

C++

黑客编程 揭秘与防范

第
3
版

冀云◎编著

CC

讲解 Windows 安全和网络编程知识，包括注册表、系统服务、文件读写、进程 / 线程操作

+

介绍 Windows 下 PE 格式、调试技术、挂钩技术等各种软件安全知识

+

阐述了 Windows 内核编程以及软件逆向知识，Windows 消息机制、病毒分析、反汇编、脱壳、免杀、扫描器

C++

黑客编程

揭秘与防范

第3版

冀云◎编著

人民邮电出版社
北京

图书在版编目 (CIP) 数据

C++ 黑客编程揭秘与防范：第3版 / 冀云编著. —
北京：人民邮电出版社，2019.2
ISBN 978-7-115-49372-9

I. ①C… II. ①冀… III. ①C语言—程序设计 IV.
①TP312.8

中国版本图书馆CIP数据核字(2018)第212260号

内 容 提 要

市面上关于黑客入门的书籍较多，比如黑客图解入门类、黑客工具详解类、黑客木马攻防类等，但是，对于很多读者来说，可能并不是掌握简单的工具就够了。很多读者学习黑客知识是为了真正掌握与安全相关的知识。与安全相关的知识涉及面比较广，包括数据安全、存储安全、系统安全、Web 安全、网络安全等，本书围绕 Windows 系统下应用层的开发来介绍一些安全方面的知识。

本书是在第 2 版的基础上新添加了一些内容，同时也删除了一些过时的内容。本书以 Win32 应用层下的安全开发为中心，介绍 Windows 系统下的安全开发。

本书介绍了操作系统的相关操作，比如进程、线程、注册表等知识。当读者掌握了关于进程、线程、注册表等的开发知识后，就可以用代码实现一些常规的操作，如对进程、注册表、文件等进行操作，这样，一些日常的操作可与学习的编程知识相结合。除了操作外，本书还介绍了网络应用程序的开发技术，了解 Winsock 的开发后，读者就会明白在应用层客户端与服务器端通信的原理。当然，除了介绍 Win32 开发外，本书还介绍了 PE 结构、调试接口、逆向等知识。本书最后剖析了恶意程序、专杀工具、扫描器等的开发技术，以帮助读者更好地了解安全知识，为开发安全的软件打下基础。

◆ 编 著 冀 云
责任编辑 张 涛
责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
涿州市京南印刷厂印刷

◆ 开本：787×1092 1/16
印张：32

字数：758 千字

2019 年 2 月第 1 版

印数：1—2 400 册

2019 年 2 月河北第 1 次印刷

定价：99.00 元

读者服务热线：(010) 81055410 印装质量热线：(010) 81055316

反盗版热线：(010) 81055315

广告经营许可证：京东工商广登字 20170147 号

第2版序

我与冀云兄因参加黑客反病毒论坛会议结识，在认识初期就能感觉到冀云兄是一个非常踏实且又富有思想的人，对安全编程的诸多方面也有自己独到的认识，这令我十分欣赏。认识几个月后，通过一次无意的聊天，我有幸读到冀云兄《C++黑客编程揭秘与防范》第1版，而后承冀云兄高看，才得以诞生此序。

通过阅读《C++黑客编程揭秘与防范》第1版，我有一种相见恨晚的感觉。这本书从最基本的 Windows 编程到 Windows 下的各种安全编程技术都有涉及，例如 PE 文件、DLL 注入技术、各种 Hook 技术、后门编写的技术关键点，乃至像 MBR 的解析这种很难涉及的点与 Rootkit 编程这样比较深入的知识，都有恰到好处的介绍与详解。

因此，就整书而言，将诸如文件/注册表操作、网络通信、PE 文件、Rootkit、逆向工程等数个知识点有效组织在一起，是一个非常巨大的工程。对于有过类似写作经验的我来说，这方面的体会尤其深刻。但是不得不说，作为读者，我真的非常幸运，首先拜读了这本书。就我个人而言，这本书至少可以被当成一部“技术字典”来使用。当我在实际的工作中对某种技术生疏后，可以拿起这本书翻一翻，顿时会感觉受益匪浅。

从整书的结构以及知识的组织方式来看，不难发现，这其实是一本相当重视初学者技术的图书。第1章对于工作环境的搭建以及对应 IDE 的使用都做了必要的介绍，第2章使用一个非常有趣且简单的例子教读者如何打造一个木马的雏形，这些无不体现出了作者对于基础薄弱的读者的细心照顾。

