

NK

COMPUTING

计算机技术教育丛书

现代C语言程序

徐德民 编著

设计教程



南开大学出版社

TP312

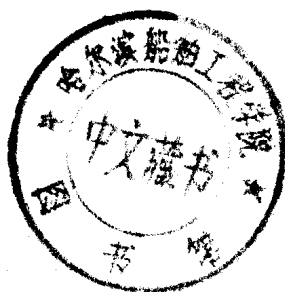
379101

X720

计算机技术教育丛书

现代 C 语言 程序设计教程

徐德民 编著



南开大学出版社



[津]新登字 011 号

现代 C 语言程序设计教程
徐德民 编著

南开大学出版社出版
(天津八里台南开大学内)
邮编 300071 电话 3358542
新华书店天津发行所发行
南开大学出版社印刷厂印刷

1994 年 5 月第 1 版 1994 年 5 月第 1 次印刷
开本:787×1092 1/16 印张:16.75 插页:2
字数:429 千 印数:1—11000

ISBN 7-310-00682-8
TP·22 定价 12.50 元

前 言

C语言是70年代初期推出的编译型程序设计语言。经过二十多年,C语言表现出强大的生命力,已经从早期在少数小型机机种上使用的程序语言,发展成当前在各种大型机、小型机、工作站和微型机上普遍使用的通用程序设计语言。由于C语言具有较强的处理功能,使用C语言进行程序设计已经成为当前软件开发工作的主流。因此,熟练掌握C语言是现代计算机工作者应该具备的基本条件,也是学习更高层次的面向对象的C++语言必须具备的基础。

在C语言发展的二十年多中,其体系日趋完善,功能不断增强。与此相应,各种新型的C编译系统陆续问世,C语言已经从早期的传统模式发展成为当前的现代模式。本书采用当前最新的现代C语言标准——ANSI C为主线,兼顾当前流行的各种最新版本C编译系统,如Microsoft C 5.0~7.0、Turbo C 2.0等,并且从系统深度讲述C语言功能的实质。使读者能够从本质上掌握C语言,达到能够正确熟练地应用C语言进行软件开发的水平。总而言之,本书的宗旨在于向广大读者呈献一本内容新颖、系统全面、按现代风格介绍C语言程序设计的书籍。

全书共分为十章。第一章介绍C程序的格式和结构特点以及C程序开发过程。第二章给出C语言处理的各种数据及其特性。第三章介绍C语言的运算种类及运算顺序。第四章从结构化程序设计角度,介绍C语言的流程控制。第五章从实质上讨论了C语言的重要概念——指针及其特性。第六章全面给出了C语言各种类型的函数以及函数间数据传递的方法。第七章讨论C语言的复合数据类型——结构和联合以及枚举类型。第八章从实用性出发,介绍使用C语言处理文件的方法。第九章给出了C语言程序设计中常用的标准函数。第十章介绍编译预处理和分割编译。

为了便于读者直观地理解和牢固地掌握本书的知识,并能尽快地把它们应用到软件开发实际工作中去,书中给出了大量的实用程序做为例题,同时每章都配备了一定数量的习题可供上机实习使用。

本书是作者在多年从事C语言教学和软件开发经验的基础上,参考了国内外有关C语言的著作,并参照了国家教委考试管理中心计算机等级考试委员会编写的《计算机等级考试大纲》中的C语言部分,完成了编著工作。因此,本书既可作为大专院校的教材使用,又可供参加计算机等级的人员学习和参考。

在本书的编写过程中,得到了天津大学计算机科学与工程系柏家球教授、管理工程系李宗耀教授和电子工程系滕建辅教授的热情帮助和指导。柏家球教授审阅了全书并提出了许多宝贵意见。在此一并地向他们表示衷心的感谢。

