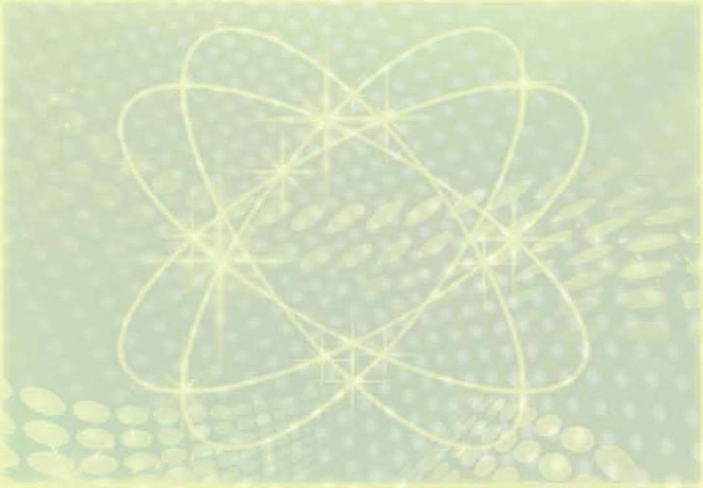
C语言程序设计

王文东　主编



西北大学出版社

21世纪高等教育规划教材
陕西省教育厅重点教材建设项目

C语言程序设计

主　编/王文东
副主编/李竹林

图书在版编目(CIP)数据

C 语言程序设计 / 王文东主编 . —西安:西北大学出版社,2014. 8

ISBN 978-7-5604-3455-1

Ⅰ. ①C… Ⅱ. ①王… Ⅲ. ①C 语言—程序设计—高等学校—教材 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 197403 号

C 语言程序设计

主　　编:王文东
出版发行:西北大学出版社
地　　址:西安市太白北路 229 号
邮　　编:710069
电　　话:029-88303593
经　　销:全国新华书店
印　　装:西安华新彩印有限责任公司
开　　本:787 毫米×1092 毫米　1/16
印　　张:23. 75
字　　数:536 千字
版　　次:2014 年 8 月第 1 版　2014 年 8 月第 1 次印刷
书　　号:ISBN 978-7-5604-3455-1
定　　价:44. 00 元

前　言

C语言自问世以来，由于其功能丰富、表达灵活方便、目标代码效率高、实用性强，特别是可移植性好、既有高级语言的优点又有低级语言的许多（类似汇编语言）特性，使得C语言已成为当今最为流行的程序设计语言之一，也已成为专家公认的理工科各专业学生训练程序设计的基础，是培养计算机应用能力的首选语言。

目前，C语言已不仅是计算机科学与技术专业必不可少的主要课程，而且也已成为广大非计算机专业人员参加全国计算机等级考试、全国计算机应用技术证书考试（NIT）的科目之一。显而易见，掌握C语言程序设计已成为计算机基础素质培养的重要组成部分。

本书是作者在总结多年来本课程教学经验的基础上，听取各方面的意见和建议，进行充分研讨与论证，根据理工科各专业计算机程序设计基础教育要求而编写的。它不仅可作为高等院校理工科各专业程序设计的教学用书，也可作为工程技术人员的自学参考书。

本书由延安大学王文东负责全书编写的总体设计、组织和统稿工作。第1、2章由延安大学西安创新学院李显峰编写，第3、7章由延安大学西安创新学院何志明编写，第4、5、6章和附录部分由延安大学李竹林编写，第8、9、10、11、12章由延安大学王文东编写。

本书主要有以下特点：

1. 按初学者已掌握程序设计的入门语言安排学习内容，由浅入深、循序渐进，重点和难点分散。只要求读者具有计算机文化基础和一定的数学知识，即可学习并掌握本书的内容。

2. 在内容安排上注重了教材的简洁性和条理性，力求做到将烦琐内容通俗化、简单化，尽可能减少读者对概念及内容上的记忆负担。

3. 将“算法设计”基本知识概念渗透到程序设计举例之中，以使读者在具体问题中掌握算法设计的要领与方法，一方面可解决抽象概念的难教难学问题，另一方面可使学生更深刻领会“程序＝算法＋数据结构”的思想。

4. 取材与选例力求典型和少而精，目的在于减少读者的阅读量。编者认为，程序设计语言学习不单是掌握语言本身，训练学习者的程序设计能力则更为重要。因而本书力求将C语言最基本的知识教给读者，突出重点培养读者程序设计的基本思想和技术

