大学计算机基础教育丛书

C语言程序设计

主编　欧阳春娟　朱　平

同济大学出版社
TONGJI UNIVERSITY PRESS

大学计算机基础教育丛书

C语言程序设计

主　编　欧阳春娟　朱　平

副主编　吴兰英　孙凌宇

编　委（按姓氏笔画顺序为序）

李金忠　刘　欢　刘新明

欧阳春娟　彭宣戈　彭嵩松

孙凌宇　吴兰英　朱　平

同济大学出版社
TONGJI UNIVERSITY PRESS

内容提要

本书结合大量的实例，系统介绍了C语言基础法和C语言程序设计。

本书共分11章和4个附录，主要内容包括：C语言基础知识、C语言程序控制结构、数组、函数、指针、结构体、编译预处理、位运算、文件、C语言图形设计等。

本书内容丰富，结构合理，文字流畅，通俗易懂，每章后配备了丰富的编程习题，习题中溶入了大量近年来全国计算机等级考试(C语言)中出现频率较高的知识。

本书可作为高等院校"C程序设计"课程的教材以及计算机水平考试培训、各类成人继续教育学校开设程序设计课程的教材，也可供计算机爱好者自学使用。

图书在版编目(CIP)数据

C语言程序设计/欧阳春娟，朱平主编. —上海：同济大学出版社，2007.2

ISBN 978-7-5608-3507-5

Ⅰ. C… Ⅱ. ①欧… ②朱… Ⅲ. C语言—程序设计 Ⅳ. TP312

中国版本图书馆CIP数据核字(2007)第010829号

C语言程序设计

欧阳春娟 朱 平 主编

策划编辑 卞玉清 责任编辑 高晓辉 责任校对 谢惠云 封面设计 陈益平

出版发行 同济大学出版社 www.tongjipress.com.cn

(地址：上海市四平路1239号 邮编：200092 电话：021—65985622)

经 销 全国各地新华书店

印 刷 江苏大丰印刷二厂

开 本 787mm×1092mm 1/16

印 张 19.25

印 数 1—3100

字 数 480000

版 次 2007年2月第1版 2007年2月第1次印刷

书 号 ISBN 978-7-5608-3507-5/TP·284

定 价 27.00元

前　言

C语言是目前国内外广泛使用的程序语言之一，是许多大学开设的重要的基础课之一。C语言功能丰富、表达能力强、使用方便灵活、程序执行效率高、可移植性好，既具有高级语言的优点，又具有汇编语言的特点，有较强的系统处理能力。因此，它被广泛应用于系统软件和应用软件的开发。本书从C语言程序设计的基本原理及程序设计的基本思想出发，贯穿“基础—应用—练习”这一主线，紧扣基础，面向应用，循序渐进地引导读者学习程序设计的思想和方法。本书的编写者一直工作在高校教学第一线，主要承担“C语言程序设计”课程的教学任务，有较丰富的教学经验。本书的主要特点是：

(1) 自始至终贯穿两条主线，即C语言的学科主线和实例主线。

(2) 全书通俗易懂，突出基本概念、基本原理与基本应用的介绍与应用。

(3) 本书中的程序代码有详细的注释，以便于读者更好地理解代码，同时，书中所有的程序代码均在TC3.0环境下调试通过。

(4) “图形设计与应用”这一章节的介绍，使读者在很好地完成功能实现的基础上设计出界面更加友好的程序。

(5) 书中每一章后设计的习题不仅尽量多地覆盖各章知识点，而且溶入大量近年来全国计算机等级考试(C语言)中出现频率较高的知识点。相配套的《C语言程序设计上机指导和习题解答》，可供读者进一步提高上机编程能力。

本书可作为高等院校、计算机水平考试培训以及各类成人继续教育学校开设的程序设计课程教材，也可供计算机爱好者自学使用。

本书共分11章，第1章由彭宣戈编写，第2章由刘新明编写，第3章由彭嵩松、李金忠编写，第4章由吴兰英编写，第5章由朱平编写，第6,11章及附录由欧阳春娟编写，第7,8章由孙凌宇编写，第9,10章由刘欢编写，全书由欧阳春娟统稿。

本书参考了国内外同类优秀教材，在此特别加以说明并向作者致以衷心的感谢！井冈山学院教务处为本书的组织工作提供了支持，谨向有关领导和职员致谢！感谢读者选择和使用本书！

