

21世纪计算机科学与技术系列教材

高职高专

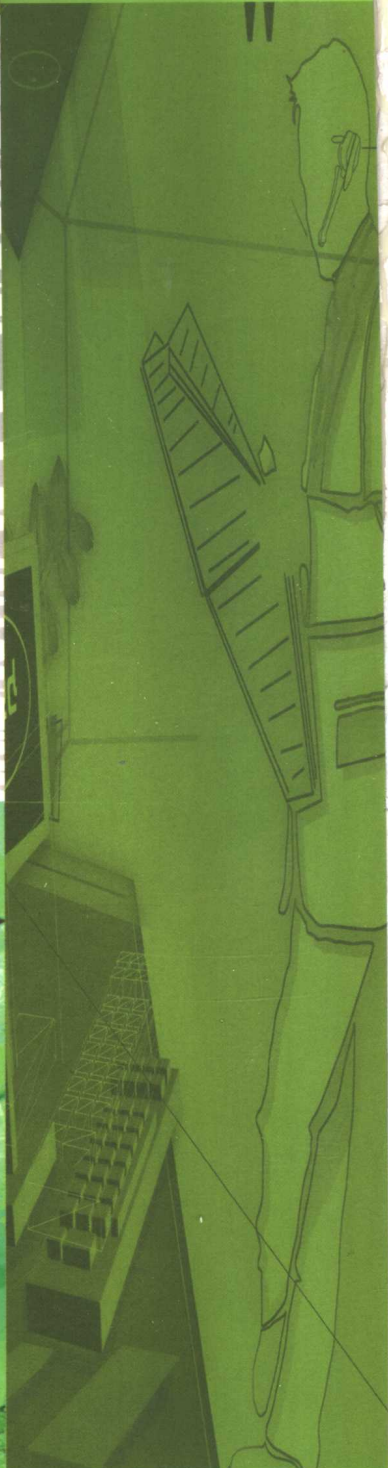
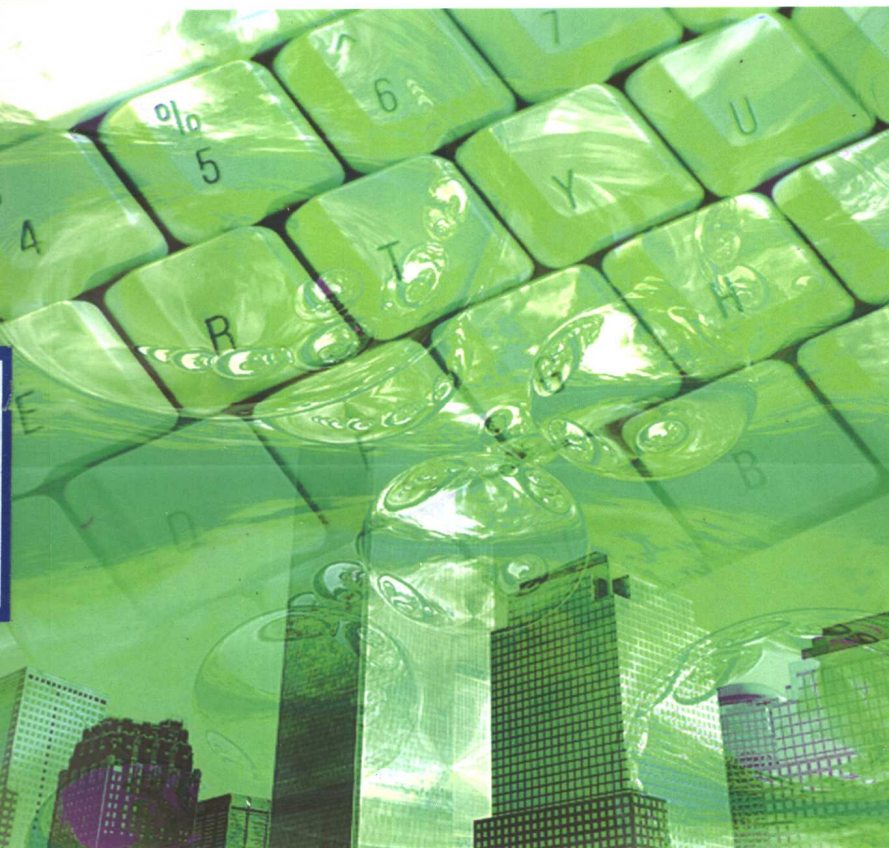
# 易学易教 C 语言

