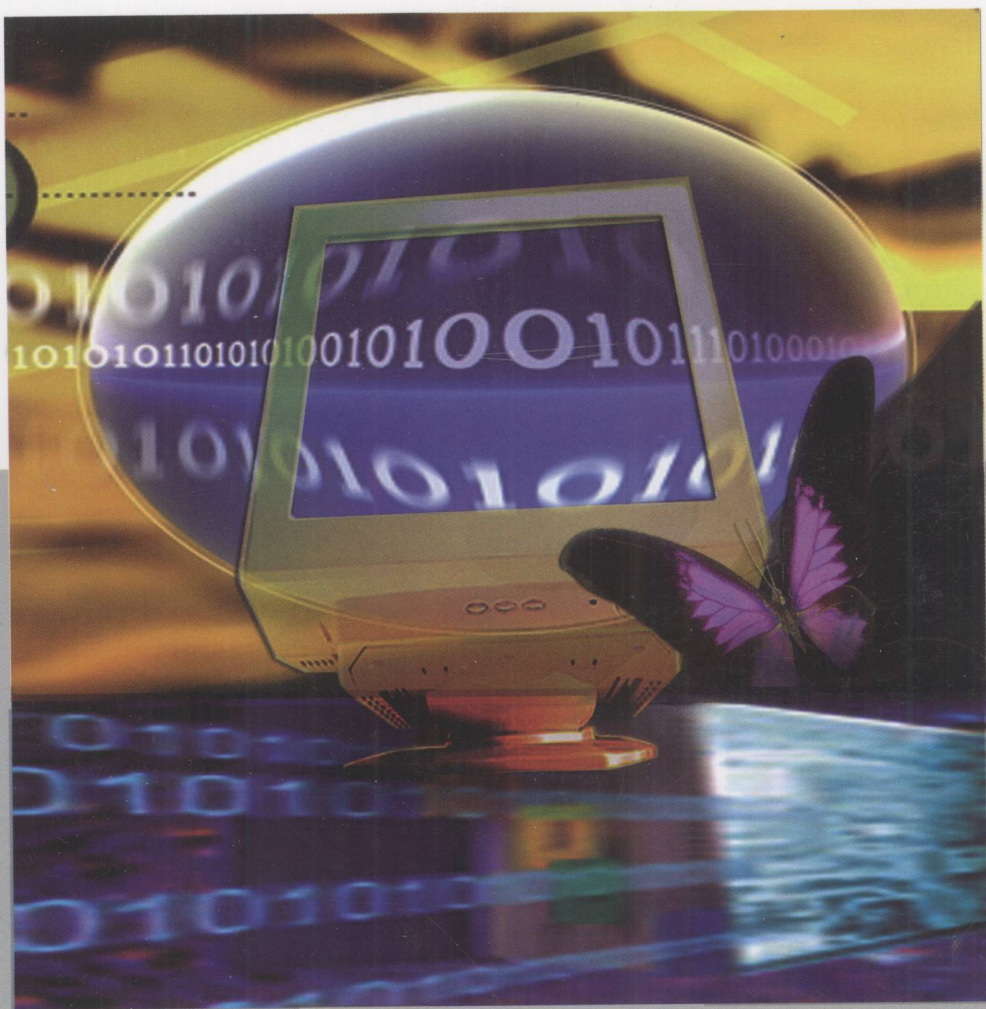


新世纪高等学校计算机系列教材

C语言程序设计

雷于生 胡成松 主编



高等教育出版社
中山大学出版社

策划:湖北省计算机学会·武汉高联教科文中心

新世纪高等学校计算机系列教材

C 语言程序设计

主 编	雷于生	胡成松	
副主编	彭文艺	程海英	韩丽娟
	鲁 力	定 会	

高等教育出版社·北京

中山大学出版社·广州

图书在版编目(CIP)数据

C 语言程序设计/雷于生,胡成松主编. —北京:高等教育出版社. —广州:中山大学出版社, 2009. 7
(新世纪高等学校计算机系列教材/湖北省计算机学会·武汉高联教科文中心策划)

ISBN 978-7-306-03267-6

I. C… II. ①雷… ②胡… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 019974 号

内 容 提 要

本书从应用的角度出发,较全面、系统地介绍了 C 语言程序设计的基本原理、基本方法和基本技术。其内容包括:C 语言概述,C 语言语法基础,顺序结构、选择结构、循环结构、函数、数组、指针、结构与联合等程序设计方法和技术,以及编译预处理和文件操作等。本书对基本概念、基本原理与基本应用的介绍,重点突出,主次分明;书中所有例题、习题都通过精选,并兼顾等级考试。

