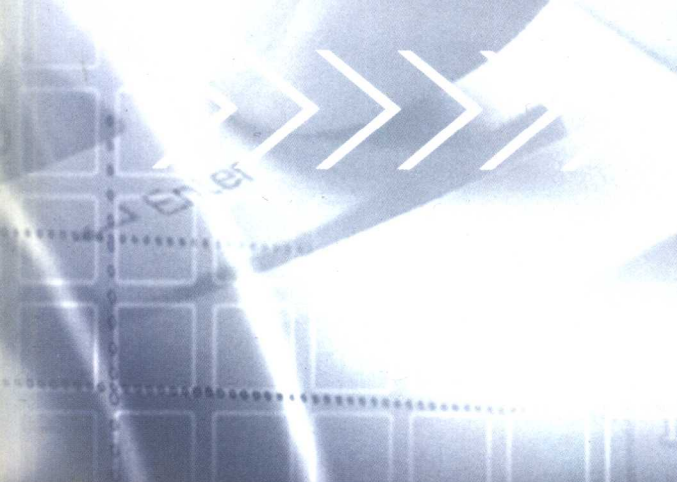




杨超宇 赵光明 编 著
孟祥瑞 李铁锋

C语言程序设计



华东理工大学出版社

EAST CHINA UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS

C 语言程序设计

杨超宇 赵光明 编著
孟祥瑞 李铁锋

 华东理工大学出版社

图书在版编目(CIP)数据

C 语言程序设计 / 杨超宇等编著. —上海:华东理工大学出版社,
2006.8

ISBN 7-5628-1947-5

I. C... II. ①杨... III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 087405 号

C 语言程序设计

.....

编 著 / 杨超宇 赵光明 孟祥瑞 李铁锋

责任编辑 / 李国平

封面设计 / 王晓迪

责任校对 / 张 波

出版发行 / 华东理工大学出版社

地 址:上海市梅陇路 130 号,200237

电 话:(021)64250306(营销部)

传 真:(021)64252707

网 址:www.hdlgpress.com.cn

印 刷 / 江苏通州市印刷总厂有限公司

开 本 / 787mm×1092mm 1/16

印 张 / 18

字 数 / 432 千字

版 次 / 2006 年 8 月第 1 版

印 次 / 2006 年 8 月第 1 次

印 数 / 1—4050 册

书 号 / ISBN 7-5628-1947-5/TP·150

定 价 / 26.00 元

(本书如有印装质量问题,请到出版社储运部调换)

前 言

C 语言是一种通用的程序设计语言,近年来在国内外广泛使用。C 语言的特点是功能丰富、表达能力强、使用灵活方便、应用面广、目标效率高、可移植性好、不受限于任何一种操作系统或机器。它很适合用来编写编译器和操作系统,它也同样适合于编写不同领域中的大多数程序。C 语言是大多数软件开发人员必须掌握的一种语言,各类高等学校中普遍开设了 C 语言课程,全国计算机等级考试二级和三级中也包括 C 语言的考试。

本书内容共分 12 章。第 1 章为 C 语言概述,主要介绍程序设计的基本概念、C 语言的特点、程序的开发流程。第 2 章为数据类型、运算符与表达式,主要介绍 C 语言的基本数据类型、运算符和表达式。第 3 章为数据的输入与输出,主要介绍 C 语言中如何输入输出数据。第 4 章为 C 程序的三种基本结构,主要介绍基本语句、复合语句、控制语句的格式和应用。第 5 章为数组,主要介绍一维数组、二维数组和字符数组及其应用。第 6 章为函数,主要介绍函数定义及调用、变量和函数的作用域。第 7 章为编译预处理,主要介绍宏定义、文件包含和条件编译三种命令格式、功能和用法。第 8 章为指针,主要介绍指针的概念、指针的定义、指针的基本运算以及指针与数组和指针与函数的关系。第 9 章为结构体与共用体,主要介绍 C 语言构造类型的定义及应用。第 10 章为文件,主要介绍文件的基本操作和使用规则。第 11 章为位运算与内存管理,主要介绍各种位运算及其应用,内存管理函数的使用。第 12 章为 C 程序高级应用,主要介绍常用库函数的应用。

