黑客防线

黑客攻防

编程解析

潘志翔　岑进锋　编著

机械工业出版社
CHINA MACHINE PRESS

黑客防线

黑客攻防编程解析

潘志翔　岑进锋　编著

机 械 工 业 出 版 社

黑客的攻击与防守是矛盾的双方，本书从编程和网络技术的角度，深入探讨了“攻”与“防”的技术，提供了一些主要用 C/C++ 来描述的有参考价值的代码。这些代码都已在 Visual C++ 6.0 或 C++ Builder 5.0 环境下编译调试通过。

本书首先介绍了攻防编程的基础知识，如 Windows 内部机制简述、编程的方法、工具和技巧等；然后剖析了特洛伊木马、扫描器、病毒等程序代码，揭露了多种黑客攻击的技术内幕；介绍了很有参考价值的 API 函数及编程的好书、好网站，以便开拓读者的视野。本书还着重介绍了个人防火墙方面的编程技术，为个人防火墙编程爱好者提供了实用的学习参考资料。

图书在版编目（CIP）数据

黑客攻防编程解析/潘志翔，岑进锋编著．—北京：机械工业出版社，2003.6
（黑客防线）
ISBN 7-111-12243-7

Ⅰ.黑… Ⅱ.①潘…②岑… Ⅲ.计算机网络－安全技术 Ⅳ.TP393.08

中国版本图书馆 CIP 数据核字（2003）第 039396 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
责任编辑：田 梅
责任印制：付方敏
北京忠信诚胶印厂印刷·新华书店北京发行所发行

2003 年 6 月第 1 版·第 1 次印刷
787mm×1092mm 1/16· 15.25 印张·376 千字
0001—5000 册
定价：23.00 元

出版说明

近年来，计算机网络在国内得到了迅速的发展。在网络的大量应用中，安全正面临着前所未有的挑战。信息安全已经成为一个综合的工程，甚至将成为一个新兴的研究学科，需要我们在网络安全领域进行长期的研究和攻关。

网络的基础在于资源的共享，这一直是网络的基本准则。随着 Internet 的飞速发展，网络上的资源共享越来越强化。随之而来的，网络安全问题也越来越突出了。网络在带给人们诸多便利的同时，也成了许多犯罪分子攻击的目标。他们以计算机为工具，同时又以计算机为目标，在网上对计算机数据信息进行恶意的修改、删除，从而造成计算机系统难以正常运行甚至瘫痪。如果我们从另一方面去看问题，黑客也使我们发现自己网络的缺陷并改进它。从某种意义上说，日益完善的安全系统和逐渐完美的防火墙，是和黑客技术密不可分的。黑客的存在是网络发展的必然结果，尤其在我国，互联网络还处于雏形阶段，存在着不可忽视的缺陷与漏洞。如何改良网络结构，完善网络安全体系，是我们的当务之急。政府部门也对网络信息安全非常重视，并鼓励大力发展信息安全事业，以使我国在全球信息网络化的发展中占据主动地位。

目前，社会上对精通网络与信息安全知识的人才需求越来越强烈，广大技术人员和网络用户也十分希望能迅速提高自己应对安全问题的能力。由此，机械工业出版社联合北京地海森波网络技术有限公司《黑客防线》编辑部共同策划出版了“黑客防线”丛书，旨在为读者提供有关网络安全方面的知识和技术，从不同侧面阐述网络安全的相关技术。在丛书撰写过程中，切实考虑读者对知识的需求，内容做到通俗易懂，其中涉及的很多技术都是工作在网络安全第一线作者的心血结晶。对从事网络安全事业的技术人员来说，本套丛书是一个很好的帮手，从中可学到很多实用技术和宝贵经验，从而得心应手地应对各种网络安全问题。对于那些想学习网络安全知识和技术的读者而言，本套丛书也不失为好的学习工具书，通过学习不仅能迅速掌握网络安全知识，提高自身防范能力，而且为走上网络安全事业的道路奠定了基础。