限于作者的经验和水平,书中难免存在错误和不足之处,殷切希望得到专家和读者的批评指正。

作 者

1994年3月于天津大学

目 录

第一章 概述	(1)
§ 1.1 C语言的发展和特点	(1)
§ 1.2 C语言程序的格式和结构特点	(3)
一、C语言程序的格式	(4)
二、C语言程序的结构特点	(6)
§ 1.3 C语言程序的开发过程	(7)
§ 1.4 简单的输入与输出	(9)
一、格式化输入输出函数.....	(10)
二、字符输入与输出函数.....	(12)
习题一	(13)
第二章 常量、变量和数组	(15)
§ 2.1 常量.....	(15)
一、数.....	(15)
二、字符常量.....	(17)
三、字符串常量.....	(17)
四、换码序列.....	(18)
五、符号常量.....	(19)
§ 2.2 变量及其数据类型.....	(20)
一、变量和变量的地址.....	(21)
二、变量的数据类型.....	(21)
三、数据类型的转换.....	(23)
§ 2.3 变量的存储类型和使用范围.....	(24)
一、变量的存储类型.....	(24)
二、变量的使用范围.....	(24)
§ 2.4 变量的初始化.....	(27)
§ 2.5 数组.....	(29)
§ 2.6 字符型数组与字符串.....	(31)
§ 2.7 多维数组.....	(32)
习题二	(34)
第三章 运算和运算表达式	(37)
§ 3.1 赋值和算术运算.....	(37)
一、简单赋值.....	(37)
二、二项算术运算.....	(38)
三、算术赋值运算.....	(39)
§ 3.2 单项算术运算.....	(39)

一、单项算术运算	(39)
二、前置运算和后置运算	(40)
§ 3.3 关系运算和逻辑运算	(41)
一、关系运算	(41)
二、逻辑运算	(41)
§ 3.4 位操作	(42)
一、位逻辑运算	(43)
二、移位操作	(44)
三、位操作赋值运算	(46)
§ 3.5 三项条件运算和逗号结合运算	(46)
一、三项条件运算	(47)
二、逗号结合运算	(47)
§ 3.6 运算顺序	(48)
习题三	(50)
第四章 流程控制	(53)
§ 4.1 结构化程序设计和流程控制	(53)
一、结构化程序设计	(53)
二、C 语言的流程控制语句	(54)
三、复合语句	(55)
四、空语句	(57)
§ 4.2 条件分支	(57)
一、if~else 分支	(58)
二、if 分支	(59)
三、条件分支嵌套	(59)
四、else if 结构	(61)
§ 4.3 开关分支	(62)
§ 4.4 while 循环	(65)
§ 4.5 for 循环	(69)
§ 4.6 do~while 循环	(71)
§ 4.7 多重循环	(74)
§ 4.8 循环的中途退出	(76)
一、break 语句	(76)
二、continue 语句	(80)
§ 4.9 goto 语句与标号	(81)
习题四	(84)
第五章 指针	(87)
§ 5.1 指针变量	(87)
§ 5.2 指针的说明和初始化	(89)
一、指针的说明	(90)
二、指针的初始化	(92)

§ 5.3 指针运算	(93)
一、指针的算术运算	(94)
二、指针的关系运算	(97)
三、指针的赋值运算	(97)
§ 5.4 指针与数组	(98)
§ 5.5 字符指针与字符串	(101)
§ 5.6 指针数组	(104)
§ 5.7 多级指针	(107)
§ 5.8 命令行参数	(110)
习题五	(112)
第六章 函数	(114)
§ 6.1 模块化软件与 C 程序的模块结构	(114)
一、模块化软件及其优越性	(114)
二、C 程序的模块结构	(115)
§ 6.2 函数的定义和说明	(117)
一、函数的存储类型和数据类型	(117)
二、函数的定义	(117)
三、函数的说明和调用	(119)
§ 6.3 函数间的参数传递	(121)
一、数据复制方式传递数据	(122)
二、地址传送方式传递数据	(123)
三、利用参数返回结果	(125)
§ 6.4 使用返回值和全局变量传递数据	(126)
一、使用返回值传递数据	(127)
二、使用全局变量传递数据	(128)
§ 6.5 数组在函数间的传递	(129)
§ 6.6 字符串在函数间的传递	(131)
§ 6.7 指针型函数	(134)
§ 6.8 递归函数	(137)
§ 6.9 指向函数的指针	(139)
习题六	(142)
第七章 结构、联合和枚举	(145)
§ 7.1 结构的定义和说明	(145)
一、结构的定义	(146)
二、结构的说明	(147)
三、sizeof 运算	(148)
§ 7.2 结构体的使用特性和初始化	(149)
一、结构体的使用特性	(150)
二、结构体的初始化	(152)
§ 7.3 结构数组	(152)

