

21世纪 计算机基础教育系列教材

谭浩强 主编

C语言程序 设计教程

■ **孟宪福 李盘林 编著**



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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



21 世纪计算机基础教育系列教材

谭浩强 主编

C 语言程序设计教程

孟宪福 李盘林 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

WJS91/33

内 容 简 介

C语言是一种应用非常广泛的结构化高级程序设计语言,既适合于编写应用软件,又特别适合于编写系统软件。本书由12章组成,按照循序渐进的原则,详细地介绍了C语言的基本概念和语法规则,在此基础上,通过精选的典型例题分析,使得读者能够尽快掌握利用C语言进行程序设计的技巧和方法,特别是,为了使读者能够尽快利用C语言编写大型实用程序,在书中的最后一章详细分析了一个图形编辑程序的设计与实现过程,并给出了完整的源程序。

本书是作者根据多年的教学经验编写而成的,在内容编排上尽量体现出易学的特点,在文字叙述上力求条理清晰、简洁,以便于读者阅读。

本书可作为高等院校计算机专业及其相关专业的教材或教学参考书,也可作为自学用书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

C语言程序设计教程/孟宪福等编著. —北京:电子工业出版社,2003.7

(21世纪计算机基础教育系列教材)

ISBN 7-5053-8650-6

I. C… II. 孟… III. C语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 026822 号

责任编辑:应月燕

印刷者:北京兴华印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:15.25 字数:403 千字

印 次:2004 年 1 月第 2 次印刷

印 数:3 000 册 定价:20.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010)68279077。质量投诉请发邮件至 zlt@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

《21 世纪计算机基础教育系列教材》序

21 世纪是信息时代,是科学技术高速发展的时代。计算机技术与网络技术的结合,使人类的生产方式、生活方式和思维方式发生了深刻的变化。在新世纪中,信息技术已成为当代人类文化的一个重要组成部分。我们要将信息技术知识和应用向一切有文化的人普及。

高等学校承担着为社会培养高层次人才的任务,大学生毕业后应当成为我国各个领域中的计算机应用人才,成为向全社会推广计算机应用的积极分子。在大学里应当把计算机教育放在十分重要的位置。

我国高校的计算机基础教育起步于 20 世纪 80 年代初。20 年来从无到有迅速地发展,从理工科专业发展到所有专业,从最初只开设一门语言课到按三个层次设置课程,学时也从三四十小时增加到一二百小时。计算机基础教育已经先后上了几个台阶,现在又需要上一个新的台阶。在新世纪,计算机基础教育已发展为对全体大学生的信息技术教育,我们要引导大学生学习和掌握先进的信息技术,学会在各专业领域中应用信息技术,以适应科学技术和经济发展的需要。

计算机不仅是一种工具,也是一种文化,工具是可选的,文化却是必备的。对学生来说,它还是全面素质教育的一个重要部分,通过学习计算机知识能激发学生对先进科学技术的向往,启发学生对新知识的学习热情,培养学生的创新意识,提高学生的自学能力,锻炼学生动手实践的能力。多年来的实践证明,对计算机感兴趣的学生,绝大多数都是兴趣广泛、思想活跃、善于思考、自学能力较强、喜欢动手实践的。他们绝不是只会死背书本的书呆子。

我们必须认真分析计算机基础教育的特点,根据教学上的需要与可能,制定出恰当的教学要求,使学生在有限的时间内能学到最多的有用的知识。计算机基础教育实际上是计算机应用的教育,应当以应用为目的,以应用为出发点。全国高等院校计算机基础教育研究会曾提出了在计算机基础教育中应当正确处理的 10 个关系,即:(1) 理论与应用的关系,(2) 深度与广度的关系,(3) 当前与发展的关系,(4) 硬件与软件的关系,(5) 追踪先进水平与教学相对稳定的关系,(6) 课内与课外的关系,(7) 课程设置与统一考试的关系,(8) 计算机课程与其他课程的关系,(9) 要求学生动手能力强与当前设备不足的矛盾,(10) 计算机技术发展迅速与师资现状的矛盾。在教学实践中,许多学校都创造了丰富的经验。