限于编写人员的水平有限，加上时间仓促，书中错漏在所难免，恭请读者不吝赐教！

编　者

2006年11月

目　次

第1章　C语言概述

C语言是继BASIC语言、FORTRAN语言、COBOL语言和PASCAL语言之后问世的一种通用计算机程序设计语言。随着计算机的迅速发展和广泛应用，C语言在计算机软件开发中的作用日益重要，越来越显示出它的魅力，已成为世界上广泛流行的、最有发展前途的计算机高级语言。C语言适用于编写各种系统软件，也适用于编写各种应用软件。当前在Internet上最为流行的电子商务软件，大部分程序都是用C语言编写的。随着C语言功能的不断增强，它已经受到越来越多用户的青睐。

1.1　C语言的发展与特点

1.1.1　C语言的发展

C语言是在B语言的基础上发展而来的。

在C语言出现以前，主要使用汇编语言来编写计算机操作系统等系统软件(包括UNIX操作系统在内)。由于汇编语言对计算机硬件依赖性强，使用局限性大，程序的可读性和移植性都比较差，故许多功能难以实现，不适合实际需要。为了打破这种局面，人们不断寻找一种新的语言来替代它。1960年，产生了一种面向问题的高级语言ALGOL60，但它离计算机硬件比较远，不适合用来编写计算机系统程序。1963年，英国剑桥大学推出了CPL(Combined Programming Language)语言。CPL语言离硬件较近，但规模大，难以缩写系统程序。1967年，剑桥大学的Matin Richards对CPL语言进行了优化，推出了BCPL(Basic Combined Programming Language)语言。BCPL语言是CPL语言的改良版，尽管有许多地方作了改进，但还是有很大的局限性，使用不方便。1970年，美国贝尔实验室的Ken Thompson以BCPL语言为基础，对其进一步简化，设计出了很接近硬件的B(取BCPL的第一个字母)语言，并用B语言编写了第一个UNIX操作系统。然而，B语言过于简单，功能有限。1972年左右，贝尔实验室的D. M. Ritchie在B语言的基础上设计出了C(取BCPL的第二个字母)语言，并首先在安装UNIX操作系统的DEC PDP－11计算机上实现，因此说，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。到了1973年，K. Thompson和Dennis M. Ritchie两人合作把UNIX的90%以上内容用C语言进行了改写，即大家熟知的UNIX第五版。多年来，UNIX V系统配备的C语言一直是C语言的公认标准，在Brian W. Kernighan和Dennis M. Ritchie合著的《C Programming Language》中对此也进行了介绍。因此，最初的C语言是为开发UNIX操作系统而研制的，它随着UNIX的出名而闻名。随着微型计算机的普及，出现了较多的C语言系统，其中多数系统接受的源程序都能高度兼容，然而，没有统一的标准，再怎么兼容，总存在一定的差异，这种差异对于计算机应用技术的发展，显然不利。为了克服此不利局面，ANSI(美国国际标准化协会)于1983年成立了专门定义C语言标准委员会，花了6年时间使C语言迈向了标准化。随着C语言的广泛应用又不断推出新的C语言版本，其性能也越来越强。到了1975年，随着UNIX第6版的推出和面向对象程序设计技术的

出现，C 语言的突出优点引起了人们的普遍关注，ANSI C 标准于 1989 年被采用，该标准一般称为 ANSI/ISO Standard C，于是，1989 年定义的 C 标准被称为 C89。到了 1995 年，出现了 C 语言的修订版，其中增加了一些库函数，出现了初步的 C++，在此基础上，C89 成为 C++的子集。此后，C 语言不断发展，在 1999 年又推出了 C99，C99 在基本保留了 C 语言特点的基础上增加了一系列新的特性。随后又几经修改和完善，C 语言从面向过程的编程语言发展到面向对象的程序设计语言。目前，可在微机上运行的 C 语言版本主要有 Microsoft C/C++，Turbo C，Quick C，Visual C/C++等版本。

C 语言是目前国际上广泛流行的一种结构化的程序设计语言，它不仅是开发系统软件的很好的工具，而且也是开发应用软件的很好的程序设计语言。因此，它深受广大程序设计者欢迎。

1.1.2 C 语言的特点

C 语言之所以能被推广并被广泛使用，概括地说主要有如下特点：

(1) 语言简洁、紧凑，使用方便、灵活

C 语言一共有 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，语法控制相对自由，没有严格的格式要求，源程序简练，编辑快捷。

