主编 吴平 李林

C语言程序设计教程

科学技术文献出版社

C语言程序设计教程

C语言程序设计教程

主编 吴 平 李 林

编委 陆诗雷 陈 薇

科学技术文献出版社

(京)新登字 130 号

责任编辑/ 沈　扬
策划编辑/ 王大庆
责任校对/ 李正德
责任出版/ 永　京
封面设计/ 宋雪梅

内 容 简 介

本书以 Turbo C V2.0 为背景，配合内容要点、学习难点及常见错误分析，深入浅出地介绍了C语言的基本语法和基本数据结构，突出体现了算法思想和程序实现方法。本书在长期教学实践基础上编写而成，内含大量例题和习题，内容设置循序渐进，讲授方法新颖独特，既是初学者的入门教材，又是程序设计、开发人员的实用手册，同时还可作为计算机等级考试人员的参考书。

图书在版编目(CIP)数据

C语言程序设计教程/吴平，李林主编 .-北京：科学技术文献出版社，1998.9
ISBN 7-5023-3144-1

Ⅰ.C…　Ⅱ.①吴…　②李…　Ⅲ.C语言-程序设计-教材　Ⅳ.TP312

中国版本图书馆 CIP 数据核字(98)第 23601 号

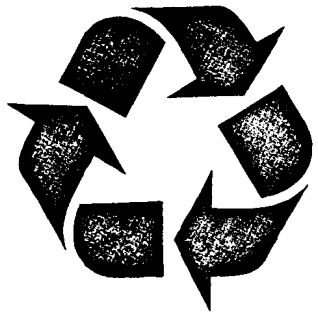
出　版　者/ 科学技术文献出版社
地　　址/ 北京市复兴路 15 号(中央电视台西侧)/100038
发　行　者/ 新华书店北京发行所
印　刷　者/ 北京国马印刷厂
版(印)次/ 1998 年 9 月第 1 版，1998 年 9 月第 1 次印刷
开　　本/ 787×1092　16 开
字　　数/ 428 千
印　　张/ 16.75
印　　数/ 1—4000 册
定　　价/ 24.00 元

目录

第一部分　程序控制结构

第二部分　数据结构

第一部分



程序控制结构

第一章

C语言程序设计

【导读提要】本章从程序设计的角度介绍了计算机语言的发展过程和各类语言的特点，其中重点介绍了计算机语言的工作环境、C语言的发展、C语言的基本概念及特点。

本章要求熟练掌握的内容：

- 计算机操作系统的基本概念
- 使用高级语言编程的过程
- C语言的特点和基本语法
- 有关程序设计算法的基础知识

1.1　程序设计语言及其工作环境

1.1.1　程序设计语言的发展

程序设计语言伴随着计算机技术的发展而不断升级换代——从机器语言到高级语言，从面向过程的语言到面向对象的语言。一般计算机语言可以分为两类：

- 低级语言：包括机器语言和汇编语言。
- 高级语言：标准化的、描述方式接近自然语言的计算机语言（C语言就是一种高级语言）。

在开始学习C语言之前，首先介绍一下各类计算机语言的基本特点。

1. 机器语言

机器语言是计算机的指令系统，它是CPU（计算机中央处理器）可以识别的由01码构成的指令。例如，可以用10000000表示加法操作，

10010000 表示减法操作。显然，机器语言指令的功能含义不直观，记忆和理解都比较困难。因此，使用机器语言编程效率比较低。再者，由于不同类型计算机的指令系统不同，机器语言对于计算机的硬件环境就具有了依赖性。

2. 汇编语言

汇编语言是用助记符描述的计算机指令系统。如用 ADD A,B 表示 A+B=>A 的操作，用 SUB A,B 表示 A-B=>B 的操作。一般情况下，一种机器的机器语言与汇编语言的指令是一一对应的，从形式上来看汇编语言比机器语言容易理解和记忆，所以使用汇编语言编程比使用 01 码的机器语言要容易一些。但是，汇编语言编写的程序必须翻译成机器语言才能使程序被 CPU 理解和执行。这个翻译转换过程就称为“汇编”。汇编后得到的机器语言程序称为目标程序(object program)，汇编以前的程序称为源程序(source program)。汇编语言源程序的翻译可以通过命令由汇编系统自动进行。

