

BORLAND[®] C++程序设计

(第二版)

(美) Ben Ezzell 著
李光志 张知恒 包仕波 译



含Windows[™] 编程技术



电子工业出版社

Borland C++ 程序设计

(第二版)

(美)本·艾泽尔 著
李光志 张知恒 包仕波等 译

電子工業出版社

(京)新登字 055 号

内 容 简 介

Borland C++是在广受欢迎的 Turbo C 和 Turbo C++的基础上推出的新一代面向对象的程序设计语言。任何 C++ 和 C 程序员,均可利用 Borland C++提供的功能强大、快速高效的编译器,开发出各种应用程序。

本书分为三部分:第一部分简介 Windows 环境下的编程技巧;第二部分介绍资源工作台的原理与使用;第三部分介绍 Windows 环境下的图形编程。

本书有助于 C++ 及 C 程序员学习在 Borland C++环境下开发 Windows 应用程序。

本书英文版由 Addison-Wesley 出版公司出版,版权由作者所有。

Copyright ©1992 by Ben Ezzell

本书中文版经 Addison-Wesley 出版公司授权电子工业出版社独家出版。未经出版者书面许可,不得以任何手段复制或抄袭本书内容。

JS111/09

Borland C++程序设计

(美) 本·艾泽尔 著

李光志 张知恒 包仕波等 译

责任编辑:魏冬 脱全禄(特约)

*

电子工业出版社 出版

北京市海淀区万寿路 173 号信箱(100036)

电子工业出版社发行 各地新华书店经销

北京市万龙图文信息公司排版

北京市顺新印刷厂印刷

*

开本:787×1092 毫米 1/16 印张:26.75 字数:645 千字

1995 年 2 月第一版 1995 年 2 月北京第一次印刷

印数:5,000 册 定价:35.00 元

ISBN 7-5053-2661-9/TN·784

译 者 序

BORLAND C++ 是在广受欢迎的 Turbo C 和 Turbo C++ 的基础上推出的新一代的面向对象的程序设计语言,是其他 C 和 C++ 的进一步发展。Borland C++ 适用于任何 C++ 和 C 程序员,他们可以利用 Borland C++ 所提供的功能强大、快速且高效的编译器,开发出任何应用程序,包括 Microsoft Windows 应用程序。

本书的目的在于帮助有经验的 C++ 和 C 程序员学习在 Borland C++ 环境下开发 Windows 应用程序。本书在讨论 Windows 编程技术的同时,还介绍了 Windows 的原理与内部结构,是程序员深入理解 Windows 的逻辑原理的有力帮手。

本书共分为三部分。第一部分简要介绍了 Windows 环境下的编程技巧;第二部分详细介绍了 Borland C++ 的新技术:资源工作平台的原理与使用;最后一部分讲述了如何在 Windows 环境下进行图形编程。

本书的翻译工作由李光志、张知恒以及包仕波等同志完成。同时还有许多同志在本书的翻译过程中做了大量的工作,如郭林、曾林、王庆德、陈林生等,在此特别致以感谢。

译者

1994. 6

献 辞

自从第一台计算机在二战以后诞生以来的几十年间,计算机一直是一种十分稀罕的设备,它由一帮电子技术专家和数学家们组成的奇特的小组精心地照料着。渐渐地,产生了一种新的职业,有了诸如系统分析员、程序员等等不同的称谓。然而,此行业的大部分从业者一开始并不是软件专家。他们来自各行各业,如果要把他们原来的职业列出一份清单来的话,它就象一本包罗各种职业的百科全书。

当一种新的行业刚刚诞生,而人们还来不及建立一种机制来考核谁可以从事该行业和怎样去从事该行业的时候,就总是这种情形。正是因为有了这种自由度,才使得具有各种背景和各种天赋的强手们给这一行业带来生机和活力,因为它不可能由哪一个单一的行业来派生。

这一切在过去的年代里已经发生。

在十五世纪中叶,欧洲的各种行业都由封建行会控制着,技术只是通过死记硬背的方式机械地流传。这时,在德国的梅兹,一个三十岁的金匠与他的行会和其他的政治势力发生了冲突,结果,他开始了一系列实验,最终改变了整个世界。虽然这并不是他本来的意图,但他的工作最后导致了不仅是行会,而且是当时整个的政治、社会和宗教系统的崩溃——不只限于欧洲,而是整个世界。

这件大事的余音迄今不绝,发聋振聩!