在计算机基础教育的教学中,首先需要解决的问题是:准确定位,合理取舍教学内容。我们必须分清楚:哪些内容是需要,哪些内容是不需要的;哪些内容是目前暂时可以不学而留待以后学的,哪些内容是目前不必学而以后也不必学的;哪些内容是主要的,哪些是次要的。绝不可不加分析、不问主次,贪多求全,使学生感到难以入门。

在教学方法和教材的编写上,要善于用通俗易懂的方法和语言说明复杂难懂的概念。传统的教学三部曲是:提出概念—解释概念—举例说明。我在多年教学实践中对于计算机应用课程总结了新的三部曲:提出问题—介绍解决问题的方法—归纳出必要的概念和结论。从具体到抽象,从实际到理论,从个别到一般。这是符合人们的认识规律的。实践证明,这样做已取得了很好的效果。

为了推动高校的计算机基础教育,我在 1996 年主编了《计算机教育丛书》,由电子工业出

版社出版。编写这套丛书的指导思想是 20 个字：“内容新颖、实用性强、概念清晰、通俗易懂、层次配套。”（也可简单地概括为：“新颖、实用、清晰、通俗、配套”。）先后出版的近 20 种供大学计算机基础教育使用的教材，受到高校广大师生的欢迎，几年内已发行近 100 万册，大家认为它定位准确、程度适当、内容丰富、通俗易懂、便于自学。

在进入 21 世纪之际，我们根据新时期的要求，按照上面所述的指导思想，重新进行规划，对原有的教材进行了筛选，淘汰了部分内容已过时的教材，同时根据计算机技术和高校计算机基础教育的发展，组织编写了一些新教材，并对原有教材进行了修订和补充，以实现推陈出新、不断提高。

我们遴选了具有丰富教学经验的高校老师编写这套教材。在这套系列教材中，我们提供了多种计算机课程的教材供各校选用，其中包括必修课和选修课。不同专业、不同层次的学校都可以从中选到合用的教材，我们还将根据计算机基础教育的需要不断推出新的教材。

本系列教材是由浩强创作室策划、组织和编写的。参加工作的有：谭浩强、薛淑斌、秦建中、史济民、吴功宜、边莫英、徐士良、李盘林、孟宪福、张基温、宋国新、徐安东、毛汉书、李凤霞、许向荣、周晓玉、张玲、刘星、王兴玲、蔡翠平、訾秀玲、索梅、李宁、赵丹娅、仇芝仙等。电子工业出版社对本丛书的出版给予了大力的支持，使得本丛书得以顺利出版。

由于我们的水平和经验有限，加以计算机科学技术发展很快，本丛书肯定会有不少缺点和不足，诚恳地希望专家和读者不吝指正，我们将继续努力工作，使本丛书能尽量满足广大读者的要求。

全国高等院校计算机基础教育研究会会长
《21 世纪计算机基础教育系列教材》主编

谭浩强

2003 年 3 月 1 日

再版前言

C语言是目前应用最为广泛的计算机高级程序设计语言之一,它短小精悍,功能齐全,是一种结构化程序设计语言。C语言能够运行于多种操作系统环境下,不仅编写了著名的操作系统软件 UNIX,在软件史上立下了丰碑,而且也编写了许许多多的应用软件。因此,学好 C 语言程序设计是非常必要的,它为进一步学习和掌握 C++ 语言程序设计打下坚实基础,同时也是深刻理解 C++ 语言程序设计的一条捷径。

《C语言简明教程》自 1996 年 6 月问世以来,先后印刷了十几次,发行了数万册,得到了广大师生的认可,成为 C 语言程序设计领域中的一个“品牌”。为了更好地满足广大读者的需要,我们对该书在内容上进行了修订,并将书名改为《C 语言程序设计教程》。本书除保持《C 语言简明教程》的简明、易学和实用等特点外,并在下面三个方面进行了修改和强化:

1. 对部分文字做了仔细修改,以使读者更容易理解。

2. 为了扩大读者视野和更深入掌握 C 语言程序设计的方法,本书增加了部分新内容,如二叉树处理等,同时增加了部分例题和习题,并给出了各章习题的参考答案,供初学者参考。

3. 增加了“C 语言综合应用”一章。它详细介绍了利用 C 语言编写的程序规模较大的图形编辑程序 Panda,在该程序中几乎包含了 C 语言的各个方面,其中包括菜单设计、图形处理(图形输入和图形编辑)以及文件管理等。仔细地阅读和分析此程序,无疑会在短期内提高利用 C 语言来编写大型实用程序的能力。

本书的编写和出版得到了谭浩强教授的大力支持和帮助,在此表示感谢。

限于作者水平,书中难免还会有不少缺点和错误,恳切希望高等院校师生、计算机工作者和广大读者批评指正。

编 者
2003 年 3 月

前 言

C语言是一种短小精悍的计算机高级程序设计语言,它是根据结构化程序设计原则设计并实现的。C语言具有丰富的数据类型,它为结构化程序设计提供了各种数据结构和控制结构,能够实现汇编语言中的大部分功能,同时,用C语言编写的程序具有良好的可移植性。尽管当初C语言是为编写UNIX操作系统而设计的,但它并不依赖于UNIX操作系统,目前C语言能在多种操作系统环境下运行,并且已经在广阔的领域里得到了应用,是目前国际上应用最为广泛的高级程序设计语言之一。

多年来,作者一直从事C语言的教学工作,同时也利用C语言来开发实际的课题,本书就是作者根据多年的教学经验和应用C语言的体会写成的。本书的主要特点可归纳如下:

1. 按照循序渐进的原则,逐步引出C语言中的基本概念,如C语言中的运算符比较丰富,其优先级也比较复杂,在本书中,根据运算符的种类,把运算符分散在不同的章节中进行讲解,这样将有助于读者的消化和掌握。

2. 在讲解C语言中的基本概念时,除了阐述理论之外,还通过典型的例题,着重强调了基本概念在程序设计中的应用,以利于读者理解和掌握。

3. 本书的重点是放在C语言的使用上,书中没有深奥的理论和算法,在例题中出现的每一个算法,都给出了比较详细的解释,因此特别适合于初学者和自学者使用。

4. 在本书的每一章中都包括“应用举例”一节,其中的例题包含了本章讲解的主要内容,有些例题还具有一定的难度,通过阅读和分析这些例题,能使读者对本章内容的应用有一个全面的了解。

5. 每章的最后都附有一定量的习题,其中包括程序分析题和编程题,这些习题对于读者巩固已学习的内容是大有益处的。

6. 在文字叙述上力求条理清晰、简洁,以利于读者阅读。

作者认为,要学好C语言,除了掌握C语言的基本理论之外,还必须加强实践环节,本书中的所有例题都在微机上使用Microsoft C调试通过,读者可以边学习边上机,刚开始时可以调试本书中的例题,待学习一段时间之后,就可以调试自己编写的程序了,只有这样,才能加快学习进度,提高学习效率。