本书适合应用型高等学校计算机本科各专业作为教材使用,在适当减少部分内容或适当减少部分例题讲解的情况下,也适合理工类本科其他各专业及专科作为教材使用。

(注:凡需要本书或其电子原稿备课者,可与执行编委唐元瑜老师联系(027—87807919,13907198295))

版权所有 盗印必究

C 语言程序设计

© 雷于生 胡成松 主编

出版人:叶侨健

责任编辑:李文

责任校对:赖艳艳

策划编辑:李文 唐源

封面设计:曹巩华 梁希

责任技编:潘 隆

出版发行:高等教育出版社(地址:北京市西城区德外大街 4 号)

邮编:100011)

中山大学出版社(地址:广州市新港西路 135 号)

邮编:510275)

经 销:广东新华发行集团公司

武汉高联科教信息有限公司(电话:027—87807752 87807919(带传真) 邮编:430074)

印 刷:安陆市鼎鑫印务有限责任公司

开 本:787mm×1092mm

1/16

印 张:17.75

字 数:450 千字

版 次:2009 年 7 月第 1 版

印 次:2009 年 7 月第 1 次印刷

印 数:1—4 000 册

定 价:29.00 元

前 言

C 语言是目前全球范围内流行和使用最为广泛的一种程序设计语言。C 语言具有语言简洁紧凑、目标代码质量高、语言表达能力强、流程控制结构化、弱类型、“中级语言”特性、书写自由灵活,以及可移植性好等一系列优点,它既具有面向过程的特性,又具有汇编语言的特点。因此,C 语言被广泛应用于各类系统软件和应用软件的开发,它已是任何一位希望迈向信息及相关领域殿堂的人们所必需掌握的一种程序设计语言。

C 语言语法规则多、概念复杂、用法灵活,初学者较难于理解和掌握,因而学习难、深入难是广大初学者的共同感受。针对这些问题,本书按照紧扣基础、面向应用、强调算法的原则,力图将 C 语言复杂的概念通俗化,内容及文字叙述条理化,在使之适合初学者学习和掌握的前提下,以标准 C 为基本框架,较全面地介绍了使用 C 语言进行程序设计的基本理论、方法和技术,为读者进一步学习或使用程序设计技术打下较坚实的基础。

全书共分 11 章。其中,第 1 章是 C 语言程序设计概述;第 2 章是 C 语言程序的语法基础;第 3~5 章分别介绍顺序结构、选择结构、循环结构程序设计;第 6~8 章分别较详细地介绍数组、函数和指针程序设计及其应用;第 9~11 章分别介绍编译预处理、结构与联合、文件。同时,为了满足实践教学和学生课后练习的需要,还出版了与本教材配套使用的《C 语言程序设计实验与习题集》一书(彭文艺,韩丽娟编)。

本书的特点是:从应用角度出发,循序渐进,由浅入深,通俗易懂;对基本概念、基本原理与基本应用的介绍,其重点突出,主次分明;同时,书中所有的例题、习题都是作者精选的,并兼顾等级考试。

本书是编者们在总结多年来从事 C 语言教学实践经验的基础上,并参考国内外其他有关资料而写成的。它既适合应用型高等学校计算机本科各专业作为教材使用,在适当减少部分内容和部分例题讲解的情况下,也适合理工类本科其他各专业或专科作为教材使用。

本书由雷于生和胡成松老师主编,具体撰写各章的老师是:雷于生(第 1、9 章和附录),胡成松(第 6、11 章),彭文艺(第 7 章),程海英(第 2、3 章),韩丽娟(第 8 章),定会(第 4、5 章),鲁力(第 10 章)。雷于生老师(教授)还负责了本书大纲的制定和统稿、审定与部分修改工作,以及有关的组织工作。

