21世纪应用型本科系列教材

C语言程序设计

高 涛 陆丽娜 编
刘德安 审

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

TP312
2286
2007

21世纪应用型本科系列教材

C语言程序设计

高 涛 陆丽娜
焦向锋 王思桦 编
刘德安 审

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS
·西安·

内容简介

欢迎使用本书进入美妙的C语言世界。

本书立足于让读者学习语言的同时，掌握面向过程编程技术的一般的思想和方法，从而真正地掌握软件的开发艺术。在语言风格上，深入浅出、易于理解、文字流畅；在体系结构上体系合理、逻辑性强，类似“滚雪球”一样，通过从一个简单的小示例发展成一个完整的软件系统，读者会在不知不觉中开发出来一个独立的系统，并且能够使读者得到设计一个软件的实践经验。

全书共11章，介绍了C语言的基本概念、语法规则以及高级应用，最终顺利过渡到面向对象的C++语言。每章中的技术提示和编程经验帮助读者在学习的过程中能够领悟到高质量的C编程，并且配有大量习题。有辅导教材《C语言程序统计——实验·习题解答·课程设计》配套使用。

本书可作为本、专科学生学习计算机编程语言的教科书，也可以作为广大的编程爱好者学习和提高的参考书。

图书在版编目(CIP)数据

C语言程序设计/高涛等编. —西安：西安交通大学出版社，2007.2
(21世纪应用型本科系列教材)
ISBN 978-7-5605-2401-6

Ⅰ.C... Ⅱ.高... Ⅲ.C语言-程序设计
Ⅳ.TP312

中国版本图书馆CIP数据核字(2007)第002208号

书　　名：C语言程序设计
编　　者：高 涛　陆丽娜　焦向锋　王思桦
出版发行：西安交通大学出版社
地　　址：西安市兴庆南路10号(邮编：710049)
电　　话：(029)82668357　82667874(发行部)
　　(029)82668315　82669096(总编办)
印　　刷：陕西友盛印务有限责任公司
字　　数：590千字
开　　本：787 mm×1 092 mm　1/16
印　　张：24.25
版　　次：2007年2月第1版　2007年2月第1次印刷
书　　号：ISBN 978-7-5605-2401-6/TP·487
定　　价：29.80元

版权所有　侵权必究

前　言

要真正地掌握软件的开发艺术，首先要掌握程序设计语言，在众多的程序设计语言中，C语言以其灵活性和实用性受到了广大计算机应用人员的喜爱。C语言自从1972年诞生于贝尔实验室以来，得到了迅速推广并发展成了一种广泛应用的高级语言。C语言已经发展成了工业界选用的系统实现语言，它是各大操作系统的基础，Unix、Linux、Windows等操作系统的内核都采用了C语言和汇编语言作为开发工具。

近年来，即使随着C++、Java、C#等语言的出现，都没有动摇C语言的基础地位。C语言几乎具备现代程序设计语言的所有语言成分，一旦掌握了C语言，就可以较为轻松地学习其他任何一种程序设计语言。

C语言由于其复杂性和灵活性使得不少初学者感到了学习的困难，但是学生们正为他们学习一种非常有用并且实用的语言而感到鼓舞，并且充满热情，正是这种学习的热情帮助他们克服了很多障碍。即便如此，本书尽可能地以一种通俗易懂的讲述，加上丰富的图解以及生动有趣的范例，使广大的学生们和计算机爱好者能够深刻的理解和掌握作为程序设计的第一门语言。

本书根据读者需求和我们的教学经验，力图体现以下编写特色：

1. 本书主要讲解C语言最基本，最常用的内容，重点主要放在语言本身的难点（如指针和内存的操作）和程序设计技巧方面，对一些C语言中出现频率很低或者与语言编译环境有关的内容只做了简单的介绍。

2. 本书各章节之间密切结合并且以阶梯式前进。章节内部将难点进行分散讲述，使读者在学习的过程中能够循序渐进。且从软件结构设计为出发点，讲述了自上而下的设计理念和方法，使得读者不仅仅局限于语言的学习，而是掌握一种编程的思想。

3. 本书中的示例程序不仅介绍了一些小程序，还演示了一个开发完整软件系统的过程。从第3章至第9章的综合举例，是根据本章的知识逐渐完善，像“滚雪球”一样，从一个简单的小示例发展成一个完整的软件系统，读者会在不知不觉中开发出一个学生信息管理系统，并且能够使读者得到设计一个软件的实践经验。

