

新
版

21世纪

高职高专系列教材

C语言程序设计 实用教程

◎李庆亮 狄文辉 陈 震 等编著



机械工业出版社
CHINA MACHINE PRESS





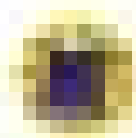
清华大学出版社

清华大学出版社

C语言程序设计 实用教程

清华大学出版社

清华大学出版社



21 世纪高职高专系列教材

C 语言程序设计实用教程

李庆亮 狄文辉 陈震 等编著



机械工业出版社

本书从实用的角度出发,本着强调程序设计的概念和思想、深入浅出、注重实践的原则,介绍了 C 语言程序设计的基本概念和方法。通过实例,使初学者尽快掌握程序设计的基本概念,建立程序设计的思想,通过实训环节,引入常用算法解析和综合实例等方式,培养读者分析问题、解决问题的能力。本书与全国计算机等级考试相结合,各章均配有大量的习题,便于读者课后练习。同时还配有实验,便于读者上机练习巩固所学知识。

本书既可作为高等院校理工科各专业学生程序设计的入门教材,也可作为计算机应用开发人员的技术参考书。

图书在版编目(CIP)数据

C 语言程序设计实用教程/李庆亮等编著. —北京:

机械工业出版社,2005.3

(21 世纪高职高专系列教材)

ISBN 7-111-16076-2

I. C... II. 李... III. C 语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 006774 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 王 颖

责任印制: 石 冉

保定市印刷厂印刷·新华书店北京发行所发行

2005 年 3 月第 1 版·第 1 次印刷

787mm×1092mm $\frac{1}{16}$ ·17.5 印张·424 千字

0 001—7 000 册

定价: 24.00 元

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话:(010)68993821、68379646

68326294、68320718

封面无防伪标均为盗版

21 世纪高职高专计算机专业系列教材 编委会成员名单

主 任 周智文

副主任 周岳山 林 东 王协瑞 赵佩华
程时兴 吕何新 陈付贵 朱连庆
陶书中

委 员 (按姓氏笔画排序)

马 伟 马林艺 卫振林 于恩普
王养森 王 泰 王德年 刘瑞新
余先锋 陈丽敏 汪赵强 姜国忠
赵国玲 赵增敏 顾可民 贾永江
顾 伟 陶 洪 龚小勇 眭碧霞
曹 毅 鲁 辉 翟社平

秘书长 胡毓坚

出版说明

根据《教育部关于以就业为导向深化高等职业教育改革的若干意见》中提出的高等职业院校必须把培养学生动手能力、实践能力和可持续发展能力放在突出的地位,促进学生技能的培养,以及教材内容要紧密结合生产实际,并注意及时跟踪先进技术的发展等指导精神,机械工业出版社组织全国40余所院校的骨干教师,对在2001年出版的“面向21世纪高职高专系列教材”进行了修订。

在几年的教学实践中,本系列教材获得了较高的评价。因此,在修订过程中,各编委会保持了第1版教材“定位准确、注重能力、内容创新、结构合理和叙述通俗”的编写特色。同时,针对教育部提出的高等职业教育的学制将由三年逐步过渡为两年,以及强调以能力培养为主的精神,制定出了本次教材修订的原则:跟上我国信息产业飞速发展的节拍,适应信息行业相关岗位群对第一线技术应用型操作人员能力的要求,针对两年制兼顾三年制,理论以“必须、够用”为原则,增加实训的比重,并且制作了内容丰富而且实用的电子教案,实现了教材的立体化。

针对课程的不同性质,修订过程中采取了不同的处理办法。核心基础课的教材在保持扎实的理论基础的同时,增加实训和习题;实践性较强的课程强调理论与实训紧密结合;涉及实用技术的课程则在教材中引入了最新的知识、技术、工艺和方法。此外,在修订过程中,还进行了将几门课程整合在一起的尝试。所有这些都充分地体现了修订版教材求真务实、循序渐进和勇于创新的精神。在修订现有教材的同时,为了顺应高职高专教学改革的不深入,以及新技术新工艺的不断涌现和发展,机械工业出版社及教材编委会在对高职高专院校的专业设置和课程设置进行了深入的研究后,还准备出版一批适应社会发展的急需教材。

