

21世纪高职高专计算机系列规划教材

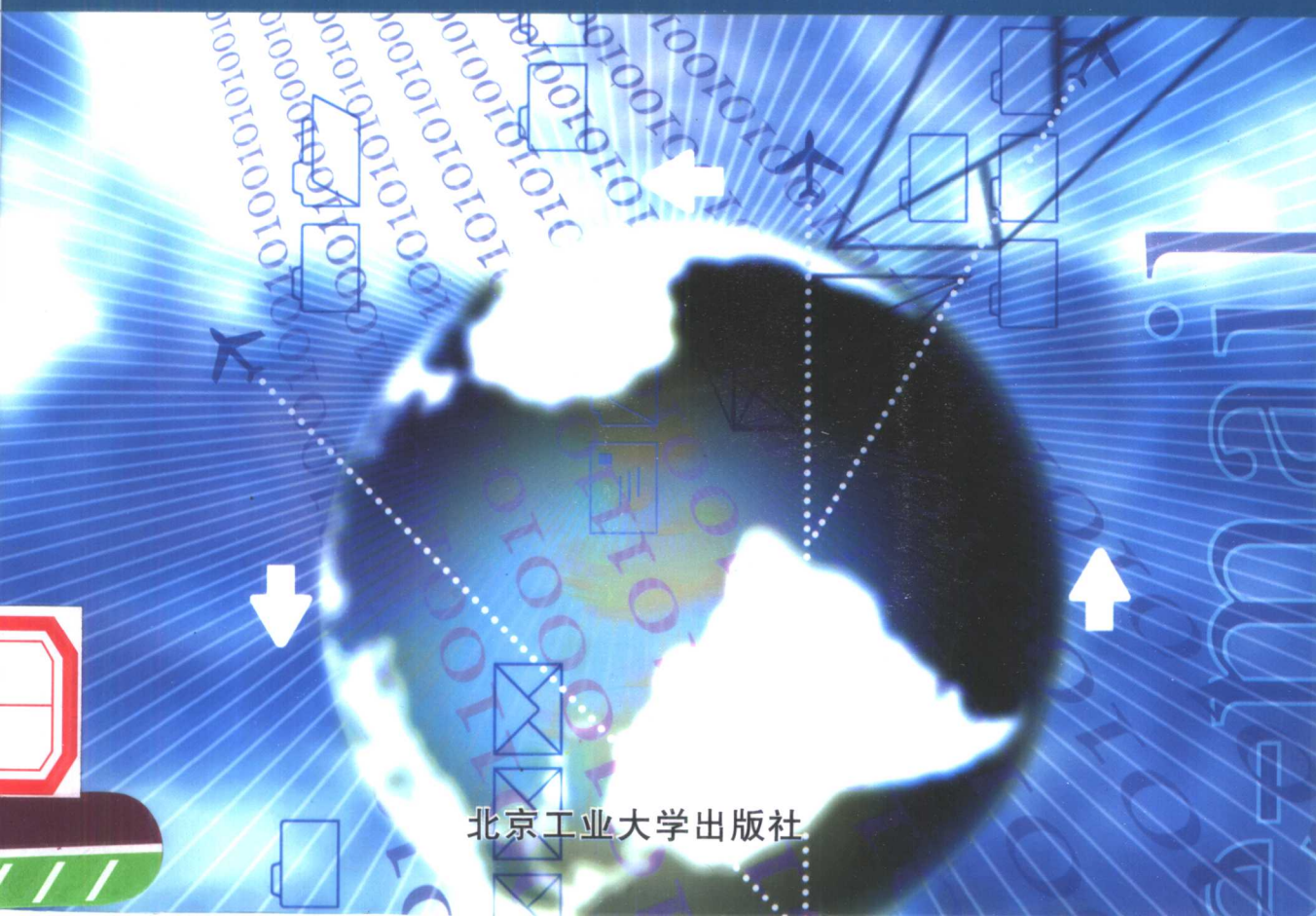
根据教育部最新高职高专教育教学大纲要求编写

C语言

程序设计

武马群 主编

匡松 吕峻闽 缪春池 编著



北京工业大学出版社

高职高专计算机系列规划教材



C 语言程序设计

武马群 主编

匡 松 吕峻闽 缪春池 编著

北京工业大学出版社

内 容 提 要

本书介绍了 C 语言的基本语法、结构和程序设计方法。内容包括 C 语言的基本数据类型、数据的输入与输出、运算符与表达式、结构控制语句、数组、函数、指针、预处理、构造型数据结构、位运算和文件等知识。书中文字流畅、通俗易懂、概念清楚、深入浅出、例题丰富,每章都附有习题和上机实验,用以加深和巩固学生所学的知识。

本书可作为高职高专学生的教材,也可供一般工程技术人员自学使用。

图书在版编目(CIP)数据

C 语言程序设计/匡松,吕峻闽,缪春池编著. —北京:
北京工业大学出版社, 2005.6

(高职高专计算机系列规划教材/武马群主编)

ISBN 7-5639-1520-6

I. C… II. ①匡…②吕…③缪… 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 065742 号

C 语言程序设计

武马群 主编

匡 松 吕峻闽 缪春池 编著

※

北京工业大学出版社出版发行

邮编: 100022 电话: (010) 67392308

各地新华书店总经销

北京东方圣雅印刷有限公司印刷

※

2005 年 6 月第 1 版 2005 年 6 月第 1 次印刷

787mm × 1 092mm 16 开本 印张 17.75 字数 452 千字

印数: 1~5 000 册

ISBN 7-5639-1520-6/T·254

定价: 25.00 元

序

进入 21 世纪以来,随着国民经济发展水平的提高和教育改革的不断深入,我国的职业教育发展迅速,进入了一个新的历史阶段。社会主义现代化建设需要大量高素质的专业人才,而作为我国高等教育重要组成部分的高等职业教育,正肩负着前所未有的使命,为社会主义现代化建设培养大量高素质的劳动者。

