C 程序设计

（第二版）

夏宝岚　张慕蓉　夏耘　编著

华东理工大学出版社

图书在版编目（CIP）数据

C 程序设计/夏宝岚，张慕蓉，夏耘编著.—上海：
华东理工大学出版社，2003.1
ISBN 7-5628-1349-3

Ⅰ. C… Ⅱ. ①夏… ②张… ③夏… Ⅲ. C 语言-程序设计-教材 Ⅳ. TP000

中国版本图书馆 CIP 数据核字（2002）第 00000 号

C 程序设计（第二版）

夏宝岚 张慕蓉 夏耘 编著

出版	华东理工大学出版社	**开本**	787×1092 1/16
社址	上海市梅陇路 130 号	**印张**	
邮编	200237 **电话** 64250306	**字数**	千字
网址	www.hdlgpress.com.cn	**版次**	2003 年 1 月第 2 版
经销	新华书店上海发行所	**印次**	2003 年 1 月第 1 次
印刷		**印数**	

ISBN 7-5628-1349-2/TP・117 定价： 元

前言

C语言以其小巧、灵活、高效等诸多优点成为当今软件开发的主要编程语言，在国内外得到了广泛的应用。我国的大专院校在多年计算机教学的改革、实践与探索中，大多将“C程序设计”课程作为大学本科生的“第一计算机语言”，作为他们获取程序设计能力的首选课程，作为他们进一步学习数据结构和 C++、Visual C++、Java 等第三代编程语言的先修课程。

编写本书的目的是为广大的初学者提供一本既全面、系统介绍 C 语言又能结合本地区特点，适应上海市普通高校计算机应用能力考试需要的教材。本书以美国国家标准化协会颁布的C语言的最新版本 87 ANSI C 为基础，全书共安排了 13 章内容，前 11 章基本涵盖了C程序设计必须具备的语言、算法、数据结构、程序设计方法等方面的全部知识点。为了提高读者的学习兴趣，充分显示C语言的特色和精华，使读者更全面地掌握C语言及其综合知识运用能力，第 12 章“图形处理初步”，帮助读者掌握计算机绘图的方法与技巧。该章通过形象生动的绘图实例，将前面所学的知识进行综合应用，所举实例都具有较强的趣味性和实用性。本书第二版对初版的内容进行了补充和调整，并增加了第 13 章“算法基础”，使程序设计与算法有机结合，同时还增加了习题参考答案，为学生自学、提高编程技巧与能力提供了方便。第 12、13 章内容教师可根据自身的教学时数酌情安排，亦可作为学生的自学内容。

相对其他常用程序设计语言而言，C语言内容多，用法灵活并涉及函数、指针、结构等比较复杂的概念，常令初学者感到难以掌握。为此，本书在内容选取和结构编排上遵循“循序渐进、由浅入深”的原则，在文字叙述上力求条理清晰、简明扼要、通俗易懂。每一章都先从实际问题切入，引起读者思考，然后才引出概念，阐述语法和语义，最终使问题得以解决。这样做使读者对每一章学习都有明确的目的，有利于理解、消化和吸收所学的知识。另外，本书在编写风格上还特别注重实例化，即每一个知识点都列举了大量例子（全书的每一个例题均在 Turbo C 环境下调试通过），而且举例实用、有针对性，只涉及本章和此前章节所介绍过的内容，以减少读者阅读程序和分析问题时的困难，使他们既能掌握知识要点又不感到抽象、枯燥，并在程序设计的能力方面得到训练。

本书由夏宝岚主编，第 1、6 章由黄俊民编写，第 2、8、11 章由夏耘编写，第 3、4、5

章由张慕蓉编写，第 7、9、10、12 章由夏宝岚编写，第 13 章由周建中编写。

本书在编写过程中得到了华东理工大学计算机系主任邵志清教授、计算中心主任龚正良教授、华东师范大学汪燮华教授、复旦大学江圣扬教授以及黄建华、龚骏、杨睿等同志的悉心指导与帮助，特别是汪燮华教授和江圣扬教授还对部分章节作了审稿和修改，在此一并表示衷心的感谢。

编者

2002

目录

1 C 语言概述

随着计算机技术的不断发展，计算机程序设计语言的研究开发也得到迅猛发展。在数以百计的高级程序语言中，C 语言以其强大的功能和优良的特点，成为国际上公认的最重要的少数几种通用程序设计语言之一。本章将简要介绍 C 语言的由来、发展与特点，并通过简单程序实例说明程序的基本结构和编译运行等知识，使初学者对 C 语言有一个初步的了解。

1.1 概述

1.1.1 程序设计与高级语言

在人类社会的活动中，人与人之间的交流需要通过一种双方都能接受的语言进行。与此相仿，人与计算机对话也需要借助于“语言”。但是人类所使用的自然语言，计算机是不懂的。例如，当我们将 a*b 这一信息输入给计算机后，计算机是不会懂得它的含义的，当然，也就不可能实现 a 和 b 的相乘。

由于计算机自身的物理性质，它只能接受由“0”和“1”组成的信息。因此早期时，人们只能用由“0”和“1”组成的代码来编写计算机程序，要计算机执行某一个操作，就要编写出相应的代码，例如，00111010 00100010 00000000 等等。这种让计算机能够接受并执行的代码，我们称之为机器指令。每个计算机都有自己一套特定的机器指令，一条指令用来完成一个指定的最基本操作，机器指令的集合称为机器语言，用机器语言编写的程序称为机器程序。显然，用机器指令来编写程序是一件十分烦琐、枯燥的工作。要记住每一条指令代码和它的含义是十分困难的，而且程序的直观性差，与人们习惯用的自然语言和数学语言差别太大，由于难学、难记、难写、难查、难改，给计算机的推广使用造成了很大的困难。人们习惯使用的自然语言和数学语言，计算机又不能接受，而计算机能接受的机器语言人们又不易理解。为了解决这一矛盾，人们找出了一种过渡性的语言，它既接近于人们的自然语言和数学表达式，又能使计算机接受，这就是“汇编语言”。

用形象直观的英语单词和数字来代替烦琐的机器指令，就得到了汇编指令。汇编指令的集合就是汇编程序。例如，前面的那条机器指令用 Z80 汇编语言可以记为：LD A，(0022H)，显然这比前面那一串串的“0”和“1”组成的机器指令容易记忆得多。但由于汇编指令仍然与机器挂钩，即不同的机器有不同的汇编指令系统，因此还是一种面向机器的语言。使用汇编语言，同样必须熟悉计算机的指令系统，否则就会感到困难，难以编写程序，因此它很难被广大普通用户所接受。

20 世纪 50 年代末，人们创造了程序设计语言。用程序设计语言编写的程序，描述十分接近于人的自然语言和数学表达式，而且与机器无关。例如，在 C 语言中，求 x 和 y 两个数中的一个大数，可以方便地用条件语句描述为：

```
if(x<y)
   a=x;
else
   a=y;
```

显然，这种写法人们很容易理解，因此程序设计语言比机器语言和汇编语言从本质上前进了一大步，给使用者带来了极大的方便。于是，指令代码语言被称为低级语言，而程序设计语言则被称为高级语言。

事实上，计算机并不能直接接受和执行用高级语言编写的程序，它需要经过“翻译”，即用高级语言编写的程序（称为源程序）先翻译成由机器指令组成的目标程序，然后再让计算机去执行翻译好的机器指令。这个“翻译”工作不是由人担任的，而是由一个充当翻译角色的“编译程序”来担任的。人们事先将这个编译程序放置到计算机内，然后由它对高级语言的源程序进行自动“翻译”。人们可以不必考虑源程序是怎样被翻译成机器指令的，也不必懂得计算机的机器指令，甚至可以不必懂得计算机的工作原理和内部结构，就能应用自如地操作计算机来进行科学计算或数据处理工作。图 1.1 展示了高级语言程序的执行过程。

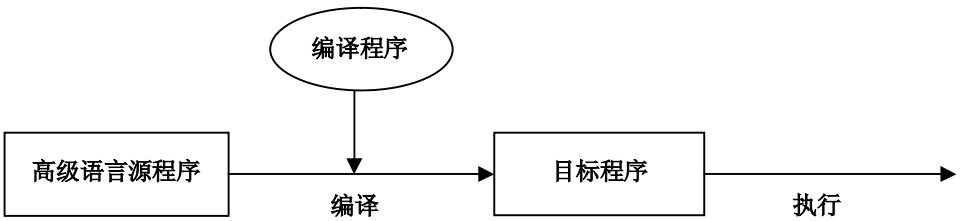

图 1.1　高级语言程序的执行过程

在用高级语言编程的过程中，首先需要调用文本编辑软件帮助你输入和编辑源程序，然后再由编译程序将源程序编译成目标程序，最后由计算机执行目标程序，输出计算结果。为了方便用户编写高级语言程序，软件商开发出了将文本编辑功能和源程序编译功能集成在一个环境中的各种软件，例如 Turbo C 就是一个最常用的 C 编译器，它提供了源程序的编辑和编译集成环境。

目前，世界上的程序设计语言有很多种，其中最常用的是 C、PASCAL、FORTRAN 等，在使用上各有自己的长处。例如：FORTRAN 语言常用于科学计算； C 语言既可用于计算机操作系统、系统实用程序以及需要对硬件进行操作的场合，也可用于科学计算和管理领域；PASCAL 语言是计算机专业较理想的结构化教学语言。每一种高级语言都有相应的编译程序。例如，用 C 语言编写的源程序只能用 C 编译程序才可将它“翻译”成机器指令。

1.1.2　C 语言的产生与发展

C 语言的产生与发展起源于高级语言和 UNIX 操作系统的发展要求。众所周知，早期的系统软件（如操作系统），都使用汇编语言开发。这主要是由于汇编语言能够体现计算机硬件指令的特性，由汇编语言程序形成的代码有较高的质量。但是，汇编语言依赖于计算机硬件和指令系统，程序员编程的工作强度很大，程序的可读性和可移植性都比较差，描述问题的

性能远不如高级语言，因而，人们希望能找到一种具有足够表达能力的高级语言来进行系统软件的设计。1972 年，C 语言被研制出来。

1960 年，主要用于数值计算和某些逻辑处理的一种程序设计语言诞生了，命名为 ALGOL 60(Algorithmic Language 60)。1963 年，在 ALGOL 60 语言的基础上，发展了 CPL 语言(Combined Programming Language)。1969 年，英国剑桥大学的 M.Richards 发表了 BCPL 语言，对 CPL 语言作了改造。1970 年，美国贝尔实验室的 K.Thompson 以 BCPL 语言的设计思想为基础，开发了 B 语言，并在 PDP-11/20 机上实现，同时还用 B 语言编写了 UNIX 系统的应用程序。这些语言都可作为 ALGOL 60 语言语系的分支，是在改造 ALGOL 60 语言时，使其接近于计算机硬件，并逐渐精简而形成的新语言。1972 年，美国贝尔实验室的 D.M.Ritchie 研制成功 C 语言。C 语言保持了 BCPL 和 B 语言的精练以及接近计算机硬件的优点，恢复了那些语言所失去的通用性，具有简洁、灵活、高效和可移植的特性。1973 年，UNIX 系统用 C 语言进行改写，加进了多道程序的功能，从而发生了本质的变化。由于 UNIX 系统建立在 C 语言的基础上，使 C 程序具备了良好的可移植性。从 20 世纪 70 年代中叶起，UNIX 系统在大学中得到了普遍使用，C 语言也随之迅速推广。

除了系统软件外，C 语言还成功地用于数值计算、文字处理、数据库、计算机网络、多媒体等领域。它成功地取代了汇编语言，它所呈现的高级语言的强有力的表达能力和效率，使得 C 语言成为“近十年来在计算机程序设计实践中作出重大贡献的一种语言”，成为微、小、超小、大和巨型机上共同使用的一种语言。

1.1.3 C 语言的特点

作为目前应用最广泛的一种高级程序设计语言，C 语言有如下特点：

1) 语言描述简洁、灵活、高效

C 语言只有 32 个标准的关键字和 9 种控制语句，并且以易读易写的小写字母为基础，使程序书写规整紧凑。

2) 有丰富的数据类型和运算符

C 语言具有四种基本数据类型（char,int,float,double）、多种组合类型（数组和结构）以及复杂的导出类型，并允许使用简单的组合构造复杂的数据类型或干脆由用户自己定义数据类型；C 语言提供了 45 种标准的运算符和多种获取表达式值的方法，并提供了与地址密切相关的指针及其运算符。C 语言将数据类型与运算符结合使用，实现对运算对象的值与流程的控制。

3) 提供了功能齐全的函数库

C 语言标准函数库提供了功能极强的各类函数，其量大面广在常用程序设计语言中是屈指可数的。对诸如串、数组、结构乃至图形的处理，只需调用一下库函数即可实现，为编程者提供了极大的方便。

4) 具有结构化程序设计风格

作为结构化程序设计语言，C 语言提供了编写结构化程序所需的基本结构和控制语句。C 程序由具有独立功能的函数构成，通过函数调用可以实现复杂的程序功能，多种存储属性的数据可共享同一段内存，从而保证了模块化的程序设计风格。另外，对于复杂的源程序文件，可以分割成多个较小的源文件，分别编译和调试，最后组装、链接，得到可执行的目标程序。

5) 具有汇编语言特征

C 语言是从改写汇编语言的 UNIX 系统开始形成的，因此它具有很多高级语言所不具有的特征。例如，它具有字位运算、字段存取等方面的数据结构和运算符，能满足软件工程的需要。寄存器存储变量的应用，提高了程序运行效率，方便了并发程序的开发。C 语言与 UNIX 系统相结合，为用户提供了良好的软件开发环境，在 UNIX 系统的支持下，用户可以得到很多强有力的软件开发工具。如标准 I/O 函数库、UNIX 系统调用、实用程序和应用程序包等，从而使程序开发更能得心应手，事半功倍。

6) 具有良好的通用性和程序的可移植性

除了系统程序的开发广泛采用 C 语言外，它还是众多领域开发计算机应用程序的工具。当前世界上正在运行的计算机系统，基本上都安装 C 语言的编译系统，都以 C 语言作为软件开发的首选语言，这是因为 C 语言的通用性好，与硬件有关的操作（例如数据的输入与输出等）都是通过调用系统提供的库函数来实现的，这就使 C 程序能容易地从一种计算机环境移植到另一种计算机环境中。

当然，C 语言的缺陷还是十分明显的。C 语言缺乏一致公认的标准，表现在语法限制不太严谨，运算符的优先级和结合性比较复杂，不容易记忆。C 语言对数据类型缺乏一致性的检测，对数组、结构等类型的对象的整体运算存在一定的限制。C 语言在程序设计方面灵活性有余，而安全性和可靠性不足。

对于上述问题，计算机科学家和工程技术人员正在不断寻求解决的办法，并不断提出改进的方案。1994 年制定的 ANSI C++标准（草案），就是 C 语言不断发展的一个佐证。

1.2 C 程序简介

1.2.1 简单的 C 程序实例

通过几个简单的 C 程序实例，使我们对 C 程序有一个初步的感性认识。

[例 1.1] 编写程序，在屏幕上打印字符串“We learn to use C language well.”。

程序如下：

```
main( )
{ printf("\n We learn to use C language well. \n");
 }
```

C 程序是由一个或多个函数组成的，但其中必须有，而且只有一个名字为 main 的函数，它是 C 程序的主函数，是真正的执行模块。程序运行时总是从 main 函数中的第一个语句开始执行。本例是个简单程序，只有一个 main 函数，函数体中也只有一句语句，调用 printf 函数在屏幕上印出“We learn to use C language well.”。字样，“\n”的作用是换行。

[例 1.2] 由键盘输入两个整数，计算出它们的商和余数，并在屏幕上输出。

程序如下：

```
main( )
{ int a,b,c,d;
```

```
  scanf("%d%d",&a,&b);
  c=a/b;
  d=a%b;
  printf("\n %d / %d = %d \n %d %% %d=%d\n",a,b,c,,a,b,d);
 }
```

本例也只有一个 main 函数。首先定义变量 a、b、c 和 d 为整型变量（int），接着函数调用 scanf 函数,从键盘输入两个整数，存入变量 a 和 b 的存储单元中，然后求 a 与 b 相除的商和余数，并分别存入变量 c 和 d 的存储单元中，最后执行 printf 语句输出计算结果。若输入的两数为 11 和 4，则屏幕输出：

```
11 / 4 = 2
11 % 4 = 3
```

[例 1.3] 由键盘输入圆的半径和高，计算圆柱和圆锥的体积，并将计算结果显示在屏幕上。

程序如下：

```
/* A program to compute volume of cylinder and cone */
float volm(r,h)  /* function */
float r, h;
{ float x;
  x=3.14159*r*r*h;
  return  x;
 }
main( )
{ float radius, height, volume;
  scanf("%f%f",&radius, &height);
  volume=volm(radius, height)
  printf("volume of cylinder is:\n", volume);
  volume=volume/3;
  printf("volume of cone is:\n", volume);
 }               /* end of main  */
```

本例中包含两个函数，一个是主函数，另一个是名字为 volm 的辅函数。volm 函数有两个参数 r（圆的半径）和 h(柱体高)，其功能是对 r 和 h 计算圆柱体的体积，最后通过 return 语句将结果返回给主调函数。main 函数首先对实型变量 radius 和 height 输入半径和高，然后调用 volm 函数分别计算圆柱体的体积和圆锥体的体积，并在屏幕上输出计算结果。程序中凡是以“/*”开头，以“*/”结束的内容都是注解，不影响程序的执行。

1.2.2 C 程序设计规则

(1) 数据是程序加工的对象，因此 C 程序中使用到的一切数据都必须在使用它们之前对其类型和存储属性加以定义和说明。例如，在[例 1.3]的 main 函数中出现的 float radius, height, volume; 说明语句表示对原始数据半径、高以及将要计算的圆柱体积在使用前先给

出说明。

(2) 函数是 C 程序的基本模块,它应该包括函数名、参数说明表和函数体三个部分。函数体必须以“{”开始，以“}”结束。任何一个 C 程序都必须含有一个名字为 main 的函数，它是主函数，也是程序的真正执行模块，而其他名字的函数也称为辅函数，程序中可以没有，也可以有一个、两个或多个。

(3) 语句是函数的基本单位，在 C 程序中，一行上允许写多个语句，或一句语句分写在多行上，但是语句结束必须加分号。

(4) 为了增加程序的可读性，C 语言允许在程序中插入注释。注释行必须以规定的符号“/*”开头，以符号“*/”结束。注释可插在程序中需要作解释的任意位置上，注释仅起到提示语的作用，它不是程序的组成部分，在程序的编译中不产生编译代码行，因此对程序的执行不产生任何影响。适当加入注释，对于程序的阅读和维护是十分必要的。

(5) C 程序是自由格式书写的程序，因此程序的正文排列与程序功能没有必然联系。但是为了提高程序的可读性，通常书写 C 语言程序时，应该按结构化程序设计原则，采用“缩进”的方式将程序设计成“层次型”，即从外层到内层逐层缩进。例如：

```
for(i=0;i<10;i++)
   for(j=0;j<6;j++)
      if(x[i][j]>0)
         sum1=sum1+x[i][j];
      else
         sum2=sum2+x[i][j];
```

1.2.3 C 程序的编译与运行

C 程序编辑好以后还必须经过编译、链接等步骤处理，使源程序“代码化”，生成可执行程序之后才能在计算机上运行。用于 C 程序编译、链接处理的软件和环境很多，这里介绍几种常用的集成环境。

1) 在 UNIX 操作系统环境下编译与运行 C 程序

首先必须使用 UNIX 系统的编辑程序（例如,行编辑程序 ed 或屏幕编辑程序 vi）,将源程序输入计算机，并存入磁盘。注意，C 源程序的文件名可自己定义，但其后缀应为 “.c”。 然后调用 C 编译程序 cc.exe 对源程序文件进行编译，当编译成功，即源程序被代码化成目标文件后，编译程序会自动调用链接程序 ld.exe，将目标文件和库文件链接起来，形成可执行文件。假设源程序文件名为 hjm01.c，则编译、运行的具体操作方法为：

(1) 输入编译命令行：cc hjm01.c -o hjm01（cc 与 -o 之间给出的是源程序文件名，-o 后面给出的则是将要生成的可执行文件名，这里是 hjm01.exe)。

(2) 输入运行目标程序的命令行：hjm01。假若编译中出现错误，或者运行时不能达到预期结果，都需要重新运行编辑程序对源程序进行修改。源程序一旦被修改，必须经过重新编译方可执行。

2) 在 Turbo C 集成开发环境下编译与运行 C 程序

这是目前最常用的 C 编译环境。若计算机的硬盘上已经安装了 Turbo C（或 Turbo C++）的软件包，那么要启动 Turbo C（或 Turbo C++)，只需键入 tc 并按回车键，即可进入 Turbo