



普通高等教育“十一五”国家级规划教材
新世纪高职高专实用规划教材
计算机系列

C语言 程序设计

(第2版)



高福成 主 编

赠送
电子课件



清华大学出版社

普通高等教育“十一五”国家级规划教材
新世纪高职高专实用规划教材 计算机系列

C 语言程序设计(第 2 版)

高福成 主编

清华大学出版社
北 京

内 容 简 介

本书以程序设计为主线,以程序设计的需要带动语言知识的学习,系统介绍了C语言程序设计的基本知识、基本数据类型和数据运算、程序控制结构、数组、指针、函数、结构和联合、文件和编译预处理等,并通过丰富的程序设计实例,加强程序设计思维方法和实际编程的训练。每章都安排了上机指导和大量的练习题,帮助读者提高实际操作水平以及阅读程序、调试程序和编制程序的实际能力,并自我检测C语言知识的掌握程度。

本书可作为高职院校“C语言程序设计”课程的教材,也可作为全国计算机等级考试(C语言)培训班教材或供从事C语言程序开发的工程技术人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C语言程序设计/高福成主编. —2版. —北京:清华大学出版社,2009.5

(普通高等教育“十一五”国家级规划教材)

(新世纪高职高专实用规划教材 计算机系列)

ISBN 978-7-302-20109-0

I. C… II. 高… 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(2009)第067655号

责任编辑:朱 颖

封面设计:山鹰工作室

责任校对:王 晖

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦A座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京国马印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:23.5 字 数:562千字

版 次:2009年5月第2版 印 次:2009年5月第1次印刷

印 数:1~4000

定 价:35.00元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:028446-01

前 言

C 语言曾以高效、灵活、功能丰富、表达力强和移植性好等特点,在计算机语言中声誉卓著,并成为此后程序设计语言进化的模板。在面向对象程序设计已经成为软件开发主流的今天,C 语言在程序开发中逐步被其后代所取代已经是不争的事实,但在开发系统程序(如操作系统和嵌入式系统等)和底层应用程序(如接口程序、通信和自动控制等)时,仍然是非 C 语言莫属,C 语言的编程思想依然是一棵常青树。目前,仍有相当多的学习 C 语言的人群。

为使 C 语言的教学能够与时俱进,本书较之第一版有以下几个方面的修订。

(1) 随着 C 语言新标准 C99 的颁布,原先沿用的 ANSI C 标准,已引起许多热衷于学习 C 程序设计的人群的不满,因此,更新教材内容、使用更新的开发环境,已成为 C 语言教学的一个迫切需要解决的问题,也是培养优秀 C 编程人员和设计人员的当务之急。为此,本书使用的开发环境由 Turbo C 2.0 改为 VC++ 6.0,这使得 C99 标准中的部分改进与扩充功能能得到支持。

(2) 充分体现 C99 标准中的新功能,删除那些副作用大并已能够被替换的陈旧元素;删除原教材中图形与图像处理一章,对全书内容进行精炼和压缩。

(3) 书中的例题和习题全部重写,以适应 VC++ 6.0 的开发环境。

本书仍坚持以程序设计为中心,语言介绍服从程序设计的需要,同时强化实践、引导探索,在阅读程序、编制程序和设计程序三个层次上将理论与实例设计紧密结合,同时在小机机上实习和中型设计性试验相结合方面进行了有益的探索:

(1) 以提高程序设计能力为目标,以程序设计方法学为依据,以综合、创新设计为重点,本着把程序设计逻辑思维方法作为一种科学方法来训练,把语言知识作为一种工具和技能来培养的理念,根据程序设计的需要安排语言知识的介绍,紧扣程序设计实例,重点介绍程序设计的步骤和方法,引导学生理解理论、掌握方法,在编程实践中熟悉语言,提高设计能力和创新能力。