信息技术以前所未有的速度飞快地向前发展,信息技术已经成为经济发展的关键手段,作为与之相关的教材要抓住发展的机遇,找准自身的定位,形成鲜明的特色,夯实人才培养的基础。为此,担任本系列教材修订任务的教师,将努力把最新的教学实践经验融于教材的编写之中,并以可贵的探索精神推进本系列教材的更新。由于高职高专教育正在不断的发展中,加之我们的水平和经验有限,在教材的编审中难免出现问题和错误,恳请使用这套教材的师生提出宝贵的意见和建议,以利我们今后不断改进,为我国的高职高专教育事业作出积极的贡献。

机械工业出版社

前 言

程序设计能力是高等院校理工科各专业计算机应用能力培养的重点之一,是利用计算机解决实际工程计算问题的基础。C语言是目前国内外广泛使用的程序设计语言之一,具有功能丰富、表达能力强、使用方便灵活、代码效率高、可移植性好等特点,广泛应用于数据处理、科学计算、系统软件设计和计算机控制等领域,是当今许多高校计算机语言教学入门的首选语言。

在长期的教学实践中,我们深刻地体会到,由于大多数学生在学习程序设计课程之前对计算机的基本知识了解得很少,很难在短时间内建立起程序及程序设计的概念,因此感到程序设计课程难学、枯燥无味,结果达不到预期的教学目的,并影响了后续课程的学习。为此,我们组织长期从事计算机程序设计教学的骨干教师,本着强调程序设计的概念和思想、深入浅出、注重实践的原则,编写了本教材。

本教材力求把知识传授与能力培养融合在一起,将理论与实践有机地结合起来,通过实例入手,使初学者尽早掌握程序设计的基本概念,建立程序设计的思想,通过实训环节,引入常用算法解析和综合实例等方式,与工程实际相结合,培养学生分析问题、解决问题的能力。本书与全国计算机等级考试相结合,各章均配有大量的习题,便于学生课后练习。

参加本书编写的有李庆亮、狄文辉、陈震、茹庆云、郭祖华、孙冬等,由李庆亮统稿。编写过程中得到了许多老师的大力支持和帮助,在此深表感谢。

由于时间仓促,书中难免存在不妥之处,请读者原谅,并提出宝贵意见。

编 者

目 录

出版说明

前言

第1章 概述	1
1.1 发展历史和特点	1
1.1.1 发展历史	1
1.1.2 特点	1
1.2 程序举例、结构特点和书写规则	2
1.2.1 程序举例	2
1.2.2 结构特点	3
1.2.3 书写规则	4
1.3 开发与调试	5
1.3.1 Turbo C的集成开发环境	5
1.3.2 上机步骤	9
1.4 关于程序设计课程的学习	10
1.5 习题	12
第2章 数据类型、运算符和表达式	13
2.1 常量与变量	13
2.1.1 常量和符号常量	13
2.1.2 变量	14
2.2 基本数据类型	15
2.2.1 整型数据	15
2.2.2 实型数据	17
2.2.3 字符型数据	18
2.3 运算符与表达式	21
2.3.1 算术运算符与算术表达式	22
2.3.2 关系运算符与关系表达式	24
2.3.3 逻辑运算符与逻辑表达式	25
2.3.4 条件运算符与条件表达式	27
2.3.5 赋值运算符与赋值表达式	27
2.3.6 逗号运算符与逗号表达式	31
2.3.7 位运算符与位运算	31
2.4 习题	34
第3章 基本程序设计	38
3.1 C语言的语句	38
3.2 算法与结构化程序设计方法	39
3.2.1 算法	39
3.2.2 结构化程序设计方法	42

3.3 顺序结构程序设计	42
3.3.1 数据输出函数	42
3.3.2 数据输入函数	45
3.3.3 实训	48
3.4 选择结构程序设计	50
3.4.1 if语句	50
3.4.2 switch语句	55
3.4.3 实训	57
3.5 循环结构程序设计	61
3.5.1 while循环语句	61
3.5.2 do while循环语句	62
3.5.3 for循环语句	62
3.5.4 break语句和continue语句	65
3.5.5 循环的嵌套	66
3.5.6 实训	68
3.6 习题	70
第4章 数组	77
4.1 一维数组	77
4.1.1 定义	77
4.1.2 引用	78
4.1.3 初始化	79
4.1.4 实训	80
4.2 二维数组	82
4.2.1 定义	82
4.2.2 引用	82
4.2.3 初始化	83
4.2.4 实训	84
4.3 字符数组与字符串	85
4.3.1 定义	85
4.3.2 初始化	86
4.3.3 引用	86
4.3.4 输入输出	87
4.3.5 字符串处理函数	88
4.3.6 实训	90
4.4 习题	91
第5章 指针	94
5.1 概念	94