我们始终坚持以普及网络安全知识，加强全民安全意识，提高我国信息技术和网络安全水平为己任，希望这套丛书的出版能满足读者的需求，并请广大读者批评指正，提出宝贵意见。

机械工业出版社

前　言

随着 Internet 的普及，在大家安逸地享受着 Internet 带来“恩赐”的同时，随之而来的却是黑客的攻击、数据的被窃取，一声“现在的网络极不安全，你随时都会面临惨重的损失！”终于敲醒了人们沉睡的头脑！于是，网络信息安全自然成为当前的热门话题。

随着计算机网络的发展和黑客攻击技术的层出不穷，网络安全技术也争先恐后地浮出水面。各大电脑保护软件、防火墙、入侵检测系统的开发公司纷纷拔地而起，在这个虚拟的 Internet 世界里掀起了“百花齐放，百家争鸣”的浪潮。

为了增加读者对编程和网络技术的了解，深入探索“攻”与“防”这两个相互矛盾的技术，一方面，让读者从另外一个角度来了解“攻”并不只是拿现成的工具用用就可以了，还应该深入了解其中的原理，不但“知其然”，还要“知其所以然”；另一方面，读者在了解“攻”的技术的同时，还要了解如何“防”，增强读者的防范意识和动手解决问题的能力。让读者在阅读本书的同时，能从另一个角度来理解黑客技术，这也是本书的一个重要指导思想。

本书提供了一些有参考价值的代码，代码主要用 C/C ++ 来描述，并且均在 Visual C ++ 6.0 或 C ++ Builder 5.0 环境下编译调试通过。书中部分章节还涉及到了汇编语言，所有的代码都附上了详细的解释，力求让每一位有一定 C 语言和汇编语言基础的读者都能明白本书的程序。全书内容实用，示例丰富，适合初学者和有一定编程基础想深入 Windows 编程的朋友学习和参考。

本书遵循由浅入深的原则，在内容安排上做到层次分明，由易到难、由攻到防。本书共分 11 章：第 1 章的 Windows 编程基础讲述了 Windows 编程的一些基础知识，其中包括一些操作的方法和技巧。第 2 章的迈进攻防编程介绍了 Windows 网络编程的基础、PE 文件结构等内容。第 3 章的特洛伊木马详尽地描述了木马的方方面面，并附有一个自制木马的源程序。第 4 章的扫描器细致地讲述了扫描器的工作原理和实现方法等。第 5 章的恶作剧程序剖析深入探讨了恶作剧程序内幕。第 6 章的病毒详细介绍了病毒的代码分析以及防范措施。第 7 章的缓冲区溢出攻击将从编程的角度来分析它的根源。第 8 章的 DoS 与 DDoS 攻击将介绍拒绝服务攻击的基本编程方法，同时也介绍防范措施。第 9 章的文件安全将介绍如何加密数据文件来防止它的泄漏。第 10 章的 Sniffer 的编写主要介绍它的实现原理及流程。第 11 章的个人防火墙编程技术将把个人防火墙从原理到实现一一展示在读者面前。附录 A 将介绍相关的 API 知识，相当一部分是微软没有公开的 API。附录 B 将介绍部分编程方面的好书。

在本书的写作过程中得到了刘思阳、谭登元等同志的大力支持，并提出不少指导性的建议，在此对他们表示由衷地感谢！此外还要特别感谢《黑客防线》丛书执行主编郭聪辉给我们的帮助与关心！

