

Borland C++实用编程指南

(京)新登字 161 号

内 容 提 要

本书主要讲述 Borland C++ 编译软件包,包括 C++ 编译器、汇编程序、调试器、剖析器,同时还讲述了 C++ 和汇编语言混合编程方法,以及如何用 Borland C++ 编写 Windows 应用程序。本书从 C、C++ 和汇编语言基础讲起,对拥有最新 Borland C++ 软件包的初学者或熟练程序员,以及想进一步学习面向过程和面向对象编程知识的人来说,都有很大的参考价值。本书内容全面、实例丰富,并指导程序员如何克服在编程时遇到的各种问题。读完本书后,能熟练使用 C++ 软件包编写、调试复杂的 C、C++ 和汇编语言程序。本书力求通俗易懂,为进一步的研究打下基础。

图书在版编目(CIP)数据

Borland C++ 实用编程指南/森林编. —北京:航空工业出版社,1994.9

ISBN 7-80046-815-1

I. B... I. 森... III. C 语言—程序设计 IV. TP312C

中国版本图书馆 CIP 数据核字(94)第 12218 号

航空工业出版社出版发行

(北京市安定门外小关东里 14 号 100029)

北京地质印刷厂印刷

全国各地新华书店经售

1994 年 10 月第 1 版

1994 年 10 月第 1 次印刷

开本:787×1092 1/16

印张:39.25

字数:952 千字

印数:1—2000

定价:49.00 元

0000158

目 录

第一章 概述	(1)
1.1 Borland C++ 软件包的配备	(1)
1.2 系统安装	(3)
第二章 Borland C++ 编译器	(5)
2.1 主窗口	(5)
2.2 帮助	(6)
2.3 第一个程序	(7)
2.4 安排窗口.....	(15)
2.5 多个源文件管理.....	(16)
2.6 其它菜单项.....	(19)
第三章 Borland 汇编	(24)
3.1 建立 Assembler	(24)
3.2 汇编过程:第一个例子	(25)
3.3 Assembler 任选项和开关	(28)
3.4 连接程序的任选项和开关.....	(29)
3.5 重要的应用程序和文件.....	(30)
3.6 汇编过程:第二个例子	(35)
3.7 汇编模式:MASM 和 IDEAL	(39)
3.8 混合模式:第三和第四个例子	(39)
3.9 跟踪汇编语言错误:第五个例子	(43)
3.10 补充说明	(45)
第四章 Borland 调试器和剖析器	(46)
4.1 调试器——寻找错误.....	(46)
4.2 剖析——效率专家.....	(46)
4.3 调试器.....	(46)
4.4 剖析器(Profiler)	(57)
4.5 程序开发步骤.....	(61)
第五章 C 和 C++ 基础	(63)
5.1 C 语言的历史	(63)
5.2 ANSI C 标准	(68)
5.3 C++ 的发展和面向对象的程序设计	(69)

5.4	C++的历史	(69)
5.5	C程序的基本元素	(73)
第六章	数据	(82)
6.1	什么是标识符	(82)
6.2	关键字	(83)
6.3	标准C和C++数据类型	(84)
6.4	存取修正符	(90)
6.5	pascal, cdecl, near, far, huge	(92)
6.6	数据类型转换	(94)
6.7	存储级别	(96)
6.8	操作符	(99)
6.9	了解操作符的优先级	(106)
6.10	标准C和C++库	(106)
第七章	控制	(110)
7.1	条件语句	(110)
7.2	循环语句	(122)
第八章	函数	(137)
8.1	函数形式和原型	(137)
8.2	函数参数	(141)
8.3	函数类型	(147)
8.4	main的函数参数	(151)
8.5	C++特殊功能	(155)
8.6	涉及作用域的编程问题	(157)
第九章	数组	(161)
9.1	什么是数组	(161)
9.2	数组的基本性质	(161)
9.3	定义数组	(161)
9.4	数组初始化	(162)
9.5	使用数组下标	(164)
9.6	在数组中使用 size of	(166)
9.7	数组边界检查	(167)
9.8	数组与字符串	(168)
9.9	多维数组	(170)
9.10	数组与函数	(172)
9.11	使用数组的字符串函数	(177)