5.1.1 变量的地址与变量的内容	94	7.2 累加/累乘求和与求积问题 算法	154
5.1.2 直接访问与间接访问	95	7.3 解决不确定性问题的穷举 算法	156
5.1.3 指针与指针变量	95	7.4 排序问题算法	158
5.2 指针变量的定义与引用	96	7.5 数值积分算法	161
5.2.1 定义	96	7.6 多项式计算问题算法	163
5.2.2 引用	96	7.7 非线性方程求解问题算法	168
5.3 指针变量的运算	99	7.8 产生随机数算法	169
5.3.1 赋值运算	99	7.9 习题	170
5.3.2 算术运算	101	第 8 章 结构体、共用体和枚举类型 ...	172
5.3.3 关系运算	102	8.1 结构体类型	172
5.4 指针与数组	102	8.1.1 结构体类型的定义	172
5.4.1 指针与一维数组	103	8.1.2 结构体变量的定义与引用	173
5.4.2 指针与二维数组	105	8.1.3 结构体数组	178
5.5 指针与字符串	108	8.1.4 结构体指针	181
5.6 指针数组与指向指针的指针 ...	111	8.2 共用体类型	184
5.6.1 指针数组	111	8.2.1 共用体类型的定义	184
5.6.2 指向指针的指针	115	8.2.2 共用体变量的定义与引用	185
5.7 习题	116	8.3 枚举类型	187
第 6 章 函数	119	8.3.1 枚举类型的定义	187
6.1 定义与调用	119	8.3.2 枚举类型变量的赋值和使用	188
6.1.1 函数定义的一般形式	119	8.4 习题	189
6.1.2 调用	121	第 9 章 编译预处理	195
6.1.3 函数的参数和函数的值	125	9.1 宏定义	195
6.2 嵌套调用和递归调用	127	9.1.1 不带参数的宏定义	195
6.2.1 嵌套调用	127	9.1.2 带参数的宏定义	197
6.2.2 递归调用	128	9.2 文件包含	201
6.3 变量的作用域	131	9.3 条件编译	201
6.3.1 局部变量	131	9.4 习题	204
6.3.2 全局变量	132	第 10 章 文件操作	205
6.4 变量的存储类别	134	10.1 概述	205
6.5 内部函数与外部函数	138	10.1.1 文件的概念	205
6.6 指针与函数	139	10.1.2 数据流的概念	206
6.7 用指针作函数参数	141	10.1.3 C 的文件系统及其与流的 关系	206
6.8 数组作为函数参数	143	10.2 文件类型指针	206
6.8.1 数组元素作函数实参	143	10.3 文件的打开与关闭	207
6.8.2 数组名作为函数参数	144	10.3.1 打开	208
6.9 返回指针的函数	146	10.3.2 关闭	209
6.10 main 函数的命令行参数	147		
6.11 习题	149		
第 7 章 常用基本算法的 C 语言实现	153		
7.1 统计与计数问题算法	153		

10.4	文件的读写操作	210	12.3	实验 3 顺序结构程序设计 ...	238
10.4.1	单个字符的读写	210	12.4	实验 4 分支结构程序设计 ...	239
10.4.2	字符串的读写	213	12.5	实验 5 循环结构程序设计 ...	242
10.4.3	数据块的读写	215	12.6	实验 6 数组	243
10.4.4	格式化数据的读写	216	12.7	实验 7 指针	245
10.5	文件的定位	217	12.8	实验 8 函数	247
10.5.1	rewind 函数	217	12.9	实验 9 常用基本算法	248
10.5.2	fseek 函数	217	12.10	实验 10 结构体与共用体 ...	249
10.5.3	ftell 函数	218	12.11	实验 11 编译预处理	251
10.6	习题	218	12.12	实验 12 文件	251
第 11 章	综合示例	225	12.13	实验 13 综合练习	252
11.1	一个通信录的程序示例	225	附录		254
11.2	一个有趣的迷宫小游戏	231	附录 A	运算符的优先级与结合性 ...	254
第 12 章	实验	235	附录 B	Turbo C 2.0 的常用库 函数	255
12.1	实验 1 C 语言的集成开发 环境	235	附录 C	Turbo C 2.0 常见错误	262
12.2	实验 2 简单的 C 语言程序 设计	237	附录 D	ASCII 字符表	266