## —— C 语言程序设计基础

主 编 朱建芳 李学军

副主编 罗移祥

主 审 柳 青



华南理工大学出版社

21 世纪计算机科学与技术系列教材(高职高专)

# 易学易教 C 语言

## ——C 语言程序设计基础

主 编 朱建芳 李学军

副主编 罗移祥

主 审 柳 青

华南理工大学出版社

·广州·

## 内 容 简 介

本书详细介绍了C语言的语法知识和如何利用C语言进行程序设计,另外还介绍了程序设计的基本方法以及数据结构与算法的基础知识。主要内容包括:C语言概述、Turbo C的操作使用、C语言的基础知识、3种基本结构的C语言程序设计、程序设计的基本方法、C语言模块化程序设计、数据结构与算法的基础知识和C语言的综合应用。每一章后面均配有丰富的练习题,并提供紧扣本章理论知识的上机实践内容。

本书是“21世纪计算机科学与技术系列教材(高职高专)”之一,主要面向高职高专院校计算机类专业及其他工科类相关专业的学生,也可作为全国计算机等级考试(二级C)的辅导教材,同时也可以作为其他人员学习C语言的自学教材和参考书。

### 图书在版编目(CIP)数据

易学易教C语言: C语言程序设计基础/朱建芳,李学军主编. —广州:华南理工大学出版社, 2005.2

(21世纪计算机科学与技术系列教材(高职高专))

ISBN 7-5623-2145-0

I. 易… II. ①朱…②李… III. C语言-程序设计-高等学校-教材  
IV. TP312

中国版本图书馆CIP数据核字(2004)第129025号

总发行:华南理工大学出版社(广州五山华南理工大学17号楼,邮编510640)

发行部电话:020-87113487 87111048(传真)

E-mail:scut202@scut.edu.cn

http://www.scutpress.com

责任编辑:欧建岸(ouja2@163.com)

印刷者:广东省阳江市教育印刷公司

开本:787×960 1/16 印张:25 25 印数:509千

版次:2005年2月第1版第1次印刷

印数:1~3000册

定 价:36.50元

版权所有 盗版必究

## 编写委员会

顾问：赵仕滨

主任：苏运霖

副主任：柳青

委员：(按姓氏笔画为序)

王敏 任淑美 朱定善 何超 何瑞麟  
吴卫祖 李洛 李克勤 李学军 杨宇  
杨小元 陈晓辉 卓尚富 贺平 雷渭倡

## 审稿委员会

主任：柳青

副主任：贺平

委员：(按姓氏笔画为序)

朱定善 何文华 吴卫祖 李洛  
苏运霖 赵仕滨 陈晓辉 宋专茂

策划编辑：欧建岸 赖淑华 王魁葵

# 前 言

近年来,C语言已经发展成为应用最广泛的计算机程序设计语言之一,是程序设计工作者必须熟练掌握的一种语言工具。高等院校几乎所有的计算机类专业和许多理工科专业都开设了C语言程序设计课程。全国计算机等级考试、全国计算机应用技术证书考试和全国大学生计算机统一考试(专升本、研究生入学考试等)都将C语言作为一个重要的考核内容。因此,学好C语言成为一种越来越普遍的需要。

