Windows程序设计

——用 Borland C++& Turbo C++for Windows

章生立　董三立

北京航空航天大学出版社

Windows 程序设计

—用 Borland C++ & Turbo C++ for Windows

章生立　董三立等　编

北京航空航天大学出版社

(京)新登字 166 号

内 容 简 介

本书介绍利用 Borland C++和 Turbo C++ for Windows 进行 Windows 程序设计的技术和参考信息。首先介绍 Windows 的优点、概念和术语，如何控制 Windows 环境和用 C 与 C++编写简单的应用程序。接着讨论如何控制 Windows 窗口；如何设计图标、光标和位图；如何开发菜单和键加速器；如何编写使用对话框和字体的代码；如何绘制图形[科学图、数学图、饼形图、条形图和线图等]；如何设计多媒体的程序。再阐述使用 ObjectWindows 类库(OWL)开发 Windows 应用程序的基本概念和技巧；介绍了从 Windows 3.1 移植程序到 Windows NT 的方法。最后列出了 Windows API 和 ObjectWindows 类库参考和 Borland 集成环境工程管理的用法。适合程序员使用，也可供计算机专业高年级学生、研究生参考。

图书在版编目(CIP)数据

Windows 程序设计：用 Borland C++和 Turbo C++ for W
indows/章生立等编. —北京：北京航空航天大学出版社
1995.6
ISBN 7-81012-577-X

Ⅰ.W… Ⅱ.章… Ⅲ.①C 语言-程序设计②操作系统，W
indows-程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(95)第 04672 号

- 书　　名：Windows 程序设计——用 Borland C++&Turbo C++ for Windows
- 编 著 者：章生立　董三立等编
- 责任编辑：王鉴莉　许传安
- 出 版 者：北京航空航天大学出版社
- 地　　址：北京市海淀区学院路 37 号(100083)，发行科电话：2015720
- 印 刷 者：朝阳科普印刷厂印刷
- 发　　行：新华书店总店科技发行所
- 经　　售：全国各地书店
- 开　　本：787×1092　1/16
- 印　　张：45
- 字　　数：1168 千字
- 印　　数：5000 册
- 版　　次：1995 年 6 月第 1 版
- 印　　次：1995 年 6 月第 1 次印刷
- 书　　号：ISBN 7-81012-577-X/TP・165
- 定　　价：54.00 元

编者前言

Borland C++的前身为 Turbo C。从 2.0 版开始，Turbo C 演化为 Borland C++和 Turbo C++两个系列。Borland C++后来又推出了 2.0，3.0，3.1 和 4.0；Turbo C++推出了 2.0 和 3.0。Borland C++非常庞大，主要被专业程序员使用，而 Turbo C++相对较小，主要用于教学，为初学者所喜爱。

从 Turbo C++ for Windows 1.0 开始，集成环境运行于 Windows，是一个真正的 Windows 应用程序。而 Borland C++从 4.0 开始，是一个很好用的 Windows 应用程序开发环境。要特别指出的是，新的 IDE(集成开发环境)可在 Windows 中直接调试程序，工程管理比以前更方便。

Borland C++和 Turbo C++ for Windows 配备了 Turbo Debugger for Windows 用来调试设计的 Windows 应用程序，增强了包括图标、光标、位图、对话框、菜单、热键和字体等的资源设计。特别是在 Borland C++ 4.0 中，资源设计已得到显著增强，用来设计 Windows 应用程序的界面。Borland C++在 OWL(ObjectWindows Class Library)中封装了 Windows API(Application Program Interface)包括函数、数据结构和消息等，可以加快读者用C++开发 Windows 应用程序的速度，减轻读者用C++开发 Windows 应用程序的难度。

本书适用于 Borland C++和 Turbo C++ for Windows 以及各种演化版本。书的最后，还介绍了用 Borland C++ 4.0 开发 Windows NT 应用程序的技术。

书中的例子针对国内程序员的需要，全部采用中文的菜单、对话框和提示界面，每个程序都有很好的注释和解说，便于理解和学习。

目 录

第一章 Windows 和 Windows NT 介绍

Borland C++编译器以及相关的 Windows 和 Windows NT 的开发工具为 C/C++程序员编辑、连接 Microsoft Windows 和 Windows NT 程序提供了很大方便。用户对 Microsoft Windows NT 程序设计感兴趣的主要原因是 Microsoft 成功地基于图形的操作环境。Windows 提供了按钮式控制方式和下拉式菜单。

本书中，Windows 3.1 术语适用于只同基于 OS 环境的 Windows 有关的概念，Windows NT 术语适用于只同基于 Windows NT 环境有关的概念。相对于环境，Windows 术语的程序设计概念都会用到。

