

全国高校计算机基础教育研究会组编

计算机教育丛书

谭浩强 主编

上册



谭浩强 刘炳文 编著

中国科学技术出版社

TT 312

T 2 2-13

1

385157

全国高校计算机基础教育研究会组编

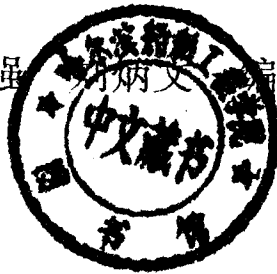
计算机教育丛书

谭浩强 主编

C++程序设计教程

上册

谭浩强 编著



中国科学技术出版社

· 北京 ·

图书在版编目(CIP)数据

C++程序设计教程 上册/谭浩强编著. —北京:中国科学技术出版社,1995.7
(计算机教育丛书)

ISBN 7-5046-1952-3

I. C…

Ⅰ. 谭…

Ⅱ. C++-程序设计-教程

Ⅳ. TP312C

中国版本图书馆 CIP 数据核字(95)第 06216 号

JS168/17



中国科学技术出版社出版

北京海淀区白石桥路 32 号 邮政编码:100081

新华书店北京发行所发行 各地新华书店经售

印 刷:武清县长宏印刷厂

*

开本:787×1092 毫米 1/16 印张:20 字数:540 千字

1995 年 7 月第 1 版 1995 年 7 月第 1 次印刷

印数:1—5000 册 定价:22.00 元

《计算机教育丛书》序

近 10 多年来,我国的计算机应用和教育事业在蓬勃地发展,愈来愈多的人认识到:没有计算机就没有现代化,计算机知识已经成为当代知识分子知识结构中不可缺少的一个重要组成部分。高等学校中几乎所有的专业都已开设了计算机课程。在中专、职业高中和中小学中也普遍进行着计算机教育。各个领域的在职干部,无论科技人员还是管理人员,都日益感觉到掌握计算机知识推进各项工作的迫切性。总之,一次新的计算机普及的高潮即将到来,我们对此应有充分的准备。

普及计算机应用首先遇到的问题是:缺乏足够的能够驾驭计算机的人才。当务之急是全面地开展计算机教育。这种教育应当是全方位多层次的,不同领域不同层次的人都需要在原有基础上学习和提高。我们着眼点首先是大多数。我们的目标是:把计算机从少数人手中解放出来,成为广大群众手中的有力武器。这个任务是十分繁重的,需要众多的有识之士共同投入,通力合作,经过长时间的努力才能实现。

为此我们愿贡献微薄之力,拟编辑一套《计算机教育丛书》促进计算机与普及。这套丛书的主要读者是计算机的初学者和初、中级应用人员。在选题上强调以应用为目的。面向应用。在写法上尽量做到通俗易懂,力求科学性、先进性与通俗性的统一。我们将根据计算机科学技术的发展和读者的要求不断扩充丛书的书目,使之符合社会需要。专家和读者能够给我们指出方向、提出要求、提供信息,参加写作。

本丛书的出版得到全国高等学校计算机教育研究会和中国科学技术出版社的大力支持,使丛书得以问世。期望本丛书能在专家和社会各界的关心爱护下逐步发展和壮大,为计算机教育作出贡献!

主编 谭浩强

1993 年 6 月于北京

内 容 提 要

本书内容包括:C++程序设计基础;数据输入输出与程序基本结构;模块化程序设计;数组;结构体和共用体;指针等。

本书可作大专院校教材及自学者的参考用书。

前 言

C++是C语言的超集,它保持了与80年代最流行的C语言的高度兼容性,继承了C语言的功能丰富、表达能力强、使用灵活方便、目标程序效率高、可移植性好等特性,同时融合了面向对象的能力,支持面向对象的程序设计。与其他面向对象和支持面向对象的程序设计语言(诸如Modula-2、Smalltalk、Ada等)相比,C++具有更完善的体系结构和更加强大的功能。从应用的角度来说,它可以充分地发挥硬件的潜力,在降低开发成本的同时,提供了强有力的软件开发工具。正是由于具有这些优良特性,C++可以广泛用于各种系统软件和应用软件的开发实践中。在“对象+消息”的程序设计范式取代“数据结构+算法”的程序设计范式的过程中,C++无疑有着举足轻重的作用。

