Borland C++ 3.1 和 4.0 使用手册

[美] Chris H. Pappas & William H. Murray

黄 昱 李 英 王慧敏 译

周志宇 审校

清华大学出版社

目 录

第三部分 高性能程序设计

第四部分 Windows 和 Windows NT 程序设计

此为试读,需要完整PDF请访问: www.ertongbook.com

第五部分 附 录

第一部分

绪　　论

第1章

整体是部分的和

读者在一个硬件商店中准备购买一种商品时，是否曾面临必须在两种工具中做出选择？一种工具标有“通用”，另一种工具标有“专业”。根据已有的经验，通用工具使用的机会很多，但从长远的观点看，通用工具不能被扩充使用。另一方面，专业化的工具通常设计得比较好，含有附加的功能，且经得起频繁使用。读者将购买哪种工具呢？通常要根据价格、性能、质量和使用目的来做决定。

当购买 Borland C++ 软件包时，读者很可能正在寻找一种编程工具，或一组工具，这些编程工具应具有经得起频繁使用的灵活性和性能。这一软件包不会因为读者编程技巧的增加而过时。选择 Borland C++ 软件包便是最佳的选择。使用 Borland C++ 软件包并不要求读者是一个专业的程序员。

本书将试图做三项基本工作。首先，本书将帮助读者研究和理解 Borland C++ 软件包中个别组成的定义和用法。其次，本书是一本完整的 C、C++ 和汇编语言的教科书。它将教给读者编写 C、C++ 和汇编语言程序的规则和技巧。读完本书后，读者所学的技巧将使读者从初级变为这些语言的中级水平的程序员。第三，本书将向读者示出集成该编译软件包的组成的方法以便使读者以最佳方法混合编程 C、C++ 和汇编语言。

本章将向读者介绍 Borland C++ 软件包及其所有组成部分，且读者将学会如何在计算机上装入和运行该软件包。这样，不管读者的程序是在具有 Borland C++ 3.1 编译程序的 DOS 环境下，还是在具有 Borland C++ 4.0 编译程序的 DOS 或 Windows NT 环境下，一个充满编程乐趣的新世界都将在等着读者。

1.1 BORLAND C++ 软件包的配置

当用户从包装盒中取出 Borland C++ 软件包时，可能会面对一大堆东西。但是，不要被这么多东西吓倒。先将说明书、磁盘和文献分开摆放。把磁盘和文献放在一边，先检查一下说明书。用户可将这一堆说明书分成两类：用户指南和参考指南。用户指南中含有软件包的每个组成部分的安装指导（本章后面将讲述安装）。其中还含有关于每个组成部分的特征的详细信息。另一方面，参考指南中含有每种语言的参考信息。其中包括函数调用及其它助记符的详细信息。切勿丢失这些说明书。

这样，便可将说明书分成三类：C++、汇编语言和调试程序/剖析程序。下面几节将解释这三种产品是什么，能干什么用，以及如何使用它们。

1.1.1 C 和 C++ 编译程序

用户可使用 C++ 软件包为 Windows 3.1 或 Windows NT 编写标准的且是独立的 C 或 C++ 程序或者应用程序。C 语言可提供结构化、模块化且是可编译的通用的编程环

境。C 正在很快成为系统程序员和初学者所需的语言。用户装入 C++ 软件之后,便可选择是在集成开发环境下工作还是命令行方式下使用编译程序。

C++ 编译程序将程序代码转换为目标代码。当用户编写 C 或 C++ 程序时,源代码名应以 *.c* 或 *.cpp* 扩展名结束。下面是一些源代码名的例子:

```
myfirst.c
mysecond.cpp
counter.c
multiplier.cpp
```

编译成功后,磁盘上将含有目标文件,例如:

```
myfirst.c
mysecond.cpp
counter.c
multiplier.cpp
myfirst.obj
mysecond.obj
counter.obj
multiplier.obj
```

目标文件中含有原始源代码翻译过来的计算机可直接读取的代码。要获得最终的可执行文件,应使用连接程序连接目标代码。连接程序设置程序执行时可执行代码所处的内存地址。连接程序通常产生带有 *.exe* 扩展名的文件。目标文件被连接后,磁盘中将含有下列文件:

```
myfirst.c
mysecond.cpp
counter.c
multiplier.cpp
myfirst.obj
mysecond.obj
counter.obj
multiplier.obj
myfirst.exe
mysecond.exe
counter.exe
multiplier.exe
```