本书在编写过程中,得到了华中科技大学、华中科技大学武昌分校、中南

民族大学工商学院的有关领导，以及湖北省计算机学会和《新世纪高等学校计算机系列教材》编审指导委员会有关专家的大力支持与帮助，在此一并致谢。

由于编者水平有限，书中难免有疏漏及不妥之处，殷切希望广大读者批评指正。

编 者

2009 年 3 月于武汉

目 录

第 1 章 C 语言程序设计概述	(1)
1.1 程序设计的基本概念	(1)
1.1.1 程序和程序设计	(1)
1.1.2 程序设计语言与程序设计方法	(3)
1.2 C 语言的产生、发展及标准化	(5)
1.3 C 语言的体系结构及特征	(6)
1.3.1 C 语言的体系结构	(6)
1.3.2 C 语言的语言特征	(7)
1.4 C 语言程序的基本组成	(9)
1.5 算法及其表示	(11)
1.5.1 算法的基本概念	(11)
1.5.2 算法的特性及组成要素	(14)
1.5.3 算法的描述及实现	(15)
1.6 学习 C 语言程序设计的方法	(18)
习题一	(20)
第 2 章 C 语言程序的语法基础	(21)
2.1 C 语言的字符集和标识符	(21)
2.1.1 字符集	(21)
2.1.2 标识符、关键字及分隔符	(21)
2.2 C 语言的基本数据类型	(23)
2.2.1 数据类型的分类	(23)
2.2.2 基本数据类型的名词及长度	(24)
2.3 常量与变量	(26)
2.3.1 整型常量和整型变量	(29)
2.3.2 实型常量和实型变量	(31)
2.3.3 字符常量和字符变量	(33)
2.3.4 字符串常量	(36)
2.4 算术运算符和算术表达式	(36)
2.5 赋值运算符和赋值表达式	(41)
2.6 关系运算符和关系表达式	(43)
2.7 逻辑运算符和逻辑表达式	(45)
2.8 条件运算符和条件表达式	(47)
2.9 逗号运算符和逗号表达式	(48)
2.10 位运算符和位运算表达式	(49)

习题二	(51)
第 3 章 顺序结构程序设计	(54)
3.1 C 程序的基本语句	(54)
3.1.1 表达式语句	(54)
3.1.2 复合语句	(55)
3.1.3 控制语句	(57)
3.1.4 函数调用语句	(58)
3.1.5 空语句	(58)
3.2 赋值语句	(58)
3.3 数据的基本输入/输出	(59)
3.3.1 字符数据的输入/输出	(60)
3.3.2 格式标准输入/输出	(62)
3.4 顺序结构程序设计	(73)
习题三	(75)
第 4 章 选择结构程序设计	(78)
4.1 if 语句	(78)
4.1.1 if 语句的三种形式	(78)
4.1.2 if 语句的嵌套	(83)
4.2 条件运算符和表达式	(85)
4.3 switch 语句	(86)
4.4 选择结构程序设计举例	(90)
习题四	(94)
第 5 章 循环结构程序设计	(97)
5.1 while 循环语句	(97)
5.2 for 循环语句	(99)
5.3 do-while 循环语句	(101)
5.4 循环的嵌套	(103)
5.5 break 和 continue 语句	(107)
5.6 goto 语句与 if 语句构成的循环	(109)
5.6.1 goto 语句和标号	(109)
5.6.2 goto 语句与 if 语句构成的循环	(109)
5.7 循环结构程序设计举例	(110)
习题五	(113)
第 6 章 数组	(118)
6.1 一维数组	(118)
6.1.1 一维数组的定义	(118)
6.1.2 一维数组元素的引用	(119)

6.1.3	一维数组元素的初始化	(120)
6.1.4	一维数组应用举例	(120)
6.2	二维数组	(124)
6.2.1	二维数组的定义	(124)
6.2.2	二维数组元素的引用	(125)
6.2.3	二维数组元素的初始化	(125)
6.2.4	二维数组应用举例	(126)
6.3	字符数组和字符串	(129)
6.3.1	字符数组的定义	(129)
6.3.2	字符数组的初始化	(129)
6.3.3	字符数组元素的引用	(130)
6.3.4	字符串的存储	(131)
6.3.5	字符数组的输入/输出	(131)
6.3.6	字符串处理函数	(133)
6.3.7	字符数组程序设计举例	(135)
习题六		(136)
第7章	函数	(138)
7.1	C程序的一般结构	(138)
7.1.1	结构化程序设计的概念	(138)
7.1.2	C程序的一般结构	(139)
7.2	函数的分类和定义	(139)
7.2.1	函数的分类	(139)
7.2.2	函数的定义	(140)
7.3	函数的调用	(143)
7.3.1	函数调用的一般形式	(143)
7.3.2	函数调用的方式	(143)
7.3.3	函数调用的执行过程	(144)
7.3.4	被调用函数的声明和函数原型	(146)
7.4	数组作为函数参数	(147)
7.4.1	数组元素作函数实参	(147)
7.4.2	数组名作为函数参数	(149)
7.5	函数的嵌套与递归调用	(154)
7.5.1	函数的嵌套调用	(154)
7.5.2	函数的递归调用	(156)
7.6	局部变量和全局变量	(161)
7.6.1	局部变量	(161)
7.6.2	全局变量	(163)
7.7	变量的存储类别	(164)
7.7.1	动态存储方式变量与静态存储方式变量	(164)
7.7.2	auto 变量	(164)