本书在编写上力求在体系结构上安排合理、重点突出、难点分散、便于掌握;在语言描述上注重概念清晰、逻辑性强、通俗易懂、便于自学。

本书由杨超宇、赵光明、孟祥瑞、李铁锋编著,张波、程元栋、沈长霞、曹营等也参与了部分编写工作或对编写提供了帮助。

由于作者的水平所限,书中难免存在许多缺点及不足之处,恳请广大读者和同行批评指正。

编者

2006. 7. 20

内 容 提 要

C 语言是一种国内外广泛使用的结构化的程序设计语言,是计算机应用人员应掌握的一种程序设计语言。

本书共分 12 章,全面系统地介绍了程序设计的基本概念,基本数据类型,运算符和表达式,C 语句和结构化程序设计,数组,函数,编译预处理,指针,结构体,共用体,位运算,内存管理,文件,库函数及其应用。

本书内容安排合理,讲解通俗易懂,对重点难点做了十分系统、清楚和详细的阐述,并辅以大量的典型例题,使广大读者能够通过学习本书掌握 C 语言的基本知识,并灵活运用所学知识编写应用程序。

本书可作为高等院校计算机专业或相关专业的教材,也可作为全国计算机等级考试的培训教程。

目 录

第 1 章 C 语言概述

1.1 C 语言的历史	1
1.2 C 语言的特点	2
1.3 C 程序入门	3
1.4 程序开发流程	5
1.4.1 程序开发前的准备工作	5
1.4.2 程序的开发	6
1.4.3 C 语言的运行环境	7
习题 1	17
实验 1	17

第 2 章 数据类型、运算符与表达式

2.1 常量与变量	19
2.1.1 常量与符号常量	19
2.1.2 变量	21
2.2 基本标识符	22
2.2.1 关键字	22
2.2.2 预定义标识符	23
2.2.3 用户定义标识符	23
2.3 基本数据类型	24
2.3.1 C 语言的数据类型	24
2.3.2 整型数据	25
2.3.3 实型数据	30
2.3.4 字符型数据	32
2.4 运算符与表达式	35
2.4.1 C 语言的运算符与表达式	35
2.4.2 算术运算符与算术表达式	36
2.4.3 赋值运算符与赋值表达式	38
2.4.4 关系运算符与关系表达式	40
2.4.5 逻辑运算符与逻辑表达式	41
2.4.6 条件运算符与条件表达式	45
2.4.7 逗号运算符与逗号表达式	47

2.4.8 复合赋值运算符与复合赋值表达式	48
2.4.9 运算符的优先级	48
习题 2	49

第 3 章 数据的输入与输出

3.1 C 语言的数据输入输出的实现	50
3.2 格式输入输出函数	51
3.2.1 格式输出函数 printf	51
3.2.2 格式输入函数 scanf	57
3.3 字符的输入输出	59
3.3.1 字符输出函数 putchar	59
3.3.2 字符输入函数 getchar	60
3.4 转义字符	61
习题 3	62
实验 3	63

第 4 章 C 程序的三种基本结构

4.1 C 程序的三种基本结构	64
4.2 C 语句	65
4.3 选择结构程序设计	67
4.3.1 if 语句	67
4.3.2 switch 语句	72
4.4 循环结构	74
4.4.1 goto 语句	74
4.4.2 while 语句	75
4.4.3 do-while 语句	76
4.4.4 for 语句	78
4.4.5 嵌套循环	80
4.4.6 break 与 continue 语句	81
习题 4	83
实验 4.1	83
实验 4.2	84

第 5 章 数组

5.1 一维数组	85
5.1.1 一维数组的定义	85
5.1.2 一维数组元素的引用	86
5.1.3 一维数组的初始化	87
5.2 二维数组	88

5.2.1 二维数组的定义	88
5.2.2 二维数组的引用	89
5.2.3 二维数组的初始化	89
5.2.4 二维数组程序举例	90
5.3 字符数组	91
5.3.1 字符数组的定义	91
5.3.2 字符数组的引用	92
5.3.3 字符数组的初始化	92
5.3.4 字符串和字符串结束标志	93
5.3.5 字符数组的输入输出	93
5.3.6 字符数组程序举例	96
习题 5	97
实验 5	97