区别于传统的本科教育,高等职业教育以培养应用型人才为主。正是基于发展我国高等职业教育的需要,通过大量调研、反复讨论和修改,我们组织了一批长期工作在教学第一线的教师编写了这套《21 世纪高职高专计算机系列规划教材》。

本套教材在编写上具有以下特点:

1. 具有鲜明的高职高专的特点。教材的策划和编写紧密地围绕培养技术应用性专门人才展开,体现了教育部“以应用为目的,以必需、够用为度,以讲清概念、强化应用为教学重点”的教育方针。本套书的作者都是长期从事高职高专教学工作的教师,有着丰富的教学经验,对高职高专学生的认知规律有深入的了解。本套教材适合高等职业学校、高等专科学校、以及本科院校举办的二级职业技术学院和民办职业高校使用。

2. 理论联系实际,强化应用。本套教材章后配有习题和实验题,突出实践技能和动手能力的培养。对于传统的教材,一般按照“提出概念→解释概念→举例说明”这样一种方法,先抽象后具体;本套教材采用“提出问题→解决问题→归纳总结”的方法,先具体后抽象。显而易见,后者更适合高职高专的教学模式,更能培养出具有较强综合职业能力,能够在生产、服务、技术和管理第一线工作的高素质的职业技术专门人才。

3. 适应行业技术发展,体现教学内容的先进性和前瞻性。在教材中注意突出本专业领域的新知识、新技术、新软件,尽可能实现专业教学基础性与先进性的统一。

为了方便教师教学,我们免费为使用本套教材的师生提供电子教学参考资料包:

- ◆ PowerPoint 多媒体课件
- ◆ 习题参考答案
- ◆ 教材中的程序源代码
- ◆ 教材中涉及的实例制作的各类素材

有需要的教师可以登录教学支持网站免费下载。在教材使用中有什么意见或建议也可以直接和我们联系,电子邮件地址: scqcwh@163.com。

希望本套教材,在教学实践的过程中,能够得到教师和学生的欢迎,同时期待得到更多的建议和帮助,以便提高本套教材的质量,更好地为培养社会主义现代化建设的高素质人才服务。

