

精通

Windows

程序设计 —— 基于 Visual C++ 实现

朱娜敏 魏宗寿 李红 编著

本书全面讲解了Windows程序设计的方方面面
32个综合实例，覆盖Windows程序设计的常用领域
操作步骤详细，读者按照操作即可得出书中结果



人民邮电出版社
POSTS & TELECOM PRESS



书中全部源程序

精通

Windows

程序设计 — 基于 Visual C++ 实现

朱娜敏 魏宗寿 李红 编著

||

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

精通Windows程序设计: 基于Visual C++实现 / 朱娜敏, 魏宗寿, 李红编著. —北京: 人民邮电出版社, 2009. 4
ISBN 978-7-115-19673-6

I. 精… II. ①朱…②魏…③李… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字 (2009) 第009401号

内 容 提 要

Visual C++是专业程序员使用的工具, 开发人员可以利用 Visual C++ 6.0 基于两种方式编写 Win32 应用程序。一种是 Windows API 的 C 方式, 另一种是 MFC 的 C++ 方式。C 方式是传统的编程方式, 且久经考验, 但是其开发工作量大; 而 C++ 方式源代码效率高, 开发难度和工作量都较小。本书大部分应用程序都是基于 MFC 的 C++ 方式的, 一共分为 5 篇: Windows 编程基础, Windows 绘图程序设计与消息, Windows 界面设计与调试技巧, Windows 数据库、网络通信与 ActiveX 控件, Windows 内部处理技术。循序渐进地讲述了 Windows 编程技术, 从基本原理到具体实践都进行了详细阐述, 并对具体知识点进行了详细的讲解。

本书的特点是全面详细地介绍了 Windows 编程, 突出了编程的原理性与实用性。本书在帮助开发人员理解原理的基础上还应用了大量的实例, 详细介绍了各原理的应用, 以及代码编写技巧, 方便开发人员应用实践。

精通 Windows 程序设计——基于 Visual C++ 实现

- ◆ 编 著 朱娜敏 魏宗寿 李 红
责任编辑 屈艳莲
执行编辑 黄 焱
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷
- ◆ 开本: 787×1092 1/16
印张: 30.25
字数: 736 千字 2009 年 4 月第 1 版
印数: 1—4 000 册 2009 年 4 月北京第 1 次印刷

ISBN 978-7-115-19673-6/TP

定价: 59.00 元 (附光盘)

读者服务热线: (010)67132692 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前 言

Visual C++ 一直都是微软公司综合性最高、最复杂的软件开发平台，其作为专业程序员的使用工具是比较严谨的，并且在目前很多行业应用中起着重要的作用。目前许多开发人员对 Visual C++ 的理论知识以及原理在项目开发中的应用了解得不够深入；或者开发人员清楚如何操作，但却不清楚其中的原因。

C++ 方式源代码效率高，开发难度和开发工作量都较小。本书应用程序都是基于 Visual C++ 的，循序渐进地讲述了 Windows 编程技术，从基本原理到具体实践都进行了阐述，并对具体知识点进行了详细的讲解。本书突出了编程的原理性与实用性。本书在帮助开发人员理解原理的基础上还应用了大量的实例，详细介绍了各原理的应用以及代码编写技巧，方便开发人员应用实践。开发人员学完本书后，可以用掌握的编程知识和技巧开发出目前可用的大多数的 Windows 应用程序。

本书除全面介绍了一般 Windows 应用程序的开发之外，还讲解了数据库开发技术、网络通信程序的设计以及 Windows 系统内部处理技术，其中包含了多线程同步与内存管理、动态链接库 DLL 与钩子、Windows 内核与防火墙开发技术等。

本书的特点

本书从 Windows API、Win32 着手深入分析了 Visual C++ 开发技术，全面说明了 Windows 框架结构与编程特性。本书本着理论与技术相结合的宗旨，使开发人员对所需理论和相关技术有全局的理解和把握，并在实践中加深理解，使读者的应用能力螺旋式上升。

本书的特点主要体现在以下几个方面。

- 注重理论与实践的结合，从理论到实践，在实践中学习理论，从根本上提高编程人员的实际开发能力。
- 提供了大量的 Windows 编程实例，从设计思路、原理到具体实现方式，包含了 Windows 编程的各个领域。对于 Windows 实例，本书提供了完整的源代码，通过书中代码读者基本可以完整编写实例。
- 调试程序是开发过程中经常要做的事情，因此本书加入了调试程序这一章，此章可以在调试程序中随时翻阅，不受基础知识的限制。
- 每一章的最后都有本章知识点的总结，有助于开发人员快速了解本章的内容和有方向地学习。本书包含大量的实用技巧、说明与注意，重点突出，便于灵活掌握，可达到事半功倍的效果。

本书的内容安排

本书分为 5 篇，共 17 章。本书以 Visual C++ 6.0 作为开发工具，深入剖析 Visual C++ 开发的原理与相关技术，全面介绍 Windows 的设计与编程。本书针对项目中经常用到的知识或技术做了针对性的讲解，内容从整体上进行安排和布局，实用性和可读性较强。

