



高职高专“十一五”计算机类专业规划教材

C语言 程序设计

李素萍 主编

C YUYAN
CHENGXU SHEJI

赠电子课件



机械工业出版社
CHINA MACHINE PRESS

TP312 C
269
1=

高职高专“十一五”计算机类专业规划教材

C 语言程序设计

主 编 李素萍
副主编 段海英
参 编 韩俊芳 段新娥
李军红 白利军



机械工业出版社

本书是高职高专“十一五”计算机类专业规划教材。主要讲解 C 语言的基本语法知识和程序设计方法。全书共分 12 章,内容包括: C 语言概述、数据类型、运算符与表达式、顺序结构程序设计、选择结构程序设计、循环控制、数组、函数、编译预处理、指针、结构体与共用体、位运算、文件。在讲解内容的过程中,穿插讲解了常用算法、程序设计技巧等。

本书按照高职高专 C 语言程序设计课程教学大纲的基本要求编写,以“必需、够用”为原则,注重把握深度,突出实践应用,简明扼要,通俗易懂。每章配有实训和精选的习题。

本书可作为高等职业学校、高等专科学校、成人高等学校及本科院校工科各专业的教材,也可作为计算机培训和等级考试辅导用书,还可供计算机工程技术人员自学或参考。

图书在版编目(CIP)数据

C 语言程序设计/李素萍主编. —北京:机械工业出版社, 2007.9

高职高专“十一五”计算机类专业规划教材

ISBN 978 - 7 - 111 - 22313 - 9

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 (2007) 第 139757 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 王玉鑫

责任编辑: 张 芳 版式设计: 霍永明 责任校对: 姚培新

封面设计: 马精明 责任印制: 邓 博

北京京丰印刷厂印刷

· 2008 年 1 月第 1 版 · 第 1 次印刷

184mm × 260mm · 16 印张 · 393 千字

0 001—4 000 册

标准书号: ISBN 978 - 7 - 111 - 22313 - 9

定价: 24.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

销售服务热线电话: (010) 68326294

购书热线电话: (010) 88379639 88379641 88379643

编辑热线电话: (010) 88379541

封面无防伪标均为盗版

前 言

随着计算机的普及和发展,对计算机应用人员的技术要求越来越高,需要计算机应用人员掌握一定的程序设计技术解决实际问题。而程序设计的前提是掌握程序设计语言。C语言具有功能丰富、表达能力强、使用灵活、应用广泛、目标程序效率高、可移植性好等特点,既可用于编写应用软件,又可用于编写系统软件;而且几乎任何一种机型、任何一种操作系统都支持C语言开发;同时C语言又具备可拓展面向对象语言的成分,发展成为较有影响力的面向对象程序设计语言C++。据此,高职高专将C语言定为各专业的必修课,作为程序设计基础。

C语言作为高职高专工科类计算机专业基础课程,主要讲解C语言的语法规则及其程序设计方法,重点培养学生用C语言编写程序的基本能力。但C语言涉及到的概念、规则繁多,使用又特别灵活,给初学者带来很大困难。为此,本教材在编写过程中重点体现如下特色,以确保读者顺利地掌握和使用C语言。

1) 课程教育目标的定位,体现了新时期对应用型人才培养的新要求。本教材是按照高职高专工科类C语言程序设计课程教学大纲的基本要求,结合计算机专业的专业特性,组织长期从事C语言程序设计课程教学的教师和经验丰富的软件工程建设技术人员反复研讨、精心设计的,经典实例和实用程序并重,能使读者深刻、透彻地理解相关理论,轻松、快捷地解决实际问题。

2) 传输给学生的知识要素,与高职高专学生就业需求相适应。在编写本教材时,考虑到高职高专学生文化基础相对低、没有计算机语言基础、该课程学时数少等因素,以“必需、够用”为原则,理论知识的阐述力争简明扼要、通俗易懂,知识的应用方面尽可能照顾到各专业,选用的例题典型、精练、贴近工程实际。

3) 概念准确、结构严谨,便于学生科学而系统地掌握知识。本教材在强调语言讲解通俗性的同时,还保证概念、术语使用的准确性,以防理解其他教程出现偏差;同时考虑到C语言知识点多,学生思维容易发生混乱,每章内容在整体结构安排上始终采用总一分一总的形式,以体现逻辑性和系统性。

4) 知识结构组织的模块化,便于由浅入深、循序渐进地进行学习。本教材将“必需”的教学内容由浅入深、循序渐进地分为“C语言概述”、“数据类型、运算符与表达式”、“顺序结构程序设计”、“选择结构程序设计”、“循环结构”、“数组”、“函数”、“编译预处理”、“指针”、“结构体与共用体”、“位运算”和“文件”等12章,以突出重点、分散难点,既强化了知识间的纵向联系,也便于各内容之间的横向比较,有利于学生对基本内容的学习、理解、记忆和掌握,也利于教师的教学。