3. 面向过程的高级语言

汇编语言和机器语言是面向机器的语言，随机器的类型不同而不同(一般情况下，不同 CPU 的指令系统是不同的)。高级语言是具有国际标准的、描述形式接近自然语言的计算机语言。常用的计算机高级语言有 BASIC、FORTRAN、PASCAL 和 C。不同的高级语言具有不同的特点，因而处理起不同领域中的问题各有所长。这些计算机高级语言，不再直接面向某种具体的机器，语言指令的描述方式比较接近人们生活中使用的自然语言，指令的格式由于有统一的国际标准而与计算机的类型基本无关，就是说掌握了一种计算机高级语言的程序设计方法，就基本具备了在不同类型计算机上进行同一语言程序设计的能力。高级语言的指令形式接近自然语言，所以指令便于记忆，程序易于理解。由于使用上述语言编制的程序是按照算法处理过程在程序中详细描述操作的所有步骤，所以具有这种特征的计算机高级语言又称为“面向过程的计算机语言 ”。使用面向过程的计算机高级语言编程时，编程者的主要精力应放在算法过程的设计和描述上。

下面是一个计算圆柱体体积的 C 语言程序段，程序详细地描述了计算机处理过程中的每一步操作。

```
main()                          /*C语言程序的主函数*/
{ float v, r,h;                 /* 定义3个实型变量v、r、h */
  r=2; h=2;                     /* 给圆柱体的半径r和柱高h赋值 */
  v=3.14159*h*r*r;              /* 计算圆柱体体积v的值 */
  printf("Volume=%fn", v);      /* 输出体积v的值 (printf是C语言的输出函数) */
}                               /* 程序结束 */
```

在 C 语言程序中，“/*”、“*/”以及放在它们中间的内容是程序注释部分。程序注释部分对程序本身不起作用。

4. 非过程化的高级语言

在使用面向过程的语言编程时，首先要解决的问题是题目要求“做什么”，然后利用计算机语言说明“怎么做”，“怎么做”的过程描述就是一个“程序”。在计算机高级语言中还有一种非过程化的语言，在使用它编程时，无须告诉机器“怎么做”，用户只要告诉计算机“做什么”，机器便会完成求解过程。

非过程化的高级语言使用更为简单。但是，程序的运行效率和语言的灵活性不如过程化的语言高。因此，学习用过程化的高级语言进行程序设计是计算机应用人员的一项基本训练。

1.1.2　C语言发展简况

C语言产生于70年代，最初版本的C语言是1972年由美国贝尔实验室的丹尼斯·里奇设计，在PDP-11机器上实现的一种小型语言。当时的C是为UNIX操作系统设计的系统语言，UNIX操作系统的研制者肯·汤姆逊先使用汇编语言和一种命名为B的语言在1970年发表了UNIX操作系统的最初版本。后来，为了克服B语言的一些局限性，而产生了C语言。

1978年在贝尔实验室的布莱恩 W·科宁汉和丹尼斯·里奇合著的《C程序语言》（《The C Programming Language》）中详细描述了C语言的基本定义。从此，C语言开始作为一种比较通用的计算机语言得到普及。

到目前为止，C语言已经成为最流行的计算机语言之一。C语言能如此迅速地被程序设计者接受，并越来越受到青睐，是由于它具有大多数计算机语言不可比拟的优势。C语言的优越性主要体现在下述几个方面：

(1) 良好的可移植性

由于C语言系统的设计并不过分依赖于计算机的硬件环境，所以当一个程序需要在不同的机器环境中运行时，基本不需要改动或者只需要进行很少的改动就可以使之适合于新的工作环境。目前，不仅广泛使用的大型、中型和小型计算机都支持C语言，而且同一机型的不同操作系统也都支持C语言，因此C语言具备广泛应用的优势。

(2)适合于开发系统软件

C语言不仅具有一般计算机高级语言的基本功能，还具有一些过去只能通过计算机低级语言才能实现的功能。例如，对计算机内存的地址操作、位操作、寄存器操作和对操作系统的系统功能调用等等。这些能力使得C语言特别适合于进行系统软件的设计。因此使用者们称之为“高级语言中的低级语言”，亦或“中级语言”。目前，一些广泛流传的著名的软件系统如UNIX和数据库管理系统就是用C语言编写的。

(3)C语言是便于模块化软件设计的程序语言。C语言程序的函数结构，有利于把大型软件模块化，对软件工程技术方法提供了强有利的支持。

(4)C语言具有种类丰富的运算符和数据结构，具有编译预处理功能，这些功能有效地提高了软件开发的工作效率。

C语言的优点使其受到程序员和系统程序员广泛的欢迎。下面让我们从C语言基本语法开始，一步步迈进程序设计的神秘殿堂，去领略无穷的奥秘与惊喜。

1.1.3　程序设计语言的支持环境——操作系统

在我们所生活的信息时代，大家对于“系统”这个词并不陌生，但是我们是否知道它的确切含义？能够简单明确地回答这个问题，对于学习本书所要介绍的内容，是十分重要的前提。

1. 系统

同类事物按一定的关系组成的整体称为系统。

2. 计算机系统

能够完成计算机任务的全部部件。它们可分为两部分。

(1) 硬件：计算机的机器设备，如机器的电路和器件、显示器、打印机等等。

(2) 软件:指挥机器进行工作的程序。

软件和硬件的概念已经越来越多地出现在人们的生活中。例如,一个音响系统,其中 Hi-Fi 录音机、CD 驱动器、录音带和 CD 盘是硬件,在磁带、唱盘和光盘上存储的内容则是软件。通过我们熟悉的音响设备,很容易地想象到仅仅有硬件——设备是不能正常工作的。必须有音乐软件,音响才能发出声音。计算机也是同样,具备了电路和器件还不够,必须有能够对计算发出正确指令的软件,才能使计算机正常运转,完成指定的任务。因此,这就产生了对计算机管理软件的需求。

3. 计算机操作系统

接触过计算机的人,都听说过操作系统,它到底是什么呢? 简单地说,操作系统就是管理计算机硬件设备进行工作的一组程序。如上所述,要使计算机完成任何一项工作,都需要对其发出指令。当我们面对一台计算机,要使它能够在机器的屏幕上显示出字符,能够接受我们从键盘上输入的数据,……如果对机器的组织结构和工作原理不了解,是无法实现的。但是如果要求想利用计算机进行工作的所有人都必须先了解计算机的原理与结构,那么,计算机在今天不可能得到如此广泛的应用。正是这种使用的需求,促进了计算机操作系统的发展。计算机的设计者或制造者向用户提供一组管理计算机进行工作的程序,用户可以通过指定的命令调用这些程序,实现自己所需要的功能。这组管理计算机进行工作的程序就称为计算机操作系统。

- 操作系统的功能:实现计算机 CPU 管理、存储管理、文件管理、设备管理、作业管理。
- 操作系统的目的:方便用户;使计算机设备得到充分利用。

显然,没有操作系统,人们要方便且充分地利用计算机是不可能的。各种程序设计语言就是在操作系统的支持下运行的,计算机设备、操作系统、高级语言编译系统和用户自己编制的应用程序之间按层次构成的关系如下所示:

IV　用户的应用源程序 ……应用软件级
III　高级语言软件等系统软件 ……系统软件级
II　操作系统 ……操作系统级
I　裸机 ……硬件级

1.1.4　使用高级语言编程的过程

使用计算机高级语言对某一问题进行编程,需要经历如下过程:

(1) 分析问题。

(2) 确定算法。

(3) 使用指定的高级语言描述算法(编程),并得到用该语言写出的程序(源程序)。

(4) 将源程序输入计算机(编辑)。

(5) 源程序必须翻译为机器码的程序后计算机才能识别,进而得到可以在机器上进行工作的程序(可执行程序)。对于其中的翻译过程,不同的计算机高级语言是使用不同的方式实现的。

将高级语言的源程序翻译为可执行程序的方式一般分为两类。

(1)解释执行

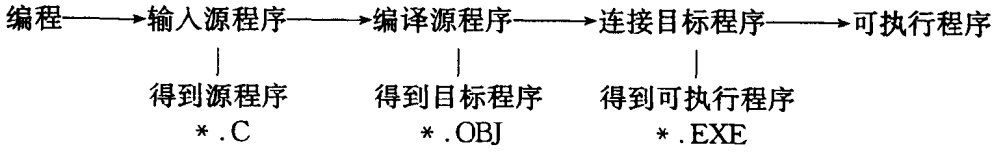
一边翻译一边执行。每次执行都是从源程序开始执行。因此,每次程序执行所占用的时

间实际是翻译和计算两部分时间。以解释方式执行的高级语言程序便于调试，因为它的工作方式是一边翻译，一边执行，一旦发现错误，就停止下来。调试者这时可以方便地直接观察当时程序工作的现场状态，有利于发现逻辑错误。但解释执行的程序运行速度较慢，而且程序必须在语言系统中才能工作。

(2)编译执行

这种执行方式先要经过编译和连接两个步骤，才能得到可执行程序。在执行程序的过程中，只需直接使用可执行程序，而不必每次进行编译。因此，与解释运行的方式相比较，这种方式执行程序的速度更快。又由于可执行程序是完全独立于语言系统本身的，所以使用起来很方便。

本书所讲述的C语言就是一种编译执行的高级语言，使用C语言进行工作的过程可以通过下述流程描述：



1.2　C语言的一般特点与C语言程序的构成

1.2.1　C语言的特点

1.C语言的底层操作能力强——适合于进行系统程序设计

由于C语言是介于低级语言和高级语言之间的中级语言，它能实现过去只能由低级语言才能实现的许多功能，例如，对计算机寄存器操作、二进制位操作以及对操作系统的系统功能的调用等等。

2. 丰富的数据结构

C语言的数据类型十分丰富，它不仅有一些基本数据类型，如整型、实型、字符型等等，而且还支持用户利用基本数据类型构造新的数据类型，如结构、联合、枚举等等，为用户能够设计出满足自己需要的数据结构提供了很高的自由度。

3. 丰富的运算符

利用C语言丰富的基本运算符，可完成算术运算、逻辑运算、位运算、指针运算、赋值运算、条件运算等各种运算和控制操作，在很大程度上提高了C语言程序的灵活性。

4. 丰富的系统函数

C语言系统中具有丰富的系统函数。C语言本身语句很少，大多数功能都是通过函数来完成的。系统函数存放在不同的库文件当中，这种处理方式使C语言系统的核心部分规模较小，外围函数可以做得很丰富，并且可以根据需要随时增加、调整、修改函数，这不仅提高了语言系统本身的灵活性，而且极大地提高了系统可移植性。

1.2.2　C语言的系统函数

C语言的功能很强,其中绝大部分功能是通过其系统函数实现的。系统中的主要函数可以分为12类(Turbo C V2.0 中的函数统计):

1. 类型转换函数　16个
2. 类型判断函数　16个
3. 目录控制函数　13个
4. 图形屏幕函数　96个
5. 输入输出函数　90个
6. 接口处理函数　60个
7. 字符串处理函数　36个
8. 数学函数　48个
9. 内存分配函数　15个
10. 进程控制函数　20个
11. 标准化函数　30个
12. 杂函数　10个

利用系统函数,程序设计人员可以方便灵活地完成各种比较复杂的操作。因此,C语言丰富的系统函数成功地降低了程序员的工作强度和工作复杂性。

1.2.3　C语言程序的程序构成和基本语法

1. C程序的整体结构

C语言程序的一般构成:

头文件:是C系统中特有的文件。它可以分门别类地对函数所需要的特殊数据结构进行定义说明,文件用ASCⅡ码方式存储,可以方便地阅读或补充所需要的内容。

全程变量说明:用于定义在整个程序中有效的变量。

主函数、子函数:定义程序各个部分的处理过程。

2. C语言的程序单位

由C程序的整体结构可知C语言程序是由函数组成的,每个程序中必须有一个且只能有一个主函数(名称为main),整个程序由一个主函数和0~N个子函数构成。因此,函数是C程序的基本单位。由此可见,函数的概念在C语言中占有十分重要的地位。

3. 函数的结构

C语言的函数定义的组成:

- 函数名称(参数列表)
- 参数说明
- {　局部变量说明;
  　处理过程描述;　}

花括号{　}中的内容称为函数体,处理过程由C语言语句组成。

4. C语言程序的基本语法规定

(1) 语句规则:

- 每条语句使用“;”为结束符。
- 每行可以写多个语句。

(2) 标识符:

- 标识符指程序中为标识变量、常量、函数、数组及其他数据结构所使用的名称。
- 标识符必须由字母或下划线开头,由字母、数字和下划线组成。
- 标识符的长度不限,但是系统的识别长度是有限的。只有在系统识别长度范围内有区别的标识符,系统才认为是不同的标识符。例如,假设在一个识别长度为8的系统中,存在标识符Student-1和Student-2,由于这两个标识符在8个字符的识别长度范围内是相同的而被系统认为是同一个标识符,如果这两个标识符处在识别长度为10的系统中,则可以被识别为两个不同的标识符。
- 程序能够识别标识符的大小写。即对于使用大写和小写标识的同一个英文字母,系统认为是两个不同的标识符。例如,如果在C程序中定义了名称为A的标识符,则在使用变量时必须使用A来实现对该变量的调用,若使用了a,则系统会提示错误——“变量a未经过定义”。如果在程序中定义了变量A和a系统能够确认它们是两个不同的变量。

5. 程序综合实例

```
#include <stdio.h>          /* 头文件连接 */
int a;                      /* 全程变量定义 */
main()                      /* 主函数开始 */
{ int b;                    /* 主函数中局部变量说明 */
  a=6;
  scanf("%d",&b);           /* 输入变量内容 (scanf是C语言的输入函数) */
  a=a+b*3;                  /* 计算 */
  printf("%d %d",a,b);      /* 输出变量内容 (printf是C语言的输出函数) */
}
```

1.3　程序设计的概念

1.3.1　什么是程序

简而言之,程序=算法+数据结构。

数据是程序操作的对象。因此,在编制程序时,首先应确定被处理数据的表达形式,其次再明确施加在数据上的操作,进而设计操作的具体步骤。操作步骤就是算法。

上述公式仅仅是对规模较小的程序而言,对于规模较大的程序系统来说,还必须考虑如何进行程序系统的设计。简单地说,就是把一个总体的大任务先分解为若干个规模较小的子任务,然后分别对每个小任务(模块)进行程序设计,以便降低程序设计的复杂程度,提高工作的效率,提高程序系统的质量。程序设计人员多年来使用的结构化程序设计和软件工程的方法,其目的就在于使程序设计工作规范化,求得软件开发的高速度和高质量。