第 1 篇（第 1~3 章）Windows 编程基础。

本篇包括：Windows 程序设计的特点及结构、Windows 程序的发展及编程工具、Windows 程序设计注意事项及常用事项、Win32 编程思想及流程、Windows 基本应用程序的创建等。系统地介绍了 Windows 程序设计的基础知识，并对重点知识点着重讲述，使读者能把握程序设计的根本。

第 2 篇（第 4~6 章）Windows 绘图程序设计与消息。

本篇包括：Windows 应用程序中鼠标和键盘等输入设备、Windows 窗口中的绘图、Windows 应用程序窗口消息等。本篇中使用大量的实例代码、图示解释尽可能让读者透彻理解程序运行的过程及原理。

第 3 篇（第 7~10 章）Windows 界面、文档/视图设计与调试。

本篇包括：用户界面窗口中基本控件、菜单、工具栏、状态栏和对话框等的设计，Windows 文件系统与序列化，文档与视图，程序调试与编程技巧等。本篇中仍然使用较多的实例代码和图表解释，做到让读者见图知义，能够更透彻地理解 Visual C++ 程序运行的情况。

第 4 篇（第 11~13 章）Windows 数据库、网络通信与 ActiveX 控件。

本篇包括：Windows 数据库的程序设计、网络通信程序设计、COM、OLE 与 ActiveX 技术等。本篇讲解了 Windows 程序的拓展应用，为读者以后进行大型的数据库或网络通信开发做了知识储备。

第 5 篇（第 14~17 章）Windows 内部处理技术。

本篇包括：多线程应用程序的设计、Windows 系统与内存的管理、动态链接库（DLL）和钩子、Windows 内核程序设计与防火墙开发等。本篇内容属于高级应用，读者通过本篇的学习可以了解 Windows 操作系统底层知识和内部处理技术，从而编写出好的 Visual C++ 程序。

适合阅读本书的读者

本书适合以下人员学习使用。

- (1) C/C++ 爱好者。
- (2) Visual C++ 窗体设计人员。
- (3) Visual C++ 界面设计爱好者。
- (4) Visual C++ 软件架构人员。
- (5) 其他的 C++ 爱好者。

由于笔者水平有限、时间仓促，书中错误和疏漏之处在所难免，恳请广大读者批评指正，读者可以通过发送 E-mail 至 huangyan@ptpress.com.cn 与本书责任编辑联系。

