

Designed for



Microsoft®
Windows NT®
Windows 98



CD-ROM
Included



Microsoft® 程序设计系列

完全修订
和更新!

Windows® 程序设计

(第5版) 上册



Win32 API
的权威指导

北京大学出版社

Microsoft Press

Windows 程序设计

(第 5 版)

上册

[美] Charles Petzold 著

北京博彦科技发展有限公司 译

北京大学出版社

· 北 京 ·

著作权合同登记 图字：01-1999-1925 号

Programming Windows (Fifth Edition)

Charles Petzold

本书版权为 Charles Petzold 所有,1999。(Copyright © 1999 by Charles Petzold. All rights reserved.)

本书中文版由美国 Microsoft 出版社授权北京大学出版社独家出版,1999。

本书封面贴有北京大学出版社的激光防伪标签,无标签者不得销售。

版权所有,翻印必究。

图书在版编目(CIP)数据

Windows 程序设计:(第5版)/[美] Charles Petzold 著;北京博彦科技发展有限公司译.一北京:北京大学出版社,1999.11

ISBN 7-301-04187-X

I. W… II. ①C… ②北… III. 窗口软件,Windows 程序设计 IV. TP316

中国版本图书馆 CIP 数据核字(1999)第 31642 号

书 名: Windows 程序设计(第5版)

著作责任者:[美] Charles Petzold 著 北京博彦科技发展有限公司 译

责任编辑:邱淑清 沈承凤 柯 纪 张 华

标准书号:ISBN 7-301-04187-X/TP·463

出 版 者:北京大学出版社

地 址:北京市海淀区中关村北京大学校内 100871

网 址: <http://cbs.pku.edu.cn>

电 话: 出版部 62752015 发行部 62754140 编辑室 62757065

电子信箱: zpup@pup.pku.edu.cn

排 印 者:北京大学印刷厂

发 行 者:北京大学出版社

经 销 者:新华书店

787 毫米×1092 毫米 16 开本 86 印张 1922 千字

1999 年 11 月第 1 版 2003 年 6 月第 6 次印刷

印 数:23001~26000 册

定 价:160.00 元(上下两册,含光盘)

对《Windows 程序设计》的评论

“第一流的 Windows 程序设计的书籍当然是 Charles Petzold 的《Windows 程序设计》。如果您还没有一本,您就需要得到一本。”

Windows Tech Journal

“重要的程序员指南……现在是并且继续是 Charles Petzold 的《Windows 程序设计》。如果您想要编写 Windows 应用程序并且还没有读过这本书,您就需要好好地读一读它。”

IEEE Micro

“我认为仅有三本书对于 Windows 程序设计是至关重要的:第一本,当然是由 Charles Petzold 著的经典的《Windows 程序设计》。”

INFOWORLD

“当您收藏最好的 Windows 书籍时,应该确认包括了经典的——Petzold 的最畅销的《Windows 程序设计》……”

PC Techniques

“它一直是经典的 Windows 程序设计指南。”

Programmer's Journal

“如果您想要编写 Windows 程序,请买这本书。随着时间的流逝,它会给您带来比书的价钱多得多的利益。”

Computer Language

“包含的范围很广而遗漏的很少,这本书对于认真对待 Windows 的任何人都是必需的。”

Byte Magazine

作者的话

关于此书的更新信息请访问 Web 站点: www.cpetzold.com, 其中包括了可能存在的错误报告和新的代码清单。如果您对本书有什么问题, 可以发电子邮件给 charles@cpetzold.com。虽然我会尽量回答您问的任何简单的问题, 但是我不能做任何保证。

我要感谢微软出版社中使本书能够出版而做出大量工作的每一个人。我认为《Windows 程序设计》的“十周年版本”^{*}是最好的一个版本。Microsoft 的许多人(包括 Microsoft Windows 的早期开发者)也在我写前几版时, 帮助我解决了很多难题, 这些杰出的人们都已列在那些版本中。

也感谢我的家人和朋友, 尤其是那些新朋友, 是这些朋友的支持使这本书得以问世。我真诚地将此书献给你们。

Charles Petzold
1998 年 10 月 5 日

* 即指这本书:《Windows 程序设计》(第 5 版)。——编者注

作者简介

Charles Petzold 从 1984 年开始编写个人计算机程序,从 1985 年开始编写 Microsoft Windows 程序。他在《Microsoft Systems Journal》1986 年 12 月号上发表了第一篇关于 Windows 程序设计的杂志论文。从 1986 年到 1995 年,他为《PC Magazine》撰写“Environments”专栏,给读者介绍了 Windows 和 OS/2 程序设计的许多方面。