5) 通用性,给学生提供了极大的方便。本教材介绍的理论知识及其程序例题,基本适用于任何计算机系统和C语言版本。不过,不同C语言版本有细微差别,学生使用时请自行修改。

6) 配套资料满足师生使用需求。为帮助学生更好地学习和掌握教程内容,每章后都配



有实训部分。习题模拟全国计算机等级考试样题，既可帮助学生牢固掌握 C 语言程序设计的内容，又为学生参加 C 语言等级考试提供了极大的方便。同时，提供与教材配套、制作精良、使用方便的多媒体教学课件。

本教材由李素萍任主编，负责拟定全书框架并对参编所写章节提出具体的修改意见；段海英任副主编；韩俊芳、段新娥、李军红、白利军参编。各章具体分工如下：李素萍编写第 8、9、11、12 章及附录部分，段海英编写第 4、5 章及两套模拟题，韩俊芳编写第 1、6 章，段新娥编写第 2、7 章，李军红编写第 10 章，白利军编写第 3 章。

虽然作者有十余年的 C 语言教学经验，但由于水平有限，加之时间仓促，错漏之处在所难免，诚恳接受读者的宝贵意见，以便于进一步完善。作者的电子信箱：lsplb@126.com。

编 者

目 录

前言

第1章 C语言概述 1

1.1 C语言的发展过程和特点 1

1.1.1 C语言的产生与发展 1

1.1.2 C语言的特点 1

1.2 C语言的程序组成 2

1.2.1 几个简单的C语言程序实例 2

1.2.2 C语言程序的结构特点 4

1.2.3 C语言程序的书写风格 4

1.3 C语言程序上机步骤 4

本章小结 6

习题 6

实训 7

第2章 基本数据类型、运算符与表达式 11

2.1 C语言数据类型概述 11

2.2 标识符、常量和变量 12

2.2.1 标识符 12

2.2.2 常量和符号常量 12

2.2.3 变量 13

2.3 整型数据 13

2.3.1 整型数据的分类 13

2.3.2 整型常量的表示 13

2.3.3 整型变量的定义和初始化 14

2.3.4 整型数据在内存中的存储形式 14

2.4 实型数据 15

2.4.1 实型数据的分类 15

2.4.2 实型常量的表示 15

2.4.3 实型变量的定义和初始化 15

2.4.4 实型数据在内存中的存储形式 16

2.5 字符型数据 16

2.5.1 字符型常量的表示 16

2.5.2 字符串常量 17

2.5.3 字符型变量定义和初始化 17

2.5.4 字符型数据在内存中的存储形式 18

2.5.5 字符型数据与整型数据的关系 18

2.6 不同类型数据间的转换 19

2.6.1 隐式类型转换 19

2.6.2 显式类型转换（强制转换） 20

2.7 运算符和表达式 20

2.7.1 C语言运算符简介 21

2.7.2 算术运算符和算术表达式 21

2.7.3 赋值运算符和赋值表达式 22

2.7.4 逗号运算符和逗号表达式 22

本章小结 23

习题 23

实训 25

第3章 顺序结构程序设计 27

3.1 C语言语句概述 27

3.2 C语言赋值语句 28

3.3 数据输入输出的概念及在C语言中的实现 29

3.4 字符数据的输入输出 29

3.4.1 putchar函数(字符输出函数) 29

3.4.2 getchar函数(字符输入函数) 30

3.5 格式输入输出 31

3.5.1 printf函数(格式输出函数) 31

3.5.2 scanf函数(格式输入函数) 36

3.6 顺序结构程序设计举例 39

本章小结 40

习题 40

实训 42

第4章 选择结构程序设计 44

4.1 关系运算和逻辑运算 44



4.1.1 关系运算符和关系表达式	44	6.4.5 字符串的引用和输入输出	75
4.1.2 逻辑运算符和逻辑表达式	45	6.4.6 字符串常用处理函数	76
4.2 if 语句	46	6.4.7 字符数组应用举例	77
4.2.1 if 语句的一般形式	46	本章小结	79
4.2.2 if 语句的嵌套	48	习题	79
4.3 条件表达式构成的选择结构	49	实训	79
4.4 switch 语句	50		
4.5 程序举例	51	第7章 函数	82
本章小结	53	7.1 概述	82
习题	53	7.1.1 模块化程序设计	82
实训	54	7.1.2 函数的分类	84
第5章 循环控制	55	7.2 函数的定义	85
5.1 概述	55	7.3 函数参数和函数的值	86
5.2 goto 语句以及用 goto 语句构成 循环	55	7.3.1 形式参数和实际参数	88
5.3 while 语句	56	7.3.2 函数的返回值	89
5.4 do-while 语句	57	7.4 函数的调用	89
5.5 for 语句	58	7.4.1 函数调用的形式	90
5.6 循环的嵌套	60	7.4.2 函数调用的方式	90