本书在编写过程中,一直得到谭浩强教授的支持和帮助,在此表示深深的谢意。

由于作者水平有限,经验不足,书中一定有不少缺点和错误,敬请有关老师、计算机工作者和广大读者批评指正。

编 者
1996年6月

目 录

第1章 绪言	(1)
1.1 C语言简介	(1)
1.1.1 C语言发展简史	(1)
1.1.2 C语言的特点	(1)
1.2 C语言程序的开发步骤	(2)
1.3 C语言的程序结构	(3)
习题	(5)
第2章 数据、运算符和表达式	(7)
2.1 基本概念	(7)
2.1.1 标识符	(7)
2.1.2 常量	(7)
2.1.3 变量	(7)
2.1.4 关键字	(8)
2.2 基本数据类型	(8)
2.2.1 整型变量及其常量	(8)
2.2.2 浮点型变量及其常量	(9)
2.2.3 字符型变量及其常量	(9)
2.2.4 长整型、短整型和无符号整型	(10)
2.2.5 类型定义 typedef	(11)
2.3 算术运算符、赋值运算符及其表达式	(12)
2.3.1 算术运算符和算术表达式	(12)
2.3.2 赋值运算符和赋值表达式	(13)
2.4 关系运算符、逻辑运算符及其表达式	(13)
2.4.1 关系运算符和关系表达式	(13)
2.4.2 逻辑运算符和逻辑表达式	(14)
2.5 变量的初始化	(15)
2.6 不同类型数据之间的转换	(16)
2.6.1 自动类型转换	(16)
2.6.2 强制类型转换	(16)
2.7 sizeof 运算符	(17)
2.8 应用举例	(17)
习题	(18)
第3章 数据的输入和输出	(20)
3.1 数据的输入	(20)
3.1.1 字符输入函数 getchar	(20)
3.1.2 格式输入函数 scanf	(20)
3.2 数据的输出	(22)
3.2.1 字符输出函数 putchar	(22)

3.2.2 格式输出函数 printf	(22)
3.3 应用举例	(24)
习题	(25)
第4章 基本语句	(26)
4.1 结构化程序设计简介	(26)
4.2 语句和复合语句	(27)
4.3 条件语句	(27)
4.3.1 if 语句	(28)
4.3.2 条件运算符	(30)
4.3.3 switch 语句	(30)
4.3.4 应用举例	(33)
4.4 循环语句	(35)
4.4.1 while 循环语句	(35)
4.4.2 do-while 循环语句	(37)
4.4.3 for 循环语句	(39)
4.4.4 break 语句和 continue 语句	(41)
4.4.5 goto 语句	(42)
4.4.6 逗号运算符和空操作语句	(43)
4.4.7 应用举例	(44)
习题	(46)
第5章 数组	(48)
5.1 一维数组	(48)
5.1.1 一维数组的定义和引用	(48)
5.1.2 一维数组元素的初始化	(49)
5.2 二维数组	(51)
5.2.1 二维数组的定义和引用	(51)
5.2.2 二维数组元素的初始化	(52)
5.3 字符数组和字符串	(55)
5.3.1 字符数组	(55)
5.3.2 字符串	(56)
5.4 应用举例	(60)
习题	(63)
第6章 函数	(65)
6.1 函数的概念	(65)
6.2 函数的定义和调用	(66)
6.2.1 函数的定义	(66)
6.2.2 函数的调用	(67)
6.3 函数的返回值	(67)
6.4 函数的参数及其传递方式	(70)
6.4.1 非数组作为函数参数	(70)
6.4.2 数组作为函数参数	(71)
6.5 函数的嵌套调用和递归调用	(73)
6.5.1 函数的嵌套调用	(73)
6.5.2 函数的递归调用	(75)

6.6 变量的作用域及其存储类型	(77)
6.6.1 局部变量及其存储类型	(77)
6.6.2 全局变量及其存储类型	(79)
6.7 内部函数和外部函数	(81)
6.7.1 内部函数	(81)
6.7.2 外部函数	(82)
6.8 应用举例	(82)
习题	(85)
第7章 编译预处理	(87)
7.1 宏定义	(87)
7.2 文件包括	(89)
7.3 条件编译	(91)
7.4 应用举例	(94)
习题	(95)
第8章 结构和联合	(97)
8.1 结构类型变量的定义	(97)
8.2 结构类型变量的引用	(99)
8.3 结构变量的初始化	(100)
8.4 结构和函数	(101)
8.4.1 结构变量做函数参数	(101)
8.4.2 函数的返回值是结构类型变量	(101)
8.5 结构和数组	(102)
8.5.1 结构中包含数组	(102)
8.5.2 结构数组	(103)
8.6 结构的嵌套	(104)
8.7 联合	(106)
8.8 枚举	(108)
8.9 应用举例	(109)
习题	(112)
第9章 位运算	(113)
9.1 二进制位运算	(113)
9.2 位段	(118)
9.3 应用举例	(119)
习题	(120)
第10章 指针	(122)
10.1 指针的基本概念	(122)
10.2 指针变量的定义和引用	(122)
10.2.1 指针变量的定义	(122)
10.2.2 指针变量的引用	(123)
10.3 指针和结构	(125)
10.3.1 指向结构的指针	(125)
10.3.2 结构中包含指针	(126)
10.3.3 链表	(127)
10.3.4 二叉树	(131)