§ 7.4 结构指针	(155)
§ 7.5 结构体在函数间的传递	(158)
一、结构体复制方式	(158)
二、结构体地址传送方式	(160)
三、结构数组在函数间的传递	(161)
§ 7.6 结构型和结构指针型函数	(163)
一、结构型函数	(163)
二、结构指针型函数	(164)
§ 7.7 结构嵌套	(166)
§ 7.8 位字段结构体	(169)
§ 7.9 联合体	(173)
§ 7.10 枚举型和标识常量	(176)
一、枚举型	(176)
二、标识常量	(179)
§ 7.11 类型定义	(180)
习题七	(182)
第八章 文件的输入和输出	(185)
§ 8.1 文件的概念	(185)
一、文件的概念	(185)
二、标准设备文件的换向	(187)
§ 8.2 标准设备文件输入输出函数	(188)
一、字符输入输出函数	(188)
二、字符串输入输出函数	(189)
§ 8.3 格式化输入输出函数	(190)
一、格式化输出函数 printf()	(190)
二、格式化输入函数 scanf()	(192)
§ 8.4 文件的打开和关闭	(193)
一、文件打开函数 fopen()	(194)
二、文件关闭函数 fclose()	(195)
§ 8.5 文件的字符输入和输出	(195)
一、字符输入函数 getc()	(195)
二、字符输出函数 putc()	(196)
§ 8.6 文件的字符串输入和输出	(199)
一、字符串输入函数 fgets()	(199)
二、字符串输出函数 fputs()	(200)
§ 8.7 文件的格式化输入和输出	(201)
一、格式化输出函数 fprintf()	(202)
二、格式化输入函数 fscanf()	(204)
§ 8.8 文件的随机读写	(205)
一、文件位置指针	(205)

二、文件指针设置函数 fseek()	(206)
三、文件指针位置检测函数 ftell()	(208)
习题八	(210)
第九章 常用标准函数	(211)
§ 9.1 字符和字符串处理函数	(212)
一、字符分类函数	(212)
二、字符串处理函数	(214)
三、数字字符串转换函数	(216)
§ 9.2 存储分配函数	(217)
§ 9.3 数学函数	(223)
§ 9.4 系统调用函数	(228)
一、ROM-BIOS 系统调用	(228)
二、DOS 系统调用	(231)
§ 9.5 图形函数	(234)
习题九	(238)
第十章 编译预处理和分割编译	(240)
§ 10.1 宏定义	(240)
一、符号常量	(240)
二、带参数的宏	(242)
三、局部宏定义	(243)
§ 10.2 文件包括	(244)
§ 10.3 条件编译	(246)
§ 10.4 分割编译	(250)
习题十	(253)
附录 A ASCII 字符代码表	(254)
附录 B ANSI 与 K&R 的主要差异	(255)
参考资料	(258)

概 述

随着电子计算机的迅速发展和广泛应用,C 程序设计语言(简称 C 语言)在计算机软件开发中的作用日益显著。当前几乎各种型号的微型计算机和 workstation,以及某些大中型计算机上都配置有 C 语言编译系统。尤其在我国目前广泛使用的 IBM-PC 及其兼容机系列的各种档次(8086、80286、80386 和 80486 等)微机上,都配备了多种版本的 C 语言编译系统。这样良好的开发环境为 C 语言的广泛应用开辟了美好的前景。

本章简要介绍 C 语言的发展和特点,C 语言程序的格式和结构特点,及其开发过程。此外,对 C 语言程序中经常使用的输入输出给予简单介绍。

§ 1.1 C 语言的发展和特点

电子计算机,自从 1946 年问世以来,随着其应用领域的迅速扩大,硬件和软件都取得了高度的发展。作为计算机软件的基础,程序设计语言也得到不断充实和完善。功能全面、使用便利的程序语言相继问世。在种类繁多的计算机程序语言家族中,70 年代初期又增添了一名新成员——C 语言。

C 语言是一种编译型程序语言,它的前身是马丁·理查德(Martin Richards)在 60 年代开发的 BCPL 语言。BCPL 语言是计算机软件人员在开发系统软件时,作为记述语言使用的一种结构化程序设计语言。它能够直接处理与机器本身数据类型相近的数据,具有与内存地址对应的指针处理方式。