十五世纪五十年代,约翰·古登堡发明了可实际使用的活字,从而掀起了传播史上的革命。活字首先被用于印刷学校的语法书、教堂的免罪书,甚至还印了一本《圣经》。他把书从抄写员手中拿走,置之于普通的平民大众的手中,——从那时起,这世界就再未曾沉默过!如同印刷技术开始了传播业的革命一样,计算机继续着对旧秩序的破坏,它使得信息为更多的人所知。

传播——一开始以印刷的形式——带来了十六世纪的宗教改革运动,然后是《自由宣言》(更近的则是柏林墙的倒塌)。未来就掌握在你的手中,正如此刻传播的工具正掌握在你手中一样——因为传播有多种方式。

几年前,我的妻子十分满足地说:“印刷的力量属于掌握它的人。”可是,学生们却告诉我们,使用传真机组成的地下网络,印刷不过只是传播力的一种……正如 Windows 3. x 是另外一种。

本书——它也是印刷力的一个例子——谨献给我们的第一位传播者:

约翰·古登堡

1394—1468

传播业之父

致 谢 辞

经常地,当完成了这样一本书之后(在经历了校对,改正,和一系列不小的关键性的细节以后),作者往往只顾上欣喜若狂,而忘记了整个事情的前前后后。因此,他可能会忘记对准备这本书的其他人(他们也花费了不少的时间和精力)说一声衷心的、毫不过份的“谢谢”。

因此,为免于忘却,我想利用这个机会感谢几位人士,为把这本书编纂成册,他们花费了血汗(有时)甚至是泪水。他们是:

Borland International 公司的 Chris Ohlsen,他审读了此书,提出了许多有价值的意见,对此,我深表感激。

Borland International 公司的 Nan Borreson,她高效地为我处理很多细节问题,甚至对我最为简单而愚蠢的问题也始终表现出极大的耐心。

同样的感激之情献给 Benchmark Productions 公司的 Chris Williams, Amy Pedersen 和 Andrew Williams,感谢他们的热忱。

谢谢你们每一位,不论你们是作为集体还是以个人的名义,非常感谢!

本·艾泽尔

目 录

第一部分:Windows 3.0 编程介绍	(1)
第一章:安装 C/C++ for Windows 3.0	(3)
BC++ 和 TC++ 的比较	(3)
QEMM-386 6.0 版	(5)
环境和术语	(7)
编译 Windows 程序	(7)
WinHello 一个 Windows 入门介绍性程序	(8)
消息编程或事件驱动编程	(13)
WndProc 过程	(14)
记号、常量和变量	(16)
小结	(22)
第二章:应用文本到 Windows 的传送	(25)
非确定环境设计	(25)
WM_PAINT 消息的处理	(27)
图形文本显示要点	(28)
Windows 字体制式	(29)
文本输出的窗口化	(32)
滚动条	(34)
窗口尺寸/尺寸重置	(37)
滚动错误	(38)
小结	(39)
第三章:键盘、光点和滚动条	(47)
键盘驱动器	(47)
WM_XXXX 事件消息	(48)
虚拟键码值	(51)
字符事件消息	(54)
KeyCodes 程序	(54)
文本输入处理	(56)
光点(光标)定位	(59)
字体和字符宽	(63)
ANSI 字符集的支持	(66)
ANSI(扩展 ASCII)字符的产生	(65)
小结	(66)