(2) 对 C 语言中数组、指针等不易理解的概念,有创新性的阐述,使难点更易理解和掌握。

(3) 强化实践环节,各章中典型例题的筛选和编排围绕三个层次进行:一是阅读程序的能力,二是根据指定算法编制程序的能力,三是针对实际问题自行设计算法和程序的能力。特别是第二种能力的培养,既是高职高专生的迫切需要,也是本书的重点目标。

(4) 每一章都附有上机指导和大量的习题,提供了大量的上机实习。这些题目的规模从小到大、解决问题的方法从易到难、提供的指导从具体到抽象,以培养学生从简单的模仿到举一反三,使其从感性认识升华到理性高度,真正在方法论上有所探索。

(5) 书中附录部分给出了 VC++ 6.0 操作指导及各章上机练习题和习题的参考答案,

以方便读者自学。

本书由高福成, 梁静毅, 王桂荣主编, 参加编写的有潘旭华、赵玉刚、李军、曲立敏、王梦倩、边佳、毛莹、高菡、杨玉兰等老师。由于作者水平有限, 疏漏和错误之处难以避免, 恳请使用本书的老师和读者提出宝贵意见。

在本书编写及出版过程中, 得到了天津大学边莫英教授的悉心指导, 清华大学出版社朱颖编辑付出了辛勤的劳动, 在此一并表示感谢。

联系方式: gaofc126@126.com

编 者

目 录

第 1 章 C 程序设计的初步知识..... 1	
1.1 C 语言的发展历史..... 1	
1.2 C 程序的结构特点..... 3	
1.2.1 一个简单的 C 程序..... 3	
1.2.2 C 程序的结构特点..... 3	
1.3 C 程序的书写风格..... 7	
1.4 简单的 C 程序设计..... 7	
1.4.1 赋值语句的简单使用..... 7	
1.4.2 格式输入输出函数的简单使用..... 7	
1.4.3 库函数和标题文件..... 8	
1.4.4 简单程序设计举例..... 9	
1.5 C 程序的开发过程..... 10	
1.6 Visual C++ 6.0 集成开发环境的使用..... 12	
1.7 上机指导..... 12	
习题..... 17	
第 2 章 基本数据类型和数据运算..... 20	
2.1 基本数据类型..... 20	
2.2 常量及其类型..... 22	
2.3 变量及其类型..... 26	
2.4 符号常数..... 29	
2.5 算术运算..... 31	
2.5.1 算术运算符..... 31	
2.5.2 算术表达式..... 32	
2.6 赋值运算..... 36	
2.7 逗号运算..... 39	
2.8 关系运算和逻辑运算..... 41	
2.9 测试数据长度运算..... 45	
2.10 位操作..... 46	
2.10.1 位逻辑运算..... 46	
2.10.2 移位运算..... 48	
2.10.3 位操作运算符组成的复合赋值运算符..... 49	
2.11 常用数学函数..... 50	
2.12 不同类型数据的输入输出..... 52	
2.12.1 printf()函数..... 52	
2.12.2 scanf()函数..... 55	
2.12.3 单字符输入输出函数 (getchar()和 putchar())..... 58	
2.12.4 单字符输入函数 (getche()和 getch())..... 60	
2.13 上机指导..... 61	
习题..... 63	
第 3 章 程序流程控制..... 69	
3.1 结构化程序设计的概念..... 69	
3.2 选择结构程序设计..... 70	
3.2.1 if 语句..... 70	
3.2.2 条件表达式..... 76	
3.2.3 switch 语句..... 78	
3.3 循环结构程序设计..... 82	
3.3.1 for 循环..... 82	
3.3.2 while 循环..... 86	
3.3.3 do-while 循环..... 89	
3.3.4 循环的嵌套..... 91	
3.4 转移控制语句..... 93	
3.4.1 break 语句..... 93	
3.4.2 continue 语句在循环结构中的作用..... 95	
3.4.3 goto 语句和标号..... 96	
3.5 上机指导..... 98	
习题..... 101	
第 4 章 数组和字符串..... 112	
4.1 数组的概念..... 112	