1970 年,肯·苏姆普逊(Ken Thompson)在软件开发工作中,继承和发展了 BCPL 语言的上述特点,进而提出了“B 语言”。当时最新型的小型计算机,美国 DEC 公司的 PDP-7 型机中的操作系统,就是使用 B 语言记述和开发的。

此后,在美国贝尔研究所进行的更新型的小型机 PDP-11 的 UNIX 操作系统的开发工作中,戴尼斯·M·利奇(Dennis M. Ritchie)和布朗·W·卡尼汉(Brian W. Kernighan)对 B 语言

做了进一步的充实和完善,于1972年推出了一种新型的程序语言——C语言。因此,早期的C语言又称为K&R C语言。

C语言和其它高级语言的发展关系如图1.1所示。

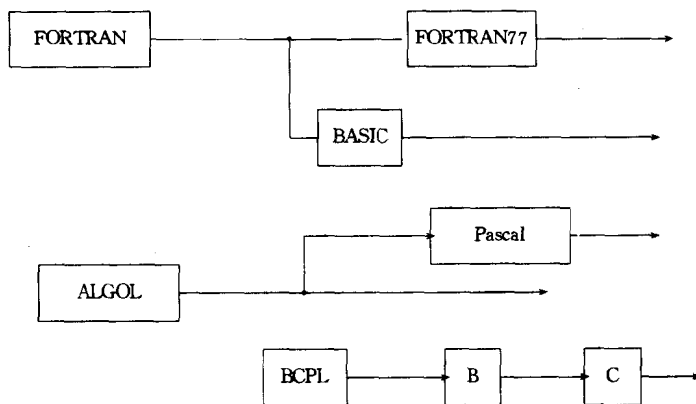


图1.1 C语言的发展历史

在K&R C语言出现之后的十几年中,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语言的应用推向了一个更加广泛普及的阶段。在C语言的实际应用中,它又得到了进一步的发展,其性能不断提高,功能逐步完善。1987年美国国家标准组织推出了比K&R C功能更完善的现代标准C语言——ANSI C。近几年出现的各种C编译系统版本,如Microsoft C 5.0~6.0、Turbo C 2.0等都是与ANSI C兼容的新版本,它们称为现代C语言。

不同版本的现代C编译系统,它们的语法和语句功能是一致的。它们之间的差异主要体现在它们各自的标准函数库中收纳的函数的种类、格式和功能上稍有差别。本书以当前最新的美国标准C语言ANSI C为基础,同时兼顾现代C语言编译系统不同版本中通用性、一致性的内容予以叙述。

进入80年代,面向对象程序设计的概念日益普及。C语言在发展的同时,也朝着成为支持面向对象的程序设计语言的方向迈出了步伐。1986年美国AT&T的贝尔研究所推出了C语言的超集——C++语言。它成为对大型和复杂软件开发的具有吸引力的语言。由于C++语言是以C语言为基础的,所以在学习C++语言之前,必须全面掌握C语言的概念和程序设计方法。

C语言在当前成为软件开发中主流程序语言,是由它的特点所决定的。C语言的主要特点如下:

1. C语言是处于汇编语言和高级语言之间的一种记述性程序语言。C语言在程序语言世界中所处的位置,如图1.2所示。可以看出,C语言是比较靠近硬件与系统的,即与汇编语言比较接近。C语言既有面向硬件和系统,象汇编语言那样可以直接访问硬件的功能,

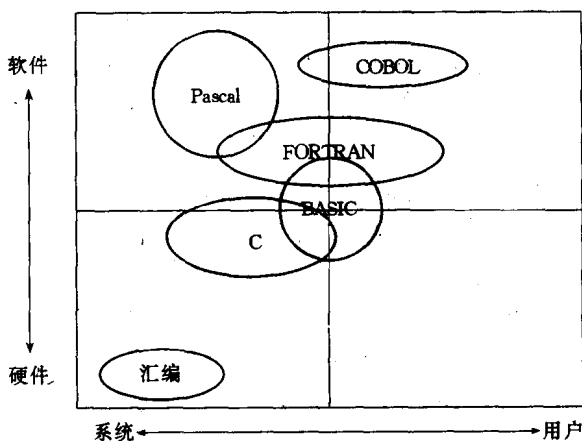


图1.2 C语言和其它语言的关系

又有高级语言面向用户、容易理解、便于阅读和书写的优点。

2. C语言是一种结构化程序设计语言,即程序的逻辑结构可以用顺序、选择和循环三种基本结构组成。C语言具有诸如 if~else、for、while、do~while、switch~case 等结构化语句,十分便于采用由顶向下、逐步细化的结构化程序设计技术。因此,用C语言编制的程序,具有容易理解、便于维护的优点。

3. C语言是便于模块化软件设计的程序语言。C语言程序的函数结构,十分利于把整体程序分割成若干个功能相对独立的程序模块,并且为程序模块间的相互调用以及数据传递提供了便利。这一特点也为把大型软件模块化,由多人同时进行集体性开发等软件工程技术方法提供了强有力的支持。

4. C语言具有与内存地址对应的指针处理方式,可以直接处理内存中各种类型的数据。指针的使用为C语言程序进行数据处理,以及数据在程序模块中的传递提供了有力的支持,并为控制和管理系统硬件及接口提供了便利的手段。

5. C语言备有种类丰富的运算符。除一般高级语言使用的+、-、*、/四则运算及与(AND)、或(OR)、非(NOT)等逻辑运算功能外,还可以实现以二进制位(bit)为单位的位与、位或、位非、位异或以及移位操作等位运算,并且具有如 a++、b--等单项运算和 +=、-=、*=、/=等复合运算功能。

6. C语言的数据类型丰富。基本类型有字节型(8bit)、单精度整数(16bit)、双精度整数(32bit)和单精度与双精度实数型。同时,它还具有结构和联合两种复合数据类型,利用它们可以便利地处理包含着不同数据类型的复杂数据结构。因此,C语言具有较强的数据处理能力。

7. C语言程序中可以使用如 #define、#include 等编译预处理语句,能够进行字符串或特定参数的宏定义,以及实现对外部文本文件的读取和合并。同时还具有 #if、#else 等条件编译预处理语句。这些功能的使用提高了软件开发的工作效率,并为程序的组织和编译提供了便利。

8. C语言具有较高的移植性。C语言的语句中没有依存于硬件的输入/输出语句,程序的输入/输出功能是通过调用输入/输出函数实现的。而这些函数是来自于由系统提供的独立于C语言的程序模块库。因此,C语言程序本身并不依存于机器硬件系统,从而便于在硬件结构不同的机种间实现程序的移植。

由于C语言具有上述诸多特点,近年来迅速地得到广泛普及和应用。C语言被称为“高级汇编语言”,特别是在微处理机和微型计算机的软件开发、以及各种软件工具的开发中,使用C语言的趋势日益增强。最近呈现出C语言在某种程度上有可能取代汇编语言的发展倾向。

§ 1.2 C语言程序的格式和结构特点

任何一种计算机程序语言,都具有特定的语法规则和一定的表现形式。程序的书写格式和程序的构成规则是程序语言表现形式的一个重要方面。按照规定的格式和构成规则书写程序,不仅可以使程序设计人员和使用程序的人员容易理解,更重要的是,把程序输入给计算机时,计算机能够充分认识它,从而能够正确执行它。

一、C 语言程序的格式

在介绍 C 语言程序格式之前,让我们首先回顾一下读者较熟悉的 BASIC 程序的格式,然后通过两种语言程序的比较,读者就会更直观地了解 C 语言程序的格式特点。

使用 BASIC 语言,编制计算半径为 R、高度为 H 的圆柱体体积的程序。要求 R 和 H 的数值由键盘输入。该程序如下:

```
10 INPUT R,H ✓  
20 V=3.14159 * R * R * H ✓  
30 PRINT "V=";V ✓  
40 END ✓
```

用 C 语言编写的具有同样功能的程序如例 1.1。

例 1.1 圆柱体体积计算程序

```
main()  
{  
    int r,h;  
    float v;  
  
    scanf("%d%d",&r,&h);  
    v=3.14159 * r * r * h;  
    printf("V=%f\n",v);  
}
```