第四章:Windows 中的鼠标	(75)
鼠标信息	(76)
Windows 中的鼠标事件	(78)
Mouse1:跟踪鼠标	(78)
Mouse2:鼠标的光标	(79)
Mouse3:鼠标按键测试	(82)
非用户窗口消息	(83)
鼠标结束消息	(84)
小结	(84)
第五章:子窗口和控制单元	(93)
Windows 按钮	(93)
按钮类型	(94)
按钮操作	(95)
按钮控制通信:发送和接收	(98)
按钮(窗口)标签	(100)
第二部分:资源工作平台介绍	(101)
资源文件	(101)
资源编程——不是源代码编写	(102)
资源工作平台与资源工具箱	(102)
资源元件	(103)
第六章:应用资源与资源文件	(105)
可编辑性和不可修改性	(106)
文件与文件类型	(107)
资源管理器	(108)
建立新的资源元件	(111)
编辑已存在的资源元件	(112)
资源标识符	(113)
小结	(115)
第七章:位图、光标、图标与字体	(117)
图象编辑	(117)
检测 SVGA 与 VGA 颜色	(119)
图形编辑工具	(120)
光标资源	(123)
图标资源	(125)
用于 FileView.C 的两个图标	(125)
用户字体	(126)
小结	(128)
第八章:对话框编辑器	(129)

对话框编辑器	(129)
Tools 菜单	(130)
对话框风格	(131)
对话框控制元件	(134)
Controls 菜单	(138)
控制元件的对齐、定位与大小设置	(140)
对话框菜单	(142)
头文件	(142)
对于 FileView 应用程序的 3 个对话框	(143)
小结	(145)
 第九章:菜单编辑器	(147)
菜单编辑器的使用	(147)
为 FileView 建立一个菜单	(151)
小结	(152)
 第十章:快捷编辑器	(153)
快捷定义	(153)
FileView 的快捷定义	(155)
小结	(156)
 第十一章:字符串资源的编辑	(157)
字符串定义	(157)
字符串编辑器	(157)
有关 FileView 的字符串表	(159)
小结	(159)
 第十二章:资源工作平台中的头文件	(161)
 第十三章:程序装配:FileView 程序	(163)
编译器警告信息的预防	(164)
WinMain 中资源的装载	(164)
对话框的建立	(165)
对话框初始化	(166)
小结	(167)
 第十四章:信息对话框	(183)
例一:信息对话框	(183)
例二: Borland 窗口用户控制	(186)
 第三部分:在 Windows 下的图形设计	(195)
 第十五章:GDI(图形设备接口)的介绍	(197)

设备的选择	(198)
设备内容信息	(198)
映象模式	(203)
屏幕(窗口)原点	(204)
以物理单位来应用映象模式	(205)
可变的映象模式	(206)
MODES.C	(206)
小结	(207)

第十六章:Windows 中的颜色

Colors.C	(225)
颜色和画图模式	(228)
小结	(229)

第十七章:图形及数字的绘制

图形工具	(239)
图形的形成	(241)
商业图表	(244)
多边形的画法	(246)
小结	(248)

第十八章:刷子、位图、BLT 和 DIB

位图刷子: I	(263)
位图刷子: II	(264)
较大位图的应用	(266)
图象的存储	(267)
SetBitmapBits/GetBitmapBits	(268)
单色位图	(268)
与设备无关的位图	(270)
与设备无关的位图的形成和应用	(274)
位图的放大和缩小	(276)
位图运算的分析:PenDraw5	(277)
小结	(278)

第十九章:元文件操作

元文件的存储	(289)
把元文件写入磁盘	(291)
磁盘元文件的获得	(293)
使用元文件的注意事项	(295)
小结	(296)

第二十章:图形的版面和风格

TextOut 和 SetTextAlign 函数	(301)
---------------------------------	-------

修改设备内容	(304)
文字字体	(304)
有关字体历史的简短回顾	(305)
版面	(306)
字体资源文件	(308)
逻辑(常用)字体的使用	(309)
创建逻辑字体	(313)
Fontsl 演示程序	(314)
计算逻辑 TWIPS 中的字号大小	(315)
小结	(315)