本书的编写目的是为C语言的初学者和教授C语言的教师们提供一本易学易教的C语言程序设计基础教材,面向高职高专计算机类各个专业及相关专业的学生和教育工作者。

本书共分13章,每章均由学习目标、教学指引、正文、习题和实验五部分组成。读者通过阅读“学习目标”可明确本章的学习目标,增强学习的积极性和主动性;教师通过阅读“教学指引”可快速明了本章的重点、难点、学时安排以及实验和作业要求等;习题部分提供了与本章及已学内容密切相关的丰富的练习题,以检验学习效果;实验部分则提供紧扣本章理论知识的上机实践内容,以达到加深理解和学以致用的目的。

本书内容的组织和编排由浅入深、由简单到复杂、由局部到系统,层次分明,在重要的知识点后给出“注意”提示以加深读者的印象,在很多例题后给出“想一想”提问,引导读者思考,以期达到开拓思维、举一反三和融会贯通的效果。

语法知识讲解清晰详尽,例题丰富,举例以通俗易懂、贴近生活和富有趣味性的内容为主,淡化在复杂的数值计算方面的应用,便于自学,特别适合高职高专层次的读者和计算机编程语言的初学者。

本书在讲解C语言语法知识的同时强调程序设计能力的培养。在应用编程例题后给出“思路导航”,指引读者如何分析问题、组织数据、设计算法和编写程序以及运行测试等。另外,书中还结合C语言程序设计需要介绍了程序设计的基本方法和数据结构与算法的知识,并进行应用案例分析。

最后一章阐述了一个综合应用案例的完整的分析设计过程,使读者在学完了C语言后懂得如何应用C语言编程解决实际问题。该章的实验部分可作为课程设计的内容。

本教材讲课时数为54~72学时,上机时数为18~30学时。教师可根据学时数、专业和学生的实际情况选讲书中第13章的部分内容,对于书中各章的实验,也

可以根据实际情况选做或不做部分内容。

本书由朱建芳、李学军主编。其中第1、7、11、13章(13.4节除外)由朱建芳编写,第2、3、4、5、6章和第13.4节由李学军编写,第8、9、10、12章由罗移祥编写。全书由朱建芳负责统稿和定稿。

在本教材的编写过程中,柳青副教授对全书认真进行了审阅,并提出了许多宝贵意见。在此表示衷心的感谢!

另外,特别感谢苏运霖教授认真审阅了本书的编写大纲并提出了宝贵的修改意见;感谢李克勤老师为本书的编写提了许多好的建议;还要感谢广州航海高等专科学校和华南理工大学出版社对本书的编写和出版给予的大力支持。

本书是编者在总结多年从事C语言应用编程与教学经验的基础上,参考大量关于C语言的教科书、应用编程指导书以及全国计算机等级考试资料编写而成,精选例题和习题,并且书中所有源程序都在Turbo C 2.0环境下调试通过。

由于水平有限,书中难免有错误和疏漏之处,敬请读者批评指正。

编 者

2005年2月

# 目 录

1 C语言概述 .....	1
1.1 C语言的发展简史 .....	1
1.2 C语言的特点 .....	2
1.3 为什么选择C语言 .....	3
1.4 认识C程序 .....	4
1.4.1 一个简单的C程序 .....	4
1.4.2 C程序的基本结构 .....	5
1.4.3 C程序中的单词 .....	9
1.4.4 如何编写可读性良好的C程序 .....	11
1.5 C程序的运行与调试 .....	14
1.5.1 C程序设计的基本步骤 .....	14
1.5.2 如何使用Turbo C2.0运行与调试C程序 .....	16
练习与思考 .....	23
实验一 .....	24
2 基本数据类型、运算符与表达式 .....	27
2.1 C语言的数据类型 .....	27
2.2 常量与变量 .....	28
2.2.1 常量 .....	29
2.2.2 变量 .....	29
2.3 整型数据 .....	31
2.3.1 整型常量 .....	31
2.3.2 整型变量 .....	32
2.4 实型数据 .....	33
2.4.1 实型常量 .....	33
2.4.2 实型变量 .....	33
2.5 字符型数据 .....	34
2.5.1 字符常量 .....	34
2.5.2 字符变量 .....	35
2.5.3 字符串常量 .....	36
2.6 各类数据混合运算 .....	37