除此之外，当前的政策环境以及社会整体的大环境都对信息安全产业释放了大量的利好信号，无论是国家将信息安全提到国家战略层面，还是发生在美国的著名“棱镜门”事件，抑或是当前的移动互联网大潮，都在预示着信息安全领域人才在未来势必将摆脱“边缘群体”，进而成为“主流群体”中重要的一员。这些改变势必将极大地加剧当前信息安全领域人才的稀缺现状，但是，我相信本书定会为中国的信息安全领域崛起贡献一份力量，进而使得更多的读者从信息安全的“门外汉”成为“圈内人”，以缓解信息安全领域人才稀缺的现状。

——任晓琿[A1Pass]，北京蓝森科技有限公司创始人，
15PB 计算机高端培训品牌创始人，《黑客免杀攻防》作者

前言

备受关注的黑客到底是什么

什么是黑客？百度百科里黑客的含义如下（摘自百度百科，略有改动）：

热衷研究、撰写程序的专才，精通各种计算机语言和系统，且必须具备乐于追根究底、穷究问题的特质。“黑客”一词是由英语 Hacker 音译出来的，是指专门研究、发现计算机和网络漏洞的计算机爱好者。早期在美国的电脑界是带有褒义的。

看到上面百度百科给出的黑客含义后，很多只会使用工具的所谓的“黑客”就能明白一个道理，即黑客是要会编写程序的。

再看一下百度百科里对只会使用工具的黑客的解释（摘自百度百科）：

脚本小子（script kiddie 或 script boy）“指的是用别人写的程序的人。脚本小子是一个贬义词，用来描述以黑客自居并沾沾自喜的初学者。”

那些自以为是的工具黑客只不过是一个“脚本小子”，是不是心里觉得不是很舒服了？是不是觉得自己应该提高了？如果是的话，那么就请抛开以前当工具黑客的想法，开始学习编写程序吧！

思想准备

新手可能会问：编写自己的黑客工具是不是很难？是不是要懂编程语言？要懂哪种编程语言呢？笔者的回答是肯定的。抛开用工具的想法，其实是让大家抛开浮躁的想法，认真地学一些真正的技术，哪怕只是一些入门的知识。想做黑客就要有创新、研发的精神，如果只是做一个只会用软件的应用级的计算机使用者，那么必定永远达不到黑客级的水平，因为工具人人都会用，而你只是比别人多知道几个工具而已。抛开浮躁，静下心来从头开始学习基础，为将来的成长做好足够的准备。

攻防的广义性

黑客做得最多的就是“入侵”，这里所说的入侵不是一个狭义上的入侵，因为它不单单针对网络、系统的入侵。这里说的是一个广义上的入侵，“入侵”一词是指“在非授权的情况，试图存取信息、处理信息或破坏系统以使系统不可靠、不可用的故意行为。”由此可以看出，入侵并非单指网络或系统。这里说的“入侵”包括两个方面，一个是针对网络（系统）的入侵，另一个是针对软件的入侵。网络的入侵是通常意义上的入侵，而软件的入侵通常就是人们说的软件破解（包括漏洞挖掘等内容）。无论是侵入别人系统，还是破解某款软件，都是在

非授权的情况下得到相应的权限，比如系统权限或者软件的使用权限。这些“入侵”都是为了寻找系统的安全漏洞，以便更好地完善系统的安全。

本书内容

本书针对“网络入侵”和“软件入侵”两方面来介绍黑客编程，从攻防两个角度来学习黑客编程的知识，通过一系列知识体系完成“黑客编程”的养成计划。

本书会介绍大量的基础知识，这些基础知识看起来与普通的应用程序编程没有什么差别。其实，所谓“黑客编程”（也称为“安全编程”），是指“采用常规的编程技术，编写网络安全、黑客攻防类的程序、工具”。因此，普通的编程技术与黑客编程技术并没有本质的差别，只是开发的侧重点不同。普通的编程注重的是客户的需求，而黑客编程注重的则是攻与防。

黑客编程有其两面性，按照攻防角度可以分为“攻击类入侵编程”和“防范类安全编程”。结合上面提到的“网络”和“软件”两方面来说，常见的“网络攻击”程序有扫描器、嗅探器、后门等；常见的“软件攻击”程序有查壳器、动态调试器、静态分析器、补丁等（这些工具是一些调试工具和逆向分析工具，因为软件破解、漏洞挖掘等会用到这些调试工具，所以称其为“软件攻击”工具）。常见的“网络（系统）防范”程序有“杀毒软件”“防火墙”“主动防御系统”等；常见的“软件防范”程序有“壳”“加密狗”“电子令牌”等。