4. 本书的示例中加入很多生动有趣的元素，例如图形、声音、游戏等等的设计，使读者在学习枯燥的语法过程中感受到程序开发的乐趣，并可以开发一些具有一定界面的感兴趣的小程序，这样可以提高开发程序的热情。

5. 本书中使用结构化的流程图表示方法有助于理解和掌握模块化的程序设计思路和控制结构。书中使用大量的示意图来说明程序执行的过程和一些重要的数据结构及算法。

6. 本书每章的小结、技术提示和编程经验主要总结了一些易错的概念和知识点及在实际的软件开发应用中的编程经验，使读者在学习的过程中就能领悟到高质量的C编程知识，强调使用结构化的思想编写清晰的程序，避免模糊不清的结构和语法规范，为后期的软件开发打好了坚实的基础。

此为试读，需要完整PDF请访问：www.ertongbook.com

7. 本书每章配有大量的习题，这些题目主要来源于笔者的精心设计和一些经典的程序，以及国家计算机等级考试中部分考题。为读者学习后参加等级考试和程序员考试打下了基础。

8. 本书针对目前很多学生学完C语言不会设计程序这一问题，在第9章介绍了主要的数据结构和算法，以便读者能够开发出更复杂和更有效率的C语言程序。

9. 程序设计技术，已经从面向过程的设计技术向面向对象程序设计技术过渡。对一个大规模的应用程序，总体框架是由面向对象程序设计构搭而成，而在局部实现时仍需采用结构化程序设计技术。因此，本书中第10章从实用的角度简要介绍了面向对象的程序设计语言C++。

本书由西安交通大学城市学院高涛、陆丽娜、王思桦老师，西北大学现代学院焦向锋老师编写。感谢西安交通大学出版社屈晓燕、贺峰涛老师多次组织我们讨论如何编写适合应用型人才的教材，给了我们许多启发。感谢西北大学现代学院理工系主任刘德安教授认真审阅了书稿，并为我们提出许多许多宝贵的意见。感谢西北工业大学戴玉超、权怀韦、李旭博士等对本书提出很好的建议。感谢西安交通大学城市学院领导对我们的关心、支持和帮助。

欢迎使用本书进入美妙的C语言世界。C语言博大精深，本书只能从实用易懂的角度去描述它，希望能够抛砖引玉，使读者尽可能地达到一种专业的编程境界。

由于笔者的水平和编程经验有限，本书中肯定有不少的不足或者错误，希望能得到专家和读者的指正。

编者

于西安交通大学城市学院

2006年11月

目 录

第1章　程序设计基础

“什么是程序”?“什么是程序设计”?“什么是程序设计语言”? 本章首先以比较直观的问题建立起对程序、程序设计、程序设计语言的基本认识。而后将简单介绍C程序设计语言,并通过简单实例介绍C语言的一些基本概念。最后指出程序设计中经常遇到的一些问题。

1.1　程序设计与程序语言

1.1.1　程序

你可能已经注意到计算机已应用于航空、工业、农业等许多领域,将数据输入到计算机中,并由计算机生成结果,结果可以显示在屏幕上,也可以打印在纸上。让我们以航空订票为例。如果你想在一个特定的航班上订机位,则只要提供订票所需要的信息:目的地、起程日期和时间、座舱级别,订票柜台将这些信息输入到计算机中,随后关于能否获得机票的信息就显示在屏幕上。

看一看计算机是如何运作的? 它被设计为具有接收输入、进行处理、产生输出的功能。然后,还必须给它一系列指令,指令中说明下列内容:

(1) 需要提供的输入信息。如起程日期、时间、舱位等级、目的地。

(2) 期望输出的信息。如座位存在与否的状态。

(3) 需要进行的处理。如接收数值、检验座位存在与否、显示结果。我们把完成一个特定工作的一系列指令叫程序。程序通常指完成某些事务的一种既定方式和过程,即程序可看作对一系列动作的执行过程的描述。日常生活中也可以找到许多“程序”实例。例如,一个学生早上起床后的行为可以描述为:

起床→刷牙→洗脸→吃饭→上早自习

这是一个最简单形式的程序。描述这种程序就是给出一个包含其中各个基本步骤的序列。如果按顺序实施这些步骤就完成了该项事务。

现在考虑另一个复杂些的过程:到图书馆借教学参考书。这一常见过程可以描述为:

(1) 进入图书馆。

(2) 查找书目。

(3) 填写借书单。

(4) 交图书馆工作人员取书。

(5) 如果该书已经借出,读者可以有两种选择:

① 回到第(2)步(进一步查找其他参考书的书目);

② 放弃借书，离开图书馆。

(6)（工作人员找到了要借的书）办理借书手续。

(7) 离开图书馆。

这个程序比前一个复杂得多。可以看到，这个程序不是一个平铺直叙的动作序列，其中出现了分情况处理和可能出现的重复性动作。如果仔细探究这个实例，我们还可以看到，这一程序还可以进一步细化，可以找出许多上面描述中未处理的情况，例如：若查找图书目录时没有找到所需书目，填写好借书单后已经到了下班时间，借书时发现自己没有带借书证，工作人员查到该读者的借书册数已经达到限额，或发现该读者有逾期未还的图书，因此拒绝借出等等。

由这些现实生活中的例子，可以初步看到程序的一些直观特征。现实生活中有许多类似于程序性活动，当我们身处其中时，通常需要按部就班地一步步完成一系列动作。对这种工作（事物、活动）过程的细节动作描述就是一个“程序”。

一个程序总有开始与结束。在执行此程序的过程中，动作者（无论是不是人）需要按照程序的描述执行一系列的动作。在到达结束位置时工作就完成了。

1.1.2 程序与程序设计

1946年人类发明的一种自动机器叫计算机，它能完成的工作就是计算。计算机的最基本功能是可以执行一组基本操作，每个操作完成一件很简单的计算工作，例如整数的加减乘除运算等等。为使计算机能按人的指挥工作，每种计算都提供了一套指令，其中的每一种指令对应着计算机能执行的一个基本动作。

作为一种看得见、摸得着的物理实体，计算机的基本原理很简单，其最本质特征是它们不仅能按指令工作，而且能自动地按程序工作。因此，人与计算机交流的基本方式就是提供要求它执行的程序。在命令计算机去执行某个程序之后，计算机就会按照程序的规定，严格地执行其中的指令，直至程序结束。

计算机是一种通用的计算机器，加上一个或一组程序后，它就会变为处理某个专门问题、完成某种特殊工作的专用机器。它还可以通过运行不同的程序，也可以在不同时候处理不同问题，甚至同时处理许多不同的问题。人们描述（编制）计算机程序的工作被称为程序设计或者编程，这种工作的产品就是程序。由于计算机的本质特征，从计算机诞生之初就有了程序设计工作。

1.1.3 程序设计语言及其发展

要说明在一个程序的运行中需要做些什么，就需要仔细给出这一程序性活动的一步步细节过程，需要描述程序运行中的各种动作及其执行的顺序。为做到这些，就需要有某种适当的描述方式。一套描述方式就形成了一种语言。

语言一词通常指人生活工作中使用的自然语言，如汉语、英语等。这些语言随着人类发展进步而自然形成，是人相互之间交流信息的工具和媒介。人们用口头语言向别人传播见闻、表达看法和想法；用书面语言写文章、书籍，以实现更大范围中的信息交流。在前面所描述现实生活的程序实例中，我们就是用汉语作为描述程序的语言，所描述的程序是为了方便人们去阅读并按人们的指令去实现所需目的。

为了和计算机交流，指挥它工作，同样需要有与之交流的方式，需要一种意义清晰、人用起

此为试读，需要完整PDF请访问：www.ertongbook.com

来比较方便、计算机也能处理的描述方式。也就是说，需要有描述程序的合适语言。可供人们编写程序用的语言就是程序设计语言，这是一类人们自己设计的语言。程序设计语言常被称为编程语言，也常常简称为程序语言。程序语言的一个突出特点就在于不仅人能懂得和掌握它，能用它描述所需的计算过程，而且计算机也可以“识别”它，并且可以按程序语言所给出的计算过程去运行，完成人所需要的计算工作。程序设计语言是人描述计算的工具，也是人与计算机进行交流信息的媒介。通过用程序语言写程序，人们能指挥计算机完成各种特定工作，完成各种计算。