5.7 控制语句	61	7.4.3 函数的说明	90
5.8 程序举例	61	7.5 函数的嵌套调用和递归调用	91
本章小结	62	7.5.1 函数的嵌套调用	91
习题	63	7.5.2 函数的递归调用	92
实训	64	7.6 数组作为函数参数	95
第6章 数组	66	7.6.1 数组元素作函数实参	95
6.1 引言	66	7.6.2 数组名作函数参数	96
6.2 一维数组	66	7.7 变量的作用域和类型	99
6.2.1 一维数组的定义	67	7.7.1 变量的作用域	99
6.2.2 一维数组的引用	68	7.7.2 局部变量	100
6.2.3 一维数组的初始化	69	7.7.3 全局变量	100
6.2.4 一维数组程序举例	69	7.8 变量的存储类别	102
6.3 二维数组	70	7.8.1 动态存储方式与静态存储 方式	102
6.3.1 二维数组的定义	70	7.8.2 auto 变量	102
6.3.2 二维数组元素的引用	71	7.8.3 static 变量	103
6.3.3 二维数组的初始化	72	7.8.4 register 变量	104
6.3.4 二维数组的初始化程序举例	72	7.8.5 用 extern 声明外部变量	104
6.4 字符数组与字符串	73	7.8.6 关于变量的声明与定义	104
6.4.1 字符数组的定义	73	7.8.7 变量属性小结	105
6.4.2 字符数组的初始化	74	7.9 内部函数和外部函数	105
6.4.3 字符数组元素的引用	74	7.9.1 内部函数	106
6.4.4 字符串	75	7.9.2 外部函数	106
		7.10 如何运行一个多文件的程序	107
		本章小结	107



习题	108	9.8 void 指针类型	157
实训	109	本章小结	157
第 8 章 编译预处理	113	习题	158
8.1 引言	113	实训	163
8.2 宏定义	114	第 10 章 结构体与共用体	166
8.2.1 不带参数的宏定义	114	10.1 结构体类型的声明和结构体	
8.2.2 带参数的宏定义	115	变量的定义	166
8.2.3 带参数的宏与函数的区别	116	10.1.1 结构体的概念	166
8.3 文件包含	116	10.1.2 结构体类型的说明	167
8.4 条件编译	119	10.1.3 结构体变量的定义	168
本章小结	120	10.2 结构体变量的引用	171
习题	121	10.3 结构体变量的初始化	172
实训	123	10.4 结构体数组	174
第 9 章 指针	126	10.4.1 结构体数组的定义	174
9.1 引言	126	10.4.2 结构体数组的初始化和	
9.2 地址和指针的概念	126	引用	176
9.3 变量的指针和指向变量的		10.4.3 结构体数组的引用举例	176
指针变量	127	10.5 结构体指针	177
9.3.1 指针变量的定义和初始化	127	10.5.1 指向结构体变量的指针	177
9.3.2 指针变量的引用	128	10.5.2 指向结构体数组的指针	179
9.3.3 指针变量作函数参数	130	10.6 结构体和函数	180
9.4 指针与数组	132	10.6.1 值传递——用结构体变量的成员	
9.4.1 指向数组元素的指针	133	作函数参数	181
9.4.2 通过指针引用数组元素	133	10.6.2 多值传递——用结构体变量	
9.4.3 指针的运算	134	作函数参数	181
9.4.4 数组名作函数参数	136	10.6.3 地址传递——用结构体指针（或	
9.4.5 指针与二维数组	141	结构体数组）作函数参数	182
9.5 指针与字符串	145	10.7 用指针处理链表	185
9.5.1 字符串的表示形式	146	10.7.1 链表概述	185
9.5.2 字符串指针作函数参数	147	10.7.2 建立简单链表	186
9.5.3 对字符指针变量与字符数组的		10.8 共用体	188
讨论	148	10.8.1 共用体的概念	188
9.6 指针与函数	149	10.8.2 共用体类型的声明和共用体	
9.6.1 指针函数	149	变量的定义	188
9.6.2 用函数指针变量调用函数	151	10.8.3 共用体变量引用	189
9.6.3 用指向函数的指针作函数		10.9 用 typedef 定义类型	192
参数	152	10.9.1 用 typedef 定义类型的	
9.7 指针数组与多级指针	153	一般形式	192
9.7.1 指针数组	154	10.9.2 用 typedef 定义类型时应	
9.7.2 多级指针	155	注意的事项	193
9.7.3 指针数组作 main 函数的形参	157	本章小结	193
		习题	194