根据前面提到的攻防两方面的内容，本书会涉及扫描器的开发、后门的开发、应用层抓包器的开发等黑客攻防方面的内容。本书还会讲解关于软件方面的知识，主要涉及 PE 结构、加壳、脱壳、逆向分析等知识。由于技术的两面性，希望读者有一个良好的学习心态，把学到的技术用到安全保护上。

读者能从本书中得到什么

通过本书，读者能学到 Windows 下基于消息的软件开发、基于 Winsock 的网络应用程序的开发、软件逆向分析和调试等方面的编程、调试及安全知识。在学习的过程中，读者应该大量阅读和参考其他相关资料，并且一定要亲自动手进行编程。编程绝对不是靠看书能够学会的！

通过本书的指导，再加上自身实践和练习，读者可以具备 Windows 下基本的应用程序开发、网络程序开发能力，基本的系统底层开发能力。除了提升相关开发能力外，读者还能学到初级的病毒分析、软件保护等相关的安全知识。

如何无障碍阅读本书

阅读本书的读者最好具有 C 和 C++ 编程的基础知识，有其他编程语言基础知识的读者也可以无障碍阅读。对于无编程知识的读者，在阅读本书的同时，只要学习了本书中涉及的相关基础知识，同样可以阅读本书。

本书涉及范围较多，知识面比较杂，但是本书属于入门级读物，专门为新手准备，只要读者具备一定的基础知识，即可顺利进行阅读。在阅读本书的基础上，读者可以接着学习更深层次的知识，希望本书能帮助读者提高自身的能力。

建议读者深入学习操作系统原理、数据结构、编译原理、计算机体系结构等重要的计算机基础知识。

免责

本书中内容主要用于教学，指导新手如何入门、如何学习编程知识，从编程的过程中了解黑客编程的基础知识。请勿使用自己的知识做出有碍公德之事，在准备通过技术手段进行蓄意破坏时，请想想无数“高手”的下场。读者如若作奸犯科自行承担 responsibility，与作者本人和出版社无任何关系，请读者自觉遵守国家法律。

由于作者水平有限，书中难免会有差错，敬请谅解。

编辑联系邮箱：zhangtao@ptpress.com.cn。

资源与支持

本书由异步社区出品，社区（<https://www.epubit.com/>）为您提供相关资源和后续服务。

配套资源

本书配套资源包括书中示例的源代码。

要获得以上配套资源，请在异步社区本书页面中单击 **配套资源**，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

如果您是教师，希望获得教学配套资源，请在社区本书页面中直接联系本书的责任编辑。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，单击“提交勘误”，输入勘误信息，单击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。

The screenshot shows a web form titled "提交勘误" (Submit勘误) with three tabs: "详细信息" (Detailed Information), "写书评" (Write a Review), and "提交勘误" (Submit勘误). The form contains the following fields and elements:

- Page number: 页码:
- Page location (line number): 页内位置 (行数):
- 勘误印次:
- A rich text editor toolbar with icons for bold (B), italic (I), underline (U), strikethrough (ABC), bulleted list (•), numbered list (1-2-3), link (🔗), and unlink (🔗).
- A large text area for entering the勘误 (勘误) details.
- A "字数统计" (Character count) label.
- A "提交" (Submit) button.

扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术 etc。



异步社区



微信服务号

目录

第 1 章 黑客编程入门..... 1

- 1.1 初识 Windows 消息..... 1
 - 1.1.1 对消息的演示测试..... 1
 - 1.1.2 对“消息测试”程序代码的解释..... 3
 - 1.1.3 如何获取窗口的类名称..... 4
- 1.2 Windows 消息机制的处理..... 5
 - 1.2.1 DOS 程序与 Windows 程序执行流程对比..... 5
 - 1.2.2 一个简单的 Windows 应用程序..... 7
- 1.3 模拟鼠标键盘按键的操作..... 12
 - 1.3.1 基于发送消息的模拟..... 13
 - 1.3.2 通过 API 函数模拟鼠标键盘按键的操作..... 16
- 1.4 通过消息实现进程间的通信..... 18
 - 1.4.1 通过自定义消息进行进程通信..... 19
 - 1.4.2 通过 WM_COPYDATA 消息进行进程通信..... 20
- 1.5 VC 相关开发辅助工具..... 24
 - 1.5.1 Error Lookup 工具的使用..... 24
 - 1.5.2 Windows Error Lookup Tool 工具的使用..... 25
 - 1.5.3 VC6 调试工具介绍..... 26
- 1.6 总结..... 32