第 1 章 概 述

本章要点

- C 语言的发展历史和特点
- C 语言程序的结构特点
- C 语言程序的开发与调试

1.1 发展历史和特点

1.1.1 发展历史

C 语言是近几年来最优秀的程序设计语言之一。它之所以被大家接受和喜爱,以至得以广泛推广,是与其特点分不开的。C 语言作为一种计算机高级语言,它既有一般高级语言的特性,又有一定的低级语言的特性,故而既适合编写系统程序又适合编写应用程序。

C 语言是 1972 年由美国的 Dennis Ritchie 设计发明的,并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。它由早期的编程语言 BCPL(Basic Combind Programming Language)发展演变而来。在 1970 年,美国电话电报公司(AT&T)贝尔实验室的 Ken Thompson 对 BCPL 语言进行了进一步简化,设计出较先进的并取名为 B 的语言。1972 年贝尔实验室的 Brain.W.Kernighan 和 Dennis.M.Ritchie 对 B 语言进行了完善和扩充,将这种语言取名为 C, C 语言就这样诞生了。

早期的 C 语言主要用于 UNIX 操作系统。由于它的强大功能和各方面的优点逐渐为人们认识,到了 20 世纪 80 年代,C 开始进入其他操作系统,并很快在各类大、中、小和微型计算机上得到了广泛的使用,成为当代最优秀的程序设计语言之一。

随着微型计算机的日益普及,出现了许多 C 语言版本。由于没有统一的标准,使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况,1983 年美国国家标准化协会(ANSI)为 C 语言制定了一套 ANSI 标准,成为现行的 C 语言标准,通常称之为 ANSI C。

目前在微型机上使用的较为流行的 C 语言版本有:Microsoft C(或称 MS C)、Turbo C、AT&T C 等。它们的编译系统虽然基本相同,但还是有一些差异。因此,读者应了解所用的计算机系统配置的 C 编译系统的特点。

1.1.2 特点

C 语言是一种结构化程序设计语言。它层次清晰,便于按模块化方式组织,易于调试和维护。C 语言的表现能力和处理能力极强,因而成为最受欢迎的语言之一。许多著名的系统软件,如 DBASE III PLUS、DBASE IV 都是由 C 语言编写的。

归纳起来,C 语言具有下列特点:

- 1) 语言简洁、紧凑,使用方便、灵活。C 语言具有 32 个关键字,9 种控制语句,程序书写形

式自由、灵活,主要用小写字母表示,压缩了一切不必要的成分。

2) 运算符和数据结构丰富。C 语言除了具有其他高级语言所具有的运算符外,还具有 C 语言特有的运算符,比如增量运算符、赋值运算符、逗号运算符、条件运算符和强制类型转换运算符等。同时 C 语言具有其他高级语言所具备的各种数据结构。

3) 语法限制不太严格,程序设计自由度大。对数组下标越界不作检查,整型与字符型以及逻辑型可以通用。

4) 结构化程序设计语言。结构化程序设计语言的显著特点是代码及数据的分离,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护以及调试。C 语言是以函数形式提供给用户的,这些函数可方便地调用,并具有多种循环、条件语句控制程序流向,从而使程序完全结构化。

5) C 是中级语言。它把高级语言的基本结构和语句与低级语言的实用性结合起来。C 语言可以像汇编语言一样对位、字节和地址进行操作,具备低级语言的特点。

6) 适用范围大,可移植性好。C 语言还有一个突出的优点就是适合于多种操作系统,如 DOS、Windows、UNIX,也适用于多种机型。本身不依赖于机器的硬件系统。

7) 语言表达能力强。C 语言是一种高级语言,能完成数值计算、字符、数据等处理,具有高级语言的通用性。同时又能对物理地址进行访问,对数据的位进行处理和运算,具备低级语言的特点。