第二十一章:打印机设备内容	(325)
打印机与视频监视器的比较	(325)
打印机的初始化	(326)
输出到打印机设备内容	(326)
Escape 子函数	(327)
对输出例程的修改	(328)
对打印机的直接操作	(329)
输出图形的分区	(329)
分区处理的应用	(329)
打印进程的退出	(330)
退出对话	(331)
作业的放弃	(333)
文本输出	(333)
可用字体与设备功能	(333)
系统的菜单入口	(335)
小结	(335)

第二十二章:剪贴板数据传输	(343)
剪贴板功能构成	(344)
剪贴板数据格式	(344)
特殊用途的格式	(346)
访问剪贴板	(346)
ClipBd.C 演示程序	(349)
文本文件剪贴板操作	(349)
位图剪贴板传输	(350)
元文件剪贴板传输	(351)
其他剪贴板格式	(354)
剪贴板延迟发送	(355)
所有者剪贴板数据显示	(355)
用户格式的其他用途	(356)
小结	(357)

第二十三章:动态数据交换(DDE)	(365)
--------------------------------	--------------

DDE 简介	(365)
DDE 术语	(366)
DDE 谈话:原子和字符串	(368)
其它 DDE 通信元素	(369)
程序 Dde_Data 和 Dde_Main	(372)
服务程序 Dde_Data	(372)
DDE 消息传送	(374)
小结	(377)
 附录 A:常量/标志/宏/结构	 (391)

第一部分

Windows 3.0 编程介绍

对许多程序员来说,编写 Windows 3.0 应用程序使他们突然离开了熟悉并习惯了的环境——如果他们在用 C 编程的话,这甚至是一个令人感到惊讶的变化,就象他们忽然被弹入一个奇怪的,几乎并行的宇宙空间一样。

对于其他一些人,尤其是使用过 Windows 以前版本或 OS/2 的人,这些新环境看起来则不那么奇怪,或者甚至会熟悉得象老朋友一样。

是否那些用过以前的 Windows 或 OS/2 的人比从未在多任务环境中编程的人占了些便宜呢? 这还是个争论未决的问题。

这个论断的部分原因是 Windows 3.0 中有许多改变和增强,使得即使是熟练的 Windows 程序员也得学些新的技术。更重要的是,因为 Borland C++ 编译器和 Whitewater Resources Toolkit 提供的特点和性能使编程的重点从 Windows 经验转移到了总体计划编程技巧上;也就是说,用 Borland C++,至关重要的是程序员的全面技术素质,而不是对 Windows 编程的熟练程度。

以上这些并非是说你可以编 Windows 程序而不懂 Windows;只是说由于这些工具(编程器及工具箱)替你干了脏活重活,你学习和理解将更容易一些。

但是,本书重点仍然是在理解 Windows 编程如何进行上;就是说,当然也得明白 Windows 是如何工作的。

因此,在第一部分,我先帮你设置起你的 BC++ 编译器及 Toolkit。然后,我将介绍 Windows 显示的基础知识,告诉你如何使用鼠标操作,并讨论在一个消息驱动的环境下编程的宗旨。

准备好了吗? 我们开始了。

第一章

安装 C/C++ for Windows 3.0

在系统硬盘上安装 Borland C/C++ 3.0 可能是你在计算机上完成的最简单的任务。只要插入安装盘,键入 A:>INSTALL。然后听从屏幕上的指导。

在开始前,必须知道完整的 3.0 软件包包括 Borland C/C++ 编译器、工具、应用、示例,总共要 26MB 硬盘空间。要想减少空间需求,可以只装特定的存储模式(在安装菜单中选),或限制示例及一些外围应用程序的安装。

除了准许裁减安装的硬盘开销,Borland 的 Install 程序还允许修改驱动器、目录及各种程序和示例安装的子目录。除了 BORLANDC 目录(BC)外,还有一项也该特别注意,那就是 Windows 目录所处的位置。

缺省时,安装程序认为 Windows 安装在 C:\WINDOWS 下。在以后的安装过程中,程序试图提供一个 Windows 安装实用程序来完成在 Windows Program Manager 下创建 C/C++ 编译器组的任务。如果在安装过程中正确地指定了 Windows 目录,在下一次 Windows 装载时,GROUPS 实用程序将自动执行,如图 1-1 所示。

