

Microsoft

Microsoft

Microsoft

Microsoft®

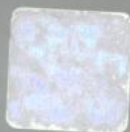
Windows® 95 Windows NT™ 3.5 高级编程技术

[美] Jeffrey Richter 著

郑全战 阿 夏 译

洛 水 张东升 审校

清华大学出版社



Windows® 95
Windows NT® 3.5

高级编程技术

[美] Jeffrey Richter 著
郑全战 阿 夏 译
洛 水 张东升 审校

清华大学出版社

(京)新登字 158 号

Windows 95, Windows NT 3.5 高级编程技术
Advanced Windows

The Developer's Guide to the Win32

API for Windows NT 3.5 and Windows 95

Jeffrey Richter

Copyright © 1995 by Jeffrey Richter.

Original English Language Edition Copyright © 1995 by Jeffrey Richter.

Published by arrangement with the original publisher, Microsoft Press, a division of
Microsoft Corporation, Redmond, Washington, U.S.A.

本书中文版由 Microsoft Press 授权清华大学出版社出版。

中华人民共和国国家版权局著作权合同登记章图字: 01—95—304

版权所有, 翻印必究。

本书贴有 Microsoft Press 激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

Windows 95 Windows NT 3.5 高级编程技术 / (美) Richter, J. 著; 郑全战, 阿夏
译. —北京: 清华大学出版社, 1996.2

ISBN 7-302-02078-7

I. W… II. ① R… ② 郑… ③ 阿… III. 窗口(软件)-编辑程序 IV. TP31

中国版本图书馆 CIP 数据核字(96)第 00973 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: 44 字数: 1039 千字

版 次: 1996 年 5 月第 1 版 1996 年 5 月第 1 次印刷

书 号: ISBN 7-302-02078-7/TP·966

印 数: 0001—5000

定 价: 79.00 元

目 录

鸣谢.....	1
前言.....	3
0.1 我对大家的期望	3
0.2 关于示例应用程序	3
0.2.1 用 C 语言写的程序	4
0.2.2 Message Cracker Macros(消息拆析宏)	4
0.2.3 有关 16 位 Windows 编程的知识	4
0.2.4 在 Windows 95 下运行示例应用程序	4
0.2.5 不相关代码.....	5
0.2.6 相互独立的应用程序.....	5
0.2.7 STRICT 一致性	5
0.2.8 错误检查.....	5
0.2.9 没有错误.....	6
0.2.10 测试平台及环境	6
0.2.11 Unicode	7
0.3 安装示例程序	7
0.3.1 Windows 95	7
0.3.2 Windows NT	8
第 1 章 WIN32 API 及支持它的平台	10
1.1 梦想: Win32 API	10
1.2 Win32s	11
1.3 Windows NT	11
1.4 Windows 95	12
1.5 现实: Win32 API	13
第 2 章 进程	15
2.1 核心对象.....	16
2.2 编写第一个 Win32 应用程序	17
2.2.1 进程的实例句柄	19
2.2.2 进程的先前实例句柄	21
2.2.3 进程的命令行	22
2.2.4 进程的环境变量	23
2.2.5 进程的错误方式	26