武马群

2005 年 5 月

前 言

C 语言是一种应用十分广泛的编程语言，具有功能丰富、语句简洁、使用方便、语法灵活、数据结构多样、能对硬件进行操作、高移植性和通用性等诸多优点，既有高级语言的特点，又有低级语言的特点，因此被称为“中级语言”，成为编制系统软件和应用软件的首选语言。

C 语言是我国各高校都开设的重要课程，在高职院校的计算机课程设置中，C 语言也是学习程序设计语言的必修课程。在编写本书过程中，作者结合自己多年从事 C 语言教学的经验，理论联系实际，力求通俗易懂。本书在体系结构安排上尽可能将概念、知识点与例题结合起来，每章末均对本章内容进行小结。所选例题比较典型、针对性强，通过一些典型程序将前后的一些知识点联系起来，使读者在对比中理解各种实现方式的特点和异同，能够融会贯通、举一反三。每章均附有必要的习题，让学生加深和巩固所学知识，提高学生的编程能力，培养良好的编程风格和习惯。

全书共 12 章：第 1 章 C 语言概述；第 2 章基本数据类型；第 3 章数据的输入与输出；第 4 章运算符与表达式；第 5 章结构控制语句；第 6 章数组；第 7 章函数；第 8 章指针；第 9 章预处理；第 10 章构造型数据结构；第 11 章位运算；第 12 章文件。本书从 C 语言的基本数据元素、基本语句和结构控制语句、构造数据类型的定义和使用、函数的定义和调用，到指针的灵活运用等主要方面进行了由浅入深的讲解。本书的特点是结构合理、层次分明、例题丰富、通俗易懂、实用性强，适合于初学者使用。

由于作者水平有限，加上时间仓促，书中缺点和错误在所难免，恳请读者批评指正。

编 者

2005 年 5 月

目 录

第1章 C语言概述.....	1
1.1 C语言概述.....	1
1.1.1 C语言的发展历史.....	1
1.1.2 C语言的特点.....	1
1.1.3 Turbo C的特点.....	3
1.2 C程序.....	7
1.2.1 C程序的结构.....	7
1.2.2 C语言程序设计基础.....	10
1.2.3 C语言程序的编制运行过程.....	11
【本章小结】.....	12
【习题】.....	12
第2章 基本数据类型.....	15
2.1 C语言的数据类型.....	15
2.2 常量与变量.....	16
2.2.1 常量.....	16
2.2.2 变量.....	18
2.3 整型数据.....	19
2.3.1 整型常量.....	19
2.3.2 整型变量.....	20
2.4 实型数据.....	23
2.4.1 实型常量.....	23
2.4.2 实型变量.....	23
2.5 字符型数据.....	24
2.5.1 字符型常量.....	24
2.5.2 字符型变量.....	25
2.5.3 字符串常量.....	27
【本章小结】.....	28
【习题】.....	28
第3章 数据的输入与输出.....	32
3.1 数据输出函数.....	32
3.1.1 按格式输出函数 printf.....	32
3.1.2 标准字符输出函数 putchar.....	35