《Windows 程序设计》由微软出版社在 1988 年首次出版,后来被认为是这方面最好的导论性读本。在 1994 年 5 月,Petzold 作为仅有的七个人之一(并且是唯一的作家)被《Windows Magazine》和 Microsoft 公司授予 Windows Pioneer 奖,以表彰他对 Microsoft Windows 的成功做出的贡献。

在 1999 年秋天,微软出版社将出版 Charles Petzold 面向普通读者的第一本书。暂时命名为《Code: The Hidden Language of Computer Hardware and Software》,这本书专门介绍数字信息的本质以及计算机处理数字信息的方式。

目录

对《Windows 程序设计》的评论	1
作者的话	3
作者简介	5

上 册

第一部分 基础知识

第一章 开始	1
1.1 Windows 环境	2
1.1.1 Windows 简史	2
1.1.2 Windows 方面	3
1.1.3 动态链接	5
1.2 Windows 编程选项	6
1.2.1 API 和内存模式	6
1.2.2 语言选项	7
1.2.3 编程环境	8
1.2.4 API 文档	9
1.3 编写一个 Windows 程序	9
1.3.1 字符模式(Character-Mode)模型	9
1.3.2 Windows 等价程序	10
1.3.3 头文件	11
1.3.4 程序入口点	12
1.3.5 MessageBox 函数	13
1.3.6 编译、链接和运行	14

第二章 Unicode 简介	17
2.1 字符集简史	17
2.1.1 美国标准	18
2.1.2 国际方面	19
2.1.3 扩展 ASCII	20
2.1.4 双字节字符集	21
2.1.5 Unicode 解决方案	22
2.2 宽字符和 C	23
2.2.1 char 数据类型.....	23
2.2.2 宽字符	24
2.2.3 宽字符库函数	25
2.2.4 维护单一源代码	26
2.3 宽字符和 Windows	28
2.3.1 Windows 头文件类型	28
2.3.2 Windows 函数调用	30
2.3.3 Windows 的字符串函数	31
2.3.4 在 Windows 中使用 printf	31
2.3.5 格式化消息框	33
2.3.6 本书与国际化	35
第三章 窗口和消息	37
3.1 自己的窗口	37
3.1.1 总体结构	37
3.1.2 HELLOWIN 程序.....	39
3.1.3 总体考虑	42
3.1.4 注册窗口类	46
3.1.5 创建窗口	52
3.1.6 显示窗口	53
3.1.7 消息循环	54
3.1.8 窗口过程	56
3.1.9 处理消息	56
3.1.10 播放声音文件	57
3.1.11 WM_PAINT 消息	58
3.1.12 WM_DESTROY 消息.....	60
3.2 Windows 编程的难点	60
3.2.1 别调用我,我会调用您	60

3.2.2 进队消息与不进队消息	62
3.2.3 行动迅速	63
第四章 输出文本	65
4.1 绘制和刷新	66
4.1.1 WM_PAINT 消息	66
4.1.2 有效矩形和无效矩形	67
4.2 GDI 简介	68
4.2.1 设备描述表	68
4.2.2 获取设备描述表句柄：方法一	69
4.2.3 绘图信息结构	70
4.2.4 获取设备描述表句柄：方法二	72
4.2.5 TextOut：细节	72
4.2.6 系统字体	74
4.2.7 字符大小	74
4.2.8 文本尺寸：细节	75
4.2.9 格式化文本	77
4.2.10 综合使用	78
4.2.11 SYSMETS1.C 窗口过程	85
4.2.12 空间不够	86
4.2.13 客户区的大小	87
4.3 滚动条	88
4.3.1 滚动条的范围和位置	89
4.3.2 滚动条消息	91
4.3.3 在 SYSMETS 中添加滚动功能	93
4.3.4 绘图程序的组织	97
4.4 建立更好的滚动	98
4.4.1 滚动条信息函数	98
4.4.2 滚动范围	100
4.4.3 新 SYSMETS	101
4.4.4 不用鼠标怎么办	108
第五章 图形基础	109
5.1 GDI 的结构	109
5.1.1 GDI 原理	109
5.1.2 GDI 函数调用	111
5.1.3 GDI 图元	112
5.1.4 其他方面	112