用户在命令行提示符下键入可执行文件的名称便可执行该文件。读者从第 2 章"使用 Borland C++ 编译程序"中可学到输入和执行简单的 C 和 C++ 代码的方法。

1.1.2 汇编程序

用户可使用汇编程序软件包用汇编语言编写独立的程序。汇编语言是机器特定的

(微处理器),因此,它不如C代码的移植性好。Borland汇编程序允许用户为Intel系列的处理器芯片开发代码。这些处理器芯片当前包括8086, 8088, 80186, 80286, 80386和80486。

汇编语言是一种意义深远的编程语言,它允许用户编写与计算机的实际机器码联系最紧密的代码。尽管许多程序员认为汇编语言很难学且很难编程,但是汇编语言可向熟练的程序员提供速度上的优势和完全的硬件控制。用户可使用Borland编辑器编写独立的汇编语言程序,然后汇编和连接编写好的程序。

汇编程序将程序代码转换为目标代码。当用户编写汇编语言程序时,源代码的名称通常以.*asm* 扩展名结束。下面是一些汇编语言源代码的名称:

```
scrclear.asm
timer.asm
```

汇编成功后,磁盘中将含有目标文件,例如:

```
scrclear.asm
timer.asm
scrclear.obj
timer.obj
```

由汇编语言汇编得来的目标文件中也含有原始源代码翻译过来的计算机可直接读取的代码。要获得最终的可执行文件,应使用连接程序连接目标代码。汇编语言程序汇编后形成的目标文件被连接后,磁盘中将含有下列文件:

```
scrclear.asm
timer.asm
scrclear.obj
timer.obj
scrclear.exe
timer.exe
```

用户在命令行提示符下键入可执行文件的名称便可执行该文件。读者从第3章"使用Borland汇编程序"中可学到输入和执行简单的汇编语言代码的方法。

1.1.3 调试程序和剖析程序

当用户编写C,C++或汇编语言程序时,总会出现错误。用户可能的错误可被分为两大类:句法错误和编程错误。当用户编译或汇编程序代码时,句法错误被报告给用户。若错误足够严重,则编译程序或汇编程序将退出编译或汇编过程且不生成任何可执行文件。要纠正这类错误,用户应进入编辑程序并找到代码中标明出错的行。Borland参考指南没对这类错误作说明。句法错误通常是最容易确定的错误(编辑和纠正代码将在第2章中详细讨论)。

用户程序可能被正确地编译或汇编——不报告任何错误——但程序却不按用户的意图工作。若出现这种情况,则错误通常是编程错误——即用户的编程逻辑有错。编程错

此为试读,需要完整PDF请访问: www.ertongbook.com

误比句法错误难跟踪。Borland 公司的调试程序是为用户尽可能快地定位编程错误而设计的。要使用调试程序，用户必须拥有一个可执行程序，即使该可执行程序不能正确执行。读者将在第 4 章“使用 Borland 调试程序和剖析程序”中学到编写集成到调试程序环境中的程序的方法。

最后说一下，剖析程序是一种工具，它将帮助用户编写更有效的代码。通过使用剖析程序，用户将知道自己的程序在执行期间在何处最耗费时间。这便可帮助用户编写效率更高的 C 和 C++ 代码并可提供一个汇编语言例程来加速这种操作。读者将在第 4 章中学到使用剖析程序的方法。

1.2 在用户系统中安装 Borland C++ 软件包

用户的 Borland C++ 编译软件包的成功与否和使用的容易程度依赖于所装入的计算机系统和用户所选择安装选项。下面几节提供优化用户编程环境的方法。

1.2.1 用户计算机

Borland C++ 软件包可在 IBM 和 IBM 兼容计算机的一个很宽的范围内运行。Borland C++ 的使用说明书中将讨论最低的系统配置，但本书建议使用下列系统剖面：

80486 计算机
DOS 6.0 或 Windows NT 操作系统
对 DOS，需 2M RAM 内存；对 Windows NT 需 16M RAM 内存
彩色监视器(图形)
300MB 硬盘
鼠标