2.2.6	进程的当前驱动器和目录	26
2.2.7	进程继承的核心对象	27
2.2.8	系统版本	29
2.3	<i>CreateProcess</i> 函数	30
2.3.1	<i>lpzImageName</i> 和 <i>lpzCommandLine</i>	31
2.3.2	<i>lpzProcess</i> , <i>lpzThread</i> 和 <i>fInheritHandles</i>	32
2.3.3	<i>fdwCreate</i>	33
2.3.4	<i>lpvEnvironment</i>	35
2.3.5	<i>lpzCurDir</i>	35
2.3.6	<i>lpzStartInfo</i>	35
2.3.7	<i>lpzProcInfo</i>	39
2.4	终止进程	40
2.4.1	<i>ExitProcess</i> 函数	40
2.4.2	<i>TerminateProcess</i> 函数	41
2.4.3	进程的结束	41
2.5	子进程	42
2.5.1	运行分离的子进程	43
第3章	线程	44
3.1	何时创建线程	44
3.2	何时不用创建线程	45
3.3	编写第一个线程函数	47
3.3.1	线程的栈	47
3.3.2	线程的 CONTEXT 结构	48
3.3.3	线程的执行时间	48
3.4	<i>CreateThread</i> 函数	50
3.4.1	<i>lpzsa</i>	50
3.4.2	<i>cbStack</i>	51
3.4.3	<i>lpStartAddr</i> 和 <i>lpvThreadParm</i>	51
3.4.4	<i>fdwCreate</i>	52
3.4.5	<i>lpIDThread</i>	53
3.5	终止线程	53
3.5.1	<i>ExitThread</i> 函数	53
3.5.2	<i>TerminateThread</i> 函数	53
3.5.3	线程终止时发生些什么	54
3.6	识别自己的身份	55
3.7	系统如何安排线程	58
3.7.1	如何使用 Win32 API 分配优先级	59
3.7.2	修改进程优先级类	61

3.7.3	设定线程相对优先级	62
3.7.4	挂起及恢复线程	64
3.8	查看系统内部	64
3.9	进程、线程和 C 运行时库	69
3.9.1	要避免使用的 C 运行时函数	72
第 4 章	Win32 存储结构	74
4.1	我所知的 CPU	74
4.2	虚拟地址空间	76
4.2.1	Windows 95 如何划分进程的地址空间	77
4.2.2	Windows NT 如何划分进程的地址空间	78
4.3	地址空间中的区域	80
4.4	在区域内提交物理存储	81
4.5	物理存储	82
4.5.1	不在页面文件中维护的物理存储	83
4.6	保护属性	84
4.6.1	写时拷贝(Copy-On-Write)访问	85
4.6.2	特殊访问保护属性标志	85
4.7	综述	86
4.7.1	区域内部	89
4.7.2	Windows 95 地址空间的差别	93
第 5 章	探索虚存	100
5.1	系统信息	100
5.1.1	系统信息示例应用程序	101
5.2	虚存状态	107
5.2.1	虚存状态示例应用程序	108
5.3	确定地址空间的状态	113
5.3.1	VMQuery 函数	114
5.3.2	虚存分配示例应用程序	123
第 6 章	使用虚存	132
6.1	在地址空间中保留区域	132
6.2	在保留区内提交存储	134
6.3	同时保留区域和提交存储	135
6.4	何时提交物理存储	135
6.5	回收物理存储和释放区域	137
6.5.1	何时回收物理存储	138
6.5.2	虚存分配示例应用程序	138
6.6	改变保护属性	149

6.7	在 RAM 中锁住物理存储	150
6.8	线程的栈	151
6.8.1	Windows 95 下的线程栈	154
6.8.2	C 运行时刻的栈检查函数	155
第 7 章	存储映射文件	158
7.1	存储映射 EXE 和 DLL	158
7.1.1	静态数据不能被一个 EXE 或一个 DLL 文件的多个实例共享	159
7.2	存储映射数据文件	161
7.2.1	方法 1: 一个文件, 一个缓冲区	161
7.2.2	方法 2: 两个文件, 一个缓冲区	162
7.2.3	方法 3: 一个文件, 两个缓冲区	162
7.2.4	方法 4: 一个文件, 零个缓冲区	162
7.3	使用存储映射文件	163
7.3.1	步骤 1: 创建或打开文件核心对象	163
7.3.2	步骤 2: 创建文件映射核心对象	164
7.3.3	步骤 3: 映射文件的数据到进程地址空间	166
7.3.4	步骤 4: 从进程地址空间解除文件数据的映射	169
7.3.5	步骤 5 和步骤 6: 关闭文件映射对象和文件对象	170
7.3.6	使用存储映射文件处理大文件	171
7.4	存储映射文件和一致性	173
7.4.1	文件反转示例应用程序	174
7.4.2	指定存储映射文件的基地址	183
7.5	存储映射文件和 Win32 实现	184
7.5.1	使用存储映射文件在进程间共享数据	186
7.5.2	由页面文件备份的存储映射文件	189
7.5.3	存储映射文件共享示例应用程序	190
7.6	稀疏提交的存储映射文件	197
第 8 章	堆	199
8.1	什么是 Win32 堆?	199
8.1.1	进程的缺省堆	200
8.1.2	创建自己的 Win32 堆	200
8.1.3	创建另外的 Win32 堆	202
8.1.4	消除 Win32 堆	205
8.1.5	在 C++ 中使用堆	205
8.1.6	16 位的 Windows 堆函数	209
第 9 章	线程同步	213
9.1	同步概述	213

