

计算机 C/C++ 语言系列丛书

# BORLAND C++

## 程序设计基础

郑雪明 编著

学苑出版社



计算机 C/C++ 语言系列丛书

# Borland C++ 程序设计基础

郑雪明 编著  
亦 鸥 审校

学苑出版社

1994

(京)新登字 151 号

## 内 容 简 介

本书是学习和使用 Borland C++ 进行程序设计的基础读物,书中介绍了进行 Borland C++ 程序设计的有关要点,并简要介绍了有关的函数。

本书对从事软件设计、开发和应用的技术人员具有重要的参考价值。

需要本书的用户,可与北京 8721 信箱联系,邮码 100080,电话 2562329。

计算机 C/C++ 语言系列丛书

**Borland C++ 程序设计基础**

---

编 著: 郑雪明  
审 校: 亦 鸥  
责任编辑: 甄国宪  
出版发行: 学苑出版社 邮政编码: 100036  
社 址: 北京市海淀区万寿路西街 11 号  
印 刷: 兰空印刷厂  
开 本: 787×1092 1/16  
印 张: 19 字数: 439 千字  
印 数: 1~5000 册  
版 次: 1994 年 5 月北京第 1 版第 1 次  
I S B N: 7-5077-0875-6/TP·24  
本册定价: 21.00 元

---

学苑版图书印、装错误可随时退换

# 目 录

<b>第一章 安装和启动 Borland C++</b> .....	1
1.1 运行安装程序 .....	1
1.1.1 安装 Borland C++ .....	1
1.2 使用集成开发环境(IDE) .....	2
1.2.1 启动 IDE .....	2
1.2.2 使用 IDE 菜单和窗口 .....	5
1.3 设置 Borland C++ .....	9
1.3.1 设置 IDE 选择项 .....	9
1.4 C 程序基本结构 .....	11
1.4.1 理解 C 结构 .....	11
1.4.2 使用编辑器编程 .....	16
1.5 库函数 .....	19
1.5.1 基本输入函数 .....	19
1.5.2 基本输出函数 .....	22
1.5.3 格式转换函数 .....	24
1.6 小结 .....	26
<b>第二章 C 语言基础</b> .....	27
2.1 基本概念 .....	27
2.1.1 源、目标和装载模块 .....	27
2.1.2 程序逻辑和运行流程 .....	28
2.2 基本数据类型 .....	34
2.2.1 C 基本数据类型 .....	34
2.2.2 定义数据对象 .....	36
2.3 C 表达式和语句 .....	39
2.3.1 C 运算符集 .....	40
2.4 类型转换 .....	43
2.4.1 隐式类型转换 .....	43
2.4.2 显式类型转换 .....	44
2.5 C 宏 .....	45
2.5.1 对象型宏 .....	46
2.5.2 函数型宏 .....	48
2.6 小结 .....	52
<b>第三章 C 函数</b> .....	53

3.1	main()函数和库函数 .....	53
3.1.1	main()函数 .....	53
3.1.2	库函数 .....	55
3.2	用户自定义函数 .....	57
3.2.1	函数说明与函数定义 .....	57
3.2.2	传递参数 .....	59
3.3	函数返回值 .....	62
3.3.1	定义和使用函数类型 .....	62
3.3.2	像数据对象一样使用函数 .....	64
3.4	存储类 .....	65
3.4.1	变量作用域 .....	65
3.4.2	变量生存期 .....	66
3.4.3	变量的连接 .....	67
3.5	高级程序控制逻辑 .....	68
3.5.1	循环控制语句 .....	68
3.5.2	改变程序流程 .....	70
3.6	可变参数表 .....	73
3.6.1	设计可变参数表 .....	73
3.6.2	使用 va_...()函数 .....	74
3.7	小结 .....	78
<b>第四章</b>	<b>使用指针和派生类型 .....</b>	<b>79</b>
4.1	理解 C 的标准派生类型 .....	79
4.1.1	C 类型的结构 .....	79
4.1.2	用旧的类型建立新的类型 .....	79
4.2	C 语言的指针 .....	82
4.2.1	理解间接地址 .....	82
4.2.2	C 的间接运算符和地址运算符 .....	87
4.3	使用数组和字符串 .....	90
4.3.1	定义和使用数组 .....	90
4.3.2	理解 C 的字符串 .....	92
4.4	使用结构和联合 .....	96
4.4.1	用不同类型建立结构 .....	96
4.4.2	联合是从另一角度观察的结构和对象 .....	99
4.5	函数指针 .....	101
4.5.1	说明和初始化函数指针 .....	101
4.5.2	引用指针调用函数 .....	102
4.6	用指针动态分配内存 .....	105
4.6.1	C 语言程序和动态内存 .....	106
4.6.2	使用动态内存 .....	107