第 2 章 黑客网络编程..... 34

- 2.1 Winsock 编程基础知识..... 34
 - 2.1.1 网络基础知识..... 34
 - 2.1.2 面向连接协议与非面向连接协议所使用的函数..... 36
 - 2.1.3 Winsock 网络编程知识..... 37
 - 2.1.4 字节顺序..... 42

- 2.2 Winsock 编程实例..... 44
 - 2.2.1 基于 TCP 的通信..... 44
 - 2.2.2 基于 UDP 的通信..... 46
 - 2.2.3 密码暴力猜解剖析..... 48
- 2.3 非阻塞模式开发..... 55
 - 2.3.1 设置 Winsock 的工作模式..... 55
 - 2.3.2 非阻塞模式下简单远程控制的开发..... 56
- 2.4 原始套接字的开发..... 65
 - 2.4.1 Ping 命令的使用..... 65
 - 2.4.2 Ping 命令的构造..... 66
 - 2.4.3 Ping 命令的实现..... 67
- 2.5 总结..... 69

第 3 章 黑客 Windows API 编程..... 70

- 3.1 API 函数、病毒和对病毒的免疫..... 70
 - 3.1.1 文件相关操作 API 函数..... 70
 - 3.1.2 模拟 U 盘病毒..... 74
 - 3.1.3 免疫 AutoRun 病毒工具的编写..... 76
- 3.2 注册表编程..... 78
 - 3.2.1 注册表结构简介..... 79
 - 3.2.2 注册表操作常用 API 函数介绍..... 80
 - 3.2.3 注册表下启动项的管理..... 84
- 3.3 服务相关的编程..... 87
 - 3.3.1 如何查看系统服务..... 88
 - 3.3.2 服务控制管理器的实现..... 89
- 3.4 进程与线程..... 94
 - 3.4.1 进程的创建..... 95
 - 3.4.2 进程的结束..... 100
 - 3.4.3 进程的枚举..... 101
 - 3.4.4 进程的暂停与恢复..... 106
 - 3.4.5 多线程编程基础..... 110
- 3.5 DLL 编程..... 116

3.5.1 编写一个简单的 DLL 程序	116	6.1.1 PE 文件结构全貌	209
3.5.2 远程线程的编程	122	6.1.2 PE 结构各部分简介	210
3.5.3 异步过程调用	132	6.2 详解 PE 文件结构	211
3.6 总结	135	6.2.1 DOS 头部 IMAGE_DOS_HEADER 详解	211
第 4 章 黑客内核驱动开发基础	136	6.2.2 PE 头部 IMAGE_NT_HEADERS 详解	213
4.1 驱动版的“Hello World”	136	6.2.3 文件头部 IMAGE_FILE_HEADER 详解	214
4.1.1 驱动版“Hello World”代码编写	136	6.2.4 可选头部 IMAGE_OPTIONAL_ HEADER 详解	215
4.1.2 驱动程序的编译	139	6.2.5 IMAGE_SECTION_HEADER 详解	219
4.1.3 驱动文件的装载与输出	140	6.3 PE 结构的地址与地址的转换	220
4.1.4 驱动程序装载工具实现	140	6.3.1 与 PE 结构相关的 3 种地址	220
4.2 内核下的文件操作	142	6.3.2 3 种地址的转换	221
4.2.1 内核文件的读写程序	142	6.4 PE 相关编程实例	224
4.2.2 内核下文件读写函数介绍	145	6.4.1 PE 查看器	224
4.3 内核下的注册表操作	151	6.4.2 简单的查壳工具	228
4.3.1 内核下注册表的读写程序	151	6.4.3 地址转换器	231
4.3.2 内核下注册表读写函数的介绍	154	6.4.4 添加节区	233
4.4 总结	156	6.5 破解基础知识及调试 API 函数的 应用	239
第 5 章 黑客逆向基础	157	6.5.1 CrackMe 程序的编写	239
5.1 x86 汇编语言介绍	157	6.5.2 用 OD 破解 CrackMe	241
5.1.1 寄存器	157	6.5.3 文件补丁及内存补丁	245
5.1.2 常用汇编指令集	160	6.6 调试 API 函数的使用	248
5.1.3 寻址方式	165	6.6.1 常见的 3 种产生断点的方法	249
5.2 逆向调试分析工具	166	6.6.2 调试 API 函数及相关结构体介绍	252
5.2.1 OllyDbg 使用介绍	166	6.7 打造一个密码显示器	262
5.2.2 OD 破解实例	170	6.8 KeyMake 工具的使用	265
5.3 逆向反汇编分析工具	174	6.9 总结	267
5.4 C 语言代码逆向基础	182	第 7 章 黑客高手的 HOOK 技术	269
5.4.1 函数的识别	183	7.1 HOOK 技术概述	269
5.4.2 if...else... 结构分析	192	7.2 内联钩子——Inline Hook	270
5.4.3 switch 结构分析	194	7.2.1 Inline Hook 的原理	270
5.4.4 循环结构分析	196	7.2.2 Inline Hook 的实现	271
5.5 逆向分析实例	201	7.2.3 Inline Hook 实例	275
5.5.1 weslen 函数的逆向	201	7.2.4 7 字节的 Inline Hook	279
5.5.2 扫雷游戏辅助工具	203	7.2.5 Inline Hook 的注意事项	280
5.6 总结	208		
第 6 章 加密与解密	209		
6.1 PE 文件结构	209		