3.1.3 字符串输出函数 puts.....	35
3.2 数据输入函数.....	36
3.2.1 按格式输入函数 scanf.....	36
3.2.2 标准字符输入函数 getchar.....	39
3.2.3 字符串输入函数 gets.....	41
【本章小结】	41
【习题】	42
第4章 运算符与表达式.....	46
4.1 运算符与表达式概述.....	46
4.1.1 表达式的组成.....	46
4.1.2 运算符的分类.....	46
4.2 算术运算符与表达式.....	47
4.3 关系运算符与表达式.....	49
4.4 逻辑运算符与表达式.....	51
4.5 其他运算符与表达式.....	52
4.5.1 赋值运算符与表达式.....	52
4.5.2 条件运算符与表达式.....	54
4.5.3 逗号运算符与表达式.....	55
4.5.4 强制类型转换运算符与表达式.....	55
4.6 运算符的优先级.....	56
【本章小结】	58
【习题】	58
第5章 结构控制语句.....	62
5.1 程序的基本控制结构.....	62
5.1.1 顺序结构.....	62
5.1.2 选择结构.....	64
5.1.3 循环结构.....	64
5.2 条件选择语句.....	64
5.2.1 if 语句的形式.....	65
5.2.2 if 语句的嵌套.....	67
5.3 多路分支语句.....	73
5.4 循环语句.....	76
5.4.1 while 型循环语句.....	76
5.4.2 do...while 型循环语句.....	78
5.4.3 for 型循环语句.....	79
5.4.4 goto 型循环语句.....	83
5.4.5 continue 语句和 break 语句.....	83

5.4.6 循环嵌套	86
【本章小结】	89
【习题】	89
第6章 数组	95
6.1 一维数组	95
6.1.1 一维数组的定义	95
6.1.2 一维数组的初始化	96
6.1.3 一维数组元素的使用	96
6.2 字符数组	102
6.2.1 字符数组的定义	102
6.2.2 字符数组的初始化	102
6.2.3 字符数组的使用	103
6.2.4 字符串处理函数	104
6.3 二维数组	110
6.3.1 二维数组的定义	110
6.3.2 二维数组的初始化	111
6.3.3 二维数组的使用	112
【本章小结】	117
【习题】	117
第7章 函数	123
7.1 用户自定义函数	123
7.1.1 函数的定义	123
7.1.2 函数的调用	124
7.1.3 函数调用中的数据传递	127
7.2 函数的嵌套调用和递归调用	135
7.2.1 函数的嵌套调用	135
7.2.2 函数的递归调用	136
7.3 变量的作用域和生存期	139
7.3.1 变量的作用域	139
7.3.2 变量的生存期	144
7.4 函数的作用范围	148
7.4.1 内部函数	149
7.4.2 外部函数	149
7.5 命令行参数	150
7.6 多个文件的连接和运行	151
【本章小结】	152
【习题】	153

第 8 章 指针	160
8.1 指针与指针变量	160
8.1.1 地址、指针与指针变量	160
8.1.2 指针变量的使用	162
8.2 指针与数组	166
8.2.1 指向一维数组的指针	166
8.2.2 指向二维数组的指针	170
8.2.3 指向字符串的指针	176
8.3 指向特殊类型数据的指针变量	178
8.3.1 指针数组	178
8.3.2 指向指针的指针	181
8.4 指针与函数	183
8.4.1 指针变量作为函数参数传递	183
8.4.2 返回指针的函数	186
8.4.3 指向函数的指针	189
【本章小结】	192
【习题】	192
第 9 章 预处理	199
9.1 宏定义	199
9.1.1 不带参数的宏定义	199
9.1.2 带参数的宏定义	200
9.2 文件包含	201
9.3 条件编译	202
9.3.1 #ifdef	202
9.3.2 #ifndef	203
9.3.3 #if	204
【本章小结】	205
【习题】	205
第 10 章 构造型数据结构	209
10.1 结构体	209
10.1.1 结构体的定义	209
10.1.2 结构体变量的使用	211
10.1.3 结构体数组	212
10.1.4 指向结构体的指针	213
10.1.5 结构体变量作为参数在函数之间进行传递	215
10.2 指针与链表	218
10.2.1 链表的定义	218