若用户系统由前面建议的剖面构成，则用户便可快速并有效地交换使用 Borland C++ 软件包的各种组成。

1.2.2 设置用户系统

将第一张盘插入磁盘驱动器并键入 install 便可开始安装。在安装的各步骤中，若用户不知道所需的另一个选项，则应接受建议的缺省值。若用户需要，则可修改这些缺省值，其中包括路径名。当用户安装 C++ 编译程序时，应确定需在用户环境中包括哪种内存模式。本书中的程序中仅使用小(“Small”)内存模式，若用户仅安装小内存模式，便可保留大量磁盘空间。若用户需要的话，可在以后添加其它内存模式。

一旦用户安装了每一个软件，都将返回到含有 C++ 编译程序的子目录中。

1.3 本书的组织

序言中已对本书的安排作了完整的描述。若读者的编程目的具有某种选择性，则下

面是本书建议读者读的章节：

快速使用每种软件：第 2 至第 4 章

只讲述汇编语言的章节：第 3、4、17 至 19 章

只讲述 C 和 C++ 的章节：第 2 章、第 4 至第 14 章

Microsoft Windows 3.1 开发：第 20 至第 25 章

Microsoft Windows NT 开发：第 26 至第 27 章

若读者希望尽可能快地开发 C 和 C++ 程序，则应去阅读第 2 章。若读者对汇编语言感兴趣，则应去阅读第 3 章。读者一旦读完了这些章节，便可立刻去阅读书中自己感兴趣的章节，继续提高自己的编程技巧。

第 2 章

使用 Borland C++ 编译程序

用户所购买的 Borland C++ 编译程序是一种强有力的软件开发环境。这一软件允许用户做下述范围内的任何工作：从检查某一个变量的内容或调用堆栈中传递给某个函数的值，到单步执行用户程序或跳转至某一特定的子例程执行。

读者将从本章中学到使用这一编程环境中的许多开发节约时间特性的方法以及使用仅适用于 C 语言的上述特性的方法。读者还可学到许多详细的附加编译操作，读者以后将用到这些操作。如果不特别注明的话，书中用到的所有引用都适用于 DOS 环境下的 Borland C++ 3.1 编译程序和 DOS 和 Windows NT 环境下的 Borland C++ 4.0 编译程序。

本章将使用简单的程序段来解释对管理多数编程问题来说是绝对必要的特性和实用程序。

2.1 主窗口

欢迎进入这一新的窗口世界。相对于用户原先使用的编译程序，用户一看到 Borland 编译程序的集成环境可能会很兴奋。这一编译程序具有窗口、控制、滚动栏、功能键以及许多菜单(参见图 2-1)。

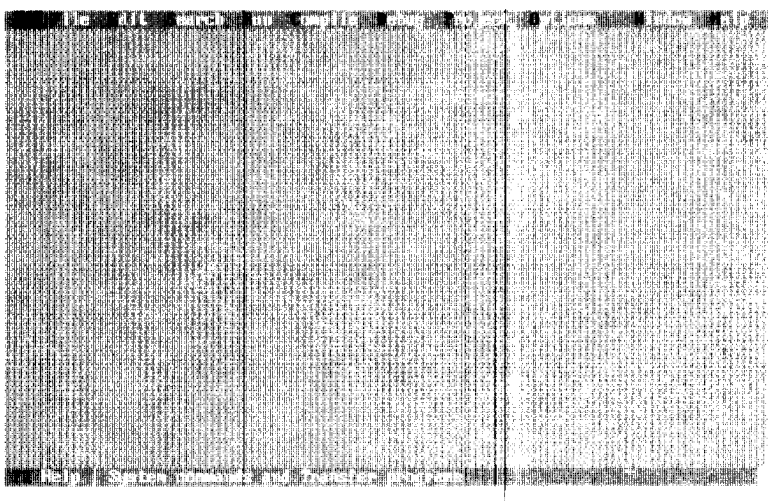


图 2-1 Borland C++ V3.1 的主窗口

横跨窗口顶部的是主菜单项："≡"(系统菜单)，"File"，"Edit"，"Search"，"Run"，"Compile"，"Debug"，"Project"，"Options"，"Window"和"Help"。该窗口的底部(Edit 方式下)提示各种功能键的用法：F1-Help(当前是活动的)，F2-Save，F3-Open，ALT-F9-Compile，F9-Make，F10-Menu。