编者
2009 年 1 月

目 录

第 1 篇 Windows 编程基础

第 1 章 Windows 程序设计基础知识.....2

1.1 Windows 程序设计知识简述.....2

1.1.1 Windows 编程特点.....2

1.1.2 Windows 应用程序的编程
模型.....3

1.1.3 Windows 程序基本结构.....4

1.2 Windows 程序的发展.....4

1.2.1 传统的 Windows 编程.....4

1.2.2 Windows 与 API 函数.....5

1.2.3 Windows 与 Win32 API.....5

1.2.4 Windows 与面向对象.....6

1.2.5 Windows 对象与 MFC 对象.....6

1.3 Windows 编程应用工具.....7

1.3.1 Windows 编程工具简介.....7

1.3.2 为何使用 Visual C++.....8

1.3.3 Visual C++ 6.0 的安装.....8

1.3.4 Visual C++ 6.0 集成开发环境.....9

1.3.5 Visual C++ 6.0 的基本使用.....9

1.4 【实例】一个显示窗口的典型的
Windows 应用程序.....10

1.4.1 创建应用程序.....10

1.4.2 Windows 程序生命期.....12

1.4.3 Windows 程序要素简介.....12

1.4.4 Winmain()入口函数.....13

1.4.5 WndProc()窗口处理函数.....13

1.4.6 进队消息与不进队消息.....14

1.4.7 设备描述表.....14

1.4.8 应用程序的退出.....15

1.5 【实例】建立第一个控制台应用
程序.....15

1.5.1 创建工程.....15

1.5.2 控制台应用程序说明.....16

1.5.3 编译并链接源程序.....16

1.5.4 应用 Visual C++调试器调试
程序.....17

1.6 Windows 程序设计注意事项.....17

1.6.1 Windows 应用程序中的调用
关系.....181.6.2 Windows 应用程序多任务
处理.....18

1.7 Windows 程序常用事项.....18

1.7.1 匈牙利式命名规则.....18

1.7.2 代码的对齐.....19

1.7.3 应用注释.....20

1.7.4 MSDN 的应用.....21

1.8 小结.....21

第 2 章 Win32 编程基础.....22

2.1 Win32 程序开发流程.....22

2.2 【实例】用 Win32 API 编写一个
简单的 Windows 程序.....23

2.2.1 创建应用程序	23	3.1.3 高层编程法	43
2.2.2 调用头文件说明	23	3.2 【实例】建立一个简单的 MFC 应用程序	43
2.2.3 WinMain() 入口函数	23	3.2.1 创建应用程序	43
2.2.4 API 函数的调用	24	3.2.2 MFC 应用程序对象	44
2.3 Win32 编程思想	25	3.2.3 建立框架窗口对象	45
2.3.1 事件驱动的程序设计思想	25	3.2.4 绘制窗口	47
2.3.2 消息和消息处理机制思想	25	3.2.5 消息映射处理	48
2.3.3 动态链接技术	26	3.2.6 有关字符串的定义	48
2.3.4 句柄	27	3.3 了解 Windows 应用程序的类型	49
2.3.5 其他 Win32 编程思想	28	3.4 关于 AppWizard 与 ClassWizard	49
2.4 【实例】一个典型的 Win32 程序	28	3.4.1 AppWizard 应用程序向导	49
2.4.1 程序入口 WinMain()	28	3.4.2 ClassWizard 类向导	50
2.4.2 窗口类的注册与产生	29	3.5 【实例】创建第一个 MFC AppWizard 工程——MDI 应用程序	51
2.4.3 消息循环	30	3.5.1 生成 MDI 应用程序框架	51
2.4.4 窗口函数的处理	30	3.5.2 浏览应用程序文件及工作区 中的项目	51
2.4.5 消息映射	31	3.5.3 调试应用程序	52
2.4.6 对话框的运行	32	3.6 关于 MDI 的说明	53
2.4.7 模块定义文件	33	3.7 关于 MDI 应用程序框架的组成 说明	53
2.4.8 资源描述文件	33	3.7.1 应用程序对象	54
2.5 进程与线程	34	3.7.2 窗口对象	55
2.5.1 内核对象	34	3.7.3 视图对象	55
2.5.2 基于 Win32 的进程模式	35	3.7.4 文档对象	56
2.5.3 进程的属性与资源	36	3.7.5 MFC 的主要类及其层次关系	57
2.5.4 进程与线程的相关性	37	3.8 MDI 应用程序的执行流程	57
2.5.5 创建进程 (CreateProcess) 函数	37	3.9 MDI 工程的相关文件说明	58
2.5.6 【实例】进程的创建	38	3.9.1 工程文件	58
2.5.7 控制与终止某进程	39	3.9.2 资源文件	59
2.5.8 与进程相关的 API 函数	40	3.9.3 预编译文件	59
2.6 Win32 程序的空闲处理	40	3.9.4 注册表项目文件	59
2.7 小结	41	3.9.5 帮助文件	60
第 3 章 创建 Windows 应用程序	42	3.10 小结	60
3.1 创建 Windows 应用程序的方法	42		
3.1.1 底层编程法	42		
3.1.2 MFC 编程法	42		

第 2 篇 Windows 绘图程序设计 with 消息

第 4 章 Windows 应用程序输入设备	62	4.1.1 鼠标设备基础知识	62
4.1 鼠标	62	4.1.2 鼠标光标的热点	63
		4.1.3 客户区与非客户区鼠标消息	63

4.1.4 鼠标跟随原理与 WM_MOUSEMOVE	64	5.4.2 坐标转换的函数应用	95
4.1.5 【实例】鼠标消息的捕获与 应用	65	5.4.3 【实例】坐标转换的应用	95
4.1.6 鼠标光标形状的设置	66	5.5 窗口与视口	96
4.1.7 【实例】不同区域显示不同 光标	67	5.5.1 概念介绍	96
4.2 键盘	69	5.5.2 窗口与视口的原点	97
4.2.1 键盘设备基础知识	70	5.5.3 窗口与视口的相关函数及其 应用	97
4.2.2 接收键盘输入焦点消息及 处理	70	5.6 窗口重绘函数	99
4.2.3 获取击键消息	71	5.6.1 OnDraw 与 OnPaint 函数	99
4.3 【实例】从键盘读取输入	72	5.6.2 添加重绘资源	101
4.3.1 为键盘数据建立存储区域	72	5.7 图形文件的操作	102
4.3.2 读取击键	72	5.7.1 创建新文档	102
4.3.3 键入的字符保存并显示	73	5.7.2 保存图形文件	102
4.3.4 运行程序并键入字符测试	73	5.7.3 打开图形文件	103
4.3.5 程序拓展——文档的永久 保存	74	5.7.4 其他相关函数	103
4.4 计时器	74	5.8 【实例】鼠标绘制直线与圆形	104
4.4.1 设置计时器	74	5.8.1 绘图原理与 WM_LBUTTONDOWN、 WM_LBUTTONUP	104
4.4.2 WM_TIMER 消息	75	5.8.2 集合数组类的应用—— 图形的保存	107
4.4.3 清除计时器 WM_KILL	75	5.8.3 图形的重绘	108
4.4.4 【实例】设置 2 个计时器 程序	75	5.8.4 坐标点的平移——图形拉伸 效果的实现	109
4.5 小结	79	5.8.5 实例拓展	110
第 5 章 Windows 窗口中绘图	80	5.9 小结	110
5.1 图形设备接口 GDI	80	第 6 章 Windows 应用程序窗口消息	111
5.1.1 设备描述表类介绍	80	6.1 窗口管理	111
5.1.2 设备描述表类的应用方法	82	6.1.1 C++ 窗口对象和 Windows 窗口对象	111
5.1.3 GDI 绘图对象	83	6.1.2 创建窗口	111
5.1.4 【实例】用 GDI 绘图	86	6.1.3 定位窗体	112
5.2 绘图模式	87	6.1.4 销毁窗体	113
5.2.1 绘图模式概述	88	6.2 主要编程对象——窗口	113
5.2.2 【实例】绘图模式的应用—— 橡皮筋矩形的实现	88	6.2.1 窗口结构与类型	114
5.3 Windows 中的映射模式	90	6.2.2 窗口关系	115
5.3.1 映射模式概述	91	6.2.3 窗口应用框架	117
5.3.2 【实例】映射模式的应用	92	6.3 窗口消息映射	117
5.4 坐标转换	93	6.3.1 Windows 消息的分类	117
5.4.1 逻辑坐标与设备坐标	94	6.3.2 消息的结构	120
		6.3.3 消息处理	120