4.2 数组的定义和初始化.....113	6.8 函数的存储类型对函数调用的 影响 210
4.2.1 数组的定义和存储结构.....113	6.9 函数的递归调用..... 210
4.2.2 数组的初始化.....115	6.10 内联函数..... 212
4.2.3 通过初始化定义隐含 尺寸数组117	6.11 上机指导..... 214
4.3 数组的基本操作.....119	习题 218
4.4 数组在数值计算中的应用.....124	第 7 章 复合数据类型和类型定义 229
4.5 数组在字符串处理中的应用.....130	7.1 结构类型..... 229
4.5.1 字符串处理函数.....130	7.1.1 结构的定义及内存分配 模式..... 229
4.5.2 字符型数组在字符串 处理中的应用.....133	7.1.2 结构成员的访问..... 233
4.6 上机指导136	7.1.3 结构变量、结构数组的赋值、 输入和输出..... 236
习题140	7.1.4 结构类型的数据在函数间的 传递..... 237
第 5 章 指针148	7.1.5 用递归结构处理链表..... 242
5.1 指针的概念148	7.2 联合类型..... 245
5.2 指针的运算153	7.3 枚举类型..... 248
5.3 用指针访问变量.....157	7.4 类型定义..... 250
5.4 用指针访问数组.....158	7.5 上机指导..... 253
5.4.1 用指针访问一维数组.....158	习题 257
5.4.2 用指针访问二维数组.....159	第 8 章 文件 266
5.5 用指针处理字符串.....164	8.1 文件的概述..... 266
5.6 二级指针169	8.2 文件的打开与关闭..... 268
5.7 用指针进行内存动态分配.....171	8.3 文件的读写操作..... 271
5.8 上机指导173	8.3.1 文本文件读写函数..... 271
习题176	8.3.2 二进制文件读写函数..... 276
第 6 章 函数184	8.4 文件检测函数..... 279
6.1 C 程序的模块结构.....184	8.5 文件的顺序存取和随机存取..... 281
6.2 函数的定义和调用.....186	8.6 上机指导..... 290
6.3 用函数返回值传递数据.....190	习题 293
6.4 用参数传递数据.....193	第 9 章 编译预处理 302
6.4.1 参数传递的过程和特点.....193	9.1 编译预处理命令..... 302
6.4.2 变量的传递194	9.2 宏定义..... 302
6.4.3 数组(含字符串)的传递197	9.3 文件包含..... 307
6.5 用全局变量传递数据.....202	9.4 条件编译..... 309
6.6 用函数指针调用函数.....203	
6.7 变量的存储类型对函数调用的 影响204	

习题	312	附录 4 上机指导参考答案	324
附录 1 C 语言运算符集	317	附录 5 习题参考答案	332
附录 2 ASCII 代码表	318	附录 6 VC++ 6.0 操作指导	354
附录 3 常用库函数及其标题文件	319		

第 1 章 C 程序设计的初步知识

教学提示: 本章主要介绍 C 语言和 C 程序设计的初步知识, 这些知识在后续章节中还会出现, 学习完这些内容后, 读者对 C 程序将会有有一个初步的了解。

教学目标: 熟悉 C 程序的结构特点和书写风格, 掌握 C 语言关键字和标识符的命名方法, 了解 C 编译系统提供的头文件的功能, 学会用赋值语句和格式输入/输出函数编制简单的 C 程序, 掌握 Visual C++ 6.0 集成环境的使用。

1.1 C 语言的发展历史

C 语言的祖先是 ALGOL60(Algorithm Language)。ALGOL60 是 1960 年由国际计算机委员会设计的一种面向过程的结构化程序设计语言, 用它编写的程序具有可读性和可移植性好的特点。但是, 它不能直接对硬件进行操作, 不宜用来编写系统程序。系统程序主要用汇编语言编写, 而汇编语言是面向机器的, 用它编写的程序可读性和可移植性都比较差。为此, 人们开始考虑设计一种集高级语言和低级语言功能于一身的语言, 以便使用它来编写可读性和可移植性都比较好的系统程序。