8) 生成的代码质量高。C 语言开发的程序生成的目标代码的效率只比用汇编语言开发同样程序生成的目标代码的效率低 10% 到 20%,而编写程序的可读性好,修改、调试容易。

1.2 程序举例、结构特点和书写规则

1.2.1 程序举例

在介绍 C 语言程序的结构特点之前,先来看一个简单的 C 语言程序例子,了解一下用 C 语言写出的程序是什么样子。

```
#include <stdio.h>

main( )                /* 主函数 */
{ int a,b,max;          /* 变量说明 */
  printf("input two numbers: \n"); /* 提示字符串,照原样输出 */
  scanf("%d%d",&a,&b);    /* 输入变量 a,b 的值 */
  if(a>b) max=a;else max=b; /* 比较 a,b 的大小,将大值送入 max */
  printf("max= %d \n",max); /* 输出 */
}
```

上面这个简单程序可分为两个基本部分:第一行是个特殊行,说明程序要用到 C 语言系统提供的标准功能,为此要参考标准库文件 `stdio.h`,有关细节在后面的章节中介绍。从第二行开始是程序基本部分,描述程序所完成的工作。该程序的功能是从键盘上任意输入两个整数,求出两个数中的较大数,并在显示屏幕上输出结果。

该程序结构上有什么特点呢? 首先第一行说明了程序执行要包含的标准库文件,称为程

序的包含部分。如果还有其他要包含的文件或其他要说明的信息的话,可以增加第二行、第三行来说明,这些将后面的章节中将进一步介绍。

该程序的基本部分是以 main 命名的一个函数,或者说是一个程序模块,“{ }”括起来的部分称为函数体。函数体内部又分为两部分,一部分为说明部分,说明程序执行过程中要用到的一些变量及类型,该程序中使用了两个整型变量 a、b 用来接受输入的数据,一个整型变量 max 用来存放计算结果。另一部分为执行部分,由一些命令语句组成。执行部分的第一行调用 printf 函数在显示器上输出提示字符串,请操作人员输入变量 a、b 的值。第二行调用 scanf 函数,接受键盘上输入的整数并存入变量 a、b 中。第三行是使用 if 语句比较变量 a、b 的大小,并将大值传送给变量 max 中。第四行是用 printf 函数输出变量 max 的值。

较为复杂一点的 C 程序还可能包括多个函数或程序模块,名字为 main 的函数是主函数的函数名,每一个 C 源程序都必须有,且只能有一个主函数(main 函数)。其他的函数名可由程序员自己定义。

在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名和关键字由系统定义外,其余都由用户自定义。C 语言规定,标识符只能是字母(A~Z, a~z)、数字(0~9)、下划线(_)组成的字符串,并且其第一个字符必须是字母或下划线。

在程序的每行后用“/*”和“*/”括起来的内容为注释部分,注释部分不执行,用于方便地阅读和理解程序。

该程序如何在计算机上执行呢? 这是初学者比较关心的另一个问题。

上面用 C 语言写出的程序称作源程序,源程序容易书写和阅读,但计算机却不能直接执行,因为计算机只能识别和执行特定二进制形式的机器语言程序。要使计算机能执行该程序,就必须先在编辑环境下把该源程序输入计算机,生成源程序文件,然后将它加工转换成二进制形式的机器语言程序文件。这种转换由 C 语言系统完成,包括“编译源程序”、“连接程序”两大步。

第一步,由编译程序对源程序文件进行分析和处理,生成相应的机器语言目标模块,由目标模块构成的代码文件称为目标文件。目标文件仍然不能执行,因为其中还缺少 C 程序运行所需要的运行系统。另外,一般 C 程序里都要使用函数库提供的某些功能。例如上面例子中需要用到标准函数库的一个标准输入函数(scanf)和一个标准输出函数(printf)。为构造出完整的可以运行的程序,还需要第二步加工——连接。连接程序负责将编译得到的目标模块与其他必要部分(运行系统、函数库提供的功能模块等)拼装起来,形成完整的可执行程序。图 1-1 说明了 C 程序的基本加工过程。