9.1.1	最坏的事情之一.....	213
9.2	临界区	214
9.2.1	创建临界区.....	216
9.2.2	使用临界区.....	217
9.2.3	临界区示例应用程序.....	223
9.3	用核心对象同步线程	235
9.3.1	互斥量.....	238
9.3.2	互斥量示例应用程序.....	243
9.3.3	信号量.....	252
9.3.4	超级市场示例应用程序.....	253
9.3.5	事件.....	277
9.3.6	Bucket of Balls 示例应用程序	278
9.3.7	SWMRG 复合同步对象	280
9.3.8	Bucket 示例源代码	285
9.3.9	文档统计示例应用程序.....	304
9.4	线程挂起	313
9.4.1	<i>Sleep</i>	313
9.4.2	异步文件 I/O	313
9.4.3	<i>WaitForInputIdle</i>	313
9.4.4	<i>MsgWaitForMultipleObjects</i>	314
9.4.5	<i>WaitForDebugEvent</i>	315
9.4.6	Interlocked 函数族	315
第 10 章	窗口消息和异步输入	317
10.1	多任务.....	317
10.1.1	抢占式调度.....	218
10.2	线程队列和消息处理.....	320
10.2.1	Win32 消息队列结构.....	320
10.2.2	发送消息给线程的消息队列.....	320
10.2.3	发送消息给窗口.....	322
10.2.4	唤醒线程.....	325
10.3	用消息发送数据.....	330
10.3.1	拷贝数据示例应用程序.....	332
10.4	非序列化输入.....	338
10.4.1	怎样使输入非序列化.....	339
10.5	局部输入状态.....	342
10.5.1	键盘输入和焦点.....	343
10.5.2	鼠标光标管理.....	346
10.5.3	局部输入状态实验示例应用程序.....	347

第 11 章 动态链接库	365
11.1 创建动态链接库	365
11.1.1 映射 DLL 到进程地址空间	367
11.2 DLL 入口和出口函数	371
11.2.1 DLL_PROCESS_ATTACH	373
11.2.2 DLL_PROCESS_DETACH	374
11.2.3 DLL_THREAD_ATTACH	375
11.2.4 DLL_THREAD_DETACH	377
11.2.5 系统如何把对 <i>DllMain</i> 的调用顺序化	378
11.2.6 <i>DllMain</i> 和 C 运行时库	381
11.3 从 DLL 中输出函数和变量	382
11.4 从 DLL 输出函数和变量	384
11.4.1 DLL 的头文件	389
11.5 在同一 EXE 或 DLL 的多个映射之间共享数据	390
11.5.1 EXE 或 DLL 的段	390
11.5.2 ModUse 示例应用程序	393
11.5.3 MultInst 示例应用程序	403
第 12 章 线程局部存储	406
12.1 动态线程局部存储	407
12.1.1 使用动态线程局部存储	408
12.1.2 动态线程局部存储示例应用程序	410
12.2 静态线程局部存储	421
12.2.1 静态局部存储示例应用程序	422
第 13 章 文件系统和文件输入输出	431
13.1 Win32 文件名约定	432
13.2 系统和卷操作	433
13.2.1 获得卷的特定信息	436
13.2.2 磁盘信息查看示例应用程序	441
13.3 目录操作	450
13.3.1 获得当前目录	450
13.3.2 改变当前目录	451
13.3.3 获得系统目录	452
13.3.4 获得窗口目录路径	452
13.3.5 创建和删除目录	452
13.4 拷贝、删除、移动及改名文件	453
13.4.1 拷贝一个文件	453
13.4.2 删除一个文件	453

