21世纪大学计算机规划教材·工程应用型

C程序设计

王 婧 刘福荣 主编 刘政宇 翟 霞 副主编

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY http://www.phei.com.cn

21 世纪大学计算机规划教材 · 工程应用型

C 程序设计

王 婧 刘福荣 主编

刘政宇 翟 霞 副主编

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书是普通应用型本科院校C程序设计教材。全书共分11章，包括C语言概述，数据类型、运算符与表达式，顺序结构程序设计，选择结构程序设计，循环结构程序设计，函数，数组，预处理命令，指针，结构体与共用体和文件。本书知识描述简洁，例题典型丰富，知识讲授和能力训练并重，为任课老师提供电子课件、习题答案、例题源代码和实验报告等教学资源。

本书既可作为计算机、信息、电子类相关专业教材，也可作为程序设计人员的参考书。

图书在版编目 (CIP) 数据

C程序设计/王婧，刘福荣主编. —北京：电子工业出版社，2009.1
21世纪大学计算机规划教材·工程应用型
ISBN 978-7-121-08102-6

Ⅰ. C… Ⅱ. ①王…②刘… 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（2009）第004274号

策划编辑：童占梅
责任编辑：余 义
印 刷：北京市海淀区四季青印刷厂
装 订：涿州市桃园装订有限公司
出版发行：电子工业出版社
北京市海淀区万寿路173信箱 邮编：100036
开 本：787×1092 1/16 印张：19 字数：486千字
印 次：2009年1月第1次印刷
印 数：5000册 定价：29.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010)88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：(010)88258888。

前　言

C 语言是目前世界上最流行、使用最广泛的高级程序设计语言之一。在对操作系统、系统应用及需要对硬件进行操作的场合中，用 C 语言明显优于其他高级语言，因此许多大型应用软件都是用 C 语言编写的。由于 C 语言功能强、使用灵活、可移植性好、目标程序质量好，从而受到编程者广泛的欢迎。

本书是一本适用于普通高等院校，尤其是侧重于应用能力培养的应用型本科院校的计算机程序设计基础教材，可供计算机专业和非计算机专业的 C 程序设计基础课教学使用，也适用于程序设计的初学者和想更深入了解 C 语言的人使用。和同类书相比，本书注重可读性和可用性，并且难点分散，用人们易于理解的方式清楚地叙述复杂的概念，具有体系合理、逻辑清楚、例题丰富、通俗易懂的特点。在编书过程中，编者遵循了知识讲授和能力训练并重的原则，在讲清基本知识的基础上，注意了例题的选择，大量增加了例题和习题的数量和类型。讲述中力求理论联系实际和循序渐进，注重培养读者分析问题和程序设计的能力，使读者养成良好的程序设计风格和习惯。

程序设计是一门实践性很强的课程，不可能靠听课和看书就能掌握 C 语言程序设计，因此应当十分重视动手编写程序和上机运行程序能力的培养。学习 C 语言时，应该注意把精力放在最基本、最常用的内容上。开始时不要钻牛角尖，在一些细节上不要死抠，因为以后会随着对 C 语言的了解逐步深入和实践经验的逐步丰富，自然而然地掌握其内容，而且有一些细节确实需要通过长期的实践才能真正熟练掌握。本书的宗旨是不仅要使读者掌握 C 语言本身，而且要能够对现实世界中较简单的问题和解决方案用 C 语言进行描述。当然，要能够描述较复杂的问题，还需要学习数据结构、面向对象的软件工程等其他课程。

本书共分 11 章，包括 C 语言概述，数据类型、运算符与表达式，顺序结构程序设计，选择结构程序设计，循环结构程序设计，函数，数组，预处理命令，指针，结构体与共用体，文件。

全书由王婧统稿，张丽杰副教授主审，刘福荣、刘政宇、翟霞、孙海龙、张振蕊参加编写，在此对哈工大华德应用技术学院计算机应用技术系高洪志主任为此书的编写提供的大力支持表示感谢。

为了方便读者学习，本书配有电子课件、习题答案、例题源代码及实验报告的电子版，以供任课老师参考，有需要者可与出版社联系。

由于编者水平有限，书中难免存在疏忽错误之处，诚挚地希望广大读者提出宝贵意见和建议，邮件请发至 hithdjsj@126.com。

编　者

2008 年 11 月

目　录

第1章　C语言概述

1.1　C语言的发展历史及特点

1.1.1　C语言的产生和发展

C语言是国际上广泛流行的计算机高级语言，它既具有高级语言的功能，又具有机器语言的一些特性，既可用来编写系统软件，又可用来编写应用软件。

C语言是在B语言的基础上发展起来的，它的根源可以追溯到ALGOL 60。1960年出现的ALGOL 60是一种面向问题的高级语言，它离硬件比较远，不宜用来编写系统程序。1963年英国的剑桥大学推出了CPL（Combined Programming Language）语言。CPL语言在ALGOL 60的基础上接近硬件一些，但规模比较大，难以实现。1967年英国剑桥大学的Matin 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，在PDP-7上得以实现。1971年又在PDP-11/20上实现了B语言，并编写了UNIX操作系统。但B语言过于简单，功能有限。1972年至1973年，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语言既保持了BCPL和B语言的优点（精练且接近硬件），又克服了它们的缺点（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90%以上的程序用C语言改写，即UNIX第5版。原来的UNIX操作系统是1969年由美国贝尔实验室的K. Thompson和D. M. Ritchie开发成功的，是用汇编语言编写的。

后来，C语言多次做了改进，但主要还是在贝尔实验室内部使用。直到1975年UNIX第6版公布后，C语言的突出优点才引起人们的普遍注意。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》，使用C语言编写的程序移植到其他机器时所需做的工作大大简化，这也推动了UNIX操作系统迅速在各种机器上实现。例如，VAX，AT&T等计算机系统都相继开发了UNIX操作系统。随着UNIX操作系统日益广泛的使用，C语言也迅速得到推广。C语言和UNIX操作系统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语言已先后移植到大型机、中型机、小型机和微型机上，已独立于UNIX和PDP了。现在C语言已风靡全世界，成为世界上应用最广泛的几种计算机语言之一。

以1978年发表的UNIX第7版中的C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》。这本书中介绍的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C。1983年，美国国家标准化协会（ANSI）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，

称为 ANSI C。ANSI C 比原来的标准 C 有了很大的发展。K&R 在 1988 年修改了他们的经典著作《The C Programming Language》，按照 ANSI C 标准重新写了该书。1987 年，ANSI 又公布了新标准——87 ANSI C。1990 年，国际标准化组织 ISO（International Standard Organization）接受 87 ANSI C 为 ISO C 的标准（ISO 9899—1990）。目前流行的 C 编译系统都是以它为基础的。本书的叙述基本上以 ANSI C 为基础。目前广泛流行的各种版本 C 语言编译系统虽然基本部分是相同的，但也有一些不同。在微型机上使用的有 Microsoft C，Turbo C，Quick C，BORLAND C 等，它们的不同版本又略有差异。因此，读者应了解所用的计算机系统所配置的 C 编译系统的特点和规定（可以参阅有关手册）。

1.1.2 C 语言的特点

一种语言之所以能存在和发展，并具有生命力，总是有其不同于（或优于）其他语言的特点。C 语言的主要特点如下：

(1) C 语言简洁、紧凑，使用方便、灵活。ANSI C 标准的 C 语言一共只有 32 个关键字（见附录 B）、9 种控制语句，程序书写形式自由，主要用小写字母表示，压缩了一切不必要的成分。

注意：在 C 语言中，关键字都是小写的。

(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。C 语言把括号、赋值、逗号等都作为运算符处理，从而使 C 语言的运算类型极为丰富，表达式类型多样化。灵活使用各种运算符可以实现其他高级语言难以实现的运算。

(3) 数据结构类型丰富，具有现代化语言的各种数据结构。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（如链表、树、栈等）的运算，尤其是指针类型数据，使用起来比 Pascal 更为灵活、多样。

(4) 具有结构化的控制语句（如 if-else 语句、switch 语句、while 语句、do-while 语句、for 语句）。用函数作为程序的模块单位，便于实现程序的模块化。C 语言是理想的结构化语言，符合现代编程风格的要求。

(5) 语法限制不太严格，程序设计自由度大。例如，对数组下标越界不进行检查，由程序编写者自己保证程序的正确。对变量的类型使用比较灵活，例如，整型数据与字符型数据可以通用。一般的高级语言语法检查比较严格，能检查出几乎所有的语法错误。而 C 语言允许程序编写者有较大的自由度，因此放宽了语法检查。程序员应当仔细检查程序，保证其正确，而不要过分依赖 C 语言编译程序的查错功能。“限制”与“灵活”是一对矛盾。限制严格，就失去灵活性；而强调灵活，就必然放松限制。一个不熟练的编程人员，编写一个正确的 C 程序可能会比编写一个其他高级语言程序难一些。也就是说，对于使用 C 语言的人，要求对程序设计更熟练一些。

(6) C 语言允许直接访问物理地址，能进行位（bit）操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。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，可用来编写系统软件。C 语言的这种双重性，使它既是成功的系统描述语言，又是通用的程序设计语言。有人把它称为“高级语言中的低级语言”或“中级语言”，意为兼有高级和低级语言的特点。

按此观点可将各语言分类如下：

高级	BASIC
	FORTRAN
	COBOL
	Pascal
	Ada
	Modula-2
中级	C
	FORTH
	宏汇编
低级	汇编语言

一般仍习惯地将地 C 语言称为高级语言，因为 C 程序也要通过编译、连接才能得到可执行的目标程序，这和其他高级语言是相同的。

(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) 与汇编语言相比，用 C 语言编写的程序可移植性好。基本上不做修改就能用于各种型号的计算机和各种操作系统。

上面只介绍了 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，我们将结合以后各章的内容进行介绍。C 语言的这些优点，使得 C 语言应用面很广。许多大的软件都用 C 语言编写，这主要是由于 C 语言的可移植性好，硬件控制能力高，表达和运算能力强。许多以前只能用汇编语言处理的问题，现在可以改用 C 语言来处理。

C 语言的以上特点，读者现在也许还不能深刻理解，待学完 C 语言以后再回顾一下，就会有比较深的体会。

下面从应用的角度出发，对 C 语言和其他传统的高级语言进行简单的比较。

从掌握语言的难易程度来看，C 语言比其他语言难一些。BASIC 是初学者入门的较好语言，FORTRAN 也比较易掌握。对于科学计算，多用 FORTRAN 或 PL/I；对于商业和管理等数据处理领域，使用 COBOL 为宜。C 语言虽然也可用于科学计算和管理领域，但并不理想，它的特长不在这里。对于操作系统、系统实用程序及需要对硬件进行操作的场合，用 C 语言明显优越于其他高级语言，有的大型应用软件也用 C 语言编写。从教学角度，由于 Pascal 是世界上第一个结构化语言，而曾被认为是计算机专业比较理想的教学语言。目前，在数据结构等课程中一般用 Pascal 语言举例。但 Pascal 语言难以推广到各实际应用领域，到目前为止基本上只是教学语言。C 语言也是理想的结构化语言，且描述能力强，同样适用于教学。操作系统课程多结合 UNIX 讲解，而 UNIX 与 C 语言密不可分，因此，C 语言已经成为被广泛使用的教学语言。C 语言除了能用于教学外，还有广泛的应用领域，因此更有生命力。Pascal 和其他高级语言的实际目标是通过严格的语法定义和检查来保证程序的正确性，而 C 语言则是强调灵活性，使程序设计人员能有较大的自由度，以适应宽广的应用面。

总之，C 语言对程序员的要求较高。程序员用 C 语言编写程序会感到限制少，灵活性大，功能强，可以编写出任何类型的程序，但较其他高级语言在学习上要困难一些。现在，C 语言已不仅用来编写系统软件，也用来编写应用软件。学习和使用 C 语言的人已越来越多。

尽管C语言有很多优点，但也存在一些缺点和不足。比如它的类型检验和转换比较随便，优先级太多不便记忆。这些都对程序设计者提出了更多要求，也给初学者增加了困难。

1.2 简单的C程序介绍

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，但可从这些例子中了解到组成一个C语言源程序的基本部分和书写格式。

【例1-1】在屏幕上输出一行文本信息“Hello World!”。

```
main()                              /* 主函数 */
{
    printf("Hello World!\n");       /* 在屏幕上输出一行文本信息“Hello World!” */
}
```

程序运行结果如下：

```
Hello World!
```

说明：

(1) main()是主函数的函数名，表示这是一个主函数。每一个C语言源程序都必须有，且只能有一个main()函数。

(2) 花括号“{}”是函数体界定符，位于花括号{……}中的内容为函数体，每个函数都必须用一对花括号将函数体括起来。

(3) 函数体中只有一条输出语句printf("Hello World!\n");，其目的是将引号中的内容“Hello World!”原样输出。

(4) printf()函数是一个由系统定义的C语言的标准输出函数，是系统提供的库函数。

(5) 语句后面有一个分号“;”，这是C语言的语句结束符。

(6) /*……*/是注释语句，用来帮助读者阅读程序，在程序编译运行时/*和*/之间的内容是不起作用的，注释语句可写在程序中的任何位置。

【例1-2】编写一个C程序，计算并输出两数之和。

```
#include "stdio.h"                  /* 编译预处理命令 */
main()                              /* 主函数 */
{
    int a,b,sum;                    /* 定义3个整型变量a,b,sum */
    a=321;                          /* 给变量a赋值 */
    b=654;                          /* 给变量b赋值 */
    sum=a+b;                        /* 计算a + b的值并送到变量sum中保存 */
    printf("The sum is %d\n",sum);  /* 输出文字“The sum is”和变量sum的值 */
}
```

程序运行结果如下：

```
The sum is 975
```

说明：

(1) 在 main( )之前的两行称为预处理命令。预处理命令还有其他几种，# include 是编译预处理命令，也称为文件包含命令，其意义是把尖括号或引号内指定的“stdio.h”文件包含到本程序来，读入到此命令的位置处，成为本程序的一部分。被包含的文件通常是由系统提供的，其扩展名为.h，因此也称为头文件或首部文件。stdio.h 文件中定义了 I/O 库所用到的某些宏和变量。C 语言的头文件中包括了各个标准库函数的函数原型，因此，凡是在程序中调用一个库函数时，都必须包含该函数原型所在的头文件。在使用 C 语言的输入/输出库函数时，一般需要# include 命令的作用及其使用方法，这将在第 8 章编译预处理中详细介绍。

(2) 在 main()函数中首先定义了 3 个整型变量 a，b，sum。

(3) 语句“a = 321; b = 654;”对变量 a, b 进行赋值。

(4) 语句“sum = a + b;”计算 a + b 的值，并将它送给 sum 变量。

(5) printf()函数调用完成 sum 的打印，即将文字“The sum is”和运算结果 975 一起输出。其中，%d 是输入/输出格式符，用来指定输入/输出时的数据类型和格式（详见第 3 章）。“%d”表示“以十进制整数形式输出”，在执行输出时，此位置上以一个十进制整数值代替。

【例 1-3】从键盘输入一个数 x，求 x 的正弦值，然后输出结果。

```
#include<math.h>
#include<stdio.h>   /* include 称为文件包含命令，扩展名为.h 的文件称为头文件 */
main()
{
    double x, s;                    /* 定义两个实数变量，以被后面程序使用 */
    printf("input number:\n");      /* 显示提示信息 */
    scanf("%lf", &x);               /* 从键盘获得一个实数 x */
    s=sin(x);                       /* 求 x 的正弦，并把它赋给变量 s */
    printf("sine of %lf is %lf\n", x, s);   /* 显示程序运算结果 */
}                                   /* main()函数结束 */
```

程序运行结果如下：

```
input number:
45↙
sine of 45.000000 is 0.850904
```

说明：

(1) 在本例中，使用了三个库函数：输入函数 scanf()、正弦函数 sin()、输出函数 printf()。sin( )是数学函数，其头文件为 math.h 文件，因此在程序的主函数前用 include 命令包含了 math.h。scanf()和 printf()是标准输入/输出函数，其头文件为 stdio.h，在主函数前也用 include 命令包含了 stdio.h 文件。

需要说明的是，C 语言规定对 scanf( )和 printf( )这两个函数可以省去对其头文件的包含命令。所以，在本例中也可以删去第 2 行的包含命令#include<stdio.h>。

同样，在例 1-1 和例 1-2 中使用了 printf()函数，也省略了包含命令。

(2) 在例题的主函数体中又分为两部分：一部分为声明部分，另一部分为执行部分。声明是指变量的类型说明。例题 1-1 中未使用任何变量，因此无声明部分。C 语言规定，源程序中所有用到的变量都必须先声明，后使用，否则将会出错。这一点是编译型高级程序设计

语言的一个特点，与解释型的 BASIC 语言是不同的。声明部分是 C 语言源程序结构中很重要的组成部分。本例中使用了两个变量 x, s，用来表示输入的变量和 sin()函数值。由于 sin()函数要求这两个量必须是双精度浮点型，故用类型说明符 double 来声明这两个变量。声明部分后的 4 行为执行部分或称为执行语句部分，用以完成程序的功能。

(3) 执行部分的第 1 行是输出语句，调用 printf()函数在显示器上输出提示字符串，请操作人员输入变量 x 的值。第 2 行为输入语句，scanf( )是一个标准输入函数，其中“&”表示“取地址”，此处 scanf( )函数的作用是接受键盘上输入的数并存入变量 x 中。第 3 行是调用 sin( )函数并把函数值送到变量 s 中。第 5 行是用 printf( )函数输出变量 s 的值，即 x 的正弦值，“%lf”表示“以双精度实数形式输出”。

运行本程序时，首先在显示器屏幕上给出提示字符串“input number:”，这是由执行部分的第 1 行完成的。用户在提示下从键盘上输入某一个数，如 45，按下回车键，接着程序会在屏幕上给出计算结果。

1.2.1 C 语言源程序的结构特点

C 语言程序的一般形式如下：

```
编译预处理命令
全局变量定义
main( )                          /* 主函数 */
{
    变量定义序列
    语句序列
}
sub1( )                          /* 自定义函数 sub1 */
{
    变量定义序列
    语句序列
}
……
subn( )                          /* 自定义函数 subn */
{
    变量定义序列
    语句序列
}
```

C 语言源程序的结构特点：

(1) C 语言程序是由函数构成的。一个 C 语言源程序至少包含一个 main( )函数，也可以包含一个 main( )函数和若干个其他函数。因此，函数是 C 语言程序的基本单位。被调用的函数可以是系统提供的库函数（例如，printf( )和 scanf( )），也可以是用户根据需要自己编制设计的函数。C 语言的函数相当于其他语言中的子程序，用函数来实现特定的功能。程序中的全部工作都是由各个函数分别完成的。编写 C 语言程序就是编写一个个函数。C 语言的库函数十分丰富，ANSI C 建议的标准库函数中包括 100 多个函数，Turbo C 和 MS C 4.0 可提供 300 多个库函数。

C 语言的这种特点使实现程序的模块化很容易。

(2) 一个函数由两部分组成：

① 函数的首部，即函数的第 1 行。包括函数名、函数类型、函数属性、函数参数（形参）名、参数类型。

一个函数名后面必须跟一对圆括号，函数参数可以没有，如 main()。

② 函数体，即函数首部下面的花括号{……}内的部分。如果一个函数内有多个花括号，则最外层的一对为函数体的范围。

函数体一般包括：

- 声明部分：在这部分中定义所用到的变量，如例 1.2 中 main()函数中的“int a,b,sum;”。在第 6 章中还将会看到，在声明部分中要对所调用的函数进行声明。
- 执行部分：由若干个语句组成。

当然，在某些情况下也可以没有声明部分（如例 1-1）。甚至可以既无声明部分，也无执行部分，例如：

```
dump()
{   }
```

它是一个空函数，什么也不干，但这是合法的。

(3) 一个 C 语言源程序有且只能有一个 main()函数，即主函数。总是从 main( )函数开始执行，而不论 main( )函数在整个程序中的位置如何（main( )函数可以放在程序的最前面，也可以放在程序的最后面，或在一些函数之前，在另一些函数之后）。

(4) 每一个声明、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。分号是 C 语言语句的必要组成部分。例如：

```
c=a+b;
```

分号不可少，即使是程序中最后一个语句，也应包含分号。

(5) 标识符，关键字之间必须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。

(6) 源程序中可以有预处理命令（include 命令仅为其中的一种），预处理命令通常放在源文件或源程序的最前面。

(7) C 语言本身没有输入/输出语句。输入和输出的操作是由库函数 scanf()和 printf()等函数来完成的。C 语言对输入/输出实行“函数化”。由于输入/输出操作牵涉到具体的计算机设备，把输入/输出操作放在函数中处理，就可以使 C 语言本身的规模较小，编译程序简单，很容易在各种机器上实现，程序具有可移植性。当然，不同的 C 语言系统需要对函数库中的函数进行不同的处理。C 语言系统除了提供函数库中的标准函数外，还按照硬件的情况提供一些专门的函数。因此，不同的系统所提供的函数个数和功能是有所不同的。

(8) 可以用/*……*/对 C 语言程序中的任何部分进行注释。一个好的、有使用价值的源程序都应当加上必要的注释，以增加程序的可读性。

1.2.2　书写程序时应遵循的规则

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) 一个声明或一个语句占一行。