由于写作时间仓促以及编者水平所限，书中难免存在错误和疏漏之处，恳请读者批评、指正。

编　者

目　录

出版说明
前言

基　础　篇

第 1 章　Windows 编程基础 …… 2
1.1　了解 Windows 内部机制 …… 2
1.1.1　引言 …… 2
1.1.2　Windwos 内部机制简述 …… 2
1.1.3　Windows 编程的几个相关概念 …… 2
1.1.4　从程序角度来看 Windows 内部机制 …… 5
1.2　学习 Windows 编程的方法论 …… 8
1.2.1　编程语言及编程工具的选择 …… 8
1.2.2　提高编程能力的两个途径 …… 8
1.3　学习使用编程工具 …… 9
1.3.1　编程工具一览 …… 9
1.3.2　两款杰出的编程工具的使用方法 …… 10
1.3.3　一个简单的编程例子 …… 13
1.4　攻防编程的几个重要技巧 …… 17
1.4.1　技巧一：如何用程序操作注册表 …… 17
1.4.2　技巧二：如何在程序中调用 API …… 22
1.4.3　技巧三：运用多线程编程技术 …… 23
1.4.4　技巧四：实现后台监控程序 …… 24
1.4.5　技巧五：使用定时触发器 …… 25
第 2 章　迈进攻防编程 …… 26
2.1　引言 …… 26
2.2　Socket 编程基础 …… 26
2.2.1　什么是 Socket …… 26
2.2.2　对 Socket 进一步理解 …… 27
2.2.3　Socket API 介绍 …… 28
2.2.4　微软对 Socket API 的扩展 …… 33
2.2.5　应用举例 …… 39
2.3　网络协议与数据报 …… 42
2.3.1　介绍 …… 42
2.3.2　以太网数据包头(Ethernet Header)介绍 …… 43
2.3.3　IP 协议数据报结构介绍 …… 44
2.3.4　TCP 协议的数据报结构介绍 …… 46
2.3.5　UDP 协议的数据报结构介绍 …… 47
2.3.6　ICMP 协议的数据报结构介绍 …… 48

2.4 PE 文件及格式介绍 …… 49
2.4.1 什么是 PE 文件 …… 49
2.4.2 PE 文件结构介绍 …… 49
2.5 小结 …… 53

攻 击 篇

第 3 章 特洛伊木马 …… 55
3.1 特洛伊木马的介绍 …… 55
3.1.1 特洛伊木马名字的由来 …… 55
3.1.2 特洛伊木马的工作机理 …… 55
3.2 特洛伊木马的编程探索 …… 56
3.2.1 特洛伊木马的编程思路 …… 56
3.2.2 特洛伊木马编程的技术细节 …… 56
3.3 特洛伊木马的源代码分析 …… 64
3.3.1 写在前面 …… 64
3.3.2 源程序解释 …… 64
第 4 章 扫描器 …… 81
4.1 扫描器简介 …… 81
4.1.1 扫描器的定义、功能及分类 …… 81
4.1.2 常见扫描器的功能介绍 …… 81
4.2 扫描器的编程思路 …… 82
4.2.1 扫描器的工作原理 …… 82
4.2.2 编程预备知识 …… 82
4.2.3 扫描器各大功能的编程实现 …… 83
4.2.4 一个简单的扫描器的原码分析 …… 84
4.3 自己动手写一个简单的扫描器 …… 88
4.3.1 准备工作 …… 88
4.3.2 程序核心部分 …… 88
第 5 章 恶作剧程序剖析 …… 99
5.1 可执行文件型恶作剧程序 …… 99
5.1.1 恶作剧程序概述 …… 99
5.1.2 恶作剧现象一览 …… 99
5.1.3 恶作剧程序的几个着眼点 …… 100
5.1.4 恶作剧程序的实现代码探索 …… 101
5.2 网页恶意攻击 …… 104
5.2.1 网页恶意攻击初探 …… 104
5.2.2 网页恶意攻击深入剖析 …… 105
5.2.3 网页恶意代码分析及相应的防范措施 …… 106
5.3 邮件型恶作剧脚本 …… 115
5.3.1 邮件型恶作剧脚本概述 …… 115
5.3.2 邮件型恶作剧原理分析 …… 115
5.3.3 一封恶意邮件的源码分析 …… 116