10.4 指针和数组	(133)
10.4.1 指向数组元素的指针及其操作	(133)
10.4.2 数组名和函数参数	(134)
10.4.3 字符串和指针	(136)
10.4.4 指针数组	(138)
10.5 指针和函数	(139)
10.5.1 指针变量作为函数的参数	(139)
10.5.2 函数的返回值是指针	(141)
10.5.3 指向函数的指针	(141)
10.5.4 命令行参数	(146)
10.6 应用举例	(148)
习题	(155)
第 11 章 文件	(158)
11.1 文件的基本概念	(158)
11.2 文件类型指针和文件号	(159)
11.3 缓冲文件系统	(159)
11.3.1 文件打开函数 fopen	(159)
11.3.2 文件关闭函数 fclose	(160)
11.3.3 文件读函数 fgetc, fread, fscanf	(160)
11.3.4 文件写函数 fputc, fwrite, fprintf	(162)
11.3.5 文件定位函数 rewind, fseek, ftell	(163)
11.3.6 应用举例	(164)
11.4 非缓冲文件系统	(166)
11.4.1 文件打开函数 open 和文件创建函数 creat	(166)
11.4.2 文件关闭函数 close	(167)
11.4.3 文件读函数 read	(167)
11.4.4 文件写函数 write	(168)
11.4.5 文件定位函数 lseek 和 tell	(168)
11.4.6 应用举例	(168)
习题	(170)
第 12 章 C 语言综合应用	(172)
12.1 图形编辑程序 Panda	(172)
12.2 图形库函数简介	(172)
12.3 Panda 的数据组织	(174)
12.4 Panda 的实现	(176)
12.4.1 Panda 的菜单设计	(176)
12.4.2 Panda 的图形处理	(177)
12.4.3 Panda 的文件操作	(177)
12.5 Panda 源程序清单	(177)
习题	(201)
附录 A 标准 ASCII 字符集	(202)
附录 B 运算符的优先级及其结合性	(205)
附录 C Turbo C 集成开发环境简介	(206)
附录 D C 语言的巴科斯范式(BNF)描述	(208)
附录 E 习题解答	(212)

第1章 绪 言

随着计算机硬件技术和软件水平的不断提高,作为人机交流主要工具的计算机程序设计语言也经历了从简单到复杂、从低级到高级的发展过程。在诸多的计算机高级语言中,C语言是目前国内外最为流行的计算机高级程序设计语言之一,它设计精巧、功能齐全、使用灵活,既适合于编写应用软件,又特别适合于编写系统软件。

C语言最初是为描述和实现UNIX操作系统而设计和实现的。在此以前像UNIX操作系统那样的系统软件,一般都是利用汇编语言那种低级语言来编写的,自C语言开发成功以来,使得利用高级语言来编写系统软件成为可能。UNIX操作系统源代码的90%以上是由C语言编写的,UNIX操作系统的一些主要特点,如便于理解、易于修改及具有良好的可移植性等,在一定程度上都受益于C语言,所以,UNIX操作系统的成功与C语言是密不可分的。最初的C语言是依赖于UNIX操作系统环境的,随着C语言的不断发展以及应用的普及,目前,C语言已经能够在多种操作系统如UNIX、DOS等环境下运行,而且,实用的C语言编译系统种类繁多,适用于IBM PC机运行的就有Microsoft C (MSC)、Turbo C (TC)、Borland C (BC)和Lattice C (LC)等。