应当指出的是,C++是面向对象的语言,但不是纯面向对象的语言,而是混合型面向对象语言,即在过程语言中增加面向对象的结构。因此,用它编写非面向对象的程序,比现有的过程语言(包括C语言)更方便。原有的C语言程序,几乎不用作任何修改就可以在C++环境中编译、运行。

C++本身的优异特性,使得它不仅为计算机专业工作者所使用,而且已经引起广大计算机应用人员浓厚的兴趣,并开始用它编写简单的应用软件。但是,由于C++涉及到一些较复杂的概念,不少初学者感到难以掌握,希望能有一本通俗易懂、适合初学者使用的C++教材或参考书。本书就是为了满足广大计算机应用人员和初学者的迫切要求而编写的。

C++是在C语言的基础上扩展而成的。目前已面世的C++著述,大多把重点放在C语言的扩展部分,这对于已学习过C语言的读者,基本上可以满足需要,但对于从未接触过C语言或对C语言不太熟悉的读者来说,用它来学习C++可能是比较困难的,因为在这之前必须先学习C语言。有鉴于此,本书将把C++作为一种独立、完整的语言来介绍。对于C++和C语言中都有的内容(诸如输入输出、动态存储分配等),则只介绍与C++有关的部分。

由于篇幅较大,本书分上、下两册出版。上册主要介绍C++的数据类型(包括基本数据类型、数组、结构体、共用体、枚举类型等),标准输入输出,流程控制,模块化程序设计及指针。下册介绍与面向对象有关的内容,包括类、继承、多态;同时介绍了位运算、C++流和文件操作以及本书使用的C++编译系统——Borland C++ 3.1集成开发环境(IDE);此外,还介绍了C++的图形处理功能。

对于已学习过C语言的读者,阅读本书时可以以下册为主;但上册中有一部分内容是C语言中所没有的,也必须认真学习。在目录中,这些章节标以星号(*)。

本书是C++和Borland C++的基础读物,面向C++的初学者,它不要求读者具有专门的计算机专业知识的基础,只要学习过一种计算机高级语言,就能学习并掌握本书的基本内容。

C++是一种现代的程序设计语言,不依赖于具体的机器和操作系统。但是,如果只是“纸上谈兵”,不编写在具体机器和编译系统中运行的程序,要掌握C++的程序设计技术是不可能的。为此,本书将通过美国Borland国际公司的Borland C++ 3.1来介绍C++的程序设计方法。Borland C++ 3.1与Turbo C 2.0完全兼容,实现了AT&T C++版本的全部功能,同时支持ANSI C和Microsoft windows应用程序的开发,并具有一定程度的功能扩展。Borland C++编译程序可编译C和C++源代码,具有较高的编译效率和很强的代码优化能力,因此已成为国际上最受欢迎的

面向对象程序设计软件。本书中所有的程序示例均已在 IBM-PC 系列机上用 Borland C++ 3.1 调试通过。但是,本书中介绍的 C++ 知识和 C++ 程序具有通用性,绝大多数程序不用作任何修改或只要进行少量修改就可以在其他 C++ 编译系统中编译、运行。

C++ 所涉及的概念比较复杂,规则繁多,较难掌握。考虑到读者已有一种以上的计算机语言的基础,对算法和数据结构已有初步的了解,为了减少学习的难度,本书把重点放在语言的使用上,介绍如何用 C++ 编写程序。所举例子主要是为了使读者了解如何正确使用 C++, 没有过多地讨论算法设计。此外,为了避免顾此失彼,本书没有从面向对象的程序设计入手介绍 C++, 而是首先把 C++ 作为一种程序设计语言来介绍。在了解了 C++ 的特点、基本概念和功能之后,再回过头来介绍 C++ 面向对象的特性。笔者认为,这样可以使读者集中精力学好 C++ 语言,实际上,掌握了 C++ 的功能之后,再理解面向对象的概念就比较容易了。

C++ 是一种大型的、全新的程序设计语言,在我国的应用还处于开始阶段。希望本书能成为引玉之砖,激发广大读者对学习和使用 C++ 的兴趣。

本书在编写过程中,参考并引用了谭浩强编著的《C 程序设计》(清华大学出版社出版)和谭浩强、刘炳文、鲍泓编著的《Turbo C 程序设计教程》(人民邮电出版社出版)。

