于本章以尽量基本为目的,所以对软件的提及都本着“什么是你真正需要的”为宗旨。

本章讨论 Windows 编程的目录结构、Windows 开发环境的安装、Windows 的零售版本和调试终端的设置,还讨论代码优化选择项和根据计划要编写程序大小的参数。

1.1 硬件要求

在选择一个好的硬件环境之前,应该考虑应用程序的可能大小,要记住的是,大的应用程序需要更多的时间来编译和连接。基于这个原因,如果主要是处理大的应用程序,则应考虑选择功能更强的微处理器以使程序执行更快。估计一下原码模块的大小和数量,如果是将已有的 DOS 应用程序移到 Windows 下,则在已有模块的大小上再增加 30%。

使用不同的硬件配置来运行 Microsoft Windows 并观察不同的选择对建立 Windows 应用程序的影响。

首先,一个好的开发系统应尽量减少资源编译和连接时间。编写 Windows 应用程序时所作的修改只需对一两个原码模块重新编译,因而,连接和资源编译是最耗时的操作。

其次,时钟和磁盘访问速度对编译和连接 Windows 应用程序的时间影响也很大。时钟速度主要影响编译时间和连接过程中的某些步骤,而磁盘访问速度会大大影响编译和连接速度。如果增加一个 2M 或更大的 RAM 驱动器,则会极大地降低编译和连接时间。

编译模块时,读 C 语言的头文件(如 windows.h)花费了相当长的时间,在本章的软件说明部分将介绍如何减少此类操作的访问时间。

表 1.1 总结了开发 Windows 应用程序所需的硬件配置。

由于新的硬件的不断推出,使得购买选择变得复杂起来,当选择新的硬件时应该记住的一点是磁盘访问速度和时钟频率是好的开发环境中的主要方面,选择新的硬件系统首先解决磁盘访问速度和时钟频率问题,然后考虑 RAM 驱动器和高分辨率显示器。好的开

发实践要求既能在速度较低、性能较低的机器上也要能在速度快、大 RAM 的机器上测试,因为你的用户并不一定使用最新、性能最强的硬件,在 486 上开发可以加快速度,但是你的程序应该在 286 或 386 SX 上测试。

表 1.1 Windows 编程的硬件配置

基本配置	4M RAM,25MHz 和 386 微处理器
	80M 硬盘
	VGA 图形卡和彩色监视器
	1 只鼠标
	单色卡和监视器(用于调试)*
	1M RAM 驱动器
高档配置	8M RAM,50MHz 和 486 微处理器
	200M 硬盘
	SVGA 图形卡和彩色监视器
	1 只鼠标
	单色卡和监视器(用于调试)*
	2M RAM 驱动器(无等待状态)

* 单色监视器和适配卡可选,因为单监视器系统可以使用 CodeView 的 Windows 版本,选择的基准是权衡费用和开发时间。

1.2 软件要求

Microsoft 软件开发工具(SDK)的正确安装可以对 Windows 开发环境的性能产生极大的影响,正是由于这种考虑,这一部分介绍正确的目录安排和 Windows 编程所需的文件。下面列出了一个组成实用编程环境所需的最少软件:

- Microsoft 软件开发工具(SDK 3.0 或更高版本)。
- Microsoft C 编译器(版本 5.1 或更高版本)。
- 你喜欢的程序编辑器。
- 一个好的查找工具,它应该能够在多个文件中查找关键字(例如,FGREP),CompuServe 和其它提供多种公共域和共享查找功能的服务平台。

讨论中,涉及的参考材料包括 SDK 版本 3.0 和 Microsoft C 编译器版本 6.0AX。

提示: 本书及磁盘上的例子已经完全在 SDK 版本 3.1 和 Microsoft C/C++7.0 下测试过。

1.3 Windows 概述

Windows 环境的这个概述将标准 DOS 应用程序与 Windows 版本进行比较。但是,要记住的是 Windows 程序具有 DOS 应用程序所没有的性能和特性,这些新特性极大地影响着应用程序的设计,这里只作一个简单的介绍,而下面的章节将进行深一步讨论。