1.1 C语言简介

1.1.1 C语言发展简史

在20世纪60年代,随着计算机科学体系的形成与完善,高级程序设计语言的研究得到了长足的发展,但是,在当时出现的高级语言中,缺乏用于编写像操作系统编译程序等系统软件的工具,系统程序的设计主要还是依赖于汇编语言。为了改变这种状况,1967年Martin Richards设计并实现了BCPL(Basic Combined Programming Language)语言,后来,这一语言被移植到了多种计算机上,并得到了广泛的应用。此后不久,Ken Thompson在BCPL语言的基础上设计并实现了B语言,并用B语言在PDP-7机上实现了第一个UNIX操作系统。接着,在1972年至1973年间,D. M. Ritchie在B语言的基础上,又重新设计了一种语言,并在PDP-11机上实现,同时用这种语言重写了UNIX操作系统。由于这一语言是在BCPL语言和B语言的基础上开发出来的,因此被称为C语言。由于BCPL语言和B语言是无类型的语言,而C语言却能支持多种数据类型,因此C语言与BCPL语言和B语言是不同的,它更能反映当代计算机的体系结构,因而得到了广泛的应用。

1.1.2 C语言的特点

C语言能够成为目前应用最为广泛的高级程序设计语言之一,完全是由其语言特点决定的。C语言的特点可大致归纳如下:

(1) C语言短小精悍,基本组成部分紧凑、简洁。

C语言一共只有32个标准的关键字、45个标准的运算符以及9种控制语句,不但语言的

组成精练、简洁,而且使用方便、灵活。

(2) C 语言运算符丰富,表达能力强。

C 语言能够处理多种运算符,其运算符包含的内容广泛,所生成的表达式简练、灵活,有利于提高编译效率和目标代码的质量。

(3) C 语言数据结构丰富,结构化好。

C 语言提供了编写结构化程序所需要的各种数据结构和控制结构,这些丰富的数据结构和控制结构以及以函数调用为主的程序设计风格,保证了利用 C 语言所编写的程序能够具有良好的结构化,同时,在 C 语言程序设计中,允许将一个复杂的程序分割为多个模块,并可由多人同时编写,当分别调试完成后,再通过连接程序连接到一起,形成一个完整的程序。

(4) C 语言提供了某些接近汇编语言的功能,有利于编写系统软件。

C 语言具有“高级语言”和“低级语言”的双重特点,它提供的一些运算和操作,能够实现汇编语言的一些功能,如 C 语言可以直接访问物理地址,并能进行二进制位运算等,这为编写系统软件提供了方便条件。

(5) 由 C 语言程序生成的目标代码质量高。

C 语言程序所生成的目标代码的效率仅比用汇编语言描述同一个问题低 20% 左右,因此,用 C 语言编写的程序执行效率高。

(6) C 语言程序可移植性好。

在 C 语言所提供的语句中,没有直接依赖于硬件的语句,与硬件有关的操作,如数据的输入、输出等都是通过调用系统提供的库函数来实现的,而这些库函数本身并不是 C 语言的组成部分。因此,用 C 语言编写的程序能够很容易地从一种计算机环境移植到另一种计算机环境中。

当然,C 语言本身也有其弱点:一是运算符的优先级和结合性比较复杂,不容易记忆;二是由于 C 语言的语法限制不太严格,这在增强了程序设计的灵活性的同时,在一定程度上也降低了某些安全性,这对程序设计人员提出了更高的要求。

总之,由于 C 语言的上述特点,使得 C 语言越来越受到程序设计人员的重视,并且已经在广泛的领域里得到了应用。

1.2 C 语言程序的开发步骤

开发一个 C 语言程序的基本步骤如图 1.1 所示。

1. 编辑

选择适当的编辑程序,将 C 语言源程序通过键盘输入到计算机中,并以文件的形式存入到磁盘中。在 DOS 系统下,经过编辑后得到的源程序文件都是以 .C 为其文件扩展名。