计算机诞生之初，人们只能直接用二进制形式的机器语言写程序。对于人的使用而言，二进制的机器语言很不方便，用它书写程序非常困难，不但工作效率极低，程序的正确性也难以保证，发现有错误也很难辨认和改正。下面是一台假设计算机上的指令系列：

0000000100000001000 ——将单元 1000 的数据装入寄存器 0
0000000100010001010 ——将单元 1010 的数据装入寄存器 1
0000010100000000001 ——将寄存器 1 的数据乘到寄存器 0 原有数据上
0000000100010001100 ——将单元 1100 的数据装入寄存器 1
0000010000000000001 ——将寄存器 1 的数据加到寄存器 0 原有数据上
0000001000000001110 ——将寄存器 0 里的数据存入单元 1110

这里想描述的是计算算术表达式 a × b + c(这里的符号 a、b、c 分别代表地址为 1000、1010 和 1100 的存储单元)，而后将结果保存到单元 1110 的计算过程(程序)。对于一个复杂程序若用二进制机器指令来书写，十分困难且难于理解。为缓解这一问题，人们发展了符号形式表示，使用相对容易些的汇编语言。用汇编语言写的程序需要用专门软件(汇编系统)加工，翻译成二进制机器指令后才能在计算机上使用。

下面是用某种假想的汇编语言写出的程序，它完成与上面程序同样的工作：

load 0 a ——将单元 a 的数据装入寄存器 0
load 1 b ——将单元 b 的数据装入寄存器 1
mult 0 1——将寄存器 1 的数据乘到寄存器 0 原有数据上
load 1 c ——将单元 c 的数据装入寄存器 1
add 0 1 ——将寄存器 1 的数据加到寄存器 0 原有数据上
save 0 d——将寄存器 0 里的数据存入单元 d

汇编语言的每条指令对应于一条机器语言指令，但采用了助记的符号名，存储单元也用符号形式的名字表示。这样，每条指令的意义都更容易理解和把握了。但是，汇编语言的程序仍然完全没有结构，仅仅是许多这样的指令堆积形成的长长序列，是一团散沙。因此，复杂程序作为整体仍然难以理解。

1954 年诞生了第一个高级程序语言 FORTRAN，它采用完全符号化的描述形式，用类似数学表达式的形式描述数据计算。语言中提供了有类型的变量，还提供了一些流程控制机制，如循环和子程序等。这些高级机制使编程者可以摆脱许多具体细节，方便了复杂程序的书写，写出的程序也更容易阅读，有错误也更容易辨认和改正。FORTRAN 语言诞生后受到广泛欢迎。

高级程序语言更接近人所习惯的描述形式，更容易被接受，也使更多的人能加入程序设计活动中，用高级语言书写程序的效率更高，这使人们开发出更多应用系统，反过来又大大推动

了计算机应用的发展。计算机应用的发展又推动了计算机工业的大发展。可以说，高级程序设计语言的诞生和发展，对于计算机发展到今天起了极其重要的作用。从FORTRAN语言诞生至今，人们已提出的语言超过千种，其中大部分只是试验性的，只有少数语言得到了广泛使用，如C、C++、PASCAL、Ada、Java、LISP、Smalltalk、PROLOG等，这些语言都曾在程序语言或计算机的发展历史上起过(有些仍在起着)极其重要的作用。随着时代发展，今天绝大部分程序都是用高级语言写的。人们也已习惯于用程序设计语言特指各种高级程序语言了。在使用高级语言(例如C语言)描述前面同样的程序片断只需一行代码：

```
d = a * b + c;
```

这表示要求计算机求出等于符号右边的表达式，而后将计算结果存入由d代表的存储单元中。这种表示方式与人们所熟悉的数学形式直接对应，因此更容易阅读和理解。高级语言程序中完全采用符号形式，使人可以摆脱难用的二进制形式和具体计算机的细节。此外，高级语言中还提供了许多高级的程序结构，供编写程序者去组织复杂的程序。

计算机能否理解用这些高级语言编写的指令呢？不，它无法做到这一点，如何使计算机执行用高级编程语言写的程序指令呢？你需要一个翻译，将用编程语言写的指令翻译成机器指令。编译器就是这样一种特别的程序，对每一种语言都有不同的编译器。编译器有如下两种方式：