由于作者水平有限,经验不足,加之时间仓促,缺点和错误在所难免。恳请专家和广大读者批评指正。

编者

1994. 8. 1

计算机教育丛书编委会

主 编:谭浩强

副主编:刘瑞挺 吴文虎 李大友

秘书长:朱桂兰 周山芙

编委会:(以姓氏笔画为序)

边奠英 史济民 刘甘娜 刘炳文

刘祖照 朱桂兰 朱继生 陈美玲

周山芙 张基温 席先觉 秦笃烈

责任编辑:朱桂兰

封面设计:赵一东

目 录

第一章 概述	(1)
§ 1.1 C++发展简史	(1)
§ 1.2 C++的主要特点	(1)
§ 1.3 Borland C++ 3.1 上机步骤	(2)
习题	(7)
第二章 C++程序设计基础	(8)
* § 2.1 C++源程序构成	(8)
* § 2.2 C++词法元素	(10)
2.2.1 标识符	(10)
2.2.2 注释	(11)
2.2.3 其他符号	(12)
§ 2.3 C++数据类型	(14)
§ 2.4 基本数据类型	(15)
2.4.1 数值数据的表示形式	(15)
2.4.2 字符数据的表示形式	(16)
2.4.3 数据的存储	(17)
2.4.4 带符号的数据类型与无符号的数据类型	(17)
§ 2.5 常量与变量	(20)
2.5.1 常量的种类	(20)
2.5.2 常量的表示	(21)
2.5.3 变量及其赋值	(24)
* 2.5.4 变量的定义和初始化	(25)
§ 2.6 运算符与表达式	(26)
§ 2.7 算术运算符与算术表达式	(29)
2.7.1 二元算术运算符	(29)
2.7.2 一元算术运算符	(31)
2.7.3 使用算术表达式有关问题的说明	(32)
§ 2.8 关系运算与逻辑运算	(34)
2.8.1 关系运算符与关系表达式	(34)
2.8.2 逻辑运算符与逻辑表达式	(36)
2.8.3 条件运算符与条件表达式	(37)
§ 2.9 逗号运算符与逗号表达式	(39)
§ 2.10 类型转换	(40)
2.10.1 隐式类型转换	(40)
* 2.10.2 强制类型转换	(41)
习题	(42)
第三章 数据输入输出与程序基本结构	(43)
§ 3.1 C++语句	(43)
* § 3.2 标准输入输出	(45)
3.2.1 字符输入输出函数	(46)

3.2.2	一般输入输出	(49)
3.2.3	格式化输入输出	(55)
§ 3.3	顺序结构	(60)
§ 3.4	选择结构	(62)
3.4.1	if 语句	(62)
3.4.2	switch 语句	(67)
§ 3.5	循环控制结构	(72)
3.5.1	while 循环	(72)
3.5.2	do-while 循环	(78)
3.5.3	for 循环	(80)
3.5.4	循环的嵌套与退出	(85)
§ 3.6	程序举例	(88)
习题	(96)
第四章	模块化程序设计	(99)
§ 4.1	C++ 程序与函数	(99)
§ 4.2	函数的定义与声明	(102)
4.2.1	函数定义	(102)
* 4.2.2	函数原型	(107)
§ 4.3	函数的调用	(109)
4.3.1	概述	(109)
4.3.2	值传递	(111)
4.3.3	嵌套调用	(114)
4.3.4	递归调用	(116)
* 4.3.5	缺省参数函数及其调用	(120)
§ 4.4	C++ 库函数	(122)
§ 4.5	变量的存储	(123)
* 4.5.1	动态变量	(123)
4.5.2	静态变量	(127)
4.5.3	外部变量	(129)
4.5.4	存储类别小结	(136)
§ 4.6	编译预处理	(138)
4.6.1	宏替换	(139)
4.6.2	文件包含	(148)
4.6.3	条件编译	(149)
* § 4.7	内置函数	(154)
* § 4.8	函数重载	(156)
习题	(159)
第五章	数组	(163)
§ 5.1	数组的定义	(163)
5.1.1	一维数组的定义	(163)
5.1.2	二维数组和多维数组的定义	(164)
§ 5.2	数组的初始化	(165)
5.2.1	一维数组的初始化	(165)
5.2.2	二维数组和多维数组的初始化	(166)
§ 5.3	数组元素的引用	(168)