第6章 病毒 …… 119
6.1 计算机病毒简述 …… 119
6.1.1 计算机病毒的定义 …… 119
6.1.2 计算机病毒的历史 …… 119
6.2 文件型病毒 …… 120
6.2.1 文件型病毒传染的机理 …… 120
6.2.2 文件型病毒源码剖析 …… 120
6.2.3 文件型病毒的防范 …… 129
6.3 电子邮件病毒 …… 129
6.3.1 电子邮件病毒的运作机理 …… 129
6.3.2 一种电子邮件病毒的源代码分析 …… 132
6.3.3 如何防范电子邮件病毒 …… 146
第7章 缓冲区溢出攻击 …… 151
7.1 缓冲区溢出简介 …… 151
7.1.1 缓冲区溢出的概念及原理 …… 151
7.1.2 缓冲区溢出的危害 …… 151
7.1.3 最新的缓冲区溢出 …… 152
7.1.4 一道最简单的缓冲区溢出程序 …… 152
7.2 缓冲区溢出的防范建议 …… 153
第8章 DoS与DDoS …… 156
8.1 DoS和DDoS攻击简介 …… 156
8.2 如何进行DoS、DDoS攻击 …… 157
8.3 构造DoS和DDoS攻击 …… 158
8.3.1 DoS、DDoS攻击的原理 …… 158
8.3.2 源代码分析 …… 159
8.4 如何防范DoS、DDoS攻击 …… 165

防 守 篇

第9章 数据文件的安全 …… 168
9.1 数据文件的加密 …… 168
9.1.1 引言 …… 168
9.1.2 基本概念 …… 168
9.1.3 加密数据的通信过程 …… 170
9.1.4 简单加密算法举例 …… 171
9.2 可执行文件的加密 …… 174
9.2.1 引言 …… 174
9.2.2 加密原理 …… 174
9.2.3 具体实现 …… 177
第10章 Sniffer的编写 …… 186
10.1 初识Sniffer …… 186
10.2 编写Sniffer程序 …… 187
10.2.1 实现原理及流程 …… 187

10.2.2 程序讲解 …… 188
第 11 章 个人防火墙编程技术 …… 196
11.1 个人防火墙介绍 …… 196
11.2 个人防火墙的编程设计 …… 196
11.2.1 功能简介 …… 196
11.2.2 用法简介 …… 197
11.2.3 主程序工作流程 …… 197
11.2.4 如何实现 SPI HOOK …… 199
11.2.5 如何实现 NDIS HOOK …… 203
11.2.6 如何判断数据包的合法性 …… 214
11.2.7 如何屏蔽指定 IP 地址的访问 …… 222
11.2.8 如何屏蔽指定端口的访问 …… 225
11.2.9 如何防止被探测 …… 226
11.2.10 小结 …… 227
附录 …… 228
附录 A 相关 API 介绍 …… 228
附录 B 有关编程的好书好网站 …… 235

基 础 篇

作为本书的基础篇，我们为读者介绍了攻防编程方面的基本知识，第 1 章是专门为初学者准备的，从最简单的基本概念入手，接着讲述编程工具的使用和提高编程能力的一些途径。并介绍了几个基本的技巧，这些技巧看起来简单，但是如果运用得好，将会使你的程序功能增强。

在第 2 章里我们带领大家步入网络编程的殿堂。如果您还是一个初学者，没有多少编程的知识，请多多关注本篇内容。

第 1 章　Windows 编程基础

1.1　了解 Windows 内部机制

1.1.1　引言

在这一章中，首先从最基础、最简单的部分开始介绍。相信大家都用过微软的 Windows 操作系统，它是一个操作极为方便的视窗系统，尽管每天都在面对和使用它，然而仍然还有很多人还不了解它的内部机制，因此在这里简单讲述一下它的内部机制，为后面的编程作个铺垫。

从理论上说，任何一门语言都可以在任何一个系统上编程，只要找到系统为该门语言提供的“接口”和编程者对系统内部机制有深入的了解，正如 C 语言可以在 Windwos 下编程，同样也可以在 Linux 上大放异彩一样。