(1) 编译方式。人们首先针对具体语言(例如C语言)开发出一个翻译软件(程序)，其功能是将采用该种高级语言书写的程序翻译为所用计算机的机器语言的等价程序。这样，用这种高级语言写出程序后，只要将它送给翻译程序，就能得到与之对应的机器语言程序。此后，只要命令计算机执行这个机器语言程序，计算机就能完成我们所需要的工作了。

(2) 解释方式。人们首先针对具体高级语言开发一个解释软件，其功能是一条一条地读入高级语言的程序，并能一步步地按照程序要求工作，完成程序所描述的计算。有了这种解释软件，只要直接将写好的程序送给运行着这个软件的计算机，就可以完成该程序所描述的工作了。

1.2 C语言简介

1.2.1 C语言出现的历史背景

1. C语言的出现

C语言是国际上流行的、很有发展前途的计算机高级语言，它是由贝尔实验室Dennis Ritchie在1973年推出的一种程序设计语言，其目的是用于写操作系统和系统程序，初期用在PDP-11计算机上写UNIX操作系统。20世纪70年代后作为Unix的标准开发语言，C语言随着Unix系统流行而得到越来越广泛的接受和应用，20世纪80年代后它被移植到大型机、工作站等的许多系统上，逐渐成为开发系统程序和复杂软件的一种通用语言。随着微机的蓬勃发展、处理能力的提高和应用的日益广泛，越来越多的人参与微机应用系统的开发工作，需要适合开发系统软件和应用软件的语言。C语言能较好满足人们的需要，因此在微机软件开发中得到日益广泛的应用，逐渐成为最常用的系统开发语言之一，被人们用于开发在微型机上

运行的各种程序。

在使用最多的以 Intel 及其兼容芯片为基础的微机上，也有许多性能良好的商品 C 语言系统可用。包括 Borland 公司早期的 Turbo C 和后续 Borland C/C++ 系列产品；Microsoft（微软）公司的 Microsoft C 和后续 Visual C/C++ 系列产品。还有其他 C/C++ 语言系统产品，使用较广的有 Watcom C/C++ 和 Symantic C/C++ 等。此外还有许多廉价的和免费的 C 语言系统。其他微型机也有多种 C 语言系统。各种工作站系统大都采用 Unix 和 Linux，C 语言是它们的标准系统开发语言。各种大型计算机上也有自己的 C 语言系统。

2. C 语言的发展

C 语言的发展经历如下：

ALGOL60 → CPL → BCPC → B → C → 标准 C → ANSI C → ISO C

（1）ALGOL60 语言是一种面向问题的高级语言。ALGOL60 离硬件较远，不适合编写系统程序。

（2）CPL（combined programming language，组合编程语言）：CPL 是一种在 ALGOL60 基础上更接近硬件的一种语言。CPL 规模大，实现困难。

（3）BCPL（basic combined programming language，基本的组合编程语言）：BCPL 是对 CPL 进行简化后的一种语言。

（4）B 语言：B 语言是对 BCPL 进一步简化所得到的一种很简单接近硬件的语言。B 语言取 BCPL 语言的第一个字母。B 语言精练、接近硬件，但过于简单，数据无类型。B 语言诞生后，Unix 开始用 B 语言改写。

（5）C 语言：C 语言是在 B 语言基础上增加数据类型而设计出的一种语言。C 语言取 BCPL的第二个字母。C 语言诞生后，Unix 很快用 C 语言改写，并被移植到其他计算机系统。

从 C 语言的发展历史可以看出，C 语言是一种既具有一般高级语言特性，又具有低级语言特性的程序设计语言。C 语言从一开始就是用于编写大型、复杂系统软件的，当然它也可以用来编写一般的应用程序。也就是说：C 语言是程序员的语言！

1.2.2　C 语言的基本特点

C 语言之所以能被世界计算机界广泛接受是由于其自身的特点，主要有：

（1）语言成分简洁，紧凑，书写形式自由，是一个比较小的语言。学习时入门相对容易，知道很少东西就可以开始编程。C 语言很好地总结了其他语言提出的程序库概念，把程序设计中需要的许多功能放在程序库（称为标准函数库）里实现，如输入输出功能等。这就使语言本身比较简单，编译程序的实现比较容易。人们早已用 C 语言写了它自己的编译程序，这种程序很容易移植到各种不同计算机上，促进了 C 语言的传播。