第十章 指针	(182)
10.1 何谓指针变量.....	(182)
10.2 函数指针.....	(197)
10.3 动态内存分配.....	(200)
10.4 指针与数组.....	(204)
10.5 C++引用类型	(213)
第十一章 C和C++中的输入输出	(215)
11.1 C中的输入输出	(215)
11.2 C++中的输入输出	(233)
11.3 高级C++输入输出	(241)
第十二章 结构、联合与杂类	(255)
12.1 结构.....	(255)
12.2 联合.....	(269)
12.3 杂类.....	(271)
12.4 链表.....	(273)
第十三章 类	(278)
13.1 基本的类概念.....	(278)
13.2 操作符重载.....	(296)
13.3 派生类.....	(298)
第十四章 面向对象程序设计介绍	(303)
14.1 C++与面向对象程序设计	(303)
14.2 面向对象程序设计的定义及观点.....	(303)
14.3 开发一个面向对象的链表程序.....	(304)
第十五章 开发重要的C和C++库	(321)
15.1 C和C++的头文件	(321)
15.2 标准库函数.....	(322)
15.3 字符函数.....	(328)
15.4 内存和字符串函数.....	(332)
15.5 数学函数.....	(338)
15.6 时间函数.....	(343)
第十六章 系统资源及图形	(349)
16.1 BIOS头文件	(349)
16.2 DOS头文件	(351)
16.3 图形头文件.....	(357)

第十七章 汇编语言	(374)
17.1 算术例程.....	(374)
17.2 使用查找表.....	(388)
17.3 BIOS/DOS 系统中断和寻址端口的使用.....	(390)
第十八章 强有力的程序设计手段—宏指令和过程	(401)
18.1 宏指令.....	(401)
18.2 过程.....	(410)
18.3 目标模块库.....	(417)
18.4 宏指令、过程和库的比较	(419)
第十九章 C 和汇编语言代码混合编程	(421)
19.1 使用插入式汇编语言.....	(421)
19.2 编写独立的 C 和汇编语言模块	(425)
19.3 一个简单的 C 和汇编语言连接的例子	(427)
19.4 使用 C++ 和汇编语言的一个硬件接口	(430)
19.5 从 C 向汇编语言传递数组	(433)
第二十章 Windows 基本概念	(436)
20.1 什么是 Windows	(436)
20.2 Windows 功能	(436)
20.3 Windows 特性	(439)
20.4 Windows 概念和术语	(440)
20.5 创建 Windows 程序的步骤	(450)
第二十一章 用 Borland C++ 编写 Windows 应用程序	(451)
21.1 为什么要使用简易 Windows 平台(SWP)模板	(451)
21.2 过程的编译和连接.....	(452)
21.3 简易 Windows 平台	(452)
21.4 每个 SWP 中所用到的重要特性	(464)
21.5 小结.....	(473)
第二十二章 使用 Borland C++ 资源库和资源编译器	(474)
22.1 Windows 资源	(474)
22.2 使用 BRW	(475)
22.3 在命令行使用资源编译程序(RC)	(488)
22.4 附加资源信息.....	(491)
第二十三章 用 Borland C++ 开发质量图表	(492)

23.1	调色板管理程序.....	(492)
23.2	在应用程序中使用字体.....	(494)
23.3	条状图表.....	(496)
23.4	使用 Windows 调试程序(Debugger)	(513)
第二十四章	面向 ObjectWindows C++ 程序开发的模板	(515)
24.1	ObjectWindows 的三个面向对象的特性	(515)
24.2	ObjectWindows 对象	(516)
24.3	SWPO.CPP 模板	(518)
24.4	试验 SWPO.CPP 模板	(523)
24.5	更高级的工作.....	(528)
第二十五章	开发与资源相关的 C++ ObjectWindows 应用程序.....	(529)
25.1	draw25,创建用户图标、光标、菜单和键盘加速器组	(529)
25.2	Pie25,含有用户图标、光标、菜单和两个对话框的高质量表示的图形	(538)
附录 A	扩展 ASCII 表	(550)
附录 B	DOS 10H、21H 和 33H 中断参数	(553)
B.1	用 BIOS 10H 中断进行屏幕控制	(553)
B.2	CRT 界面控制	(553)
B.3	字符操作	(554)
B.4	图形界面	(555)
B.5	ASCII 电传打字机输出	(555)
B.6	DOS 21H 中断的分类和要求	(556)
B.7	中断 33H 的 MOUSE 控制功能.....	(562)
附录 C	Windows 函数.....	(567)