实训	196	(fscanf 函数和 fprintf 函数)	212
第 11 章 位运算	198	12.4.3 二进制文件的输入输出 函数	213
11.1 概述	198	12.4.4 其他读写函数	214
11.2 位运算符和位运算	198	12.5 文件的定位、测试与随机读写	215
11.2.1 “按位与”运算符	198	12.5.1 rewind 函数	215
11.2.2 “按位或”运算符	199	12.5.2 fseek 函数和随机读写	215
11.2.3 “异或”运算符	200	12.5.3 ftell 函数	217
11.2.4 “取反”运算符	200	12.6 检错与处理	217
11.2.5 左移运算符	201	12.6.1 ferror 函数	217
11.2.6 右移运算符	201	12.6.2 clearerr 函数	218
11.2.7 位运算赋值运算符	202	12.6.3 exit 函数	218
11.2.8 不同长度的数据进行位运算	202	12.7 程序举例	218
11.3 位运算举例	202	本章小结	221
本章小结	203	习题	221
习题	203	实训	223
实训	204	附录	227
第 12 章 文件	207	附录 A 常用字符与 ASC II 代码 对照表	227
12.1 C 语言文件概述	207	附录 B C 语言中的关键字	228
12.1.1 文件的概念	207	附录 C 运算符和结合性	228
12.1.2 缓冲文件系统和非缓冲 文件系统	207	附录 D C 语言常用语法提要	229
12.2 文件类型指针	208	附录 E 库函数	233
12.3 文件的打开和关闭	209	试卷	239
12.3.1 文件的打开 (fopen 函数)	209	C 语言程序设计试卷一	239
12.3.2 文件的关闭 (fclose 函数)	210	C 语言程序设计试卷二	241
12.4 文件的读写	210	参考文献	246
12.4.1 文件的字符输入输出函数 (fgetc 函数和 fputc 函数)	211		
12.4.2 文件的格式化输入输出函数			

第 1 章 C 语言概述

学习目标:

- 1) 了解 C 语言的发展及特点。
- 2) 掌握 C 语言程序的结构。
- 3) 掌握 C 语言程序的上机步骤。

1.1 C 语言的发展过程和特点

计算机语言经历了较长的发展阶段, 各类语言的出现为计算机技术的发展提供了强大的动力。其中, C 语言是一种得到广泛重视并普遍应用的计算机程序设计语言, 也是国际上公认的最重要的几种通用程序设计语言之一。它适合作为系统描述语言, 既可用于开发系统软件, 也可用来开发应用软件。

1.1.1 C 语言的产生与发展

在 C 语言产生之前, 系统软件主要是用汇编语言编写的。由于汇编语言程序依赖于计算机硬件, 其可读性和可移植性都很差, 而一般高级语言又难以实现对计算机硬件的直接操作, 人们设想能否找到一种既具有一般高级语言特性, 又具有低级语言特性的语言, 于是, C 语言在这种背景下应运而生。

C 语言是贝尔实验室于 20 世纪 70 年代研制出来的, 并随着 UNIX 操作系统的广泛使用得到迅速推广。后来, C 语言又被多次改进, 并出现了许多版本。80 年代初, 美国国家标准协会 (ANSI) 根据 C 语言问世以来的各种版本对 C 语言作了扩充, 制定了 ANSI C 标准。

目前微机上广泛流行使用的有 Turbo C、Quick C、Microsoft C 等, 本书用 TC V2.0 (DOS 环境)。

1.1.2 C 语言的特点

C 语言之所以能存在和发展, 并具有强大的生命力, 这主要归功于其所具有的鲜明的特点:

- 1) C 语言有高级语言的优点, 也有低级语言的特点, 能用来编写应用软件, 也可用来编写系统软件。
- 2) 语言简洁, 使用方便、灵活。C 语言一共只有 32 个关键字和 9 种控制语句, 程序书写形式自由, 主要用小写字母表示。在一般语言中的许多结构单元在 C 语言中都通过库函数调用来完成, 库函数可根据需要方便地扩充, 压缩了一切不必要的程序组成部分。
- 3) 具有各种控制流结构, 如 while、do-while、switch 语句等, 足以满足结构化程序设计的需要。
- 4) 数据类型丰富, 有基本型和多种构造型数据, 同时各基本数据类型之间的转换也较为方便。
- 5) 语言产生的代码质量高, 即完成相同的任务 C 语言产生的代码要比其他高级语言产



生的代码短，系统开销少。

6) 可移植性好，即程序可以从一个环境不加或稍加改动就可搬到另一个环境上运行。

7) 具有高效灵活的指针运算。利用指针可以直接访问内存单元，同时指针可以作为数据来运算，指针可以提高程序的效率。

8) 运算符丰富。C 语言把括号、赋值、指针、强制类型转换等都作为运算符处理，包含范围较广，灵活应用各种运算符，可以得到形式多样的表达式。

1.2 C 语言的程序组成

1.2.1 几个简单的 C 语言程序实例

下面先介绍几个简单的 C 语言程序，然后分析 C 语言程序的特性。

【例 1-1】

/* 仅由 main() 函数构成的 C 语言程序。 */

```
main()  
{ printf("Welcome to c!\n");  
}
```

本程序的作用是在屏幕上输出以下信息：

Welcome to c!

