21世纪高等院校
计算机系列课程教材

C语言程序设计

主 编 高 巍 赵彤洲

北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

21 世纪高等院校计算机系列课程教材

C 语言程序设计

主　编　高　巍　赵彤洲
编　者　姜　楠　张丽秋
　　姬　涛　刘大革
　　王淮中

北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

C语言是一种高效使用的程序设计语言，目前仍在国内外广泛使用，是计算机应用人员应该掌握的一种程序设计工具。

本书深入浅出、循序渐进地介绍了使用标准C语言编程的必备知识，内容新颖、体系合理、通俗易懂、概念清晰，适用于教学。

如无特殊说明，本书程序均在TC 2.0集成环境下测试通过。

本书可作为高等院校各专业学生学习C语言的教材及相关领域的培训教材，也可供各类报考C语言考试的读者参考。

本书同时配有《C语言程序设计习题与上机指导》，以方便教师教学及学生自学。

图书在版编目(CIP)数据

C语言程序设计 / 高巍，赵彤洲主编. —北京：北京理工大学出版社，2005.2

(21世纪高等院校计算机系列课程教材)

ISBN 7-5640-0315-4

Ⅰ.C… Ⅱ.①高…②赵… 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(2005)第012334号

出版发行 / 北京理工大学出版社
社　　址 / 北京市海淀区中关村南大街5号
邮　　编 / 100081
电　　话 / (010)68914775(办公室)　68944990(发行部)
网　　址 / http://www.bitpress.com.cn
电子邮箱 / chiefedit@bitpress.com.cn
经　　销 / 全国各地新华书店
印　　刷 / 北京地质印刷厂
开　　本 / 787毫米×1092毫米　1/16
印　　张 / 15.25
字　　数 / 358千字
版　　次 / 2005年2月第1版　2005年2月第1次印刷
印　　数 / 1～8000册　　责任校对 / 陈玉梅
定　　价 / 22.00元　　责任印制 / 刘京凤

前　言

C语言是一门在国内外都获得广泛应用的计算机语言。自从20世纪70年代C语言诞生以来，凭借其自身的优点，表现出了强大的生命力，至今在许多学科和领域中仍然占据着绝对的统治地位。可以预见到，在竞争激烈的21世纪，C语言仍然是一门有前途的计算机语言。在大学计算机基础教学中，C语言是核心课程，我们一定要学好它，才能在今后的学习和工作中得心应手。

C语言功能丰富，使用灵活，目标程序效率高，可移植性好，具有高级语言的所有优点，同时也具有许多低级语言的特点，因此使用灵活方便。比如，许多单片机都在提供汇编语言编程工具的同时，提供了标准C语言编译器，支持使用标准C语言为单片机编程，极大地提高了工作效率。另一方面，C语言是C++的基础，有志于学习面向对象编程的读者，可以选择使用C++编程，而此时，学习C语言就成为必需的事情了。

程序设计是一门实践性很强的课程，在掌握概念的基础上，必须自己动手编程，并且上机调试运行，这样才能真正掌握程序设计的灵魂。如果仅仅将书读几遍，忽视实践环节，就希望能掌握程序设计的精髓，那么日后不可避免地会在实践中碰壁。希望读者能够重视实践，注重理论联系实际，锻炼动手能力，多思考，多练习。要将学习的目标放在能独立解决问题上，而不是其他方面——诸如考试。当然，这需要付出努力才行。

本书适合做大学计算机专业和非计算机专业的程序设计基础课教材。使用本书可达到以下培养目标：

1. 程序设计入门，领略什么是过程程序设计；

2. 掌握程序设计方法；

3. 把握C语言程序设计规律，熟练使用C语言编程。

全书共分12章。前4章介绍C语言程序设计的基础知识，这是必须具备的基础。以后各章依次介绍了数组、函数、指针、预编译、结构体与共用体、位运算及文件，在这些章节里，最后一章介绍了面向过程程序设计中常涉及的算法和编程技巧，供读者参考。

本书由高巍、赵彤洲主编，姜楠、张丽秋、姬涛、刘大革、王淮中等参编。沈阳化工学院的王志艳副教授对本书的修改提出了许多宝贵的意见和建议。本书的编写也得到了各级领导的关心和支持，在此一并表示深深的感谢！

