Windows 程序设计

王艳平　编著

人民邮电出版社

图书在版编目（CIP）数据

Windows 程序设计/王艳平编著. —北京：人民邮电出版社，2005.3
ISBN 7-115-13205-4

Ⅰ. W… Ⅱ. 王… Ⅲ. 窗口软件，Windows—程序设计 Ⅳ. TP316.7

中国版本图书馆 CIP 数据核字（2005）第 019046 号

内 容 简 介

本书编写的目的是为学习 Windows 编程的读者提供一个良好的学习方法，循序渐进，最终使他们从根本上提高编程水平，有能力独立开发出像 Windows 防火墙一样复杂的应用程序。本书首先介绍了 Win32 程序运行原理和最基本的 Win32 API 编程，然后通过模拟 MFC 中关键类、全局函数和宏定义的实现详细讲述了框架程序的设计方法和 MFC 的内部工作机制，并指出了这些机制是如何对用户程序造成影响的，最后完整讲述了开发内核驱动和 Windows 防火墙的过程。

全书语言严谨流畅，针对初学者的特点，精心策划、由浅到深，是学习 Windows 编程的理想书籍。

Windows 程序设计

◆ 编　　著　王艳平
责任编辑　刘　浩

◆ 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
读者热线　010-67132692
北京密云春雷印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：28.5
字数：696 千字　　2005 年 3 月第 1 版
印数：1 – 5 000 册　　2005 年 3 月北京第 1 次印刷

ISBN 7-115-13205-4/TP・4535

定价：49.00 元（附光盘）

前　言

许多人在刚开始接触 Windows 编程时，从 Visual Basic 开始，从 MFC 开始，虽然写出了程序，但他们自己都不知道程序是如何运行的，从而造成写程序“容易”修改难，设计程序“容易”维护难的状况。本书是为 Windows 程序设计的初学者和想从根本上提高自己编程水平的爱好者们编写的，试图使他们成为编程高手。

API 函数是 Windows 提供给应用程序的编程接口，任何用户应用程序必须运行在 API 函数之上。直接使用 API 编程是了解操作系统运行细节的最佳方式，而且熟知 API 函数也是对程序开发者的一个最基本的要求。本书将以 API 函数作为起点介绍 Windows 编程，这样做的好处是使读者撇开 C++的特性专心熟悉 Win32 编程思路和消息驱动机制。

但是，在开发大型系统的时候，往往并不完全直接使用 API 函数，而是使用 MFC 类库框架程序。MFC 对 90%以上的 API 函数进行了面向对象化包装，完全体现了对象化程序设计的特点，是目前流行的类库。

当读者熟悉最基本的 API 函数编程以后，就可以学习更高级的 MFC 编程了。但是，虽然 MFC 仅仅是对 API 函数的简单封装，由于读者对 C++语言的了解不够，不清楚框架程序的工作机制，即便是有经验的程序员在 MFC 复杂的结构面前也显得非常困惑。他们会“用”MFC，却不知道为什么这么“用”，在运行程序出错时这种现象带来的问题就很明显，他们不会改。

面对一个大的项目，代码往往需要手工添加和修改，而很少能够依靠 VC++的向导。为此，本书将从开发者的角度同读者一起来设计 MFC 中的类、函数和宏定义。通过对 MFC 类库的分析和了解，不仅能够使读者更好地使用 MFC 类库，同时，对于自己设计和实现框架和类，无疑也有相当大的帮助。

之后，本书讲述了 Windows 系统编程中当前最为热门的话题——DLL 注入技术、远程进程技术、HOOK API 技术等，并配有完整而具体的实例。

本书的最后将讨论 Windows 内核驱动程序设计和防火墙开发。这对于全面了解 Windows 操作系统的结构体系，学习独立开发应用软件是非常有帮助的。

—— 内容安排

《Windows 程序设计》从“Hello，World”这个简单的例子出发，通过多个实例，由浅入深地讲述 Win32 API 程序设计、类库框架设计、MFC 程序设计、内核模式程序设计等，使读者在实践中熟练掌握 Windows 程序设计模式，并有能力写出完成特定功能的用户应用程序和简单的内核驱动程序。

第 1 章（Windows 程序设计基础）写给从未接触过 Windows 程序的读者，使读者的知识符合本书的要求。

第 2~4 章（Win32 程序运行原理、Win32 程序的执行单元、Windows 图形界面）详细介绍 SDK 编程，这是 Windows 下最基本的编程方式，它是依靠直接调用 API 函数来编写 Windows 应用程序的。

