

C 程序设计

CHENGXUSHEJI

主编/ 王 舒

主审/ 李志蜀

编著/ 王 舒 夏 欣

张 盈 孙亚飞

C 程 序 设 计

主 编 王 舒
主 审 李志蜀
编 著 王 舒 夏 欣
张 盈 孙亚飞

四川大学出版社

责任编辑:陈正权
责任校对:贾朝辉
封面设计:文绍安
责任印制:石大明

图书在版编目(CIP)数据

C 程序设计/王舒主编,一成都:四川大学出版社,
2001.2

ISBN 7-5614-2074-9

I.C... II.王... 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(2001)第 04199 号

内容简介

本书从 C 语言作为大学本科第一计算机语言的角度出发,按循序渐进原则安排内容,采用通俗易懂的讲解方法,并辅以丰富的典型例题,全面系统地讨论了 C 语言及其程序设计方法。全书以美国国家标准 C 语言(ANSI C)为基本内容,以当前广泛使用的 Turbo C 编译系统为实现版本,共设 12 章,每章后附有习题。

本书可作为大专院校计算机专业和非计算机专业学生的 C 语言程序设计教材,也可作为其他学习 C 语言人员的自学教材或培训教材。

书 名 C 程序设计

作 者 王舒 主编
出 版 四川大学出版社
地 址 成都市一环路南一段 24 号 (610065)
印 刷 四川省地勘局测绘队印刷厂
发 行 新华书店经销
开 本 787mm×1092mm 1/16
印 张 20.625
字 数 525 千字
版 次 2001 年 2 月第 1 版
印 次 2001 年 2 月第 1 次印刷
印 数 0 001~4 000 册
定 价 29.90 元

◆读者邮购本书,请与本社发行科
联系。电 话:5412526/5414115/
5412212 邮编:610064

◆本社图书如有印装质量问题,请
寄回印刷厂调换。

版权所有◆侵权必究

前 言

C语言是一种通用的程序设计语言,它的结构简单,数据类型丰富,运算灵活方便。C语言又是一种理想的结构化语言,它既具有高级语言的优点,又具有低级语言的特点,因此可用于编写高效简洁、风格优美的应用程序,以及计算机系统程序(例如著名的UNIX操作系统就是用C语言编写的)。用C语言编写的程序,具有运算速度快、效率高、目标代码紧凑、可移植性好等优点。

作为当今最为流行的一种计算机语言,C语言越来越受到程序设计人员的青睐。广大计算机应用人员迫切要求学习C语言,各高等院校日益重视C语言程序设计教学,不少学校已经把C语言作为大学本科“第一计算机语言”进行教学。为适应这一新需求,我们在吸取国内外同类教材优点的基础上,结合C语言程序设计的教学和科研经验,编写了本书。

全书具有以下几个特点:

1.从大学本科“第一计算机语言”的角度安排本书的体系结构,力求入门容易、通俗易懂。在第1章中介绍了算法、算法的表示及结构化程序设计等基本概念,以备后续章节使用。

2.为分散难点,在章节安排上循序渐进,由每章引入一些新概念,并由浅入深、逐层展开地为读者提供大量典型的C语言示例。

3.在例题和习题的选择上贯穿以应用为主线的原则,尽量涉及C语言在各种领域的应用,以及程序设计的各种基本算法,使读者将C语言知识和编程能力融会贯通,灵活自如地把C语言应用到学习和工作中,以达到学以致用目的。

4.全书以美国国家标准C语言(ANSI C)为基本内容,以当前广为使用的Turbo C编译系统为实现版本,所有例题程序均上机调试通过,并附运行结果。这些程序在不同的计算机系统上使用,可能会有细小差别,读者可参照所用计算机系统手册,自行修改。

本书共分12章。第1章、第7章、第11章由张盈编写,第2章、第3章、第8章由夏欣编写,第4章、第5章、第6章、第12章由王舒编写,第9章、第10章由孙亚飞编写。全书由王舒主编并最后修改定稿,书中所有例题程序由夏欣上机调试通过。并请教育部工科计算机课程教学指导委员会委员、四川大学计算机学院院长李志蜀教授审阅。

由于时间仓促,加之计算机科学技术发展迅速、日新月异,书中难免有不少缺点和错误,恳请专家和读者批评指正。

编 者

2001年1月于四川大学

目 录

第1章 C语言概述

- 1.1 程序及程序设计的初步知识..... (1)
- 1.2 C语言的发展及特点..... (5)
- 1.3 C语言的格式及结构特点..... (7)
- 1.4 C语言程序的开发过程..... (9)

第2章 基本语法单位及基本数据类型

- 2.1 基本语法单位..... (13)
- 2.2 C的数据类型..... (14)
- 2.3 常量与变量..... (15)
- 2.4 整型数据..... (16)
- 2.5 实型数据..... (19)
- 2.6 字符型数据..... (20)
- 2.7 变量的初始化..... (24)

第3章 运算符与表达式

- 3.1 C运算符简介..... (26)
- 3.2 算术运算..... (26)
- 3.3 赋值运算..... (29)
- 3.4 位运算..... (33)
- 3.5 其他运算符..... (39)
- 3.6 各类型数据的混合运算..... (40)
- 3.7 运算符的优先级和运算顺序..... (42)