4.7	小结 .....	113
<b>第五章</b>	<b>建立、编译和测试程序 .....</b>	<b>114</b>
5.1	程序由几个源文件组成 .....	114
5.1.1	确定源文件的内容 .....	114
5.1.2	设置 Borland C++ 的工程文件 .....	130
5.2	理解外部引用 .....	132
5.2.1	使用 extern 关键字 .....	132
5.2.2	使用外部函数 .....	133
5.3	为外部模块写头文件 .....	134
5.3.1	确定头文件中的内容 .....	135
5.3.2	引入用户提供的头文件 .....	135
5.4	在头文件中使用条件编译指令 .....	136
5.5	在集成开发环境下编译和运行文件 .....	138
5.5.1	编译和运行简单程序 .....	139
5.5.2	编译和运行复杂程序 .....	139
5.6	集成调试器 .....	141
5.6.1	使用 Step over 命令 .....	141
5.6.2	使用 Trace into 命令 .....	143
5.6.3	设置和取消断点 .....	143
5.7	小结 .....	145
<b>第六章</b>	<b>使用 I/O 库函数 .....</b>	<b>146</b>
6.1	输入/输出的概念 .....	146
6.1.1	文件和设备 .....	146
6.1.2	文件和流 .....	147
6.1.3	文件和二进制流 .....	149
6.2	使用标准流输入/输出 .....	150
6.2.1	使用格式化 I/O 函数 .....	150
6.2.2	scanf() 函数 .....	154
6.2.3	printf() 函数 .....	157
6.2.4	字符 I/O 函数 .....	160
6.3	文件控制函数 .....	164
6.3.1	打开、关闭和控制文件 .....	164
6.3.2	控制文件缓冲区 .....	167
6.4	直接文件 I/O 函数 .....	169
6.4.1	直接 I/O 概念 .....	169
6.4.2	读写直接文件 .....	171
6.5	文件定位函数 .....	177
6.5.1	获得当前文件位置 .....	177
6.5.2	设置新的文件位置 .....	178

6.6	处理输入/输出错误	180
6.6.1	检查文件 I/O 错误	180
6.6.2	显示和清除文件 I/O 错误	180
6.7	小结	181
<b>第七章</b>	<b>存储模式和 VROOMM 技术</b>	<b>183</b>
7.1	Intel 80x86 系列微处理器	183
7.1.1	段、节和偏移量	183
7.1.2	80286 和 80386 的一些特性	184
7.1.3	CPU 地址寄存器	185
7.2	near 指针、far 指针和 huge 指针	186
7.2.1	选择指针大小	186
7.2.2	near、far 和 huge 指针的说明	188
7.3	Borland C++ 的六种存储模式	190
7.3.1	选择存储模式	190
7.3.2	混合模式编程	191
7.4	生成可执行的 COM 文件	193
7.4.1	COM 文件	193
7.5	覆盖管理程序	195
7.5.1	覆盖管理程序	196
7.5.2	Borland 的 VROOMM 技术	196
7.6	设计和生成覆盖程序	198
7.6.1	决定覆盖模式	198
7.6.2	编译和连接一个覆盖程序	199
7.7	小结	201
<b>第八章</b>	<b>Borland C++ 的视频函数</b>	<b>203</b>
8.1	IBM/PC 文本模式	203
8.1.1	IBM/PC 视频适配器及监视器	203
8.1.2	屏幕存储映象输入输出	204
8.2	文本屏幕控制	205
8.2.1	文本模式控制函数	205
8.2.2	直接控制输入输出	208
8.3	窗口函数	209
8.4	IBM/PC 的图形模式	213
8.4.1	像素和调色板	213
8.4.2	图形屏幕的控制	216
8.5	BGI 图形库简介	217
8.5.1	画图函数和填充函数的使用	218
8.5.2	屏幕及视口控制	222
8.6	在图形模式下使用文本	223

8.6.1	BGI 字模 .....	223
8.6.2	图形模式文本函数的使用 .....	224
8.7	小结 .....	226
<b>第九章</b>	<b>Turbo Debugger 与 Turbo Profiler .....</b>	<b>228</b>
9.1	设置调试环境 .....	228
9.1.1	故障和测试程序 .....	228
9.1.2	从 IDE 中运行 Turbo Debugger .....	229
9.2	使用 Turbo Debugger .....	230
9.2.1	用 Turbo Debugger 运行程序 .....	230
9.2.2	观察数据 .....	234
9.3	高级调试功能 .....	235
9.3.1	改变程序的值 .....	236
9.3.2	设置断点 .....	237
9.4	使用 Turbo Profiler .....	238
9.4.1	什么是剖析 .....	238
9.4.2	启动 Turbo Profiler .....	239
9.4.3	了解基本的剖析信息 .....	240
9.5	高级剖析功能 .....	244
9.5.1	选择要剖析的区域 .....	245
9.5.2	显示剖析程序统计信息 .....	247
9.6	选择主动或被动剖析 .....	248
9.7	小结 .....	249
<b>第十章</b>	<b>Borland C++ 的扩充功能 .....</b>	<b>251</b>
10.1	在线汇编语言 .....	251
10.1.1	在线汇编环境 .....	251
10.1.2	关键词 ASM .....	253
10.2	汇编语言程序调用接口 .....	257
10.2.1	C 语言调用汇编语言程序 .....	257
10.2.2	汇编程序调用 C 函数 .....	268
10.3	中断函数 .....	276
10.3.1	80x86 中断系统 .....	276
10.3.2	Borland 的中断接口 .....	277
10.4	中断处理程序 .....	280
10.4.1	说明中断处理函数 .....	280
10.4.2	一个时钟中断服务程序 .....	283
10.5	小结 .....	286
<b>附录 A</b>	<b>ASCII 码表 .....</b>	<b>288</b>
<b>附录 B</b>	<b>printf() 和 scanf() 函数细节 .....</b>	<b>291</b>
<b>附录 C</b>	<b>exec...() 和 spawn...() 函数细节 .....</b>	<b>294</b>