本书将集中精力介绍怎样使用 Borland C++编译器在 Windows 3.1 和 Windows NT 环境中开发程序。

本章目的是对有关 Windows 3.1 和 Windows NT 的重要概念做说明。

针对 16 位的 Windows 环境而编写的应用程序，可以在 Windows 3.1、Windows NT 和 Windows 95 环境下运行。针对 Windows NT 环境编写的 32 位应用程序只能在 Windows NT 和 Windows 95 下运行。许多标准的 DOS 程序可以从命令提示符环境，移植到这两种环境下运行。

如果要针对 16 位的 Windows 3.1 开发应用程序，必须要有 Borland C++ 3.1 的编辑器或更新版本，还要有有关 Windows 的工具。

针对 32 位的 Windows NT 环境开发应用程序，应该具备 Borland C++4.0 的编译器和有关 Windows 的工具。

1.1 什么是 Windows

Windows 是一个基于图形的多任务窗口环境，它可以使针对 Windows 编写的程序有始终如一的外形和命令结构。由于有这一功能，即使是新的程序员也能轻易学会。

Windows 提供了几个内建的例程，可以使下拉式菜单、滚动条、对话框、图标和其它的友好图形界面功能轻易完成。从 Windows 3.0 到 Windows NT，程序员可以使用新的对话控制、菜单类型和“自画控制”。通过利用大量图形程序设计语言，程序员可以轻易用不同的字体和间距格式化并输出文本。

Windows 将视频显示器、键盘、鼠标、打印机、串行端口和系统时钟按独立设备方式处理。这种处理方式可以使相同的程序在不同的硬件配置上运行一致。

1.2 Windows 的功能

Windows 操作环境同 MS-DOS 环境相比，前者为用户和程序员提供了更多的有利条件。Windows 有三个主要功能：一个是面向图形的用户界面；一个是多任务功能；另一个是硬件独

立性。单独说这三项功能没有一项是新的，但是将其合并到一个单独的微机操作环境中，就成为一种新思想。

1.2.1 标准的用户界面

在 Windows 提供的三项主要功能中，面向图形的用户界面最引人注目，对用户来说当然是最重要的。用户界面用与其一致的图形来表示驱动器、文件、子目录和许多操作系统命令和行为。图 1.1 显示了一典型 Windows 窗口的外观。通过标题条确定程序，通过点和单击鼠标可以在程序菜单中访问许多基本的文件操作函数。多数 Windows 程序都提供了键盘界面和鼠标界面。尽管 Windows 程序中的多数函数都能通过键盘控制，但鼠标控制对很多任务来说更为简单。

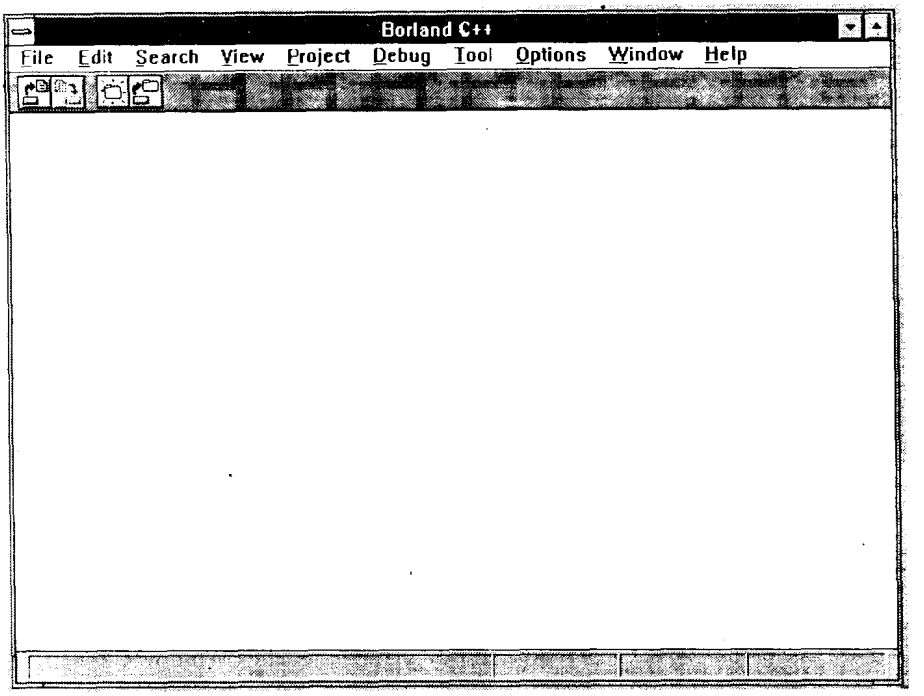

图 1.1 一个典型的 Windows 应用程序窗口

1.2.2 多任务

一个多任务操作环境允许用户同时运行几个程序或同一个程序的几个实例。现在许多用户仍然想知道多任务在微型计算机上有没有必要。图 1.2 显示了几个 Windows 程序。每个程序在屏幕上占据了一个矩形窗口。任何时候用户都能在屏幕上移动窗口，在不同的程序之间切换，改变窗口的大小，从一个窗口到另一个窗口交换信息。