6.3.4 窗口消息映射表	121	6.4.9 线程消息处理	130
6.3.5 窗口消息的定位	122	6.4.10 跨进程消息处理	131
6.3.6 消息映射宏的种类	122	6.4.11 【实例】跨进程消息交换 数据	132
6.3.7 消息的预处理	123	6.5 【实例】MFC 的窗口过程	133
6.4 窗口消息处理机制	124	6.5.1 MFC 窗口过程的指定	133
6.4.1 标准窗口消息处理	124	6.5.2 对 Windows 消息的接收和 处理	135
6.4.2 命令消息处理	125	6.5.3 对命令消息的接收和处理	136
6.4.3 消息反射处理	125	6.5.4 对控制通知消息的接收和 处理	140
6.4.4 更新命令处理	126	6.5.5 对更新命令的接收和处理	141
6.4.5 自定义窗口消息	126	6.6 小结	144
6.4.6 自定义命令	128		
6.4.7 Post 与 Send 消息	128		
6.4.8 两进程窗口通信的注册消息	129		
第 3 篇 Windows 界面设计与调试技巧			
第 7 章 用户窗口界面的设计	146	7.3.5 组合框加入工具栏	175
7.1 Windows 控件的设计与应用	146	7.4 状态栏	176
7.1.1 Windows 控件简介	146	7.4.1 创建普通状态栏	176
7.1.2 静态控件	147	7.4.2 高级状态栏	177
7.1.3 按钮	148	7.4.3 在对话框中设置状态栏	180
7.1.4 编辑框	151	7.5 对话框的设计	181
7.1.5 列表框	153	7.5.1 对话框原理介绍	182
7.1.6 组合框	156	7.5.2 模式对话框与无模式 对话框的设计	182
7.1.7 滚动条	157	7.5.3 系统通用对话框的应用	183
7.1.8 列表视图控件	158	7.5.4 自定义对话框形状	188
7.1.9 树形控件	158	7.5.5 对话框与控件的数据交互	188
7.1.10 旋转、进度条和滑块控件	160	7.5.6 对话框与文档视图的数据 交互	190
7.1.11 位图按钮、图像列表控件	162	7.6 【实例】将文本框与组合框加入 列表视图控件实例	190
7.1.12 自定义控件的颜色	163	7.6.1 准备知识	190
7.2 菜单	165	7.6.2 创建结构体	196
7.2.1 菜单基础知识	166	7.6.3 创建所需相关类	197
7.2.2 创建菜单应用程序	166	7.6.4 功能实现	198
7.2.3 创建上下文快捷/弹出式 菜单	168	7.7 小结	199
7.2.4 菜单应用程序加入对话框 窗口	169	第 8 章 Windows 文件系统与序列化	200
7.3 工具栏	170	8.1 文件 CFile 类	200
7.3.1 创建工具栏	170	8.1.1 创建、打开与关闭文件	201
7.3.2 工具栏与应用程序相关联	172	8.1.2 文件的读、写	202
7.3.3 工具栏的可见性	172	8.1.3 移动文件指针	203
7.3.4 停靠栏	174		

8.1.4 临时文件的操作	204	9.4.2 初始化视图	236
8.1.5 监控目录, 搜索文件	205	9.4.3 视图与文档的数据交互	236
8.1.6 文件属性的获取和更改	206	9.4.4 多个视图添加到单个文档	237
8.1.7 文件的分割与合并	208	9.5 文档/视图与框架的交互	238
8.1.8 锁定文件	209	9.5.1 框架窗口的处理	239
8.1.9 CFile 的派生类	210	9.5.2 框架命令的处理	239
8.2 文件的序列化	210	9.5.3 框架与文档/视图的交互	240
8.2.1 序列化基础	211	9.6 打印的设置	241
8.2.2 序列化函数 Serialize()	214	9.6.1 文档与视结构中的打印 流程	241
8.2.3 自定义可序列化类	217	9.6.2 自定义打印功能的实现	243
8.3 【实例】普通文件的编辑器	218	9.6.3 打印预览	244
8.3.1 项目框架程序	218	9.6.4 【实例】图像打印程序	245
8.3.2 初始化文档数据	219	9.7 【实例】多文档和多视图的 编程	249
8.3.3 文件串行化处理	220	9.7.1 创建多文档与多视图	249
8.3.4 文本数据的显示	220	9.7.2 定义新视图类	250
8.3.5 文本字体的设定	222	9.7.3 修改子框架窗口 CChildFrame	252
8.4 小结	223	9.8 小结	253
第 9 章 文档与视图	224	第 10 章 程序调试与编程技巧	254
9.1 文档与视图的概述	224	10.1 调试 Visual C++ 程序	254
9.1.1 文档与视图的关系	224	10.1.1 bugs 程序	254
9.1.2 文档与视图的接口	225	10.1.2 设置断点	255
9.1.3 SDI 与 MDI 应用程序的 区别	225	10.1.3 单步执行	256
9.2 文档/视图框架结构的关系	226	10.1.4 检查运行时程序的数据	257
9.2.1 MFC 各对象间的关系	226	10.1.5 编辑并继续	260
9.2.2 文档类	227	10.2 其他相关处理	260
9.2.3 视图类	227	10.2.1 异常处理	261
9.2.4 框架窗口类	228	10.2.2 诊断调试	262
9.2.5 文档模板类	228	10.2.3 用 Visual C 做远程调试	264
9.2.6 多文档	229	10.2.4 使用调试功能检测内存 泄漏	264
9.2.7 多视图	230	10.3 DEBUG 版本与 RELEASE 版本	269
9.2.8 拆分窗口	230	10.4 其他应用技巧	270
9.2.9 文档/视图的滚动处理	231	10.4.1 全面清除历史记录实例	270
9.3 文档类	233	10.4.2 清理 Visual C++ 工程	275
9.3.1 文档的生命周期	233	10.4.3 通过优化使代码效率更高	275
9.3.2 文档的操作与函数的重载	234	10.4.4 Visual C++ 编译时加快 链接	276
9.3.3 设置文档修改标志	235		
9.3.4 文档的序列化保存	235		
9.3.5 关闭并销毁文档	235		
9.4 视图类	235		
9.4.1 视图类的函数	236		