5.3.1	一维数组的引用	(168)
5.3.2	二维数组和多维数组的引用	(169)
§ 5.4	数组作为函数参数	(171)
5.4.1	数组元素作为函数参数	(171)
5.4.2	数组名作为函数参数	(173)
5.4.3	用多维数组作函数参数	(176)
§ 5.5	字符数组和字符串	(177)
5.5.1	字符数组的定义和初始化	(177)
5.5.2	字符串及其存储	(179)
5.5.3	字符串的输入输出	(180)
5.5.4	字符串函数	(183)
5.5.5	二维字符数组	(186)
5.5.6	字符数组应用举例	(187)
§ 5.6	程序举例	(191)
习题		(196)
第六章	结构体和共用体	(197)
§ 6.1	结构体	(197)
6.1.1	引例	(197)
* 6.1.2	结构体类型与结构体变量的定义	(199)
6.1.3	结构体变量的初始化及其引用	(203)
6.1.4	结构体数组	(206)
§ 6.2	结构体作为函数参数	(212)
6.2.1	结构体参数的传递	(212)
6.2.2	传递整个结构体	(213)
6.2.3	返回结构体类型值的函数	(217)
§ 6.3	共用体	(219)
6.3.1	基本概念	(219)
6.3.2	共用体变量的定义及其引用	(220)
* 6.3.3	无名共用体	(224)
§ 6.4	枚举类型	(225)
* 6.4.1	枚举常量和变量的定义	(225)
6.4.2	枚举变量的初始化和引用	(226)
§ 6.5	用 typedef 定义类型	(229)
习题		(232)
第七章	指针	(233)
§ 7.1	指针的概念	(233)
§ 7.2	指针变量的定义和引用	(234)
7.2.1	指针变量的定义	(235)
7.2.2	指针变量的初始化和引用	(236)
§ 7.3	指针运算	(239)
7.3.1	指针的算术运算	(239)
7.3.2	指针的赋值运算	(239)
7.3.3	指针的关系运算	(240)
§ 7.4	指针与数组	(241)
7.4.1	指针与数组的等价性	(241)
7.4.2	字符串与指针	(245)

§ 7.5	指针作为函数参数	(247)
7.5.1	函数参数的地址传递方式	(247)
7.5.2	数组的指针作为函数参数	(249)
7.5.3	字符串指针作为函数参数	(250)
* § 7.6	引用	(251)
7.6.1	独立引用	(251)
7.6.2	引用参数	(253)
7.6.3	返回引用的函数	(256)
§ 7.7	返回指针值的函数	(257)
§ 7.8	指针数组	(259)
7.8.1	指针数组与多维数组	(260)
7.8.2	字符指针数组与字符串	(261)
7.8.3	指针数组作为函数参数	(264)
§ 7.9	指向指针的指针	(270)
7.9.1	间接寻址	(270)
7.9.2	指向指针的指针与指针数组	(271)
§ 7.10	指向函数的指针	(272)
7.10.1	函数指针的概念	(272)
7.10.2	用函数指针变量调用函数	(274)
7.10.3	用指向函数的指针变量作为函数参数	(276)
§ 7.11	指向结构体类型的指针	(281)
7.11.1	结构体指针变量的定义和引用	(281)
7.11.2	指向结构体数组的指针	(283)
7.11.3	用指向结构体的指针作函数参数	(285)
7.11.4	结构体指针型函数	(287)
* § 7.12	指向 void 类型的指针	(288)
* § 7.13	动态存储分配	(289)
7.13.1	概述	(289)
7.13.2	存储空间分配	(290)
7.13.3	对分配内存操作的检测	(292)
7.13.4	数组的 new 操作	(293)
§ 7.14	链表	(297)
7.14.1	概述	(298)
7.14.2	链表的建立和输出	(299)
7.14.3	链表的删除和插入	(302)
习题	(308)