7.3 导入地址表钩子——IAT HOOK	283	8.5 加壳与脱壳	392
7.3.1 导入表简介	284	8.5.1 手动加壳	392
7.3.2 导入表的数据结构定义	284	8.5.2 编写简单的加壳工具	394
7.3.3 手动分析导入表	286	8.6 驱动下的进程遍历	395
7.3.4 编程枚举导入地址表	289	8.6.1 配置 VMware 和 WinDbg 进行驱动调试	396
7.3.5 IAT HOOK 介绍	290	8.6.2 EPROCESS 和手动遍历进程	397
7.3.6 IAT HOOK 实例	290	8.6.3 编程实现进程遍历	401
7.4 Windows 钩子函数	293	8.7 HOOK SSDT	402
7.4.1 钩子原理	294	8.7.1 SSDT	402
7.4.2 钩子函数	294	8.7.2 HOOK SSDT	404
7.4.3 钩子实例	296	8.7.3 Inline HOOK SSDT	406
7.5 总结	300	8.8 总结	408
第 8 章 黑客编程实例剖析	301	第 9 章 黑客编程实例	409
8.1 恶意程序编程技术剖析	301	9.1 网络安全简介	409
8.1.1 恶意程序的自启动技术	301	9.1.1 网络与网络安全的简单介绍	409
8.1.2 木马的配置生成与反弹端口技术	309	9.1.2 网络协议基础介绍	410
8.1.3 病毒的感染技术	315	9.2 网络中的破解	418
8.1.4 病毒的自删除技术	319	9.2.1 电子邮箱的破解	418
8.1.5 隐藏 DLL 文件	322	9.2.2 FTP 服务器的破解	424
8.1.6 端口复用技术	329	9.3 Web 安全	427
8.1.7 远程 cmd 通信技术	333	9.4 总结	456
8.2 黑客工具编程技术剖析	338	第 10 章 安卓软件安全初探	457
8.2.1 端口扫描技术	338	10.1 安卓可执行文件格式解析	457
8.2.2 嗅探技术的实现	347	10.1.1 准备一个 Dex 文件	457
8.3 反病毒编程技术	351	10.1.2 DEX 文件格式详解	459
8.3.1 病毒专杀工具的开发	351	10.2 实现 Dex 文件格式解析工具	483
8.3.2 行为监控 HIPS	372	10.2.1 解析工具所需的结构体	483
8.3.3 U 盘防御软件	378	10.2.2 解析 Dex 文件	483
8.3.4 目录监控工具	382	10.3 总结	493
8.4 实现引导区解析工具	385	附录 反病毒公司部分面试题	494
8.4.1 通过 WinHex 手动解析引导区	385	参考文献	496
8.4.2 通过程序解析 MBR	388		
8.4.3 自定义 MBR 的各种结构体	389		
8.4.4 硬盘设备的符号链接	390		
8.4.5 解析 MBR 的程序实现	391		

第1章 黑客编程入门

读者是否曾经用别人开发的工具尝试“入侵”，对自己的系统进行安全检查，是否希望开发出自己的“黑器”？本章将介绍 Windows 操作系统的开发基础，将带领读者进入 Windows 编程的大门。

Windows 是一个庞大而复杂的操作系统，它提供了丰富而强大的功能，不但操作灵活方便，而且有众多的应用软件对其进行支持。Windows 因有众多软件的支持，从而长期雄霸于 PC 系统。之所以有众多软件的支持，是因为 Windows 提供了良好的应用程序开发平台（接口）、完整的开发文档和各种优秀的开发环境。对于一个程序员来说，除了要掌握基本的开发语言以外，还要掌握具体的开发环境和系统平台的相关知识；在掌握编程语言和开发环境等知识后，还要掌握调试技术以及各种调试分析工具。同样，Windows 操作系统提供了良好的调试接口，并且有非常多的调试工具。