在程序中，main() 表示主函数，它后边的小括号表示函数。一个 C 语言源程序可有多函数，但只能有一个主函数，也必须有一个主函数。在 main() 下面的一对大括号内是函数体，所有函数体是以大括号开始到大括号结束。在函数体内有一条语句 printf("Welcome to c!"); 这条语句的作用是在屏幕上打印出一条信息：Welcome to c! 双引号内的内容在屏幕上按原样输出。“\n”是光标换行符。分号表示一个语句的结束。这个程序中函数体内只有一条语句，也可以有多条语句，每条语句以分号结束。printf 是标准输出函数，在后面第 3 章将详细介绍，这里只分析 C 语言程序的构成。

【例 1-2】

```
main()  
{  
    int sum;           /* 定义一个整型变量 */  
    sum = 100 + 25;     /* 为 sum 赋值 */  
    printf("sum is %d\n", sum);  
}
```

本程序的作用是求 100 和 25 之和，在 main() 函数体内有三条语句，第一条语句定义了一个整型局部变量，关于局部变量在以后各讲中详细介绍。在这条语句后边的 “/* ... */” 为注释行，在 C 语言程序中注释行是由两对斜杠加星号括起来的内容。注释行只是对程序进行说明，编译器将不理睬它，在程序执行中也不起作用。第二条语句是赋值语句，等号右边是算术表达式，经过计算后将值赋给 sum。第三条语句是标准输出语句，它在屏幕上打印出 sum 的值。双引号内的 sum is 是原样在屏幕上打印，而 %d 是输出格式码，它表示输出整型



数据。printf 函数的格式码很多在后面有关章节中详细介绍。在逗号后面的 sum 是变量，是 printf 要输出的整型变量。本程序编译后执行输出为：

```
sum is 125
```

【例 1-3】

```
#include <stdio.h>
```

```
void main()
```

```
{ int add(int x,int y);
```

```
    int mul(int x,int y); /* 对被调用函数的声明 */
```

```
    int a,b,sum,count; /* 声明部分,定义变量 */
```

```
    a = 20;b = 35; /* 为 a,b 赋初值 */
```

```
    sum = add(a,b); /* 调用函数 add,将得到的值赋给 sum */
```

```
    count = mul(a,b); /* 调用函数 mul,将得到的值赋给 count */
```

```
    printf("sum = %d,count = %d\n",sum,count);
```

```
        /* 输出 sum,count 的值 */
```

```
}
```

```
int add(int x,int y) /* 定义 add 函数,函数值为整型,形式参数 x 和 y 为整型 */
```

```
{
```

```
    int z; /* add 函数中的声明部分,定义本函数用到的变量 z 为整型 */
```

```
    z = x + y;
```

```
    return(z); /* 将 z 的值返回,通过 add 带回调用处 */
```

```
}
```

```
int mul(int x,int y) /* 定义 mul 函数 */
```

```
{
```

```
    int w; /* 定义 mul 函数中用到的变量 w 为整型 */
```

```
    w = x * y;
```

```
    return(w); /* 返回 w 的值 */
```

```
}
```

运行结果为：

```
sum = 55, count = 700
```

这个程序的目的是显示两个整数 a、b 的和与积，说明如下：

函数是构成 C 语言程序的基本单位。本程序包含 3 个函数：主函数 main()、函数 add() 和 mul()。函数 add() 的作用是求 a、b 之和，函数 mul() 的作用是求两个数的乘积。

在 C 语言程序中，一个函数可以使用函数调用的方法访问另一个函数。程序中的第 7 行和第 8 行为 main() 函数对 add() 函数和 mul() 函数的调用。在调用函数时将实际参数 a 和 b 的值分别传送给函数 add() 和 mul() 中的形式参数 x 和 y，然后执行函数相应的运算，并将结果通过 return 语句返回给主函数。

1.2.2 C 语言程序的结构特点

通过 1.2.1 节中的几个程序，可以看出 C 语言程序有如下结构特点：



(1) C 语言程序是由函数构成的。函数是构成 C 语言程序的基本单元。一个源程序有且只能有一个 `main()` 函数，也可以包含一个 `main()` 函数和若干个其他函数。通过函数调用的形式来实现函数之间的信息传递。被调用的函数可以是系统提供的库函数（如 `printf()` 和 `scanf()` 函数），也可以是用户根据需要自己编制设计的函数（如例 1-3 中的 `add()` 函数）。函数是 C 语言程序模块化结构的重要体现。

(2) `main()` 函数是程序执行的起点。一个 C 语言程序总是从 `main()` 函数开始执行的，而不论 `main()` 函数在整个程序中的位置如何。`main()` 函数可以放在程序的最前面，也可以放在程序最后，或放在一些函数之间。

(3) 函数由函数说明和函数体两部分组成

1) 函数说明包括函数名、函数类型、函数参数名和参数类型。例如，例 1-3 中的函数 `add()`，其函数说明各部分如下：

函数类型	函数名	函数参数类型	函数参数名
↓	↓	↓	↓
int	add	(int	x, int y)