(2) 运算符十分丰富

C 语言一共有 34 种运算符，运算类型极为丰富，表达式类型多样化，能实现各种复杂的运算。

(3) 数据结构丰富

C 语言具有多种数据结构，数据类型有整型、实型、字符型、数组、指针、结构体、共用体等，能实现各种复杂的数据结构(如链表、树、栈等)运算。

(4) C 语言是一种中级语言

C 语言通常被称为中级语言，这并不意味着 C 语言的功能不如高级语言，而是因为它把高级语言的先进思想与汇编语言的控制和灵活性有机地结合起来。作为中级语言，C 语言允许对位、字节和地址这些计算机功能中的基本成分进行操作。

(5) 具有结构化语言的特点

C 语言是便于进行模块化程序设计的语言，C 语言程序由一系列函数组成，这种结构便于把一个大型程序划分为若干相对独立的模块，模块间通过函数调用来实现相互链接。函数允许一个程序中的各个任务被分别定义和编码，从而使程序模块化。一个设计良好的函数可以在各种情况下正常工作，不会对程序的其他部分产生不良影响。因此，对于大型程序的设计来说，函数的设计至关重要。因为函数是 C 语言独立的子程序，是一种构件，程序的所有操作都在其中发生，一个程序员的代码不会意外地影响他人的程序。

C 语言能够把执行某个特殊任务所需要的指令和数据从程序的其余部分分离出去，隐藏起来，实现代码和数据的封装。结构化语言还提供了大量的程序设计功能，直接支持顺序、分支和循环三种典型的基本结构，使程序设计人员便于使用“自顶向下逐步求精”的结构化程序设计技术。

(6) 具有可移植性

C 语言程序具有较好的移植性，在这一点上 C 语言不同于 FORTRAN 等程序设计语言。可移植性指的是：可以把为某种计算机编写的软件运行在另一种机器或操作系统上。如在

DOS下写的程序，能够方便地在 Windows 2000 下运行，这个程序就是一个可移植的程序。C语言不包含依赖硬件的输入输出机制，其输入输出功能是由独立于 C 语言的库函数来实现。这样就使 C 语言程序本身不依赖于硬件系统，因此，便于在不同型号的计算机和各种操作系统间移植。

1.2 C语言程序的基本结构与书写规则

1.2.1 C语言程序的基本结构

为了说明 C 语言程序的结构特点，首先介绍几个简单的 C 程序实例，使读者有一个初步的感性认识。

【例 1.1】

```
main( )
{
 printf("how are you? \n");              /* 直接输出字符串 */
}
```

程序运行结果如下：

```
how are you?
```

【例 1.2】

```
#include "stdio.h"
int max(int x,int y)
{int z;
 if(x>y) z=x;
 else z=y;
 return z;
}
main( )
{int num1,num2;
 printf("Input the first integer number;");
 scanf("%d",&num1);
 printf("Input the second integer numder:");
 scanf("%d",&num2);
 printf("max=%d\n",max(num1,num2));
}
```

程序运行结果如下：

```
Input the first integer number:3 ↙
Input the second integer number:7 ↙
max=7
```

从上面的例子可以看出：

(1) 一个完整的 C 程序至少包含一个 main 函数，也可以包含一个 main 函数(又称主函数)和若干个其他函数。花括号{……}中的内容是 main 的函数体。每个 C 程序的函数都至少有一对{ }。

(2) 一个 C 程序总是从 main 函数开始执行，与 main 函数在程序中的位置无关。当 main 函数执行完毕时，整个程序也就执行完毕。

(3) 任何函数都是由说明部分和执行部分(函数体)组成的，其一般结构如下：

```
函数类型说明符  函数名(函数参数表)                    /* 函数说明部分 */
{说明语句部分;
 执行语句部分;                                      /* 函数体 */
}
```

函数的说明部分主要用于说明该函数的名称、类型、属性、参数名和参数类型等。

例 1.2 中的 max 函数各说明部分如下所示：

```
int      max     (int     x,     int     y)
 ↓        ↓       ↓       ↓       ↓       ↓
函数类型说明符 函数名 (参数类型 参数名, 参数类型 参数名)
```

函数的执行部分(函数体)写在函数说明部分后面的{ }里面，是函数的具体执行部分，主要用于描述该函数的功能和具体操作过程。函数体一般由说明语句和执行语句两部分构成：说明语句放在执行语句的前面，主要对函数体里面出现的数据结构进行定义说明；执行语句由一条或若干条执行具体操作的语句构成。

下面是例 1.2 中 main 函数的示意说明。

```
main( )
{ int num1,num2;                                    说明语句
  printf("Input the first integer number:");
  scanf("%d",&num1);
  printf(Input the second integer number:");        执行语句    函数体
  scanf("%d",&num2);
  printf("max=%d\n",max(num1,num2));
}
```

#include 是预编译程序命令，它把头文件“stdio. h”的内容展开在 #include“stdio. h”所在的行位置处。C 语言系统提供了大量的标准函数供用户使用。如果在程序中使用了 C 语言标准库函数，则必须在程序的开头用 #include 将其相应头文件包含进来。常用的标准函数库及其相应的头文件，读者可到附录三中进行查阅。

(4) C 语言没有输入、输出语句，输入和输出操作由 C 语言系统提供的标准库函数来完成。

1.2.2 C 语言程序的书写规则

C 语言程序的书写比较自由，它没有十分严格的格式要求，但在具体编写 C 语言程序时，还是有一些简单的格式要求：

(1) 程序一般用小写字母书写，个别的标识符可以用大写字母书写。

(2) 所有语句都必须以分号“;”结束，即使是函数的最后一条语句也不例外。

(3) 程序的书写格式相当自由，既允许在一行内写几条语句，也允许将一条语句分写成几行。一般而言，如果程序中的某条语句很长，应将其分写在几行中。

如例 1.2 的 main 函数，可改写成如下的格式：

```
main( )
```

```
{ int num1,num2;
 printf("Input the first integer number:");scanf("%d",&num1);
 printf("Input the second integer number:");scanf("%d",&num2);
 printf("max=%d\n",max(num1,num2));
}
```

(4) 程序中所有的变量必须遵循“先说明,后使用”的原则,必须先说明该变量的类型,然后才能使用该变量。

如例 1.2 中的 main 函数,如果写出如下所示的格式,在编译时就会发生错误。

```
main( )
{ printf("Input the first integer number:");scanf("%d",&num1);
 printf("Input the second integer number:");scanf("%d",&num2);
 printf("max=%d\n",max(num1,num2));
}
```

(5) 程序中的注释,可以用/*……*/对程序中的任何部分作解释。也可以使用“//”注释标识符,其后面的内容就是解释的内容。

1.3 C 语言程序的编译和执行

1.2 节介绍了一些 C 语言编写的程序。需利用计算机来处理问题,必须事先编写出使计算机按照人的意愿工作的应用程序。所谓程序,就是一系列遵循一定规则和思想并能正确完成指定工作的代码(也称为指令序列)。当 C 语言程序的代码编写完成后,就要在机器上运行了。我们知道,C 语言是一种程序设计语言,它很容易被人们看懂和接受;但是,对于计算机来说,却只能接受机器语言。为此必须首先把 C 语言程序翻译成相应的机器语言程序,这个工作叫编译。

任何一个源程序文件必须经过编译、连接后才能执行,生成可执行文件。一般而言,一个程序要经过多次修改、调试才能够顺利完成。图 1-1 是 C 语言程序的编辑调试过程。

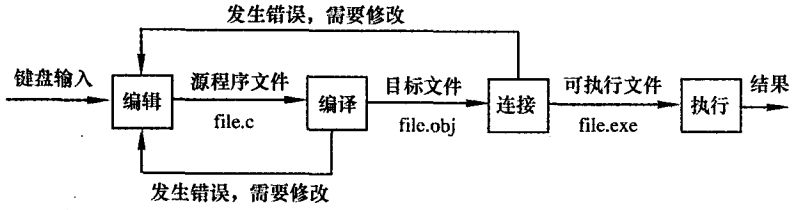


图 1-1

1. 源文件的编辑

为了编译 C 语言程序,首先要用系统提供的编辑器建立一个 C 语言程序的源文件。一个 C 源文件是一个编译单位,它是以文本格式保存的。源文件名可以自由指定,文件的扩展名(或后缀名)为“.c”或“.cpp”。例如,myfile.c 和 file.cpp。

一个比较大的 C 语言程序往往可划分为若干模块,每个模块由不同的开发者或开发小组负责编写,对每个模块可建立一个源文件。因此,一个大的 C 程序可包含多个源文件,这些源文件都要进行编辑。关于源程序的编辑环境及其应用,请参见附录二与附录三。

2. 编译

源文件建立好后，经检查无误后就可进行编译。编译是由系统提供的编译器完成，编译命令随系统的不同而不同，具体操作时可参考相应的系统手册。例如，对于 Tubo C，一般通过 Turbo C 的编辑环境界面中 Compile 菜单中的 Compile 命令进行编译，编译器在编译时对源文件进行语法和语义检查，并给出所发现的错误。用户可根据错误情况，使用编辑器进行修改，然后对修改后的源文件再度编译。用户也可以在 Compile 菜单中选 Make 命令进行编译，它能直接生成可执行的文件；此时如果系统发现用户的源程序有语法错误，就发出错误的参考信息，提示用户进行错误代码的修改，然后用户再重新进行编译。值得注意的是，C 语言的编译器不对数组的越界进行检查，因此，用户一定要注意数组的越界问题，有关数组的详细介绍，请参阅本书的后续章节。

3. 连接

若在上述步骤中，用户选择 Compile 命令进行编译，编译所生成的目标文件(*.obj)是相对独立的模块，但还不能直接执行，用户还必须用连接编辑器把它和其他目标文件以及系统所提供的库函数进行连接装配，生成可执行文件后才能执行。可执行文件的名字可自由指定，默认的执行文件的名字与源文件的名字一致，可执行文件的扩展名为“.exe”。

4. 执行

可执行文件生成后，就可执行它了。若执行的结果达到预想的结果，则说明程序编写正确；否则，就需进一步检查修改源程序，重复上述步骤，直至得到正确的运行结果为止。

习　题

一、填空题

1. C 语言程序的基本单位是_____。
2. 一个 C 源程序中至少应包括一个_____函数。
3. C 语言中，输入操作是由库函数_____完成的，输出操作是由库函数_____完成的。
4. C 源程序扩展名是_____，菜单中 RUN 命令运行 C 程序的快捷键是_____。
5. 一个名为 P1 的 C 语言程序，编译、连接、运行成功后，会生成_____个有关 P1 的文件，扩展名分别是_____、_____、_____、_____。

二、改错题

请把下面两题的 C 语言程序的语法错误找出来。

1.
```
main( )
  { int a=5; b=8;c;
    c=a * b
    print ("c=%d\n",c);
```

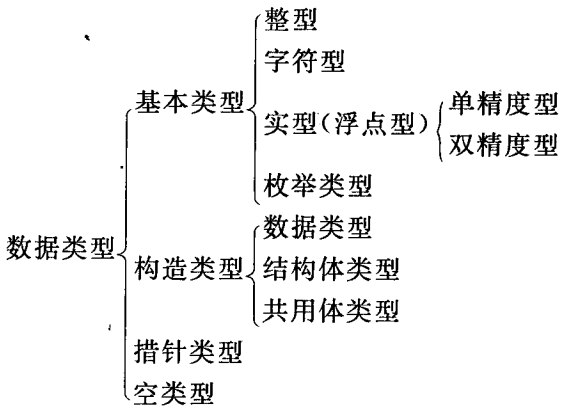
2.
```
#include sdio.h;
main( )
float r;s;
r=5.0
s=3.14169r * r;
  print "s=%f\n",s)
```

第 2 章　C 语言程序设计基础

一个程序应包括两方面内容：数据的描述（即数据结构）和操作步骤（即算法）。数据结构是指数据对象及其相互关系和构造方法，程序中的数据结构描述了程序中的数据间的组织形式和机构关系。数据结构与算法密不可分，一个良好的数据结构能使算法简单化。作为程序设计人员，必须认真考虑和设计数据结构和算法。著名计算机科学家沃斯（Nikiklaus Wirth）提出一个公式

数据结构＋算法＝程序

C 语言提供的数据结构是以数据类型形式出现的。C 的数据类型如下：



C 语言中数据有常量和变量之分，它们分别属于以上这些类型。

2.1　常量和变量

2.1.1　常量和符号常量

在程序运行的过程中，其值不能被改变的量称为常量。常量区分为不同的类型，如 9，0，－2 为整型常量，5.1，－8.63 为实型常量，'b'，'k' 为字符常量。常量的类型一般从其字面形式即可判别。也可以用一个标识符代表一个常量，如：

【例 2.1】

```
# define VALUE 50
main( )
{
  int num,sum;
  num=20;
  sum=num * VALUE;
  printf("sum=%d",sum);
}
```