这里,我们暂且不必顾及 C 语言程序中各个语句的功能,首先把注意力放在其格式特点上。比较上面两个程序,可以看出,C 语言程序有以下若干格式特点。

1. C 语言程序习惯上使用小写英文字母,这与 BASIC 程序使用大写字母恰恰相反。C 语言程序中也可以用大写字母,但它们常常是作为常量的宏定义和其它特殊用途使用。

2. 和 BASIC 程序一样,C 语言程序也是由一个个语句组成的。每个语句都具有规定的语法格式和特定的功能。通过上面的 C 语言程序与 BASIC 程序对照,可以得知,scanf 是用于输入变量数值的语句。此外,还有体积 v 的计算语句 $v=3.14159 * r * r * h$; 和输出语句 printf。

3. 和 BASIC 语言不同,C 语言程序不使用行序号。但在本书中,为了解说方便,有时在给出的程序实例中加有序号。需要注意的是,向计算机输入这些程序时,不要输入这样的行序号。

4. BASIC 程序中以回车符 ✓ 作为一行语句的终止符,以冒号:作为语句间的分隔符。与此对应,C 语言程序使用分号;作为语句的终止符或分隔符。对于初学者,这点常常被忽视,请务必予以注意。

5. 一般情况下,每个语句占用一个书写行的位置。更准确地说,C 语言程序不存在程序行的概念。一个程序可以自由地使用任意的书写行,即一行中可以有多个语句,一个语句也可以占用任意多行。但需要注意,语句之间必须用;分隔。作为极端情况,上面的程序按下列格式书写仍是正确的:

```
main(){int r,h;float v;scanf("%d%d",&r,&h);  
v=3.14159 * r * r * h;printf("V=%f\n",v);}
```

把它输入给计算机时,仍然能够正确接收和识别。

6. C 语言程序中用大括弧对 {} 表示程序的结构层次范围。一个完整的程序模块要用一对大括弧表示该程序模块的范围,如上面程序中的第二行和最后一行的大括弧对。此外,程序体中若干结构化语句,如 if、for、while、switch、else、do 等,常常是由若干语句组成的语句块,这样的语句块也要求用大括弧对包围,以表示该结构的范围(可参看后面的程序例)。

综上所述,C 语言程序的书写格式自由度较高、灵活性很强,有较大的任意性。但是,为了避免程序书写的层次混乱不清,便于人们阅读和理解,一般都采用有一定格式的习惯书写方式。这样的书写格式并不是计算机要求的,而是为了给程序人员提供便利。本书采用一种使用较多的书写格式。配合说明,下面再给出一个程序实例。程序的功能请暂且不必考虑,目前仅着眼于它的格式方面。

例 1.2 统计输入文件中,行、单词和字符数量的程序(说明格式用)

```
#include <stdio.h>
main()
{
    int c,nl,nw,nc,inword;

    inword=0;
    nl=nw=nc=0;
    while((c=getchar())!=EOF)
    {
        nc++;
        if(c=='\n')
            nl++;
        if(c==' '||c=='\t'||c=='\n')
            inword=0;
        else if(inword==0)
        {
            inword=1;
            nw++;
        }
    }
    printf("line=%d word=%d character=%d\n",nl,nw,nc);
}
```

参照上述程序,可以归纳出这种书写格式的要点是:

1. 一般情况下,每个语句占用一行。
2. 不同结构层次的语句,从不同的起始位置开始,即在同一结构层次中的语句,缩进同样的字数。如程序例中 while 和 if、else 语句,其结构中的各个语句,都缩进相同位置。向计算机输入 C 语言源程序时,一般使用 TAB 键调整各行的起始位置。
3. 表示结构层次的大括弧,写在该结构化语句第一个字母的下方,与结构化语句对齐,并占用一行,如 while 下方的 { 和倒数第三行的 } 是表示 while 结构范围的大括弧对。同样,else 下方的大括弧对也是如此。
4. 为了增强可读性,程序中适当地加些空格和空行。例如,上例中在第 4 和第 5 行之

间加有一空行,表示其上下是程序的两个部分。但是,程序中各种标识名字(变量名、函数名等)和 C 语言本身使用的单词(称为保留字,详见附录 B),不能在中间插入空格。