GROUPS 实用程序执行起来很简短,但很有礼貌;开始时,显示如图 1-1 的对话框,提供进行或退出的选项。背景的图像当然是纯属可有可无的,但 GROUPS 的图标应出现在左下角(在图中是右边数第二个图标)。

选中缺省(OK)按钮后,GROUPS 将在 Program Manager 下创建一个新任务组,标题为 Borland C++,它包括图 1-2 中所示的八个应用程序。

如图 1-2 所示,Borland C/C++ 3.0 软件包包括两个分离的编译器 Turbo C++ 和 Borland C++,还有 Borland Resource Workshop, Turbo Debugger for Windows, Turbo Profiler for Windows, Import Library 实用程序,和 Winsight(一个 Windows 间谍实用程序)。当然最重要的两项是两个编译器,Borland C++(BC)和 Turbo C++(TCW)。

BC++和 TC++的比较

Borland C/C++ 2.0 版介绍了一个新编译器叫 BC,替换了熟悉的 TC 编译器。BC 将变成

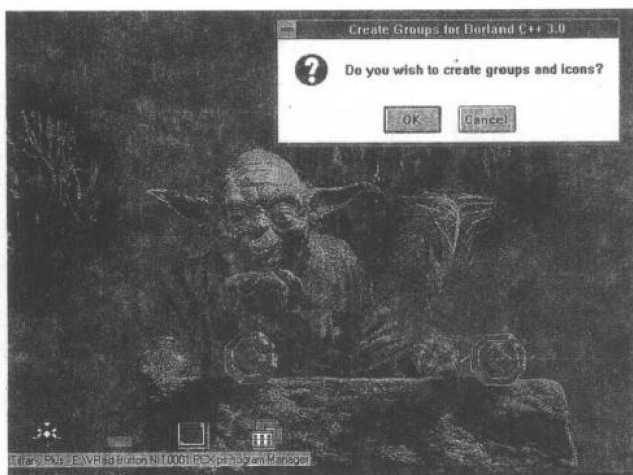


图 1-1 Borland C++ Group 的创建

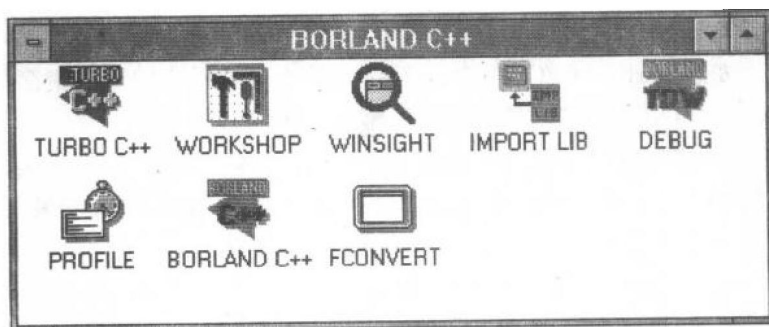


图 1-2 Borland C++应用程序

新的专业程序员使用的编译器,而旧的 TC 版本将以一种业余消遣的版本出现,只不过软件包中带的饰件和实用程序更少,显得更经济。

从 TC 到 BC 2.0 版的变化绝不止形式上的变化。BC 编译器可以处理四种类型的应用:传统 DOS 程序, DOS 覆盖程序, Windows 3.0 应用, 及 DLL (动态链接库) 单元。而当 BC 2.0 编译 Windows 应用和 DLL 时, 编译器并不在 Windows 下工作 (除非是在 Windows 的 DOS Shell 中)。而且即使在 Windows 的 DOS Shell 下编译和链接也是明显地缓慢的。

另一方面来说, 用 BC 时, 一般不必在 Windows 环境下那么慢地编译 Windows 应用。实际上, 大多数程序员更喜欢在 DOS 下编译, 然后转到 Windows 下测试可执行代码, 尽管这种从 DOS 转到 Windows 再转到 DOS 下的转换是很不方便的, 但它能节省很多时间。