编程是一项很繁杂的工作，除了熟练使用编程工具外，还要了解系统本身的内部工作机理，这是写出稳定兼容的程序所必不可少的前提条件。在哪一种系统上编程就要对该系统的机制进行研究，至少应该知道一个程序在那个系统上是如何运行的，下面我们将详细介绍 Windows 系统的内部机制。

1.1.2　Windwos 内部机制简述

Windows 是一个“基于事件的，消息驱动的”操作系统。

在 Windows 下执行一个程序，只要用户进行了影响窗口的动作（如改变窗口大小或移动、单击鼠标等），该动作就会触发一个相应的“事件”。系统每次检测到一个事件时，就会给程序发送一个“消息”，从而使程序可以处理该事件。每个 Windows 应用程序都是基于事件和消息的，而且包含一个主事件循环，它不停地、反复地检测是否有用户事件发生。每次检测到一个用户事件，程序就对该事件做出响应，处理完后再等待下一个事件的发生。

Windows 的应用程序不断地重复着这一过程，直至用户终止该程序。如果用代码来描述，实际上这是一个消息处理过程的 while 循环语句。

1.1.3　Windows 编程的几个相关概念

下面简单介绍一下与 Windows 系统密切相关的几个基本概念。

1.窗口

大家都知道，窗口是 Windows 本身以及 Windows 环境下的应用程序的基本界面单位，但是很多人都误以为只有具有标题栏、状态栏，最大化、最小化等按钮的标准方框才叫窗口，其他的不规则的窗口就不是窗口。其实窗口的概念很广，例如按钮和对话框等也是窗口，只不过是一种特殊的窗口。

从用户的角度看，窗口就是显示在屏幕上的一个矩形区域，其外观独立于应用程序，而事

实上它就是生成该窗口的应用程序与用户间的直观接口;从应用程序的角度看,窗口是受其控制的一部分矩形屏幕区。应用程序生成并控制与窗口有关的一切内容,包括窗口的大小、风格、位置以及窗口内显示的内容等。当用户打开一个应用程序后,程序即创建了一个窗口,当用户选择窗口中的选项时,程序就会对此做出响应。

2. 程序

通常说的程序都是指一个能让计算机识别的文件,我们平常接触得最多的便是.exe 型的可执行文件。

3.进程

学过操作系统的人都很清楚,所谓进程就是应用程序的执行实例(或称一个执行程序)。需要注意的是:进程是程序动态的描述,而上面说到的程序是静态的描述,两者有本质的区别。例如,从网上下载一个瑞星杀毒软件到 C 盘但是没有运行它,那个.exe 可执行文件就叫做程序,它是一个二进制码的文件。如果双击 exe 文件图标来运行该程序,那个"正在运行着的瑞星杀毒"便称为进程,它在双击的那一刻就被系统创建,当关机或者在任务栏的图标上单击鼠标右键,在弹出的快捷菜单中选择"退出"命令时,这个进程便结束了。进程经历了由"创建"到"消亡"的生命期,而程序自始至终存在于硬盘上,不管机器是否启动。

4. 线程

线程是进程中的一个执行单元,同一个进程中的各个线程对应于一组 CPU 指令、一组 CPU 寄存器以及一堆栈。进程的动态过程,我们实质上是通过线程来执行体现的,从这个意义上说,Windows 中进程的动态意义已经不是很明显了,只是给程序所占的资源划定一个范围而已,真正具有动态性意义的是线程。在后面的篇章中我们将会介绍到多线程编程的技巧,所以要求读者很清楚的理解这些内容,以便应用到实践中去。

5. 消息

在进行操作时,几乎每做一个动作都会产生一个消息,在用鼠标"指点江山"的今天,鼠标的每一次移动都会产生 WM _ MOUSEMOVE 消息,如鼠标左键被按下时会产生 WM _ LBUT-TONDOWN 消息,鼠标右键按下时会产生 WM _ RBUTTONDOWN 消息等。所有这些都可以通过 GetMessage、SendMessage 等函数得到,在以后的讲解过程中会经常接触到这些函数。