# 第一章 安装和启动 Borland C++

本章将介绍在机器上安装 Borland C++ 的方法,还介绍了怎样利用 Borland C++ 的高性能。首先,本章介绍怎样安装 Borland C++ 及如何根据用户的不同需要来设置;其次,本章将展示给读者有关增强了的集成开发环境(IDE)的一些很有用的功能;最后,当读者对 IDE 已有初步的了解后,本章将讲述怎样编写一些简单的 C 程序。

## 1.1 运行安装程序

安装 Borland C++ 是一件很容易的事,Borland 提供了很完善的安装程序,减轻了用户安装大容量软件包的负担。在实际的安装过程中,用户需要做的所有事情就是告诉 INSTALL 程序自己的选择,在提示时插入需要的磁盘;其他的工作全部由 INSTALL 完成。

### 1.1.1 安装 Borland C++

安装 Borland C++ 前,用户需要检查一下机器硬盘上的可用空间容量。Borland C++ 是一个体积很大的软件包,当安装所有的选择项时,需要 28M 字节的硬盘空间。

开始安装时,把标有 INSTALLATION DISK 的软盘插入驱动器,并键入:

```
n: install
```

这里 n: 是插入 INSTALLATION DISK(安装盘)的驱动器。

屏幕首先给出欢迎使用安装程序,并提示全安装将需要 28M 字节的磁盘空间。第二幅屏幕给出提示,询问在哪个磁盘驱动器进行安装,这时,只需要键入正确的驱动器名并回车。

第三幅屏幕是 Borland C++ 安装程序的菜单。在这个菜单上,用户需要指定安装时挑选哪些选择项和文件在该盘的存储路径。图 1.1 是 Borland C++ 安装程序的主菜单。

为修改安装主菜单上的选择项,可以用箭头键来移动屏幕上的亮度行到需要的选择项时,键入回车。这时机器将就是否对菜单作出修改给出提示。

在主菜单中,第一项用来指定 Borland C++ 程序及支持文件的存放路径,这是件很容易的事。若想使安装程序把 Borland C++ 存贮于缺省路径,只需要直接键入回车,也可以指定文件存储的路径。

安装程序同时要求就是否需要程序示例文件做出选择。这些文件是 Borland 中示范 C++ 用法的一些程序。若硬盘空间有限或不需参考这些示例文件,可以选择不把它们拷贝到硬盘中。

安装程序最重要的部分之一是选择用户需要的内存模式。假若磁盘上有足够的空间,可以使用缺省选择项并设置成全部内存模式。但这样做,往往占用了实际并不使用的空间。在选择时,用户可以指定成所需要的某种内存模式。表 1.1 归纳了每一种内存模式的基本特征。