10.4.5 代码维护技巧	277
---------------------	-----

10.5 小结	277
---------------	-----

第 4 篇 Windows 数据库、网络通信与 ActiveX 控件

第 11 章 数据库的程序设计	280
-----------------------	-----

11.1 数据库概述	280
------------------	-----

11.1.1 应用 Visual C++ 开发 数据库的特点	280
---	-----

11.1.2 Visual C++ 开发数据库的 技术	281
--------------------------------------	-----

11.2 ODBC 数据库管理	281
-----------------------	-----

11.2.1 ODBC 的体系结构	282
-------------------------	-----

11.2.2 SQL 操作	283
---------------------	-----

11.2.3 用 MFC ODBC 访问数 据库	285
-----------------------------------	-----

11.3 DAO 数据库访问对象	287
------------------------	-----

11.3.1 基于 DAO 的程序特色	287
---------------------------	-----

11.3.2 用 DAO 开发数据库应用 程序	288
----------------------------------	-----

11.4 【实例】ODBC 技术开发 数据库	290
---------------------------------	-----

11.4.1 创建数据源	290
--------------------	-----

11.4.2 创建 ODBC 应用程序	290
---------------------------	-----

11.4.3 增加数据表记录	292
----------------------	-----

11.4.4 删除数据表记录	294
----------------------	-----

11.4.5 查找数据表记录	294
----------------------	-----

11.5 小结	295
---------------	-----

第 12 章 网络通信程序设计	296
-----------------------	-----

12.1 网络体系结构与协议	296
----------------------	-----

12.1.1 OSI 参考模型	296
-----------------------	-----

12.1.2 TCP/IP 分层模型	298
--------------------------	-----

12.1.3 IP 地址	300
--------------------	-----

12.1.4 TCP/IP	302
---------------------	-----

12.1.5 UDP 协议	302
---------------------	-----

12.1.6 客户端/服务器模型	303
------------------------	-----

12.2 Windows Sockets 编程	304
-------------------------------	-----

12.2.1 Socket (套接字) 介绍	304
------------------------------	-----

12.2.2 Socket 程序设计	305
--------------------------	-----

12.2.3 【实例】WinSock 程序	308
-----------------------------	-----

12.3 MFC 套接字编程	311
----------------------	-----

12.3.1 套接字类 CAsyncSocket 的	
----------------------------	--

介绍	312
----------	-----

12.3.2 套接字类 CSocket 的应用	313
-------------------------------	-----

12.3.3 CSocket 类的应用特点	314
-----------------------------	-----

12.4 WinInet 编程	316
-----------------------	-----

12.4.1 MFC 的类 WinInet	316
-----------------------------	-----

12.4.2 WinInet 编程方法介绍	317
-----------------------------	-----

12.5 FTP 编程	318
-------------------	-----

12.5.1 FTP 简介	319
---------------------	-----

12.5.2 FTP 客户端编程	320
------------------------	-----

12.5.3 FTP 的断点续传	326
------------------------	-----

12.6 【实例】创建 Web 浏览器	326
---------------------------	-----

12.6.1 准备知识	326
-------------------	-----

12.6.2 建立 Web 浏览器框架	327
---------------------------	-----

12.6.3 应用程序实现代码	328
-----------------------	-----

12.7 小结	331
---------------	-----

第 13 章 COM、OLE 与 ActiveX 技术	332
-----------------------------------	-----

13.1 组件对象模型 COM	332
-----------------------	-----