2.7 运算符	39
2.7.1 算术运算符和算术表达式	39
2.7.2 自增与自减运算符	41
2.7.3 赋值运算符和赋值表达式	43
2.7.4 关系运算符及其表达式	45
2.7.5 逻辑运算符及其表达式	46
2.7.6 逗号运算符及其表达式	48
2.7.7 位运算符及其表达式	49
2.8 运算符的优先级及其结合性	53
练习与思考	54
实验二	59
3 C 语句和简单的 C 程序设计	62
3.1 C 语句概述	62
3.2 程序的三种基本结构	64
3.3 赋值语句	67
3.4 数据输入输出	67
3.4.1 格式化输出函数 printf()	68
3.4.2 单个字符的输出函数 putchar()	74
3.4.3 格式化输入函数 scanf()	75
3.4.4 单个字符输入函数 getchar()	78
3.5 顺序结构的 C 程序设计举例	79
练习与思考	81
实验三	85
4 选择结构程序设计	88
4.1 if 语句	88
4.2 switch 语句	98
4.3 选择结构程序设计举例	101
练习与思考	105
实验四	110
5 循环结构的 C 语言程序设计	112
5.1 循环语句	112
5.1.1 goto 语句	113
5.1.2 while 语句	116
5.1.3 do··while 语句	117
5.1.4 for 语句	120



---

5.2 break 语句和 continue 语句	123
5.2.1 break 语句	123
5.2.2 continue 语句	124
5.2.3 break 语句和 continue 语句的区别和联系	124
5.3 循环的嵌套	125
5.4 几种循环的比较	130
5.5 循环结构的 C 程序设计举例	131
练习与思考	135
实验五	139
<b>6 数组和串</b>	<b>141</b>
6.1 一维数组的定义和引用	142
6.2 二维数组的定义和引用	147
6.3 字符数组与字符串	153
6.4 数组应用举例	160
练习与思考	164
实验六	169
<b>7 函 数</b>	<b>171</b>
7.1 库函数	172
7.2 函数的定义	173
7.3 函数的返回值	175
7.4 函数的调用	179
7.4.1 函数调用的一般形式	179
7.4.2 函数间的参数传递	180
7.4.3 函数调用的方式	180
7.4.4 函数的声明	181
7.4.5 函数的递归调用	183
7.4.6 数组作为函数的参数	188
7.5 由多个函数组成的 C 程序设计举例	190
练习与思考	193
实验七	196
<b>8 指针</b>	<b>197</b>
8.1 什么是指针	198
8.2 指针和指针变量	199
8.2.1 指针变量的定义与初始化	199
8.2.2 指针变量的引用	200

8.2.3	指针变量作为函数参数 .....	202
8.3	数组指针 .....	204
8.3.1	指向数组元素的指针 .....	204
8.3.2	通过指针引用数组元素 .....	205
8.3.3	数组名作函数参数 .....	207
8.3.4	指向二维数组的指针和指针变量 .....	208
8.4	字符串指针 .....	213
8.4.1	字符串的指针表示和引用 .....	213
8.4.2	字符串指针作函数参数 .....	215
8.5	函数指针 .....	216
8.5.1	指向函数的指针变量 .....	216
8.5.2	用函数指针调用函数 .....	216
8.5.3	用指向函数的指针作函数参数 .....	217
8.6	返回指针值的函数 .....	219
8.7	指针数组和指向指针的指针 .....	220
8.7.1	指针数组 .....	221
8.7.2	指向指针的指针 .....	223
8.7.3	指针数组作 main()函数的形参 .....	225
8.8	指针应用举例 .....	226
	练习与思考 .....	230
	实验八 .....	232
9	结构体与共用体 .....	234
9.1	结构体类型与结构体变量的定义 .....	235
9.1.1	结构体类型的定义 .....	235
9.1.2	结构体类型变量的定义 .....	236
9.2	结构体变量的引用与初始化 .....	238
9.2.1	结构体类型变量的引用 .....	238
9.2.2	结构体类型变量的初始化 .....	239
9.3	结构体数组 .....	240
9.3.1	结构体数组的定义 .....	240
9.3.2	结构体数组的初始化 .....	241
9.3.3	结构体数组应用举例 .....	241
9.4	指向结构体类型数据的指针 .....	243
9.4.1	指向结构体变量的指针 .....	243
9.4.2	指向结构体数组的指针 .....	245