由于编者水平有限，加之时间仓促，书中难免有不妥之处，我们真诚希望得到广大读者的批评指正。

编　者

2005年1月

目　录

第 1 章　C 语言概述

1.1　C 语言的发展历史

电子计算机自从 1946 年问世以来，其应用已渗透到社会的各个领域，并且硬件和软件都取得了突飞猛进的发展。作为计算机软件的基础，程序设计语言也得到不断的充实和完善。一些功能全面、使用便利的计算机程序语言相继问世。20 世纪 70 年代初期，在种类繁多的程序语言家族中又增添了一名新成员——C 语言。

C 语言是一种编译型程序语言，它的前身是马丁·理查德（Martin Richards）在 20 世纪 60 年代开发的 BCPL 语言。BCPL 语言是计算机软件人员在开发系统软件时，作为记述语言使用的一种程序语言。1970 年，美国贝尔实验室的肯·苏姆普逊（Ken Thompson）在软件开发工作中，继承和发展了 BCPL 语言的特点，进而提出了“B”语言。当时最新型的小型计算机，美国 DEC 公司的 PDP-7 型机中的 UNIX 操作系统就是使用 B 语言记述和开发的。此后，在美国贝尔实验室为更新型的小型机 PDP-11/20 进行 UNIX 操作系统的开发工作中，戴尼斯·M·利奇（Dennis M Ritchie）和布朗·W·卡尼汉（Brian W Kernighan）对 B 语言做了进一步的充实和完善，于 1972 年推出了一种新型的程序语言——C 语言。

C 语言功能强大而灵活，因此很快被传播到贝尔实验室之外，世界各地的程序员都使用它来编写各种程序。然而，在 C 语言出现不久，不同的组织便开始使用自己的 C 语言版本。由于没有统一的标准，使得不同版本的 C 语言之间出现了一些不一致的地方。为了改变这种情况，美国国家标准研究所（ANSI）为 C 语言制定了一套 ANSI 标准，成为现行的 C 语言标准。目前流行的 C 语言编译器绝大多数都遵守这一标准。

1.2　为什么要使用 C 语言

在当前的计算机编程领域中，有大量的高级语言可供选择，如 C、Perl、Java 和 C++。这些都是非常卓越的计算机语言，能够完成大部分编程任务。虽然如此，但基于以下几个原因，很多人认为 C 语言是其中最佳的。

① C 语言功能强大，并且语言简洁、紧凑，使用方便、灵活。C 语言仅有 32 个关键字（见附录Ⅱ），9 种控制语句，程序的书写形式也很自由。C 语言可用于操作系统、字处理器、图形、电子表格等项目，甚至可用于编写其他语言的编译器。具体的实例可参考本书第 12 章

的内容。

② C语言具有结构化的控制语句，用函数作为程序模块以实现程序的模块化。

③ 数据类型丰富。C语言除具有基本数据类型整型(int)、实型(float和double)、字符型(char)外，还有各种构造类型。利用这些数据类型可以实现复杂的数据结构，如堆栈、队列、链表等。

④ C语言表达能力强、语言简练，可以直接访问内存物理地址和硬件寄存器，可以表达直接由硬件实现的针对二进制位(bit)的运算。在语言成分的表示方法上尽可能简洁，I/O操作不是作为C的语法成分而是通过C库函数实现，因而程序简洁，编译程序体积小。

⑤ C语言是可移植的。这意味着为一种计算机系统(如IBM PC)编写的C语言，可以在其他系统中编译并运行，而只需做少量的修改，甚至无需修改。例如，在使用Windows操作系统的计算机上编写的C程序，可以不必修改或做少量修改就可成功移植到使用Linux操作系统的计算机上。C语言的ANSI标准(有关编译器的一组规则)进一步加强了可移植性。

⑥ C语言生成的目标代码质量高，程序执行效率高。代码质量是指C程序经编译后生成的目标程序在运行速度上和存储空间上开销的大小。一般而言，运行速度越高，占用的存储空间越少，则代码质量越高。一般高级语言相对于汇编语言而言其代码质量要低得多，但C语言在代码质量上几乎可以与汇编语言媲美。

由于C语言的上述众多的特点，使它成为一个实用的通用程序设计语言，既可用于编写系统软件，又可编写应用软件，特别适用于编写各种与硬件环境相关的系统软件，不愧为一种强有力的系统程序设计语言。