第 6 章 函数

6.1 函数的定义	99
6.2 函数的调用	102
6.2.1 函数调用的一般形式	102
6.2.2 函数的调用方式	102
6.2.3 函数的嵌套调用	103
6.2.4 函数的递归调用	104
6.3 函数的参数与返回值	108
6.3.1 形式参数与实际参数	108
6.3.2 函数的返回值	109
6.3.3 函数原型与函数的声明	110
6.4 变量的作用域与存储类别	111
6.4.1 作用域和生存期	111
6.4.2 局部变量和全局变量	112
6.4.3 变量的存储类别	115
6.4.4 变量的声明与定义	120
6.5 内部函数与外部函数	120
6.5.1 内部函数	120
6.5.2 外部函数	121
习题 6	122
实验 6	122

第 7 章 编译预处理

7.1 概述	124
7.2 宏定义	124

7.2.1 不带参数的宏定义	124
7.2.2 带参数的宏定义	127
7.3 文件包含	131
7.4 条件编译	132
7.4.1 #ifdef 命令	132
7.4.2 #ifndef 命令	133
7.4.3 #if 命令	133
习题 7	134

第 8 章 指针

8.1 指针概述	135
8.1.1 指针的概念	135
8.1.2 指针变量的定义	136
8.1.3 指针变量的引用	137
8.2 数组与指针	140
8.2.1 通过指针引用数组元素	140
8.2.2 作为指针的数组名	141
8.2.3 将数组传递给函数	143
8.2.4 多维数组与指针	149
8.3 字符串与指针	154
8.3.1 字符串与字符指针	154
8.3.2 字符指针作为函数参数	156
8.3.3 对使用字符指针变量和字符数组的讨论	159
8.4 函数与指针	161
8.4.1 指针作为函数参数	161
8.4.2 指向函数的指针	162
8.4.3 返回指针的函数	163
8.5 指针数组和指向指针的指针	165
8.5.1 指针数组的概念	165
8.5.2 指向指针的指针	168
8.5.3 带参数的 main 函数	170
8.6 有关指针的数据类型的小结	171
习题 8	172
实验 8	173

第 9 章 结构体与共用体

9.1 结构体概述	174
9.1.1 定义结构体类型	174
9.1.2 结构体类型变量的定义及初始化	175

9.1.3 结构体成员的引用	178
9.2 结构体与函数	181
9.2.1 结构体类型变量作为函数参数	181
9.2.2 结构体类型变量作为函数返回值	181
9.3 结构体数组	182
9.4 指向结构体的指针	185
9.4.1 用指针作为结构的成员	185
9.4.2 指向结构体的指针	185
9.4.3 指向结构体数组的指针	187
9.4.4 将指向结构体的指针作为函数参数	188
9.5 链表	189
9.5.1 链表概述	189
9.5.2 动态链表	190
9.6 共用体	194
9.6.1 定义共用体类型	194
9.6.2 共用体类型变量的定义	195
9.6.3 共用体成员的引用	195
9.7 枚举类型	196
9.7.1 枚举类型和枚举型变量的定义	196
9.7.2 枚举型变量的使用	197
9.8 typedef 类型定义	199
习题 9	200
实验 9	200

第 10 章 文件

10.1 文件概述	201
10.2 文件指针	202
10.3 文件的打开与关闭	203
10.3.1 文件的打开	203
10.3.2 文件的关闭	205
10.4 文件的读写	205
10.4.1 格式化文件的输入输出	205
10.4.2 字符输入输出	207
10.4.3 直接文件输入输出	213
10.5 文件的定位	214
10.5.1 rewind 函数	214
10.5.2 ftell 函数	214
10.5.3 fseek 函数和随机读写	215
10.6 其他文件相关函数	217

10.6.1 检测文件结束	217
10.6.2 删除文件	217
10.6.3 文件重命名	218
10.6.4 错误处理	219
习题 10	220
实验 10	220