函数参数可以没有，如 `main()` 函数。

2) 函数体即函数说明下面花括号内的部分，包括说明语句和可执行语句。

说明语句部分：在这部分中定义所用到的变量，如例 1-3 函数 `add()` 中的 “`int z;`”。

可执行语句部分：一般由若干条可执行语句构成，如例 1-3 函数 `add()` 中的 “`z = x + y;`
`return (z);`”。

注意：函数体中的变量定义语句必须在所有可执行语句之前。

1.2.3 C 语言程序的书写风格

1) 所有语句都必须以分号 “`;`” 结束，函数的最后一个语句也不例外。分号是 C 语言程序语句必不可少的组成部分。

2) 程序行的书写格式自由，既允许一行内写几条语句，也允许一条语句分写在几行上。当一个语句写在多行上时，用 “`\`” 作续行符。如果某条语句很长，一般需要将其分写在几行上。

3) 允许使用注释，以提高程序的可读性。可在 C 程序的任何部分添加注释语句，C 语言注释由 “`/*`” 开始，至 “`*/`” 结束。

注意：

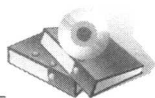
1) “`/*`” 和 “`*/`” 必须成对使用，且 “`/`” 和 “`*`” 以及 “`*`” 和 “`/`” 之间不能有空格。

2) 注释的位置，可以单占 1 行，也可以跟在语句的后面。如果 1 行写不下，可另起 1 行继续写。

3) 注释中允许使用汉字。

1.3 C 语言程序上机步骤

在了解了 C 语言的基本知识后，下面接着介绍运行 C 语言程序的相关问题。本节介绍目前使用最广泛的 C 语言程序开发软件——Turbo C 的基本知识。Turbo C 是 Borland 公司开



发的一个适用于微型计算机,同时集程序编辑、编译、链接和调试为一体的C语言编译系统。它具有良好的用户界面和丰富的库函数,同时还有运行速度快、效率高和功能强等优点,使用非常方便。

一个C语言程序要能够最终实现既定的功能,需要依次经历以下基本环节(程序的具体执行过程详细步骤请参考实训部分):

(1) 编辑 使用C语言编写程序代码,创建源文件。

(2) 编译 在C语言程序的编译过程中,可以查出程序中的语法错误。编译器将程序转换为机器代码后即可生成目标程序(.obj)。

(3) 链接 C语言程序是模块化设计程序,一个C语言程序可能由多个程序设计者分工合作编写。最后需要将库函数以及其他目标程序链接为一个整体,生成可执行文件(.exe)。

(4) 运行 运行源文件经过编译链接后生成的可执行文件(.exe)即可获得正确的结果。

下面简单认识Turbo C 2.0的环境。

1. 启动 Turbo C

进入Turbo C环境需要调用tc.exe,可以用如下两种方法进行操作:

(1) 由DOS平台进入Turbo C 使用DOS命令,假设文件存于C:\TurboC中,则为:

C:\>CD TurboC

C:\TurboC>tc

进入Turbo C环境,屏幕将显示出如图1-1所示的Turbo C工作窗口。

(2) 由Windows平台进入Turbo C 通过“资源管理器”找到文件夹Turbo C中的tc.exe文件,双击该文件即可进入Turbo C环境。进入后屏幕正中有一个Turbo C的版本信息框,标明Turbo C的版本号、生产日期和公司名称,当用户按下任一键时,此版本信息框就会消失,用户看到的将是Turbo C工作窗口。

2. 工作窗口简介

(1) 主菜单 主菜单包含8个下拉菜单,分别为File、Edit、Run、Compile、Project、Options、Debug和Break/watch,每个菜单中有命令可以使用,如图1-1所示。

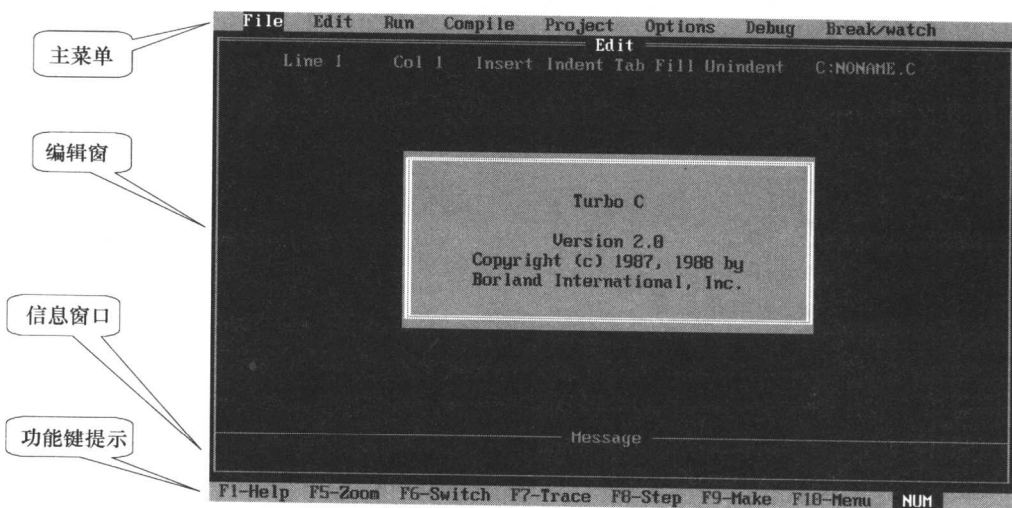
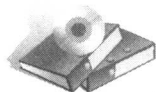