第 5~7 章（框架管理基础、框架中的窗口、用户界面设计）中，笔者将和读者一起设计自己的“类库”，这个小“类库”是 MFC 的一个缩影，它将 MFC 中核心的东西，如运行期信

息、线程/模块状态、消息映射、内存管理等都非常清晰地体现了出来。在封装 API 的时候，作者详细介绍了 C++语言中虚函数、静态函数、继承和类模板等高级特性的具体应用，叙述了框架程序管理应用程序的每一个细节，用各种精彩的实例介绍如何进行对象化程序设计，如何使用 MFC 类库简化开发周期。整个过程就是深入了解对象化程序设计模式的过程，也是彻底明白 MFC 内部工作机制的过程。

第 8~11 章（Windows 文件操作和内存映射文件、动态链接库和钩子、TCP/IP 和网络通信、内核模式程序设计与 Windows 防火墙开发）通过对各种实例的剖析详细讨论 Windows 的高级特性，使读者在实践中提高自己的编程水平。

—— 对读者的假设

（1）读者应该熟知 C 编程语言。本书的 SDK 程序设计部分使用了 C 语言格式的例子演示 Win32 程序运行原理和 API 编程的细节。

（2）读者应该懂得 C++语言的基础知识，像简单的类的封装和对象的概念等。在这个基础上本书会详细介绍 C++的高级特性以设计和实现自己的框架和类。

（3）可视化编程的经验是有用的。如果读者曾接触过 VB、PB 或 MFC 等工具，将会更容易理解 Win32 程序的结构和 Windows 的消息驱动。

（4）本书不再假设读者有任何 Windows 编程经验和其他程序设计语言的知识。

—— 关于附书光盘和读者反馈

配套光盘包含 3 个文件夹：

- <book_code>　　此文件夹下包含本书涉及的所有 VC++工程（共 77 个，工程名称前缀 01、02、03 等说明了是哪一章的程序），将这些工程代码拷贝到硬盘上，然后直接用 VC++ 6.0 或者 VC++ .NET 打开，即可正确编译连接。
- <驱动程序向导>　　此文件夹下包含开发驱动程序所需的模板向导 DriverWizard.awx，将之复制到 VC++ 安装目录的“...\Microsoft Visual Studio\Common\ MsDev98\Template”文件夹下即可使用。
- <SkinMagic>　　此文件夹下包含美化界面工具 SkinMagic 的安装程序 setup.exe，运行它即可将 SkinMagic 2.11 版安装到电脑上。

本书附书光盘中可以找到全部例子源代码，代码全部使用 Visual C++6.0 编译通过。虽然本书中的所有的例子都已经在 Windows 98、Windows 2000 和 Windows XP 下测试通过，但由于许多工程比较复杂，也有存在 Bug 的可能，如果发现代码存在的错误或者发现书中的其他问题，请告知作者（book_better@sohu.com），以便在下一版中改进。

—— 致谢

感谢我的朋友，重庆万博软件有限公司的杨长元先生，你为我提供了一些很有价值的代码及最新的技术资料，并很有耐心地帮我调试了一些程序代码。

感谢好友冯利娜女士，在我不顺利时，你安慰、鼓舞并支持我。

同时，也感谢韩松、马伟波、陈志强等河北理工大学、天津大学的工作人员，你们给了我莫大的生活上的照顾，提出了许多宝贵的意见和建议。

作者

2005 年 3 月

目　录

第 1 章　Windows 程序设计基础

本章介绍开发 Win32 程序前的准备工作，包括了解 Windows 产品，熟悉开发工具 VC++，知道如何直接从 Microsoft 获取帮助，如何写风格固定的规范的代码等。

1.1　必须了解的东西

1.1.1　Windows 产品概述

Windows 的操作系统有：

- Windows 95、Windows 98、Windows Me、Windows 2000、Windows 2003
- Windows XP Professional、Windows XP Home
- Windows XP Media Center Edition
- Windows XP Tablet PC Edition

它们都是 32 位的操作系统，即 CPU 能同时处理的数据的位数为 32 位。Win32 指的是针对 32 位处理器设计的 Windows 操作系统。

本书要讨论的是 32 位环境下 Windows 应用程序设计。Microsoft 公司为每一个平台都提供了相同的应用程序编程接口（application programming interface，即 API），这意味着如果学会了为一个系统平台编写应用程序，那么也就知道了如何为其他平台编写程序了。

本书主要讲解如何使用 Windows API 函数写 Windows 应用程序，它适用于所有的系统平台。事实上，系统间的差异是存在的，不同系统提供的函数可能是以不同的方式运行，这在书中会尽量指出。