第 11 章 位运算与内存管理

11.1 位运算	221
11.1.1 按位逻辑运算符	221
11.1.2 求补运算符	224
11.1.3 移位运算符	224
11.1.4 位段	225
11.2 分配与释放内存	228
11.2.1 malloc() 函数	228
11.2.2 calloc() 函数	229
11.2.3 realloc() 函数	229
11.2.4 free() 函数	230
11.3 操纵内存	231
11.3.1 memset() 函数	231
11.3.2 memcpy() 函数	232
11.3.3 memmove() 函数	232
习题 11	233

第 12 章 C 程序高级应用

12.1 数学函数	234
12.1.1 三角函数	234
12.1.2 指数函数和对数函数	235
12.1.3 其他数学函数	236
12.2 字符串函数	237
12.2.1 字符串处理函数	237
12.2.2 字符串和数值转换函数	240
12.3 处理时间与日期	240
12.3.1 时间与日期的表示	240
12.3.2 时间与日期函数	241
12.4 文本的屏幕输出	243
12.4.1 文本方式的控制	243
12.4.2 窗口设置	245
12.4.3 窗口字符的输入输出函数	246

12.4.4 其他处理函数	246
12.5 图形程序设计	248
12.5.1 图形系统的初始化	248
12.5.2 基本图形函数	251
12.5.3 颜色控制函数	254
12.5.4 封闭图形的填充函数	258
12.5.5 图形窗口操作函数和屏幕操作函数	262
12.5.6 图形方式下的文本输出函数	264
习题 12	268
 附录 I 常用字符与 ASCII 代码对照表	 270
 附录 II C 语言关键字	 271
 附录 III 运算符的优先级和结合性	 272
 参考文献	 273

第 1 章 C 语言概述

1.1 C 语言的历史

C 语言是国际上广泛流行的计算机高级语言。它既可用于编写系统软件,也可用来编写应用软件。

C 语言是一种通用的程序设计语言。它同 UNIX 系统之间具有非常密切的联系——C 语言是在 UNIX 系统上开发的,并且,无论是 UNIX 系统本身还是其上运行的大部分程序,都是用 C 语言编写的。但是,C 语言并不受限于任何一种操作系统或机器。由于它很适合用来编写编译器和操作系统,因此被称为“系统编程语言”,但它同样适合于编写不同领域中的大多数程序。

C 语言是在 B 语言的基础上发展起来的。1967 年英国剑桥大学的 Martin Richards 开发出了 BCPL(basic combined programming language)语言。1970 年美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础,设计出了很简单的而且很接近硬件的 B 语言(取 BCPL 的第一个字母),并用 B 语言写了第一个 UNIX 操作系统,在 PDP-7 上实现。但 B 语言过于简单,功能有限。1972 年至 1973 年间,贝尔实验室的 D.M. Ritchie 在 B 语言的基础上设计出了 C 语言(取 BCPL 的第二个字母)。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。1973 年,K. Thompson 和 D.M. Ritchie 两人合作把 UNIX 的 90% 以上用 C 改写,即 UNIX 第 5 版。原来的 UNIX 操作系统是 1969 年由美国的贝尔实验室的 K. Thompson 和 D.M. Ritchie 开发成功的,是用汇编语言写的。后来,对 C 语言多次做了改进,但主要还是在贝尔实验室内部使用。直到 1975 年 UNIX 第 6 版公布后,C 语言的突出优点才引起人们的普遍注意。1977 年出现了不依赖于具体机器的 C 语言编译文本《可移植 C 语言编译程序》,使 C 语言移植到其他机器时所需做的工作大大简化,这也推动了 UNIX 操作系统迅速地在各种机器上实现。例如 VAX、AT&T 等计算机系统都相继开发了 UNIX。随着 UNIX 的日益广泛使用,C 语言也迅速得到推广。C 语言和 UNIX 可以说是一对孪生兄弟,在发展过程中相辅相成。1978 年以后,C 语言已先后移植到大、中、小、微型机上,已独立于 UNIX 和 PDP 了。

1978 年由美国电话电报公司(AT&T)贝尔实验室正式发表了 C 语言。同时由 B. W. Kernighan 和 D.M. Ritchie 合著了著名的“THE C PROGRAMMING LANGUAGE”一书。通常简称为《K&R》,也有人称之为《K&R》标准。但是,在《K&R》中并没有定义一个完整的标准 C 语言。