1.3 C语言程序范例及其结构特点

任何计算机程序语言，都具有特定的语法规定和一定的表现形式。程序的书写格式和程序的构成规则是程序语言表现形式的一个重要方面。按照规定的格式和构成规则书写程序，不仅可以使程序设计人员和使用程序的人容易理解，更重要的是，当把程序输入给计算机时，计算机能够充分“认识”，从而能够正确执行它。

1.3.1 C语言程序范例

一个完整的C语言程序可由若干个函数构成，其中必须有且只能有一个以main命名的主函数。下面将介绍几个简单的C程序例子，我们从中分析C程序的结构。

例1.1 编写一个程序，显示出以下一行文字：I love China.

```
main( )
{
 printf("I love China.\n");
}
```

运行这个程序时，在屏幕上显示一行英文：

I love China.

这是一个仅由 main 函数构成的 C 语言程序。main 是函数名，函数名后面一对圆括号内是写函数参数的，本程序的 main 函数没有参数，故不写，但圆括号不能省略。Main( )后面被大括号{　}括起来的部分称为函数体。一般情况下，函数体由“说明部分”和“执行部分”组成。本例中只有执行部分而无说明部分。执行部分由若干语句组成。“\n”是换行符，即在输出“I love China.”后回车换行。

例 1.2　计算三个整数的和与积。

```
main( )
   {
     int a,b,c;                          /* 定义 a、b、c 为整型变量 */
     printf("Please input 3 numbers:");  /* 输出提示字符串信息 Please input 3 numbers: */
     scanf("%d,%d,%d",&a,&b,&c);         /* 输入变量 a、b、c 的值 */
     b=a+b;                              /* 计算变量 b 的值 */
     c=a*c;                              /* 计算变量 c 的值 */
     printf("a+b+c=%d\n",c+b);
     printf("a*b*c=%d\n",a*c);
   }
```

程序的运行结果如下：

```
Please input 3 number: 3, 4, 5↙
a+b+c=22
a*b*c=45
```

本程序也是一个仅由 main 函数构成的 C 语言程序。其中：

① int a,b,c;　　是变量定义语句，定义三个整型变量，名为 a、b、c。

② printf("Please input 3 numbers:");　　这是一条输出语句，通过调用 printf 库函数在显示屏上输出指定的内容，此例输出"Please input 3 numbers:"字符串。

③ scanf("%d,%d,%d",&a,&b,&c);　　这是一条输入语句，通过调用 scanf 库函数从键盘上输入 a、b、c 的值，&a、&b 和&c 中的“&”含义是“取地址”，此 scanf 函数的作用是将三个数值分别输入到变量 a、b 和 c 的地址所标识的单元中，也就是输入给变量 a、b 和 c。关于 scanf 函数的介绍详见第 2 章内容。

例 1.3　求两个整数中的大数。

```
 int max(int x, int y)
{
  return( x>y ? x : y );
}
main( )
{ int num1,num2;
  scanf("%d, %d", &num1,&num2);
```

```
    printf("max=%d\n", max(num1, num2));
}
```

运行这个程序时，输入 3,9↙　　(输入 3 和 9 给 num1, num2)

在屏幕上显示：

```
max=9
```

本程序是由 main 函数和一个被调用的函数 max 构成的。max()函数的作用是返回 num1 和 num2 中较大的值，通过 return 语句将 num1 和 num2 中较大的数返回给主调函数 main。返回值是通过函数名 max 带回到 main 函数的调用处。main 函数中第 4 行为调用 max 函数，在调用时将实际参数 num1、num2 的值分别传送给 max 函数中的形式参数 x、y。经过执行 max 函数得到一个返回值，然后输出这个值。printf 函数中双引号内“max=%d\n”在输出时，其中“%d”将由 max 的返回值代替，“max=”原样输出。

通过对上述几个例子的分析，我们可以归纳出 C 语言程序的结构特点。

1.3.2 C 语言程序结构特点

① 函数是 C 语言程序的基本单位。main()函数的作用，相当于其他高级语言中的主程序；C 语言中的其他函数，相当于其他高级语言中的子程序。

② C 语言程序总是从 main()函数开始执行。一个 C 语言程序，总是从 main()函数开始执行，而不论其在程序中的位置。当主函数执行完毕时，亦即程序执行完毕。习惯上，将主函数 main()放在最前头。

③ 分号“;”是 C 语句的一部分。

④ C 程序书写格式自由，一行内可写多条语句，且语句中的空格和回车符均可忽略不计。

⑤ 程序的注释部分应括在“/*”和“*/”之间，“/*”和“*/”必须成对出现。“/”和“*”之间不允许留有空格，注释可以用西文，也可以用中文。注释可以出现在程序中的任何位置上。注释部分对程序的运行不起作用。在注释中可以说明变量的含义、程序段的功能，以便帮助人们阅读程序。因此一个好的程序应该有详细的注释。

1.4 C 语言程序的开发过程

C 语言采用编译方式将源程序转换为二进制的目标代码。一个编写完成的 C 程序在成功运行之前，一般经过编辑源代码、编译、链接、运行四个步骤。

(1) 编辑源代码

使用一个文本编辑器（如 Turbo C 2.0 系统自带的编辑器）编辑 C 源代码，创建一个包含源代码的磁盘文件，文件的扩展名为“.c”。源代码是一系列的语句或命令，用于指示计算机要执行的任务。

(2) 编译

使用一个 C 语言编译系统（如 Turbo C 2.0 系统）对 C 源程序进行语法检查和翻译，生成同名的“.obj”目标文件。计算机只能识别二进制形式的机器语言，必须将源代码转换为机器

语言，C 语言程序才能在计算机上运行。这种转换工作是由编译器来完成的。编译器将源代码文件作为输入，如果编译器没有发现任何错误，将生成一个与源代码文件同名，扩展名为“.o”或“.obj”的目标文件，该文件中包含了与源代码语句对应的机器语言指令。编译器创建的机器语言指令被称为目标代码，而包含目标代码的磁盘文件被称为目标文件。如果发现错误，编译器将生成报告提交给程序员，并返回到第 1 步，让程序员在源代码中进行修改。

(3) 链接

将目标文件和库函数等链接在一起生成一个“.exe”的可执行文件。ANSI C 语言定义中包含一个函数库，其中包含预定义的函数的目标代码（已编译过的函数的代码）、预定义的函数包含编写好的 C 代码，由编译器软件包以可直接使用的方式提供。如前面的程序范例中使用的 printf( )和 scanf( )便是库函数，这些库函数，经常执行需要完成的任务，如在屏幕上显示信息及读取磁盘文件中的数据等。如果程序中使用了这样的函数，则必须将编译源代码时生成的目标文件和库函数中的目标代码组合起来，生成最终的可执行程序。这一过程称为链接，是由链接程序完成的。使用链接程序进行链接，如没有发生错误，则生成一个与源代码主文件名相同，扩展名为“.exe”的可执行程序。图 1.1 说明了从源代码到目标代码，再到可执行程序的过程。

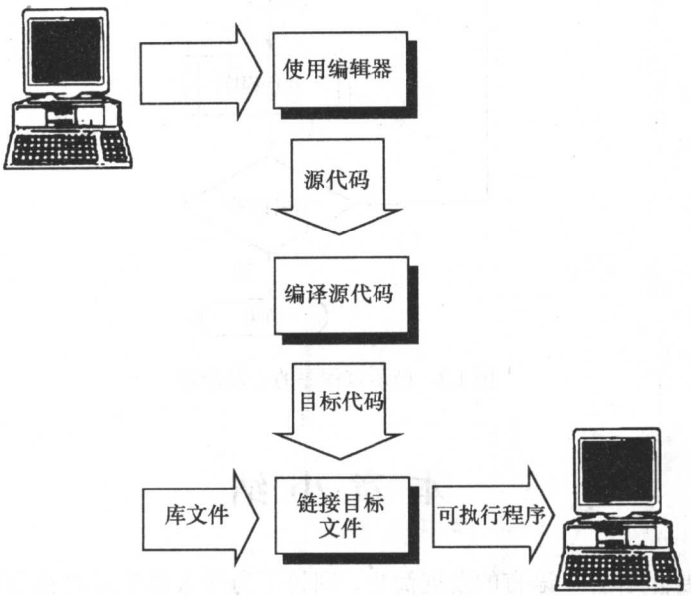

图 1.1　编译器将 C 程序源代码转换为目标代码，然后被链接程序转换为可执行文件

(4) 运行可执行文件

将程序进行编译和链接，创建出可执行文件后，就可以运行了。如果运行程序时得到的结果与期望的不同，则需要回到第 1 步，找出导致问题的原因，并在源代码中进行更正。修改源代码后，需要重新编译和链接程序，创建更正后的可执行文件，如果还有错误，则要不断循环下去，直到程序的执行情况同期望的完全相符。

图 1.2 说明了上述程序开发步骤。

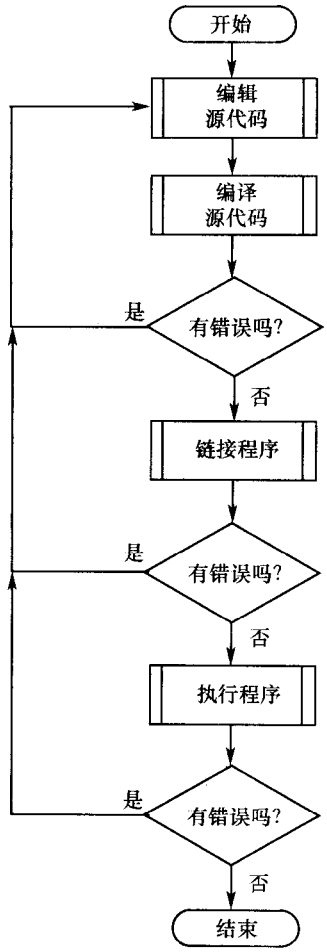

图 1.2 C 语言程序的开发步骤

本章小结

本章我们简要介绍了 C 语言的发展简史，回答了为什么要学习 C 语言这一初学者一直在思考的问题；同时介绍了 C 语言程序范例及其结构特点。要求熟练掌握 C 程序的基本结构，重点理解以下内容：

① 所有的 C 程序都由一个或多个函数组成。

② 在所有函数中，至少包含一个名为 main()的主函数。

③ C 程序总是从主函数 main()开始执行，main()函数可以放在程序的任何位置。

④ 熟练掌握 C 语言程序的开发过程。C 程序的开发包括 4 个步骤：

• 编辑源程序。使用一个文本编辑器（如 Turbo C 2.0 系统自带的编辑器）编辑 C 源程序。

- 编译。使用一个 C 语言编译系统（如 Turbo C 2.0 系统）对 C 源程序进行语法检查和翻译，生成同名的“.obj”目标文件。
- 链接。将目标文件和库函数等链接在一起形成一个扩展名为“.exe”的可执行文件。
- 运行。运行得到的可执行文件。

第2章　数 据 类 型

计算机能处理的信息包括文字、声音、图形、图像等，这些信息都是以一定的数据形式存储的。数据在内存中的存储情况由数据类型决定。任何高级语言都有其所允许的数据类型。在本章中，我们将要学习C语言的数据类型，从而利用合法的数据类型来参与运算，以解决实际问题。在今后的编程中，我们每时每刻都会与它们打交道。C语言有4种基本数据类型和4种复杂数据类型，本章将重点讨论C语言的基本数据类型。

2.1　数据类型的分类

2.1.1　什么是数据类型

C语言提供了十分丰富的数据类型，每种数据类型是对一组变量的性质及作用在它们之上的操作的描述。数据类型描述了某种数据的基础特性，它包括该数据在内存中的存储格式（空间规则）和该类型的数据能进行的算术运算、逻辑运算（运算规则）。C语言所支持的数据类型有基本类型、构造类型、指针类型和空类型，如图2.1所示。

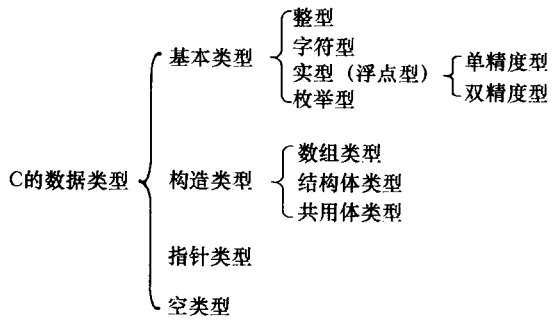

图2.1　C语言的数据类型

由图2.1可见，C语言的数据类型非常丰富，并且我们还可根据实际需求，利用这些基本数据类型构造更为复杂的数据结构。需要记住的是，在C语言中，不论程序中用到哪种数据，都要非常明确地指明其数据类型，并遵循先说明后使用的原则。

2.1.2　C语言的数据类型分类

根据前面的讲述，我们已经知道C语言所支持的数据类型有基本类型、构造类型、指