10.2.2 链表的操作	219
10.3 公用体	224
10.3.1 公用体的定义	224
10.3.2 公用体变量的引用	225
10.4 枚举类型	227
10.4.1 枚举类型的定义	227
10.4.2 枚举类型变量的使用	228
10.5 自定义类型	230
【本章小结】	231
【习题】	231
第11章 位运算	237
11.1 位运算符	237
11.1.1 按位与	237
11.1.2 按位或	238
11.1.3 按位异或	238
11.1.4 按位取反	239
11.1.5 按位左移	239
11.1.6 按位右移	240
11.1.7 位运算复合赋值运算符	240
11.2 位段	242
【本章小结】	243
【习题】	243
第12章 文件	248
12.1 文件概述	248
12.1.1 文件的存储方式	248
12.1.2 文件类型指针 (FILE 类型指针) 概念	248
12.2 文件的打开和关闭	249
12.2.1 打开文件函数 fopen	249
12.2.2 关闭文件函数 fclose	250
12.3 文件的读写	250
12.3.1 字符的读取和写入	250
12.3.2 字符串的读取和写入	252
12.3.3 按格式读取和写入	253
12.4 随机文件的读写	254
12.4.1 随机文件的读取	254
12.4.2 随机文件的写入	254
12.5 文件定位函数	256

12.5.1	rewind 函数	256
12.5.2	fseek 函数	257
12.5.3	ftell 函数	258
【本章小结】		258
【习题】		258
附录一	常用的 ASCII 码表	263
附录二	常见的 Turbo C 2.0 编译错误信息	265
附录三	常用的 Turbo C 2.0 库函数	268

第 1 章 C 语言概述

C 语言自出现到现在经历了三十多年的时间，是使用最广泛的程序设计语言之一，本章首先介绍 C 语言的基本发展历史、版本情况以及主要特点，然后介绍 C 语言程序的基本编制环境和步骤，使大家在简单程序的编制过程中初步掌握 C 程序的构成、特点和编制规范。

1.1 C 语言概述

1.1.1 C 语言的发展历史

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 (Bootstrap Combined Programming Language) 发展演变而来。

在 1970 年，AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出较先进的 B 语言，并用 B 语言写了第一个 UNIX 操作系统。1972 年，贝尔实验室的 Ritchie 在 B 语言的基础上设计出了 C 语言，取的是 BCPL 语言的第二个字母。到了 1973 年，两人又用 C 语言一起改写了 UNIX 操作系统 90% 以上的代码，也就是 UNIX 第 5 版本。1978 年，Ritchie 和 Brian Kernighan 编写了《The C Programming Language》，并于 1988 年作了修订，该书就是 C 语言版本的基础，被称为标准 C。

随着计算机的日益普及，出现了许多 C 语言版本，由于 C 语言的高可移植性的特点，使得 UNIX 系统在 AT&T、VAX 等计算机上能够迅速实现。但是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，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，它包括了 1983 年和 1987 年两个版本。到了 1990 年，国际标准化组织 (International Standard Organization, ISO) 接受了 87 ANSI C 作为 ISO C 的标准，即 ISO 9899—1990。目前比较流行的 C 语言的编译系统都是以此作为基础的，如 Turbo C、Borland C 和 Microsoft C 等。

1.1.2 C 语言的特点

C 语言发展如此迅速，并且成为最受欢迎的语言之一，主要因为它具有强大的功能。C 语言的特点有：

(1) C 语言提供的语句简洁，使用方便，格式紧凑，语法灵活。C 语言共有 32 个关键字，9 种控制语句，语句简练，书写自由。以下关键字是由系统定义的，不能用作其他定义。

① C 语言中的 32 个关键字如下。

● 与数据类型相关的关键字：

char int short long signed unsigned float
double enum struct union typedef void

● 与存储类型相关的关键字：

auto register static extern

● 与控制语句相关的关键字：

if else switch case default do
while for break continue goto return

● 其他的关键字：

const sizeof volatile

② C 语言中的 9 种控制语句如下。

if(...) ... else ... switch(...) ... case for(...) ... while(...) ...
do ... while(...) continue break goto return

(2) C 语言的运算符十分丰富，共有 34 种运算符，包括算术、关系、逻辑、位、赋值、指针、条件、逗号、下标、类型转换运算符等多种类型。

(3) 数据结构多样，有整型、实型、字符型、枚举类型等基本类型，有数组、结构体、共用体等构造类型以及指针类型，还为用户提供了自定义数据类型，特别是引入了指针概念，能够实现复杂的数据结构。