13.1.1 创建 COM 对象	333
------------------------	-----

13.1.2 对象的生存期	333
---------------------	-----

13.1.3 获取接口指针	334
---------------------	-----

13.1.4 创建 COM 服务器	334
-------------------------	-----

13.1.5 定位透明度	336
--------------------	-----

13.2 OLE 技术概述	337
---------------------	-----

13.2.1 结构化存储	337
--------------------	-----

13.2.2 Monikers 介绍	338
--------------------------	-----

13.2.3 统一数据传输	338
---------------------	-----

13.2.4 OLE 文档	339
---------------------	-----

13.2.5 OLE 自动化	339
----------------------	-----

13.3 ActiveX 技术概述	340
-------------------------	-----

13.3.1 ActiveX 控件	341
-------------------------	-----

13.3.2 ActiveX 文档	342
-------------------------	-----

13.3.3 ActiveX 控件容器	343
---------------------------	-----

13.3.4 创建 Activex 控件	344
----------------------------	-----

13.3.5 注册 ActiveX 控件	347
----------------------------	-----

13.3.6 插入 Activex 控件	348
----------------------------	-----

13.4 【实例】ActiveX 时钟控件	350
-----------------------------	-----

13.4.1 创建时钟控件的基本功能	350
--------------------------	-----

13.4.2 控件属性	351
-------------------	-----

13.4.3 使用属性页354

13.4.4 设置控件的事件355

13.4.5 添加控件方法 356

13.5 小结 356

第 5 篇 Windows 内部处理技术

第 14 章 多线程应用程序的设计358

14.1 线程基本知识概述358

14.1.1 线程的 ID 和句柄358

14.1.2 设置线程的优先级359

14.1.3 多任务与多线程360

14.2 创建和使用线程361

14.2.1 创建线程 CreateThread()
函数361

14.2.2 线程的分类363

14.2.3 启动及初始化线程363

14.2.4 悬挂和恢复线程366

14.2.5 线程的休眠366

14.2.6 终止线程366

14.2.7 【实例】创建工作线程367

14.3 实现多线程同步的相关知识368

14.3.1 线程的同步机制369

14.3.2 临界区对象369

14.3.3 互斥内核对象372

14.3.4 事件内核对象374

14.3.5 信号量内核对象378

14.3.6 线程局部存储381

14.3.7 互锁函数382

14.4 【实例】多线程的应用382

14.4.1 添加线程对象和线程函数382

14.4.2 添加鼠标左键触发函数383

14.4.3 添加状态栏坐标提示383

14.5 小结384

第 15 章 Windows 系统与内存的管理385

15.1 驱动器和文件目录385

15.1.1 驱动器卷标操作385

15.1.2 逻辑驱动器的操作386

15.1.3 格式化驱动器387

15.1.4 文件目录操作388

15.2 注册表的应用389

15.2.1 注册表的结构389

15.2.2 注册表的管理390

15.2.3 【实例】用注册表键存储

用户信息 393

15.3 任务栏状态区 395

15.3.1 任务栏状态区操作函数 395

15.3.2 任务栏状态区托盘程序的
实现 396

15.4 控制面板 398

15.4.1 控制面板的工作原理 399

15.4.2 控制面板应用程序的实现 400

15.5 内存管理 402

15.5.1 Windows 内存分布 402

15.5.2 内存属性与状态 403

15.5.3 全局内存与虚拟内存 405

15.5.4 虚拟内存管理 407

15.5.5 自由的、保留的和已提交的
虚拟内存 407

15.5.6 释放虚拟内存 409

15.5.7 改变虚拟内存页的保护 409

15.5.8 锁定虚拟内存页 409

15.5.9 查询进程的虚拟内存 410

15.6 内存映射文件 410

15.6.1 内存映射文件的用途 410

15.6.2 内存映射文件的编程流程 412

15.6.3 内存映射文件的视图 412

15.7 【实例】用内存映射文件处理

大文件 413

15.7.1 创建文件内核对象 414

15.7.2 创建文件映射内核对象 414

15.7.3 文件数据及视图的映射 416

15.7.4 关闭文件对象和文件映射
对象 416

15.8 小结 417

第 16 章 动态链接库 DLL 和钩子 418

16.1 动态链接库 DLL 418

16.1.1 动态链接的特点 418

16.1.2 Visual C++ 支持的动态
链接库类型 419

16.1.3 DLL 入口函数.....	420	17.1.1 Windows 层次结构	439
16.1.4 DLL 函数的导出.....	421	17.1.2 CPU 保护模式	440
16.1.5 连接 DLL 的方式—— 显式和隐式	423	17.1.3 内核模式和用户模式	440