采用这样格式书写的程序,结构层次清晰,充分体现了结构化程序的特点,十分便于阅读和理解。除了这里介绍的书写格式外,还有一些其它的常用书写格式,但都大同小异。感兴趣的读者可以参考其它有关资料,这里不再赘述。

二、C 语言程序的结构特点

一个完整的 C 语言程序是由一个或多个具有相对独立功能的程序模块结合而成,这样的程序模块称为函数。从格式上看,每个函数是由函数名和大括号对 {} 包围的若干语句组成的,为了更直观地了解 C 语言程序的结构特点,下面再给出一程序例子。

例 1.3 C 语言程序的结构特点

```
1: /* print string as uppercase */
2: #include <stdio.h>
3: #define SIZE 80
4: main()
5: {
6:     char str[SIZE];
7:     int putupper(char ch);
8:     int i;
9:
10:    gets(str);
11:    for(i=0;str[i]!='\0';i++)
12:        putupper(str[i]);
13: }
14: int putupper(char ch)
15: {
16:     char cc;
17:
18:     cc=(ch>='a'&&ch<='z')? ch+'A'-'a':ch;
19:     putchar(cc);
20: }
```

可以看出,上述程序是由名字称为 main 和 putupper 的两个函数组成。在组成 C 语言程序的函数中,必须有一个且只能有一个名字为 main 的函数,它叫做主函数。除主函数之外的函数由用户命名,如上例的 putupper 函数。

C 语言程序的执行是从主函数开始的,主函数中的所有语句执行完毕,则程序执行结束。如例 1.3 的程序是从第 5 行的 { 开始,执行到第 13 行的 } 结束。在执行第 12 行的语句时,程序控制转移到函数 putupper 中。执行完函数 putupper 的语句后,再返回主函数中继续运行。这个转移过程叫做调用函数 putupper。因此看出,C 语言程序中,main 函数之外的其它函数都是在执行 main 函数时,通过嵌套调用而得以执行的。在程序中除了可以调用用户自己编制的函数外,还可以调用由系统提供的标准函数,如例 1.1 中的 printf 和 scanf,以及例 1.3 中的 gets 和 putchar 等都是标准函数。

程序前部带有 # 号的语句是编译预处理语句,如例 1.3 中第 2 行和第 3 行的语句。关于它

们的作用在第十章中详细讨论。

和其它程序语言一样,C 语言程序也可以使用注释。注释部分的格式是:

```
/* 注释内容 */
```

如例 1.3 中的第 1 行就是注释部分。注释可以加在程序的任意部位,它可以占用一行以上的位置,也可以写在语句的后面。注释部分作为源程序的一个部分驻留在源程序清单中,但对源程序进行编译时,系统将无视注释部分。

C 语言程序中函数的一般形式如下所示:

```
函数名(形式参数说明)
{
    数据说明语句部分;
    执行语句部分;
}
```

函数名后的圆括号中包括着形式参数说明,如上例第 14 行中函数 putupper 的形式参数说明是 char ch。当函数不带参数时,函数名后的圆括号也不能缺省。函数名下面大括号包围的就是该函数的函数体。函数体一般由数据说明语句和执行语句两部分组成。数据说明语句部分用于说明函数中使用的数据及其属性等。执行语句部分是完成函数功能的执行语句,它们有三种:运算表达式语句、函数调用语句和流程控制语句。如上例中,main()函数中第 6、7 和 8 行是数据说明语句,第 10、11 和 12 行是执行语句。putupper 函数中第 16 行是数据说明语句,第 18 和 19 行是执行语句。

C 语言程序的函数模块结构特点,使得程序整体结构分明、层次清楚,它为模块化软件设计方法提供了有力的支持。关于程序各部分的意义及功能,后面各章中将陆续讨论。

§ 1.3 C 语言程序的开发过程

C 语言是一种编译型程序语言,和 FORTRAN、Pascal 等编译语言相同,C 语言程序的开发过程要经历四个基本开发步骤:

编辑—编译—链接—运行

如图 1.3 所示。

1. 源文件的编辑

程序开发人员把自己编制的 C 语言程序输入给计算机时,首先要使用系统提供的编辑程序(又称为编辑器)建立 C 语言程序的源文件。建立后的源文件以文本文件形式存储在计算机的文件系统(软盘或硬盘)中。源文件的名字由用户选定。C 语言源文件名字要求有一定的后缀“.c”。例如 test.c、program.c 等。