7.7.3 用 static 声明局部变量	(165)
7.7.4 register 变量	(166)
7.7.5 用 extern 声明外部变量	(166)
7.8 内部函数和外部函数	(168)
习题七	(169)
第 8 章 指针	(173)
8.1 指针与地址的概念	(173)
8.2 指针变量的定义与引用	(174)
8.2.1 指针变量的定义	(174)
8.2.2 指针变量的引用及运算	(175)
8.2.3 指针变量作为函数参数	(178)
8.3 指针与数组	(181)
8.3.1 指向一维数组的指针	(181)
8.3.2 指向二维数组的指针	(185)
8.4 指针与字符串	(188)
8.4.1 字符串与指向字符串的指针	(188)
8.4.2 使用字符串指针变量与字符数组的区别	(189)
8.5 指向函数的指针	(193)
8.6 指针型函数	(195)
8.7 指针数组与指向指针的指针	(197)
8.7.1 指针数组的概念	(197)
8.7.2 指向指针的指针	(199)
8.7.3 main 函数的参数	(201)
习题八	(202)
第 9 章 编译预处理	(204)
9.1 无参宏定义	(204)
9.2 带参数的宏定义	(206)
9.3 取消宏定义	(208)
9.4 文件包含 #include	(209)
9.5 条件编译	(211)
习题九	(214)
第 10 章 结构与联合	(217)
10.1 结构类型和结构变量	(217)
10.1.1 结构类型的声明	(217)
10.1.2 结构变量的定义与初始化	(218)
10.1.3 结构类型的引用	(221)
10.2 结构数组	(221)
10.2.1 结构数组的定义与初始化	(221)

10.2.2 结构数组元素的引用	(222)
10.3 指向结构类型数据的指针	(224)
10.3.1 指向结构类型变量的指针	(224)
10.3.2 指向结构数组的指针	(227)
10.3.3 用结构变量和结构指针作为函数参数	(229)
10.4 用结构指针处理链表	(231)
10.4.1 静态数据结构和动态数据结构的概念	(231)
10.4.2 C 语言的动态存储分配函数	(231)
10.4.3 链表的建立和操作	(233)
10.5 联合和联合数据的引用	(243)
10.6 枚举类型定义和应用	(245)
10.7 类型定义 typedef	(246)
习题十	(248)
第 11 章 文件	(255)
11.1 C 语言中文件的基本概念	(255)
11.2 文件类型指针	(255)
11.3 文件的打开与关闭	(256)
11.3.1 文件的打开	(256)
11.3.2 文件的关闭	(257)
11.4 文件的读/写	(258)
11.4.1 fputc 函数和 fgetc 函数	(258)
11.4.2 fread 函数和 fwrite 函数	(260)
11.4.3 fprintf 函数和 fscanf 函数	(263)
11.5 文件的定位	(263)
11.5.1 rewind 函数	(263)
11.5.2 fseek 函数和随机读/写	(263)
11.5.3 ftell 函数	(264)
11.6 文件的错误检测	(265)
11.6.1 ferror 函数	(266)
11.6.2 clearerr 函数	(266)
习题十一	(266)
附录	(267)
附录 1 ASCII 字符编码表	(267)
附录 2 C 标准库函数	(268)
参考文献	(272)



C 语言程序设计概述

C 语言是目前全球流传最广、使用最多的程序设计语言之一。C 语言作为一种典型的结构化程序设计语言,不仅可用于应用程序和系统程序设计,而且它的许多先进的技术思想和方法被当前程序设计的主流——面向对象程序设计语言如 C++、JAVA、C# 语言所继承,或者说 C 语言是 C++、JAVA、C# 产生和发展的基础。因此,学习 C 语言既是对程序设计应用工具的学习,也是对程序设计技术基础的学习。

本章从程序设计的角度出发,介绍程序设计的基本概念、C 语言的产生和发展、C 语言的体系和特征、简单 C 语言程序的组成、算法及其表示,以及如何学习 C 语言程序设计。