（2）提供了丰富的程序机制，包括丰富且功能强大的运算符、各种控制机制和数据定义机制，能满足构造复杂程序时的各种需要。方便易用的函数定义和使用机制使人们可以把复杂程序分解成一个个具有一定独立性的函数，以分解程序的复杂性，使之更容易控制和把握。

（3）提供了一套预处理命令，支持程序或软件系统的分块开发。利用这些机制，一个软件系统可以较方便地先由几个人或几个小组分别开发，然后再集成，构成最终系统。这种工作方式对于开发大型软件系统是必需的。

（4）可以写出效率很高的程序。一般高级语言写出的程序效率较低。这样，开发效率要

求特别高的程序时就只能使用汇编语言。但用C语言开发的程序具有较高的效率，因它还提供了一组比较接近硬件的低级操作，可用于写较低级、需要直接与硬件打交道的程序。这些特点使C语言常被用作汇编语言的“替代物”，大大提高了开发低层程序的效率。

C语言的工作得到世界计算机界的广泛赞许。一方面，C语言在程序设计语言研究领域具有一定价值，由它引出了不少后继语言，还有许多新语言从C语言中汲取营养，吸收了它的不少优点。另一方面，C语言对计算机工业和应用的发展也起了很重要的推动作用。正是由于这些情况，C语言的设计者获得世界计算机科学技术界的最高奖——图灵奖。

1.2.3 C语言的发展和标准化

在设计C语言时，设计者主要把它作为汇编语言的替代品，作为自己写操作系统的工具，因此更多强调的是灵活性和方便性。语言的规定很不严格，可以用许多不规范的方式写程序，因此也留下了许多不安全因素。使用这样的语言，就要求编程序者自己注意可能出现的问题，程序的正确性主要靠人来保证，语言的处理系统（编译程序）不能提供多少帮助。随着应用范围的扩大，使用C语言的人越来越多，C语言在这方面的缺点日益突出起来。由此造成的后果是，人们用C语言开发的复杂程序里常带有隐藏很深的错误，难以发现和改正。

随着应用发展，人们更强烈地希望C语言能成为一种更安全可靠、不依赖于具体计算机和操作系统（如Unix）的标准程序设计语言。美国国家标准局（ANSI）在20世纪80年代建立了专门小组研究C语言标准化问题，在1988年颁布ANSI C标准。这个标准被国际标准化组织和各国标准化机构接受，同样也被采纳为中国国家标准。此后人们继续工作，1999年通过了ISO/IEC 9899:1999标准（一般称为C99）。这一新标准对ANSI C做了一些小修订和扩充。

1.3 C语言程序设计简介

1.3.1 简单的C程序介绍

用C语言写的程序简称为C程序。本书将从下章开始详细讨论各种C语言结构，如何使用C语言编程。在开始这些讨论之前，先请读者看几个简单的程序例子，看看用C语言写出的程序是什么样子。

【例1-1】 一个简单的C语言程序。

```
#include <stdio.h>
main ()
{
    printf("This is a C program.\n");
}
```

说明：本程序的功能是输出一行信息。

```
This is a C program.
```

(1) 上面这个简单程序可分为两个基本部分：第一行是个特殊行，include <stdio.h>说

明程序用到 C 语言系统提供的标准功能(参考标准库文件 stdio. h)。其他几行是程序的基本部分。

(2) main 表示“主函数”。每个 C 语言程序都必须有一个 main 函数,它是每一个 C 语言程序的执行起始点(入口点)。

用{}括起来的是“主函数”main 的函数体。main 函数中的所有操作(或语句)都在这一对{}之间。也就是说 main 函数的所有操作都在 main 函数体中。

(3) “printf”语句是 C 语言的库函数,功能是用于程序的输出(显示在屏幕上),本例用于将一个字符串“This is a C program. \n”的内容输出。即在屏幕上显示:This is a C program.