本章主要介绍 Windows 的消息机制，Windows 下的开发工具、辅助工具，还有调试工具。本章的目的在于对 Windows 操作系统的消息机制进行回顾，它是 Windows 开发的基础，方便后续章节内容的学习。本章对于 Windows 编程的一些基本概念不会进行过多的介绍。除了对消息机制进行回顾外，本章还要介绍集成在 Visual C++（VC6）中的调试工具和其他一些开发辅助工具。

1.1 初识 Windows 消息

大部分 Windows 应用程序都是基于消息机制的（命令行下的程序并不基于消息机制），熟悉 Windows 操作系统的消息机制是掌握 Windows 操作系统下编程的基础。本节将带领读者认识和熟悉 Windows 的消息机制。

1.1.1 对消息的演示测试

在真正学习和认识消息之前，先来完成一个简单的任务，看看消息能完成什么样的工作。首先写一个简单的程序，通过编写的程序发送消息来关闭记事本的进程、获取窗口的标题和设置窗口的标题。

程序的具体代码如下：

```
void CMsgTestDlg::OnClose()  
{  
    // 在此处添加处理程序代码
```

```
HWND hWnd = ::FindWindow("Notepad", NULL);
if ( hWnd == NULL )
{
    AfxMessageBox("没有找到记事本");
    return ;
}

::SendMessage(hWnd, WM_CLOSE, NULL, NULL);
}

void CMsgTestDlg::OnExec()
{
    // 在此处添加处理程序代码
    WinExec("notepad.exe", SW_SHOW);
}

void CMsgTestDlg::OnEditWnd()
{
    // 在此处添加处理程序代码
    HWND hWnd = ::FindWindow(NULL, "无标题 - 记事本");
    if ( hWnd == NULL )
    {
        AfxMessageBox("没有找到记事本");
        return ;
    }

    char *pCaptionText = "消息测试";
    ::SendMessage(hWnd, WM_SETTEXT, (WPARAM)0, (LPARAM)pCaptionText);
}

void CMsgTestDlg::OnGetWnd()
{
    // 在此处添加处理程序代码
    HWND hWnd = ::FindWindow("Notepad", NULL);
    if ( hWnd == NULL )
    {
        AfxMessageBox("没有找到记事本");
        return ;
    }

    char pCaptionText[MAXBYTE] = { 0 };
    ::SendMessage(hWnd, WM_GETTEXT, (WPARAM)MAXBYTE, (LPARAM)pCaptionText);

    AfxMessageBox(pCaptionText);
}
```

编写的代码中有4个函数：第1个函数 OnClose()是用来关闭记事本程序的；第2个函数 OnExec()是用来打开记事本程序的，主要是测试其他3个函数时可以方便地打开记事本程序；第3个函数 OnEditWnd()是用来修改记事本标题的；第4个函数 OnGetWnd()是用来获取当前记事本标题的。程序的界面如图1-1所示。

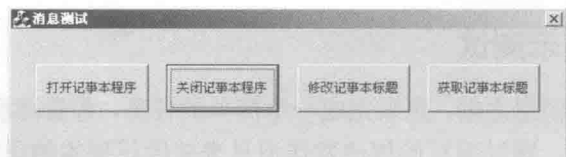


图1-1 消息测试窗口

简单测试一下这个程序。首先单击“打开记事本程序”按钮，出现记事本的窗口（表示记事本程序被打开了）；接着单击“修改记事本标题”按钮，可以发现记事本程序的窗口标题

改变了；再单击“获取记事本标题”按钮，弹出记事本程序窗口标题的一个对话框；最后单击“关闭记事本程序”按钮，记事本程序被关闭。

1.1.2 对“消息测试”程序代码的解释

上面的代码中要学习的 API 函数有两个，分别是 FindWindow()和 SendMessage()。下面看一下它们在 MSDN 中的定义。

FindWindow()函数的定义如下：

```
HWND FindWindow(
    LPCTSTR lpClassName,
    LPCTSTR lpWindowName
);
```

FindWindow()函数的功能是，通过指定的窗口类名（lpClassName）或窗口标题（lpWindowName）查找匹配的窗口并返回最上层的窗口句柄。简单理解就是，通过指定的窗口名（窗口名相对于窗口类来说要直观些，因此往往使用的是窗口名）返回窗口句柄。FindWindow()函数有两个参数，分别是 lpClassName 和 lpWindowName。通过前面的描述，该函数通常使用的是第 2 个参数 lpWindowName，该参数是指定窗口的名称。在例子代码中，为程序指定的窗口名是“无标题—记事本”。“无标题—记事本”是记事本程序打开后的默认窗口标题，当 FindWindow()找到该窗口时，会返回它的窗口句柄。例子代码中也使用了 lpClassName（窗口类名），在窗口的名称会改变的情况下，只能通过窗口类名来获取窗口的句柄了。