6. 事件

何谓事件? 从它的字面就可以明白它的含义,如在程序运行的过程中改变窗口的大小或者移动窗口等,都会触发相应的"事件"。

7. 句柄

打个比喻来说,夏天在摇扇子的时候只要抓住扇柄就可以控制整个扇子的运动,在程序中也是这个意思。通常一个句柄就可以传递我们所要做的事情,编写程序总是要和各种句柄打交道的。句柄是系统用来标识不同对象类型(如窗口、菜单等)的工具,不同类型的对象,对应着不同的句柄。

在 C++ 教材中是这样给句柄下定义的:"在 Win32 里,句柄是指向一个无值型对象(void *)的指针,是一个 4 字节长的数据"。虽然读者对它的本质是什么还并不很清楚,但应该知道句柄并不是一个真正意义上的指针。从结构上看,句柄的确是一个指针,尽管它没有指向用于存储某个对象的内存位置,而实际上句柄指向的是一个包含了对该对象进行引用的位置。在

编程时,只要明确了对象的句柄就可以对该对象进行操作了。下面举个例子来说明句柄的运用:编一个程序,使 QQ 登录窗口的号码框和密码框均变黑,相关代码及解释如下:

```
void __fastcall TForm1::FormCreate(TObject *Sender)
{
HWND hCurWindow,HC,HE;
//定义三个窗口句柄变量,hCurWindow 用于存放 QQ 用户登录窗口的句柄,HC、HE 分别存放号码框和密码框的句柄
        if((hCurWindow = FindWindow(NULL,"QQ 用户登录"))!=0||(hCurWindow = FindWindow(NULL,"OICQ 用户登录"))!=0)
        {
//调用 FindWindow()函数去获得 QQ 登录窗口的句柄
          String str;
          str.sprintf("0x%x",hCurWindow);
        }
TCHAR wClassName[255];
//类名变量
HC = GetWindow(hCurWindow, GW_CHILD);
//得到号码框的句柄
HE = GetWindow(HC, GW_HWNDNEXT);
//得到密码框的句柄
GetClassName(HE, wClassName, sizeof(wClassName));
//得到类名
GetClassName(HC, wClassName, sizeof(wClassName));
//得到类名
EnableWindow(HE,false);
//使窗口失效
EnableWindow(HC,false);
//使窗口失效
}
```

以上代码在 C++ Builder 下编译通过,只要运行此程序,QQ 登录窗口的号码框和密码框马上变黑色。如图 1-1 所示,这里只是调用了一个 EnableWindow()函数而出现的效果。

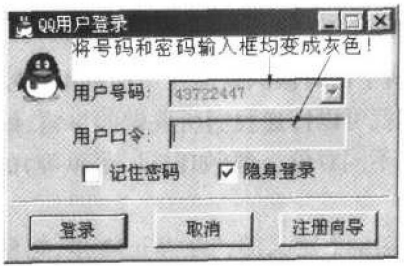

图 1-1 调用 EnableWindow()函数后的效果

还可以添加一个 Timer 控件,将上面的代码复制到 void __fastcall TForm1::Timer1Timer (TObject *Sender)函数中,并在后边加上这一句代码:

SendMessage(hCurWindow,WM_CLOSE,0,0);

这样,程序启动后,一直在监视,看有没有 QQ 登录窗口的出现,一旦有就关闭它,使别人用不了 QQ。

8. API 与 SDK

API 是英文 Application Programming Interface 的简称,意为"应用程序接口",泛指系统为应用程序提供的一系列接口函数。其实质是程序内的一套函数调用,在编程的时候可以直接调用,而不必知道其内部实现的过程,只知道它的原型和返回值就可以了,此外,初学者最好阅读一下关于《Windows API 大全》之类的书,这样对理解 API 的内容很有好处。在后面会介绍调用 API 编程的例子,调用 API 编程工作虽然烦琐,但由于 API 函数都被封装在 dll 库里,程序只有在运行的时候才调用它,因此程序的体积小而且运行效率高。