第一章 概述

当你准备编制程序时,可能会碰到以下问题:有两种工具可以选用,一种是通用工具,另一种是专用工具,究竟采用哪一种好呢?一般来说,专用工具适合于特定的应用场合,但从长远观点来看,其使用性不易扩展;而通用工具往往做得很好,还包括有各种附加功能,能经得起繁重任务的考验。

Borland C++软件包括一些方便灵活的编程工具,即使是编程新手也可以使用 Borland C++软件包。而当编程技巧提高后,该软件包仍然不会过时。

本书完成以下三件事:第一,帮你理解 Borland C++软件包各个部分的定义和使用方法;第二,它是一完整的 C、C++和汇编语言的文档,可以指导你学习用 C、C++和汇编语言编写程序的方法和技巧。当读完本书时,对上述每种语言的掌握水平,你将从一初学者变为一中级水平程序员;第三,该书讲述了如何结合编译器的各部分,将 C、C++和汇编语言组合进最后完成的程序。

本章介绍 Borland C++软件包的各个部分,从中可以找到如何在计算机上安装运行该软件包的方法。

1.1 Borland C++软件包的配备

当你第一次打开 Borland C++软件包装盒时,可能会感到不知所措。不用着急,只要将手册、软盘和文档分成不同的堆。将软件和文档放在一边,检查手册堆。手册堆分两类:用户手册和参考手册。用户手册介绍了如何安装软件包的各部分(有关安装将在本章的后面部分介绍)。其中也包括各部分特点的详细介绍。参考手册包括各种语言的参考,其中有函数调用和其它功能的详细介绍。

将手册分成三部分:C++、汇编语言和调试工具。下面解释它们分别做什么用以及如何用。

1.1.1 C和C++编译器

用 C++软件包工具可以写出独立的 C 或 C++程序。由于 C 语言提供了通用的结构化、模块化编译程序的环境,很快变成了系统程序员和初学者必不可少的语言。安装 C++软件后,既可以在集成开发环境中工作,也可以在命令行上使用编译器。

C++编译器将程序源代码转换成目标代码。编写 C 或 C++程序时,源文件名通常以扩展名.C 或.CPP 结束。下面是一些源文件名的例子:

```
tfirst.c
```

```
tsecond.cpp
```



```
counter.c
multiplier.cpp
```

编译成功后,盘上可能也包括目标文件。例如:

```
tfirst.c
tsecond.cpp
counter.c
multiplier.cpp
tfirst.obj
tsecond.obj
counter.obj
multiplier.obj
```

目标代码由源代码转换得到,计算机可直接读。为了得到最后的可执行文件,用 Linker (连接程序)来连接目标代码。Linker 设置在执行程序时可执行代码装入内存的地址。Linker 生成的文件扩展名一般为 .exe。连接了目标文件后,盘上包括下列文件:

```
tfirst.c
tsecond.cpp
counter.c
multiplier.cpp
tfirst.obj
tsecond.obj
counter.obj
multiplier.obj    tfirst.exe
tsecond.exe
counter.exe
multiplier.exe
```

可以在命令行上输入执行文件名来执行一程序。在第二章“Borland C++编译器”中,你将学会如何输入并执行简单的 C 和 C++ 程序。

1.1.2 汇编

可以通过 Assembler 用汇编语言写出独立的程序、汇编语言是机器级语言,因此不如 C 语言简便。Borland Assembler 允许开发使用 Intel 系列处理器芯片代码,包括当前的 8086, 8088、80186、80286、80386 和 80486。