(4) C 语言的控制语句形式多样、使用方便。有两路分支、多路分支和循环结构几种控制语句，便于结构化模块的实现和控制，便于程序的编制和维护。

(5) C 语言是一种模块化的程序设计语言，采用自顶向下、逐步求精的结构化程序设计方法，各模块功能独立，以函数形式编制，通过函数之间的相互调用和数据传递，实现系统整体的功能要求。这样，把大型系统的实现化整为零，便于分工合作以及共享。

C 程序中，所有的关键字都是小写字母，在 C 程序里大小写是敏感的，一般习惯使用小写字母作为变量名称和字符串常量。C 语言书写格式自由，不使用行号，可以使用空格和空行，但习惯上采用锯齿形的书写格式。良好的书写习惯，是优秀的程序员必备的素质之一。例如，使用 Tab 键缩进，花括号对齐，在适当的地方添加空行以及添加必要的注释。

(6) C 语言可以直接访问地址、进行位运算，从而能对硬件进行操作，因此 C 语言既具有高级语言编写简单方便、便于理解的优点，又具有低级语言与硬件结合紧密的优点。因此 C 语言被称为介于高级语言和低级语言之间的中级语言。

(7) C 语言具有很强的移植性，由 C 语言编写的程序基本不用太多的修改就可以用于不同型号的计算机上，程序和硬件的匹配由 C 语言的编译程序来完成，同时也可以在各种操作系统下使用。

(8) C 语言具有很好的通用性，既可用于编写应用软件，也适合编写系统软件。例如 UNIX 操作系统的源代码就是用 C 语言编制的。

1.1.3 Turbo C 的特点

1. Turbo C 的发展

Turbo C 是由美国的 Borland 公司开发的产品，从 1987 年推出 Turbo C 1.0 以来，该公司相继推出了一系列 Turbo C 版本。Turbo C 1.0 拥有一套集成开发环境，以菜单的方式把程序编制过程中要使用的命令分门别类地集中起来，提供给用户方便清晰的编程环境。1988 年，Borland 公司又推出 Turbo C 1.5 版，增加了图形库处理和文本窗口函数库等功能。1989 年推出 Turbo C 2.0，增加了纠错功能，可以在微型模式下直接生成可执行文件。此后，随着图形化操作系统的普及和推广，程序编制的模式也发生了较大的变化，出现了面向对象的程序设计方法，Borland 公司在 Turbo C 2.0 的基础上推出了面向对象的程序设计语言 Turbo C++，把 Turbo C 的集成开发环境与面向对象的程序设计结合起来，后来又发展为 Borland C++。

2. Turbo C 2.0 的组成

Turbo C 2.0 的系统文件比较精干，主要包含以下一些文件，如表 1-1 所示。

表 1-1 Turbo C 2.0 的文件

文件 名	说 明	文 件 名	说 明
INSTALL.EXE	安装程序文件	TLINK.EXE	连接器
TC.EXE	集成编译文件	TLIB.EXE	库管理工具
TCINST.EXE	系统配置文件	C0?.OBJ	不同模式的启动代码
TCHELP.TCH	帮助文件	C?.LIB	不同模式的运行库
THELP.COM	读取帮助的驻留程序	GRAPHICS.LIB	图形库
README	信息文件	EMU.LIB	8087 仿真库
TCCONFIG.EXE	配置文件转换程序	FP87.LIB	8087 库
MAKE.EXE	项目管理工具	*.H	Turbo C 头文件
TCC.EXE	命令行编译程序	*.BGI	图形驱动程序
*.C	Turbo C 源文件	*.OBJ	目标程序

表 1-1 中的“?”可以有以下的取值，其取值及表示的含义如下：

- T: Tiny (微型模式)
- S: Small (小模式)
- C: Compact (紧凑模式)
- M: Medium (中型模式)
- L: Large (大模式)
- H: Huge (巨大模式)