第一章 概述

§ 1.1 C++ 发展简史

C++是C语言的超集,是C的扩充版本,它是在C语言和Simula67的基础上发展起来的。众所周知,C语言是70年代出现的程序设计语言,由于它模块化能力强,运算符丰富,具有现代语言的各种数据结构,并且可进行位操作,因而功能十分强大,受到软件人员的欢迎。近20年来,C语言凭借其灵活性和高效性,在程序设计界占领了广泛的市场,已经在许多重要产品的开发中发挥了极为关键的作用。但是,随着软件工程规模的扩大,C语言的局限性也逐渐暴露出来。当程序量超过50000行,开发人员达几十个时,系统维护的工作量变得相当大,而且系统的整体性难以保证。

为了克服这种局限性,美国AT&T公司的Bell(贝尔)实验室的研究员Bjarne Stroustrup博士在C语言的基础上,扩充引进了Simula67语言中面向对象的特性,研制出一种新的语言。这种语言最初被称为“带类的C语言”(C with Class),1983年,正式定名为C++。

可以看出,C++具备C语言的各种优点,同时增加了面向对象的程序设计的支持。它实质上是C和Simula67这两种功能强大的程序设计方法的有机融合。因此,Bjarne Stroustrup对C语言的功能扩充是革命性的,C++并非只是C语法的扩充,它允许数据抽象、封装、继承和在相关类之间进行多态的消息传递。C++是真正面向对象的程序设计语言,而不是像Ada或Modula-2那样只是支持基于面向对象的程序设计。

1983年7月,C++第一次在Stroustrup博士的语言开发小组以外公开发表。之后,他和语言小组的同事们在总结该语言使用经验的基础上,进一步精炼提高,不断更新版本,推出了用于UNIX和MS-DOS操作系统的AT&T2.0版。

美国Borland国际公司从80年代中期也开始了对C的扩充,并于1990年推出了Turbo C的面向对象版——Turbo C++。之后,又陆续推出了Turbo C++ 2.0、Borland C++ 2.0、Turbo C++ 3.0和Borland C++ 3.0,1992年推出了该公司的最新版本——Borland C++ 3.1。此外,Borland国际公司于1993年12月推出了C++的国际化版本——Borland C++ 4.0。从目前情况来看,国内用户使用的C++版本以Borland C++ 2.0和3.0为主。本书中的例子在Borland C++ 3.1环境下运行通过。

Borland C++ 3.1可以满足C++和C程序员对于专业化和优化的编译器的要求,同时也可服务于AT&T公司的C++ 2.1版和ANSI C的程序员。Borland c++ 3.1快速而有效,可以让用户方便地建立包括Microsoft Windows应用程序在内的任何应用程序。

Borland C++ 3.1是面向对象(OOP)的程序设计语言,是C语言的进一步发展,因此也是可移植的。一个用C++编写的应用程序,可以很容易地从一个系统移植到另一个系统。

§ 1.2 C++ 的主要特点

C++是以C语言为基础,支持数据抽象和面向对象的程序设计语言。以C语言为基础,使得C++保持了C的紧凑、灵活、高效和可移植性强的优点。C++对数据抽象的支持主要基于类的概

念和机制,面向对象的支持主要通过虚拟函数来实现。

如前所述,C++实际上是对C语言的扩充,而扩充的部分绝大多数来自许多著名语言中最优秀的特征。例如,C++从Simula67中吸取了类;从Algol68中吸取了运算符重载、引用以及在程序的任何地方声明和定义变量;综合了Ada的类属和Clu的模块特点,形成了C++的抽象类;从Ada、Clu和ML中吸取了异常处理,从BCPL中吸取了用//表示注释。

由于C++既有数据抽象又有面向对象能力,又比其他面向对象语言(如Smalltalk、Eiffel、Modula-3等)的运行性能高得多,加之C语言的普及,因而从C到C++的过渡较为平滑;C++与C的高度兼容,使得大量的C程序可以方便地在C++环境中重用。这些因素使得C++能在短短的几年内迅速流行,成为当今面向对象程序设计的主流语言。

Borland C++3.1具有C和C++程序设计的全部功能,完全实现了AT & T C++ v2.1版规范指标,并提供了100%符合美国国家标准协会(ANSI)最新C语言标准的编译程序,完全支持Kernighan和Ritchie(K & R)定义。C++编译程序和C编译程序均包括两种版本,即集成开发环境(IDE)和命令行版本。总的来看,Borland C++3.1具有如下一些主要特点:

1. 全局优化。C++提供了一整套优化选择项,使用户可以控制代码生成,从而用最简便的方式来编写程序,Borland所独有的预编译头文件,可以大大缩短编译程序的编译时间;优化也很快,在短时间内即可产生短小、高速和高效的程序代码。

2. DPMMI编译程序。当编译受到内存限制的大程序时,C++3.1使用工业标准DPMMI保护模式,它允许编译程序在DOS或Windows增强模式的保护模式下运行。

3. Microsoft Windows编程。利用Borland C++3.1,可以编写Windows应用程序,包括动态连接库(.DLL)和可执行文件(.EXE)。它增加了许多新功能,例如资源编译器(Resource Compiler)、帮助编译器(Help Compiler)、资源工作库(Resource Workshop),同时提供了一些C和C++ Windows应用程序的实例。

4. EasyWin。这是一种自动的Windows转换特性,它不用改变代码行就可以把用C++编写的DOS应用程序转换为Windows应用程序。

5. 程序开发平台。Borland C++是一个集编辑、编译、调试、运行和优化等功能于一体的、具有联机帮助和热键触发等特点的优秀的软件开发环境,并支持鼠标和多重覆盖窗口。此外,还共享工程、配置和桌面文件。不管使用程序员平台,还是使用运行于Windows集成开发环境,其工作环境都相同。

§ 1.3 Borland C++3.1 上机步骤

Borland C++是大型软件,必须安装在硬盘上才能使用。具体安装方法及其或能将在第十三章详述。

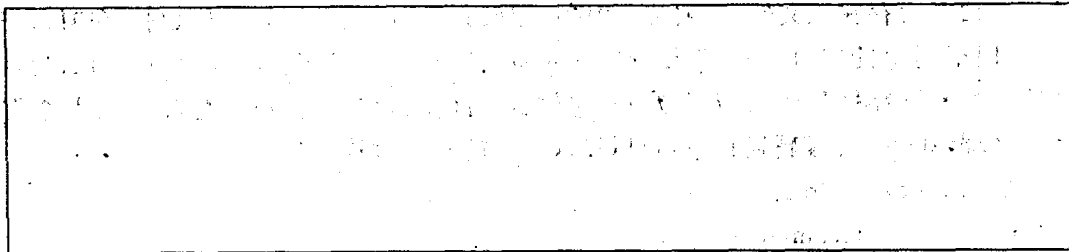
Borland C++3.1集成开发环境(IDE)为源程序的建立(编辑)和编译、连接、运行提供了极大的方便。这一节通过例子说明在Borland C++3.1集成环境中建立和运行C++程序的过程。

在Borland C++集成环境下,程序的建立及编译、连接是通过菜单来完成的。Borland C的菜单操作有着十分丰富的内容,将在第十三章详细介绍。这里只介绍编辑、编译、连接及运行所用到的最基本的菜单命令。

(一) 启动 Borland C++ 开机后,在DOS提示符后打入

bc <CR> (<CR>为回车键,下同)

即可启动 Borland C++,进入集成开发环境,其初始画面如图 1.1 所示。



F1—Help Specify environment settings

图 1.1

进入 Borland C++ 集成环境后, 显示屏幕的大部分是一块灰色图案, 它是用户工作区(目前是空的), 在屏幕的顶部是主菜单, 底部是“热键”。

在 Borland C++ 集成环境下, 必须执行指定的菜单命令, 才能完成相应的操作。Borland C++ 3.1 的主菜单显示在屏幕的顶部, 共 11 个菜单, 每个菜单都有若干个命令, 通过这些命令来实现所需要的操作。

为了执行菜单命令, 必须先“打开”含有该命令的菜单, 通常把打开菜单的操作叫做选择菜单。主菜单可通过下述三种方式中的一种来选择:

(1) 按 F10, 然后按每个菜单名的第一个字母键。例如, 为了选择 Run 菜单, 可在按 F10 后, 再按 R 键。

(2) 按 F10, 然后用 → 或 ← 键把块光标移到所需要的主菜单名, 再按回车键。

(3) 按住 Alt 键, 不要松开, 接着按所需菜单的首字母键。例如, 按 Alt, 不要松开, 再按 R 键, 即可选择 Run 菜单。本书中把这种选择方式记做 Alt-R。