1983 年,美国国家标准协会(American National Standards Institute)成立了一个委员会,以制定一个现代的、全面的 C 语言规范,最后的结果就是 1988 年完成的 ANSI 标准,即“ANSI C”。该标准的大部分特性已被当前的编译器所支持。该标准的第二个重要贡献是为 C 语言定义了一个函数库,它描述了诸如访问操作系统(如读写文件)、格式化输入/输出、内

存分配和字符串操作等类似的很多函数。该标准还定义了一系列的标准头文件, 它们为访问函数声明和数据类型声明提供了统一的方法。

1.2 C 语言的特点

(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。语言组成精炼、简洁, 而且使用方便、灵活(见表 1.1)。

表 1.1

auto	break	case	char	const	continue	default
do	double	else	enum	extern	float	for
goto	if	int	long	register	return	short
signed	static	sizeof	struct	switch	typedef	union
unsigned	void	volatile	while			

(2) 运算符丰富。

C 语言共有 34 种运算符。C 语言不仅包括其他高级语言中具有算术运算符、关系运算符、逻辑运算符等, 还把括号、赋值、强制类型转换等都作为运算符处理。从而使 C 语言的运算类型极为丰富, 表达式类型多样化, 可以实现在其他高级语言中难以实现的运算。

(3) 数据结构丰富。

C 语言的数据结构类型有: 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, 能用来实现各种复杂的数据结构(如: 链表、树、栈等)的运算。

(4) 具有结构化的控制语句。

C 语言为实现结构良好的程序提供了基本的控制流结构: 语句组、条件判断(if — else)、多路选择(switch)、终止测试在顶部的循环(while、for)、终止测试在底部的循环(do)、提前跳出循环(break)等。

(5) 具有编译预处理功能。

编译的预处理阶段将对程序文本进行宏替换、包含其他源文件以及进行条件编译。

(6) C 语言允许直接访问物理地址。

C 语言是一种相对“低级”的语言。它意味着 C 语言可以处理大部分计算机里的对象, 包括地址。

(7) 生成目标代码质量高, 程序执行效率高。

一般只比汇编程序生成的目标代码效率低 10%~20%。

(8) 书写灵活。

书写格式比较自由, 一行可以书写多个语句, 一个语句可以分布在连续的若干行。

(9) 可移植性好。

C 语言能够运行在大部分的计算机上, 它同具体的机器结构无关。可以用 C 语言编写出可移植的程序, 即可以不加修改地运行于多种硬件上。

1.3 C 程序入门

本节通过几个简单的程序,介绍C语言程序的一些基本构成和格式。

【例 1.1】 输出 "hello, world" 字符。

```
#include<stdio.h>

main()
{
    printf("hello, world!\n");
}
```

程序运行结果:

hello, world!

程序分析:

(1) main 是主函数的函数名,表示这是一个主函数。每一个C源程序都必须有,且只能有一个主函数(main 函数)。

(2) 函数调用语句,printf 函数的功能是把要输出的内容送到显示器去显示。printf 函数是C语言的一个标准输出函数,它不需定义,可直接调用。

(3) 编译预处理命令#include。

程序的第1行:#include <stdio.h> 是一个编译预处理命令。编译预处理命令必须以“#”开头,末尾不能加“;”。该编译预处理命令称为文件包含命令,功能是把一个文本文件的内容包含到该命令处。<stdio.h> 是命令的参数,它给出了所包含文件的文件名,即stdio.h。stdio.h 是系统提供的头文件名,该文件中包含有关输入输出函数的信息。在该程序中,由于调用标准输出函数 printf,所以需要在程序的开始处将头文件 stdio.h 包含进来。

【例 1.2】 设计一个程序,求半径为 2 的圆的面积。

```
#include<stdio.h>

main()
{
    int radius;
    float area;
    /*用 area 存放面积的值*/
    radius = 2;
    /*用 radius 存放半径的值*/
    area = 3.14159*radius*radius;
    printf("radius = %d, area = %f\n", radius, area);
}
```

程序运行结果:

radius = 2, area = 12.566360

程序分析:

(1) 函数及其定义。

每个 C 语言程序都由一个或多个函数组成,函数是具有特定功能的程序模块。函数定义就是用特定的格式对函数功能进行描述。函数定义是编写程序必不可少的步骤。

该程序包含一个函数定义。main 是函数名,系统将通过这个名称调用该函数。函数名后的圆括号“()”不能省略,它是识别函数的重要标志。

函数体以左花括号“{”开始以右花括号“}”结束,其间可以有说明部分和执行部分。

每个程序至少包含 1 个函数,这个不可缺少的函数称为主函数,约定的函数名是 main。在 C 语言程序中,字母的大小写是有区分的。因此 main, Main, MAIN 等是不同的名称,作为主函数的函数名只能是 main,不能混用。

(2) 变量及其赋值。

变量定义的基本格式如下:

类型符 变量名表;

变量是给用于存储信息的内存单元赋予名称。在程序执行期间,程序使用变量来存储各种信息。在 C 语言中,使用变量之前必须定义它。

在该程序中定义了 1 个整型变量、1 个单精度实型变量,变量名分别是 radius 和 area。

在 C 语言中可以通过赋值运算符“=”给变量赋值。程序中把 radius 赋值为 2,然后用 3.14159 乘以 radius 的平方,计算出结果后,赋值给 area。

(3) 程序的注释。

程序注释以“/*”开始,以“*/”结束。“/*”和“*/”必须成对出现,“/”和“*”之间不允许有空格。注释可以用西文,也可以用中文。注释的长度可以是一行或者几行。注释可以出现在程序中任意合适的地方。注释对程序的运行不起作用,可以用来说明程序段的功能,以帮助人们阅读程序。因此一个好的程序应该有必要的注释。

最新的 ANSI C 标准新增了使用单行注释的功能。

单行注释使用双斜杠来标识注释。

```
//This entire line is a comment
```

两个斜杠指出该行中余下的内容为注释。ANSI C - 99 标准将这种特性加入了 C 语言中。

【例 1.3】

```
#include<stdio.h>
int max(int a,int b);
/* 函数说明*/
/* 主函数*/
main()
{
    int x,y,z;
    /* 变量说明*/
    int max(int a,int b);
    /* 函数说明*/
    printf("input two numbers:\n");
```

```
scanf("%d%d",&x,&y);
/*输入 x,y 值*/
z = max(x,y);
/*调用 max 函数*/
printf("maxnum=%d",z);
/*输出*/
}
/*定义 max 函数*/
int max(int a,int b)
{
    if(a>b)return a;
    else return b;
    /*把结果返回主调函数*/
}
```

例 1.3 中程序的功能是由用户输入两个整数,程序执行后输出其中较大的数。

(1) 本程序由两个函数组成,主函数和 max 函数。函数之间是并列关系。可从主函数中调用其他函数。max 函数的功能是比较两个数,然后把较大的数返回给主函数。max 函数是一个用户自定义函数,在主函数中调用了 max 函数。

(2) 上例中程序的执行过程是:首先在屏幕上显示提示串,请用户输入两个数,回车后由 scanf 函数语句接收这两个数送入变量 x,y 中,然后调用 max 函数,并把 x,y 的值传送给 max 函数的参数 a,b。在 max 函数中比较 a,b 的大小,把大者返回给主函数的变量 z,最后在屏幕上输出 z 的值。

1.4 程序开发流程

1.4.1 程序开发前的准备工作

解决问题时,应采取一些特定的步骤。首先必须定义问题。如果不知道问题是什么,将无法找到解决方案。知道问题是什么之后便可以设计解决它的方案。有了方案后,你通常能够实现它。方案实现后,必须对结果进行测试,以确定问题是否得到解决。这种逻辑也适用于包含编程在内的许多其他领域。

创建 C 语言程序(或其他语言的计算机程序)时,应遵循下面类似的步骤:

- (1) 确定程序的目标;
- (2) 确定要使用什么样的方法来编写程序;
- (3) 创建程序,以解决问题;
- (4) 运行程序,以查看其结果。

目标可能是编写一个字处理器或数据库程序。一个更为简单的目标是将你的姓名显示到屏幕上。如果没有目标,将无法编写程序,因此必须首先完成第 1 步。

第 2 步是确定要使用什么样的方法来编写程序。需要使用一个计算机程序来解决问题