了解几个基本要领后，这些菜单的使用就不复杂了。用户可用任何下列三种办法之一访问一个主菜单项：用户可用鼠标单击该选择项；用户可按下 F10 键后使用左右箭头键加亮该选择项并按下 ENTER 键；或者用户可按住 ALT 键不放按下用户要激活的选择项中的用彩色表示的字符。例如，用户按 ALT-F 可选择"File"选择项(ALT-空格将选择"≡"系统菜单)。

2.2 帮助(Help)

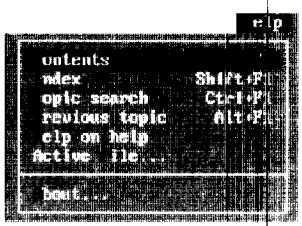
当所有操作都失败时，用户可按 F1 键请求帮助。要在编程环境中的任何地方获取上下文相关的帮助信息则只需选择所希望的菜单项并按下 F1 键。读者从下文中将读到这种特征的详细的解释。只需按下 ESC 键便可退出 Help 窗口(多数环境下，ESC 键将取消许多不同的操作)。

根据所涉及的帮助的主题，用户可能被提示转向一个附加的 Help 窗口。按 ALT-F1 键可退回原 Help 窗口。顺便提一下，假设用户正使用编辑器且嵌套了几层 Help 窗口，并且用户将不再使用 Help，则可按 ESC 键退出 Help 实用程序。用户按 ALT-F1 而不按 F1 键便可返回上一次显示的 Help 窗口。Help 实用程序在许多环境下都非常完整，因此，用户不必再去参考《Borland C++ 用户指南》。

另外，若用户正处于 Edit 窗口中，则用户可使用 Help 获取与关键语言特征相关的信息。把光标放到一个 C 关键字上并按下 CTRL-F1(或把鼠标定位在关键字之上且单击鼠标右按钮)便可获取对选中项的句法和用法的简要解释。

若用户不知道该怎么做，则应按两次 F1 键查看 Help 用法的一个 Help 窗口。图 2-2 示出了这一 Help 窗口的第二页，该窗口列出了可得到的帮助的目录。

用户还可从 Main 菜单中通过选择"Help"选项来请求帮助。列出各种选项以及与选项作用相同的热键的菜单如下所示：



"Contents"选项(图 2-3)示出了主帮助菜单的目录，而"Index"选项(图 2-4)使用户可访问可索引的 204 个窗口帮助文件。对其中的最后一个选项，用户只需加亮所希望的特征或命令并按下 ENTER 键，或双击鼠标左按钮以获取所需的帮助信息。