5.2 设备描述表	113
5.2.1 获取设备描述表句柄	113
5.2.2 获取设备描述表信息	115
5.2.3 DEVCAPSI 程序	116
5.2.4 设备的大小	119
5.2.5 关于色彩	124
5.2.6 设备描述表属性	126
5.2.7 保存设备描述表	127
5.3 画点和线	128
5.3.1 写像素	129
5.3.2 直线	129
5.3.3 边界框函数	134
5.3.4 贝塞尔样条	140
5.3.5 使用现有画笔(Stock Pens)	145
5.3.6 画笔的创建、选择和删除	146
5.3.7 填充空隙	150
5.3.8 绘图方式	150
5.4 绘制填充区域	152
5.4.1 Polygon 函数和多边形填充方式	153
5.4.2 用画刷填充内部	157
5.5 GDI 映射方式	159
5.5.1 设备坐标和逻辑坐标	161
5.5.2 设备坐标系	161
5.5.3 视口和窗口	162
5.5.4 处理 MM_TEXT	164
5.5.5 “度量”映射方式	166
5.5.6 “自作主张的”映射方式	169
5.5.7 WHATSIZE 程序	175
5.6 矩形、区域和剪裁	178
5.6.1 矩形函数	178
5.6.2 随机矩形	180
5.6.3 创建和绘制区域	184
5.6.4 矩形与区域的剪裁	186
5.6.5 CLOVER 程序	187
第六章 键盘	193
6.1 键盘基础	193
6.1.1 忽略键盘	193

6.1.2 谁获得了焦点	194
6.1.3 队列和同步	195
6.1.4 击键和字符	195
6.2 击键消息	196
6.2.1 系统击键与非系统击键	196
6.2.2 虚拟键码	197
6.2.3 lParam 信息	201
6.2.4 换档状态	202
6.2.5 使用击键消息	203
6.2.6 为键盘增强 SYSMETS	204
6.3 字符消息	212
6.3.1 四类字符消息	212
6.3.2 消息顺序	213
6.3.3 处理控制字符	215
6.3.4 死字符消息	216
6.4 键盘消息和字符集	216
6.4.1 KEYVIEW1 程序	217
6.4.2 外语键盘问题	222
6.4.3 字符集和字体	224
6.4.4 Unicode 怎么样?	234
6.4.5 TrueType 和大字体	235
6.5 插入符(不是光标)	242
6.5.1 插入符函数	242
6.5.2 TYPED 程序	243
第七章 鼠标	251
7.1 鼠标基础	251
7.1.1 一些简单的定义	252
7.2 客户区鼠标消息	253
7.2.1 简单的鼠标处理: 一个例子	254
7.2.2 处理 Shift 键	258
7.2.3 双击鼠标键	259
7.3 非客户区鼠标消息	261
7.3.1 命中测试消息	262
7.3.2 从消息产生消息	263
7.4 程序中的命中测试	263
7.4.1 一个假想的例子	264
7.4.2 示例程序	264

7.4.3	使用键盘仿真鼠标	268
7.4.4	在 CHECKER 中添加键盘接口	269
7.4.5	将子窗口用于命中测试	273
7.4.6	CHECKER 中的子窗口	274
7.4.7	子窗口和键盘	279
7.5	捕获鼠标	284
7.5.1	设计矩形	284
7.5.2	捕获的解决方案	289
7.5.3	BLOKOUT2 程序	289
7.6	鼠标轮	293
7.6.1	后会有期	301
第八章	计时器	303
8.1	计时器基础	304
8.1.1	系统和计时器	304
8.1.2	计时器消息不是异步的	305
8.2	计时器的使用; 三种方法	305
8.2.1	方法一	305
8.2.2	方法二	309
8.2.3	方法三	312
8.3	计时器用于时钟	313
8.3.1	构造数字时钟	313
8.3.2	获取当前时间	318
8.3.3	显示数字和冒号	319
8.3.4	国际化	319
8.3.5	构造模拟时钟	320
8.4	为状态报告使用计时器	326
第九章	子窗口控制	331
9.1	按钮类	333
9.1.1	创建子窗口	337
9.1.2	子窗口向父窗口发送消息	338
9.1.3	父窗口向子窗口发送消息	339
9.1.4	下压按钮	340
9.1.5	复选框	341
9.1.6	单选按钮	342
9.1.7	分组框	342
9.1.8	更改按钮文本	343