2. 编译

通过编辑程序将源程序输入到计算机后,需要经过 C 语言编译成目标程序。在对源程序的编译过程中,可能会发现程序中的一些语法错误,这时就需要重新利用编辑程序来修改源程序,然后再重新编译。在 DOS 系统下,经过编译后得到的目标文件都是以 .OBJ 为其文件扩展名。

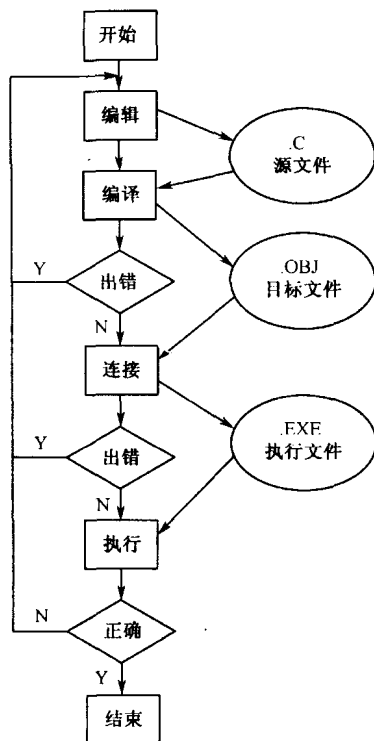


图 1.1 C 语言程序开发步骤

3. 连接

经过编译后生成的目标文件是不能直接执行的,它需要经过连接之后才能生成可执行的代码。在 DOS 系统下,连接后所得到的可执行文件都是以 .EXE 为其文件扩展名。

4. 执行

经过编译、连接之后,源程序文件就可以生成可执行的文件,这时就可以执行了。在 DOS 系统下,只要键入可执行的文件名,并按“回车”键后,就可执行文件了。

现在有许多厂家都推出一种集成环境来处理 C 语言程序,如 Turbo C,在这种集成环境下,对程序的编辑、编译和连接等操作,都可以在一个窗口下进行,使用起来非常方便。

在本书附录 C 中简单介绍了 Turbo C 集成开发环境的使用方法,感兴趣的读者可以参考使用。

1.3 C 语言的程序结构

C 语言的程序结构比较简单,很容易掌握,它主要是通过函数之间的调用来实现指定的功能。在这一节中,将通过编写几个简单的 C 语言程序,来阐述 C 语言的程序结构,同时也对 C 语言的基本语法成分进行相应的说明,以便读者对 C 语言程序有一个概括的了解,为以后的学习打下基础。

【例 1.1】 编写一个 C 语言程序,用于显示字符串“Hello,World!”。

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

```
main( )
{
    printf("Hello,World! \n");
}
```

上述程序是一个简单而完整的 C 语言程序,经过编辑、编译和连接后,其执行结果是在屏幕的当前光标位置处显示如下字符串:

Hello,World!

下面对上述程序进行一下说明:

(1) 一个 C 语言程序可以由多个函数组成,但任何一个完整的 C 语言程序,都必须包含一个且只能包含一个名为 main 的函数,程序总是从 main 函数开始执行的。

(2) 由左右花括号括起来的部分是函数体,函数体中的语句将实现程序的预定功能。在本例中,main 函数的函数体中只有 printf 一个语句,它的功能是进行格式化输出(显示),即将字符串“Hello,World! \n”显示在终端屏幕上。其中,字符串中的字符“\”和“n”合起来,表示一个“换行”字符,在“换行”字符后面输出的任何字符,将被显示在屏幕的下一行上。

(3) C 语言中的每个基本语句,都是以“;”结束的,分号是 C 语言语句的终结符。

(4) C 语言程序的书写格式比较自由,没有固定的格式要求,在一行内,既可以写一个语句,也可以写多个语句。为了提高程序的可读性,往往根据语句的从属关系,以缩进书写的形式来体现出语句的层次性。