1963 年, 英国的剑桥大学和伦敦大学首先将 ALGOL60 发展成 CPL(Combined Programming Language), 该语言已比较接近于硬件, 但规模较大, 难以实用。

1967 年, 剑桥大学的 Martin Richards 将 CPL 改制成 BCPL(Basic Combined Programming Language)。BCPL 比 CPL 大为简化, 既具有结构化程序设计语言的特点, 也能直接处理与硬件相关的数据, 被软件人员用作系统程序的描述语言。

1970 年, 美国贝尔实验室的 Ken Thompson 将 BCPL 修改成 B 语言(Boiling CPL down to its basic good feature), 并用 B 语言开发了第一个高级语言 UNIX 操作系统, 在 DEC 公司的 PDP-7 小型机上运行。

1972 年, Dennis M. Richie 将 B 语言修改设计成 C 语言。C 语言既保持了 BCPL 和 B 语言精炼和接近于硬件的特点, 也克服了它们过于简单、数据无类型等缺点。1973 年, Ken Thompson 和 Dennis M. Richie 又合作将 1969 年用汇编语言编写的 UNIX 操作系统改用 C 语言编写, 其中 C 语言代码占 90% 以上, 只保留了少量汇编语言代码。这样就使得 UNIX 操作系统向其他类型的机器上移植变得相当简单。到 70 年代中期, UNIX 操作系统和 C 语言作为软件设计的良好工具而开始风靡世界。

1978 年, 以 UNIX 第 7 版中的 C 编译程序为基础, Brain W. Kernighan 和 Dennis M. Richie 合著了影响深远的名著《The C Programming Language》。这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础, 称为 K&R C 语言或传统 C, 并作为 C 编译器所支持的最基本的 C 语言部分。其后的十几年中, 适用于不同机种和不同操作系统的 C 编译系统相继

问世,从而把 C 语言的应用推向了更加广泛普及的阶段。1983 年美国国家标准局 ANSI 制定了 C 语言标准,称为 ANSI C 或现代 C。1988 年, K&R 修改了他们的经典著作《The C Programming Language》,按 ANSI C 标准重新编写了该书。1989 年, ANSI C 标准(ANSI X3.159-1989)正式颁布。C 标准化的一个目的是扩展 K&R C。这个标准包括了一些新的特性,也使用新的函数库标准。后来, ANSI C 标准被 ISO(国际标准化组织)采纳成为 ISO 9899,简称 C89,其第一个版本文件在 1990 年出版。

在 ANSI 标准化后, C 语言的标准在一段相当长的时间内都保持不变。虽然 Normative Amendment1 在 1995 年曾经开发了一个新的 C 语言版本,简称为 C95,但这个版本很少为人所知。到了 90 年代,随着 C++标准在不断更新, C 语言标准又得到改进,这就是 ISO/IEC 9899:1999(1999 年出版),俗称 C99,该标准于 2000 年 3 月被 ANSI 采用。

但是各个公司对 C99 的支持所表现出来的兴趣不同。当 GCC 和其他一些商业编译器支持 C99 的大部分特性的时候,微软和 Borland 却似乎对此不感兴趣,例如, VC++ 6.0 就不完全支持 C99。

C 语言是一种面向过程的语言,意思是用 C 语言编程时,必须按照算法的实现过程逐条语句编写,通知计算机一步一步怎么做。进入 80 年代后,面向对象的程序设计概念日益普及。所谓面向对象,是通过类和对象把程序所涉及的数据结构和对它施行的操作有机地组织成模块,对于数据和对数据的处理细节进行最大限度的封装,从而使开发出来的软件易重用、易修改、易测试、易维护、易扩充。正如其他传统的程序设计语言都在发展自己面向对象的新版本一样, C 语言也在发展的同时,朝着支持面向对象程序设计(OOP—Object Oriented programming)的方向迈出了步伐。1986 年,美国 AT&T 的贝尔研究所的 Bjarne Stroustrup 推出了 C 语言的超集 C++语言,也叫“带类的 C”。