汇编语言写出的程序代码与实际的计算机机器指令相对应。尽管大多数程序员觉得汇编语言难学、编程复杂,但它为熟练的程序员提供了一种提高操作速度和直接控制硬件的较好方法。可以用 Borland 编辑器编出独立的汇编语言程序,然后加以汇编和连接。

Assembler 将程序源代码转换成目标代码。在编写汇编语言程序时,源文件名通常以 .asm 扩展名结束,下面是一些汇编语言源文件名的示例:

```
srclear.asm  
timer.asm
```

编译成功后,盘上也包括目标文件,例如:

```
srclear.asm  
timer.asm  
srclear.obj  
timer.obj
```

由汇编语言源文件汇编得到的目标文件计算机可直接读。为了得到最后的可执行文件,应用 Linker 来连接目标代码,连接完成后,盘上包含下列文件:

```
srclear.asm  
timer.asm  
srclear.obj  
timer.obj  
srclear.exe  
timer.exe
```

可以在命令行上输入执行文件名来执行该程序,在第三章“Borland 汇编”中你将学会如何输入并执行简单的汇编程序。

1.1.3 调试工具

当编写 C、C++ 或汇编语言程序时,很可能会出错,可能犯的错误分两类:语法错误和程序执行期间的错误。语法错误是编译或汇编时出现的错误。如果错误特别严重,则退出编译程序,不产生可执行文件。为了修正这类错误,可进入编辑器,找到出错的行。Borland 参考手册不论述这个问题。语法错误通常很容易找出(关于编辑和源代码修改的详细讨论在第二章)。

程序可能在编译时没有错误出现,但仍不能正常运行。出现这种情况,说明你的程序有逻辑错误,因而引起运行错误。运行错误比语法错误难查。Borland 调试器能帮你尽快找到出现运行错误的地方。要想使用调试器,必须有可执行程序,不管该执行程序能否正常运行。你将在第四章“Borland 调试工具”中学会如何使用 Borland 集成调试环境来编写程序。

另一调试工具—剖析器(Profiler)可以帮助写出更有效的源代码。通过剖析器,可以找出程序在执行过程中各部分所需的执行时间,用来帮助写出高效率的 C 和 C++ 源代码。这样可以编一汇编子程序来代替较费时间的操作。在第四章中将学会如何使用剖析器。

1.2 系统安装

能否成功、方便地安装 Borland C++ 编译软件包,在很大程度上依赖于计算机系统和你的安装选择。下面介绍如何优化编程环境。

1.2.1 所用的计算机

Borland C++ 软件包适用于 IBM 及与其兼容的各类微机。Borland C++ 的 GettingStarted 手册讨论了所需的最少系统。但本书建议使用下列系统配置：

PC XT/AT、286、386、486 主机

DOS5.0 以上操作系统

4M RAM 内存

彩色显示器(图形)

协处理器芯片

120MB 硬盘

鼠标器

如果你的计算机系统包括上面提到的配置,就可以在 Borland C++ 软件包各部分之间快速转换。

1.2.2 建立系统

开始安装系统时,将第一张盘放入驱动器中,并敲入 install,在安装过程中,尽量使用缺省值,除非需要别的选择项,如果必要的话,你可以在以后改变这些缺省值,如路径名等。在安装 C++ 编译器时,需要确定使用哪一种内存模式。本书的程序只使用小内存模式。如果仅仅安装小内存模式,可以留出大量的磁盘空间。如果必要的话,可以在后来加入其它内存模式。

安装完毕,返回到包含 C++ 编译器的子目录下,敲入 BC 进入集成环境。

第二章 Borland C++ 编译器

Borland C++ 编译器是一个强大的软件开发环境,利用它可以完成各种工作:检测单个变量的内容,或在函数调用中传递给函数的值,程序的单步运行或者跳转到一特定子程序。