1.1 程序设计的基本概念

程序是计算机的灵魂,程序设计贯穿于计算机理论研究与应用的各个环节中。尤其是在计算机应用领域里,程序设计是计算机应用的最核心任务。程序设计主要包含两方面的问题:一是对要处理的问题抽象出数据对象和处理对象的逻辑模式和方法;二是用程序设计语言描述成计算机能处理的代码序列。一个正确的、有效的程序的编写,必然与程序设计、程序设计方法和程序设计语言密切相关。因此,在学习 C 语言程序设计之前,介绍有关程序、程序设计、程序设计方法和程序设计语言的基本概念是很有必要的。

1.1.1 程序和程序设计

1. 程序的概念

计算机能够自动进行工作,是因为计算机可以执行存储在计算机内存中的程序。对于什么是计算机程序,首先看看人们在日常生活中对程序的理解。所谓程序就是处理事情的先后次序,像会议程序、工作程序等。以下就是用汉语编写的一个使用全自动洗衣机洗衣的程序:

- (1) 把脏衣服放到洗衣机中。
- (2) 插好电源,打开水龙头。
- (3) 设置洗衣的命令,如经济、普通、轻揉。
- (4) 放入洗涤剂。
- (5) 按下开始按钮。
- (6) 等待洗衣机清洗完毕。

当洗衣机清洗完毕后,蜂鸣器发出响声,即可将衣服取出晾晒,这就是人们洗衣服的程序。这个程序是由人的一项一项活动组成的,是有序地完成每一项活动,也就是实现了程序的目标。类似地,计算机程序是在计算机领域中,将人们要处理的问题,表达成计算机能执行的操作步骤;也即是把要让计算机解决的问题表达成计算机能识别和操作的代码序列。这个代码

序列包含计算机操作的对象(或数据)和计算机处理对象(或数据)的操作步骤。

在我国颁布的计算机软件保护条例中,对程序概念所给出的精确描述是:计算机程序是指为了得到某种结果而可以由计算机等具有信息处理能力的装置执行的代码化指令序列,或者可被自动地转换成代码化的符号化序列,或者符号化语句序列。可见,程序不仅给出了解决问题的方法和步骤,也表达了人们处理问题的系统思维。而且,程序受计算机系统条件的限制,必须适应计算机操作的要求。因此,正确完善的计算机程序应具有以下性质:

- 目的性:程序有明确的目的,程序执行完成指定的功能。
- 分步性:程序由计算机可执行的一系列基本步骤组成。
- 有序性:程序执行步骤是有序的,不可随意改变程序步骤执行顺序。
- 有限性:程序包含的指令序列是有限的。
- 操作性:有意义的程序总是对某些对象进行操作,完成预定的功能。

2. 程序设计

程序设计是程序形成的过程。具体地说,程序设计是将计算机要解决的问题,构造成计算机能执行的指令代码序列的过程。这个过程,一般分成 3 个阶段:问题分析、设计与实现(即具体语言的写作、表达)。其中,问题分析和设计是第一位的,而语言的写作、表达是第二位的。即先有问题的分析与设计,后有语言的写作和表达。

- 问题分析是定义用户的需求,明确要解决的问题,在给定的计算机环境下,使用什么样的技术和方法可以达到预定的目标,实现问题域到计算机域的映射。

- 设计是指对问题的系统设计。在计算机环境的约束和限制的条件下,问题应该用模式抽象为计算机解决问题的模式,即是对解决问题进行抽象。计算机解决问题的方式就是对数据进行处理。因此,对于程序设计来说,首先是对处理问题对象(或数据)进行抽象,抽象出能够反映本质特征的数据,并对其进行描述,即数据结构的设计;然后考虑对设计好的数据如何进行操作以及获得问题结果,即算法设计。

- 实现即具体语言的写作、表达。实现主要包含两个方面的内容:一是使用某种程序设计语言编辑程序,本质上是对问题的处理方案的描述,也即是对数据结构和算法的精确描述;二是将编辑的程序经过语言处理程序翻译成等价的目标程序,并将目标文件和其他有关文件(如库文件)链接在一起生成一个可执行的文件,最后让计算机执行该文件,给出执行的结果。

程序设计是一种创造性的劳动,程序设计的结果(即程序)本质上是人类思维模式在计算机领域中具体的映射。因此,每个程序的形成与分析、设计阶段有一定的联系,是分析、设计阶段成果的一种具体实现。这种实现局限于特殊环境和特殊语言,因此涉及环境和语言特性。对于具体的环境和语言,同一问题可以有不同的实现策略,反映不同的思维,可见程序设计本身也会涉及思维问题。相对于分析、设计阶段而言,程序设计思维的重心集中在最终系统的具体实现,如技术、语言、平台等,而与原始问题的联系相对较少。不同的思维重心反映了看待问题的不同抽象层次。尽管人们的思维重心不同,但本质上仍然是人们的思维反映,因而可以理解为人们思维模式在不同认识域中的具体映射。