随着信息时代的到来,由于设计 Internet 上的 Web 浏览器的需要,1994 年出现了 Java 语言,它不仅支持 OOP,而且具有软硬件平台无关性的特点,适合于进行网络开发。Java 脱胎于 C++,被称为 C++的衍生语言。2000 年, Microsoft 推出了 Microsoft Visual Studio.NET,它是一个具有公共语言子集的开发平台,实现了多种语言及其类库的无缝集成,使应用程序的开发更容易、更简单。C#是专为这一平台推出的全新语言,它也派生于 C 和 C++,并具有语法简洁、面向对象、与 Web 紧密结合、卓越的安全性能、灵活性和兼容性俱佳等特点,成为 .NET 平台一流的网络编程工具。

从面向过程的 C 到面向对象的 C++,继而是 Windows 平台上可视化的 C++开发工具 Visual C++,直到网络平台上面向对象的 Java 和 C#, C 语言在程序开发中逐步被其后代所取代已经是不争的事实,但在开发系统程序(如操作系统和嵌入式系统等)和底层应用程序(如接口程序、通信和自动控制等)时,仍然是非 C 莫属, C 语言的编程思想依然是一棵常青树。

1.2 C 程序的结构特点

本节通过一个程序实例来说明 C 程序的结构特点。

1.2.1 一个简单的 C 程序

下面是一个简单的 C 程序，它只有一个 `main()` 函数。

【例 1.1】 编制程序根据给定的半径 `r`，计算圆的面积 `a` 和周长 `c`。

先定义三个变量分别表示半径、面积和周长，然后指定半径 `r` 的值，由计算公式计算面积和周长并存放在变量 `a` 和 `c` 中，最后输出半径 `r`、面积 `a` 和周长 `c` 的值。程序如下：

```
#include <stdio.h>
const float PI=3.14159;    //定义符号常数
void main()                //函数名
{                            //函数体开始
    float r,a,c;            //r:半径; a:面积; c:周长
    r=2.5;                  //给定 r 的值
    a=PI*r*r;               //计算面积 a
    c=2*PI*r;               //计算周长 c
    printf("r=%f,a=%f,c=%f\n",r,a,c); //输出 r,a,c
}                            //函数体结束
```

程序第 1 行是用于包含头文件 `stdio.h` 的编译预处理命令，程序中使用的输出函数 `printf()` 就定义在该头文件中；

第 2 行定义了一个符号常数 `PI`，它代表圆周率；

第 3 行是函数名 `main()`，C99 标准规定：一个程序由若干个函数组成，至少有一个用 `main()` 命名的主函数，任何函数都必须有类型说明，`void` 表示该函数没有返回值；

第 4 行至第 10 行为函数体，是程序的变量定义和执行部分，其中左花括号表示函数体的开始，右花括号表示函数体的结束。每行右边的“//”及其后的文字称为注释，它对程序的执行没有影响，只对程序进行某些必要的说明，以帮助阅读程序。

程序运行结果如下：

```
r=2.500000,a=19.634937,c=15.707951
```

1.2.2 C 程序的结构特点

C 程序的基本组成结构是函数，一个 C 程序由一个或多个 C 函数组成，一个 C 函数由若干 C 语句构成，一条 C 语句由若干基本单词形成。

1. C 函数

C 函数是完成某个整体功能的最小单位,它是相对独立的模块。一个简单的 C 程序可能只有一个主函数,而复杂的 C 程序则可能包含一个主函数和任意多个其他函数。所有的 C 函数都具有相同的结构。

