新版

21世纪

高职高专系列教材

C语言程序设计

◎顾可民　等编著

机械工业出版社
CHINA MACHINE PRESS

本书以简明的方式介绍了C语言程序设计的主要知识，较系统地介绍了C语言的基本概念、基本语法等，并把重点放在提高学生程序设计和解题的能力上，本书最后一章介绍了有关C++的基本概念和基本语法，为读者从C语言编程迈向C++编程奠定了一定基础。

书中的例题、习题和上机实训内容丰富，通俗易懂。另外，本书还配备了电子课件，以供读者学习使用。

本书既作为高职高专院校相关专业程序设计课程的教材，也可供相关领域的科技人员参考自学。

图书在版编目（CIP）数据

C语言程序设计/顾可民等编著 .—北京：机械工业出版社，2004.8
（21世纪高职高专系列教材）
ISBN 7-111-15020-1

Ⅰ.C... Ⅱ.顾 ... 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(2004)第077073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：李利健
责任印制：石　冉

三河市宏达印刷有限公司印刷·新华书店北京发行所发行

2004年8月第1版·第1次印刷
787mm×1092mm 1/16·15.5印张·381千字
0 001—5 000册
定价：22.00元

21世纪高职高专
计算机专业系列教材编委会成员名单

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”进行了修订工作。

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断深入，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广大教师努力将最新的教学实践经验融于教材的编写，并以可贵的探索精神推进本系列教材的更新。由于高职高专教育处在不断的发展中，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

C语言不仅是软件开发者们使用最多的工具之一，也被众多读者作为程序设计的一种基本语言来学习。它兼有高级语言和低级语言的长处，其代码简捷高效，功能强大。

目前，尽管已有为数不少的有关C语言程序设计方面的书籍问世，但所介绍内容的侧重点有所不同，主要有普及性教材（语法）、程序设计技巧与实例两类。本书作者希望能够将二者有机地结合起来，以简明的方式介绍C语言程序设计的主要内容并突出其特点。此外，还要为读者在学习和理解此语言的基本知识方面提供一些更新的素材，这是编写本书的主要出发点。为此，在本书的编写过程中，我们尽量使其体现出如下的一些特点：

（1）叙述简明扼要，重点突出。考虑到读者的层次不同，本书写作方式非常通俗，以使读者不必具有太多的专业知识就可以读懂，并尽量做到叙述简明扼要，重点突出。

（2）精心安排体系结构。作者根据多年的教学经验，认真组织了本书的知识体系，使前后内容自然衔接，循序渐进，多而不乱，且使每一部分的内容相对独立完整。

（3）充分体现C语言的特征。本书的内容充分体现了C语言的高效、灵活等特点，这是C语言迅速发展并普及的根本原因所在。

（4）新的标准。在C语言的发展中，产生了多种不同的版本，它们之间的差别有些是较大的，甚至在一些问题的处理上采取了完全不同的方式和约定。如果将这些问题混在一起，将会给初学者带来较大的困难。本书的讲解采用了微机上最流行的Turbo C软件，所有的解释皆以此为依据。问题的处理方式都采用最新的标准，这种做法将会使读者养成良好的习惯。

（5）新的概念。关于C语言的指针问题，依据基本类型的概念进行了重新解释，从而使C语言中复杂的指针概念变得简单易懂。

本书共分13章。第1章介绍程序设计的基本概念、常识，以及C语言的初步知识；第2～8章分别介绍数据、运算、输入输出、流程控制、数组和函数，这些内容是组成程序的基本成分；第9章介绍指针、指针与数组以及字符串操作，这些内容集中体现了C语言的“特殊能力”和程序设计的灵活性；第10～12章分别介绍了结构体、共用体、位运算和文件；第13章介绍C++ 面向对象程序设计基础。

为了能够实现我们的想法，依据长期的教学实践体会，除了正常的例题和每章的编程实训习题之外，还安排了综合测试和数量较多的不同题型的例题和习题，以帮助学习者更好地理解语法现象，为进行程序设计奠定扎实的基础。此外，这些题目中很多可以作为面向各种测试的自修题目，相信对读者参加诸如等级考试之类的测试是很有用的。

在本书中还增加了一些辅助内容，包括上机环境和库函数浏览。书中还介绍了C语言的上机环境和简单的程序调试方法，读者可以按照书中的指导上机实践。

本书由顾可民（编写第1、2、6、7、8、13章）、高源（编写第9～12章及附录）、王晓丹（编写第

目　录

第1章　C语言概述

本章要点

- C语言的发展简史和特点
- C语言程序的结构与书写规则
- C语言的语句和关键字
- C语言程序的编辑与运行

1.1　C语言的发展简史和特点

1.1.1　C语言的发展简史

C语言于20世纪70年代初问世，1978年由美国电话电报公司(AT&T)贝尔实验室正式发表，同时由B.W.Kernighan和D.M.Ritchit合著了《The C Programming Language》一书，通常简称为《K&R》，也有人称之为《K&R》标准。但是，在《K&R》中并没有定义一个完整的C语言标准，后来由美国国家标准化协会(ANSI)在此基础上制定了一个C语言标准，于1983年发表，通常称之为ANSI C。

早期的C语言主要用于UNIX系统。由于C语言的强大功能和各方面的优势逐渐为人们所知，到了20世纪80年代，C语言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广泛应用，成为当代最优秀的程序设计语言之一。

目前最流行的C语言有以下几种：

- Microsoft C或称MS C；
- Borland Turbo C或称Turbo C；
- AT&T C。

这些C语言版本不仅实现了ANSI C标准，并在此基础上各自作了扩充，使之更加方便、完善。

在C语言的基础上，1983年，贝尔实验室的Bjarne Strou-strup推出了C++语言。C++进一步扩充和完善了C语言，成为一种面向对象的程序设计语言。C++目前流行的最新版本是Borland C++ 4.5、Symantec C++ 6.1和Microsoft Visual C++ 2.0。C++提出了一些更为深入的概念，它所支持的这些面向对象的概念容易将问题空间直接映射到程序空间，为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法，因而也增加了整个语言的复杂性，掌握起来有一定难度。但是，C是C++的基础，它们在很多方面是兼容的，掌握了C语言，再进一步学习C++就能以一种熟悉的语法来学习面向对象的语言，从而达到事半功倍的效果。

1.1.2　C语言的特点

C语言是一种“中级语言”，它兼有低级语言和高级语言的功能和特点。一方面，它继承了低级语言的大部分功能，例如，可以直接处理地址，进行位操作；另一方面，它又有高级语言的

特点，C 语言是一种结构化语言，即代码和数据分离，使各程序块互相隔离且影响小，而且具有丰富的数据类型。C 语言的主要特点如下：

1）语言简洁，紧凑灵活。C 语言只有 32 个基本关键字，9 种控制语句，书写格式自由，大量的基本函数（如输入输出等）都是调用一些外部函数实现的。

2）数据类型丰富。基本数据类型有字符、整数、浮点型等，浮点型分单精度和双精度，以解决精度要求和资源浪费之间的矛盾，还可以在基本数据类型的基础上构造出各种数据类型，其灵活性和应用能力优于其他高级语言，如指针可以作为函数参数传递，可以动态地分配内存空间，更简单地处理数组。

3）C 语言的主要结构是函数，函数中一般使用局部变量，使模块中的数据互相隔离，这使得 C 语言基本上是模块化结构语言。

4）语法检查不太严格，程序设计相对自由，对变量的类型使用比较灵活。例如，整型量与字符型数据以及逻辑型数据可以通用，但这要求程序编制人员对程序设计比较熟练，初学者在程序调试时会遇到一些困难。

5）C 程序可移植性好，汇编语言对应某种机器硬件，并且无法移植，而 C 语言编译程序代码 80%是公共的，因此，C 能适用于各种类型的机器。

6）C 源程序和生成的代码质量高。C 源程序书写紧凑，C 目标代码效率比汇编代码效率一般只低 10%～20%，比其他高级语言的系统开销小，适用于编制系统软件，而且对于较大的源程序可以把它分成若干块单独编译，最后将它们链接到一起，这样修改一次就不必将整个程序重新编译一遍，可以大大提高大程序的调试速度。

1.2 C 语言程序的结构与书写规则

为了直观地了解 C 语言程序的结构及书写规则，首先介绍一个 C 语言程序的简单例子。

【例 1-1】用 C 语言编写程序求 a+b。

```
/* This is a C program */
main()                          /* 行 1 */
{int a,b,s;                     /* 行 2 */
 a=100;b=200;                   /* 行 3 */
 s=a+b;                         /* 行 4 */
 printf("sum=%d\n",s);          /* 行 5 */
}                               /* 行 6 */
```

在【例 1-1】中，“/*”和“*/”之间的部分是注释部分。

行 1 是名为 main 的主函数，可运行的 C 程序总是从 main()开始执行。一个标识符后跟一对圆括号，在 C 语言中是函数的标志，圆括号中可放参数，但参数可有可无，【例 1-1】中的 main() 无参数（有关函数的介绍见第 7 章）。

行 2 和行 6 为一对花括号{ }，在 C 语言中，它表示一个函数体、一个分程序或一个复合语句。一个函数体至少要有一对花括号，花括号中可以是任何有意义的语句，也可以是空语句。

行 2 为变量说明语句，变量在使用之前必须说明，故说明语句一般放在函数体或复合语句的前面。

一个变量或函数语句和一条可执行语句总是用分号“;”结尾。

行3为两条语句。C程序中一行可以写多个语句，只要每个语句用“;”号结尾就可以。但是如果一行中语句太多，会降低程序的可读性。

1. C语言程序的构成

1）一个C语言源程序可以由一个或多个源文件组成。

2）每个源文件可由一个或多个函数组成。

3）一个源程序不论由多少个文件组成，都有且只有一个main函数，即主函数。

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

5）每一个说明或每一个语句都必须以分号结尾。但预处理命令、函数头和花括号“{”之后不能加分号。

6）标识符和关键字之间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。

7）源程序中需要解释和说明的部分可加注释，用“/*……*/”引出。

2. C语言程序的书写原则

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，通常表示程序的某一层次结构。{}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，以便看起来更加清晰，增加程序的可读性。

3. C语言程序设计的大致步骤

目前，由于计算机技术的迅速发展和计算机应用的日益普及，使得更多的问题可以用计算机来得到圆满的解决，也使越来越多的人希望了解与程序设计有关的内容。这里将简要地描述与程序设计相关的若干问题。

（1）问题的求解过程

简单地说，在计算机上解决一个问题，就是把解决问题的技术和方法描述成计算机可以实现的一系列步骤，并实施这些步骤，通常包含如下过程：

1）分析问题，建立数学模型。

2）设计解决问题的方法，即算法。

3）设计程序的流程。

4）编制和调试程序。

5）运行程序，得到结果。

原则上说，在解决一个问题时，程序员可以使用任何一种语言来编制程序，但每种语言都有自己的特性和适合的领域，编制出来的程序的效率也有较大差异。长期以来，人们期望能有一种既具有高级语言的特点，又兼有低级语言的高效和很强的硬件操作能力的语言出现，这恰是C语言的主要特征。

（2）运行程序

程序编制完成后，需要进行测试并修改其中存在的错误。通常，可以使用多组数据集合来

运行程序，直到不发生错误为止。确信程序无误后，再将其真正地投入运行。

4. 算法描述

程序设计主要包括算法设计和数据结构设计。算法是对特定问题求解步骤的一种逻辑描述，数据结构则体现了数据及其相互关系，二者是密不可分的。一些典型的数据结构可以方便地进行插入、查找和排序等基本操作，每种操作的实现都依赖于一定的算法。

表示一个算法可用自然语言、传统流程图、结构化流程图、伪代码以及 PAD 图等多种方法，本书则采用较直观的流程图，它依赖于国际标准 GB1526-89 所推荐的一套流程图标准化符号，这些符号与国际标准化组织 ISO 提出的流程图符号一致，图 1-1 列出了部分常用的符号。

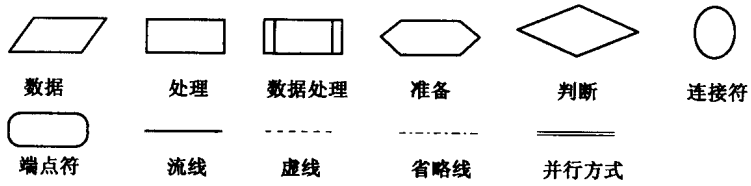

图 1-1　流程图符号

设计一个好的算法通常要考虑到正确性、易读性、健壮性、高效率和低存储量等诸多方面。由于本书以介绍语言为目的，较少涉及复杂的算法，一般也只是给出算法的简单流程或语言描述。

1.3　C 语言的语句和关键字

1.3.1　C 语言的语句

字符是组成 C 语言语句的最基本的元素。C 语言字符集由字母、数字、空格、标点和特殊字符组成。在字符常量、字符串常量和注释中还可以使用汉字或其他可表示的图形符号。C 语言的语句中经常使用的字符一般可分为五类：标识符、运算符、分隔符、常量和注释符。

1. 标识符

在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外，其余都由用户自定义。C 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。

以下标识符是合法的：a，x，b3x，BOOK 1，sum5，_ time

以下标识符是非法的：3s(以数字开头)，s * T(出现非法字符“ * ”)，-3x(以减号开头)，bowy-1 (出现非法字符“-”(减号))

在使用标识符时还必须注意以下几点：

1) 标准 C 不限制标识符的长度，但它受各种版本的 C 语言编译系统限制，同时也受到具体机器的限制。例如，在某版本 C 中，规定标识符前 8 位有效，当两个标识符前八位相同时，则被认为是同一个标识符。

2) 在标识符中，大小写是有区别的。例如，BOOK 和 book 是两个不同的标识符。

3) 标识符虽然可由程序员随意定义，但标识符是用于标识某个量的符号。因此，命名应尽量有相应的意义，以便阅读理解，做到“顾名思义”。

2. 运算符

C语言中有相当丰富的运算符,它把除了控制语句和输入输出语句以外的几乎所有的基本操作都作为运算符处理。不仅包括一般高级语言使用的算术运算符、关系运算符和逻辑运算符,还包括位运算符、自增自减运算符和一些特殊功能的运算符。运算符由一个或多个字符组成,与变量、函数一起组成表达式,表示各种运算功能。

3. 分隔符

在C语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中分隔各个变量。空格多用于语句中各单词之间作间隔符。在关键字和标识符之间必须要有一个以上的空格符作间隔,否则将会出现语法错误,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。

4. 常量

C语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种。

5. 注释符

C语言的注释符是以"/*"开头并以"*/"结尾的串,在"/*"和"*/"之间的即为注释。注释的目的是说明和解释程序或程序中的语句,有助于程序阅读者对程序的阅读和理解,是编程者和程序使用者之间进行沟通的桥梁。注释可出现在程序中的任何位置,可以是一行、几行或一行中的一部分。在调试程序中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。编译时注释部分被忽略,不产生目标代码。

1.3.2 C语言的关键字

C语言中有一种特殊的标识符,它在程序中有特定的含义,用在特定的地方,不能随便移作他用,它们是C语言本身带来的,是由C语言规定的具有特定意义的字符串,这样的标识符称为关键字或保留字,它们都是一些英文单词或缩写。用户定义的标识符不应与关键字相同。

C语言的关键字分为以下几类:

1) 类型说明符:用于定义、说明变量、函数或其他数据结构的类型。如前面例题中用到的 int。

2) 语句定义符:用于表示一个语句的功能。如 if、else 就是条件语句的语句定义符。

3) 预处理命令字:用于表示一个预处理命令。如常用到的 # include、# define 等。

C语言的关键字如下:

auto break case char const continue default do
double else enum extern float for goto if int
long register return short signed sizeof static struct
switch typedef union unsigned void volatile while

本书将在后面的章节中对这些关键字的使用方法作具体介绍。

1.4 C语言程序的编辑与运行

1.4.1 C语言程序的上机环境

C语言程序的上机环境如下:

1) 启动 Turbo C 后,界面如图 1-2 所示。

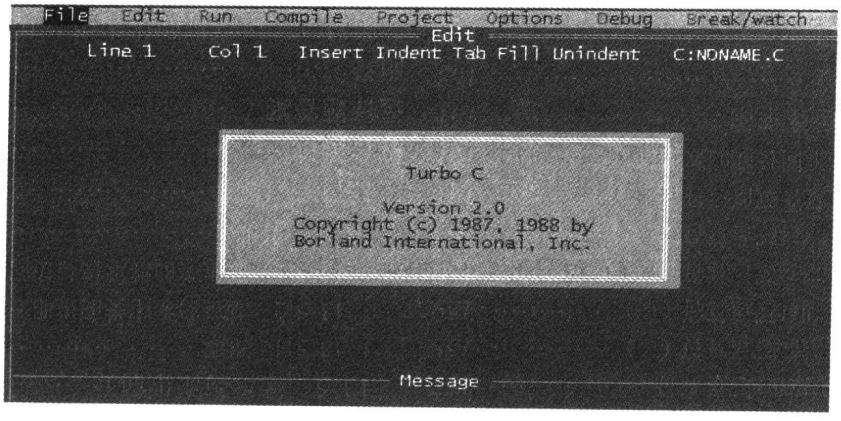

图 1-2

2) File 菜单(文件操作)如图 1-3 所示。

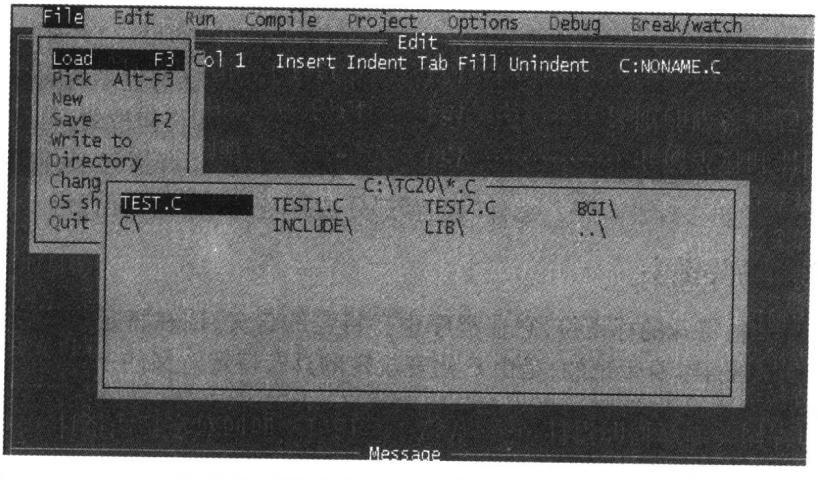

图 1-3

3) Run 菜单(各种程序运行方式)如图 1-4 所示。

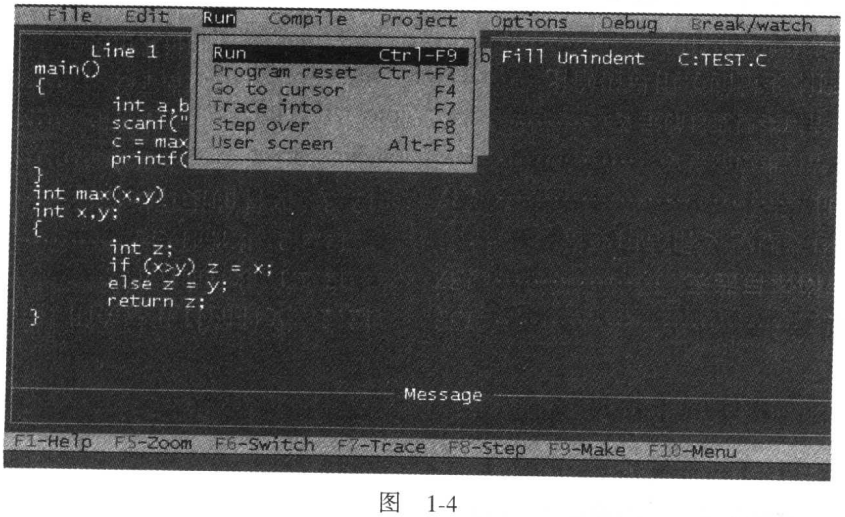

图 1-4

4) Compile 菜单(编译、连接)如图 1-5 所示。

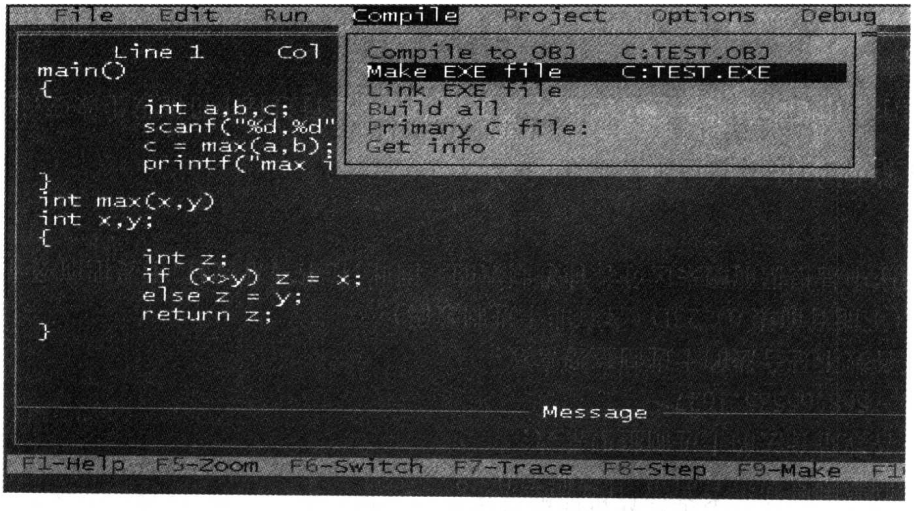

图 1-5

以上介绍的是几个常用的菜单项,余下部分请读者在上机操作时仔细体会。

5) Turbo C 环境介绍如下:

Turbo C 是一个快速、高效的编译软件,它将程序的编辑、编译、连接和执行集成在一起,形成一个集成开发环境。在 Turbo C 的集成环境下,编程和运行等功能均可以通过菜单来完成。

Turbo C 2.0 系统是 DOS 环境下的应用软件。只要在 DOS 系统下进入 Turbo C 2.0 所在的目录,输入 TC 并按〈Enter〉键后,即可进入 Turbo C 2.0 集成环境。按任意键,版本信息消失,屏幕显示见图 1-2。

Turbo C 2.0 集成开发环境的主屏幕由上至下分成四部分:主菜单、编辑窗口、编译信息窗口和功能键提示行。

一般情况下,光标停留在主菜单的 File 项上,可以通过键盘上的〈↑〉、〈↓〉、〈←〉、〈→〉光标移动键移动光标。要选择执行哪项功能,可以将光标移动到那一项上,如 Edit,然后按〈Enter〉键,便可执行该项的功能。如果在启动 Turbo C 时,已经键入了源程序的文件名(如执行 tc.exe),则光标直接处在编辑窗口的左上角,此时,可以直接键入源程序。否则,如果光标处在主菜单上,需要利用〈←〉、〈→〉光标移动键移动光标至 Edit 项上,按〈Enter〉键,光标回到编辑窗口的编辑区时,才能输入程序。

源程序编辑完成后,先按〈F2〉键保存源程序,然后用功能键〈F10〉键激活主菜单(使光标回到主菜单),利用〈↑〉、〈↓〉、〈←〉、〈→〉光标移动键依次移动光标到编译(Compile)、连接(Link)、运行(Run)的菜单项上,按〈Enter〉键,即可执行相应的功能,也可以直接利用功能键提示行上列出的功能键,如〈F9〉为编译键,直接执行各种功能。

1.4.2 C 语言程序的上机步骤

C 语言程序的上机步骤如下:

1) 启动 Turbo C:tc ↙

2) 编辑源程序。在 Turbo C 环境下,将 C 语言源程序通过键盘输入到计算机中,并以文

件形式存盘,以便将来再使用时,直接从磁盘调入到内存中执行,C 源程序的文件扩展名必须是 .c,例如 ex1.c。

3) 编译。编译过程将把 C 语言源程序转换成目标程序。目标程序是以 .obj 为扩展名的文件。若源程序有错,需要对其进行修改,然后重新编译直至没有错误,才算编译成功。

4) 连接。编译后生成的目标文件不能直接执行,需要经过连接之后才能生成可执行代码。在 DOS 系统下,连接后所得到的可执行文件扩展名是 .exe。连接的目的是将目标文件和库函数或其他目标程序连接成可执行程序。

5) 执行程序。连接生成的程序就是可执行文件了。在 DOS 系统下,只要键入可执行文件名,如 ex1,再按回车键就可以了。

6) 退出 Turbo C,按〈Alt + X〉键。

综上所述,上机步骤为:编辑(.c)——编译(.obj)——连接(.exe)——执行。

1.4.3 高级语言程序的执行过程

高级语言的程序并不是计算机能够直接识别并执行的指令集,而是一个文本文件,称为"源程序"文件。为了能够运行它,必须将其转换为机器指令,实现这一转换的程序就是编译程序或解释程序。

编译和解释是指程序的两种执行方式,或者说是对源文件的处理方式。从理论上说,每种语言都可以采取这两种形式来处理,而实际上,多数语言只使用了其中一种方式。例如,早期的 BASIC 语言采用了解释方式。而 C 语言则使用了编译方式,为此,两种系统分别提供了解释程序和编译程序。

一个解释程序将逐行处理源程序。例如,使用解释 BASIC 语言时,可以用 Run 命令直接运行源程序。此时,系统自动启动解释程序来处理用户的源代码,每次将一行源代码翻译成机器指令并执行,直到源程序全部处理完毕。此方式使程序的执行速度较慢,且离不开源程序。

一个编译程序一次读入整个源文件,并将其全部代码转换成目标代码,形成可执行的文件,从而彻底脱离源程序,并有较快的程序执行速度。实现时,高级语言源程序到可执行程序的转换通常都由两个程序完成:其一是编译程序,用于直接处理源程序,生成一个二进制代码文件,称为目标文件;另一个是连接程序,它处理由编译程序所生成的目标代码文件,最终形成一个可执行文件。

1.4.4 程序调试方法

程序设计过程中难免出错。设计完成的计算机程序必须经过测试来确定其是否可以正确地工作,这一过程称为调试。程序错误一般有两类:一类是语法错误;另一类是运行错误。语法错误是由于没有按照程序设计语言的语法规则编写语句所致,一般通过编译和连接就会找到,可根据编译系统给出的提示进行修改,直至编译通过,说明程序没有语法错误。当程序已顺利执行,但得不期望的结果时,说明程序存在运行错误。这类错误除由类型错误造成的外,其他都归为逻辑错误。计算机不能指出运行错误,只能由编程者自己查找出程序的错误所在。运行错误往往是由于编程者所采用的变量类型或(部分或全部)算法有问题,而导致开始或中间结果不正确。这就要求有能快速找出错误的方法,Turbo C 集成环境提供了方便灵活的手段控制程序的执行,可在任意语句行上暂时或永久停止程序的执行(程序停止执行的语句行称

为断点),并可追踪、观察中间(变量)结果,下面介绍一些调试方法。

1. 设置和使用观察变量

找出程序运行错误的一种有效的方法是设法了解程序执行过程中某些变量或表达值的变化,Turbo C 集成环境提供了观察变量的途径。选 Break/Watch 菜单项中的 Add watch 项或同时按〈Ctrl + F7〉键,会出现一个输入框,输入欲观察的变量名或表达式后,它们就会在 watch(屏幕底部)窗口中。重复使用 Break/Watch 菜单项中的 Add watch 项或同时按〈Ctrl + F7〉键可以在 Watch 窗口添加多个变量或表达式。变量或表达式设置完成后,执行程序。Watch 窗口中变量或表达式的值随程序的执行将发生变化,由于程序执行速度非常快,往往观察不到这种变化。因此,需要人为地控制程序逐条地执行语句,以观察变量或表达式的值。这可以通过单步执行程序、设置和使用断点来实现。

2. 单步执行程序

选择 Run 菜单下的 Step over 命令项或按〈F8〉键,亮条就出现在主函数处,说明程序从主函数开始执行。每次选 Run 菜单中的 Step over 命令项或按一次〈F8〉键则执行一行程序,遇到函数时不进入函数(定义),即将函数作为一条语句或一个表达式处理。

执行 Run 菜单中的 Trace into 命令项或按〈F7〉键也是跟踪执行程序,与单步不同的是,遇到函数时进入其内部逐行执行。

执行 Run 菜单中的 Goto cursor 命令项或按〈F4〉键,从主函数开始执行程序到光标所在处(跟踪到光标)。欲让程序在某条语句暂停,只需将光标移到那条语句所在行,选 Run 菜单中的 Goto cursor 命令项或按〈F4〉键,程序即可暂停在光标所在行。然后再移动光标到欲暂停的语句行上,再选 Run 菜单中的 Gto cursor 命令项或按〈F4〉键,由此可以方便地调试程序。

3. 设置和使用断点

通过设置断点,可以分段解决问题,把出现问题的范围缩小。调试程序时,可以在程序中设置多个断点。当程序执行到断点时,就会在有断点的那行上暂停程序的执行,显示所需观察变量或表达式的当前值,据此可判定断点前的程序是否有问题。

设置断点的步骤为:

1) 在编辑窗口中将光标移到欲设置断点的那一行上。

2) 选 Break/Watch 菜单项中的 Toggle breakpoint 命令项或同时按〈Ctrl + F8〉键,被设置断点的一行呈高亮度显示。

3) 欲设多个断点,重复步骤 1)、2),断点所在的行均呈高亮度显示;

4) 要删除断点,将光标移到欲删除断点所在行,选 Break/Watch 菜单项中的 Toggle - le-breakpoint 命令项或同时按〈Ctrl + F8〉键。

以上方法可结合使用。程序连续执行时,一遇到断点就会暂时停止,并返回编辑状态。断点不会因为被执行而清除。

程序的调试技巧与程序设计一样重要。在调试过程中可以积累编程经验和技巧,为设计优秀的程序奠定基础。

1.5 习题

1. 目前最流行的 C 语言有哪几种版本?