16.2 【实例】应用共享 MFC DLL 创建一个 DLL	424	17.2 Windows 系统服务.....	441
16.3 【实例】测试 DLL 文件	424	17.2.1 常用 API 服务函数	441
16.3.1 创建应用程序框架	424	17.2.2 设计服务程序.....	443
16.3.2 编辑对话框并添加相应 执行函数	424	17.3 内核驱动程序设计.....	447
16.3.3 连接待测试的 DLL 库文件.....	425	17.3.1 内核及内核对象.....	447
16.4 Windows 钩子	426	17.3.2 内核模式编程环境.....	447
16.4.1 钩子的定义及类型	426	17.3.3 内核钩子	449
16.4.2 钩子链表和钩子子程	429	17.3.4 【实例】最小化驱动程序	451
16.4.3 钩子的安装与卸载	429	17.4 Windows 防火墙.....	454
16.4.4 全局钩子与线程钩子	430	17.4.1 防火墙核心功能概述.....	454
16.4.5 传递事件的函数	430	17.4.2 代理防火墙与服务服务器.....	455
16.5 【实例】应用钩子机制截获 鼠标点击次数.....	431	17.4.3 防火墙的工作流程.....	456
16.5.1 创建钩子 DLL 文件	431	17.4.4 防火墙模块的划分.....	457
16.5.2 创建调用钩子 DLL 文件的 程序.....	432	17.5 网络封包截获技术.....	457
16.5.3 运行并测试	433	17.5.1 过滤驱动程序的特性.....	457
16.6 Win32 API 的挂钩技术	433	17.5.2 传输层截获网络封包技术	457
16.6.1 网络监控技术	434	17.5.3 网络驱动程序接口规范.....	457
16.6.2 屏幕取词技术	434	17.5.4 中间驱动程序的特性.....	458
16.6.3 防止进程结束技术	435	17.5.5 网络中间驱动程序截获 网络封包技术	458
16.6.4 DLL 注入机制.....	435	17.6 【实例】Windows 个人防火墙 系统的实现	459
16.6.5 系统服务的挂钩	437	17.6.1 Filter-Hook 驱动.....	459
16.7 小结	438	17.6.2 创建内核模式驱动.....	459
第 17 章 Windows 内核程序设计与 防火墙开发	439	17.6.3 过滤函数.....	462
17.1 Windows 操作系统和 CPU 的 保护模式.....	439	17.6.4 设置过滤函数.....	465
		17.6.5 防火墙应用户操作界面的 实现.....	466
		17.6.6 防火墙应用程序的实现.....	467
		17.7 小结	470

第 1 篇 Windows 编程基础

第 1 章 Windows 程序设计基础知识

第 2 章 Win32 编程基础

第 3 章 创建 Windows 应用程序

第 1 章

Windows 程序设计基础知识

Windows 为用户提供了一种多任务、面向图形的多窗口环境。在此环境下可运行专门为 Windows 设计的程序。该类程序具有一致的外观和命令结构，因此，它比传统的 MS-DOS 程序更加易学易用。

用户可以在不同的 Windows 程序间切换，从而使程序间数据的交换更加方便。许多为 MS-DOS 编写的程序也能在 Windows 下运行。虽然这些程序无法利用 Windows 的许多优点，但它们在很多情况下也可以窗口化，并与 Windows 程序并发执行。

Windows 为程序开发人员提供了丰富的内部例程，使程序员可以使用菜单、对话框、滚动条等友好的用户界面。Windows 还给出了一种外在的图形程序设计语言，这种语言可以对各种不同的字模进行格式化。程序员可以用设备无关方式来处理键盘、鼠标、视频显示器及打印机等。

本章主要讲解用户编写 Windows 应用程序必需的基础知识，包括用户应该备有什么软件、如何建立一个 Windows 应用程序及 Windows 程序设计中的注意事项等。

1.1 Windows 程序设计知识简述

本节简要介绍 Windows 编程特点及基本概念，理解这些概念是进一步学习的基础。只有理解了这些概念，才能更好地学习其他章节。

1.1.1 Windows 编程特点

Windows 程序设计完全突破了传统 MS-DOS 系统的局限性，系统信息都在统一的图形窗口下交换。Windows 编程特点如下所述。

● 用户界面一致。

Windows 操作系统提供了一个漂亮、友好、统一的图形界面。不同的 Windows 应用程序有着大致相同的用户界面，大大提高了学习、使用计算机的效率。大部分操作都可以用键盘和鼠标来完成或使用下拉菜单来处理，更方便的是可以直接用鼠标进行拖放操作。Windows 应用程序还可以通过对话框来获取外部输入信息。

● 多窗口、多任务管理。

Windows 环境提供了多个窗口，每个应用程序占用一个窗口，用户可以随意操作每个窗口，并可以在不同的窗口间切换并传递消息。

Windows 支持多任务操作，一个操作环境中允许多个应用程序同时运行。Windows 将 CPU 分配给要同时运行的各个应用程序，从而可在同一时刻运行多个程序，并可在程序间进行信息交互。

● 内存管理高效、完善。

在 Windows 环境中，内存是一种共享资源，由 Windows 统一管理和调度。在同一时间内会有多个程序运行，所以任何一个应用程序都不能独占内存。一个程序开始运行时，必须占去一段内存空间。一个程序结束运行后，又会释放回一段内存空间。

● 设备无关性。

Windows 的图形设备接口 (Graphics Device Interface, GDI) 实现了程序与设备的无关性，即为 Windows 编写的应用程序可以运行于任何具有 Windows 设备驱动程序的硬件环境中。不论硬件设备如何更新，在 Windows 环境中只要有适当的设备驱动程序，应用程序便可以通过它来控制新的硬件，而无需修改程序的源代码。

1.1.2 Windows 应用程序的编程模型

传统的应用程序是过程化的模型。程序从头到尾按顺序执行，程序执行路径是可预测的。而 Windows 程序不是这样执行的，它是通过处理操作系统发来的消息来响应事件的，其编程模型如图 1.1 所示。