上面所说的程序设计的三个阶段是相辅相成的。作为一个程序设计新手,只能从简单问题入手,而且简单问题的解决方法也较简单,所以算法设计不会有什么困难。但是,C 语言对于初学者来说是全新的,因此初学程序设计时用 C 语言编程常常是较为困难的阶段,但可以肯定的是随着对 C 语言语法的了解和编程实践的不断增加,编程会变得越来越简单。而与此

相反的倒是,随着遇到的问题越来越复杂,其算法设计会变得更加困难和更具有挑战性。程序设计既是一门科学技术,又是一门艺术,就像练习写作一样,必须不断地进行编程实践,并且大量阅读他人的程序,以便积累经验,提高程序设计的能力。

1.1.2 程序设计语言与程序设计方法

1. 程序设计语言

程序设计语言(或计算机语言)是人类与计算机交流的工具,程序是用程序设计语言描述和表达的指令序列。显然,对于人们要解决的问题,一种有效的程序设计语言能提高程序的开发效率和对问题的表达力,能填补通常人们非结构化的思维方式与计算机执行所要求的精确性之间的沟壑。程序设计语言正是沿着为更好地满足这方面的需要而向前发展的。

在计算机发展的初期,人们直接使用机器语言编写程序。机器语言是以二进制表示的机器基本指令的集合,是计算机系统唯一不需要翻译就可以直接识别和执行的程序设计语言。机器语言程序的特点是,计算机运行速度快、效率高,每条指令均为0和1的代码串。指令代码包括操作码和操作对象,例如 $A+B$ 用机器语言表示为1000110010100000。机器语言是对计算机硬件逻辑的抽象,直接对应机器的操作。用机器语言编写程序时,要求程序员熟练地记住所有机器指令的二进制代码,数据单元的地址和指令地址,因而编程困难、工作量大、易于出错、程序可读性差,程序出错时的修改更是如此。同时,由于计算机机型的不同,计算机指令系统也往往各不相同,所以在一台计算机上执行的程序,要想在另一台计算机上执行,必须另编程序。因此,目前已很少用机器语言直接编程。

与机器语言相比,汇编语言是对机器语言的进一步抽象表示。用汇编语言编写程序可以用简洁的英文字母、符号串,即助记符来替代一个特定的二进制串,如用指令“ADD”代表加法,用“MOV”代表数据的传递,用标号和符号代表地址、常量、变量,等等。这样,人们较容易理解程序在干什么,对程序的改错和维护变得更方便。例如, $A+B$ 用汇编语言表示则为“A ADD B”。然而,计算机是不能直接识别这些符号串的,需要一个专门的程序来负责将这些符号翻译成二进制数的机器语言,这种翻译程序被称为汇编程序。

汇编程序通常由三部分组成:指令、伪指令和宏指令。汇编语言是与机器语言一一对应的一种程序设计语言。虽然汇编语言不是用0、1代码编写,但其实质和机器语言是相同的,也是对硬件直接操作,只不过指令采用助记符标识,更容易识别和记忆。由于汇编语言的每一条指令只能对应实际操作过程中的一个很细微的动作,例如移动、自增等,因此汇编源程序一般比较冗长、复杂、容易出错,而且使用汇编语言编程需要更多的计算机专业知识。因此,人们只有在直接编写面向硬件的驱动程序时才采用汇编语言。

机器语言和汇编语言均与特定的计算机硬件有关,它们都属于低级语言。

高级语言出现于20世纪50年代中期,是一种能被计算机接受的,同时接近人类语言描述方式的程序设计语言。相对于低级语言系统,高级语言屏蔽了机器硬件的许多细节,用接近人类自然语言的模式构成计算机指令系统,实现对机器操作的更高级层次的抽象。

相对于汇编语言来说,高级语言不但将许多相关的机器指令组合成为单条指令,并且去掉了与具体操作有关但与完成工作无关的细节,如堆栈、寄存器的使用等。这不仅在很大程度上简化了程序中的指令,而且由于省略了很多细节,编程者不需要有太多的专业知识。因此,使用高级语言编写程序不仅编程相对简单,易于理解,不容易出错,而且高级语言程序是独立于