1) 函数名

函数名是标识和调用函数的依据。在 C 程序中,主函数有固定的名称 **main**,其他函数则可以根据标识符的命名方法任意取名。主函数通常包括了整个程序的轮廓,由它再调用其他函数。

2) 形式参数

形式参数是函数调用时进行数据传递的主要途径。形式参数简称形参或虚参,必须放在函数名后面的一对圆括号中。当形式参数超过一个时,相互之间用逗号隔开。有的函数也可以没有形式参数,但圆括号不能省略。

3) 函数体

函数体用来描述函数的功能。函数体要用花括号括住,主要有两大部分,第一部分是本函数内部用到的局部变量类型定义(C 语言中,所有的变量都要先定义后使用),第二部分是语句序列,完成本函数的功能。

图 1.1 描述了 C 程序的一般格式,除了 **main()** 函数外,函数 **fun1()** 到 **funn()** 均为用户自行命名的函数。程序执行时,无论各个函数的书写位置如何,总是先执行 **main()** 函数,由 **main()** 函数调用其他函数,最后终止于 **main()** 函数(有关函数的详细介绍见第 6 章)。

```
#include <头文件>
全局变量定义
void main()
{ 局部变量定义
  语句序列
}
int fun1(形式参数表)
{ 局部变量定义
  语句序列
}
float fun2(形式参数表)
{ 局部变量定义
  语句序列
}
.....
char funn(形式参数表)
{ 局部变量定义
  语句序列
}
```

图 1.1 C 程序的一般格式

2. C 语句

C 语句是完成某种程序功能(如赋值、输入、输出等)的最小单位,所有的 C 语句都以分号结尾。C 语句可分为表达式语句、复合语句和空语句三类。

1) 表达式语句

任何一个 C 表达式的末尾加上分号后,就构成表达式语句。如:

```
i=0;  
x=x+1;
```

2) 复合语句

复合语句是将一组 C 语句用花括号括住,它在语法上被视为一条语句。如:

```
while (i<10)  
{ sum=sum+i;  
  i++;  
}
```

复合语句通常用在条件分支或循环语句中。有时为了数据隐藏的目的,用复合语句形成一个代码块,块中定义的局部变量不会对程序的其他部分发生副作用。

(3) 空语句

空语句指的是只有一个分号的语句。如:

```
for(i=0; i<1000; i++)  
    ;
```

由一条 for 语句(循环语句)和一条空语句组成。空语句用作循环语句的循环体,表示什么也不做。事实上,这个循环的功能是延迟一小段时间。有时,空语句被用作转向点。

3. 基本单词

一个 C 语句由若干个基本单词组成。C 语言共有五种基本单词,即关键字(亦称保留字)、标识符、常数、操作符和分隔符。例如,语句

```
float r,a,c;
```

中, float 是关键字(代表数据类型), r、a 和 c 是标识符(表示变量);又如,语句

```
a=PI*r*r, c=2*PI*r;
```

中, “=”、“*”和“,”是操作符, “2”是常数, PI 是符号常数。

1) 关键字

关键字是 C 语言中有特定意义和用途、不得作为它用的字符序列,其中 C99 标准规定的关键字有 37 个,较之 C89 增加了 5 个,见表 1.1。

表 1.1 C99 关键字

流程控制	break	case	continue	default	do	else
	for	goto	if	return	switch	while
存储属性	auto	extern	static	register		
数据类型	_bool	_complex	_imaginary	char		
	const	double	enum	float	int	long
	restrict	short	signed	struct	union	unsigned
	void	volatile				
其他	inline	sizeof	typedef			

注意：所有的 C 关键字都必须小写。

2) 标识符

标识符用来表示程序中的变量名、数组名、函数名、指针名、结构名、联合名、枚举常数名、用户定义的数据类型名及语句标号等用途的字符序列,可由 1~31 个字符组成(C99 标准规定 63 个字符),第一个字符必须是字母或下划线,后面的字符可以是字母、数字或下划线。例如:

```
AB, Ab, aB, ab, A_b, _ab, ab_, s2d, W_length
```