现在，主流的操作系统是 Windows XP，“XP”代表的英文单词是“experience”，象征着此系列的操作系统能够给用户带来新的体验。本书中的例子都是在 XP 系统下完成的。同样，笔者也假定您使用的操作系统是 Windows XP 或 Windows 2000，或者是更高的版本。

1.1.2　开发工具 Visual C++

Visual C++是 Windows 环境下最优秀的 C++编译器之一，它是 Microsoft 公司开发的 Visual Studio 系列产品的一部分。Visual C++ .NET 2003 是目前此系列产品的最新版本，但是由于 Visual C++ 6.0 小巧易用，对计算机的软硬件环境要求比较低，而且能够胜任几乎全部的 Windows 应用程序的开发工作，所以大部分软件开发公司主要的开发平台还是 Visual C++ 6.0。本书的例子代码是用 Visual C++ 6.0 编写并编译的，不过它们也可以在.NET 下编译通过。

建议您也使用 Visual C++ 6.0 来学习 Windows 程序设计，等有了一些软件开发经验之后再去研究.NET 提供的新功能。为了能够在 Visual C++ 6.0 中使用操作系统的新特性，只需更新一下 SDK 工具即可。

SDK 是 Software Development Kit 的缩写，意思是软件开发工具箱。Microsoft 公司的 Platform SDK 为开发者提供了开发 Windows 应用程序必要的文档、头文件和例子代码。

VC++ 6.0 自带的 SDK 工具太老了（1998 年的），对许多新的特性都不支持。在写这本书时 Microsoft 公司刚刚发布了适用于操作系统 Microsoft Windows XP Service Pack 2 的最新的 SDK，可以很方便的从“ http://www.microsoft.com/msdownload/platformsdk/sdkupdate/ ”站点下载。这个 SDK 既能够被 Visual C++ 6.0 使用，也能够被 Visual C++ .NET 使用。它可以保证您拥有适用于 Windows XP Service Pack 2 发行版的最新的文档、例子和 SDK 构造环境（包括头文件、运行期库和工具）。

此 SDK 中的 SDK 文档提供了为各种版本的 Windows 系统开发应用程序所需的应用程序编程接口（Application Programming Interfaces，API 函数）的帮助信息，它还包含关于 Windows Server 2003 的最新信息。

当然，没有必要非得更新 VC++ 6.0 自带的 SDK，本书会在需要的地方明确地指出。

1.1.3 Windows 资料来源——MSDN

MSDN 是微软程序员开发网络（Microsoft Developer Network），是为帮助开发人员使用 Microsoft 的产品和技术写应用程序的一系列在线或者离线的服务。

在使用 VC++ 6.0/.NET 编写程序时，如果想动态地获取帮助，那么就应该安装 MSDN。MSDN 中包含了编程信息、技术论文、文档、工具、程序代码以及新产品的 Beta 测试包。而且，MSDN 也包含相应版本的 SDK 工具。想成为高手必须学会自己查阅 MSDN（或 SDK 文档）来解决问题。

1.1.4 Win32 API 简介

API 是 Application Programming Interface 的简写，意思是应用程序编程接口。可以把它想像成一个程序库，提供各式各样与 Windows 系统服务有关的函数，例如 CreateFile 是用来创建文件的 API 函数，C 的标准库函数 create 也提供了创建文件的函数，但是它是靠调用 CreateFile 函数完成创建文件功能的。事实上，在 Windows 下运行的程序最终都是通过调用 API 函数来完成工作的，因此，可以把 Win32 API 看成是最底层的服务。

通常所说的 SDK 编程就是直接调用 API 函数进行编程。但是 API 函数数量众多，详细了解每一个函数的用法是不可能的，也是完全没有必要的。只需知道哪些功能由哪些 API 函数提供就行了，等使用它们时再去查阅帮助文件。

Win32 API 是指编制 32 位应用程序时用的一组函数、结构、宏定义。在 Win32 的环境下，任何语言都是建立在 Win32 API 基础上的，只不过 Visual FoxPro、Visual Basic 等软件对 API 封装得很深。以后本书讨论的应用程序全是通过直接调用 API 函数来实现的（介绍内核驱动的章节除外）。