用于建立源文件的编辑程序种类很多,如 UNIX 下的 vi、ed,MS-DOS 下的 EDLIN 等,也可以使用功能更强的专用编辑软件,如 Personal Editor 和 WordStar 等。

2. 编译过程

在对源文件进行编译前,要再次进行检查和确认。在 UNIX 下用 cat 命令,MS-DOS 下用 type 命令可以查看源文件的内容。发现问题后,则重新启动编辑程序对源文件进行修改。确认

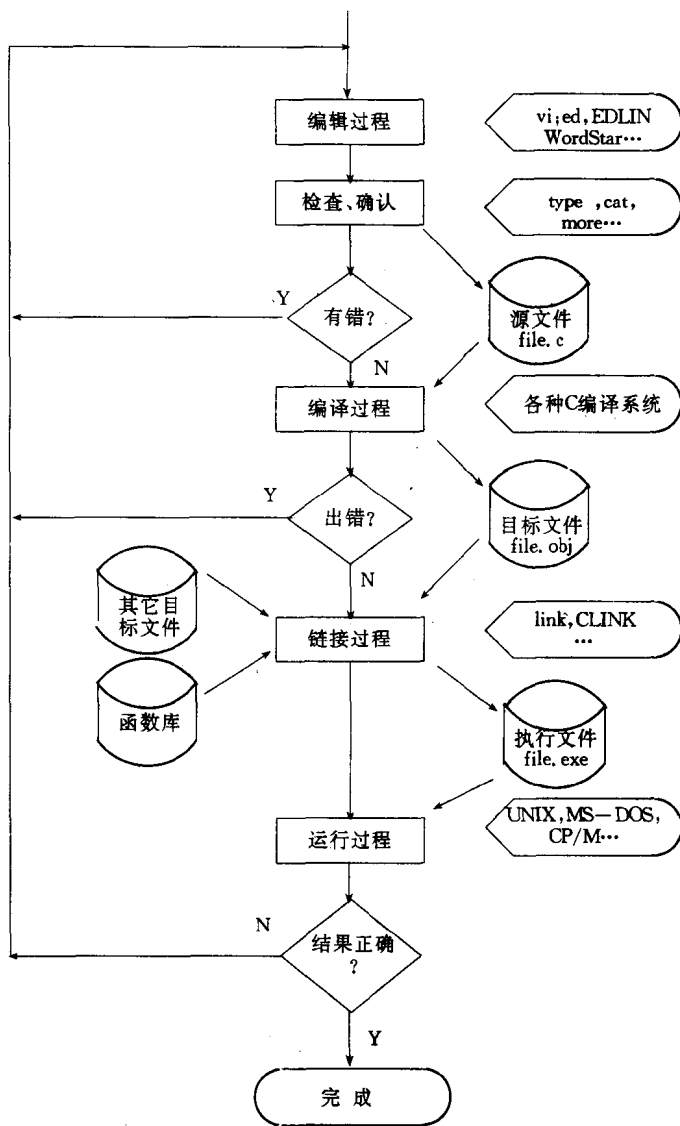


图1.3 C语言程序的开发过程

正确后,则开始对源文件进行编译。

编译过程是使用系统提供的编译程序(又称为编译器)进行的。不同版本的C编译系统中编译程序的启动命令不同。程序人员应该按照有关技术手册提供的方法,正确使用编译命令完成编译过程。

系统对源文件进行编译的同时,还对源程序的语法和程序的逻辑结构等进行检查。当发现错误时,就在显示器上列出错误的位置和种类。此时要再次使用编辑程序对源文件进行排错修正。

正确的源文件经过编译后生成目标文件,存储于文件系统中。目标文件带有规定的后缀,在UNIX下为.o,MS-DOS下为.obj,CP/M下为.CRL等。

3. 链接过程

编译后产生的目标文件是浮动的程序模块,不能直接用于运行。目标文件要经过链接过程,把它和其它目标程序模块,以及系统提供的标准函数库等进行链接后,才生成可以运行的