注意，用户可按下 ALT-F1 键把前一次选中的 Help 窗口放回前台。用户可使用这一

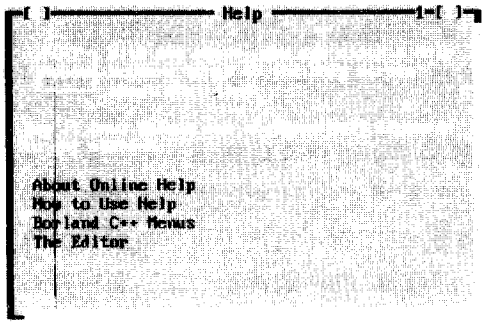


图 2-2　列出帮助的主题的 Help 窗口

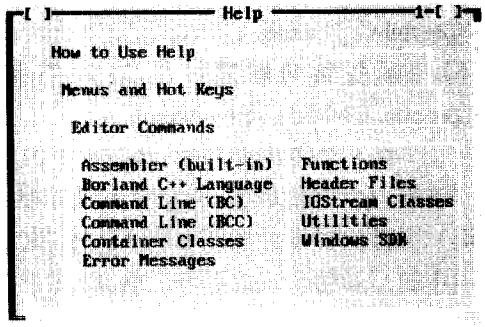


图 2-3　Help 的上下文表

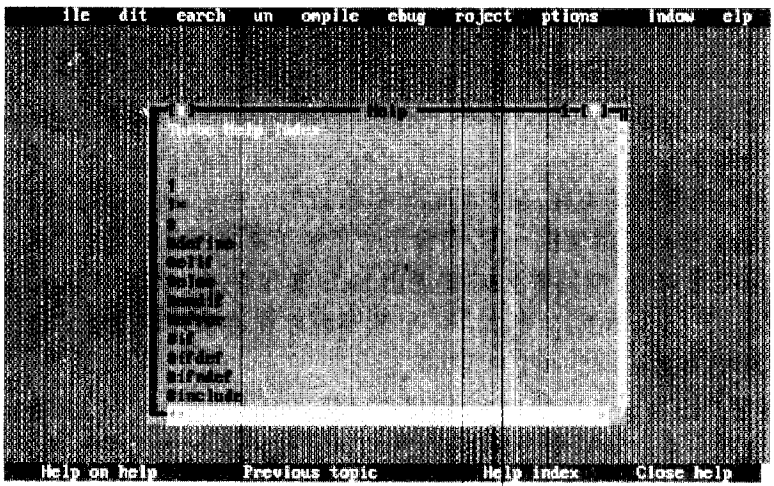


图 2-4　Help 目录

键组合来一步步地从整个编辑会话的帮助申请返回。

2.3　用户的第一个程序

本节将向读者介绍输入、编辑、编译、运行、调试、存盘以及重新装入一个简单程序的必要的基础。但在用户输入自己的第一个程序之前，还需做一些必要的设置。

2.3.1　最低设置

用户可很快地将 Borland C++ 编译程序装入到一个硬盘子目录中。现在的问题是把文件放到何处。若用户拥有一个硬盘，则用户可能希望将文件放到该硬盘上(可能放到编译程序所在的同一个目录，也可能放在其它地方)。用户也可能希望把自己的程序文件存到软盘上以便将文件从办公室带回家。

定义输出文件的一种简单方法是在主菜单中使用 ALT-O 选项，然后为“Directories”

选项敲入 D。用户既可在标有“Output Directory”的框内单击鼠标，也可用 tab 键转换到该框，并输入自己的输出文件的位置。例如，若用户正在开发一个工资报表程序，则用户可能会希望将该文件存到 A 驱动器上的 PAYROLL 目录(参见图 2-5)。

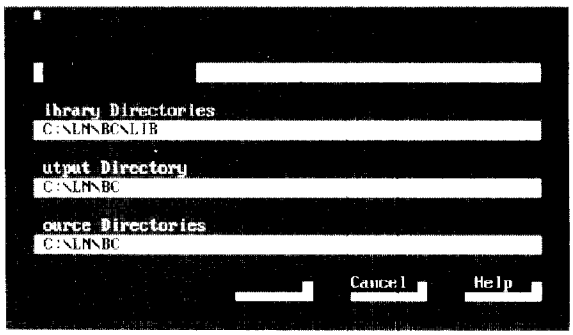


图 2-5　Borland C++ V3.1 的 Directory 窗口

如果目前用户希望把自己的所有文件放到 A 驱动器的 PAYROLL 目录，则用户在按下 ESC 键离开 Directories 子菜单之前应确保已存盘了所定义的内容。用户通过敲入 S 选择“Save”选项完成这一操作。

当然，用户每次创建新文件时，可指定选择“Save”(F2)选项时文件的保存位置。用户可通过按下 F2 键然后将文件命名为 A:\PAYROLL\PAYROLL.C 完成同一文件的存储定位。

2.3.2　创建

若用户是第一次在 DOS 环境下使用 Borland 编译程序，则用户通过敲入 BC 启动该编译程序时将看到类似于图 2-1 中示出的一个窗口。另一种方法是在 Windows 3.1 或 Windows NT 下选择适当的图标来进入该集成环境。用户显示器上的最大一部分将是白色块形模式。这部分是用户的工作区并且此区域当前是空的。

不论用户所创建的是第一个程序，还是用户已具有使用几个程序的经验，创建一个新文件都只是在“File”选项上单击鼠标，然后再在“New”选项上单击鼠标。注意，此时该工作区变成带有一个亮边界的实的背景(参见图 2-6)。

为获得使用编辑器的经验，用户可输入下列 C 程序，包括其中的错误：

```
/* A simple demonstration program */
#include <stdio.h>

#define SIZE 5

void print_them(int index,char continue, int int_aray[SIZE]);

main()
{
  int index;
  int int_aray[SIZE];
```


此为试读，需要完整PDF请访问：www.ertongbook.com