第4章 最简单的C程序设计

- 4.1 C语句概述..... (44)
- 4.2 表达式语句..... (45)
- 4.3 复合语句..... (46)
- 4.4 数据输出..... (47)
- 4.5 数据输入..... (53)
- 4.6 程序举例..... (57)

第5章 逻辑运算和选择结构程序设计

5.1 关系运算符和关系表达式	(60)
5.2 逻辑运算符和逻辑表达式	(61)
5.3 条件运算符和条件运算	(63)
5.4 if 语句	(64)
5.5 switch 语句和 break 语句	(70)
5.6 程序举例	(72)

第6章 循环结构程序设计

6.1 while 语句	(80)
6.2 do-while 语句	(83)
6.3 for 语句	(85)
6.4 几种循环的比较	(87)
6.5 循环的嵌套	(89)
6.6 break 语句在循环中的应用	(91)
6.7 continue 语句	(92)
6.8 goto 语句	(93)
6.9 程序举例	(94)

第7章 数 组

7.1 一维数组的定义及引用	(102)
7.2 二维数组与多维数组的定义及引用	(109)
7.3 字符数组	(113)

第8章 函 数

8.1 C 语言程序的结构	(124)
8.2 函数的定义	(125)
8.3 函数的调用	(131)
8.4 函数的嵌套调用	(138)
8.5 递归函数	(141)
8.6 数组与函数	(146)
8.7 变量的存储类型	(153)

第9章 编译预处理

9.1 宏替换	(175)
9.2 文件包含	(180)
9.3 条件编译	(185)
9.4 预定义宏	(189)
9.5 运算符#和##	(190)

第 10 章 指 针	
10.1 指针的概念	(193)
10.2 指针变量的定义	(194)
10.3 指针变量的引用	(196)
10.4 指针作为函数参数	(200)
10.5 指针与数组	(204)
10.6 指针运算	(228)
10.7 指向字符串的指针	(234)
10.8 指针与函数	(238)
10.9 指针数组与指向指针的指针	(242)
第 11 章 结构体、共用体及枚举类型	
11.1 概述	(250)
11.2 结构体的定义	(251)
11.3 结构体的引用	(252)
11.4 结构体变量的初始化	(256)
11.5 结构体数组	(257)
11.6 指向结构体类型数据的指针	(262)
11.7 结构体和函数	(266)
11.8 结构体和指针的综合应用	(271)
11.9 共用体	(274)
11.10 枚举类型	(279)
11.11 用 typedef 定义数据类型	(283)
第 12 章 文 件	
12.1 C 文件概述	(289)
12.2 文件类型指针	(290)
12.3 文件的打开和关闭	(291)
12.4 文件的读写	(293)
12.5 文件的定位	(301)
12.6 出错的检测	(303)
12.7 非缓冲文件系统	(304)
附 录	
附录 A C 语言中的关键字	(309)
附录 B 运算符优先级和结合方向	(310)
附录 C C 库函数	(311)
附录 D Turbo C 编译出错信息	(318)

第 1 章 C 语言概述

C 语言是非常流行的计算机程序设计语言,由于其性能卓越,早已风靡全球。同时,作为一门程序设计基础课的教学语言,它又具有很多无可比拟的优势:其一,它是目前实际程序设计中
使用最广泛的语言之一,有很好的兼容性;其二,目前许多流行的新一代的程序设计语言都借鉴了 C 语言
的思想和描述方式,有些本身就是 C 语言的扩充和发展,掌握 C 语言对学习和掌握其他语言大有裨益;其三,在众多的程序设计语言中,通过学习 C 语言更易于理解程序设计原理、掌握程序设计的基本思维方法。

本章是引论性的,首先讲解程序及程序设计的初步知识,然后介绍 C 语言的发展和特点,以及 C 语言的格式和结构特点,最后给出在计算机上开发 C 语言程序的过程。

1.1 程序及程序设计的初步知识

初学者在刚刚开始学习高级语言程序设计时,最先遇到的问题可能是下面的几种:

什么是程序?

什么是程序设计语言?

程序与程序设计语言之间存在怎样的关系?

怎样才能进行程序设计?

这些问题都是在进行程序设计之前亟待解决的。下面,我们就围绕着上述问题阐述关于程序及程序设计的初步知识。

今天,计算机对于人们已不再陌生,它早已同家用电器一样走进了千家万户,成为大众化的工具,深刻地影响着人们的生产和生活。计算机能有现在的发展,正是因为人们为它开发设计出了无数或简单或复杂的程序,使计算机能够完成各种工作。那么,什么是程序呢?

众所周知,计算机是一种具有内部存储能力、由程序自动控制的电子设备。人们把需要计算机完成的工作写成一定形式的指令序列,并把它们存储在计算机的内部存储器中,当给出启动命令之后,它就按指令操作顺序自动运行。人们就把这种可以连续执行一条条指令的集合称为“程序”。可以说,程序就是人与机器之间进行交流或“对话”的语言。

语言是不同主体之间进行信息交流的工具。计算机从诞生之日起,就存在着与人进行交流对话的问题,人和计算机之间所用的语言就是程序设计语言,它是专门为人和计算机之间进行交流而设计的,例如 C 语言。用程序设计语言来描述问题的求解过程,以及对其中参与运算的数据进行合理地组织和安排,就叫做程序设计。

如何进行程序设计呢?设计程序,是一项十分艰苦的脑力劳动,如同解决一个棘手的问题。一般要经过以下几个步骤:

(1) 分析问题,提出解决问题的可行方案。

(2) 确定算法。针对提出的可行方案确定解决问题、完成任务的每一个细节步骤。好的算法是解决问题的关键,它应当满足:包含有限个操作步骤(有穷性);每一条指令都需要有确

定含义,不能模棱两可(确定性);算法中的操作是可以在执行有限次后实现的(可行性);有零个或多个输入,有一个或多个输出等。

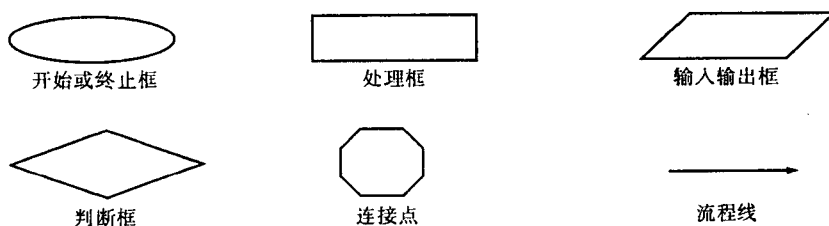
(3) 编程。使用程序设计语言把上述算法严格的描述出来(即写成程序),输入到计算机里并存盘。

(4) 在计算机上运行并调试这个程序。如果在运行过程中发现了错误,就仔细分析出错的原因,弄清楚后更正错误,再次运行该程序,直到程序准确无误并得到正确的输出结果为止。

(5) 总结。写出书面报告。

程序设计过程中,有很多种方法可以用来描述算法,最常见的是流程图和 N-S 图。

(1) 流程图。流程图是很好的描述算法的工具,它由几种基本框组成:



用流程图来表示算法,很直观、很形象,也很简单,但是这种流程图对于流程线的走向没有任何限制,在描述复杂算法时所占篇幅较多。比较适合于手工绘制。

(2) N-S 图。它是随着结构化程序设计方法的出现而产生的一种新的流程图形式。这种流程图完全去掉了流程线,算法中的每一步都用一个矩形框来描述。把一个一个一个的矩形框按执行的次序连接起来就是一个完整的算法描述。但是,手工画 N-S 图很麻烦,它更主要面向使用计算机辅助工具的场所。

对于程序设计,目前使用较广的程序设计方法有两大类,即结构化程序设计方法和面向对象程序设计方法。C 语言是结构化的程序设计语言,它提供了完善的程序控制结构和模块化程序设计手段。而结构化程序设计方法就是只采用三种基本的程序控制结构来编制程序,从而使程序结构清晰、易于阅读、易于验证。这三种基本结构是顺序结构、选择结构和循环结构。下面我们就结合流程图和 N-S 图简单介绍一下这三种基本结构:

① 顺序结构。即自顶向下,顺序执行程序中的每一条语句,没有分支没有转移,如图 1.1。

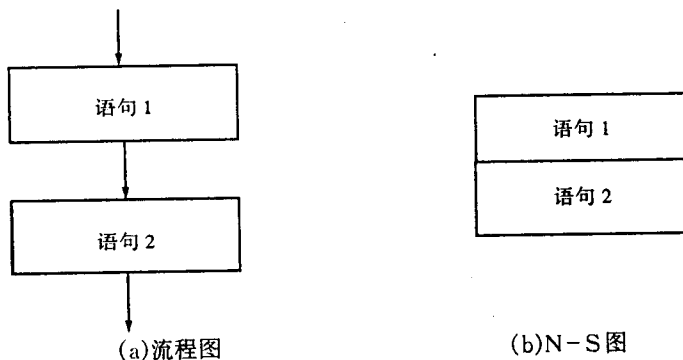


图 1.1

② 选择结构。程序中当执行到这些语句时,将根据给定的条件的成立与否去执行不同分支中的语句:当判断条件表达式为真(即条件成立)时,执行语句 1,否则执行语句 2。注意:在程序的某一次执行过程中,只能执行到一条分支上的语句,如图 1.2。

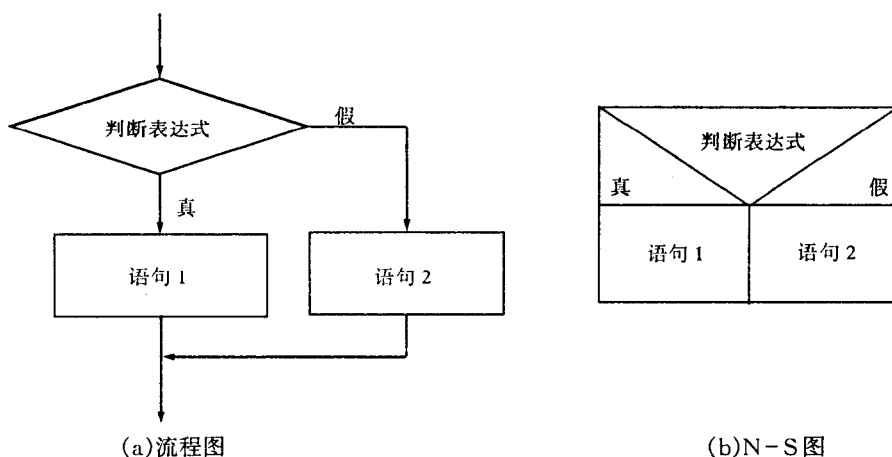


图 1.2

③ 循环结构。一组特殊的语句序列,执行时将根据给定的条件,决定同一组语句(循环体部分)重复执行 0 次或多次。循环结构分为当型循环和直到型循环两种。当型循环的特点是,当指定的判断条件表达式为真时,就执行循环体,否则退出循环。如图 1.3。

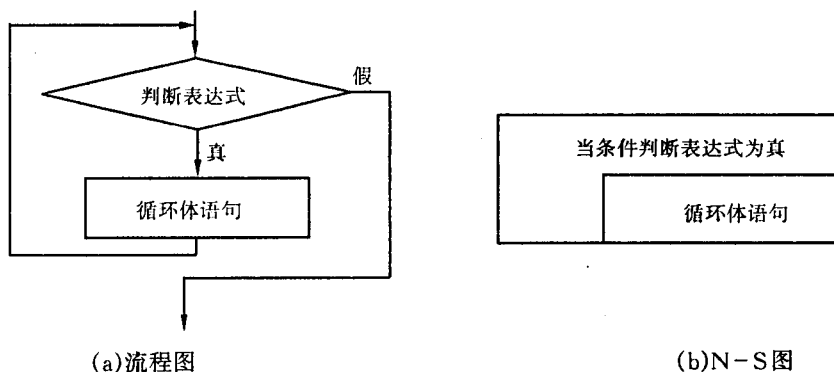


图 1.3

直到型循环的特点是,先执行循环体,直到指定的判断条件表达式为假时,退出循环。如图 1.4。

在实际程序设计中,常常是三种基本结构有机地组合在一起,由它们所形成的算法结构可用来解决任何复杂问题。请看下面的例子。

【例 1.1】输入一个年份,判断该年是否闰年。要求分别用流程图及 N-S 图表示出算法。这道题的解题思路是:先输入一个数,把它放到变量 year 中。然后对 year 进行判断,如果某年的年份能被 4 整除但不能被 100 整除,或者能被 400 整除,该年就是闰年,输出“是闰年”,否则

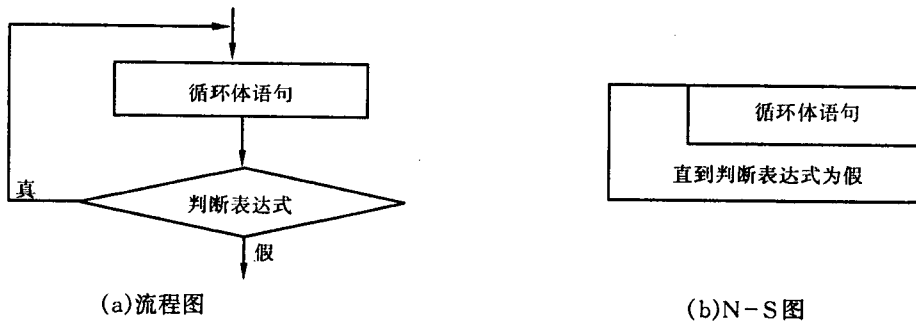


图 1.4

输出“不是闰年”,其流程图如图 1.5 及 N-S 图如图 1.6。

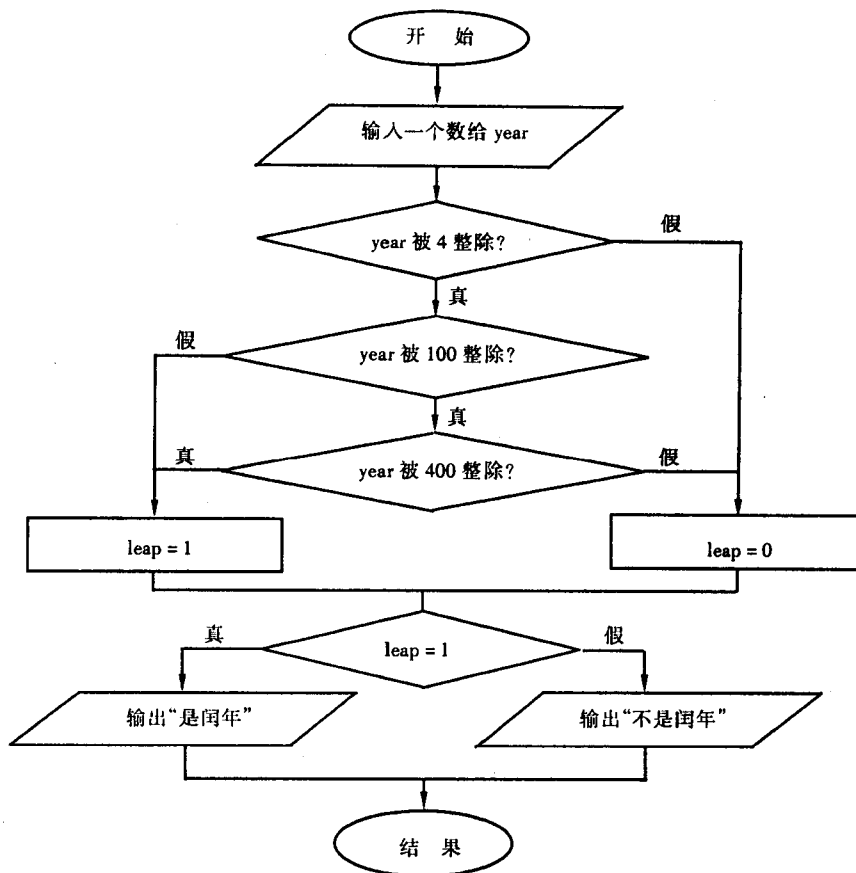


图 1.5

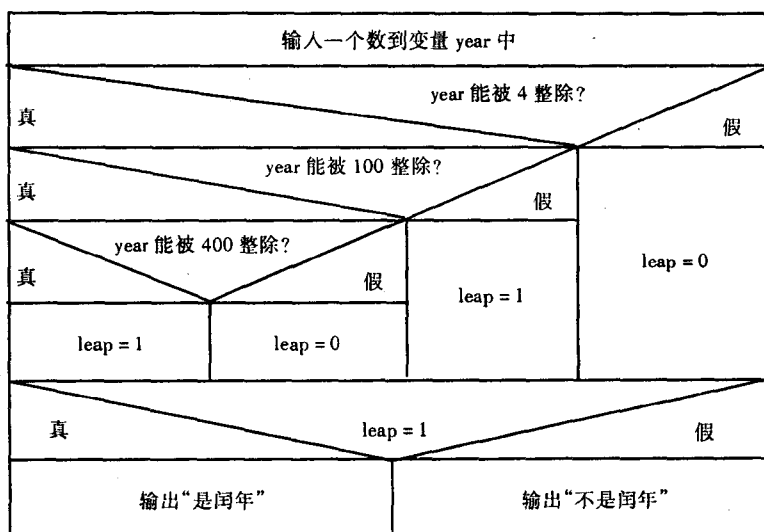


图 1.6

1.2 C 语言的发展及特点

C 语言是非常流行的计算机程序设计语言。其卓越的性能,良好的可读性,使之成为最受欢迎的编程语言之一。它的应用范围很广,既可以用来编写系统软件,又可以用来编写应用软件。

说到起源,C 语言是在 B 语言的基础上发展而来的。在 C 语言的发展过程中,UNIX 系统功不可没。C 语言曾是 UNIX 系统的主力语言,它与 UNIX 系统有着互相依存、休戚与共的紧密关系。UNIX 系统是美国贝尔实验室的 K. Thompson 和 D. M. Ritchie 从 1969 年开始用不到两个人年的时间研制成功的。在 UNIX 系统上实现了汇编语言后,UNIX 系统又以汇编语言编写。但由于汇编语言的不可移植性,并且描述问题的效率不如高级语言,特别是可读性差,所以 K. Thompson 在 PDP-11/20 上设计出并实现了很简单又很接近硬件的 B 语言,并且用 B 语言编写了 UNIX 操作系统和绝大多数实用程序。B 语言主要思想源于 BCPL(Basic Combined Programming Language)语言,BCPL 是 M. Richards 基于 CPL(Combined Programming Language)语言在 1969 年发表的一种单一数据类型语言,至今仍显示出足够的生命力。不过,由于种种原因,B 语言没有盛行起来。例如,现代的高级语言要求有强有力的类型结构,B 语言过于简单,无类型,在描述客观世界许多事物时会遇到相当多的困难。于是 1972 年至 1973 年间,贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出了 C 语言。C 语言既保持了 B 语言的优点(精练,接近硬件),又克服了它的缺点(过于简单,数据无类型等)。最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工作语言而设计的。1973 年,K. Thompson 和 D. M. Ritchie 两人合作把 UNIX 操作系统的 90% 以上程序用 C 语言改写。后来,C 语言又做了多次改进,但还是在贝尔实验室内部使用。直到 1975 年 UNIX 操作系统第六版公布后,C 语言的突出优点才引起人们广泛注意。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》,于是 C 移植到其他机器时所需做的工作大大简化

了,这也推动了 UNIX 操作系统迅速地在各种机器上推广实现。C 语言和 UNIX 可以说是一对孪生兄弟,在发展过程中相辅相成。1978 年以后,C 语言先后移植到大、中、小型微型计算机上,渐渐独立于 UNIX 和 PDP 而蓬勃发展。现在已经成为世界上应用最广泛的计算机程序设计语言之一。事实上,现今流行的操作系统都是用 C 语言开发的,例如 Linux,几乎全部用 C 编写。

C 语言之所以起名为 C,有人说因为 C 是由 BCPL 经过 B 演化来的,所以延接 B(取 BCPL 的第二个字母)而成为 C,而另有人说这是按字母 A、B、C、D……的顺序,已有 B,接下去就应该是 C。C 语言的开发者 D. M. Ritchie 对这两种说法均不置可否。C 之后的发展是叫 P 还是叫 D,也任人猜测。不过从它仅一个字母的名字倒常常令人联想到它的简洁性。

尽管 C 语言是一种计算机高级语言,但从被开发之始,它就介于高级语言和汇编语言之间,因而它的身上既有高级语言的特性(程序有较好的可读性和可移植性),又有低级语言的优点(可以对硬件直接编程)。具体地讲,C 语言的特点可归纳为以下几个方面:

(1) 语言本身简洁、灵活,便于学习和使用。C 语言共有 32 个关键字,9 种控制语句;程序书写形式自由,一般用小写字母表示,压缩了一切不必要的成分,使得编译程序小而精。在表示方法上力求在明白易懂的前提下简单易行,如使用花括号对 { } 来表示复合语句的开始和结束。

(2) 语言表达能力强,能对硬件进行编程,可实现汇编语言的大部分功能。C 语言是面向结构程序设计的语言,通用性好,不局限于某种机器。它可以直接处理字符、数字、地址,也可以完成通常由硬件实现的算术和逻辑运算。它反映了当前计算机的性能,足以取代汇编语言来编写各种系统软件和应用软件。

(3) 运算符类型丰富。C 语言共有 34 种运算符,它把括号、赋值、强制类型转换等都作为运算符处理,从而使表达式类型多样化,可以实现在其他语言中难以实现的复杂运算。

(4) 数据类型丰富,并有多种数据结构。C 语言中的数据类型大致可分为两大类:一类是如整型、实型、字符型等的简单类型;另一类是在简单类型基础上按层次产生的各种构造类型,如数组类型、指针类型、结构体类型和共用体类型等,能用来实现各种复杂的数据结构的运算,如链表、树、栈等。

(5) 具有结构化的控制语句。如 if...else 语句、do...while 语句、switch 语句、for 语句等,功能很强,足以描述结构良好的程序。此外,用函数作为程序模块以实现程序的模块化,是结构化的理想语言,符合现代编程风格要求。

(6) 高效率的目标代码。高级语言能否用来描述系统语言,特别是操作系统、编译程序等,除了要求语言表达能力强之外,很重要的一个因素是语言生成的代码质量如何。如果代码质量低,系统开销就大,无实用价值。许多试验证明,针对同一问题,用 C 语言编写的程序生成代码的效率比用汇编语言编写的仅低 10%~20%。

(7) 同汇编语言相比,用 C 语言写的程序可移植性好。可移植是指程序可以从一个环境不加或稍加改动就可以搬到另一个完全不同的环境中运行。汇编语言因为依赖机器,所以根本不可移植。C 语言基本上不作修改就能用于各种型号的计算机和操作系统。

(8) 程序设计较自由,语法限制不严格。如:对数组下标越界不做检查,由程序编写者自己保证程序的正确性。一些变量的数据类型可以通用,如整型量与字符型数据及逻辑型数据。不像其他的高级程序设计语言,对语法检查严格,能检查出几乎所有的语法错误,而 C 语言放宽限度,在程序设计上灵活自由,相应地,检查错误的任务也转到了编程者的身上。因而要求

读者在编程时仔细检查自己的程序,保证其正确,而不要过分依赖 C 语言编译程序去查错。

上面我们介绍了 C 语言的最容易理解的一般特点,至于 C 语言内部的其他特点以后将结合各章的内容作详细介绍。正是由于以上几个突出优点,C 语言吸引人们对它倾注越来越多的关心,因而成为国际上广泛流行的,很有发展前途的计算机高级程序设计语言。

1.3 C 语言的格式及结构特点

在这一节里,我们先介绍几个简单的 C 程序,使读者对 C 程序有一个初步的印象,以便从中分析和了解 C 语言的格式及结构特点。

【例 1.2】一个简单的 C 程序。

```
main( )                                /* 主函数 */
{                                       /* 函数体开始 */
    printf("Hello World !");          /* 输出一字符串 */
}                                       /* 函数体结束 */
```

上面程序的运行结果是在屏幕上输出一行信息:

Hello World !

其中,main 表示主函数,C 语言规定每一个 C 程序都必须有一个 main 函数,其后的一对圆括号不能省略;在一对花括号之间的部分是函数体,函数体是程序的定义和执行部分,左花括号表示函数体的开始,右花括号表示函数体的结束。在本例中函数体内只有一条输出语句 printf("Hello World !"),其中的 printf 是 C 语言中的输出函数,用于输出一串字符。

语句最后有一分号。“/* */”部分表示注释,可以加在程序的任何部分,一般放在 C 语句的右边。

注意:一个 C 程序可由若干个函数(这些函数可在一个或多个文件中)组成,其中至少有一个是 main 函数。如果一个完整的 C 程序中没有 main 函数,编译时将出现错误信息。

【例 1.3】给定圆的半径,求圆面积及周长。

```
# include <stdio.h>
# define PI 3.1415926                    /* 定义 PI 的值 */
main( )                                  /* 主函数 */
{                                       /* 函数体开始 */
    float r,a,c;                        /* 定义变量。r:半径;a:面积;c:周长 */
    r=2.5;                              /* 给定 r 的初值 */
    a=PI*r*r;                           /* 计算面积 a */
    c=2*PI*r;                           /* 计算周长 c */
    printf("r= %f,a= %f,c= %f\n",r,a,c); /* 输出结果 */
}                                       /* 函数体结束 */
```

运行结果如下:

r=2.500000,a=19.634954,c=15.707963

在这个程序里,有两个用“#”开头的语句,它们的含义将在后面的章节中详细介绍,这里略过不讲。函数体由五条语句组成。第一句是定义变量部分,说明 r、a、c 是实型(浮点型)数据。第二句是赋值语句,使 r 的值为 2.5。第三句和第四句都是求值,分别求面积和周长。第

五句的作用是输出结果,其中的“%f”是输入输出的格式字符串,用来指定输入输出时的数据类型和格式,这里的“%f”表示输出浮点数且保留 6 位小数(float),在执行输出时其所在位置为一个具体的浮点数;括号内右边的 r、a、c 是要输出的变量,它们的值分别是 2.500000、19.634956、15.707965,从而得到上面的运行结果。

【例 1.4】从键盘输入两个数,求其中的最大值。

```
main( )                /* 主函数 */
{
    int a,b,c;           /* 定义三个变量为整型 */
    scanf("%d,%d",&a,&b); /* 输入变量 a,b 的值 */
    c=max(a,b);          /* 调用 max 函数,并将结果赋给变量 c */
    printf("max=%d",c);  /* 输出结果 c */
}                        /* 主函数体结束 */

/* 下面定义 max 函数,函数型为整型,形式参数 x,y 也为整型 */
int max(int x,int y)
{
    int z;               /* 定义变量 z 为整型 */
    if (x>y)             /* 条件语句,如果 x 大于 y */
        z=x;            /* 将 x 的值赋给 z */
    else                 /* 条件语句中的转折,表示“否则” */
        z=y;            /* 将 y 的值赋给 z */
    return(z);           /* 返回 z 的值,通过 max 带回调用处 */
}
```

该程序运行结果如下:

3,4↵ (输入 3 和 4 给 a 和 b,↵表示回车换行)

max=4 (程序输出结果)

这个程序包含两个函数,一个主函数 main 和一个被调用的子函数 max。

函数 max 的作用是将 x 和 y 中较大的值赋给变量 z。return 语句将 z 的值返回给主函数 main。主函数中的 scanf 是输入函数,它和 printf 一样都是 C 语言提供的输入输出函数,作用是从标准输入设备(键盘)接收数据 a 和 b,并按格式转换说明符将数据转换成相应的格式放到对应的变量中。&a 和 &b 中的“&”的含义是“取地址”。整个语句(scanf 所在行)的作用是将两个数值分别输入到变量 a、b 的地址所标志的单元中,也就是输入数值给 a 和 b。&a 和 &b 前面的“%d,%d”的含义同“%f”类似,表示输入十进制整数(decimal),因此用于指定输入的两个数据应按十进制整数形式从键盘输入。

main 函数中第三句 c=max(a,b) 调用了 max 函数。在调用时将实际参数 a 和 b 的值分别传给 max 中的形式参数 x 和 y,经过执行 max 函数后得到了一个返回值 z,然后将 z 的值传递给变量 c,最后输出 c 的值。

上面提到的一些术语(如函数调用,形式参数,实际参数等),这里不予详细介绍,在以后的章节中读者自然会迎刃而解。这里只是想让读者了解一个 C 程序的基本格式和大体结构。

综合上面三个小程序,归纳如下:

(1) 一个 C 程序由一个或多个函数组成,其中必须有一个用“main”命名的主函数。一个

程序总是从它的主函数开始执行的。因此,函数是构成 C 程序的基本单位,是完成某个整体功能或某个模块的最小组成,相当于一个子程序。被调用的函数(子函数)可以是系统提供的库函数(例 1.4 的 `scanf` 和 `printf`),也可以是用户根据需要自己编辑完成的自定义函数(例 1.4 中的 `max`)。函数能够帮我们实现任何特定的功能,掌握函数对学习 C 程序有极其重要的意义。C 程序的全部工作就是由大大小小的函数完成的,可以说,C 是函数式的语言。

(2) 每个函数由头部和函数体两部分组成,函数头部说明返回值的类型、函数名和形式参数;函数体由若干条 C 语句组成;以“`{`”和“`}`”作为函数的开始和结束标志。

(3) 每个 C 语句以“`;`”(分号)结尾,分号是组成 C 语句的重要部分。C 语句是完成某种程序功能的最小单位。

(4) C 程序的书写格式比较自由,一个语句可以写在一行,也可以分成几行书写;一行内可以写一个语句,也可以写多个语句。但过于自由的程序书写格式,可能让人不易读懂程序,初学者应该从一开始就养成一个良好的习惯,在适当的地方分行、保持格式对齐,使程序清晰、易读,能够很好的体现程序内部的层次结构,反映程序各部分间的关系。

(5) 可以在程序的任何位置用“`/* ..... */`”对 C 程序中的任何部分作注释。“`/*`”和“`*/`”必须成对出现,但“`*`”和“`/`”之间不可以有空格。在程序中适当的地方加上注释,可以增加程序的可读性。

1.4 C 语言程序的开发过程

C 语言是一种高级程序设计语言。用高级语言编写的程序(即源程序),计算机是不能直接运行的,计算机只能识别和运行由二进制代码形式组成的机器语言程序。为了使计算机能够完成高级语言程序所描述的工作,必须把高级语言源程序转化成机器可以识别的二进制代码形式的机器语言目标程序,这个过程必须借助于专门的系统软件才能完成,这个系统软件就叫做编译系统。编译系统将高级语言书写的程序翻译成机器语言程序,然后让计算机运行。一个 C 语言源程序要生成可执行文件,一般需要经过以下几个步骤:

(1) 编辑。可使用任何文本编辑软件(如 `Edit`)输入 C 语言程序清单,构成源程序文件,存盘时文件名应以“`.c`”为扩展名。

(2) 编译。使用 C 语言的编译系统,对源程序文件进行编译。编译系统在编译时,对源程序进行语法和语义检查,并在发现错误时输出“出错信息”,告诉用户第几行有什么样的错误。一般 C 语言程序在编译过程发现的错误有两大类:一是局部语法错误,如拼写错误、少写了必要的符号;二是程序里上下文关系方面的错误,如使用变量时却没有预先定义等。用户可回到编辑状态对源程序修改,然后再进行编译,如此直到编译通过,编译系统再将源程序编译成机器代码程序,即目标程序。

(3) 连接。使用连接软件,将目标程序、库函数或其他目标程序连接组装起来,形成可直接执行的绝对指令代码程序,即生成可执行文件。其主文件名与源文件名相同,扩展名为“`.exe`”。

(4) 调试。一个程序编译连接产生了可执行程序后,就可以开始程序的调试运行了。通常人们需要输入一些实际数据考验程序执行结果的正确性。如果程序执行中出现问题,或者发现程序的输出结果不正确,就需要设法找到出错的原因,回到前面步骤:修改程序、重新编译、重新连接,再调试,反反复复,直到确信程序正确无误。调试的目的是为了排除错误。一个

程序是不是正确,这是一个很难回答的问题。荷兰计算机科学家 Dijkstra 有一句很有名的话,“调试可以确认一个程序里有错误,但是不能确认其中没有错误。”

(5) 运行。经过了前面的步骤,计算机里已经生成并存在可以直接独立执行的“.exe”程序文件。任何时候,用户都可以在 DOS 环境系统提示符下直接输入“.exe”文件的主文件名,然后回车,程序就会被运行了。

经过上面的分析,读者已经知道,开发一个 C 程序,要经过上述五个步骤。一般来说,这五个过程都是在一个集编辑、编译、连接等为一体的系统软件中完成的。

目前比较流行的软件是 Turbo C,它是一个优秀的集成化开发软件。现在我们就简单介绍一下 Turbo C 软件及 Turbo C 的使用方法。

Turbo C 系统是由一组文件组成的。一般情况下,这些文件存放在硬盘或软盘的一个子目录下,如 Tc、Turbo、Tc20 等,这些子目录名也可以是用户指定的任意目录名。在指定目录的所有文件中,有一个文件是主文件,一般情况下,文件名是 tc.exe。现在我们假设在 C 盘 TC 子目录下存储了 Turbo C 的所有文件,其中主文件名为 tc.exe,则启动 Turbo C 的步骤为:

(1) 进入 DOS 系统(Windows 环境下,也可选择进入“MS-DOS 方式”)。

(2) 假设当前盘是 C 盘,在系统提示符下键入“CD \ TC”命令后回车,将当前工作目录转到 C 盘的 TC 子目录。

(3) 键入“TC”后回车,即可启动 Turbo C 软件系统。

注意:在 Windows 环境下,如果在桌面上创建了 Turbo C 的快捷方式,以后每次进入 Turbo C 只需双击相应图标即可。需要指出的是,Turbo C 是一个 DOS 环境下的系统软件。

执行了上面两个步骤后,屏幕上将显示进入 TC 软件的初始画面,如图 1.7 所示:

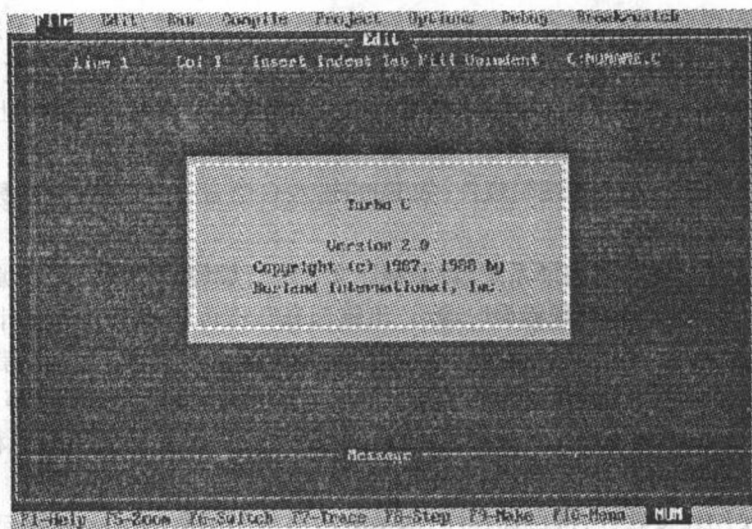


图 1.7

屏幕第一行是菜单行,列出了 Turbo C 提供的所有菜单名:

File Edit Run Compile Project Options Debug Break/watch

用户可以使用键盘上的 F10 键激活菜单栏,结合“→”和“←”键来移动菜单栏上的光标,以便选择不同的命令菜单。也可以使用 Alt + “热键字母”来选择命令菜单,如按下“Alt + F”键