上述三种方式作用相同, 在下面的叙述中, 我们将以第三种方式, 即 Alt 菜单名加首字母键为主。

如果系统配有鼠标器, 则选择菜单的操作将是十分简单的。只要把鼠标光标移到指定的菜单名上, 然后按鼠标左按钮, 即可打开该菜单。

选择某个菜单后, 在该菜单下面以下拉方式显示该菜单的命令, 例如, 按 Alt-F 选择“File”菜单, 就在其下面显示 File 菜单的命令, 如图 1.2 所示。可以用下述三种方式中的一种执行菜单命令:

(1) 用 ↑ 或 ↓ 移动条形光标, 让条形光标盖住要执行的菜单命令, 然后按回车键。

(2) 按命令中呈红色显示(在黑白显示器上高亮显示)的字母键, 即可执行相应的命令。例如, 按 File 菜单中的“O”键, 即可执行“Open”命令。

(3) 用鼠标点按需要执行的菜单命令。

如果用第三种方式选择菜单, 则选择菜单和执行菜单命令可连续进行。例如, 按 Alt-F 选择 File 菜单, 然后按“O”键, 即可执行“Open”命令。本书中把上述过程记做 Alt-F, O。

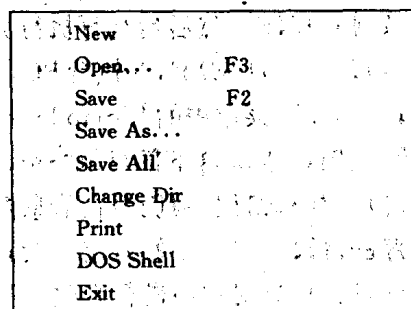


图 1.2

(二)建立(编辑) Borland C++ 程序 启动 Borland C++ 后,打开 File 菜单,然后选择 Open 命令,此时屏幕上显示一个小窗口。在这个小窗口中键入要建立或编辑的文件的名字,然后按回车键。如果当前磁盘上存在该文件,则被装入内存;如果文件不存在,则建立该文件。在窗口中打入 bc1_1 并按回车键,就建立了一个名为 bc1_1.cpp 的文件(C++的扩展名为 .cpp,由系统自动加上)。此时工作区分为两部分:下部仍为灰色图案,上部变为单色背景(彩色显示器上通常为蓝色),这就是编辑窗口。在编辑窗口中,可以输入如下的 C++ 程序:

```
//Program example 1-1
#include <iostream.h>
main()
{
    char name[10];
    cout<<"What's your name? \n";
    cin>>name;
    cout<<"Hello,"<<name<<"\n";
}
```

上述程序定义一个字符数组,接着提示输入名字,然后显示“Hello,”和输入的名字。

程序一行一行地输入,输入方法与用普通的编辑软件或字处理软件(如 Wordstar)输入其他信息时基本相同。

(三)运行 C++ 程序 键入上述程序后,为了保险起见,最好先将程序存盘。其操作是:按 Alt-F 打开 File 菜单,然后按 s 键,选择 Save 命令,上述程序即以 BC1_1.CPP 的文件名存入磁盘的当前目录下。

可以用 Run 菜单中的 Run 命令运行程序。输入程序并存盘后,按 Alt-R,然后再按 R 键,即可运行该程序。如果没有错误,则屏幕上的信息将被清除,并在屏幕顶部显示“ What's your name?”,等待输入名字,如果输入

```
LiMing <CR>
```

则显示:“Hello, LiMing”。不过,这行信息显示后马上消失,回到 Borland C++ 3.1 集成环境。按 Alt-F5 可以转换到“用户屏幕”,查看显示内容,按任意键仍回到集成环境。

如果存在语法错误,则会在编辑窗口下面的信息窗口(Message)显示相应的错误信息,此时按 F6 键或回车键,可使光标回到编辑窗口,并停留在有错误的程序行上,可以对程序进行修改,然后再一次运行程序。