计算机的,其通用性和移植性均较好。

计算机是不能直接执行高级语言程序的。为此,必须通过某种方式将高级语言程序转换成计算机能够执行的机器语言程序。在程序设计中,一般将高级语言编写的程序称为高级语言源程序,将计算机可直接执行的程序称为机器语言程序或目标程序,而把具有转换功能的程序称为语言处理程序。语言处理程序有两种方式:解释方式和编译方式。

解释方式类似于英语、汉语之间的“同声翻译”。即一边将源程序代码由相应语言的解释器(也称解释程序)“翻译”成机器执行的目标代码,一边执行,因此效率比较低,而且不能生成可独立执行的可执行文件。但这种翻译方式比较灵活,可以及时发现错误,可动态地调整、修改应用程序。如 BASIC 语言就是采取解释方式执行的。

编译方式是指在源程序被执行之前,将源程序先由编译器(也称编译程序)“翻译”成机器可执行的机器语言程序(一般对源程序的编译处理要经过词法、语法和语意分析,代码生成和代码优化等阶段,即将所处理的源程序转换成用二进制代码表示的目标程序),然后通过链接程序的处理将程序中的函数调用、系统功能调用等嵌入到目标程序中,并构成一个可连续执行的二进制程序。因此,用二进制代码表示的目标程序可以脱离其语言环境独立执行,使用比较方便,效率较高。但程序一旦需要修改,则必须先修改其源代码,再重新经编译处理后用二进制代码表示的才能执行。如 PASCAL、FORTRAN、C 等大多数高级语言都采用编译方式执行。

2. 程序设计方法

程序是程序设计语言描述、表达的操作序列。同一问题组成的不同操作序列,不仅反映了程序设计的方法不同,同时也说明了程序设计方法与程序设计语言密切相关,即程序设计方法与具体的程序语言相捆绑。所谓程序设计方法就是如何组织程序的问题,即用什么样的逻辑模式组成所要解决的对象和相应的操作。随着计算机技术的发展,为了简化程序设计和提高软件生产效率,程序设计方法的改进伴随着程序设计语言的进化,程序设计语言的发展推动着程序设计方法的变革。目前,程序设计方法主要是结构化程序设计方法和面向对象的程序设计方法。

1) 结构化程序设计方法

结构化程序设计方法是对所描述数据(或对象)的操作序列采用顺序、选择和循环三种基本控制结构的方法。在结构化程序设计方法中,程序的过程体和组成部分以模块表示,过程体主要用函数来实现,并采用从上到下逐步细分的方法,即在一定的数据结构之下,设计对应的算法,实现模块化和结构化。用结构化方法设计的程序称为结构化的程序。

2) 面向对象程序设计方法

面向对象程序设计方法就是将要解决的问题分解为对象进行设计的方法。面向对象的程序是由对象组成的,即程序=对象+对象+对象+……。对象是客观世界可独立存在的、被区分的实体(包含概念上的实体,如一项计划等),并且任何系统都是由众多对象构成的。对象有自己的数据(属性),也包括作用于数据的操作(方法),对象状态的变化和对象之间的相互作用通过消息传送来实现,面向对象程序设计的模式为“对象+消息”。

面向对象程序设计的方法主要使用抽象机制将对象的共性抽象成类,由类实现对象及对象操作的封装。由于对象之间的包含、并列和继承关系,面向对象程序设计不仅要关心每个类的细节,而且还要关心类与类之间的关系。因此,面向对象程序设计的关键在于分类、分层和

抽象。由于面向对象的程序是由类构成且是用类实现模块的,所以面向对象程序设计的主要特点就是封装、继承和多态。

1.2 C语言的产生、发展及标准化

计算机应用和计算机技术的迅速发展,推动着程序设计语言的发展。在众多的程序设计语言的应用中,C语言能够脱颖而出、独树一帜地成为全球范围内流行和使用最为广泛的一种程序设计语言,是与其产生、发展与应用需求背景有着密切联系的。

1. C语言的产生

在20世纪60年代,FORTRAN、ALGOL、COBOL、LISP这样一些面向过程的语言较好地解决了科学计算、商用程序、表处理方面的应用问题。但是,由于这些语言都是面向特定的应用,缺乏对机器硬件的直接操作与控制能力是这些语言的共同缺陷。另一方面,汇编语言虽然具有良好的硬件操纵能力,但其面向机器的基本特性,对硬件指令系统的依赖性,仅依赖助记符来描述机器操作使之缺乏面向问题的数据抽象能力和算法描述能力。这样一些不足使得用汇编语言编程的难度大、效率(劳动生产率)低。因此,能否设计出一种既具有面向过程的特性,又具有汇编语言特点的新的程序设计语言是当时计算机界迫切需要解决的问题。