SDK 是英文 Software Development Kit 的缩写,指"软件开发工具包",在防火墙的设计中就经常涉及到 SDK。

有关基本概念就先介绍这些,至于 C/C++ 的基本语法、面向对象的含义等知识,读者可以查阅相关的书籍,这些相关概念在很多书籍上都可以见到。

1.1.4 从程序角度来看 Windows 内部机制

上面所提到的都是一些理论知识,是从文字的角度去理解 Windows 的,现在从代码的角度来认识。下面的代码虽然有点长,但这是一道用 C 语言写的最简单的程序,是在屏幕上显示一个标准窗口,并在窗口的工作区内显示一串字符串。笔者在代码中间加上了详细的解释来帮助读者理解,一旦读者读懂了这道简单的程序就会初步明白 Windows 系统是如何工作的、程序是如何运行的。

1. 程序的头文件 Simpwin.h

```
//函数原型

//过程
LRESULT CALLBACK MainWndProc(HWND,UINT,WPARAM,LPARAM);
//函数
BOOL InitApplication(HINSTANCE);
BOOL InitInstance(HINSTANCE,int);
char *hello="中秋节快乐";
//定义一串字符串,这是窗口输入的文本
```

2. 程序的主文件 Simpwin.c

```
#include<windows.h>   //嵌入基本的 Windows 功能

#include<string.h>
//包含自定义的头文件
#include"simpwin.h"
HINSTANCE hInst;
//当前实例句柄
```

```
HWND hWndMain;
//主窗口句柄
//程序主函数 WinMain 调用初始化函数,处理消息循环
int APIENTRY WinMain(
HINSTANCE hInstance,
//当前实例句柄
HINSTANCE hPrevInstance,
//前一个实例句柄
LPSTR lpCmdLine,
//命令行字符串
int nCmdShow)
//窗口显示方式
{
MSG msg;
//初始化窗口数据,并注册窗口类
if (! InitApplication(hInstance))
return(FALSE);
//创建和显示窗口,对应用程序进行必要的初始化
if(! InitInstance(hInstance,nCmdShow))
return(FALSE);
//当进入消息循环,从应用程序消息队列中检取消息
//当检取的消息是一条 WM_TUIT 消息时,就退出消息循环
while (GetMessage(&msg,NULL,0,0))
{
    //把虚拟键消息翻译为字符消息
    TranslateMessage(&msg);
    //把消息分配到相应的窗口过程中去
    DispatchMessage(&msg);
}
    return(msg.wParam);
}
BOOL InitApplication(HINSTANCE hInstance)
//当前实例句柄
{
WNDCLASS wcSimpwin;
//填写窗口类结构,使得其参数描述主窗口各方面的属性
wcSimpwin.style = 0;
wcSimpwin.lpfnWndProc = (WNDPROC) MainWndProc;
wcSimpwin.cbClsExtra = 0;
wcSimpwin.cbWndExtra = 0;
wcSimpwin.hInstance = hInstance;
wcSimpwin.hIcon = LoadIcon(NULL,IDI_APPLICATION);
wcSimpwin.hCursor = LoadCursor(NULL,IDC_ARROW);
```

```
 wcSimpwin.hbrBackground = GetStockObject(WHITE_BRUSH);
 wcSimpwin.lpszMenuName = NULL;
 wcSimpwin.lpszClassName = "SimpwinWClass";
 //对窗口类进行注册
 return(RegisterClass(&wcSimpwin));
 }
 BOOL InitInstance(HINSTANCE hInstance,int nCmdShow)
 {
          hInst = hInstance;
hWndMain = CreateWindow("SimpwinWClass","我爱你们",WS_OVERLAPPEDWINDOW,
CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT, NULL, NULL, hInstance,
NULL);
// 如果创建窗口失败,则返回 FALSE
if (! hWndMain)
return(FALSE);
//让窗口显示出来,并更新其客户区,最后返回 TRUE
ShowWindow(hWndMain,nCmdShow);
UpdateWindow(hWndMain);
return(TRUE);
}
LRESULT CALLBACK MainWndProc(
          HWND,hWnd,          //窗口句柄
          UINT message,       //消息类型
WPARAM wParam,                //消息附带信息
LPARAM lParam)                //消息附带信息
{
HDC hdc;
PAINTSTRUCT ps;
switch (message)
{
     case WM_PAINT:hdc = BeginPaint(hWnd, &ps);
     TextOut(hdc,20,10,hello,lstrlen(hello));
     EndPaint(hWnd, &ps);
     break;
     case WM_DESTROY:      //消息:本窗口正将被销毁
     //请求退出窗口和应用程序
     PostQuitMessage(0);
     break;
     default:
     //调用默认窗口过程对未处理的消息进行必要的处理
     return(DefWindowProc(hWnd, message, wParam,lParam));
}
return(0);
```