9.5 共用体 .....	247
9.5.1 共用体类型的定义 .....	247
9.5.2 共用体变量的定义 .....	247
9.5.3 共用体变量的引用 .....	249
9.6 枚举类型 .....	251
9.7 用 typedef 定义类型 .....	253
9.8 结构体应用举例 .....	254
练习与思考 .....	261
实验九 .....	264
10 编译预处理 .....	266
10.1 宏定义 .....	266
10.2 文件包含 .....	269
10.3 条件编译 .....	271
练习与思考 .....	273
实验十 .....	273
11 C 语言的模块化程序设计 .....	275
11.1 程序设计的基本方法 .....	275
11.1.1 自顶向下、逐步求精的设计过程 .....	276
11.1.2 模块化程序设计 .....	279
11.1.3 结构化编程 .....	280
11.2 C 语言对模块化编程的支持 .....	280
11.2.1 局部变量与全局变量 .....	280
11.2.2 变量的存储类型 .....	284
11.2.3 内部函数与外部函数 .....	289
11.3 多个源程序文件组成的 C 程序设计案例分析 .....	292
练习与思考 .....	295
实验十一 .....	298
12 文件 .....	299
12.1 C 语言文件概述 .....	300
12.1.1 文件的概念及分类 .....	300
12.1.2 文件的操作流程 .....	301
12.1.3 缓冲文件系统与非缓冲文件系统 .....	301
12.1.4 文件指针 .....	302
12.2 文件的打开与关闭 .....	303
12.2.1 文件的打开 .....	303

12.2.2	文件的关闭	304
12.3	文件的读写	305
12.3.1	读写一个字符	305
12.3.2	读写一个字符串	307
12.3.3	读写一个数据块	309
12.3.4	对文件进行格式化读写	311
12.4	文件定位	313
12.4.1	rewind()和 fseek()函数	313
12.4.2	返回文件当前位置 ftell()函数	316
12.5	文件的出错检测	316
	练习与思考	317
	实验十二	318
13	C 语言的综合应用	321
13.1	数据结构与算法的概念	321
13.1.1	数据结构	322
13.1.2	算法	326
13.2	顺序表的插入与删除	330
13.2.1	顺序表的定义	330
13.2.2	顺序表的插入和删除	331
13.3	链表的插入与删除	336
13.3.1	链表的定义	336
13.3.2	单链表的建立、插入与删除	337
13.4	绘制图形实例	344
13.4.1	图形程序设计步骤	344
13.4.2	图形程序设计基础	344
13.4.3	设置和关闭图形模式的函数	345
13.4.4	常见图形模式下的图形函数	347
13.4.5	简单 C 图形程序设计举例	362
13.5	综合应用案例分析	363
	练习与思考	382
	实验十三	384
附录一	ASCII 码表	385
附录二	C 语言的词法、句法和章法提要	386
附录三	Turbo C 常用库函数	387
参考文献		392

# 1 C 语言概述

## 【学习目标】

读者可通过本章的学习,达到以下目的:

- 了解 C 语言的发展简史和特点;
- 掌握 C 语言的基本结构及如何编写可读性良好的 C 程序;
- 掌握一个 C 程序的编辑、编译、运行和调试的一般过程。

## 【教学指引】

教师可将本章讲课重点放在对 C 语言基本结构的解说和示范 Turbo C 环境的操作使用上,关于“如何编写可读性良好的 C 程序”这一节,可引导学生自学及提醒学生在以后的编程实践中随时注意并养成良好习惯。