方法。

5.图形处理是计算机应用的基本内容。本书专门用一章内容来介绍计算机图形处理C程序设计，教给读者计算机处理的图形基本技术和思想方法，以培养其图形处理的编程能力。

本书编写过程中，参考了国内外有关书籍与教材，吸收了和引用了其中的一些编写经验和材料，在此谨向各书的作者和出版社表示深切感谢。西北大学出版社为本书出版给予了大力支持，在此一并致谢。

本教材受到延安大学教材出版专项基金的资助，出版过程中得到延安大学西安创新学院闫文耀、白海涛、安蕾、张婷曼、樊志华、黄昌军、柴荣军等老师的支持与帮助，对于参与本书讨论和提出宝贵意见的领导、老师及同仁致以谢意。

由于我们水平有限，编写经验不足，加上时间仓促，书中缺点、错误在所难免，欢迎读者批评指正。

编　者

2014年5月

目　　录

第1章 概 述

计算机程序设计语言是计算机应用的语言，用以描述解决问题的方法，供计算机阅读和执行。随着计算机应用领域的扩大，计算机语言也进一步完善，功能不断强大，并朝着易读、易维护和易编程的方向发展。通过本章的学习，希望同学们了解C语言的特点、掌握C程序的组成及书写要求、掌握上机运行简单程序的操作步骤、熟悉简单C程序结构。

1.1 程序与程序设计语言

自1946年世界上出现第一台电子计算机以来，计算机改变了世界，改变了人类的生活。但是计算机并不是天生“自动”工作的，它是由程序控制的。

程序设计在信息技术中占有重要的地位。高效程序的设计基于良好的信息组织和优秀算法，而不是基于“编程小技巧”。实际上，一切问题解决的过程都是有效数据组织的过程，是寻找、设计和实现算法的过程。

1.1.1 算法初识

1. 算法的概念

科学技术的进步，社会生产力的发展，都是由于相关问题不断地得到解决的结果。在当今的信息社会中，许多问题的解决都使用了电子计算机(以下简称计算机)。

使用计算机解题，首先要正确理解题意，接着是寻找或设计解题方法，并对解题方法的正确性进行论证。按照正确的解题方法，可以设计正确的算法，即：规定每一个解题步骤中要求计算机执行的处理，以及各个解题步骤的执行次序。

算法就是计算机求解某一问题而采取的具体的方法和步骤，称为计算机算法，简称算法。计算机算法是以一步接一步的方式来详细描述计算机如何将输入转化为所要求的输出的过程，或者说，算法是对计算机上执行的计算过程的具体描述。

为了更好地理解计算机算法，请看下面几个例子。

【例1-1】计算$1+2+3+\cdots+100$，可采取以下两种算法中的一种。

算法一：可以设两个变量(变量是指其值可以改变的量)，一个变量代表和(s)，一个变量代表加数(i)，用循环算法表示如下：

第一步：$0\Rightarrow s$，$1\Rightarrow i$。

第二步：$s+i\Rightarrow s$。

第三步：$i+1\Rightarrow i$。

第四步：如果$i\leqslant 100$，转第二步；否则，转第五步。

第五步: 输出结果 s,结束。

算法二 : 只有两步。

第一步: $100 \times 101/2 \Rightarrow s$。

第二步: 输出 s,结束。

【例 1-2】判断一个大于等于 3 的正整数是不是素数。

所谓素数是指除了 1 和该数本身之外,不能被其他任何正整数整除的数。例如 23 是素数,因为它不能被 2,3,4,…,21,22 整除。

判断素数的方法很简单,例如判断 $n(n \geqslant 3)$ 是不是素数,只需将 n 作为被除数,将 2 到 $(n-1)$ 各个整数轮流作除数,做除法运算,如果都不能被整除(余数不为 0) ,则 n 是素数。算法描述如下:

第一步: 输入 n 的值。

第二步: i 作除数,$2 \Rightarrow i$。

第三步: n 除以 i,得余数 r。

第四步: 如果 $r=0$,表示 n 能被 i 整除,则打印 n 不是素数,转第七步; 否则执行第五步。

第五步: $i+1 \Rightarrow i$。

第六步: 如果 $i \leqslant n-1$,返回第三步; 否则打印 n 是素数,转第七步。

第七步: 结束。

实际上,除数 i 只需要自增到 $n/2$ 即可。这样,可大大缩短程序执行的时间。

2. 算法的特点

从上面解决问题的过程、步骤中,我们可以看到一个算法应该具有如下的特征:

1) 有穷性。一个算法应包含有限的操作步骤,而不能是无限的。广义地说,“有穷性”一般是指操作步骤或完成操作的时间在合理的范围之内。如果让计算机执行一个历时 1000 年才结束的算法,这虽然是有穷的,但超过了合理的限度,那么这种算法也不能算是有效的算法。

2) 确定性。算法中的每一个步骤都应当有确切的含义,而不应当是含糊的、模棱两可的。算法中的每一个步骤应当不致被解释成不同的含义,而应是十分明确的。也就是说,算法的含义应当是唯一的,而不应当产生“歧义性”。

3) 有 0 个或多个输入。所谓输入是指在执行算法时需要从外界取得必要的信息,其目的是为算法的某些阶段建立初始状态。如果建立初始状态所需的信息已经包含在算法中了,那就不再需要输入。

4) 有 1 个或多个输出。算法的目的是为了求解,没有输出的算法是没有意义的。

5) 有效性。算法中的每一个步骤都应当能有效地执行,并得到确定的结果。

事实上,在日常生活中经常要用算法解决问题,只是人们一般不叫它算法罢了。例如,乐谱是乐队指挥和演奏的算法; 菜谱是厨师做菜的算法; 等等。在漫长的岁月中,人们发现了很多算法。例如,欧几里得(Euclid) 发明的求两个自然数的最大公约数算法。电子计算机的出现,开创了算法研究的新时代。人们可以将算法编写成程序提交给计算机执行,从而迅速获得解题结果。著名计算机科学家克努特(D. E. Knuth) 认为,计算机科学是算法的

学习。

1.1.2 程序和程序设计语言

1. 程序

要让计算机按照人们的愿望工作,必须由人们事先编制好程序,输入到计算机,执行程序才能使计算机产生相应的操作。为了解决某个任务而编写的一组计算机能识别和执行的指令序列,称为计算机程序,简称程序。

虽然算法与计算机程序密切相关,但二者也存在区别:计算机程序是算法的一个实例,是将算法通过某种计算机语言表达出来的具体形式;同一个算法可以用任何一种计算机语言来表达。

【例 1-3】从键盘输入两个整数 x,y,再交换这两个变量的值并将结果进行输出。

解 设有整型变量 x,y 和 temp。

步骤 1:输入两个整数,分别赋予变量 x,y;

步骤 2:x→temp;

步骤 3:x←y;

步骤 4:temp→ y;

步骤 5:输出 x 和 y 的值,结束。

针对上述算法,用 C 语言描述如下:

```
#include <stdio.h>
void main()
{
   int x,y,temp;
   printf("please input x and y:");
   scanf("%d%d",&x,&y);
   temp = x;
   x = y;
   y = temp;
   printf("x = %dy = %d\n\n\n",x,y);
}
```

程序运行结果如下:

```
please input x and y:20  8(其中20和8为用户从键盘输入,回车后则显示结果如下)
x = 8      y = 20
```

2. 程序设计语言

人们常把编写程序的过程称为程序设计。按照某种计算机语言的语法编写的程序,我们把它称为该语言的程序。用 C 语言编写的程序,就称为 C 语言程序。类似的有机器语言程序、汇编语言程序和 basic 语言程序等。

人和计算机怎么沟通呢？计算机并不懂得人类的语言，它只能识别二进制的信息。在计算机产生的初期，人们为了让计算机工作，必须编写出由 0 和 1 所组成的一系列的指令，通过它指挥计算机工作。在研制计算机时，要事先设计好该型号计算机的指令系统，规定好：一条由若干位 0 和 1 组成的指令使计算机产生哪种操作。一个型号机器的指令的集合称为该计算机的机器语言。机器语言是紧密依赖于计算机的硬件的，不同型号的计算机的机器语言不同。用机器语言写程序难学、难记、难写、难修改、难维护，而且在不同计算机之间互不通用，这样计算机的推广应用就会很困难。后来，人们采用了汇编语言，它是用一些特定的“助记符号”代替 0 和 1 来表示指令，如“ADD A，B”就是一条执行加法的指令。用汇编语言编写程序与用机器语言编写程序的步骤相似，它们的指令是一一对应的。由于机器语言和汇编语言都依赖于具体机器，所以被称为“低级语言”。用低级语言编写程序很不直观，且烦琐枯燥，工作量大，无通用性。20 世纪 50 年代出现了用于程序设计的“高级语言”，它比较接近于人们习惯使用的自然语言（英文）和数学语言，如用 read 表示从输入设备“读”数据，用 write 表示向输出设备“写”数据，用 sin 表示求正弦函数。用高级语言编写程序直观易学，易理解，易修改，易维护，易推广，通用性强（不同型号计算机之间通用）。从 1954 年出现第一种高级语言 FORTRAN 以来，全世界先后出现了 2500 种以上的高级语言，每种高级语言都有其特定的用途。其中应用比较广泛的有 100 多种，影响最大的有 FORTRAN 和 ALGOL（适合数值计算）、BASIC（适合初学者的小型会话语言）、COBOL（适合商业管理）、Pascal（适合教学的结构程序设计语言）、PL/1（大型通用语言）、LISP 和 PROLOG（人工智能语言）、C（系统描述语言）、C++（支持面向对象程序设计的大型语言）、Java（适于网络的语言）等。

显然，用高级语言编写的程序，计算机是不能直接识别和执行的（计算机只能直接识别二进制的指令），必须事先把用高级语言编写的程序翻译成机器语言程序，这个“翻译”工作是由称为“编译系统”的软件来实现的。

高级语言的出现被认为是计算机发展史上的惊人成就，它为计算机的推广普及提供了可能。

3. 程序设计过程

程序设计（Programming）是给出解决特定问题程序的过程，是软件构造活动中的重要组成部分。程序设计往往以某种程序设计语言为工具，给出这种语言下的程序。程序设计过程应当包括分析、设计、编码、测试、排错等不同阶段。简单的程序设计一般包括：

1）分析问题，建立数学模型，确定数据结构。对于接受的任务要进行认真的分析，研究所给定的条件，分析最后应达到的目标，找出解决问题的规律，选择解题的方法，完成实际问题。

2）确定算法，并对算法进行描述。即设计出解题的方法和具体步骤。

3）编写程序。将算法翻译成计算机程序设计语言，对源程序进行编辑、编译和链接。

4）调试程序，测试程序的正确性。运行可执行程序，得到运行结果。能得到运行结果并不意味着程序正确，还要对结果进行分析，看它是否合理。不合理要对程序进行调试，即通过上机发现和排除程序中的故障的过程。

5）整理并写出文档资料。许多程序是提供给别人使用的，如同正式的产品应当提供

产品说明书一样，正式提供给用户使用的程序，必须向用户提供程序说明书，内容应包括程序名称、程序功能、运行环境、程序的装入和启动、需要输入的数据以及使用注意事项等。

1.2 C语言的历史与特点

C语言具有功能强大、语句精少、程序简练、灵活方便、效率高、移植性好等特点，使其成为国际上广泛流行的计算机高级语言。它适于作为系统描述语言，既可以用来编写系统软件，也可用来编写应用软件，因此使用非常广泛，不仅在计算机软件专业人员中被广泛使用，而且受到广大计算机应用人员的青睐。与计算机语言有关的各类考试中，也都将C语言作为考试科目之一。由此可见，C语言是初学者学习计算机语言的首选语种。

1.2.1 C语言的历史

C语言是一门通用的、模块化、程序化的编程语言，被广泛应用于操作系统和应用软件的开发。由于其高效和可移植性，适应于不同硬件和软件平台，深受开发员的青睐。

1. C语言出现的历史背景

C语言的起源可以追溯到1960年出现的ALGOL 60。1963年英国剑桥大学推出了CPL(Combined Programming Language)语言，但规模比较大，难以实现。1967年英国剑桥大学的Martin Richards对CPL语言做了简化，推出了BCPL(Basic Combined Programming Language)语言。1970年美国贝尔实验室的Ken Thompson以BCPL语言为基础，又做了进一步简化，设计出了很简单而且很接近硬件的B语言(取BCPL的第一个字母)，并用B语言编写了第一个UNIX操作系统。此时的B语言过于简单，功能有限。1972—1973年，美国贝尔实验室的D. M. Ritchie在B语言的基础上设计出了C语言(取BCPL的第二个字母)。C语言的诞生是和UNIX操作系统的开发密不可分的，原先的UNIX操作系统都是用汇编语言写的，1973年UNIX操作系统的核心用C语言改写，从此以后，C语言成为编写操作系统的主要语言。1977年，出现了首个不依赖于具体机器的C语言编译文本。随着UNIX的日益广泛使用，C语言也迅速得到推广。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、微型计算机上，很快风靡全世界，成为世界上应用最广泛的程序设计高级语言。

图1-1 C语言之父——里奇

2. C语言的标准化发展

以1978年发布的UNIX第7版中的C语言编译程序为基础，Brian W. Kernighan和Dennis M. Ritchie(合称K & R)合著了影响深远的名著《The C Programming Language》(中文译名《C程序设计语言》)，书中介绍的C语言成为后来广泛使用的C语言版本的基础，被称为标准的C，人们称这个版本的C语言为“K & R C”。1988年Brian Kernighan和Dennis Ritchie

修改此书，出版了《The C Programming Language》第 2 版。第 2 版涵盖了 ANSI C 语言标准，从此成为大学计算机教育有关 C 语言的经典教材。

为统一 C 语言版本，1983 年美国国家标准局(American National Standards Institute，简称 ANSI) 成立了一个委员会，根据 C 语言问世以来各种版本对 C 语言的发展和扩充，制定了新的标准草案，来制定 C 语言标准，即 83 ANSI C。83 ANSI C 比原来的标准 C 有了很大的发展。1987 年，ANSI 又公布了新标准——87 ANSI C。1989 年，ANSI 又公布了一个新的 C 语言标准——ANSI X3，159—1989(简称 C89) 。1990 年，国际标准化组织 ISO 接受 C89，作为国际标准 ISO/IEC 9899: 1990，通常简称为 C90。ISO 的 C90 和 ANSI 的 C89 基本上是相同的。1995 年，ISO 对 C90 作了一些修订，称为 C95。1999 年，ISO 又对 C 语言标准进行修订，在基本保留原来的 C 语言特征的基础上，增加了一些面向对象的特征，命名为 ISO/IEC 9899: 1999 标准，通常被称为 C99。但是各个公司对 C99 的支持所表现出来的兴趣不同。因此大多数编译系统都是以 C89 为基础开发的，本书的叙述基本上也以 C89 为基础。

目前，几乎所有的开发工具都支持 ANSI C 标准，是 C 语言用得最广泛的一个标准版本。

3. C 语言的不同版本和其对其他语言的影响

C 语言已经有了 40 年的历史，在高级语言中占据着重要的地位。C 语言有很多版本，如 MC(Microsoft C) 、QC(Quick C) 、TC(Turbo C) 、TC++ (Borland C++) 、VC(Visual C) 、VC++ 、C#等，目前被广泛使用的有 TC、TC++ 、VC++ 和 C#。

很多编程语言都深受 C 语言的影响，比如 C++ (原先是 C 语言的一个扩展) ，C#，Java，PHP，Java Script，Perl，LPC 和 UNIX 的 C Shell。也正因为 C 语言的影响力，掌握 C 语言的人，再学其他编程语言，大多能很快上手，触类旁通。

本书介绍的是传统的 C 语言所涉及的程序都是在 Visual C++ 6.0 的环境中调试运行，因此书中实例的主函数前面都带有 void 标记。

1.2.2 C 程序的特点

一种语言之所以能存在和发展并具有较强的生命力，总是有其优于其他语言的特点。早期的 C 语言主要是用于 UNIX 系统，由于 C 语言的强大功能及其优点逐渐为人们认识，到了 20 世纪 80 年代，C 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了广泛的使用，成为当代最优秀的程序设计语言之一。下面，介绍 C 语言的优秀之处。

1. 语言简洁、紧凑，使用方便、灵活

C 语言是现有程序设计语言中规模最小的语言之一，而小的语言体系往往能设计出较好的程序。C 语言只有 32 个关键字(表 1 - 1) 和 9 种控制语句(表 1 - 2) 。所有的关键字用小写字母表示，压缩了一切不必要的成分，因此 C 语言程序比其他许多高级语言简练，源程序短，输入程序时工作量少。C 语言的书写形式比较自由，表达方法简洁，使用一些简单的方法就可以构造出相当复杂的数据类型和程序结构。

表1-1 32个关键字(由系统定义,不能重作其他定义)

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	unsigned	union	viod
volatile	while			

表1-2 9种控制语句

if...else	while	for
do...while	continue	break
switch	goto	return

实际上,C是一个很小的内核语言,只包括极少的与硬件有关的成分,C语言不直接提供输入和输出语句、有关文件操作的语句和动态内存管理的语句等(这些操作是由编译系统所提供的库函数来实现的),C的编译系统相当简洁。

2. 可移植性好

用C语言编写的程序基本上不用修改就能用于各种型号的计算机和各种操作系统。

用过汇编语言的读者都知道,即使是功能完全相同的一种程序,对于不同的单片机,必须采用不同的汇编语言来编写。这是因为汇编语言完全依赖于单片机硬件。而现代社会中新器件的更新换代速度非常快,也许我们每年都要跟新的单片机打交道。如果每接触一种新的单片机就要学习一次新的汇编语言,那么也许我们将一事无成,因为每学一种新的汇编语言,少则几月,多则上年,那么我们还有多少时间真正用于产品开发呢?

C语言是通过编译来得到可执行代码的,统计资料表明,不同机器上的C语言编译程序80%的代码是公共的,C语言的编译程序便于移植,从而使在一种单片机上使用的C语言程序,可以不加修改或稍加修改即可方便地移植到另一种结构类型的单片机上去,这大大增强了我们使用各种单片机进行产品开发的能力。

3. 表达能力强

C语言具有丰富的数据结构类型,可以根据需要采用整型、实型、字符型、数组类型,C语言的语法规则不太严格,程序设计的自由度比较大,程序的书写格式自由灵活。程序主要用小写字母来编写,而小写字母是比较容易阅读的,这些充分体现了C语言灵活、方便和实用的特点,指针类型、结构类型、联合类型、枚举类型等多种数据类型可实现各种复杂数据结构的运算。

4. 运算符丰富,表达方式灵活

C语言的运算符包含的范围非常广泛,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。利用C语言提供的多种

运算符,可以组成各种类型的表达式,还可采用多种方法来获得表达式的值,从而使用户在程序设计中具有更大的灵活性,从而可以实现在其他高级语言中难以实现的运算。

5. C 语言是结构化语言,可进行结构化程序设计

C 语言是完全模块化和结构化的语言,具有结构化的控制语句(如 if... else 语句、while 语句、do... while 语句、switch 语句和 for 语句) 。具有结构化的控制语句用函数作为程序设计的模块单位,便于实现程序的模块化。C 语言程序中的函数相当于汇编语言中的子程序。C 语言对于输入和输出的处理也是通过函数调用来实现的。各种 C 语言编译器都会提供一个函数库,其中包含有许多标准函数,如各种数学函数、标准输入输出函数等。此外,C 语言还具有自定义函数的功能,用户可以根据自己的需要编制满足某种特殊需要的自定义函数。实际上 C 语言程序就是由许多个函数组成的,一个函数即相当于一个程序模块,因此,C 语言可以很容易地进行结构化程序设计。C 语言是完全模块化和结构化的语言。

6. 可以直接操作计算机硬件

C 语言具备高级语言和低级语言的特征。C 语言具有直接访问单片机物理地址的能力,可以直接访问片内或片外存储器,可以直接对硬件进行操作。C 语言允许进行位(bit) 操作,能按地址访问字节,可以实现汇编语言的大部分功能。因此 C 语言既具有高级语言的功能,又具有低级语言的许多功能; 既可用来编写系统软件,又可用来编写应用软件。有人把 C 语言称为“高级语言中的低级语言”或“中级语言”,意为其兼有高级和低级语言的特点。但一般仍习惯将 C 语言称为高级语言,因为 C 语言程序也要通过编译、链接才能得到可执行的目标程序,这是和其他高级语言相同的。

7. 生成的目标代码质量高

C 语言程序比其他高级语言执行效率高。众所周知,汇编语言程序目标代码的效率是最高的,这就是为什么汇编语言仍是编写计算机系统软件的重要工具的原因。但是统计表明,对于同一个问题,用 C 语言编写的程序生成代码的效率仅比用汇编语言编写的程序低 10% ~20% 。

尽管 C 语言具有很多优点,但也有其自身的缺点,如不能自动检查数组的边界,各种运算符的优先级别太多,某些运算符具有多种用途等。但总的来说,C 语言的优点远远超过了它的缺点。由于 C 语言的这些优点,使其应用面很广。自 20 世纪 90 年代初以来,我国学习和使用 C 语言的人越来越多,C 语言成了学习和使用人数最多的一种计算机语言,熟练掌握 C 语言成为计算机开发人员的一项基本功。

1.3 C 程序简介

1.3.1 C 语言程序的书写格式

为了说明 C 语言源程序结构的特点,先看以下几个程序。这几个程序由简到难,表现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍,但可从这些例子中初步了解到组成一个 C 源程序的基本部分和 C 语言源程序的书写格式。