等都是正确的标识符,而

```
A+B, A'B, A.B, 2abc, α, β, d%
```

等是错误的标识符。

 **注意：** 标识符不能与 C 关键字相同,而且要区分大小写。例如 AB 和 Ab 是两个不同的标识符,ELSE 可以作为标识符,它不会与 C 关键字 else 混淆。

顺便指出,在 C 编译系统的库函数中,经常使用以下划线“_”打头的函数名或变量名,所以在程序中也应尽量避免使用以“_”打头的标识符,免得与库函数冲突。

3) 常数

C 语言中的常数包括数值常数(如 123、-23.5、1.2E4 等)、字符常数(如'a'、'B'、'c'等)、字符串常数(如"xyz"、"good morning"等)、符号常数以及枚举常数(枚举将在第 7 章将介绍)。

4) 操作符

操作符包括各种运算符(如+、-、*、/),有特定意义的标点符号(如花括号、方括号、圆括号、逗号)等。

5) 分隔符

分隔符用来分隔相邻的标识符、关键字和常数。最常用的分隔符是空格,此外还可以用制表符、换行符、换页符等作为分隔符。

1.3 C 程序的书写风格

C 语言采用自由的程序书写风格:

(1) 每个函数在整个程序文件中的位置任意, 主函数不一定出现在程序的开始处, 但不管主函数位于程序中的何处, 程序运行时总是从主函数开始。

(2) 每个程序行中的语句数量任意, 既允许一行内写几条语句, 也允许一条语句分几行书写, 但每条语句都必须以分号(;)结束。有时还可以在程序的适当地方(如两个函数之间)加进一个或多个空行, 使程序结构更加清晰。

(3) 注释的位置任意, 注释可以出现在程序的任何地方, 既可以独占一行或几行, 也可以出现在某语句的开头或结尾处。单行注释一般用“//”开始, 后跟注释内容(这时 C99 标准新增的注释方式); 多行注释通常以“/*”开始, 以“*/”结束, 其间的内容均视为注释, “*”和“/”之间不能有空格。注释不是 C 语句, 它对程序的编译和运行没有影响, 使用注释的唯一目的是增加程序的可读性。

(4) 尽管 C 程序的书写几乎没有限制, 但为使程序清晰易读, 通常每行写一条语句; 不同结构层次的语句从不同的位置开始, 即按缩进格式书写成阶梯形状, 可以用 Tab 键或空格键调整各行的起始位置。

1.4 简单的 C 程序设计

一个程序通常具备三个功能, 即输入数据、数据运算和输出结果。在 C 语言中, 数据运算主要是由赋值语句完成的, 数据的输入、输出则需要调用 C 编译系统提供的输入/输出函数。本节简要介绍赋值语句和格式输入、输出函数的使用, 以便能进行最简单的程序设计。

1.4.1 赋值语句的简单使用

赋值语句具有计算和存储两大功能, 其一般格式为:

`v=e;`

其中, `v` 是变量名, `e` 是表达式。例如, `x=a+b`, 其功能是先计算表达式 `a+b` 的值, 并把该值转换成和 `x` 相同的数据类型后保存在变量 `x` 中。

1.4.2 格式输入输出函数的简单使用

输入指的是通过输入设备将原始数据送入计算机, 以便计算机对它们进行处理; 输出

指的是将保存在内存中的计算结果送到输出设备上,以便人能阅读。由于 C 语言不提供输入输出语句,C 程序中的输入和输出主要是通过 C 编译系统提供的输入输出函数实现的。其中用得最多的是格式输出函数 `printf()` 和格式输入函数 `scanf()`。本节对它们做简单介绍,详细讲解见第 2 章。

1. 格式输出函数 `printf()`

格式输出函数 `printf()` 用来按指定的格式输出数据,是内存与显示器之间进行数据交换的主要手段,也是用户获得程序运行结果的主要途径,其一般调用形式为

```
printf("格式控制字符串",输出项目清单);
```