图 1-1



(2) 编辑窗口 在主菜单窗口的下面,正上方有 Edit 字样作为标志。其作用是对 Turbo C 源程序进行输入和编辑。源程序都在这个窗口中显示,因而编辑窗口占据了屏幕的大部分面积。窗口中顶端横线为双线显示的,表示该窗口是活动窗口。

在编辑窗口的上部有一行英文:

Line 1 Col 1 Insert Indent Tab Fill Unindent C: NONAME. C

其中“Line 1”和“Col 1”表示当前光标的位置在第 1 行第 1 列。当光标移动时,Line 和 Col 后面的数字也随之改变,它用来告诉用户光标当前所在的位置。

该行最右端显示的是当前正在编辑的文件名,对新文件自动命名为 NONAME. C。如果从磁盘调入一个已存在的文件则在该位置上显示的不再是 NONAME. C,而是该文件的名字。

(3) 信息窗口 在屏幕的下部,用来显示编译和连接时的有关信息。在信息窗口上方有“Message”字样作为标志。在编辑源程序时不用此窗口。

(4) 功能键提示行 它显示一些功能的作用。例如:“F10”键用于激活第一个菜单 File;“F1”键显示帮助信息;“F5”键用来扩大编辑窗口;“F6”键用来激活信息窗口;“F9”键用来进行编译和连接。

3. 退出 Turbo C 系统

- 1) 按“Alt + X”组合键。
- 2) 选择 File 菜单下的“Quit”命令。

本章小结

1) 本章首先介绍了 C 语言的发展简史。C 语言诞生于美国贝尔实验室,经过不断地修改、完善,C 语言目前已成为人们广泛使用的语言之一。

2) C 语言程序由函数组成。一个 C 语言源程序有且仅有一个 main() 函数,但是可以包含任意多个其他函数,因此,函数是 C 语言程序的基本单元。C 语言开发者编制了内容丰富的库函数,编程者可以通过函数调用的形式使用这些库函数,方便、快捷、编程效率高。

3) C 语言程序中,每一个基本语句和数据定义都是以分号“;”结束,书写格式自由,一行可写多个语句,一个语句也可以分别写在几行上。可以使用/*...*/对程序添加注释以方便读者阅读、理解程序的编写意图。

4) C 语言是编译型语言,源程序必须经过编译和链接生成可执行文件后才能运行。

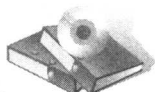
习 题

1-1 填空题

1. C 语言源程序的基本单位是_____。
2. 一个 C 语言源程序至少应包括一个_____。
3. 在一个 C 语言源程序中,注释部分两侧的分界符分别为_____和_____。
4. 在 C 语言中,输入操作是由库函数_____完成的,输出操作是由库函数_____完成的。

1-2 简答题

1. C 语言有何特点?



2. 一个C语言程序由哪几部分组成?
3. 一个C语言程序的开发应经过哪几个步骤? 简述各步骤的作用。

实 训

1. 实训概要

通过实训, 让读者掌握如何编写C语言源程序及编译、运行的整个过程。

2. 实训内容

编写程序求两数的最大值并运行。按原样输入以下程序, 并进行调试运行和看结果。

源程序:

```
int max(int x,int y)
{
    int z;
    if(x > y) z = x;
    else z = y;
    return(z)
}

main()
{
    int a,b,c;
    scanf("%d,%d",&a,&b);
    c = max(a,b);
    printf("max = %d",c);
}
```

3. 实训过程

(1) 进入 Turbo C

(2) 编辑保存源文件 如果要输入和编辑一个新的C语言程序, 应该先选主菜单中的File菜单(如果是刚刚进入TC环境, 系统会自动激活主菜单中的File菜单; 如果在编辑过其他程序后想编一个新程序, 则要按“F10”键激活主菜单中的File菜单), 按“回车”键后出现下拉菜单, 用键盘上的“↓”和“↑”键找到子菜单项New, 然后按“回车”键, 编辑窗口就被清空, 光标定位在左上角(第1行、第1列), 输入源程序(见图1-2)。

输入源程序后应对程序作认真检查, 并改正已发现的错误。这时应及时将源程序保存起来, 按“F10”键并按“回车”键, 再用“↓”和“↑”键找到子菜单项Save, 按“回车”键后, TC会自动弹出一个对话框, 要求用户指定文件名(见图1-3)。从图中可以看出, 在对话框中有一个缺省的文件名为NONAME.C, 一般不要采用缺省名字, 应另起文件名, 以便于辨别, 如C5.c, 则可以修改对话框中的文件名为: C:\TURBOC2\C5.C, 然后按“回车”键, 此时, 编辑窗口中右上角的文件名自动改为C5.C。

(3) 编译和连接 编辑好源程序并存盘后, 应当对源程序进行编译、连接和运行。一般, 可以将编译和连接合起来作为一步进行, 然后再运行。按“F10”键并选择Compile得到编译菜单, 从中选择Make EXE file项, 按“回车”键后就可一次生成一个目标文件C5.OBJ和一个可执行文件C5.EXE, 这一步也可直接按“F9”键一次性完成编译和连接。

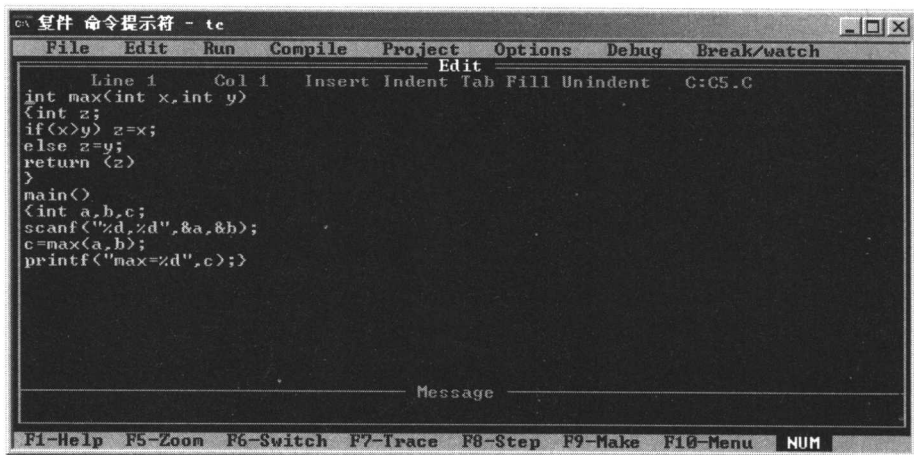


图 1-2

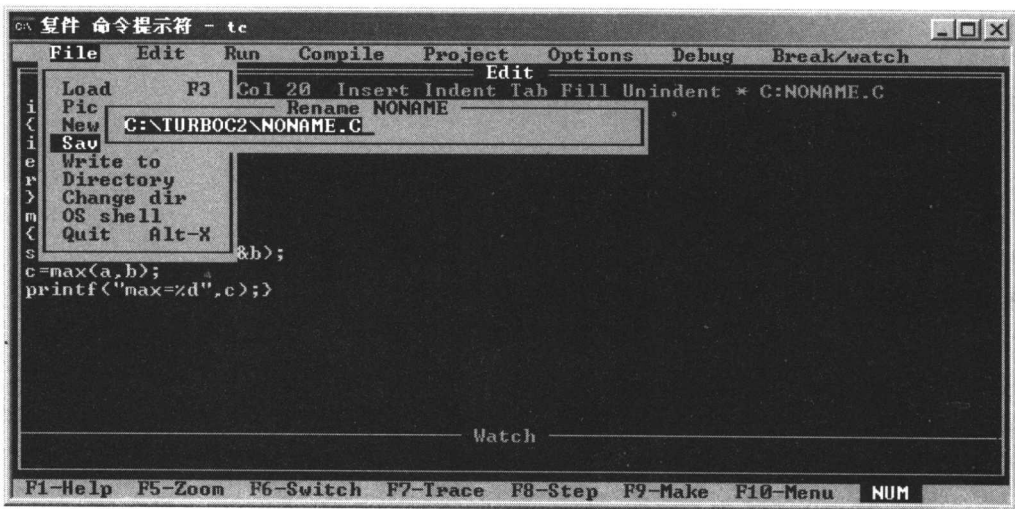


图 1-3

如果系统发现源程序有错误，会出现一个编译信息窗口，并且会提示有几个错误（见图 1-4），从图中可看出源程序有一个错误。按任何一键后，中间的编译窗口消失，在屏幕下部的消息窗口中显示了具体的出错信息，在源程序中出错的地方也以高亮度显示，便于用户查找修改错误，该程序“return (z)”句缺少了分号（在看到出错行后，不能只简单地从系统显示的出错行数去找错，应该连同上下行一起检查）。修改程序后重新进行编译和连接。

(4) 运行源程序 编译连接成功后，系统就会生成目标文件 C5.OBJ 和可执行文件 C5.EXE，这时可以运行源程序。常用操作方法为：按“F10”键，然后在主菜单中选择 Run 菜单，并按“回车”键，出现下拉菜单，从中选择 Run 项并按“回车”键（见图1-5），即开始运行。也可以按“Ctrl + F9”组合键使之运行。

选择“Run”命令并按“回车”键后，屏幕显示切换到用户屏。程序运行时，如由键盘