13.4.3	移动一个文件	453
13.4.4	改名文件	455
13.5	创建、打开和关闭文件	456
13.6	同步读写文件	460
13.6.1	定位文件指针	462
13.6.2	设置文件尾	463
13.6.3	强制缓冲的数据写入磁盘	463
13.6.4	锁定及解锁文件的某区域	463
13.7	异步读写文件	466
13.7.1	同时执行多个异步文件 I/O 操作	472
13.7.2	告警异步文件 I/O	473
13.7.3	告警 I/O 示例应用程序	475
13.8	操作文件属性	488
13.8.1	文件标志	488
13.8.2	文件大小	489
13.8.3	文件时间戳	489
13.9	搜寻文件	492
13.9.1	目录漫游示例应用程序	495
13.10	文件系统变化通知	504
13.10.1	文件变化示例应用程序	507
第 14 章	结构化异常处理	521
14.1	终止处理程序	522
14.1.1	通过例子理解终止处理程序	522
14.1.2	关于 <i>finally</i> 块的几点说明	533
14.1.3	SEH 终止示例应用程序	535
14.2	异常过滤程序和异常处理程序	545
14.2.1	通过例子理解异常过滤程序和异常处理程序	546
14.2.2	EXCEPTION-EXECUTE-HANDLER	547
14.2.3	EXCEPTION-CONTINUE-EXECUTION	548
14.2.4	EXCEPTION-CONTINUE-SEARCH	550
14.2.5	全局展开	553
14.2.6	停止全局展开	555
14.2.7	关于异常过滤程序的进一步讨论	556
14.2.8	<i>GetExceptionInformation</i>	560
14.2.9	SEH 异常情况示例应用程序	565
14.2.10	SEH SUM 示例应用程序	575
14.3	软件异常情况	582
14.3.1	SEH 软件异常情况示例应用程序	584

14.3.2	未处理的异常情况	595
14.3.3	没有调试器附属时的未处理异常情况	596
14.3.4	关闭异常情况消息框	598
14.3.5	自己调用 <i>UnhandledExceptionFilter</i>	600
14.3.6	未处理的核心态异常情况	600
第 15 章	UNICODE	602
15.1	字符集	602
15.1.1	单字节与双字节字符集	602
15.1.2	Unicode: 宽字节字符集	603
15.2	为什么要使用 Unicode	604
15.3	如何编写 Unicode 源代码	604
15.3.1	Windows NT 和 Unicode	605
15.3.2	Windows 95 和 Unicode	605
15.3.3	C 运行时库中对 Unicode 的支持	605
15.3.4	Win32 定义的 Unicode 数据类型	610
15.3.5	Win32 中的 Unicode 和 ANSI 函数	611
15.4	使你的应用程序区分 ANSI 和 UNICODE	613
15.4.1	Win32 中的字符串函数	613
15.4.2	资源	615
15.4.3	文本文件	616
15.4.4	在 Unicode 与 ANSI 之间的字符串转换	616
15.4.5	Windows NT: 窗口类和过程	620
第 16 章	闯过进程的边界墙	622
16.1	为什么需要打破进程边界墙: 一个例子	622
16.2	使用 Registry 来注入一个 DLL	624
16.3	使用窗口钩子来注入一个 DLL	626
16.3.1	Program Manager 恢复示例应用程序	627
16.4	使用远程线程来注入一个 DLL	642
16.4.1	如何装入一个 DLL	642
16.5	影响其它进程的 Win32 函数	642
16.5.1	<i>CreateRemoteThread</i>	643
16.5.2	<i>GetThreadContext</i> 和 <i>SetThreadContext</i>	644
16.5.3	<i>VirtualQueryEx</i> 和 <i>VirtualProtectEx</i>	648
16.5.4	<i>ReadProcessMemory</i> 和 <i>WriteProcessMemory</i>	648
16.6	创建一个函数来把 DLL 注入到任意进程的地址空间内	649
16.6.1	版本 0: 为什么显而易见的方法就是不起作用	649
16.6.2	版本 1: 手工编码的机器语言	650