9.1.9 可见的和启用的按钮	343
9.1.10 按钮和输入焦点	344
9.2 控制与颜色	345
9.2.1 系统颜色	345
9.2.2 按钮颜色	347
9.2.3 WM_CTLCOLOREBTN 消息	348
9.2.4 所有者绘制按钮	348
9.3 静态类	355
9.4 滚动条类	356
9.4.1 COLORS1 程序	357
9.4.2 自动键盘接口	364
9.4.3 窗口子类化	365
9.4.4 给背景着色	366
9.4.5 给滚动条和静态文本着色	367
9.5 编辑类	368
9.5.1 编辑类风格	370
9.5.2 编辑控制通知	371
9.5.3 使用编辑控制	372
9.5.4 发送给编辑控制的消息	372
9.6 列表框类	373
9.6.1 列表框风格	374
9.6.2 将串放入列表框	374
9.6.3 选择和获取项	376
9.6.4 接收来自列表框的消息	377
9.6.5 一个简单的列表框应用程序	378
9.6.6 文件列表	382
9.6.7 Windows 的 head 程序	383
第十章 菜单及其他资源	389
10.1 图标、光标、字符串和定制资源	390
10.1.1 将图标添加到程序	390
10.1.2 获取图标句柄	395
10.1.3 在程序中使用图标	398
10.1.4 使用自定义光标	399
10.1.5 字符串资源	399
10.1.6 定制资源	401
10.2 菜单	409
10.2.1 菜单概念	410

10.2.2	菜单结构	410
10.2.3	定义菜单	411
10.2.4	在程序中引用菜单	411
10.2.5	菜单和消息	412
10.2.6	示例程序	414
10.2.7	菜单设计规范	421
10.2.8	定义菜单的复杂方法	421
10.2.9	浮动弹出式菜单	423
10.2.10	使用系统菜单	428
10.2.11	更改菜单	431
10.2.12	其他菜单命令	432
10.2.13	创建菜单的非正规方法	433
10.3	键盘加速键	438
10.3.1	为什么要使用加速键	438
10.3.2	安排加速键的几条规则	439
10.3.3	加速键表	439
10.3.4	加载加速键表	440
10.3.5	键盘代码转换	440
10.3.6	接收加速键消息	441
10.3.7	菜单与加速键应用程序 POPPAD	442
10.3.8	启用菜单项	448
10.3.9	处理菜单项	449
第十一章	对话框	453
11.1	模态对话框	454
11.1.1	创建 About 对话框	454
11.1.2	对话框及其模板	458
11.1.3	对话框过程	461
11.1.4	激活对话框	462
11.1.5	不同的主题	463
11.1.6	更复杂的对话框	466
11.1.7	使用对话框控制	473
11.1.8	OK 和 Cancel 按钮	476
11.1.9	避免全局变量	477
11.1.10	Tab 停留位和组	478
11.1.11	在对话框上绘图	480
11.1.12	将其他函数用于对话框	481
11.1.13	定义自己的控制	481

11.2 非模态对话框	488
11.2.1 模态对话框与非模态对话框的区别	489
11.2.2 新的 COLORS 程序	491
11.2.3 HEXCALC: 窗口还是对话框?	496
11.3 通用对话框	504
11.3.1 增强 POPPAD	504
11.3.2 Unicode 文件 I/O	527
11.3.3 更改字体	528
11.3.4 查找与替换	528
11.3.5 只调用一个函数的 Windows 程序	529
第十二章 剪贴板	531
12.1 剪贴板的简单使用	531
12.1.1 标准剪贴板数据格式	532
12.1.2 内存分配	533
12.1.3 将文本传送到剪贴板	535
12.1.4 从剪贴板上获取文本	536
12.1.5 打开和关闭剪贴板	537
12.1.6 剪贴板和 Unicode	538
12.2 复杂的剪贴板用法	544
12.2.1 利用多个数据项	544
12.2.2 延迟生成	546
12.2.3 私有数据格式	547
12.3 实现剪贴板查看器	549
12.3.1 剪贴板查看器链	550
12.3.2 剪贴板查看器的函数和消息	550
12.3.3 一个简单的剪贴板查看器	553

第一部分 基础知识

第一章

开始

本书介绍了在 Microsoft Windows 98、Microsoft Windows NT 4.0 和 Windows NT 5.0 下编写程序的方法。这些程序用 C 语言编写并使用原始的 Windows 应用程序编程接口 (API)。正如在本章稍后所讨论的,这不是编写 Windows 程序的唯一方法。然而,无论最终您使用什么方式编写代码,了解 Windows API 都是非常重要的。