1.2 用 Visual C++的基本用法

本节讲述编写控制台应用程序的方法和如何在程序中调用 API 函数。这些知识非常简单，

目的是让从没有接触过 VC++ 6.0 的读者能够轻松入门。

1.2.1　应用程序的类型

Windows 支持两种类型的应用程序：一种是基于图形用户界面（Graphical User Interface，简称 GUI）的窗口应用程序，这是大家常见的 Windows 应用程序；另一种是基于控制台用户界面（Console User Interface，简称 CUI）的应用程序，即“ MS-DOS ”界面的应用程序。不要以为使用控制台环境的程序就不是 Windows 程序，它可以使用所有的 Win32 API，甚至可以创建窗口进行绘图。所以，这两种应用程序类型间的界限是非常模糊的。

控制台应用程序不需要创建自己的窗口，其输入输出方式也很简单。从这里开始有利于初学者抛开复杂的 Windows 界面管理和消息循环，而去专心研究 API 函数的细节，了解常用的内核对象。下一小节具体介绍如何用 VC++ 6.0 创建控制台应用程序。

1.2.2　第一个控制台应用程序

本节不会全面介绍 VC++ 6.0 的使用方法，而是在后继章节中陆续地介绍。等看完本书后，相信您对集成编译器的使用就很清楚了。下面是使用 VC++ 6.0 创建控制台应用程序的整个过程：

（1）运行 VC++ 6.0，选择菜单命令“ File/New ”，在打开的 New 对话框中打开 Projects 选项卡，选项卡左侧的列表框中有多种工程类型，单击 Win32 Console Application（控制台程序）选项，然后在右侧的“ Project name ”中输入工程名 01FirstApp，在“ Location ”中输入存放工程文件的路径“ E:\MYWORK\BOOK_CODE ”，如图 1.1 左图所示。

（2）单击 New 对话框的 OK 按钮，出现如图 1.1 右图所示的对话框。在这个对话框里，VC 要求你选择一个控制台应用程序的类型，它们分别是空的工程（An empty project），简单的程序（A simple application），能够打印出“ Hello, World! ”字符串的程序（A “Hello, World!” application）、支持 MFC 的应用程序（An application that supports MFC）。在这里选择第三种。

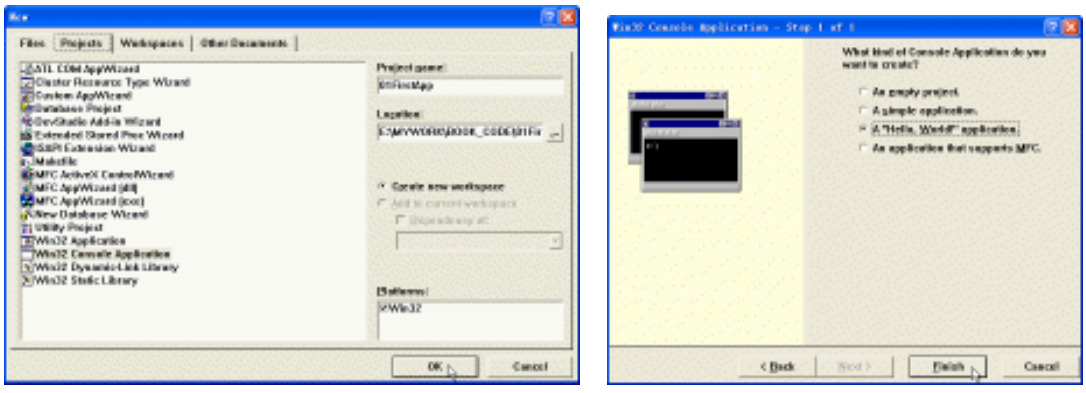

图 1.1　创建 Win32 Console Application 工程

（3）单击 Finish 按钮，弹出一个消息框，直接单击 OK 按钮即可建立一个简单的工程框架，其中含有 VC 自动生成的程序入口函数 main，如图 1.2 所示。