16.6.3	版本 2: <i>AllocProcessMemory</i> 和 <i>CreateRemoteThread</i>	652
16.6.4	ProcMem 实用函数	656
16.6.5	<i>InjectLib</i> 函数	659
16.7	测试 <i>InjectLib</i> 函数	669
16.7.1	注入库测试示例应用程序	669
16.7.2	Image Walk 动态链接库	671
16.8	小结	674
附录 A	消息拆析器	676
A.1	消息拆析器	677
A.2	子控制宏	680
A.3	API 宏	681
附录 B	BUILD 环境	682
B.1	ADVWIN32.H 头文件	682
B.1.1	Warning Level 4(四级警告)	682
B.1.2	STRICT 宏	682
B.1.3	Unicode	683
B.1.4	ARRAY_SIZE 宏	683
B.1.5	BEGINTHREADEX 宏	683
B.1.6	链接器指令	684
B.2	不能在源文件中设置的项目 Settings(设置)	687

鸣 谢

虽然这部书的封面上只出现了我一个人的名字,但许多朋友都以这样或那样的方式为本书的编写出了力。如果没有以下这些人的帮助和支持,仅凭我个人的努力和专业知识,我是不能完成这部书的编写工作的。感谢大家!

在整个编书过程中, Susan “Q” Ramee 给了我爱和支持。Sue 还校对了各章节,并帮我想出了示例程序中的一些主意。当然,我还要感谢她的两只猫咪 Nalt 和 Cato。当我在深夜无法入睡而决定写作时, Nalt 和 Cato 经常陪伴着我。当我打字时,她们经常在键盘上跳来跳去而且扔掉我的笔记。我敢保证,读者在本书中发现的任何打字错误都是 Nalt 和 Cato 所为。

Jim Harkins 是我的一位最好的朋友。他对本书的直接贡献可以在 Directory Walker, Alertable File I/O 和 File Change 示例程序中找到。Jim 还帮我考虑了线程同步和 *InjectLib* 的与 CPU 无关的版本中的许多问题。

Scott Ludwig 和 Valerie Horvath 是我最亲密的朋友。Scott 是 Microsoft 公司里带头开发第一版 Windows 的开发人员。在我写本书第一版时,他总是极其耐心地回答我所有的问题。经过我们的讨论, Scott 赢得了我最大的尊敬和爱戴。

Lucy Gooding 为我的生活带来许多乐趣。她应该为容忍我的时间表而获得一枚勋章。

Jeff Cooperstein 总是知道如何与一个系统捣乱并让它做设计目标之外的事情。Jeff 发现了几种绕过 Windows NT 安全性机制的办法(这些全都在 Windows NT 3.5 中完善了)并正要开发他自己的 Virus Developer's Kit(病毒开发工具箱, VDK)。

Jonathan Locke 帮我校读了许多章节,并建议我对许多正文做了调整。

Lou Perazzoli, Steve Wood 及 Marc Lucovsky 是 Windows NT 开发组的成员,他们审阅了部分章节并回答了有关线程及存储管理的许多问题。

