



全国高职计算机专业教材

院士教授、企业资深从业人员、职教一线教师共同打造

◎顾问 张效祥 院士 ◎总主编 邱玉辉 教授

C语言程序设计

徐受蓉 主编



西南师范大学出版社



全国高职计算机专业教材

院士教授、企业资深从业人员、职教一线教师共同打造

● 顾问 张效祥 院士 ● 总主编 邱玉辉 教授

C语言 程序设计

西南师范大学出版社

图书在版编目(CIP)数据

C 语言程序设计/徐受蓉主编. —重庆:西南师范大学出版社,2006.6

ISBN 7-5621-3613-0

I. C... II. 徐... III. C 语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 054566 号

全国高职计算机专业教材

顾问:张效祥 院士

总主编:邱玉辉 教授

总策划:周安平 李远毅

执行策划:周 松 张浩宇

C 语言程序设计

徐受蓉 主编

责任编辑:李虹利

封面设计:唐小慧

出版发行:西南师范大学出版社

(重庆·北碚 邮编 400715)

网址:<http://www.xscbs.com>)

印刷者:重庆师范大学印刷厂

开 本:787mm × 1092mm 1/16

印 张:16.5

字 数:430 千字

版 次:2006 年 8 月 第 1 版

印 次:2006 年 8 月 第 1 次印刷

书 号:ISBN 7-5621-3613-0/TP · 51

定 价:25.00 元

《全国高职计算机专业教材》编委会联系方式

联系人:周 松 张浩宇

电 话:023-68254356 13908317565 13883206497

地 址:重庆市北碚区西南师范大学出版社内

邮 编:400715

E-mail:qggzsjjc@yahoo.com.cn

《全国高职计算机专业教材》总编委会

总编委会顾问

张效祥 中国科学院院士、著名计算机专家、“两弹一星”功臣

总编委会主任

邱玉辉 西南大学人工智能研究所所长、教授、博士生导师

总编委会副主任

黄国兴 华东师范大学软件学院 院长、教授

王能忠 四川托普信息技术职业学院 院长、教授

张为群 西南大学计算机与信息科学学院 院长、教授

汪林林 重庆邮电大学软件学院 原院长、教授

李吉桂 华南师范大学计算机科学系 原系主任、教授

张 杰 西北大学软件职业技术学院 院长、教授

徐受容 重庆电子职业技术学院计算机系 主任、教授

丛书总序

CHONGXING JIAOYU

总主编 邱玉辉

高等职业教育是我国高等教育体系的重要组成部分。近年来，国家高度重视职业教育，并为推动我国职业教育跨越式发展，颁发了《国务院关于大力推进职业教育改革与发展的决定》，提出了将高等职业教育学制逐步由目前的三年改为两年的改革方向。

教材是提高教育质量的关键之一。信息产业部电子教育中心调查后认为，现在使用的教材多数是普通高校本科教材的压缩和简化，偏重理论知识的介绍，而案例教学、项目教学的内容极少，实用技能的训练更是不足，课程内容滞后于专业技术的更新与发展，与社会需求和行业发展相脱节，从而导致学生分析问题和解决问题的能力，特别是职业能力较弱，毕业的学生很少能直接顶岗工作。

为落实国家大力发展职业技术教育的重大决策和解决目前缺乏面向两年学制的高职计算机专业系列教材的问题，我们组织开发了这套《全国高职计算机专业教材》。

这套教材由我国著名计算机专家、“两弹一星”功臣张效祥院士担任顾问，并得到中央教育科学研究所的大力支持。其编写指导思想是：需求牵引，改革驱动，理论适度，着眼技术，立足实用，培养能力。我们通过总结当前职业教育专家教学改革的最新研究成果，紧紧依靠高职院校从事计算机教育的一线教师，以培养技能型紧缺人才为目标，让学生明白Why，知道What，重点学会How，把理论与实践融为一体，既考虑了每门课程本身的科学性，又兼顾了课程间的联系与衔接。全套教材具有重点突出，针对性强；结构清晰，循序渐进；模块结构，易教易学等特点。此外，我们还将为教材配备包含教参和习题解答等内容的光盘，供教师参考和学生自学。

总之，这套教材经过长期策划，精心打造，认真审读，终于问世了。它倾注了编写教师、总编委会以及出版社的大量心血。如果它能够对我们的高职计算机教育有所助益，那么我们的目的就达到了。

前言

C 语言是目前国内外广泛使用的程序设计语言之一。C 语言功能强、使用灵活、可移植性好、目标代码质量高,它既具有高级语言的优点,又具有低级语言的许多特点,既适合编写应用程序,又适合开发系统软件。C 语言程序设计已成为高等院校普遍开设的一门计算机程序设计课程。

本书根据高职学院的办学宗旨,从程序设计的基本概念和基本原理出发,立足“理论适度,着眼技术,注重实用,培养能力”的原则,对 C 语言知识进行了系统介绍。全书共分十章。主要内容包括 C 语言概述,数据类型、运算符及表达式,顺序结构程序设计,选择结构程序设计,循环结构程序设计,数组,函数与编译预处理,指针,结构体与共用体(该书仅涉及到结构体与共用体的初步知识,较深入的内容会在后续课程如:《数据结构》中学习)和文件。我们精选教材内容,灵活运用有实用意义的案例教学,融入以应用技术为核心的职业能力培养。

本书突出 C 语言程序设计课程的应用性、实践性特点,始终贯穿能力培养主线。通过强化应用性教学内容,力求达到在应用中学习知识、培养能力的目的。

本书由徐受蓉主编,武春岭、李怡平副主编,参加本书编写的还有余辉、胡钢、杨俭、黄从云、王亚宁、赵波、陈磊、蒋文豪、罗中华。全书由徐受蓉统稿定稿。

由于作者水平有限,书中的不足和错误在所难免,恳请广大读者批评指正。

编 者

2006 年 2 月

 第一章 C 语言概述	(1)
第一节 C 语言出现的历史背景	(1)
第二节 C 语言的特点	(2)
第三节 简单的 C 程序介绍	(4)
第四节 C 语言的运行步骤	(8)
第五节 实 训	(14)
  第二章 数据类型、运算符及表达式	(17)
第一节 C 语言的数据类型	(17)
第二节 常量和变量	(18)
第三节 整型数据和实型数据	(21)
第四节 字符型数据	(25)
第五节 运算符和表达式	(28)
第六节 实 训	(39)
  第三章 顺序结构程序设计	(43)
第一节 C 语句概述	(43)
第二节 程序的三种基本结构	(46)
第三节 数据的输入输出	(48)
第四节 顺序结构程序设计举例	(53)
第五节 实 训	(56)
  第四章 选择结构程序设计	(59)
第一节 if 语句	(59)
第二节 switch 语句	(65)
第三节 选择结构程序设计举例	(69)
第四节 实 训	(73)
  第五章 循环结构程序设计	(81)
第一节 while 语句	(82)

第二节	do - while 语句	(83)
第三节	for 语句	(86)
第四节	多重循环	(90)
第五节	break、continue 和 goto 语句	(92)
第六节	循环结构程序设计举例	(96)
第七节	实 训	(102)
 第六章	数 组	(109)
第一节	数组的概念	(109)
第二节	一维数组	(110)
第三节	排序和查找	(114)
第四节	二维数组	(118)
第五节	字符数组和字符串	(122)
第六节	实 训	(131)
 第七章	函数与编译预处理	(135)
第一节	函数的定义	(135)
第二节	函数的参数与返回值	(138)
第三节	函数的调用	(141)
第四节	局部变量和全局变量	(148)
第五节	编译预处理命令	(156)
第六节	实 训	(162)
 第八章	指 针	(171)
第一节	指针的概念	(171)
第二节	指向变量的指针变量	(174)
第三节	指向一维数组的指针变量	(185)
第四节	指向字符串的指针变量	(189)
第五节	实 训	(192)
 第九章	结构体和共用体	(201)
第一节	结构体类型的定义	(201)
第二节	结构体变量的定义、初始化及引用	(203)
第三节	结构体数组	(207)
第四节	指向结构体数据的指针变量	(210)

第五节 共用体	(216)
第六节 实 训	(218)
 第十章 文 件	(223)
第一节 文件概述	(223)
第二节 文件的打开与关闭	(226)
第三节 文件的读写函数	(232)
第四节 实 训	(239)
附录 I 常用字符与 ASCII 编码对照表	(245)
附录 II C 语言中的关键字	(246)
附录 III 运算符和结合性	(247)
附录 IV C 常用库函数	(248)
参考文献	(252)

第一章 C 语言概述

学习要求:

要求了解 C 语言的特点,掌握 C 程序的基本结构和书写格式,熟悉 C 程序的编辑、编译、调试和运行过程。

主要内容:

本章介绍了 C 语言出现的历史背景和特点,然后通过三个简单的 C 程序实例分析,总结 C 程序的总体结构和书写格式。最后重点介绍了 C 程序的开发环境和运行步骤。

第一节 C 语言出现的历史背景

C 语言是国际上广泛流行的、很有发展前途的计算机高级语言。它适合作为系统描述语言,既可以用来写系统软件,也可用来写应用程序。

C 语言是由 D. M. Ritchie 于 1972 年在 B 语言的基础上设计的,主要用于编写 UNIX 操作系统。后来 C 语言进行了多次改进,1977 年出现了可移植的 C 语言编译程序,使得 C 语言编写的 UNIX 操作系统可以在各种计算机上使用。随着 UNIX 的广泛应用,C 语言得到了普及和推广,并最终独立于 UNIX 而成为世界上应用最广泛的计算机语言之一(C 语言发展过程如图 1-1 所示)。

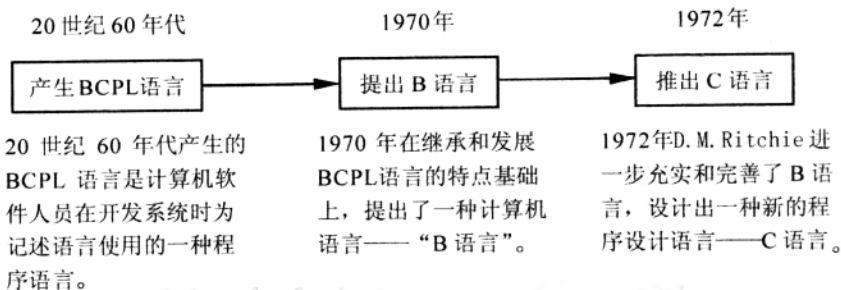


图 1-1 C语言的发展过程

提醒: 以前的操作系统等系统软件主要是用汇编语言编写的(包括UNIX操作系统在内)。由于汇编语言依赖于计算机硬件,程序的可读性和可移植性都比较差。为了提高可读性和可移植性,最好改用高级语言,但一般高级语言难以实现汇编语言的某些功能(汇编语言可以直接对硬件进行操作,例如:对内存地址的操作、位操作等)。人们希望能否找到一种既具有一般高级语言特性、又具有低级语言特性于一身的语言,于是,C语言诞生了。

在C语言的推广过程中,出现了多种版本。为了解决这些版本的兼容问题,1983年美国国家标准协会(ANSI)制订了C语言标准,即ANSI C。目前流行的C编译系统都是以此为基础的,如Microsoft C、Turbo C、Quick C、Borland C等。这些版本的C语言虽然都以ANSI C为标准,但也有些差异,读者可参考相关手册。本书以ANSI C为基础。

第二节 C语言的特点

- C语言具有以下特点:
- 1. 语言简洁、紧凑,使用方便、灵活。
 - C语言一共只有32个关键字,9种控制语句,程序书写形式自由,语句简练。
 - 2. 运算符丰富,适用的范围广泛。
 - C语言的运算符包含的范围很广,共有34种运算符,它把括号、赋值符号、强制类型转换符号等都作为运算符处理,从而使C语言的运算符类型极为丰富,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和操作。

3. 数据结构丰富,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。这些丰富的数据类型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 C 语言的指针类型,功能强大、灵活多样,恰当地使用不仅可以简化程序结构,而且可以节省存储空间、提高运算速度。

4. 具有结构化的控制语句(如 if...else 语句、while 语句、do...while 语句、switch 语句、for 语句)。

这些语句可以实现程序中所有的控制结构,C 语言的控制语句符合结构化程序设计要求,并且用函数作为程序模块,使得程序结构清晰、可读性强、易于调试。

5. 编程限制少,程序设计自由度大。

例如对数组下标越界不作检查,由程序编写者自己保证程序的正确性。对变量的类型使用比较灵活,如整型量与字符型数据以及逻辑型数据可以通用,使得某些运算变得更加简单、直接。

一般的高级语言的语法规则和检查比较严格,能检查出几乎所有的语法错误。而 C 语言允许程序的编写有较大的自由度,因此放宽了语法检查。编写者应当仔细检查程序,保证其正确性,而不要过分依赖于编译软件去查错。就这一点来说,C 语言比其他语言对程序员的要求要高。

6. 可直接对硬件操作。

C 语言允许直接访问物理地址,直接对硬件操作,提供对字节、位、内存和寄存器操作,能实现汇编语言的大部分功能。这个特点使得 C 语言既具有高级语言的功能,又兼备低级语言的许多功能,可以用来写系统程序。

7. 生成目标代码质量高,程序执行效率高。

C 语言生成的目标代码一般只比汇编语言生成的目标代码的效率低 10% ~ 20%。

8. 用 C 语言写的程序可移植性好。

与汇编语言相比,用 C 语言写的程序基本上不作修改就能用于各种型号的计算机和操作系统,使程序具备了很好的移植性。

以上所述的是 C 语言的一般特点,至于其内部的其他特点将结合以后各章节内容逐一进行介绍。正是 C 语言的这些特点,使得它的应用非常广泛。许多大的软件都用 C 语言编写,这主要是由于 C 语言的可移植性好和对硬件的控制能力高,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都改用 C 语言来处理了。

总之,C 语言对编程者要求较高。由于使用 C 语言编写程序会感到很少限制,灵活性大,功能强,可以编写出任何类型的程序,因此学习和使用 C 语言是学习计算机编程的基础。

第三节 简单的 C 程序介绍

一、C 语言程序的总体结构

一个完整的 C 语言程序,是由一个 main() 函数(又称主函数)和若干个函数组合而成的,或仅由一个 main() 函数构成。下面我们通过几个例子,从中分析 C 程序的组成和形式。

例 1.1 仅由 main() 函数构成的 C 语言程序。

```
/* 程序 L1_1. C 功能:在屏幕上显示一句话:Hello World! */
#include <stdio.h>          /* 包含输入输出头文件 */
main()                     /* 定义主函数 */
{
    printf(" Hello World! \n"); /* 在屏幕上输出“Hello World!”,光标移到下一行 */
}
```

运行结果为:

Hello World!



提醒: (1) C 程序的注释信息可以出现在程序的任何地方,它总是以“/*”符号开始,以“*/”符号结束。注释是给人看的,目的是方便程序的阅读和调试,对编译和运行不起作用,不会出现在目标程序中。

(2) 只要程序用到输入输出函数,如 printf() 函数,就需要在程序中包含标准输入输出头文件 stdio.h,即 #include <stdio.h>。

(3) main 表示“主函数”,每一个 C 程序都必须有且仅有一个 main() 函数,这样程序运行时才可以找到入口。

(4) 大括号 {} 括起来的部分称为函数体,在其中写入该函数要执行的功能。本例中主函数内使用的 printf() 是 C 语言的输出函数,将双引号中的字符串原样输出。“\n”是换行符,即在“Hello World!”输出之后回车换行。

(5) 每一个语句后用“;”(分号)表示结束。

例 1.2 仅由 main() 函数构成的 C 语言程序(函数体中使用到变量)。

```
/* 程序 L1_2. C 功能:求两个整数之和 */
#include <stdio. h >
main()
{int a,b,sum;          /* 声明部分,定义整型变量 a、b、sum */
a = 123;b = 456;        /* 给变量 a、b 赋值 */
sum = a + b;            /* 将变量 a 和 b 之和赋给变量 sum */
printf("sum is %d\n",sum); /* 输出变量 sum 的值 */
}
```

运行结果为:

sum is 579



提醒: %d 表示执行输出时此位置将由变量 sum 的十进制整数值代替。

例 1.3 由 main() 函数和 1 个其他函数 max() 构成的 C 语言程序。

```
/* 程序 L1_3. C 功能:通过键盘输入两个实数,输出其中的最大数 */
#include <stdio. h >
float max(float x,float y) /* 定义求实数 x,y 的最大数的函数 max */
{
float z; /* max 函数中的声明部分,定义本函数中用到的实型变量 z */
if (x > y) z = x; /* 如果 x 大于 y,则 z 等于 x */
else z = y; /* 否则 z 等于 y */
return(z); /* 返回 z,则 max 函数值为 z 的值 */
}
main() /* 定义主函数 */
{float a,b,c; /* 声明部分,定义 a、b、c 三个变量为实型变量 */
printf("Please input two float numbers (a,b):"); /* 输出屏幕提示信息 */
scanf("%f,%f",&a,&b); /* 从键盘输入变量 a 和 b 的值 */
c = max(a,b); /* 调用 max 函数,将得到的最大数返回值赋给变量 c */
printf("max = %f",c); /* 输出最大数 c 的值 */
}
```

运行结果为:

Please input two float numbers (a,b): 3.1,8.9 ↵

max = 8.900000

! 提醒: (1) 程序由主函数 `main()` 和用户自定义函数 `max(float x, float y)` 组成, 用户自定义函数 `max` 用于求出给定两个实数 x 、 y 的最大数。

(2) 用户自定义函数 `max(float x, float y)` 内容在大括号内, 包括: 定义存放最大数的变量 z ; 进行运算(用 `if-else` 语句实现 z 等于 x 、 y 的最大值); 最后 `return z` 使得 `max(float x, float y)` 的结果等于 z 。

(3) 主函数内容同样在大括号内, 包括: 定义所有要使用的变量, 在执行时屏幕先给出提示信息, 然后 `scanf()` 输入函数可以等待用户通过键盘输入两个数 3.1, 8.9 给 a 、 b (注意两个数之间用逗号间隔), 接着 c 等于 `max(a, b)` (即 a 、 b 的最大数), 最后输出实数 c 的值的位用 `%f` 代替。

(4) 每一个语句结束时加“;”。



重点: C 语言程序结构特点。

通过对以上几个例子的分析, 我们可以总结出 C 语言程序结构具有如下特点:

1. 函数是 C 程序的基本组成单位。

`main()` 函数的作用相当于其他高级语言的主程序, 其他函数的作用相当于子程序。

其中被调用的函数可以是系统提供的库函数 (`printf()` 和 `scanf()` 函数), 也可以是由用户根据需要编制设计的函数 (如 `max()` 函数)。C 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。可以说 C 是函数式的语言, 程序中的全部工作都是由各个函数分别完成的。编写 C 程序就是编写一个个函数。C 的函数库十分丰富, 此特点使之容易实现程序的模块化。

2. C 语言程序总是从 `main` 函数开始执行的, 而不论 `main` 函数在整个程序中的位置如何。

3. C 语言本身没有输入和输出语句。输入和输出的操作是由库函数 `scanf()`, `printf()` 等函数来完成的, 即 C 对输入输出实行“函数化”。

4. 一个函数由两部分组成:

(1) 函数的首部 (函数的第一行)。

例: float	max	(float	x,	float	y)
↓	↓	↓	↓	↓	↓
函数类型	函数名	函数参数类型	函数参数名	函数参数类型	函数参数名



提醒: 一个函数名后面必须跟一对圆括弧, 函数参数可以没有, 如 `main()`。

(2) 函数体,即函数首部下面的大括弧内的部分。

如果一个函数内有多个大括弧,则最外面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。

执行部分,由若干个语句组成。



提醒: (1) 函数体中的变量定义语句,必须在所有可执行语句之前。

如下面程序中变量定义语句“int sum;”的位置是非法的:

```
main()
{
    int a,b;          /* 变量定义语句:定义两个整型变量 a,b */
    a=2;              /* 可执行的赋值语句:将2赋值给变量 a */
    b=3;              /* 可执行的赋值语句:将3赋值给变量 b */
    int sum;           /* 变量定义语句:出现在可执行的赋值语句“a=2;”和
                        “b=3;”之后,非法! */
    sum = a + b;
```

二、C 语言程序的书写格式

1. C 程序书写格式自由,一行内可以写几个语句,一个语句可以分写在多行上。各语句之间用分号分隔,每个语句和数据定义的最后必须有一个分号。

2. 可以用“/*……*/”对 C 程序中的任何部分作注释,以提高程序的可读性。



提醒: “/”和“*”以及“*”和“/”之间不能有空格,否则会出错。



技巧: 为了避免遗漏必须配对使用的符号,例如注释符号、函数体的起止标识符(大括号)、圆括号等,在输入时,可连续输入这些起止标识符,然后再在其中进行插入来完成内容的编辑。这在起止标识符嵌套时,以及相距较远时,更有必要。

3. 注释的位置,可以单独占一行,也可以跟在语句的后面。另外如果一行写不下,可另起一行继续写。

4. 注释中允许使用汉字。在非中文操作系统下,看到的会是一串乱码,但不影响程序运行。

第四节 C 语言的运行步骤

一、从输入到运行一个 C 语言程序的一般过程

从输入到运行一个 C 程序,一般要经历编辑、编译、连接、运行四个步骤:

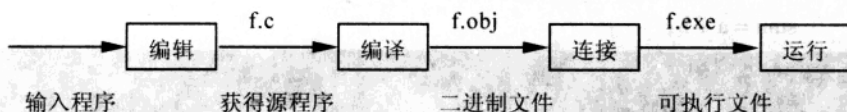


图 1-2 从输入到运行一个 C 程序的过程示意图



重点: (1) 用高级语言所提供的语句和函数写出的语句序列叫源程序。用户通过编辑器,如 Turbo C 的集成环境内的编辑器、Windows 的记事本等其他文字处理软件,将自己编写的 C 语言程序输入计算机的过程称为 C 程序源文件的编辑。编辑生成的文件以文本形式存储,扩展名为“.c”,如上面示意图中“编辑”生成的文件即为源程序。

(2) “编译”将 C 的源程序翻译成计算机可以识别的二进制形式的目标程序,扩展名为“.obj”。

(3) “连接”将目标程序和其他目标程序模块以及系统提供的 C 库函数等进行连接,生成可执行文件,扩展名为“.exe”。

(4) 计算机的语言分为低级语言和高级语言,二进制语言和汇编语言属于低级语言,C 语言、BASIC 语言等属于高级语言。只有用二进制语言写的程序,计算机才可以直接执行。汇编语言编写的程序,需要通过汇编程序的翻译后才能执行。高级语言写的程序(称为源程序),也必须翻译成二进制程序后才能执行。源程序的翻译过程有两种方式:一种是解释方式,如 BASIC 语言写的源程序,通过解释程序翻译一句,执行一句,不生成目标程序;另一种是编译方式,如 C 语言,通过编译程序一次翻译产生目标程序,然后执行。