当使用 FindWindow()函数获取窗口句柄时，指定窗口名是比较直观和容易的。但是，如果窗口名经常发生变化时，那么就不得不使用窗口类名了。

使用 FindWindow()函数返回的窗口句柄是为了给 SendMessage()函数来使用的。

SendMessage()函数的定义如下：

```
LRESULT SendMessage(
    HWND hWnd,
    UINT Msg,
    WPARAM wParam,
    LPARAM lParam
);
```

该函数的作用是根据指定窗口句柄将消息发送给指定的窗口。该函数有 4 个参数，第 1 个参数 hWnd 是要接收消息的窗口的窗口句柄，第 2 个参数 Msg 是要发送消息的消息类型，第 3 个参数 wParam 和第 4 个参数 lParam 是消息的两个附加参数。第 1 个参数 hWnd 在前面已经介绍过了，该参数通过 FindWindow()函数获取。

在程序的代码中，SendMessage()函数的第 2 个参数分别使用的了 WM_CLOSE 消息、WM_SETTEXT 消息和 WM_GETTEXT 消息。下面来看这 3 个消息的具体含义。

WM_CLOSE: 将 WM_CLOSE 消息发送后，接收到该消息的窗口或应用程序将要关闭。WM_CLOSE 消息没有需要的附加参数，因此 wParam 和 lParam 两个参数都为 NULL。

WM_SETTEXT: 应用程序发送 WM_SETTEXT 消息对窗口的文本进行设置。该消息需要附加参数，wParam 参数未被使用，必须指定为 0 值，lParam 参数是一个指向以 NULL 为结尾的字符串的指针。

WM_GETTEXT: 应用程序发送 WM_GETTEXT 消息，将对应窗口的文本复制到调用者

的缓冲区中。该消息也需要附加参数，wParam 参数指定要复制的字符数数量，lParam 是接收文本的缓冲区。

例子代码在 VC6 下进行编译连接，生成可执行文件后，可以通过按钮的提示进行测试，以便读者感性认识消息的作用。

1.1.3 如何获取窗口的类名称

编写程序调用 FindWindow()函数的时候，通常会使用其第 2 个参数，也就是窗口的标题。但是有些软件的窗口标题会根据不同的情况进行改变，那么程序中就不能在 FindWindow()函数中直接通过窗口的标题来获得窗口的句柄了。而窗口的类名通常是不会变的，因此编程时可以指定窗口类名来调用 FindWindow()函数以便获取窗口句柄。那么，如何能获取到窗口的类名称呢？这就是将要介绍的第 1 个开发辅助工具——Spy++。

Spy++是微软 Visual Studio 中提供的一个非常实用的小工具，它可以显示系统的进程、窗口等之间的关系，可以提供窗口的各种信息，可以对系统指定的窗口进行消息的监控等。它的功能非常多，这里演示如何用它来获取窗口的类名称。

打开“开始”菜单，在 Visual Studio 的菜单路径下找到 Spy++，打开 Spy++窗口，如图 1-2 所示。

选择工具栏中的“Find Window”按钮，如图 1-3 所示。

单击“Find Window”按钮，出现如图 1-4 所示的窗口。

在图 1-4 中，用鼠标左键单击“Finder Tool”后面的图标，然后拖曳到指定的窗口上，会显示出“Handle”（窗口句柄）“Caption”（窗口标题）和“Class”（窗口类名），其中“Class”是编程时要使用的“窗口类”名称。