Windows 95 开发小组的 Brain Smith, Jon Thomason 和 Michael Toutonghi 回答了关于 Windows 95 存储和线程管理的一些问题。

Asmus Freytag(aka Dr. Unicode)校读了 Unicode 那章。

Windows NT NTVDM 开发小组的 Dave Hart 回答我关于在 Windows NT 下运行 16 位 MS-DOS 和 Windows 应用程序的许多问题。

Chuck Mitchell, Steve Salisbury 和 Jonathan Mark 这几位是 Visual C++ for Windows NT 开发组的成员,他们回答了我有关结构化异常处理、局部线程存储、C 运行时库及链接方面的问题。

我还要感谢 Mark Durley 和 Cezary Marcjan,他们找出了第一版中的许多错并给我提供了关于 Win32 程序设计的许多主意。

我还要感谢 Visual C++ 开发小组的其他几位成员。他们是: Byron Dazey, Eric

Lang, Dan Spalding, Matthew Tebbs, Bruce Johnson, Jon Jorstad, Dave Henderson, 和 T. K. Brackman。

Bernie McIlroy 帮助我在 DEC Alpha 机上测试了这些应用程序。

Mark Cliggett, Cameron Ferroni, Eric Fogelin, Randy Kath 和 Steve Sinofsky 在我不熟悉的问题上给了我帮助。

Rebecca Gleason 是 Microsoft Press 负责本书第二版的编辑。Rebecca 在所有方面都很精通,并且总能回答我所有的问题。

Jim Fuchs 是 Microsoft Press 负责本书第二版的技术编辑。Jim 为校对我的源代码和资源文件付出了很大的努力。

Nancy Siadek 应该因其为这部书所作出的努力和贡献而获得奖励。她是本书第一版的编辑。我敢肯定她都不知道她为这部书付出了多少心血。Nancy 在我和她相处写书的这短暂的时间内教我的比我这一生学的都要多。

Jeff Carey 是本书第一版的技术编辑。他代我回答了 Nancy 的许多问题,从而使我能够有空重写某些部分。

我也要感谢 Microsoft Press 的其他成员。虽然他们中的许多人我从未见过,但我很欣赏并感谢他们的努力。对第二版:Shawn Peck, John Sugg, Jim Kramer, Michael Victor, Kim Eggleston, David Holter, Penelope West, Richard Carter, Elisabeth Thebaud, Peggy Herman 和 Barbara Remmele。对第一版:Erin O'connor, Laura Sackerman, Deborah Long, Peggy Herman, Lisa Iversen 和 Barb Runyan。

还要感谢:

Borland International 的 Dan Horn,因为有一些章节中有他的建议和他编写的材料。

Jim Lane, Tom Van Back 和 Rich Peterson,他们在 DEC Alpha 编译器方面给了我帮助。

Microsoft Press 的 Dean Holmes,他接受了我的书稿并容忍了我编写第一版时因为买房子而造成的延误。

Gretchen Bilson 和 *Microsoft System Journal* 的每一位成员都鼓励我不断写作。

Charles Petzold 向我介绍了 Microsoft Press 和酸辣汤。

Carlos Richardson 帮我安装了 TJ-NET(我的家庭网络),使之能在我的新家开始运行。

Donna Murray,她在这几年中给了我爱护、支持和友谊。

我的兄弟 Ron 一直在帮我找一本 Patrick Moraz 的“Salamander”(火怪)。虽然他从未找到它,但我知道他尽了力。

还有我的母亲和父亲,我要感谢他们许多年来给我的爱和支持。

前言

我很高兴能够编写这部书。与其它事情相比我更喜欢处于科技的前沿并能学到新东西。Windows 95 和 Windows NT 显然就处于技术的前沿,这里有很多新的知识可学。如果你已经是一位 16 位 Windows 程序员,当你仅学到一点修正你已有程序的简单技术之后,你会发现你已经能编写 Win32 应用程序了。但这些被修正的程序不会利用 Win32 提供的崭新而又强大的功能。