正如您可能知道的,Windows 98 是最新的图形操作系统的代表,已成为围绕 32 位 Intel 微处理器(例如 486 和 Pentium)的 IBM 兼容的个人计算机事实上的标准。Windows NT 是运行于 PC 兼容机及一些 RISC(精简指令集计算机)工作站的 Windows 的工业增强型版本。

使用本书有三个先决条件。首先,您应该从用户的角度熟悉 Windows 98。您不要期望不了解 Windows 用户界面就可编写其应用程序。基于此,我建议您在开发程序(或在进行其他工作)时,使用基于 Windows 的机器,并使用 Windows 应用程序。

第二,您应了解 C 语言。如果您不了解 C 语言,一开始就编写 Windows 程序并不是一个好主意。我建议您在字符环境中,例如在 Windows 98 MS-DOS 命令提示窗口下提供的环境中学习 C 语言。Windows 编程有时包括一些非字符编程方面的 C 语言部分;在这些情况下,我将提供一些针对性的讨论。但大多数情况下,您应非常熟悉该语言,特别是 C 语言的结构和指针。了解标准 C 语言运行时库的某些知识是有帮助的,但不用必须了解。

第三,您应在机器上安装一个适于进行 Windows 编程的 32 位 C 语言编译器和开发环境。在本书中,假定您正在使用 Microsoft Visual C++ 6.0,该软件包可独立购买,也可作为 Visual Studio 6.0 软件包的一部分购买。

到此为止,我将不再假设您已经具有任何图形用户界面(例如 Windows 等)的编程经验。

1.1 Windows 环境

Windows 几乎不需要介绍。然而人们很容易忘记 Windows 给办公室和家庭桌面计算机带来的巨大变化。Windows 在其早期走过了坎坷的道路,征服桌面市场的前途曾一度非常渺茫。

1.1.1 Windows 简史

在1981年秋天 IBM PC 推出之后不久,就已经很明显,PC 上的主流操作系统将是 MS-DOS。MS-DOS 代表 Microsoft Disk Operating System (Microsoft 磁盘操作系统)。MS-DOS 是一个小型的操作系统。MS-DOS 提供给用户一种命令行界面,以诸如 DIR 和 TYPE 这样的命令为特征,它可以将应用程序装入内存来执行。对于应用程序员,它提供了一组函数调用进行文件的输入输出(I/O)。对于其他的任务——尤其是将文本或图形写到视频显示器——应用程序可以直接访问 PC 的硬件。

由于内存和硬件的限制,成熟的图形环境是慢慢来到小型计算机中的。当苹果公司那不幸的 Lisa 在 1983 年 1 月发布时,它提供了不同于字符模式环境的另一种可能的选择,并在 1984 年 1 月成为 Macintosh 上图形环境的一种标准。尽管 Macintosh 的市场占有率在下降,但是它仍然被认为是衡量所有其他图形环境的标准。包括 Macintosh 和 Windows 的所有图形环境都要归功于 Xerox Palo Alto Research Center(PARC) 在 70 年代中期所做的开拓性的工作。

Windows 是由 Microsoft 在 1983 年 11 月(在 Lisa 之后 Macintosh 之前)宣布,并在两年后的 1985 年 11 月推出。在此后的两年中,紧随着 Microsoft Windows 早期版本 1.0 之后,又进行了几次更新,以支持国际商业市场,并提供新型视频显示器和打印机的驱动程序。

Windows 2.0 是在 1987 年 11 月推出的。该版本对用户界面做了一些改变。这些改变中最有效的是使用了层叠式窗口,而 Windows 1.0 中使用的是“平铺”窗口。Windows 2.0 还增强了键盘和鼠标接口,特别是加入了菜单和对话框。

至此,Windows 还只要求 Intel 8086 或者 8088 微处理器,以“实模式”运行,访问 1 MB 的内存。Windows/386(在 Windows 2.0 之后不久推出的)使用 Intel 386 微处理器的“虚拟 86”模式,实现直接访问硬件的多个 MS-DOS 程序的窗口化和多任务化。为了统一起见,Windows 2.1 被更名为 Windows/286。

Windows 3.0 是在 1990 年 5 月 22 日发布的。它将 Windows/286 和 Windows/386 结合到一种产品中。Windows 3.0 有了一个很大的改变,这就是对 Intel 的 286、386 和 486 微处理器保护模式的支持。这能使 Windows 和 Windows 应用程序访问高达 16 MB 的内存。Windows 用于运行程序和维护文件的“外壳”程序得到了全面的改进。Windows 3.0