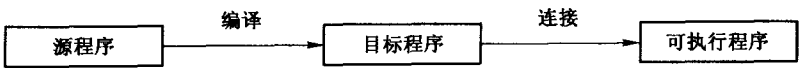


图 1-1 C 程序的加工过程

1.2.2 结构特点

C 语言程序的结构特点:

1) 一个 C 语言程序可以由一个或多个源程序文件组成。对于较大的程序,一般不能将其全部放在一个文件中,而是分别放在若干个源程序文件中。

2) 每个源程序文件可由一个或多个函数组成。一个源程序文件是一个编译单位,而不是以函数为单位进行编译的。

3) 一个源程序不论由多少个文件组成,都有一个且只能有一个 main 函数,即主函数;也可以包含一个 main 函数和若干其他函数,因此函数是构成 C 程序的基本单位。无论 main 函数在整个程序中的位置如何,一个 C 程序总是从 main 函数开始执行。

4) main 函数有两部分组成,即函数首部和函数体。函数首部很简单,就是 main()。函数体由大括弧{}括起来,内部又分为说明部分和执行部分。说明部分是对变量的类型说明。C 语言规定,源程序中所有用到的变量都必须先说明,后使用。执行部分由若干语句组成,用以完成程序的功能。

5) C 语言中的字母要区分大小写,习惯上用小写字母。只有常量、宏定义等多用大写,所有的关键字必须小写。如 int、if 等只能小写,a 与 A 表示两个不同的变量。

6) 每一个数据定义,每一个语句都必须以分号“;”结尾,分号“;”是语句的必不可少的组成部分。但预处理命令、函数首部和花括号“{}”之后不能加分号。

7) 标识符、关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符,也可不再加空格来间隔。例如把 int a; 写成 inta; C 编译器会把 inta 当成一个标识符处理,其结果必然出错。

8) C 语言程序的书写格式非常自由,一行内可以写几个语句、一个语句可以分写在多行上。

9) C 语言没有输入、输出语句,输入、输出的操作是由库函数 scanf 和 printf 等函数来完成的。

10) C 语言的注释符是以“/*”开头并以“*/”结尾的串,即/* 注释内容 */。可以在程序的任何位置给程序加上注释。注释用来向用户提示或解释程序的意义。加上必要的注释,可以增强程序的可读性。注释一般放在程序的开始,用以说明该程序段的功能;或者放在某个语句的后面,对该语句进行说明。程序编译或执行时,不对注释作任何处理。所以,在调试程序时,对暂不使用的语句也可用注释符括起来,使翻译跳过不作处理,待调试结束后再去掉注释符。

11) 源程序中可以有预处理命令(include 命令仅为其中的一种),预处理命令通常应放在源文件或源程序的最前面。

关于预处理命令、由多个函数构成的源程序文件以及由多个源程序文件构成的 C 语言程序将在后面的章节中陆续讲到。作为初学者,首先应掌握只有一个 main 函数组成的 C 语言程序的编程方法与技巧。

1.2.3 书写规则

从书写清晰,便于阅读、理解、维护的角度出发,在书写程序时应遵循以下规则:

1) 一个说明或一个语句占一行。

2) 用“{}”括起来的部分,通常表示了程序的某一层结构。“{}”一般与该结构语句的第一个字母对齐,并单独占一行。

3) 低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写,即采用缩进并对齐的书写方式。以便看起来更加清晰,增加程序的可读性。在编程时应力求遵循这些规则,

以养成良好的编程风格。

1.3 开发与调试

1.3.1 Turbo C 的集成开发环境

Turbo C 是美国 Borland 公司的产品, Borland 公司是一家专门从事软件开发、研制的大公司。该公司相继推出了一套 Turbo 系列软件, 如 Turbo BASIC, Turbo Pascal, Turbo Prolog, 这些软件很受用户欢迎。该公司在 1987 年首次推出 Turbo C 1.0 产品, 其中使用了全新的集成开发环境, 即使用了一系列下拉式菜单, 将文本编辑、程序编译、连接以及程序运行一体化, 大大方便了程序的开发。1988 年, Borland 公司又推出 Turbo C 1.5 版本, 增加了图形库和文本窗口函数库等, 而 Turbo C 2.0 则是该公司 1989 年出版的。Turbo C 2.0 在原来集成开发环境的基础上增加了查错功能, 并可以在 Tiny 模式下直接生成 .COM (数据、代码、堆栈处在同一 64KB 内存中) 文件。还可对数学协处理器 (支持 8087/80287/80387 等) 进行仿真。