- 学时安排:讲课 2~3 学时,实验 2 学时。
- 重点:C 语言基本结构和开发一个 C 程序的完整的上机过程,包括编辑、编译、运行和调试等步骤。
- 难点:熟悉 Turbo C 编译系统工作环境的设置和快捷键的使用,能够对有错误的程序进行改错、调试与运行。
- 实验要求:完成实验一。
- 作业要求:能模仿书中的例程写出符合 C 语言规范的简单 C 程序并上机运行,完成习题 2、4、5。

## 1.1 C 语言的发展简史

C 语言发源于著名的美国贝尔实验室(Bell Laboratory),是该实验室的研究人员 Dennis Ritchie 和 Ken Thompson 两人于 20 世纪 70 年代初在设计 UNIX 操作系统时开发出来的。

以前的操作系统等系统软件主要是用汇编语言编写的。原来的 UNIX 操作系统就是 1969 年由美国贝尔实验室的 Ken Thompson 和 Dennis Ritchie 利用汇编语言开发成功的。由于汇编语言依赖于计算机硬件,程序的可读性和可移植性都比较差。为了解决这个问题,贝尔实验室的 Ken Thompson 于 1970 年以 BCPL 语言(英国剑桥大学的 Martin Richards 设计的比较接近硬件的一种高级语言)为基

础,设计出一种简单而且接近硬件的高级语言——B语言,并用B语言写了UNIX操作系统。

由于B语言是无类型的语言,过于简单且功能有限。随着对软件任务要求的日益提高,迫切需要设计一种更好、功能更强的语言。1972年到1973年间,Dennis Ritchie在B语言基础上设计了C语言,既保留了B语言精炼和接近硬件的优点,又增加了函数、整型数、浮点数、字符、几十种操作符和其他成分。后来,C语言又经过多次改进,1977年出现了不依赖于具体机器的C语言编译文本《可移植C语言编译程序》,大大提高了C语言的可移植性,同时也推动了UNIX操作系统迅速在各种机器上的实现。到1978年,C语言已经发展成最流行的几种计算机编程语言之一。

随着C语言使用的日益频繁,出现了各种C编译系统。为了使C语言更加规范化,1983年美国国家标准化协会(ANSI)开始对C的发展和扩充制定了新的标准,称作ANSI C,1987年又公布了新标准——87 ANSI C,该标准于1990年被国际标准化组织(ISO)接受为国际标准。Microsoft C、Turbo C、Quick C、Borland C等流行的C编译系统都是以ANSI C为基础,但是不同版本也略有不同。本书内容的阐述以ANSI C为基础,以Turbo C为编译环境,所有范例程序均在Turbo C环境调试通过。

C语言的发展一直没有结束。20世纪80年代后,许多著名的软件公司都对C语言的功能做了大量的扩充,先后出现了C++、Visual C++、C#,使C语言继续向面向对象、可视化和网络化的程序设计语言发展。

## 1.2 C语言的特点

C语言是一种应用广泛的编程语言。用C语言不仅可以编写系统软件如UNIX、Microsoft Windows、DBASE以及当今应用越来越广泛的嵌入式系统软件等,还可以编写出各种应用软件如游戏软件、商业软件、工具软件、文字处理软件、图表软件、数学运算软件等。

C语言之所以能够发展成为当今最受欢迎和最优秀的语言之一,是因为C语言具有如下几个方面的特点:

①语言简洁、灵活,程序执行效率高。C语言只有32个关键字,对数据的描述简单,各种语句的形式简单,程序书写自由,源程序的编写方便。用C语言编写的程序比用其他高级语言编写的程序执行效率高,比用汇编语言编写的代码低10%~20%。