以上代码说明了 Windows 程序的工作方式，只有读懂了这道程序才算对 Windows 内部机制有了初步的了解。

第1章

1.2 学习 Windows 编程的方法论

1.2.1 编程语言及编程工具的选择

下面谈谈语言和编程工具的选择问题，这也是初学者们最感到困惑的问题。

从上面的介绍中读者对 Windows 有了进一步的了解，现在就该开始行动了，选择要学的语言和工具是第一步，并且是非常重要的一步，建议读者一切以简单易行、易接受为原则，这样会比较容易开始，为以后打下良好基础。

在刚开始的时候很多人都会感到迷惑，目前的编程语言那么多，有 C、C++、C#、Java、汇编、HTML 等，究竟该选择哪一样？刚开始该学什么呢？甚至有人将 VC、C++ Builder 也列为两种不同的语言，这些都是对编程语言缺乏了解造成的。

从实用的角度来讲，C++ 是最好的选择，而 VC 和 C++ Builder 是其相应开发工具的两大主流，在这里推荐初学者使用 C++ Builder，因为这个工具简单易行，一开始就用 VC 作为入门语言，可能会打击你的自信心，不容易深入学习下去。

1.2.2 提高编程能力的两个途径

如果你是一个黑客技术的狂热者，到雅虎网(http://www.yahoo.com.cn)去搜索“黑客教程”的时候就会发现，很多的中文教程在谈到如何进行黑客编程的时候，十有八九都会介绍两大最佳途径，即读程序和写程序，并且都提出了教程作者的看法，下面谈一下笔者在这方面的个人观点。

1. 读程序

将读程序放在前面来讲是有原因的。在没有阅读过一份完整的源代码之前，不要希望能写出有多好的程序来。这是对每一位初学者的忠告也是警告，而且必须具备一定的语言基础知识，这里所说的基础知识主要是指语法知识，即最少要能读懂别人程序的每一行意思。有没有程序的设计思想，在这个时期并不重要，只要具备了一定的语法基础就可以了，思想可以通过阅读完别人的源程序后分析得来。

编程的能力包括经验、技巧、耐心等几个因素，但并非像想像中的那样简单，更不要以为编程就是简简单单地写程序。

其实学一门语言并不需要刻意去记那些条条框框的语法，在看代码的时候，遇到了不明白的地方再去查相关的资料，一点点地补充基础知识再配合源程序的思路，这时的理解才是最深刻的。可以肯定地说，这个时候对语法的接受程度绝对比你刚开始的死记硬背要有效得多。

读程序也不能单纯地读，要真正做到“俯而读，昂而思”。好的代码是百读不厌的，可以将从网上搜集来的代码打印到纸上，然后边看边做好眉批，遇到一个新函数就记下它的功能，一些忘记了的知识在旁边标出来，还可以写上对程序的看法等。特别是遇到了一些新的 API 函数，最好标出来，以后编程的时候也许会用得着，最后别忘了分析一下程序的思路，这样对你以后编写类似的程序是很有帮助的。