Turbo C 2.0 可运行于 IBM-PC 系列微机, 包括 XT、AT 及 IBM 兼容机。此时要求 DOS 2.0 或更高版本支持, 并至少需要 448KB 的 RAM, 可在任何彩、单色 80 列监视器上运行。支持数学协处理器芯片, 也可进行浮点仿真, 这将加快程序的执行。

Turbo C 是在微机上广泛使用的编译程序, 它具有方便、直观、易用的界面和丰富的库函数, 它向用户提供了一个集成环境, 把程序的编辑、编译、连接和运行等操作全部集中在一个界面上进行, 使用十分方便。

为了能使用 Turbo C, 必须先将 Turbo C 的编译程序装入磁盘某一个目录下, 当用户的当前目录是 Turbo C 编译程序所在的子目录时, 只需从键盘键入“tc”即可进入 Turbo C 的集成开发环境中, 屏幕显示如图 1-2 所示。

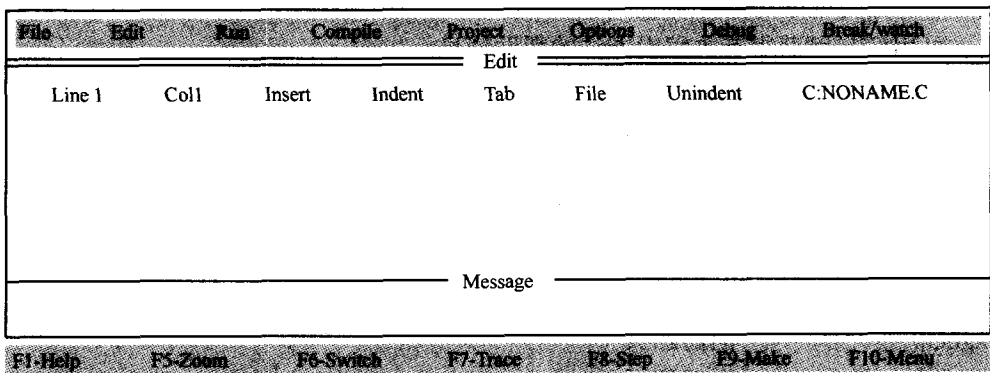


图 1-2 Turbo C 2.0 主屏幕

其中最上面行为 Turbo C 2.0 主菜单, 中间窗口为编辑区, 接下来是信息窗口, 最底下一行为参考行。这四个窗口构成了 Turbo C 2.0 的主屏幕, 以后的编辑、编译、调试以及运行都将在这个主屏幕中进行。下面详细介绍主菜单的内容。

主菜单在 Turbo C 2.0 主屏幕顶上一行, 显示下列内容:

File Edit Run Compile Project Options Debug Break/watch

除 Edit 外,其他各项均有子菜单,只要用<Alt>键加上某项中第一个字母(即大写字母),就可进入该项的子菜单中,如通过<Alt + F>键即可进入 File 菜单。

1. File(文件)菜单

按<Alt + F>键可进入 File 菜单,该菜单包括以下内容:

- Load(加载):装入一个文件,可用类似 DOS 的通配符(如 * .C)来进行列表选择。也可装入其他扩展名的文件,只要给出文件名(或只给路径)即可。该项的热键为<F3>,即只要在主菜单中按<F3>键即可进入该项,而不需要先进入 File 菜单再选此项。
- Pick(选择):将最近装入编辑窗口的 8 个文件列成一个表让用户选择,选择后将该程序装入编辑区,并将光标置在上次修改过的地方。其热键为<Alt + F3>。
- New(新文件):说明文件是新的,默认文件名为 NONAME.C,存盘时可改名。
- Save(存盘):将编辑区中的文件存盘,若文件名是 NONAME.C 时,将询问是否更改文件名,其热键为<F2>。
- Write to(存盘):可由用户给出文件名将编辑区中的文件存盘,若该文件已存在,则询问要不要覆盖。
- Directory(目录):显示目录及目录中的文件,并可由用户选择。
- Change dir(改变目录):显示当前目录,用户可以改变显示的目录。
- Os shell(暂时退出):暂时退出 Turbo C 2.0 到 DOS 提示符下,此时可以运行 DOS 命令,若想回到 Turbo C 2.0 中,只要在 DOS 状态下键入 EXIT 即可。
- Quit(退出):退出 Turbo C 2.0,返回到 DOS 操作系统中,其热键为<Alt + X>。