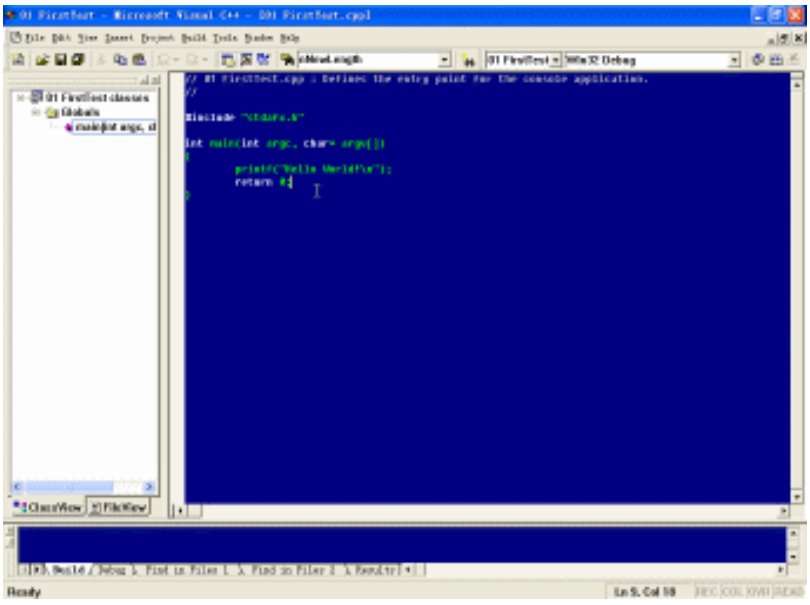

图 1.2　最终生成的工程

在第二步时，也可以选择其他的控制台工程类型，其结果大同小异，只是 VC 自动生产的代码不同而已。例如，可以选中第一个选项“ An empty project ”，即建立一个空的工程，然后自己向工程中添加文件（通过菜单命令“ File/New ”的 Files 选项卡）并定义入口函数 main。

现在，向 01FirstApp 工程中添加自己的文件或代码就行了。程序编写完毕可以按<Ctrl>＋F7 键编译，按 F7 键编译连接，按<Ctrl>＋F5 键运行程序。

如果要将存在的文件添加到工程中，使用菜单命令“ Project/Add To Project/Files ”即可。

1.2.3　API 函数的调用方法

在 VC++ 6.0 下使用 API 函数是非常方便的，只要在文件的开头包含上相应的头文件，然后在程序中直接调用它们就可以了。下面是一个调用 API 函数的例子，修改 01FirstApp 工程中 main 函数的实现代码如下。

```
#include "stdafx.h"        // 这是 VC 自动添加的头文件。为了减少文件间的依赖性，本书建议不使用它
#include <windows.h>   // 包含 MessageBox 函数声明的头文件


int main(int argc, char* argv[])
{
    // 调用 API 函数 MessageBox
    int nSelect = ::MessageBox(NULL, "Hello, Windows XP", "Greetings", MB_OKCANCEL);
    if(nSelect == IDOK)
        printf("  用户选择了“确定”按钮 \n");
    else
        printf("  用户选择了“取消”按钮 \n");
```

```
    return 0;
}
```

运行程序，除了显示一个控制台外还会弹出一个小对话框，如图 1.3 所示。

图 1.3　MessageBox 函数的调用结果

MessageBox 是众多的 API 函数中的一个，它声明在 windows.h 文件中，用于显示一个指定风格的对话框。在自己的程序中调用 API 函数的方法非常简单，具体步骤如下：

（1）包含要调用函数的声明文件。

（2）连接到指定的库文件（即 lib 文件）。VC 默认已经连接了常用的 lib 文件，所以一般情况下，这一步对我们是透明的。如果需要显式设置的话（比如在网络编程时需要添加 WS2_32.lib 库），可以在文件的开头使用‘ #pragma comment(lib, "mylib.lib") ’命令。其中 mylib.lib 是目标库文件。

（3）在 API 函数前加“ :: ”符号，表示这是一个全局的函数，以与 C++类的成员函数相区分。

如果想获得某个 API 函数详细信息，只需将光标移向此函数，按键 F1 即可。也可以打开 MSDN 文档（或 SDK 文档），直接将函数名输入到索引栏来查找函数。

1.3　本书推荐的编程环境

程序开发者长时间盯着屏幕，对自己眼睛的伤害比较大。在编写程序时可以通过改变编程工具的默认颜色来减少显示器对眼睛的伤害。具体做法如下：

（1）单击菜单“ Tools/Options ”，弹出 Options 窗口，在 Format 页中选取 Category 中的 All Windows 项，如图 1.4 所示。

（2）在 Colors 栏中对文本颜色、背景色、关键字的颜色等进行设置。

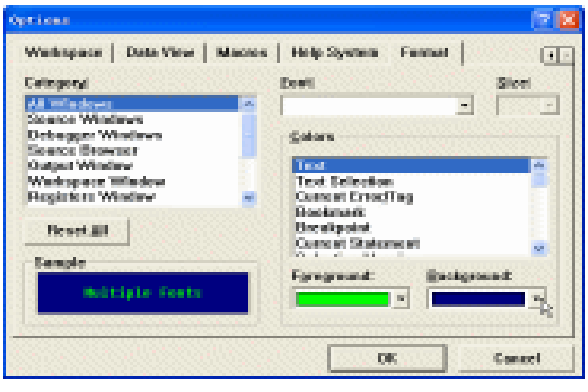

图 1.4　设置编辑器的颜色