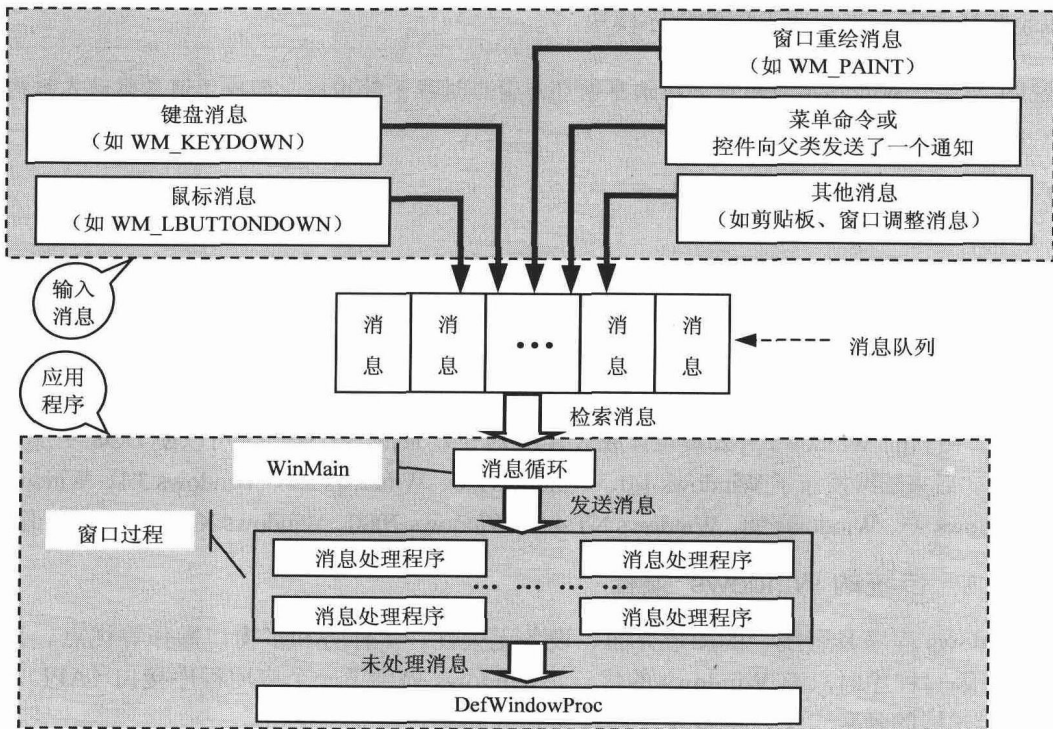


图 1.1 Windows 编程模型

Windows 应用程序的入口点为 `WinMain()` 函数，但是大多数操作都是在窗口过程的函数中进行的。窗口过程函数处理发送给窗口的消息。

 **注意** 应用程序不能处理的消息被传递给 `DefWindowProc` 的 `API (Application Programming Interface)` 函数，该函数对传来的消息提供默认响应。

1.1.3 Windows 程序基本结构

Windows 应用程序的结构基本固定，由 `WinMain()` 入口函数和自定义的窗口过程函数两大部分构成基本框架。

1. `WinMain()` 函数

`WinMain()` 函数的作用类似于 C 语言或 C++ 控制台方式运行程序的 `main()` 函数，它是所有 Windows 应用程序的入口。`WinMain()` 函数完成定义窗口类、建立用户窗口等初始化工作，并产生消息循环。

消息循环是 `WinMain()` 函数中最重要的部分，是整个应用程序运行的核心。其作用是获取与本用户程序有关的信息。程序执行至此就一直处于该消息循环队列中，等待接收消息。每获取一个消息便将其发送到与之相对应的窗口过程函数，由窗口过程函数处理该消息事件。

2. 自定义的窗口过程函数

应用程序中定义的每个窗口都必须有一个与之对应的窗口函数，用来处理来自该窗口的消息。Windows 应用程序中对不同消息的响应均在窗口函数中定义，由消息处理分支结构（如 `switch` 选择语句）来响应不同的程序模块。

 **说明** 整个 Windows 应用程序由消息循环与窗口过程函数构成，形成了消息驱动式结构。

1.2 Windows 程序的发展

20 世纪 80 年代初，美国 Xerox 公司推出的世界上第一个商用的图形用户界面（GUI）是 Windows 系统的雏形。1983 年 Apple 公司推出了世界上第一个成功的 GUI 系统，并制定了图形环境标准。

基于 Intel x86 微处理器芯片的 IBM 兼容微机出现后，微软公司有了很大的发展空间来开发 Windows。微软的 Windows 开始被设计成 MS-DOS 的扩展版本，是一个可以使 DOS 更加友好的图形界面。后来陆续发布了 Windows 1.0、Windows 2.0、Windows 3.0、Windows 3.1、Windows NT 3.0、Windows 95、Windows 98、Windows NT 4.0、Windows 2000、Windows XP 等版本的操作系统。

1.2.1 传统的 Windows 编程

Windows 程序具有统一的图形界面、设备无关性、多任务和多窗口操作等优点。为了使程序员在编写程序时具有 Windows 的优点，Windows 提供了一个应用程序接口（API）。它是 Windows 支持的函数定义、参数定义和消息格式的集合。

直接采用应用程序接口（API）进行的 C/C++ 程序设计称为传统的 Windows 编程。