例如, 一个应用在 DOS 下用 BC 编译链接只需要 16.20 秒 (在 33MHz 386 系统), 在 Windows 的 DOS Shell 下编译链接就需要 261.63 秒——比 DOS 下慢 16 倍。但使用 3.0 版的 TCW, Windows 下编译链接时间几乎与 DOS 下用 BC 是一样的。

在 3.0 版中, BC 编译器还为 DOS 和 Windows 编程提供 4 个编译选项。但在 3.0 版, 以前的 TC 编译器重新以 TCW 形式出现, 提供了一个真正的 Windows 兼容的编译器。更重要的是, TCW 编译和链接 Windows 应用和 DLL 没有 BC 编译器的时间障碍。

请注意, 不要以为有了 TCW 就可以删除掉 BC 版本了, 因为 TCW 只提供了 Windows 应

用和 DLL 编译器,而未提供任何 DOS 应用的编译器。

因此,BC 和 TCW 编译器都是很重要的,要在他们各自适当的环境中发挥作用。

同时,对于少数的仍坚持不肯用集成环境的死硬派,Borland C/C++也提供了一个命令行编译器(当然,BCC 对于你偶尔编的很大的程序不能调入内存时也是很有用的)。

QEMM-386 6.0 版

QEMM 是一个扩展内存管理器。除了其他功能,在 80386/80486 系统中,QEMM 还允许将 TSR 实用程序调入高端内存(内存地址在 640K 与 1024K 之间)。因为这些内存区域传统上是为系统硬件保留的,但只有一部分被真正地使用。通过使用这些空间装入 TSR,如你的鼠标驱动、ANSI.SYS、其他驱动程序、实用程序等,640K 低端内存将有更多空闲空间可以供你的主要应用(如编译器)和 Windows 等来使用。

不幸的是,以前 QEMM-386 版本不是和 Windows 百分之百兼容的,这使得在使用 DOS 或 Windows 3.0 时得分别重新启动不同的配置。

Quarterdeck 最近发布了与 Windows 3.0 兼容的新版本 QEMM-386。升级到 QEMM-386 6.0 版(或 5.1 版以上),可提供在高端内存装入 DOS 终端驻留程序的种种便利,并且,对于 DOS 或 Windows,它不需要不同的启动配置。

在大多数情况下,QEMM-386 用一种与 Windows 3.0 全兼容方式安装自己,并且使用 OPTIMIZE 实用程序,在正确的操作中很少或根本不需要修改。

但是,大多数情况并非所有情况。根据视频系统的不同,你可能会在试图装入 Windows 3.0 时得到如下消息:Cannot run Windows because of video device conflict。

有一种解决办法。你可以修改 CONFIG.SYS 文件使一部分特定内存区不受 QEMM-386 的内存管理。在安装后,CONFIG.SYS 的第一行应如下:

```
DEVICE=C:\QEMM\QEMM386.SYS RAM
```

如果和 Windows 3.0 有视频冲突,Quarterdeck 建议进行如下修改:

```
DEVICE=C:\QEMM\QEMM386.SYS RAM X=0000-3FFF X=A000-C7FF X=F000-FFFF
```

权衡后的 CONFIG.SYS 文件可能如下所示:

```
BREAK=ON
```

```
STACKS=0,0
```

```
BUFFERS=25 /X
```

```
FILES=30
```

```
LASTDRIVE=E
```

```
SHELL=C:\DOS\COMMAND.COM /P /E:256
```

```
DEVICE=C:\QEMM\LOADHI.SYS /R:1 C:\DOS\ANSI.SYS
```

```
INSTALL=C:\QEMM\LOADHI.COM /TSR /R:1 C:\DOS\FASTOPEN.EXE
```

```
C:=(50,25)
```

```
DEVICE=C:\QEMM\LOADHI.SYS /R:1 D:\WINDOWS\SMARTDRV.SYS 256
```

注意 HIMEM.SYS(由 Windows 安装的)已被去掉,被 SMARTDRV.SYS 使用的内存大小也减少了。

这些都是可选的变化,但第一行的包含语句却是很重要的修改。在前 1 兆字节内存中排除了以下区域: