



世纪高职高专规划教材
高等职业教育规划教材编委会专家审定

C YUYAN CHENGXU SHEJI

C 语言程序设计

主 编 段智毅 杨 辉
副主编 张 淼 孙 坤 高 阁



北京邮电大学出版社
www.buptpress.com

21 世纪高职高专规划教材

高等职业教育规划教材编委会专家审定

C 语言程序设计

主 编 段智毅 杨 辉

副主编 张 森 孙 坤 高 阁

北京邮电大学出版社

· 北 京 ·

内 容 提 要

C 语言功能强大,使用灵活,可移植性好,既具有高级语言的优点,又具有低级语言的许多特点,既可以用来编写系统软件,又可以用于编写应用软件,是国内外广泛使用的计算机语言,也是计算机应用人员应掌握的一种程序设计工具。

本书针对 C 语言初学者和高职高专学生的特点,从 C 语言的基本概念、基本知识、基本技能、基本的编程思想入手,力求内容简捷明快、重点突出、定位准确、深入浅出,是高职高专学生学习 C 语言程序设计的理想教材,也是 C 语言初学者自学的好教材。

图书在版编目(CIP)数据

C 语言程序设计/段智毅,杨辉主编. —北京:北京邮电大学出版社,2009

ISBN 978-7-5635-2022-0

I. C… II. ①段…②杨… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 112511 号

书 名: C 语言程序设计

作 者: 段智毅 杨 辉

责任编辑: 彭 楠

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编:100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京源海印刷有限责任公司

开 本: 787 mm×1 092 mm 1/16

印 张: 12.5

字 数: 303 千字

印 数: 1—3 000 册

版 次: 2009 年 7 月第 1 版 2009 年 7 月第 1 次印刷

ISBN 978-7-5635-2022-0

定 价: 22.00 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

编 委 会

主 编：段智毅 杨 辉

副主编：张 森 孙 坤 高 阁

委 员：（按姓氏笔画排序）

王 兰 李荣涛 曾丽娟

谭鹤毅

前 言

C语言功能强大,使用灵活,可移植性好,既具有高级语言的优点,又具有低级语言的许多特点,既可以用来编写系统软件,又可以用于编写应用软件,是国内外广泛使用的计算机语言。掌握了C语言,在进一步学习C++时就能以一种熟悉的语法来学习面向对象的语言,从而达到事半功倍的目的。但在长期的高职高专教育教学工作过程中,我们发现不少学生感到C语言涉及的概念复杂、规则繁多、灵活性强、容易出错,他们认为学习C语言比较困难,迫切需要一本“易学、易懂、易用”的教材。加之为体现高职高专教育教学的特点,以学生能力培养为主线,力求掌握基本概念、基本知识和基本理论,达到“学会、会用、够用”的目的,我们组织了部分教学经验丰富的高职高专院校的一线教师,针对初学者和高职高专学生的特点,精选内容,降低难度,突出重点,通俗易懂地编写这本教材,力求达到“好懂、好学、好用”的目的,使学生和初学者不至于对C语言望而止步,培养学生浓厚的学习兴趣和热情。

本书共分11章。其中,第1章、第10章以及附录由杨凌职业技术学院段智毅老师编写;第2章由辽宁信息职业技术学院孙坤老师编写;第3章由辽宁职业技术学院张森老师编写;第4章和第8章由辽宁信息职业技术学院王兰老师编写;第5章由河南职业技术学院曾丽娟老师编写;第6章和第7章由辽宁职业技术学院李荣涛老师编写;第9章由辽宁信息职业技术学院高阁老师编写;第11章由南充职业技术学院谭鹤毅老师编写。

本书由杨凌职业技术学院段智毅老师担任第一主编,山东水利职业学院杨辉老师担任第二主编。同时,为了帮助读者学习本书,作者还编写了一本配套实训教材,由山东水利职业学院杨辉老师担任第一主编,杨凌职业技术学院段智毅老师担任第二主编。由于作者水平有限,书中难免存有缺点和错误,敬请广大专家与读者批评指正。

编 者

目 录

第 1 章 C 程序设计概述	(1)
1.1 C 程序设计语言概述	(1)
1.1.1 C 程序设计语言的发展	(1)
1.1.2 C 语言的特点	(1)
1.2 简单的 C 程序介绍	(2)
1.3 C 程序的基本结构及规则	(2)
1.4 C 语言基本语法成分	(3)
1.4.1 C 语言字符集	(3)
1.4.2 标识符	(3)
1.4.3 关键字	(3)
1.5 C 语言程序的开发环境	(4)
习题一	(5)
第 2 章 C 程序设计基础	(6)
2.1 常量、变量和标识符	(6)
2.1.1 常量与标识符	(6)
2.1.2 变量	(7)
2.2 整型数据	(7)
2.2.1 整型常量	(7)
2.2.2 整型变量	(7)
2.3 实型数据	(9)
2.3.1 实型常量	(9)
2.3.2 实型变量	(9)
2.4 字符型数据	(10)
2.4.1 字符型常量	(10)
2.4.2 字符型变量	(11)
2.4.3 字符串常量	(12)
2.5 不同数据类型间的混合运算	(13)
2.6 算术表达式	(13)
2.6.1 算术运算符	(13)

2.6.2 运算符的优先级、结合性及算术表达式	(14)
2.6.3 强制类型转换表达式	(14)
2.7 赋值表达式	(15)
2.7.1 赋值运算符	(15)
2.7.2 简单赋值运算	(15)
2.7.3 复合赋值运算	(16)
2.8 自加、自减运算符和逗号运算符	(17)
2.8.1 自加、自减运算符	(17)
2.8.2 逗号运算符和逗号表达式	(18)
习题二	(19)
第 3 章 顺序结构	(21)
3.1 赋值语句	(21)
3.2 数据输入	(22)
3.2.1 scanf 函数的一般调用形式	(22)
3.2.2 scanf 函数中常用的格式	(22)
3.3 数据输出	(25)
3.3.1 printf 函数的一般调用形式	(25)
3.3.2 printf 函数中常用的格式	(26)
3.4 程序举例	(28)
习题三	(29)
第 4 章 选择结构	(31)
4.1 关系运算符和逻辑运算符	(31)
4.1.1 关系运算符和关系表达式	(31)
4.1.2 逻辑运算符和逻辑表达式	(32)
4.2 if 语句	(35)
4.2.1 单臂 if 语句	(35)
4.2.2 双臂 if 语句	(35)
4.2.3 嵌套的 if 语句	(37)
4.3 条件表达式	(41)
4.4 switch 语句	(42)
4.4.1 switch 语句	(42)
4.4.2 break 语句	(44)
4.5 程序举例	(45)
习题四	(47)
第 5 章 循环结构	(52)
5.1 while 语句	(52)

5.2 do-while 语句	(53)
5.3 for 语句	(54)
5.4 循环结构的嵌套	(55)
5.5 break 和 continue 语句在循环结构中的作用	(57)
5.5.1 break 语句	(57)
5.5.2 continue 语句	(58)
5.6 程序举例	(59)
习题五	(61)
第 6 章 字符型数据	(63)
6.1 调用 scanf 和 printf 函数输入和输出字符	(63)
6.2 调用 putchar 和 getchar 函数输入和输出字符	(64)
6.3 程序举例	(65)
习题六	(66)
第 7 章 数组	(67)
7.1 一维数组	(67)
7.1.1 一维数组的定义	(67)
7.1.2 一维数组元素的引用	(69)
7.1.3 一维数组的初始化	(69)
7.1.4 一维数组程序举例	(70)
7.2 二维数组	(73)
7.2.1 二维数组的定义	(73)
7.2.2 二维数组元素的引用	(74)
7.2.3 二维数组的初始化	(75)
7.2.4 二维数组程序举例	(76)
7.3 字符数组	(77)
7.3.1 字符数组的定义	(77)
7.3.2 字符数组与字符串的关系	(78)
7.3.3 字符数组的引用	(78)
7.3.4 字符数组的输入输出	(80)
7.3.5 字符串处理函数	(80)
7.3.6 字符串数组程序举例	(83)
习题七	(85)
第 8 章 函数	(88)
8.1 函数的定义和返回值	(88)
8.1.1 函数的分类	(88)
8.1.2 函数的定义	(89)
8.1.3 函数的返回值	(90)

8.2 函数的调用方式	(90)
8.3 函数的说明	(93)
8.3.1 函数说明的形式	(93)
8.3.2 函数说明的位置	(94)
8.4 函数调用时数据的传递	(95)
习题八	(96)
第 9 章 指针	(101)
9.1 变量的地址和指针	(101)
9.2 指针变量	(102)
9.2.1 指针变量的定义	(102)
9.2.2 指向变量的指针	(103)
9.2.3 指向数组的指针	(108)
9.2.4 指向字符串的指针	(119)
9.2.5 指向函数的指针	(123)
9.3 程序举例	(128)
习题九	(131)
第 10 章 结构体、共用体和用户自定义类型	(139)
10.1 用 typedef 说明一种新类型名	(139)
10.2 结构体类型变量的定义及引用	(140)
10.2.1 结构体类型的定义	(140)
10.2.2 结构体类型变量的定义	(141)
10.2.3 结构体变量、数组的初始化与引用	(142)
10.3 指向结构体类型的数据指针	(146)
10.3.1 指向结构体类型的数据指针的定义和初始化	(146)
10.3.2 用结构体指针引用结构体成员	(146)
10.3.3 指向结构体数组的指针	(147)
10.4 共用体类型的定义	(147)
10.4.1 共用体类型变量的定义	(147)
10.4.2 共用体变量的引用	(148)
10.5 枚举类型	(149)
10.6 预处理命令	(150)
10.6.1 宏定义	(150)
10.6.2 文件包含	(153)
习题十	(155)
第 11 章 文件	(158)
11.1 文件的概念	(158)

11.1.1 文件与文件名	(158)
11.1.2 文件分类	(159)
11.2 文件指针	(160)
11.2.1 FILE 结构体	(160)
11.2.2 文件类型指针的定义	(160)
11.3 打开文件	(160)
11.4 关闭文件	(162)
11.5 fgetc 和 fputc 函数的调用	(163)
11.5.1 读字符函数	(164)
11.5.2 写字符函数	(164)
11.6 fscanf 函数和 fprintf 函数	(166)
11.6.1 格式读函数	(166)
11.6.2 格式写函数	(167)
11.7 fgets 函数和 fputs 函数	(168)
11.7.1 读字符串函数	(168)
11.7.2 写字符串函数	(169)
11.8 fread 函数和 fwrite 函数	(169)
11.8.1 读数据函数	(169)
11.8.2 写数据函数	(170)
11.9 文件定位函数	(170)
11.9.1 fseek 函数	(171)
11.9.2 ftell 函数	(172)
11.9.3 rewind 函数	(172)
习题十一	(174)
附录一 C 运算符的优先级与结合性	(177)
附录二 ASCII 码表	(178)
附录三 Turbo C 2.0 常用库函数及其标题文件	(179)
附录四 Turbo C 2.0 编译错误信息	(183)
参考文献	(185)

第 1 章

C 程序设计概述

本章学习目标

- 了解 C 程序设计语言的发展历史。
- 正确理解 C 语言的特点。
- 正确掌握 Turbo C 的基本操作。
- 深刻理解 C 程序的基本结构及规则。

本章要点内容

- 熟悉 C 程序的基本结构和书写风格,掌握 C 语言关键字和标识符的命名方法,了解 C 程序的基本结构。
- 掌握在 Turbo C 集成开发环境中编写 C 程序的方法。

本章学前要求

- 对程序设计语言有初步了解。

1.1 C 程序设计语言概述

1.1.1 C 程序设计语言的发展

在 C 语言诞生以前,系统软件主要是用汇编语言编写的。由于汇编语言程序依赖于计算机硬件,其可读性和可移植性都很差;但一般的高级语言又难以实现对计算机硬件的直接操作(这正是汇编语言的优势),于是人们盼望有一种兼有汇编语言和高级语言特性的新语言,因此,人们设法寻找一种既具有一般高级语言特性,又具有低级语言特性的语言,C 语言就在这种情况下应运而生。

早期的 C 语言主要是用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们所认识,到了 20 世纪 80 年代,C 语言开始进入其他操作系统,并很快在各类大、中、小和微型计算机上得到了广泛的使用,成为当代最优秀的程序设计语言之一。目前,在微型计算机上使用的有 Microsoft C、Quick C、Turbo C 等多种版本。

1.1.2 C 语言的特点

C 语言的特点如下:

- (1) 具有结构化语言的特点,用函数作为程序的组成单位,设计出的程序简捷、紧凑、结构清晰;
- (2) 既有高级语言的特点(可移植性好),又有低级语言的许多功能(能对硬件操作);

- (3) 提供了丰富的数据类型;
- (4) 语法限制不太严格,程序设计自由度大;
- (5) 生成的目标代码质量高,程序执行速度快。

对 C 语言的上述特点,读者现在也许还不能深刻理解,待学完 C 语言以后再回顾,就会有较深刻的体会。总之,C 语言对程序员要求较高,使用 C 语言编写程序会感到限制少、灵活性强、功能强,可以编写出任何类型的程序。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。同时,C 程序设计语言是 C++ 程序设计语言的基础,C++ 语言和 C 语言在很多方面是兼容的。因此,掌握了 C 语言,再进一步学习 C++,就能以一种熟悉的语法来学习面向对象的语言,从而达到事半功倍的目的。

1.2 简单的 C 程序介绍

为了说明 C 语言源程序的结构和特点,先看例 1.1 所示的一个简单的 C 程序,可以从这个例子中了解组成一个 C 源程序的基本部分和书写格式。

例 1.1 简单的 C 程序。

```
/*example1.1 The first C Program*/  ←——注释
#include<stdio.h>  ←——文件包含命令
main()  ←——主函数
{
    printf("Hello,World!");  ←——语句
}
```

main 是主函数的函数名;printf("Hello,World! \n");是函数调用语句,把要输出的内容送到显示器去显示。每一个 C 源程序都必须有且只能有一个主函数(main 函数)。printf 函数是一个由系统定义的标准函数,可在程序中直接调用。

上述程序输出的结果为

Hello,World!

例 1.2 已知 3 个整数 8、12、6,按公式 $s=a+b \cdot c$ 计算并显示结果。

```
#include<stdio.h>  /*标准输入输出头文件*/
main()
{
    int a,b,c,s;  /*定义四个整型变量*/
    a=8;b=12;c=6;  /*变量赋值*/
    s=a+b*c;  /*算术运算并赋值*/
    printf("s= %d\n",s);  /*输出结果*/
}
```

1.3 C 程序的基本结构及规则

从上述简单的 C 程序中不难看出 C 程序的基本结构及规则,现归纳如下。

(1) 一个完整的 C 程序由若干个函数组成,至少有一个 main 函数,每个函数包括函数说明和函数体两部分,函数体又包含变量定义部分与语句部分。

- (2) 一行可以写多条语句,但语句必须以分号“;”作为结束标志。
- (3) “/*”与“*/”之间的内容构成 C 语言程序的注释部分。
- (4) 用预处理命令 #include 可以包含有关头文件的信息。
- (5) 在 C 语言中区分大小写,如 Main、MAIN、main、maiN 是不相同的。
- (6) 用 { } 括起来的部分,通常表示程序的某一层结构。{ } 一般与该结构语句的第一个字母对齐,并单独占一行。
- (7) 低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写,以便看起来更加清晰,增加程序的可读性。

1.4 C 语言基本语法规成分

1.4.1 C 语言字符集

C 语言字符集如下。

- (1) 英文字母:大小写各 26 个,共计 52 个。
- (2) 阿拉伯数:0~9 共 10 个数字。
- (3) 下划线: _。
- (4) 特殊符号:通常指由 1~2 个符号组成的运算符。
 - 算术运算符: + - * / % ++ --
 - 关系运算符: < > >= <= == !=
 - 逻辑运算符: & & || !
 - 位运算符: & | ~ ^ >> <<
 - 条件运算符: ? :
 - 赋值运算符: =
 - 其他分隔符: () [] { } . , ;

1.4.2 标识符

标识符是用来标识变量名、符号常量名、函数名、类型名、文件名等的有效字符序列。

C 语言规定:标识符只能由字母、数字和下划线 3 种字符组成,且第一个字符必须为字母或下划线。

例如,合法标识符有 _22A,lea_1,avg3,day,ABCde43xyw8;不合法标识符有 M. J. YORK,\$_238,#xy,a*b,8Tea。



注意:在 C 语言中,大小写字母不等效。因此,a 和 A、I 和 i、Sum 和 sum 分别是两个不同的标识符。

1.4.3 关键字

关键字就是具有特定含义的标识符,用户不能用来作自定义标识符。由 ANSI 标准推荐的关键字有 32 个,常用的关键字如下。

- (1) 与数据类型有关的关键字

char int float double signed unsigned short long void struct union

typedef enum sizeof

(2) 与存储类别有关的关键字

auto extern register static

(3) 与程序控制结构有关的关键字

do while for if else switch case default goto continue break return

1.5 C 语言程序的开发环境

为了使计算机能够按照人们的意志进行工作,必须根据问题的要求,编写出相应的程序。所谓程序,就是一组计算机能够识别和执行的有序指令集合。每一条指令使计算机执行特定的操作。用高级语言编写的程序称为“源程序”。从根本上说,计算机只能识别和执行由 0 和 1 组成的二进制的指令,而不能识别和执行高级语言的指令。为了使计算机能够执行高级语言源程序,必须先用一种称为“编译程序”的软件,把源程序翻译成二进制形式的“目标程序”,然后将该目标程序与系统的函数库和其他目标程序连接起来,形成可执行的目标程序。具体 C 程序的计算机执行过程如图 1-1 所示。

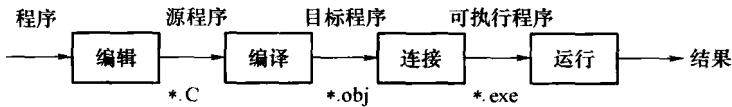


图 1-1 C 程序的计算机执行过程

C 程序的上机过程如图 1-2 所示(详细操作过程见配套的实训教材)。

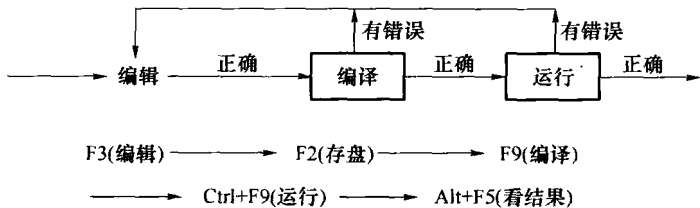


图 1-2 C 程序的上机过程

Turbo C 主菜单画面如图 1-3 所示。

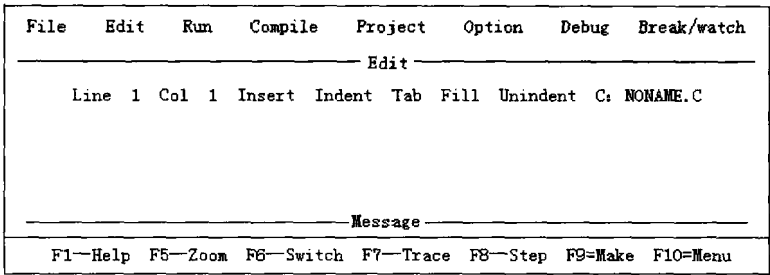


图 1-3 Turbo C 主菜单画面

在编辑过程中,为了操作的方便,应记住 Turbo C 中的一些常用功能键。

- F1:帮助。
- F2:将当前文件存盘。
- F3:装载原有文件或给新文件命名。
- F4:程序运行到光标所在行。
- F5:放大或缩小活动窗口切换。
- F6:开或关活动窗口切换。
- F7:单步运行程序,跟踪进入函数内部运行。
- F8:单步运行程序,不跟踪进入函数内部。
- F9:生成可执行文件。
- F10:菜单与活动窗口相互切换。
- Ctrl+F9:在当前编辑环境下进行编译、连接且运行程序。
- Alt+F5:将窗口切换到DOS环境,用来查看程序运行结果。

习 题 一

- 1.1 写出一个C程序的基本构成。
- 1.2 C程序有哪些主要特点?
- 1.3 函数是C程序的基本单位,有什么好处?
- 1.4 请参照本章例题,试编写一个C程序,输出以下信息:

C program

- 1.5 上机运行本章例1.1和例1.2,熟练使用Turbo C系统的上机方法与步骤。

第 2 章

C 程序设计基础

本章学习目标

- 正确理解常量和变量的概念及使用。
- 了解 C 语言中的基本数据类型。
- 掌握各种数据类型的定义和使用。
- 正确掌握运算符及表达式的功能和使用。

本章要点内容

- 熟悉 C 语言中常量、变量的使用,掌握各数据类型的特点。
- 掌握算术运算符、赋值运算符及自加自减运算符的使用及相应表达式的求解。

本章学前要求

- 对二进制编码有初步了解。

2.1 常量、变量和标识符

2.1.1 常量与标识符

常量又可称为常数,是指在程序运行期间其值不变的量,如 5、3.2、'A'、“北京”等均为常量。常量可分为不同的类型,如 5 为整型常量,3.2 为实型常量,'A'为字符型常量,“北京”为字符串常量。

常量也可以用一个标识符来表示,将其称为符号常量,如例 2.1 所示。

例 2.1 符号常量的使用。

```
#define PI 3.14159                                /*定义符号常量 PI 代表 3.14 */
#include <stdio.h>
main()
{
    float r,c,area;                                /*定义变量*/
    r = 2.5;
    c = 2 * PI * r;                                /*求圆的周长*/
    area = PI * r * r;                              /*求圆的面积*/
    printf("c = %f,area = %f\n",c,area);
```

```
}

```

使用符号常量的好处如下。

- (1) 简化输入,避免出错。用符号代表冗长的数据,输入起来更加方便。
- (2) 含义清楚,如用 PI 代表圆周率。
- (3) 便于修改。如例 2.1 中,若要将圆周率修改为 3.14,只需将第一行改为“# define PI 3.14”即可。

习惯上,为方便与变量名相区别,符号常量名多用大写。

2.1.2 变量

变量是指在程序运行过程中,其值可以改变的量。

每个变量都有一个名字,称为变量名。编程时,用变量名来标识不同的变量。变量被分为不同的类型,在内存中占用不同的存储单元,以便用来存放相应变量的值。

1. 变量使用的规定

C 语言中要求:变量必须“先定义,后使用”。如例 2.1 中的 float r,c,area;定义了 3 个变量。之后,才可以对这些变量进行使用。

2. 变量的定义

定义变量的一般形式为

数据类型 变量名 1,变量名 2,...

如 int a,b;。

数据类型为系统中定义的数据类型名,如 int 代表整型。变量名为一标识符,应符合第 1 章所讲的标识符的命名规则。

2.2 整型数据

2.2.1 整型常量

整型常量即整数。C 语言中,整型常量可用 3 种方式表示。

- (1) 十进制整型常量:直接以数字开头的整型数,如 3、-4。
- (2) 八进制整型常量:以数字 0 开头的整型数,如 012、-035。

八进制是逢 8 进位的数制。八进制数据中的数字只能包括 0~7,019 这样的表示是错误的。012 代表的是八进制的 12,而不是十进制的 12,它换算成十进制数为: $1 \times 8^1 + 2 \times 8^0 = 10$ 。

- (3) 十六进制整型常量:以数字 0 加字母 x 或 X 开头的整型数,如 0x2a、0X39。

十六进制是逢 16 进位的一种数制。十六进制中的数字可为 0~9、a~f(或 A~F)。0X2A 换算成十进制数为 $2 \times 16^1 + 10 \times 16^0 = 42$ 。

如上所述,25、027、0X19 代表的是同一个数,它们都代表的是整型数 25。

2.2.2 整型变量

1. 整型变量的分类

整型变量用来存放整型数据,可分为 3 种类型。