1.2.3 内存管理

在 Windows 环境中，内存是一项重要的共享资源。当同时运行程序的数目超过一个时，每

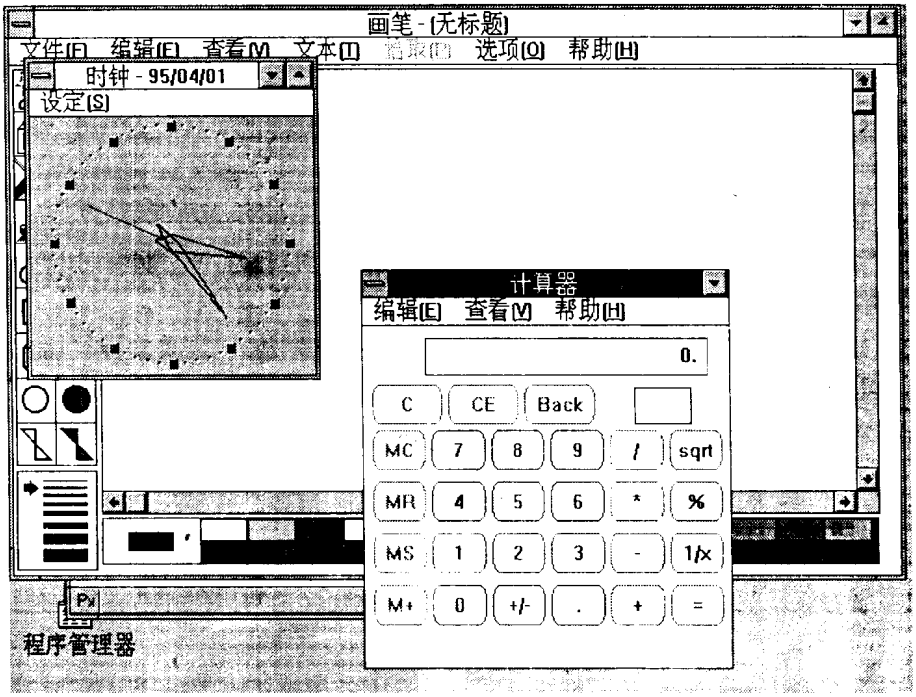

图 1.2 在 Windows 环境下多任务的几个应用程序

个程序都必须协作共享内存，目的是不浪费储备。当新程序开始运行旧程序终止时，内存可能会分裂。Windows 能够通过在系统内存中移动代码块和数据块来统一自由内存空间。

1.2.4 队列输入

另一项重要的共享资源方式是从键盘和鼠标上输入。正因为如此，不应该用 C 和 C++中的许多输入输出语句，例如 getchar，而用 Windows 中的具体函数调用代替。在 Windows 环境下，一个应用程序不能利用明确的调用从键盘或鼠标输入设备上读取信息。相反，Windows 是按系统顺序从键盘、鼠标和计时器上接受所有的输入。队列的任务是将输入信息从系统行列拷贝到程序行列，使其改道在合适的程序中。此时，当程序准备接受输入时，程序从其本身的队列中读取并向正确的窗口分发一个消息。

1.2.5 消息

多任务环境中传播数据的主要手段是 Windows 消息系统。从程序的观点看，消息以通知的形式看到，其中一些有意义的消息已经存在，它们关联(也可不)一项具体的行为。这些事件可由用户自己启动，例如单击或移动鼠标，改变窗口的大小，或选择菜单。消息也可以由程序本身产生。比如，一个基于图形的电子表格由于需要更新一个显示条图而应该进行循环。这时程序将向其自身发出一个“更新”窗口的消息。

Windows 本身也能产生消息，遇到“关闭过程”时产生。在这种情况下，Windows 将关闭的意图通知给每一个程序。最后，会从设备监视程序中产生一个额外的消息源，它指明一个重要

的化学过程已经到达了指定温度。不考虑消息源时，用户的程序必须有适当的行为。

1.2.6 设备的独立性

Windows 提供的另一项功能是硬件设备的独立性。Windows 将程序员从必须考虑监视器、打印机等不同种类的现代输入设备中解放出来。在 Windows 的环境中，每个设备——不管是视频显示器、打印机还是其它——都只需写一次，不必每个软件公司都去编写自己完整的集合，因为生产硬件的公司对自己生产的系统已经编写了驱动程序。Windows 中有许多驱动程序，其它的可以从制造商那里得到。