3. Turbo C 2.0 的安装和启动

Turbo C 2.0 的安装非常简单，只要将 1#盘插入软盘驱动器中，在 DOS 的“A>”提示符后键入“INSTALL”然后回车即可，此时屏幕上显示以下三种选择：

- (1) 在硬盘上创建一个新目录来安装整个 Turbo C 2.0 系统。

(2) 对 Turbo C 1.5 更新版本。这样的安装将保留原来对选择项、颜色和编辑功能键的设置。

(3) 为只有两个软盘而无硬盘的系统安装 Turbo C 2.0。

这里假定按第一种选择进行安装，只要在安装过程中按提示，顺序插入各个软盘，就可以顺利地进行安装，安装完毕将在 C 盘根目录下建立一个 TC 子目录，TC 下还建立了两个子目录 LIB 和 INCLUDE，LIB 子目录中存放库文件，INCLUDE 子目录中存放所有头文件。

如果不是上述的安装盘，则直接将文件拷贝到相应的目录下即可。

Turbo C 2.0 可以在 DOS 和 Windows 环境下运行。运行 Turbo C 2.0 时，只要在 TC 子目录下运行主程序 tc.exe 即可进入 Turbo C 2.0 集成开发环境。

在 DOS 环境下，如果当前目录是 TC 子目录，键入“tc”后回车即可启动，例如：

```
C:\TC>tc
```

如果当前目录不是 TC 子目录，则应先改变当前目录为 TC 子目录，或在“tc”之前加上路径即可启动，例如：

```
C>CD TC
```

```
C:\TC>tc
```

或

```
C>c:\TC\tc
```

在 Windows 环境下，只要通过“我的电脑”或“资源管理器”，找到 Turbo C 2.0 系统文件夹的位置，进入文件夹，双击 tc.exe 文件，就可以启动进入 Turbo C 2.0 集成环境。

4. Turbo C 2.0 的集成开发环境

进入 Turbo C 2.0 集成开发环境后，屏幕显示如图 1-1 所示。

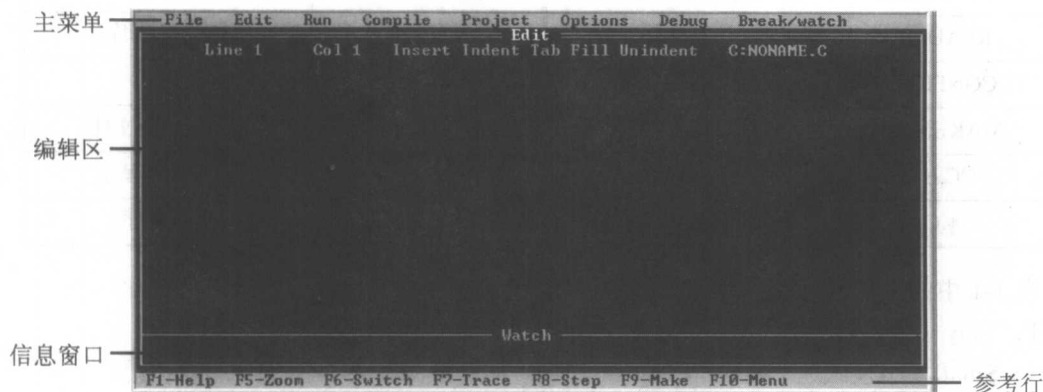


图 1-1 Turbo C 2.0 主窗口

其中最上面一行是 Turbo C 2.0 的主菜单，中间窗口为编辑区，接下来是信息窗口，最下面一行为参考行。这几个部分构成了 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 主屏幕的最上一行，显示的菜单项有 File、Edit、Run、Compile、Project、Options、Debug、Break/watch。除 Edit 外，其他各项均有子菜单，用 Alt 键加上某项中第一个字母（即大写字母）；或者按 F10 键，再按光标键选择，就可进入菜单。以上各项可用光标键移动色条进行选择，回车则执行。也可以用每一菜单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，以后不再说明。