②运算符丰富。C语言共有34种运算符,运算类型极为丰富,表达式类型多样化,能够实现在其他高级语言中难以实现的运算功能。

③数据类型丰富。C 语言的数据类型有整型、实型、字符型、数组、指针、结构体、共用体、枚举等类型,可实现各种复杂的数据结构的运算。

④灵活的结构化控制语句。C 语言具有条件选择、多分支选择、循环等多种控制语句,是理想的结构化程序设计语言,而且 C 语言以函数为程序基本单位,容易实现模块化程序设计。

⑤语法不严,程序设计自由。C 语言对语法检查相对较宽松。如对变量类型使用比较灵活,整型数据与字符型数据可以通用,逻辑表达式中参与运算的量值为非 0 即代表“真”,而不像其他语言只有值为 1 才代表“真”,等等。语法限制不严,编程人员可以更加灵活地进行程序设计。

⑥具有直接对硬件进行控制的功能。C 语言能够通过位运算直接对硬件进行操作,可以实现汇编语言的大部分功能。这是 C 语言有别于其他高级语言的最突出的一大特点,也是 C 语言可用来编写系统软件的原因。所以 C 语言具有“高级语言”和“低级语言”的双重特性,被称作“中级语言”。

⑦可移植性好。C 语言和汇编语言相比,不仅具有很强的硬件控制能力,同时具有更强的表达和运算能力,并且 C 语言本身能够独立于硬件,从而具有很好的可移植性,所以很多大型的系统软件选用 C 语言编写。

⑧C 语言的另一面。正是由于 C 语言语法限制不严、编程灵活,所以要求 C 语言程序员必须花更多的时间去学习、熟悉 C 语言语法,尤其是指针的应用常常使初学者难以理解,从而产生 C 语言难学的印象。其实,这只是 C 语言给你难以接近的第一印象,一旦跟“她”熟悉后,就会逐渐体会到“她”的魅力无穷,并深深地喜欢上“她”。

## 1.3 为什么选择 C 语言

任何一种计算机语言都有其发展的目的和特点。Basic 语言的主要特点是初学者容易用来编写程序,语法近似英文,浅显易懂。FORTRAN 语言是为了解决科学计算问题而产生的,而 COBOL 语言则是为了解决商业和管理上的数据处理问题而发展起来的。还有 PASCAL 语言,它是第一个结构化语言。但是这些语言都因其发展背景以及语言本身的限制而无法兼顾实用和功能。

为什么选择 C 语言?

首先,C 语言继承和发扬了许多高级语言的成功经验,同时弥补了它们的缺陷,因而发展成为任何专业程序员都必须熟悉精通的一门程序设计语言。

其次,C 语言也逐渐代替了过时的 Pascal 语言而成为高等院校计算机专业以及其他一些工科类专业学生必修的一门计算机语言课程。现在计算机专业学生必修的“数据结构”以及“操作系统”课程基本上都以 C 语言作为先修课程,因为这两

门课程中的算法描述和程序设计大多使用 C 语言,所以学好 C 语言非常必要,而且掌握了 C 语言之后,就可以毫不困难地挑选并掌握其他程序设计语言。当今被广泛应用于视窗与网络程序设计的 Visual C++、C# 以及 Java 等语言均以 C 语言为基础。因此,学好 C 语言是很有必要的。

## 1.4 认识 C 程序

### 1.4.1 一个简单的 C 程序

下面让我们从一个简单的 C 程序开始进入 C 的世界。你可以在任何喜爱的文本编辑器中编写 C 程序,编写完后再移植到 C 的编译器中编译执行。C 的编译系统一般都提供相应的程序编辑器,如 Turbo C、Microsoft C 和 Microsoft Visual C++ 等,所以人们一般都在 C 的编辑器中编写程序。下面就让我们在 Turbo C 的环境里编写第一个 C 语言程序(关于 Turbo C 的详细操作说明请看 1.5.2 节)

①打开 Turbo C 环境,在编辑窗口输入下面程序。

例 1-1 在屏幕显示字符串“Welcome to the C world!”。

```
/* exam1_1 */
#include <stdio.h>
void main()
{
    printf("Welcome to the C world! \n");
}
```