说明:以上各项可先用光标键移动色棒进行选择,然后键入回车即可执行;也可用每一项的第一个大写字母直接选择。若要退到主菜单或从它的下一级菜单列表框退回均可用<Esc>键, Turbo C 2.0 所有菜单均采用这种方法进行操作,不再赘述。

2. Edit(编辑)菜单

按<Alt + E>键可进入编辑菜单,若再回车,则光标出现在编辑窗口,此时用户可以进行文本编辑。可用<F1>键获得有关编辑方法的帮助信息。

- 与编辑有关的功能键如下:

F1 获得 Turbo C 2.0 编辑命令的帮助信息

F5 扩大编辑窗口到整个屏幕

F6 在编辑窗口与信息窗口之间进行切换

F10 从编辑窗口转到主菜单

- 编辑命令简介:

PageUp 向前翻页

PageDn 向后翻页

Home 将光标移到所在行的开始

End 将光标移到所在行的结尾

Ctrl + Y 删除光标所在的一行

Ctrl + T 删除光标所在处的一个单词

Ctrl + N 插入行

Ctrl + V 或 Ins	插入模式
Ctrl + B 或 Backspace	删除光标左边的字符
Ctrl + G 或 Del	删除光标处的字符
Ctrl + KB	设置块开始
Ctrl + KK	设置块结尾
Ctrl + KV	块移动
Ctrl + KC	块复制
Ctrl + KY	块删除
Ctrl + F1	如果光标所在处为 Turbo C 2.0 库函数, 则获得有关该函数的帮助信息

说明: Turbo C 2.0 在编辑文件时还有一种功能, 就是能够自动缩进, 即光标定位和上一个非空字符对齐。在编辑窗口中, <Ctrl + OL> 为自动缩进开关的控制键。

3. Run(运行)菜单

按 <Alt + R> 键可进入 Run 菜单, 该菜单有以下各项:

- Run(运行程序): 运行由 Project/Project name 项指定的文件名或当前编辑区的文件。如果对上次编译后的源代码未做过修改, 则直接运行到下一个断点(没有断点则运行到结束)。否则先进行编译、连接后才运行, 其热键为 <Ctrl + F9>。
- Program reset(程序重启): 中止当前的调试, 释放分给程序的空间, 其热键为 <Ctrl + F2>。
- Go to cursor(运行到光标处): 调试程序时使用, 选择该项可使程序运行到光标所在行。光标所在行必须为一条可执行语句, 否则提示错误, 其热键为 <F4>。
- Trace into(跟踪进入): 在执行一条调用其他用户定义的子函数时, 若用 Trace into 项, 则执行长条将跟踪到该子函数内部去执行, 其热键为 <F7>。
- Step over(单步执行): 执行当前函数的下一条语句, 即使用户函数调用, 执行长条也不会跟踪进函数内部, 其热键为 <F8>。
- User screen(用户屏幕): 显示程序运行时在屏幕上显示的结果, 其热键为 <Alt + F5>。

4. Compile(编译)菜单

按 <Alt + C> 键可进入 Compile 菜单, 该菜单有以下内容:

- Compile to OBJ(编译生成目标码): 将一个 C 源文件编译生成 .OBJ 目标文件, 同时显示生成的文件名, 其热键为 <Alt + F9>。
- Make EXE file(生成执行文件): 此命令生成一个 .EXE 的文件, 并显示生成的 .EXE 文件名。其中 .EXE 文件名是下面几项之一:
 - ① 由 Project/Project name 说明的项目文件名。
 - ② 若没有项目文件名, 则由 Primary C file 说明的源文件。
 - ③ 若以上两项都没有文件名, 则为当前窗口的文件名。
- Link EXE file(连接生成执行文件): 把当前 .OBJ 文件及库文件连接在一起生成 .EXE 文件。
- Build all(建立所有文件): 重新编译项目里的所有文件, 并进行装配生成 .EXE 文件。该命令不作过时检查(上面的几条命令要作过时检查, 即如果目前项目里源文件的日