当你开始使用 Win32 时,可以逐渐在你的应用程序中越来越多地利用这些特性。Win32 的许多特性使编程变得极为简单。而且,正如我很快发现的那样当我修正我以前的程序时,我可以删去我原有程序中的很大一部分代码,并可通过调用 Win32 提供的特性来代替它们。

利用这些崭新的功能是如此的方便,以致于我现在只进行 Win32 编程了,我还经常在公司和会议上解释开发人员如何能够有效地编写 Win32 应用程序。

这部书是我在 Windows 95 和 Windows NT 上工作经验的结晶。自从本书第一版以来,我已经学会了许多东西;在这一版中,我几乎重写了所有的章节,并且扩展了深度以覆盖 Win32 的高级特性。我也重新组织了材料并用我自以为更清楚的方式加以表达。

我确信 Win32 API 将会成为一种标准的 API——不管是在大型机和小型机上(Windows NT)还是在微机上(Windows 95 和 Windows NT)。本书将帮助你准备好在这一注定要成为工业标准的环境上开发应用。

0.1 我对大家的期望

这部书是针对那些已具有编写有关 16 位 Windows 程序经验的 Windows 开发者而编写的。不过读者并不需要有关 16 位 Windows 深入的知识,而只要具备 Windows 编程的基本知识就够了,这些基本知识包括窗口过程、窗口消息、对话框和存储管理。这本书的内容涵盖了运行在 Windows NT 操作系统下的已被引入 Win32 API 中的新功能。本书并不是为了讲解 Windows 编程的入门知识。这本书也包含了当你将 16 位 Windows 应用程序移植到 Win32 API 时所需解答的问题。

0.2 关于示例应用程序

编写示例应用程序的目的是以实际程序演示如何利用 Windows 的先进功能。你永远也不能通过阅读足够的书籍来代替通过编写应用程序而获得的知识 and 经验。我在学习 Windows NT 时也有同样的体会。本书所出现的许多示例应用程序是我在努力理解 Windows NT 如何工作的过程中编写的。

0.2.1 用 C 语言写的程序

当在决定用哪一种语言编写示例应用程序时,我在选择 C 还是 C++ 之间犹豫不决。对大项目,我总是使用 C++,但实际上大多数 Windows 的编程人员目前还未使用 C++,而且我也不想因选择了 C++ 而疏远了我最大部分的潜在读者。

0.2.2 Message Cracker Macros(消息拆析宏)

如果读者不利用 C++ 和 Windows 类库(类似 Microsoft Foundation Classes)编写程序,那么我建议读者利用定义在 WINDOWSX.H 头文件内的消息拆析宏,这些宏定义使你的程序更便于编写、阅读和保存。我认为消息拆析宏的功能非常强大,因此我在本书附录中描述了消息拆析宏存在的必要性及如何有效地利用它们。

0.2.3 有关 16 位 Windows 编程的知识

虽然很需要有 16 位 Windows 编程的经验,但是书中所有的程序都不依赖于有关 16 位 Windows 编程的深入知识。示例应用程序是在假设读者已熟悉对话框及其子控制的建立和管理的前提下编写的,读者仅需具有极少的有关 GDI 及 Kernel 功能的知识即可。

在本书中讨论许多题目时,我确实对 16 位 Windows 和 Windows NT 进行了比较。如果读者已经理解了 16 位 Windows 是如何行事的,那就不难理解 Win32 中的变化。

0.2.4 在 Windows 95 下运行示例应用程序

Windows 95 的目标是在只有 4MB 的 RAM 的机器上运行。为此,Microsoft 在建造 Windows 95 时不得不敲掉一些边边角角。对于软件开发人员来讲,这就意味着某些 Win32 函数在 Windows 95 上没有完全实现。就我的示例程序而言,这就意味着某些应用程序在 Windows NT 下会有更多的功能。