在本章中,将学会如何利用 C 语言所独有的特点在程序环境中节省开发时间,还能学会以后会用到的很多附加的编译器操作。

本章用简单的程序段来解释该编译器的特点和一些实用例子,这在以后解决编程问题时是绝对有用的。

2.1 主窗口

Borland C++ 编译器的集成环境非常漂亮,编译器有窗口、控制键、滚动条、功能键,还有很多菜单(见图 2-1)。

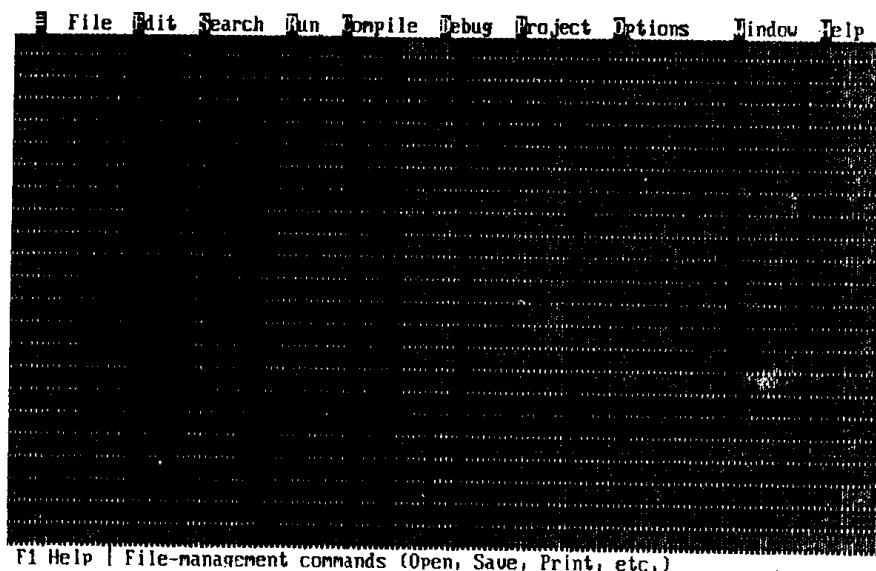


图 2-1 主窗口

窗口的顶部是主菜单选择项“三”(系统菜单),“File”、“Edit”、“Search”、“Run”、“Compile”、“Debug”、“Project”、“Options”、“Window”和“Help”。窗口的底部(在编辑模式下)以高亮度显示了不同功能键的用法:F1—Help(当前一直处于激活),F2—Save、F3—Open、ALT + F9—Compile、F9—Make 和 F10—Menu。

学习一点基础知识你会感到更清楚一些。可以用下列三种方法之一来访问主菜单的选、

择项:在选择项上点一下鼠标;按 F10,再用向左或向右光标键移到选择项上并按回车;同时按下 ALT 键和你要激活的选择项的以彩色显示的字母键,例如,按 ALT+F 可以选择 File 项(ALT+空格键选择“三”系统菜单)。

2.2 帮助

有问题时,可以按 F1。在编程环境中,选择你所希望的选择项并按 F1 键可以得到关于它的帮助信息,其中有详细解释。按 ESC 键可以从帮助窗口中退出(在大多数情况下,ESC 键可以撤消各种各样的操作)。

根据不同的帮助项目,可能会提示用户进入一附加的帮助窗口。为了返回前面的帮助窗口,按 ALT+F1。需要说明的是,如果你正在使用编辑器,并且有多层帮助窗口嵌套,这时按 ESC 则退出帮助。如果不是按 ESC 键而是按 ALT+F1 键,则只返回到上一级帮助窗口中。帮助包括的内容很全面,在很多场合下,几乎不需再参考 Borland 的 C++ 用户手册。

另外,如果你位于编辑窗口中,可以用帮助功能快速得到关于编程语言的解释。将光标移到 C 关键字处并按 CTRL+F1(或将鼠标移到关键字处并按鼠标右钮),可以得到简短的语法解释和用途说明。