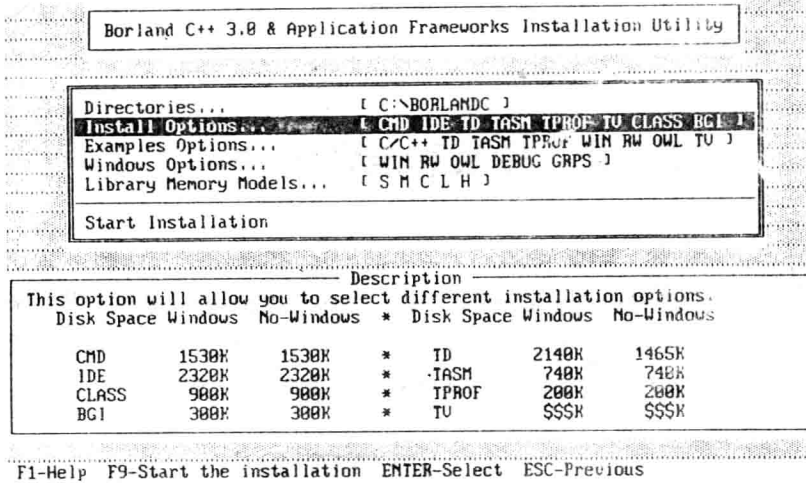


图 1.1 安装程序主菜单

表 1.1 内存模式说明

内存模式	特 性
Tiny(微型)	64K 空间被用于代码段、数据段和堆栈段。同时使用近指针。
small	64K 空间用于代码段,另 64K 空间用于数据段和堆栈段。使用近指针。
Medium(中型)	1M 空间用于代码段,另 64K 空间用于数据段和堆栈。代码段可以使用远指针,但数据段和堆栈段只能使用近指针。
Compact(紧凑型)	64K 空间用于代码段,1M 空间用于数据段和堆栈段。代码段使用近指针,数据段使用远指针。
Large(大型)	1M 空间用于代码段,另 1M 空间用于数据段和堆栈段。使用远指针。
Huge(巨型)	数据段、代码段都使用远指针,常规的静态数据存取的 64K 距离限制被超越。

当安装程序菜单的所有项目被设置以后,就可以开始真正的安装过程。这时用户需要做的所有事情就是在机器提示时插入软盘,安装程序会做好其余的一切事情。

安装完成后,需要修改 AUTOEXEC. BAT 文件中的 PATH 命令和 CONFIG. SYS 文件中的 FILES 命令。例如,用户需要在当前 PATH(路径)命令中增加以下驱动器和路径参数:

```
PATH = C:\Borland C \ bin
```

在 CONFIG. SYS 文件中,需要增加以下 FILES(文件)命令:

```
FILES = 20
```

重新启动计算机后,用户修改的 AUTOEXE. BAT 和 CONFIG. SYS 文件开始生效,这时,只要在 DOS 提示符下键入 bc,就可以运行 Borland C++了。

Borland C++ 3.0 特别设计成可以在 Microsoft Windows 下运行。为方便用户使用 Borland C++3.0 的最新版本。当下次用户运行 Windows 时,安装程序将自动生成程序组,Borland C++的全部程序和用户选择的公用程序都将包括在这个程序组内。

在 Windows 下运行 Borland C++ 是很容易的。当下次运行 Windows 时, 机器将给出选择, 提示是否建立 Borland C++ 程序组。只要简单地回答“yes”, 在安装时所选择的全部 Borland C++ 程序将被置入一个新的 Borland C++ 程序组中, 这样, 就可以很方便地在 Windows 下运行 C++。

## 1.2 使用集成开发环境(IDE)

用户拥有了新的集成开发环境(IDE), 就可以在一个方便的环境下完成所有的编程工作。IDE 集成了一个高产高效率程序员在开发软件时所必须的各种工具。本节将向读者介绍集成开发环境, 并在怎样启动和如何使用 IDE 的特性方面提供帮助。

### 1.2.1 启动 IDE

启动 Borland C++ 是很简单的, 只需要把目录转到 Borland C++, 在 DOS 提示符下键入 bc 就行了。只用这个命令启动 Borland C++ 后, 效果很好, 大部分需要都能被考虑。但使用了命令行选择项后, 用户程序和工程文件就会更快地调入内存, 使 IDE 更加高效地使用系统。

启动 Borland C++ 加命令行选择项的句法如下:

```
bc 源名(可选) | 工程文件名(可选) 方式选择(可选) ...
```

源名可以是任何 ASCII 码文件。通常, 源名是用户需要设计的 C 或 C++ 程序名; 工程文件是当前使用的工程文件名, 同时, 它必须具有扩展名 PRT。可以指定源名和工程文件名, 但二者不能同时指定。可供使用的选择项有 /b、/d、/e、/l、/m、/p、/rx 和 /x。下面将分别解释这些选择项:

#### 1. /b (建立所有文件)

/b 选择项使 Borland C++ 可以重新编译和连接工程中所有的文件。在编译和连接的过程中, 所有可能产生的错误信息都被送往标准输出设备。当编译和连接完成后, Borland C++ 中止并返回 DOS 提示符状态下。

为决定建立什么样的工程文件, Borland C++ 首先查看命令行参数, 然后查看调入编辑器的当前文件。Borland C++ 先检查命令行参数以得知是否指定了文件或工程文件名, 若指定了其中一个, Borland C++ 就对其进行编译和连接。若命令行未包含文件名或工程名, Borland C++ 就对当前调入编辑器的文件进行编译和连接。

#### 2. /d (双监视器方式)

使用 /d 选择项时, Borland C++ 工作在双监视器方式下, 这时可以同时使用两个监视器, 但它们必须工作在不同的视频方式下, 例如, 其中一个工作在彩色方式下, 另一个工作在单色方式下。为运行双监视器方式, Borland C++ 必须检测支持显示器的硬件, 找不到必须的硬件时, /d 选择项将被忽略。当运行和调试程序时, 以及在使用 Borland C++ 的 DOS shell 功能时, 双监视器方式是很有用的。