此外,在我写这本书的时候,Windows 95 还在开发中,所有示例程序都是在 Windows 95 build 275 上测试的,但却没法在 Windows 95 的最终版本上测试。在我完成此书时,下列程序不能在 Windows 95 build 275 上运行:ALERTIO.EXE(第 13 章)和 TINJLIB.EXE(第 16 章)——原因我在介绍它们的相应章节里会有说明。

要想获得 Windows 95 的最新改动,我建议你周期性地访问 WIN-NEWS 论坛。你可以在下列地方找到它们:

在 CompuServe 上:	GO WINNEWS
在 Internet 上:	ftp://ftp.microsoft.com/peropsys/Win-News http://www.microsoft.com
在 AOL 上:	关键字 WINNEWS
在 Prodigy 上	jumpword WINNEWS
在 Genie 上	Windows RTC 上的 WINNEWS 文件域

你也可以加入(subscribe to)Microsoft 的电子新闻(newsletter)WinNews。要想加入,就给 enews@microsoft.nwnet.com 发送电子邮件,邮件正文中应包括 SUBSCRIBE

WINNEWS 这两个词。

0.2.5 不相关代码

我曾想从示例应用程序中删去那些与我所要阐明的技术不直接相关的代码。不幸的是,无论编写什么样的 Windows 程序这都是不可能的。例如,大多数 Windows 编程书重复了本书中每一个例程中出现的注册 Windows 类的代码。我已尽了最大努力来减少这一类不相关的代码。

我减少不相关代码的方法之一是使用对 Windows 编程者来说不总是显而易见的那些技术。例如,大多数示例应用程序的用户界面是对话框。事实上,大多数示例应用程序在 *WinMain* 中有一行简单地调用 *DialogBox*(对话框)功能的代码。其结果是,没有一个示例应用程序初始化 *WNDCLASS* 结构或者是调用 *RegisterClass* 函数。另外,只有第 13 章中的 *FileChng* 应用程序中有一个消息循环。

0.2.6 相互独立的应用程序

我已尽力保持应用程序互相独立。例如,存储映射文件这一章是包含存储映射文件应用程序的唯一的一章。因为我在设计程序时已使它们相互独立,因此读者阅读前后各章时不必互相对照查找。

读者偶尔会发现在一应用程序中使用了前面几章已出现的技术及信息。比如,第 14 章“结构化异常处理”中的 *SEHExcept* 应用程序,它演示了如何操作虚存。因为 SEH 是控制虚存的一个非常有用的工具,所以我决定将这两个专题写在一个应用程序中。为了完全理解这一应用程序,读者在研究 *SEHExcept* 应用程序之前应阅读第 3 章。

不过,有一个示例程序,似乎有点什么都沾点边儿:*TInjLib*(在第 16 章)。要想完全理解这一应用程序,你必须对核心对象、虚存、进程、线程、线程同步、动态链接库、结构化异常处理以及 Unicode 都有很好的理解。我敢说,理解了 *TInjLib* 的这些性质之后,你就可以在面试时说你自己懂 Win32 编程了。

0.2.7 STRICT 一致性

所有的示例应用程序都在定义了 *STRICT* 标识符的情况下进行了编译,这一标志能够发现常见编码错误。例如,传递给函数的不正确的句柄类型是在编译期间,而不是在运行期间被发现。有关利用 *STRICT* 标识符的更多的信息,参见包含在 Win32 SDK 中的编程技术这一部分。

0.2.8 错误检查

错误检查在任何软件项目中都占很大部分工作量。不幸的是,适当的错误检查会使软件工程的规模及复杂程度呈指数上升。为了使示例应用程序更简单明了,我未增加太多的错误检查代码。如果读者利用我的一些代码片段并将它们组合到自己设计的程序中,最好能详细地检查这些代码并增加适当的错误检查项。