为了查看全部帮助,按两次 F1,这时显示一帮助屏,图 2-2 显示了该帮助窗口的第二页,其中列出了有效的帮助类目。

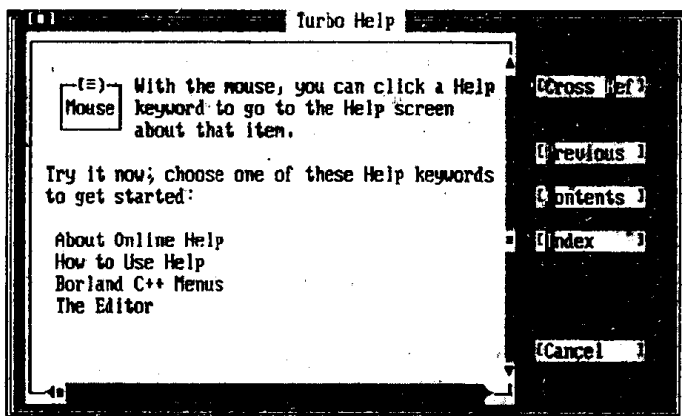


图 2-2 列出帮助项目的帮助窗口

也可以从主菜单中选择“Help”任选项来寻求帮助。Help 菜单列出了不同的选择项和它们的热键,如下所示:

Help	
Contents	
Index	Shift+F1
Topic search	Ctrl+F1
Previous topic	Alt+F1
Help on help	
Active file...	
About...	

2.3.1 最小设置

可以在一硬盘子目录中快速安装 Borland C++ 编译器,问题是在哪儿存放文件。如果有硬盘,可以把文件放在硬盘上(可能和编译器在同一子目录下,也可能在别处),也可以把程序存放在软盘上,这样就可以把它从办公室带回家。

可以用一个很简单的办法来定义输出文件的位置:从主菜单中用 ALT+O 任选项,敲入“D”选择“Directories”。在“Output Directory”框上点鼠标键,或用 tab 移到该框上,然后输入输出文件的位置。例如,如果你想开发一个工资单程序,可能希望将它存在 A 驱动器 PAYROLL 目录下(见图 2-5)。

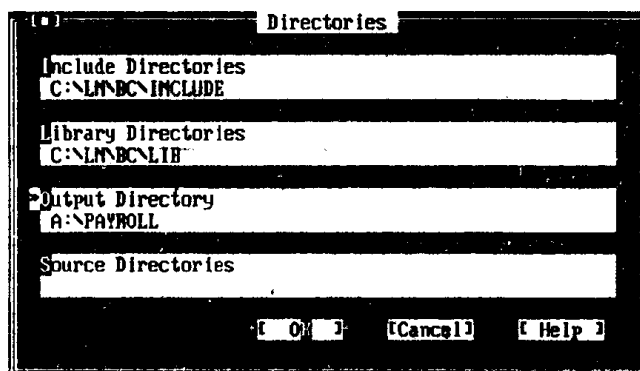


图 2-5 目录窗口

如果你总是想将文件存到 A 驱动器 PAYROLL 目录下,必须在按 ESC 退出 Directory 子菜单前将前面定的目录存起来(通过按“S”来完成存入功能)。

当然,在每次建立文件时,通过选择“Save(F2)”任选项可以指定该文件存到哪儿。按 F2 并且给出类似 A:\PAYROLL\PAYROLL1.C 的文件名,可以指定单个文件的存放位置。

2.3.2 生成文件

如果你是第一次使用 Borland 编译器,当敲入 BC 进入编译器时,将看到和图 2-1 一样的屏蔽区。最大的显示区是一灰色方块,它是工作区,当前是空的。

无论这是你第一次编程还是你已试过几个现有程序。开始一个新文件很简单。可以在字“File”上点一下鼠标,再在“New”上点一下鼠标即可。注意,现在工作区变成了一带亮边的整块(见图 2-6)。