Turbo C 2.0 的主菜单包括 8 个子菜单，下面简单介绍一下各子菜单的基本功能，以及一些常用菜单命令的位置和功能。

(1) File (文件) 菜单：按 Alt+F 组合键可进入 File 菜单，主要进行一些与文件相关的操作。该菜单包括以下命令。

- 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。该命令的快捷键为“F3”，即只要在主菜单中按“F3”即可进入该命令，而不需要先进入 File 菜单再选择。
- 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，选择后将该文件装入编辑区，并将光标置在上次修改过的地方。其快捷键为 Alt+F3。
- New (新文件)：新建一个文件，缺省文件名为 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，其快捷键为 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，其快捷键为 Alt+X。

(2) Edit (编辑) 菜单：按 Alt+E 组合键可进入编辑菜单，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。用 F1 键可以获得有关编辑方法的帮助信息。

1) Turbo C 2.0 的双界符包括以下几种符号：花括号{和}，尖括号<和>，圆括号(和)，方括号[和]，注释符/*和*/，双引号"，单引号'。

2) 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 组合键可进入 Run 菜单，主要用于程序的运行和跟踪调试。该菜单有以下命令。

- Run (运行程序)：运行由 Project 菜单中 Project name 命令指定的文件名或当前编辑区的文件。如果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，其快捷键为 Ctrl+F9。
- Program reset (程序重启)：中止当前的调试，释放分给程序的空间，其快捷键为 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，其快捷键为 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 命令，则执行长条将跟踪到该子函数内部去执行，其快捷键为 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, 其快捷键为 F8。

- User screen (用户屏幕): 显示程序运行时在屏幕上显示的结果。其快捷键为 Alt+F5。

(4) Compile (编译) 菜单: 按 Alt+C 组合键可进入 Compile 菜单, 主要用于程序的编译。该菜单有以下命令。

- 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, 其快捷键为 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。
- 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。其快捷键为 F9。
- 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期和时间与目标文件相同或更早, 则拒绝对源文件进行编译)。
- Primary C file (主 C 文件): 当在该命令中指定了主文件后, 在以后的编译中, 如没有项目文件名则编译此命令规定的主 C 文件; 如果编译中有错误, 则将此文件调入编辑窗口, 不管目前窗口中是不是主 C 文件。
- Get info: 获得有关当前路径、源文件名、源文件字节大小、编译中的错误数目、可用空间等信息。

(5) Project (项目) 菜单: 按 Alt+P 组合键可进入 Project 菜单, 主要用于工程管理和多个文件的连编。该菜单包括以下命令。

Project name (项目名): 项目名具有 .prj 的扩展名, 其中包括将要编译、连接的文件名。例如, 有一个程序由 file1.c、file2.c 和 file3.c 组成, 要将这 3 个文件编译装配成一个 file.exe 的执行文件, 可以先建立一个名为 file.prj 的项目文件, 将要连编的文件名写入该文件, 每一个文件名各占一行, 内容如下:

```
file1.c
file2.c
file3.c
```

然后将项目文件名 file.prj 写入 Project name 项中, 以后进行编译时将自动对项目文件中规定的 3 个源文件分别进行编译, 最后连接成 file.exe 文件。

如果其中有些文件已经编译成 .obj 文件, 而又没有修改过, 可直接写上 .obj 扩展名。此时将不再编译而只进行连接。例如, 项目文件的内容为:

```
file1.obj
file2.c
file3.c
```

将不对 file1.c 进行编译, 而直接连接。当项目文件中的每个文件无扩展名时, 默认为 .c 源文件。另外, 其中的文件也可以是库文件, 但必须写上扩展名 .lib。

- Break make on (中止编译): 由用户选择是否在有 Warning (警告)、Errors (错误)、Fatal Errors (致命错误) 时或 Link (连接) 之前退出 Make 编译。
- Auto dependencies (自动依赖): 当开关置为 on, 编译时将检查源文件与对应的 .obj 文件日期和时间, 否则不进行检查。