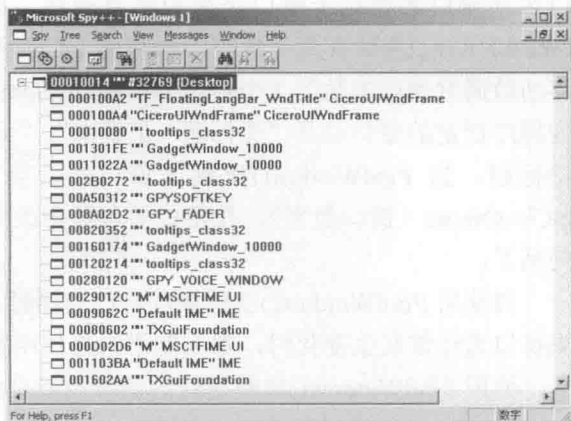


图 1-2 “Microsoft Spy++”窗口



图 1-3 “Find Window”按钮

“Hide Spy++”是一个比较实用的功能，它用来隐藏 Spy++主窗口界面。选中该复选框后，拖曳“Finder Tool”后的图标时，图 1-2 所示为窗口将被隐藏。这个功能的实用之处在于，有些应用软件有反 Spy++的功能，隐藏 Spy++主窗口有助于避免被反 Spy++的软件检测到。为什么隐藏 Spy++的“Find Window”窗口会有反检测的功能，反检测的原理是什么？原理很简单，目标程序也是通过调用 FindWindow()函数来查找 Spy++窗口的，如果有该窗口，就进行一些相应的处理。



注：通过 Spy++找到的窗口句柄是不能在编程中使用的，每次打开窗口时，窗口的句柄都会改变。

将“Finder Tool”后的图标拖曳到记事本的标题处，Spy++的 Find Window 窗口显示的内容如图 1-5 所示。

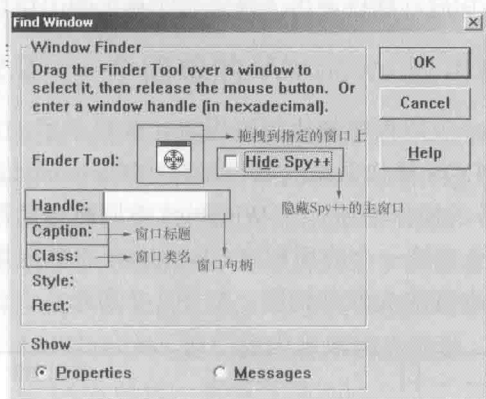


图 1-4 Find Window 窗口

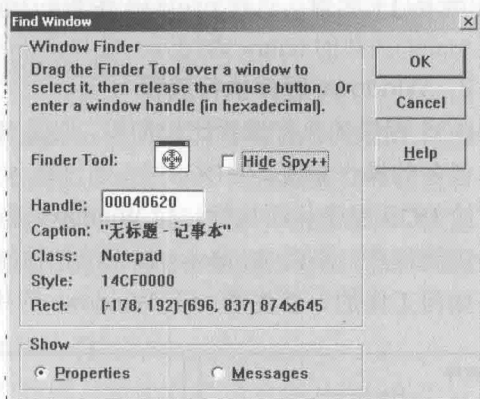


图 1-5 获取到信息的 Find Window 窗口

从图 1-5 中可以得到记事本程序的标题和类名称。当编写程序调用 FindWindow()函数，不能通过程序的标题文本得到窗口的句柄时，可以通过窗口类名称得到窗口的句柄。

1.2 Windows 消息机制的处理

SendMessage()将指定的消息发送给指定的窗口，窗口接收到消息也有相应的行为发生。那么窗口接收到消息后的一系列行为是如何发生的？下面通过熟悉 Windows 的消息机制来理解消息处理背后的秘密。

1.2.1 DOS 程序与 Windows 程序执行流程对比

Windows 下的窗口应用程序都是基于消息机制的，操作系统与应用程序之间、应用程序与应用程序之间，大部分都是通过消息机制进行通信、交互的。要真正掌握 Windows 应用程序内部对消息的处理，必须分析实际的源代码。在编写一个基于消息的 Windows 应用程序前，先来比较 DOS 程序和 Windows 程序在执行时的流程。

1. DOS 程序执行流程

在 DOS 下将编写完的程序进行执行，在执行时有较为清晰的流程。比如用 C 语言编写程序后，程序执行时的大致流程如图 1-6 所示。

在图 1-6 中可以看出，DOS 程序的流程是按照代码的顺序（这里的顺序并不是指程序控制结构中的顺序、分支和循环的意思，而是指程序运行的逻辑有明显的流程和流程依次执行。大致步骤为：DOS 程序从 main()主函数开

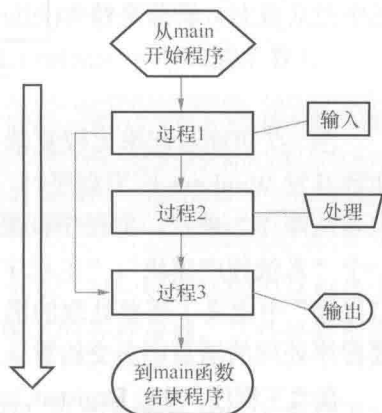


图 1-6 传统 DOS 程序执行流程