在双监视器方式下, 其中一个监视器是现用的, 另外一个处在待用状态。现用的监视器

接收数据,而待用的不接收任何数据,但它仍显示先前送给它的数据。

用户可以用 DOS 中的 MODE 命令来决定哪个监视器处在使用状态下,这条命令允许用户选择视频方式。若选择了彩色方式,数据将被送往彩色监视器;若选择单色方式,数据将被送往单色监视器。使用下面这条命令后,彩色监视器处在现用状态:

```
MODE Co80
```

而下面这条命令使单色监视器处在使用状态下:

```
MODE MoNo
```

当在双监视器方式下启动 Borland C++ 时,Borland C++ 使用待用监视器,而把程序输出送往现用监视器。因此,若想在 Borland C++ 后运行程序,并想在彩色方式下显示程序,必须在启动 Borland C++ 前,使彩色监视器处在现用状态下。

当使用了 /d 选择项启动 Borland C++ 后,要避免以下情况:

- 在 Borland C++ 的 DOS Shell 下改变显示方式。
- 直接使用待用监视器运行程序。
- 同时使用两个监视器运行或调试程序。

### 3. /e (扩展内存)

/e 选择项使机器扩展内存得到应用。当分配存储区时,Borland C++ 通常换到硬盘中去;使用了 /e 选择项后,Borland C++ 换到机器的扩展内存中去,这样可以提高 Borland C++ 的运行性能。

### 4. /l (液晶显示屏)

/l 选择项使 Borland C++ 优选使用液晶显示屏。

### 5. /m (MAKE.EXE)

/m 选择项作用和 /b 选择项相似,但使用 /m 时,文件被制作而不是被建立。使用 /b 选择项时,所有文件都被重新编译和连接,而使用 /m 选择项时,只编译和连接过了时的文件。

### 6. /p (恢复调色板)

/p 选择项是为 EGA 适配器而使用。如果程序改变了 EGA 调色板,使用 /p 选择项可以在转换屏幕时恢复调色板。

### 7. /rx (RAM 盘)

/rx 选择项可以使机器 RAM 盘得到利用。当分配存储区时,Borland C++ 通常交换到硬盘上;而使用了 /rx 选择项后,Borland C++ 则交换到 RAM 盘上。这样可以提高 Borland C++ 的性能,/rx 中的 x 指定使用哪一个 RAM 盘。

### 8. /s (使用可用的内存)

选择项允许 Borland C++ 在编译过程中,可以使用机器中的大部分可用内存,用于建立表。这样可提高编译处理的速度。

## 9. /x(扩充内存)

/x 选择项使机器的扩充内存得到利用。当分配存储区时,Borland C++通常交换到硬盘中去。使用了/x 选择项后,Borland C++ 2.0 交换到机器的扩充内存中,这样可相当大地提高了编译性能。注意 Borland C++3.0 并不支持这个选择项,它已经提高了存储管理技术。

### 1.2.2 使用 IDE 菜单和窗口

新版本的集成开发环境比以往任何的 Borland 编译程序都有效而且实用。用户拥有了 IDE,就具有了在简便易用的单环境下所需要的所有编程工具。

本节中将介绍怎样应用集成开发环境的菜单和窗口。图 1.2 显示了集成开发环境的主要部分。

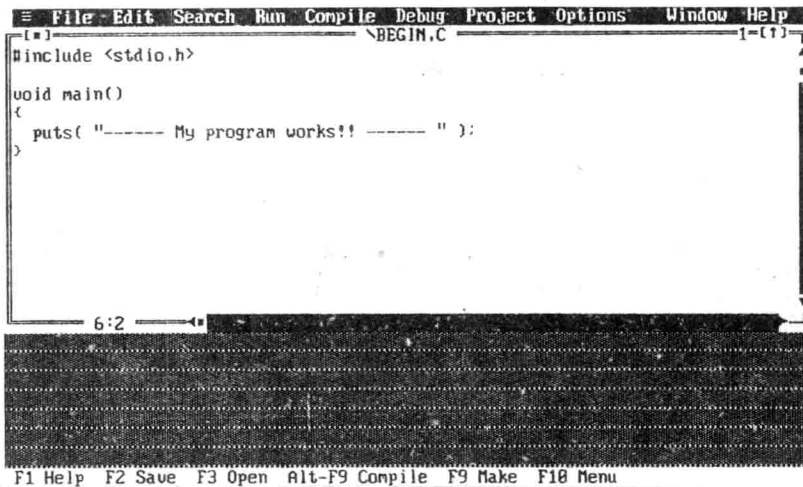


图 1.2 集成开发环境(IDE)

IDE 由以下三个主要部分构成:

(1)菜单系统。菜单系统使用户能够使用 Borland C++ 的高效公用程序和功能,这是通过选择菜单中的功能项目实现的。从图 1.2 可以看出,主菜单在屏幕中第一行,其中每一个项目都和一组公用程序相联系。

(2)窗口系统。窗口系统是 IDE 中最有用和可见的部分,它提供各种窗口系统。在编辑程序、调试程序和观察程序的运行状况时,窗口是很有用的。其他一些窗口使用户可以选择功能项或用于显示一些重要的信息。图 1.2 显示了一个编辑窗口,在窗口中可以建立或修改用户的程序。窗口的文件是 FIRST.C。

(3)状态行。在屏幕底部,状态行给出了一些常用命令的简捷使用键,或告知 Borland C++ 的运行状态,或提供对用户选择命令的一些暗示。在图 1.2 中,状态行显示了一个简捷键的清单。

## 菜单系统

菜单系统包括了一些实用程序和功能,这些对有效管理编程任务是很必要的。这些实用程序和功能包括:处理文件、编辑程序、编译、调试、运行。用户可以通过三种途径进入菜单系统,前两种通过键盘,第三种通过鼠标器。

1. 进入菜单系统的第一种方法是:进入主菜单,移动亮光条到某一命令项时,回车选择。在 IDE 中任何状态下,按 F10 键就可以进入主菜单。这时功能行中出现亮块。用箭头键移动亮块到某一项目处,键入回车,出现下拉菜单。可以用箭头键来选择某一命令,或直接键入命令行中发亮的字母。选择好命令后,键入回车就可执行。图 1.3 显示了一个 Run 命令的下拉菜单。

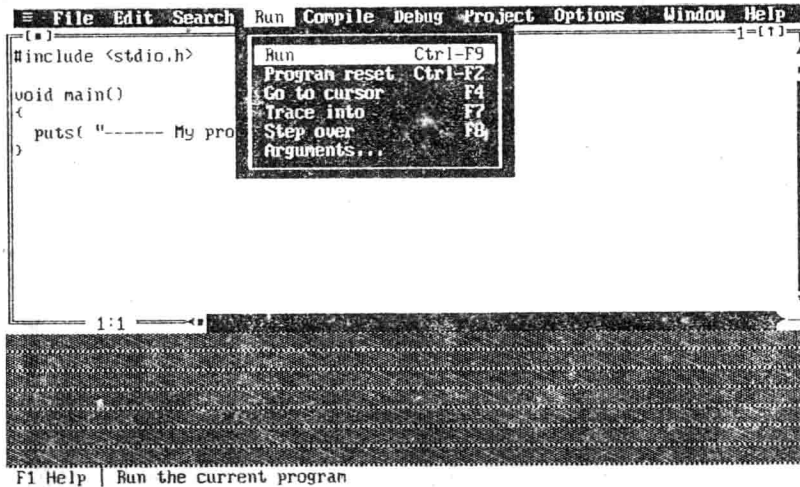


图 1.3 IDE 的下拉菜单

2. 进入菜单系统的第二种方法是使用 Alt-? 组合键。同时按下 Alt 和主菜单中命令的发亮字母,会出现下拉菜单。这时用箭头键移动光条到需要的命令处,键入回车。选择命令后可能出现三种结果:

- (1) 执行命令。
- (2) 显示子菜单。

(3) 显示包含菜单的对话框,提示已选择命令的其他选择。在执行命令前,用户必须向对话框提供必要的信息。

3. 第三种进入菜单系统的方法是当鼠标指向主菜单中要选择的命令时敲键,同样用这种方法选择下拉菜单中的命令。如果选择命令后出现了对话框,用户就需要键入必要的命令。还有一些情况下,仍然是用鼠标器在对话框内进行选择。本节后面还将详细介绍对话框的内容。

Borland C++ 具有两套菜单:一套完全式和一套省略式。两套菜单之间的区别在于完全菜单有更多可供选择的项。尽管省略式能够处理大部分的需要,但完全式在编程时可以提供更多的控制。可以用两种办法来激活完全菜单。第一种办法是选择主菜单中的 Options 功能项,然后选择下拉菜单中的 Full menus,第二种办法是使用 BCINST 程序设置 Borland C++

2.0,以便自动使用完全菜单。Borland C++3.0在BCINST中并不支持这项选择,因为IDE可以完成所有的功能。

### 窗口系统

用户使用 Borland C++ ,将在窗口中完成大部分工作。IDE 有数种不同的窗口,可以提供区域用来写程序、观察程序输出、调试问题代码、显示状态信息以及跟踪机器运行的情况。大多数窗口都具有相似的功能(参见图 1.4)。