其中, `printf` 是函数名,其后的括号中是该函数的参数:格式控制字符串用双引号括住,用来规定输出格式,例如, `%d` 用来输出十进制整数, `%f` 用来输出实数;输出项目清单中包含零个或多个输出项,它们可以是常数、变量或表达式,当有多个输出项时,相互之间用逗号隔开。例如:

```
printf("%f,%f",x,y);
```

用来按小数形式输出变量 `x` 和 `y` 的值。

2. 格式输入函数 `scanf()`

格式输入函数 `scanf()` 用来按指定的格式接收输入数据,是内存与键盘之间进行数据交换的主要手段,也是用户为程序运行提供原始数据的主要途径,其一般调用形式为:

```
scanf("格式控制字符串",输入项目清单);
```

其中, `scanf` 是函数名,其后的括号中是该函数的参数:格式控制字符串用双引号括住,用来规定输入格式,其用法和 `printf()` 相同;输入项目清单中至少包含一个输入项,且必须是变量的地址(变量地址的表示形式是在变量名前面加一个“&”),当有多个输入项时,相互之间用逗号隔开。例如:

```
scanf("%f%d",&a,&b);
```

用来接收从键盘输入的一个实数和一个整数,并分别存放在变量 `a` 和 `b` 中。

1.4.3 库函数和标题文件

上面介绍的 `printf()` 和 `scanf()` 是 C 编译系统提供的库函数。库函数不是 C 语言的组成部分,而是由 C 编译系统提供的一些非常有用的功能函数,例如各种输入输出函数、数学函数、字符串处理函数等,可供用户在自己的程序中直接调用。这些库函数的说明、类型和宏定义都分门别类地保存在相应的标题文件(也称头文件)中,而对应的子程序则存放在运行库(.lib)中,当需要使用系统提供的库函数时,只要在程序开始用

```
#include <头文件>
```


或

```
#include "头文件"
```

就可以调用其中定义的库函数。`printf()`和`scanf()`是在头文件`stdio.h`中定义的,因此,在调用它们之前,应先用

```
#include <stdio.h>
```

或

```
#include "stdio.h"
```

将它们所在的标题文件包含进来。这里,由#开头的`include`行以及前面用到的`#define`都称为命令行,它们是C语言的编译预处理命令(详见第9章),不是C语句,命令动词`include`和`define`也不是C关键字。

C99提供的常用标题文件如下(标题文件名不区分大小写):

<code>ctype.h</code>	字符处理函数
<code>math.h</code>	数学函数
<code>stdio.h</code>	标准输入输出及文件操作函数
<code>stdlib.h</code>	标准实用函数
<code>string.h</code>	字符串处理函数
<code>stdbool.h</code>	支持布尔数据类型,以便兼容于C++(C99新增)
<code>complex.h</code>	支持复数算法(C99新增)

从技术角度讲,任何函数都可以自行编制,完全可以不使用C编译系统提供的函数库。但实际中很少这样做,因为函数库C编译系统提供的一种重要的软件资源,充分利用它们,避免重复自行编制,可达到事半功倍的效果;另一方面,C语言为了减少对硬件的依赖,不提供任何执行I/O操作的方法,这时,自行编制I/O函数也存在很大困难。读者应在学习C语言本身的同时,逐步了解和掌握各种库函数的功能和用法,以简化编程。

需要说明的是,不同的C编译系统支持C99标准的程度也是不同的。

1.4.4 简单程序设计举例

程序设计是学习和使用C语言的出发点和归宿。本节介绍两个最简单的C程序设计例子。

【例 1.2】计算任意立方体的表面积和体积。

设一个立方体的三条边分别为`a`、`b`、`c`,若从键盘输入这三条边长,就可以计算任意立方体的表面积和体积。程序如下:

```
#include <stdio.h>
void main()
{ float a,b,c,s,v;
```