(4) 注意:函数体中每条语句都要用“;”号结束。

【例 1-2】 用 C 语言实现求和问题。

```
main()                    /* 计算两数之和 */
{
  int a,b,sum;            /* 这是定义变量 */
  a=123;b=456;            /* 以下 3 行为 C 语句 */
  sum=a+b;
  printf("sum=%d\n",sum);
}
```

说明:本程序是计算两数之和,并输出结果。

(1) 同样此程序也必须包含一个 main 函数作为程序执行的起点。{}之间为 main 函数的函数体,main 函数所有操作均在 main 函数体中。

(2) 用/* */括起来的部分是一段注释,注释只是为了改善程序的可读性,在编译、运行时不起作用(事实上编译时会跳过注释,目标代码中不会包含注释)。注释可以放在程序任何位置,并允许占用多行,只是需要注意“/*”与“*/”配对使用,一般不要嵌套注释。

(3) int a,b,sum;是变量声明。声明了三个具有整数类型的变量 a,b,sum。C 语言的变量必须先声明再使用。

(4) a=123;b=456;是两条赋值语句。将整数 123 赋给整型变量 a,将整数 456 赋给整型变量 b。a,b 两个变量的值分别为 123,456。注意这是两条赋值语句,每条语句均用“;”结束。也可以将两条语句写成两行,即:

```
a=123;
b=456;
```

由此可见 C 语言程序的书写可以相当随意,但是为了保证容易阅读要遵循一定的规范。

(5) sum=a+b;是将 a,b 两变量内容相加,然后将结果赋值给整型变量 sum。此时 sum 的内容为 579。

(6) printf("sum=%d\n",sum);是调用库函数输出 sum 的结果。%d 为格式控制,表示 sum 的值是以十进制整数形式输出。程序运行后,输出(显示):sum=579。

【例 1-3】 C 语言中函数的应用。

```
main()                              /* 主函数 */
{                                   /* main 函数体开始 */
    int a,b,c;                      /* 声明部分定义变量 */
```

```
    scanf("%d,%d",&a,&b);
    c=max(a,b);                    /* 调用 max,将调用结果赋给 c */
    printf("max=%d",c);
}                                  /* main 函数体结束 */
int max(int x,int y)               /* 计算两数中较大数的函数 */
{                                  /* max 函数体开始 */
    int z;                         /* 声明部分,定义变量 */
    if(x>y)z=x;
    else z=y;
    return z;                      /* 将 z 值返回,通过 max 带回调用处 */
    }                              /* max 函数体结束 */
```

说明:本程序是输入两个整数,计算两者较大的数,并输出。

(1) 程序包括两个函数。其中主函数 main 仍然是整个程序执行的起点;函数 max 计算两数中较大的数。

(2) 主函数 main 调用 scanf 函数获得两个整数,存入 a,b 两个变量,然后调用函数 max 获得两个数值中较大的值,并赋给变量 c,最后输出变量 c 的值(结果)。

(3) int max(int x,int y)是函数 max 的函数头,函数 max 的函数头表明此函数获得两个整数,返回一个整数。

(4) 函数 max 同样也用{}将函数体括起来。max 的函数体是函数 max 的具体实现。从参数表获得数据,处理后得到结果 z,然后将 z 返回调用函数 main。

(5) 本例还表明函数除了调用库函数外,还可以调用用户自己定义、编制的函数。

1.3.2 C 程序结构

综合上述三个例子,我们对 C 语言程序的基本组成和形式(程序结构)有了一个初步了解:

(1) 程序由函数构成:

① 一个 C 源程序至少包含一个 main 函数,也可以包含一个 main 函数和若干个其他函数。函数是 C 程序的基本单位。

② 被调用的函数可以是系统提供的库函数,也可以是用户根据需要自己编写设计的函数。C 是函数式的语言,程序的全部工作都是由各个函数完成。编写 C 程序就是编写一个个函数。

③ 函数库非常丰富,ANSI C 提供 100 多个库函数,Turbo C 提供 300 多个库函数。

(2) main 函数(主函数)是每个程序执行的起始点。一个 C 程序总是从 main 函数开始执行,而不论 main 函数在程序中的位置。可以将 main 函数放在整个程序的最前面,也可以放在整个程序的最后,或者放在其他函数之间。

(3) 一个函数由函数首部和函数体两部分组成:

① 函数首部:一个函数的第一行。其定义格式如下:

返回值类型 函数名([函数参数类型 1 函数参数名 1][,...,函数参数类型 n,函数参数名 n])

此为试读,需要完整PDF请访问: www.ertongbook.com