Run 菜单中的 Run 命令具有三种功能,即对源程序进行编译、连接生成可执行文件(.EXE)并运行程序。因此,在执行该命令时,先编译源程序,生成目标文件(.OBJ),接着连接目标文件,生成可执行文件(.EXE),然后运行可执行文件。在对上面的程序执行 Run 命令时,先生成 BC1_1.OBJ,再生成 BC1_1.EXE,并执行 BC1_1.EXE。如果只想生成 .OBJ 文件,不生成 .EXE 文件,则应选择 Compile 菜单中的“Compile to OBJ”命令,其方法是:按 Alt-C,再按 C 键。编译完成后,如果没有错误,则屏幕下部显示“Success:press any key”,表示编译成功,此时按任意键即可回到集成环境。如有错误,则仍会在信息窗口显示错误信息,应回到编辑窗口对程序进行修改。

(四)程序的修改 对程序的修改一般有两种情况:一种是在编译中发现错误时,应根据信息窗口中显示的错误信息对程序进行修改。另一种是对存放在磁盘上的旧文件进行修改。

1. 修改错误 在编译(Alt-C,C)或运行(Alt-R,R)程序时,如果有错误,则在窗口底部显示“Error:press any key”,此时按空格键,信息窗口即被激活,亮条出现在第一个错误或警告

信息上,同时在编辑窗口中也有一亮条,用以标示给出的错误或警告在源代码中的相应位置。如果用↑或↓上下移动信息窗口中的亮条,则编辑窗口中的亮条也随着跟踪源代码中发生错误的位置。如果把亮条置于“Compiling”,则编译器显示源文件的最后位置。当一行中信息太长时,可用→或←键使窗口水平滚动,以查看“隐藏”在窗口之外的信息。

为了修正错误,可将信息窗口中的亮条置于相应的错误位置上,然后按回车键,光标移到编辑窗口中发生错误的地方,即可对错误进行修改。

出现错误后,屏幕上两个窗口,上面是编辑窗口,下面是信息窗口。在编译过程中发生错误时,光标位于信息窗口中。为了修改错误,必须使光标回到编辑窗口。除上面讲的按回车键外,按F6也可以使光标回到编辑窗口。如果再按一次F6,则光标又移到信息窗口。也就是说,F6使得光标在两个窗口中来回移动。

2. 修改旧文件 为了修改已存放在磁盘上的文件,必须先把该文件装入内存,然后才能修改。可以用下面的方法装入文件:

按 Alt-F,O,执行 File 菜单中的 Open 命令,此时屏幕上显示出一个“对话框”,对话框中有一个文件名输入框,可以在输入框中键入要打开(即装入)的文件名,然后按回车键,文件即被装入内存。如果该文件在当前目录下,则直接键入文件名即可,否则应给出其所在的路径。此外,在对话框中还列出了当前目录下所有 C++ 源程序的名字,用→、←、↑或↓键把条形光标移到要装入的文件名,按回车键,该文件即被装入内存。

3. 简单编辑操作 修改程序实际上是对程序进行编辑操作。下面介绍几个简单的编辑命令,利用这些命令,可以对程序进行一般的编辑处理。编辑操作有着十分丰富的内容,将在第十三章详述。

(1) 光标移动 为了修改程序,必须把光标移到需要修改的位置。可以用下列键移动光标:

→	右移一个字符
←	左移一个字符
↑	上移一行
↓	下移一行
PgUp	上滚一屏
PgDn	下滚一屏

(2) 插入与重写 为了插入字符,必须使编辑器处于插入状态。启动 Borland C++ 后,编辑器处于插入状态。此时把光标移到要插入的字符的左边,键入字符,该字符即被插入。此外,把光标移到一行的开头,按回车键,即可在该行的上面插入一个空行。

在插入状态下,按一次 Ins 键,编辑器被置为重写状态。在这种情况下,键入的字符将取代光标所在位置的字符。

Ins 键是一个“开关”,按一次变为重写状态,再按一次又变为插入状态。

(3) 删除

删除字符: Del 键删除光标所在位置的字符,退格键删除光标左边的字符。

删除行: 按 Ctrl-Y 键可删除光标所在的行。

以上是一些简单的编辑操作。请读者用这些操作对程序 BC1_1.CPP 进行修改,步骤如下:

① 启动 Borland C++:

bc <CR>

② 按 Alt-F,O(或按 F3),然后键入:

BC1_1 <CR>