(5) #include 命令是编译预处理命令,其作用是将由双引号或尖括号括起来的文件中的内容,读入到该语句的位置处。在使用 C 语言输入、输出库函数时,一般需要使用 #include 命令将“stdio. h”文件包含到源文件中。有关 #include 命令的作用及其使用方法,将在第 7 章中作详细介绍。

【例 1.2】 从键盘输入 2 个整数,并将这 2 个整数之和显示出来。

```
#include "stdio. h"

int ADDxy(a,b)    /* 计算 2 个整数之和 */
{
    int a,b;

    {
        int c;
        c=a+b;
        return(c);
    }
}

main( )
{
    int x,y,z;
    scanf("%d%d",&x,&y);    /* 读入 2 个整数,存入变量 x 和 y 中 */
    z=ADDxy(x,y);
    printf("The sum of %d and %d is %d",x,y,z);
}
```

上述程序经过编辑、编译和连接后,执行情况说明如下:

当此程序从 main 函数开始执行,执行到 scanf 语句时,将等待用户从键盘输入 2 个整型数据后再继续执行,假如用户输入 10 和 20(此时,x 的值为 10,y 的值为 20),则屏幕将显示如下信息:

The sum of 10 and 20 is 30

下面,对上述程序进行一下说明:

(1) 程序中由/*和*/括起来的内容是程序的注释部分,它是为了增加程序的可读性而设置的。注释部分对程序的编译过程和执行结果没有任何影响。

(2) C语言中的所有变量都必须定义为某种数据类型,同时必须遵循“先定义、后使用”的原则,如语句:

```
int x,y,z;
```

定义了x、y、z是3个整型变量,以后就可以使用这3个变量来存放整型数据。

(3) 一个C语言程序可以由多个函数组成,通过函数之间的调用来实现相应的功能。程序中所使用的函数,既可以是系统提供的库函数,也可以是用户根据需要自己定义的函数。如上述main函数中调用的scanf函数和printf函数,就是系统提供的库函数,这些函数不需要用户自己定义,在需要时,只要按照指定的格式进行调用即可。C语言编译系统提供的库函数非常丰富,特别是与硬件打交道的部分,很多工作只要调用库函数就可以实现,而函数ADDxy是自定义的函数,它是用户为了实现加法功能而自己编写的函数。每一个函数都用于完成一特定的功能,ADDxy函数的功能是将其参数a、b之和通过return语句返回给main函数中的变量z。正确地使用函数,将有助于编写易于理解的、结构化好的程序。有关函数的详细说明请参阅第6章。

(4) 程序中调用的scanf函数的作用是进行格式化输入,其中由圆括号括起来的部分是函数的参数部分,不同的函数需要不同的参数,scanf函数中的参数主要包括两部分内容:一是“格式控制”部分,它用于对输入数据的格式进行说明;二是“地址表”部分(本书中出现的表的概念,如地址表、输出表等,是指用逗号分隔的有限个元素序列),它使用的是存放输入数据的变量的地址。

程序中调用的printf函数的作用是进行格式化输出,其参数也包括两部分内容:一是“格式控制”部分,用于对输出数据的格式进行说明;二是“输出表”部分,它使用的是存放输出数据的变量本身。有关数据的输入、输出以及函数的调用形式,将在以后作详细的介绍。

习题

1.1 试简述C语言的主要特点及其用途。

1.2 请参照本章例题,编写一个C语言程序,用于显示以下信息:

Happy New Year!

1.3 请参照本章例题,分析下面的C语言程序,并给出运行结果。

```
#include "stdio.h"
main( )
{
    int a,b,c;
    a=100;
    b=20;
    c=a-b;
    printf("sum=%d",c);
}
```

1.4 请参照本章例题,编写一个C语言程序,用于显示以下信息:

#####

Wonderful!

#####

1.5 请参照本章例题,分析下面的 C 语言程序,并给出模拟运行结果。

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

```
main( )
```

```
{
```

```
    int x,y,z;
```

```
    scanf("%d%d",&x,&y);
```

```
    z=x/y;
```

```
    printf("Result=%d",z);
```

```
}
```