用户的程序是同 Windows 打交道，而不是其它的具体设备，不需要知道打印机连接的是什么。比如程序通知要画一个实心矩形，Windows 就会考虑如何完成这项任务。同样的原因，每个设备驱动程序将通过 Windows 中的一个程序来进行工作。

1.2.7 动态连接库

大多数 Windows 的功能是由新动态连接库提供的，称为. DLL。通过提供有利而又灵活的图形用户界面增强了基本操作系统。动态连接库含有预定义函数。将其装入时，它同应用程序动态地连接，而不是在. EXE 文件生成时静态地连接。

动态库的概念并非起源于 Windows。例如，对 C 编译器来说，要想实现不同系统的标准功能，对库的依赖性就很强。连接器将运行的库函数，例如 getchar 和 printf，复制到程序的执行文件中。函数库使程序员避免了把对象读入字符或格式化输出一类的一般操作都重建一个新过程的工作。程序员可以建立自己的库来包含附加的功能，例如改变字体和对齐文本。把可利用的函数当成通用工具以消除多余的设计。

当一个 Windows 程序调用一个 Windows 函数时，编译器必须为该函数的调用生成一段机器代码，该函数位于一个 Windows 库中的一代码段上。这也存在一个问题，因为在 Windows 程序中实际运行之前，Windows 函数的地址还不知道。

解决这个问题的方法用户已很熟悉——即迟后联编或动态连接。连接器允许用户调用的函数在连接时不必完全确定。只有当程序装入内存准备运行时，远程函数调用才确定。

1.3 Windows 的特征

由于使用了 Windows 3.0 版本，大大提高了计算机的运行、内存管理和多任务功能，还有许多其它的主要变化：

- 直接访问扩充内存
- 完整的内存管理
- 调色板管理程序能使一个应用程序充分利用设备的彩色功能。
- 与设备无关的彩色图标和位图
- 与设备无关的光标和位图，能从程序提供的一套预先定义的图象中自动选择合适的指定设备的图象。
- 一个增强的用户界面，能允许应用程序使用新的对话框控制、菜单类型、“自画”控制和选择更多种类的字体。

- 一个适应性更强的安装过程。
- Windows 帮助编译器。

1.4 Windows 的概念和术语

可以将有关 Windows 的概念和术语分成两种主要类型：Windows 中可见的功能，比如菜单，图标等等；另一种是屏幕后面的操作，比如消息，函数访问等等。为使 Windows 程序开发人员之间能有效地交流，对 Windows 中的所有性能都给出一个名字和有关的使用方法。本节将介绍开发 Windows 程序所需的最少用语。

1.4.1 一个窗口的定义

对用户来说，一个 Windows 窗口就是显示器上的一个矩形部分，同时带有一个与程序无关的、始终如一的外形，这就是用户和程序产生的窗口之间的可见界面。对一个应用程序来说，窗口就是在屏幕上由这个程序控制创建并控制的一块矩形区域。该程序创建和控制主窗口中的所有功能，包括它的大小和形状。当用户启动一个程序时，就创建了一个窗口。用户每次单击一个窗口选项，程序就会做出相应的反应。关闭一个窗口，就终止了一个程序的运行。就象这种可视用户/应用程序信息交换一样，窗口在表示 Windows 系统基本的子结构中起了重要的作用。

例如，当窗口本身表示面向图形的用户界面时，它也是多任务和硬件无关性的可视表示。正是这种运行 Windows 程序的可见重叠性，才为用户提供了真正的多任务功能。通过将屏幕分成不同的窗口，用户可以利用键盘或鼠标选择一个当前运行的程序，将输入直接对着多任务环境中某个指定的程序。然后 Windows 截取用户的输入，并在需要时分配所有必要的资源(例如处理器)。

1.4.2 可视界面

Windows 程序的某些功能和行为是相同的：边界、控制框和 About 等等。这为判断各个程序的属性提供了方便。图 1.3 图示了一个窗口的主要成分。

1.4.2.1 边 界

所有的 Windows 窗口周围都有边界。边界由线组成，最后形成一个窗口。边界看起来象是从别处描绘了一个程序的屏幕视图口，但是，仔细研究几个重叠的窗口程序，就会发现，其间有很大差别。边界不只是作为屏幕真实的等级界线，同时也表示了这个活动窗口。移动鼠标指针到一条边界，按下左按钮，用户就可以改变窗口的大小。

1.4.2.2 标题条

标题条在窗口的上端显示了这个应用程序的名字。标题条位于上端，相关窗口的中间处。这对帮助记忆当前是哪个程序在运行很有用处。

1.4.2.3 控制框

Windows 程序用到了一个控制框，这是一个小方框，左上角有一短划线。在控制框内单击鼠标指针就会使窗口显示系统菜单。