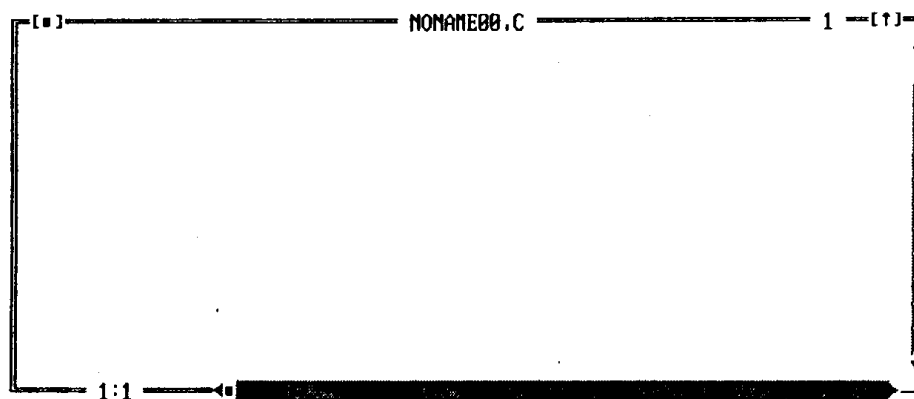


图 2-6 开始一新文件

为了练习使用编辑器,照下面完整输入 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_array[SIZE]);

main()
{
    int index;
    int int_array[SIZE];
    char continue=0;

    print-them(index, continue, int_array);

    printf("\n\nWelcome to a trace demonstration!");
    printf("\nWould you like to continue (Y/N) ");
    scanf("%c", continue);
    if(continue == 'Y')
        for(index=0; index < SIZE; index++) {
            printf("\nPlease enter an integer: ");
            scanf("%d", &int_array[index]);
        }
    print-them(index, continue, int_array);

    return(0);
}

void print-them(int index, char continue, int int_array[SIZE])
{
    printf("\n\n%d", index);
    printf("\n\n%d", continue);

```



```

for(index=0; index < SIZE, index++)
    printf("\n%d",int_array[index]);
}

```

如果你用过通用文字处理软件,会发现大多数编辑键(BACKSPACE、INS、DEL、PGUP、PGDN等)在 Borland C 和 C++ 编辑器中可同样使用。如果要移动到程序第一行,同时按 CTRL 和 PGUP 即可,同时按 CTRL 和 PGDN 可移动到程序最后一行。

也可使用窗口右边的滚动条。用鼠标点向上箭头符号▲将显示内容向后移动,点向下箭头符号将显示内容向前移动。滚动条上的■符号(不是“[■]”)指示当前屏幕相对整个长度上的文件的相对位置,将鼠标移到■符号的两边并按动鼠标键,能向前或向后移动一整屏,就象 PGUP 和 PGDN 键一样。用鼠标将■符号拖动到滚动条上的新的位置,可在整个文件长度上快速前后显示。

如果想使某一窗口占满整个屏幕,按 F5 或在[↑]箭头上点鼠标。反复按 F5 可在原始尺寸和满屏幕之间交替变换。在[■]符号处点鼠标可以关闭当前窗口。

编辑器有很多先进功能,按 F1 和 PGDN 可以查看这些功能。几种值得一试的功能是 Find(CTRL-QF)、Find&Replace(CTRL-QA)和块命令(CTRL-KB,开始;CTRL-K K,结束)。

用块命令来高亮度显示(选择)一部分代码,然后将它们移动(CTRL-KV)、拷贝(CTRL-KC)、删除(CTRL-KY)或写盘(CTRL-KW)。在一大型程序中,有些程序的一部分可能在别的程序中也需要,这时块写盘功能是很有用的。读入一存在盘上的块的方法很简单,只需将光标移到要插入的地方,并按 CTRL-KR(读盘),编辑器提示你输入文件名。在指定不在当前缺省目录下的文件时,不要忘记加入驱动器名和子目录名。

2.3.3 存盘

在编辑器中,千万不要丢失文件,为了存储一文件,按 F2 或在“Save”处点鼠标键。应该在每输入一定数量的内容后存储一次。

2.3.4 编译

为了编译一程序,按 ALT-F9 或在主菜单中 Compile 项处按鼠标再在 Compile 命令按鼠标。当按了 ALT+F9 后,出现一编译窗,显示编译的行数,同时也显示出现警告和错误信息的行数,正如下面显示的一样: