高职高专“十二五”计算机系列规划教材

C语言程序设计

主　编　黄毅斌　郑　麟
副主编　郑锦材　邱　彬　胡越梅

浙江大学出版社

C语言程序设计

主　编　黄毅斌　郑　麟
副主编　郑锦材　邱　彬　胡越梅

ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

内容提要

C 语言是以其灵活、高效、可移植性强等特点发展至今仍保持着强大的生命力，被大多数计算机专业作为第一程序设计基础课程。本书作者根据长期的教学经验，认真编排了教材结构，精心选择教学案例，强调实践与应用，重点讲解程序设计的思想和方法，力求培养学生的程序设计能力，同时也培养学生的独立思考能力，注重启发学生的发散思维，拓宽学生用计算机程序解决问题的思路。教材把握学习程序设计的规律和特点，注重实例教学，从实例中总结出一般规律，运用通俗易懂的文字，由浅入深、由易到难、循序渐进，力求把抽象的概念形象化，把复杂的算法简单化，让学生更加易学易懂。

本教材可以作为各类大专院校的计算机专业或非计算机专业的教学用书，以及各类培训用书，也可作为 C 语言程序自学用书。

图书在版编目（CIP）数据

C 语言程序设计 / 黄毅斌等主编 . —杭州 :浙江大学出版社 , 2012 .1
ISBN 978-7-308-09539-6

Ⅰ . ①C… Ⅱ . ①黄… Ⅲ . ①C 语言—程序设计—高等职业教育—教材 Ⅳ . ①TP312

中国版本图书馆 CIP 数据核字（2012）第 002448 号

C 语言程序设计

主编　黄毅斌　郑　麟

责任编辑　吴昌雷
封面设计　俞亚彤
出版发行　浙江大学出版社
（杭州市天目山路 148 号　邮政编码 310007）
（网址 :http ://www .zjupress .com）
排　　版　杭州中大图文设计有限公司
印　　刷　杭州日报报业集团盛元印务有限公司
开　　本　787mm×1092mm　1/16
印　　张　14 .75
字　　数　383 千
版 印 次　2012 年 1 月第 1 版　2012 年 1 月第 1 次印刷
书　　号　ISBN 978-7-308-09539-6
定　　价　32 .00 元

前　言

目前软件开发语言工具大都使用可视化编程语言工具，面向对象的程序设计语言也逐步替代结构化设计语言，但是C语言仍被许多院校的计算机专业选为入门语言。这是因为C语言具有完备的高级语言特性、丰富灵活的控制和数据结构、简洁高效的语句表达、清晰的程序结构和良好的可移植性等特点，使其发展至今仍保持强大的生命力，在各个领域依然有着广泛的应用。以C语言作为入门基础语言，能学习到较全面的程序设计语言概念、语法、语句，在此基础上再学习其它语言，往往能事半功倍。

但是，把C语言当作入门语言对初学者来说也有不利的一面，它内容丰富、规则繁杂、语法灵活，尤其指针的部分不容易掌握。如果初学者没有掌握好的方法是难以较全面掌握语言设计基础的。C语言的实践性很强，如果都是从理论上或语法规则上进行介绍，学生还是难以真正掌握和理解的。

本教材从应用出发，强调实践与应用，重点讲解程序设计的思想和方法，力求培养学生的程序设计能力，同时也培养学生的独立思考能力，注重启发学生的发散思维，拓宽学生用计算机程序解决问题的思路。教材把握学习程序设计的规律和特点，注重实例教学，从实例中总结出一般规律，运用通俗易懂的文字，由浅入深、由易到难、循序渐进，力求把抽象的概念形象化，把复杂的算法简单化，让学生更加易学易懂。

本教材可以作为各类大专院校的计算机专业或非计算机专业的教学用书，以及各类培训用书，也可作为C语言程序自学用书。教材共分为10章，为使教材保持学科的系统性，我们仍保留了位运算、指针与数组、位段等章节，这部分知识点可根据实际情况选讲。本书由黄毅斌和郑麟主编，其中第1～3章由黄毅斌编写，第4章由胡越梅编写，第5、6、10章由郑麟编写，第7、9章由邱彬编写，第8章由郑锦材编写。

由于编者水平所限，书中难免有错漏之处，我们恳切希望得到读者们的宝贵意见，我们的邮箱地址为xszuoye@163.com，欢迎大家交流。

编　者

2012年1月

目　录

第 1 章

学习前准备

C 语言是计算机专业学生必修的一门基础程序设计语言,学好 C 语言能为计算机专业学生今后进行软件开发培养必要的程序逻辑能力,并为后继课程的深入学习奠定强有力的基础。为更好地进行 C 语言的学习,我们先来了解一些基本概念。

1.1 程序与程序设计语言

1.1.1 什么是程序

计算机本身并不具备独立思考的能力,它的优势在于它能够按照人给定的指令快速地运算,而且对于繁杂的运算,它不会感到枯燥和厌烦。那么人们要让计算机完成某一个计算任务,就必须设计好一系列的指令,然后输入到计算机,启动计算机执行这些指令。这个由人设计好的指令系列就称为程序。也就是说:程序是计算机指令的集合。

在计算机系统中,把计算过程描述为由许多命令按一定顺序组成程序,然后把程序和数据一起输入计算机内存,CPU 自动从内存中取出一条指令,对其进行译码、执行,然后按照顺序自动取出下一条指令继续译码执行,如此反复,这就是计算机自动运行程序的基本模式。

对于程序员来说,要让计算机完成某一项任务或完成某一项工作,可以编制计算机程序让计算机执行。进行程序设计大致可以按以下步骤进行:首先要分析好用户的需求,界定好程序的功能,即弄清楚程序要“做什么”的问题;其次要构建好解决问题的思路,设计好解决问题的算法;第三,用一种程序语言编写程序,实现该算法;第四,进行调试修改,直到程序正确完成所要求功能。

1.1.2 什么是程序设计语言

程序设计语言是用于书写计算机程序的语言,由一组标识记号和一组语法规则组成,是

人与计算机进行交流的工具。从语言的发展历程来看,程序设计语言已经历了四代:机器语言、汇编语言、高级语言和第四代语言。

1.机器语言

计算机能识别的指令是由0和1组成的一串代码,称为机器指令。不同的型号的CPU有不同的指令系统。如某种计算机的指令为1011011000000000,它表示让计算机进行一次加法操作;而指令1011010100000000则表示进行一次减法操作。它们的前八位表示操作码,而后八位表示地址码。早期的程序员设计程序用的就是机器语言,但是使用机器语言设计程序存在下列弊端:

(1)繁杂琐碎的程序设计细节使程序员把大量的精力放在确保程序的正确性、高效性上,而难以进行创造性的劳动。

(2)复杂的程序设计细节使得程序的可靠性差,且开发周期长。

(3)由于用机器语言进行程序设计的思维和表达方式与人们的习惯大相径庭,只有必须经过较长时间职业训练的程序员才能掌握。

(4)程序由二进制代码构成,可读性差,不便于交流与合作。

(5)不同的计算机系统有不同的指令系统,所以针对某台计算机编写的程序难以在另一型号的计算机上运行,所以程序的可移植性和重用性很差。

2.汇编语言

这些弊端造成当时的计算机应用未能迅速得到推广。因此,人们创造了汇编语言,汇编语言用助记符代替操作码,用地址符号或标号代替地址码,比如,用“MOV”表示数据传递,用“ADD”代表加法运算等。这样用符号代替机器语言的二进制码,更利于人类记忆和理解。但是使用汇编语言编写的程序,机器不能直接识别,要由一种程序将汇编语言翻译成机器语言,这种起翻译作用的程序叫汇编程序,汇编程序是系统软件中语言处理系统软件。汇编语言同样十分依赖于机器硬件,移植性不好,但运行效率仍十分高,针对计算机特定硬件而编制的汇编语言程序,能准确发挥计算机硬件的功能和特长,程序精炼而质量高,所以至今仍是一种常用而强有力的软件开发工具。

3.高级语言

高级语言是面向用户的、基本上独立于计算机种类和结构的语言。其最大的优点是:形式上接近于算术语言和自然语言,概念上接近于人们通常使用的概念。高级语言的一个命令可以代替几条、几十条甚至几百条汇编语言的指令。因此,高级语言易学易用,通用性强,应用广泛,用高级语言开发的程序可读性较好,便于维护。同时,由于高级语言并不直接和硬件相关,其编制出来的程序移植性和重用性也要好得多,现代应用程序设计多数都是使用高级语言。

高级语言按其思维模式常被分为面向过程程序设计语言和面向对象程序设计语言。

所谓面向过程就是以要解决的问题为思考的出发点和核心,并使用计算机逻辑描述需要解决的问题和解决的方法。针对这两个核心目标,面向过程的程序设计语言注重高质量的数据结构和算法,研究采用什么样的数据结构来描述问题,以及采用什么样的算法来高效

地解决问题。在20世纪70年代和80年代,大多数流行的高级语言都是面向过程的程序设计语言,如Basic、Fortran、Pascal和C等。

面向对象(Object Oriented)不仅仅是一种程序设计语言的概念,应该说是一种全新的思维方式。面向对象的基本思想就是以一种更接近人类一般思维的方式去看待世界,把世界上的任何一个个体都看成是一个对象,每个对象都有自己的特点,并以自己的方式做事,不同对象之间存在着通讯和交互,以此构成世界的运转。用计算机专业的术语来说,对象的特点就是它们的属性,而能做的事就是他们的方法。常见的面向对象的程序设计语言包括C++和Java等。面向对象方法大大提高了程序的重用性,而且从相当程序上降低了程序的复杂度,使得计算机程序设计能够对付越来越复杂的应用需求。

1.2　C语言的特点

C语言是在20世纪70年代初问世的,1978年由美国电话电报公司(AT&T)贝尔实验室正式发表。C语言是一种结构化语言,它层次清晰,便于按模块化方式组织程序,易于调试和维护。C语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型,便于实现各类复杂的数据结构,它还可以直接访问内存的物理地址,进行位(bit)一级的操作。由于C语言实现了对硬件的编程操作,因此C语言集高级语言和低级语言的功能于一体。既可用于系统软件的开发,也适合于应用软件的开发。此外,C语言还具有效率高,可移植性强等特点。因此广泛地移植到了各种类型的计算机上,从而形成了多种版本的C语言。主要特点如下:

(1)简洁紧凑、灵活方便。C语言一共只有32个关键字,9种控制语句,程序书写形式自由,主要用小写字母表示。它把高级语言的基本结构和语句与低级语言的实用性结合起来。C语言可以像汇编语言一样对位、字节和地址进行操作,这大大增加了程序的灵活性。

(2)运算符丰富。C语言的运算符包含的范围很广泛,共有34种运算符。C语言把括号、赋值、强制类型转换等都作为运算符处理。从而使C语言的运算类型极其丰富,表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

(3)数据结构丰富。C语言的数据类型有:整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。能用来实现各种复杂的数据结构的运算。并引入了指针概念,使程序效率更高。另外C语言具有强大的图形功能,支持多种显示器和驱动器。且计算功能、逻辑判断功能强大。

(4)是面向结构的语言。结构化语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护以及调试。C语言是以函数形式提供给用户的,这些函数可方便的调用,并具有多种循环、条件语句控制程序流向,从而使程序完全结构化。

(5)C语法限制不太严格,程序设计自由度大。虽然C语言也是强类型语言,但它的语法比较灵活,允许程序编写者有较大的自由度。

(6)C语言允许直接访问物理地址,可以直接对硬件进行操作。由于C语言允许直接访

问物理地址,可以直接对硬件进行操作,因此它既具有高级语言的功能,又具有低级语言的许多功能,能够像汇编语言一样对位、字节和地址进行操作,而这三者是计算机最基本的工作单元,可用来写系统软件。

(7)生成目标代码质量高,程序执行效率高。一般只比汇编程序生成的目标代码效率低10～20%。

(8)C 语言适用范围大,可移植性好。C 语言有一个突出的优点就是适合于多种操作系统,如 DOS、UNIX;也适用于多种机型。C 语言具有强大的绘图能力,可移植性好,并具备很强的数据处理能力,因此适于编写系统软件,三维,二维图形和动画,它也是数值计算的高级语言。

1.3 如何学习 C 语言

在讨论如何学习 C 语言之前,我们先来了解为什么要学习 C 语言,明确学习的目的是很重要的,只有明确了学习目的,才会有持久的学习动力。C 语言是很多程序员学习的第一个计算机语言,许多程序员经过了多年的开发经验后都感觉到 C 语言对于一个程序员来说是多么的重要,如果不懂 C 语言,你难以编写底层程序,如果不懂 C 语言,你难以写出优秀高效的程序,为什么说 C 语言如此重要呢?

第一,C 语言语法结构很简洁精练,程序效率高。C 语言本身只提供必要的语言特性,其他复杂一点的功能如文件处理、数学计算等等都以库函数方式提供,便于描述算法,大多数的程序员愿意使用 C 语言去描述算法本身,如果你想在程序设计方面有所建树,那么你应该学习 C 语言。

第二,C 语言的可移植性非常好。实际上 C 语言从早期开始就几乎达到了“一次编写,处处编译”,ANSI 在 1989 年统一了 C 语言标准以后(称之为 C89),只要特定平台上的编译器完整实现了 C89 标准,而且你的代码没有使用某些特殊的扩展,那么代码一定可以编译通过,再实现一下操作系统相关的函数库,C 语言的移植就是很简单的事情。

第三,C 语言能够深入系统底层,甚至直接和硬件打交道。许多的操作系统都是 C 语言写的。例如 Windows,Unix,Linux,Mac,Os/2 等,如果你将来希望能够编写内核程序,那么你必须懂 C 语言,才可以深入到这些操作系统当中去。

第四,很多新型的语言都是衍生自 C 语言,例如 C++,Java,C#,J#,perl,等等。掌握了 C 语言,可以说你就为掌握这些语言打好了基础,经过进一步的学习,你就可以用这些新型的语言去开发了,所以说 C 语言是程序设计的重要基础,许多招聘和考试都必考 C 语言。

明确了学习 C 语言的重要性之后,我们再来讨论一下如何学习 C 语言。很多初学者刚开始都有要学好 C 语言的愿望,但是经过一段时间后,渐渐对 C 语言失去了兴趣,再下来对 C 语言产生了厌倦甚至是恐惧的情绪。是 C 语言难学吗?与其说是 C 语言难学,倒不如说是这些人没有掌握好学习方法。什么才是好的学习方法呢?每个人的学习习惯和理解方式不同,适合自己的学习方法也就不同。在这里提出一点看法,仅供参考。

首先,不要浮躁,静下心来认真学习每一个细节。心态很重要,做一名出色的程序员要

耐得住性子，耐得住寂寞，学习时才能真正掌握每一个语句的每一个细节，当你成为一名程序员时，才能有缜密的逻辑思维，才能让你的程序滴水不漏。不少学习者心浮气躁，语句看了个大概，就自以为懂得差不多了，编写程序遇到些许困难就不愿再思考，急于找答案来看，一看似乎又懂了，下次再编写又不懂了。切记，心浮气躁是编程的大忌。

其次，仔细研读教材。教材的内容是经过精心组织安排的，知识点往往循序渐进，除有细致的讲解之外，还有详细的案例、习题，是初学者必备的工具。有知识点看不懂，请多找几本参考书，再仔细研读，“书读百遍，其意自现”。

第三，多阅读源代码。除书上的例题外，可以多找些源程序代码来阅读。阅读源程序时，对不懂的语句、函数要查阅相关资料及时弄懂，这样一来可以学习或巩固语句语法。另外，要弄清楚作者的解题思路，程序是思维的体现，不是简单的语句叠加，对初学者来说，有两件难事：一是解决问题的思路，也就是算法，二是如何用C语言实现算法。在阅读源代码时，我们总会有这方面的收获。

第四，多编写程序是掌握C语言的不二法门。一定要动手编写，不要以为看得懂程序代码就能写程序。很多失败的学习者就是在这里吃的亏，他们在学习教材时，能读懂所有内容，却懒于对习题进行编写训练，结果真正写起程序却不知如何下手。应该说编写程序不仅仅是一门知识，它还是一种技能，你不训练，就不能掌握，就如你能通过书本知道游泳的方法，但是，如果你不练习，最终你还是不会游泳。

第五，调试修改程序。程序写完后并不一定就正确，要让编译运行，让机器来检验它的正确性。当然，语法正确了还不能代表逻辑也正确了，应该多设计几个区间合理的输入数据，看看能否得到预期的输出结果，通过调试修改，不断完善程序。

第六，其他建议。多参与讨论，同一题目要求，往往有多种算法，参考他人的算法，拓宽自己的思路，尽量让程序的时间、空间效率提高。在学习的后期，设计一款小软件，让它能实际应用在某一方面。分章节的学习，我们学到的是零散的知识，如何综合运用，只有通过实际的开发才能体会，同时，也让我们学以致用，把学到的东西，转换成实实在在的好处。

总之，具体的学习方法因人而异，但是有些步骤是必须经历的，希望读者能尽快找到一种适合自己的学习C语言的方法。

1.4　在Visual C++环境中运行C程序

C程序通常可以在Turbo C或者Win－Tc环境下编译执行，但也可将它放在Visual C++环境中运行。在Visual C++6.0中调试运行C程序的技巧在附录A中有详细说明，读者可以自行参阅，在这里我们只介绍基本的运行步骤。

步骤一：在“开始”菜单打开Microsoft Visual C++6.0。

步骤二：选择“文件”菜单中的“新建”命令，在出现的界面中单击“文件”选项卡，如图1.1所示。

步骤三：选择“C++Source File”选项，在“文件”框中输入文件名，单击“确定”按钮。

步骤四：在程序编辑区中录入程序。

图 1.1 Microsoft Visual C++6.0“新建”窗口

步骤五:选择“编译”菜单中的“编译”命令,或是工具栏中的按钮。

步骤六:根据“编译”窗口中(如图 1.2 所示)的提示信息,修改程序,直至错误数为 0。

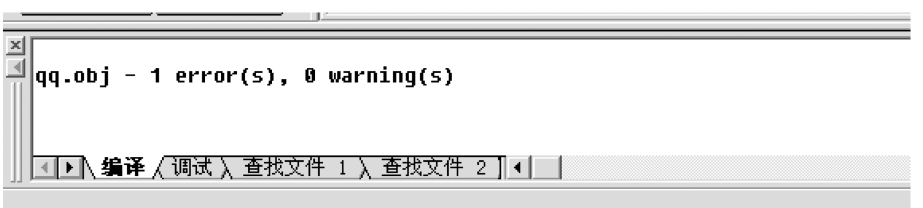

图 1.2 Microsoft Visual C++6.0“编译”窗口

步骤七:选择“编译”菜单中的“执行”命令,或是工具栏上的按钮,执行程序。

步骤八:检查程序运行结果,如果有错,继续修改调试,直到程序结果正确。

1.5 本章小结

本章简单介绍了程序、程序设计语言的概念,介绍了 C 语言的特点和在 Microsoft Visual C++6.0 环境下运行 C 程序的方法,同时还介绍了 C 语言的学习方法,掌握一种合适的学习方法是非常重要的。

第 2 章

C 语言基础

【本章主要内容】

本章的内容比较多,主要介绍 C 程序的基本格式,算术运算符,基本数据类型(包括整型、实型、字符型),赋值运算符和赋值表达式,格式化输入与输出函数的使用。通过本章的学习,学生应能掌握简单的 C 程序写法,并对基本语法结构有所认识。

【本章要求】

(1)掌握简单 C 程序的格式。
(2)掌握算术运算符及其用法。
(3)了解整型常量与变量、实型常量与变量、字符型常量与变量的定义与使用。
(4)掌握赋值表达式的使用方法,包括复合赋值运算符、自增、自减运算符。
(5)掌握格式化输出函数 printf()函数和格式化输入函数 scanf()函数的使用方法。

2.1 简单的 C 程序

首先,请读者先运行下面这个程序,并观察它的运行结果。

【例 2.1】

【程序】

```
#include<stdio.h>
main (    )
{int a;
 float b,z;
 printf("单价:");
```