图 1-1 是在 Turbo C 环境输入程序之后的情形。

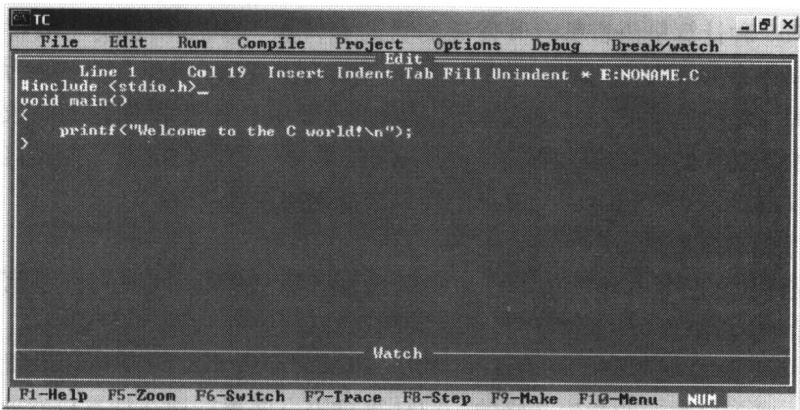


图 1-1 TC 的编辑界面



②编写完程序后,按功能键 F9 对程序进行编译和连接,生成可执行文件。图 1-2 是编译、连接后的情形。

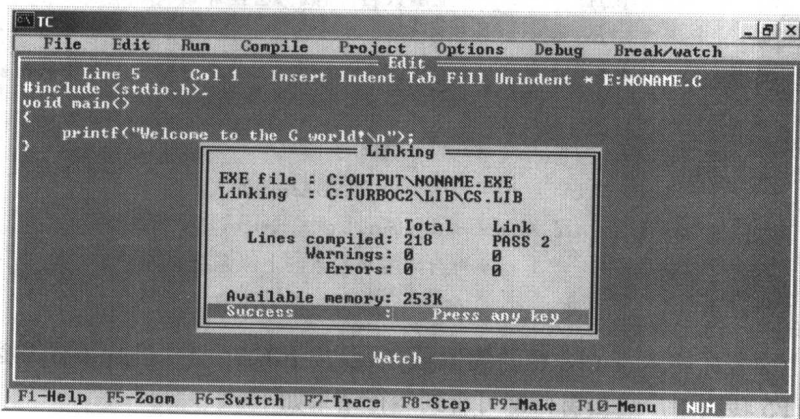


图 1-2 TC 的编译、连接提示界面

③按下功能键 Ctrl+F9 执行程序,再按 Alt+F5 转到用户输出窗口即可看到该程序的运行结果。再按任意键可从用户输出窗口切换到 Turbo C 环境。图 1-3 是程序运行后的用户输出窗口。

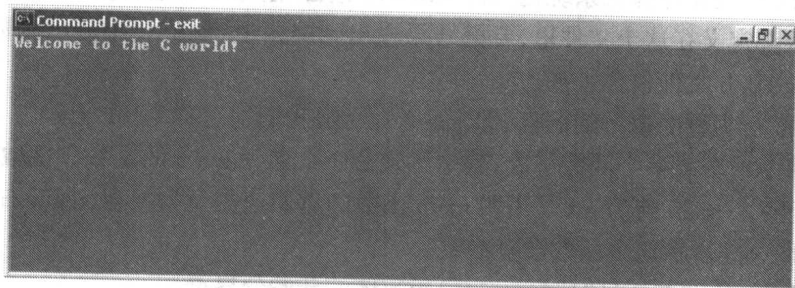


图 1-3 TC 的运行结果输出窗口

程序运行结果输出“Welcome to the C world!”

## 1.4.2 C 程序的基本结构

下面再来看另一个简单的 C 程序,并从这个例子出发说明 C 程序的基本结构(程序行最左边的行号是为了方便介绍程序而添加的,实际输入程序时是不需要的)。

例 1-2 计算两个数的和并输出结果。