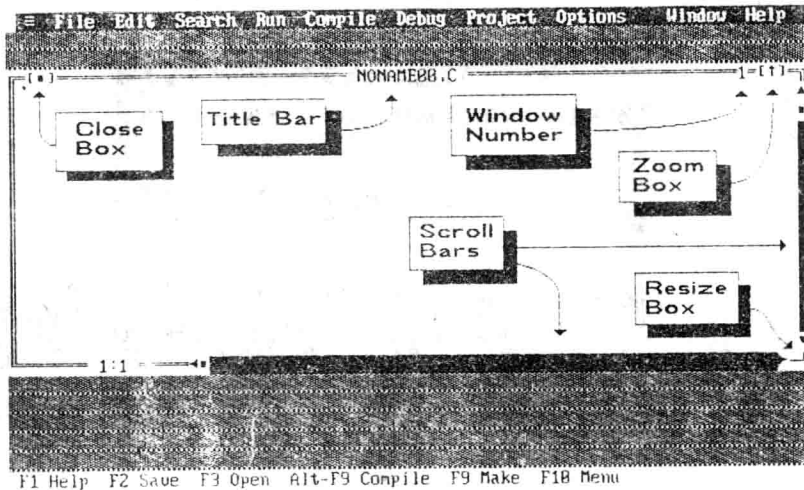


图 1.4 Borland C++窗口特征

以下是 Borland C++窗口具有的基本功能。

1. 闭合箱。闭合箱位于窗口的左上角。用鼠标器选择闭合箱时就关闭整个窗口。若没有鼠标器,可以用功能项 Window 下拉菜单中的 Close 命令来关闭整个窗口。

2. 名称行。名称行位于窗口顶部,它显示窗口名。当鼠标指向名称行时,连接两次则改变窗口大小;用鼠标拖动名称行时则移动窗口。

3. 窗口号。窗口号有助于用户跟踪打开的窗口。在 Borland C++中,用户可以有最多九个编号窗口。通过按 Alt-O 组合键或者在 Window 子菜单中选择 List 命令,可以得到所有打开窗口的清单。在窗口中进行工作前,必须先激活窗口。可以使用以下方法:

(1) 按下 Alt 键并键入想激活的窗口号数字。

(2) 使用 Window 子菜单中的 List 命令,并选择想激活的窗口号。或使用 Window 子菜单中的 Next 命令。

(3) 在窗口任意处敲鼠标。

激活的窗口显示在其他窗口之上,并围以双线边界。

4. 缩放框。用户使用缩放框可以把窗口放大到最大幅度。这时需要用鼠标器选择缩放框,若没有鼠标器,可以选择 Window 子菜单的 Zoom 命令。若窗口已在最大幅度时再使用缩放框,窗口恢复到以前尺寸。

5. 卷滚行。垂直方向和水平方的卷滚行位于窗口的右部和下部。使用卷滚行可以翻卷窗口内容。卷滚行的两端有箭头,用鼠标器单击箭头可以滚动一行。箭头边是阴影框,用鼠

标器单击阴影框一次将改变一页的内容。也可以用鼠标器拖卷滚框来快速改变屏幕内容。卷滚框同时标明了信息在窗口中相对于文件其他部分的位置。

6. 变幅框。用鼠标器拖变幅框可以改变窗口的大小。

用户可能对对话框还不太熟悉。当选择菜单中一些命令后,窗口中会弹出对话框,这时用户对使用的命令可以做一些选择。当菜单中所有带有省略号(…)的命令被选中时,都会产生对话框。图 1.5 显示了 Search 子菜单中 Find 命令的对话框。

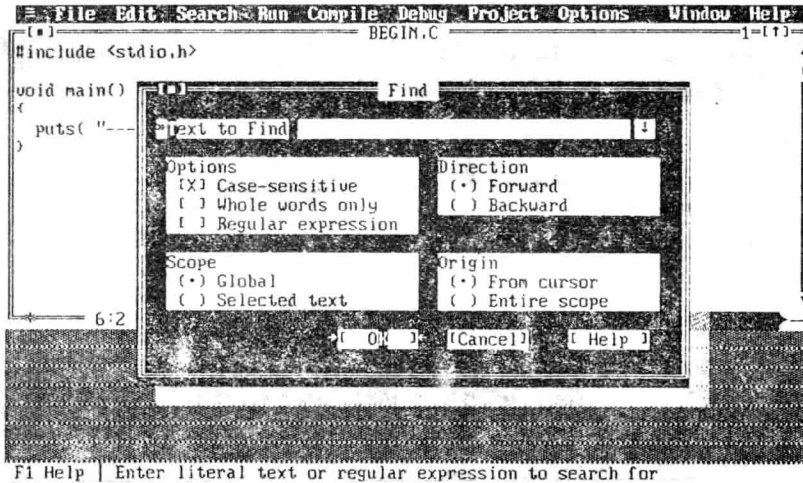


图 1.5 对话框示例

大部分对话框具有以下项目:

(1) 输入框。把正文输入到框中,用户也可以用基本编辑键对输入的正文进行编辑。若需要向输入框中输入控制码,则在输入之前要先键入 ^ P。后面有向下键头的对话框联系着一个历史清单,其中记录着先前在输入框中使用过的项。当用户需要重复或修改以前的命令时,这个特征是很有用的。敲向下箭头键,或用鼠标器键向下箭头符号,就可以激活历史清单。若不想使用历史清单中的任何选择项,只需按 Esc 就能退出清单。图 1.5 中,对话框的名称是 Text to Find(要找到的文本)。

(2) 清单框。清单框使用户不需离开对话框就能够从清单中选择项目。显示的清单长度是不同的。清单框的一个典型例子是 File 子菜单中 Open 命令显示的文件清单。

(3) 动作按钮。用户在对话框中做好选择或键入必需的信息后,就可以使用合适的动作按钮来执行(或取消)命令。标准的动作按钮是 OK、Cancel 和 Help(参见图 1.5)。激活动作按钮的方法是:使用鼠标或同时按下 Alt 和动作按钮上发亮的字母。

(4) 无线按钮。用户使用无线按钮,可以在一组互不相关的命令中选择一项,也只能选择一项。无线按钮的使用方法是:对准按钮按鼠标器,或同时按下 Alt 键和无线按钮中发亮的字母。也可以先用 Tab 键先移动光标到一组无线按钮中,再使用箭头键挑选一个无线按钮。图 1.5 中在 Scope、Direction 和 Origin 选择项下有无线按钮。

(5) 开关框。用户可以使用开关框来打开或关闭一个选择项。当开关框中有 X 时,那个选择项被打开;若开关框没有 X 时,选择项被关闭。对应一个主题的开关框组中,用箭头键挑选其中一个开关框,按空格键就可以打开或关闭选择项。若用户使用鼠标器时,在开关框



处按鼠标键就可以打开或关闭选择项。图 1.5 中的 Option 选择清单就是一组开关框。

## 1.3 设置 Borland C++

Borland C++ 高效率的一个方面在于它的多用途。用户完全不必只局限于一个 Borland C++ 给出的完善的编程环境,用户能够改变环境以适合自己的兴趣和需要。用 Options 子菜单就可以改变 Borland C++ 的环境。

### 1.3.1 设置 IDE 选择项

Borland C++ 允许用户在 IDE 中改变编程环境,这一功能可以由菜单系统中 Options 下的命令来完成。Options 子菜单下的内容分为三部分:编译选择项、环境选择和环境信息。图 1.6 为一个 Options 子菜单。

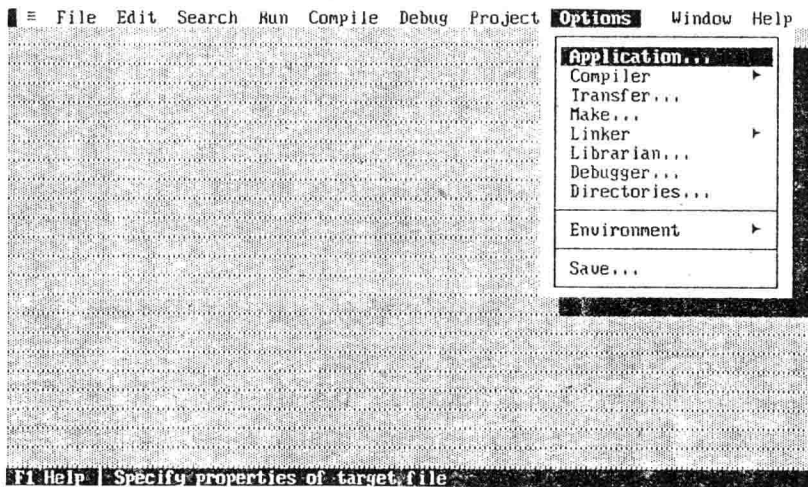


图 1.6 Borland C++ Options 子菜单

Options 菜单的第一部分控制着用于编程工程中的功能和应用程序。这部分内容大都用于设置编译器。Options 菜单下的 Compiler 命令被选择时弹出菜单,其中包括用于控制编译 C 和 C++ 程序的命令。内容如下:

- Code Generation(生成目标码)命令给出对话框,允许用户改变 Borland C++ 产生通常目标代码的方式。
- Advanced Code Generation(生成高级目标码)选择项包括要使用的数学协处理器、CPU 类型及其他选择项。
- Entry、Exit(入口、出口代码)选择项有助于设置编译器以产生不同类型的代码,比如 DOS 程序、DOS 覆盖程序、窗口程序和窗口 DLL 等。其他选择项用于决定使用 C 或 Pascal 类型调用的哪一个。
- C++ Options 命令使你改变为 C++ 程序生成目标代码的方式。
- Advanced C++ Options(高级 C++ 选择命令,只适用于 C++ 3.0)显示一些附加设置项,用于控制 C++ 程序的编译方式。用户在这时可以控制建立多少数据指针和虚