1969~1973年,美国贝尔实验室的Dennis M Ritchie为了在DEC公司的PDP-11机器上实现UNIX操作系统,在B语言的基础上首先设计并实现了C语言。

2. C语言的发展

1973年,Dennis M Ritchie和Ken Thompson合作,又用C语言重写了UNIX操作系统的内核,先后推出了UNIXV5和UNIXV6,但此时的C语言还是附属于UNIX操作系统的。1973~1977年,用C语言编写的UNIX操作系统先后被移植到Honeywell 635、IBM 360/370、Interdata 8/32、DECVAX 11/780以及AT&T的机器上。1978年,Brian Kernigha和DMRitchie以UNIXV7的C编译程序为基础,写出了被称为“C语言的圣经”的影响深远的名著“The C Programming Language”。它介绍的C语言被称为K&RC,或称传统C,并且对C语言的普及和推广起到了不可估量的作用。自1978年以后,传统C已经被移植到各种类型的计算机上,并独立于UNIX和其他操作系统。

3. C语言的标准化

1978年,由于UNIX V7和UNIX System V的相继推出,使UNIX系统得到了广泛的使用。伴随UNIX取得的巨大成功,C语言的突出优点也被人们广泛认同,从而极大地推动了C语言的普及和推广,同时也导致了各种各样C语言版本的出现。由于传统C在许多语言细节方面尚显得不够精确,于是美国ANSI在1983年成立了C语言标准化的委员会X3J11,并由它负责C语言的标准化。X3J11于1989年底公布了美国第一个C语言的国家标准ANSI 89,俗称C89。

1990年,国际标准化组织ISO将ANSI 89接受为C语言的国际标准,称为ISO/IEC 9899:1990。它是C语言的第一个国际标准,也称为标准C,即ANSI C,俗称C90。

在20世纪90年代,随着对宽字符集支持需求的出现,对整型数据和浮点数据表示范围扩

大的需求,以及数值计算精度提高的要求,C 语言的标准需要不断扩充和完善。ISO/IEC 在 1995 年公布了一个新的 C 语言标准的草案,俗称为 C95,供讨论和征求意见。接着 ISO/IEC 在 1998 年又公布了新标准的草案 WG14/N843 和 WG14/N897,进一步就 C 语言标准的完善征求意见。

在长期征求意见和修改的基础上,ISO/IEC 于 1999 年 12 月公布了最新的 C 语言国际标准 ISO/IEC 9899:1999(E),它是 C 语言国际标准 9899 的第二版,俗称 C99。该新标准在字符集方面支持三重图形字符序列(Trigraph sequences)和多字节字符集(Multibyte characters),在整型数据方面支持 long long int 类型,在浮点数运算方面支持 long double 类型,并且新引入了三种复数类型和布尔类型等。

从以上的介绍可知,C 语言的产生与演变体现出继承与创新的特点。例如,C 语言渊源于 B 语言,B 语言又渊源于 BCPL 语言,BCPL 语言又渊源于 CPL 语言,CPL 语言又渊源于 ALGOL60 语言,而 ALGOL60 语言又渊源于 FORTRAN 语言。C 语言的这种继承、创新、发展与完善的关系可以用图 1.1 来描述。

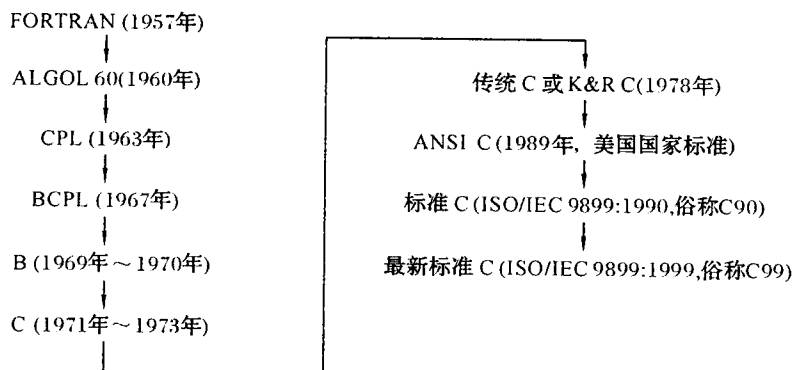


图 1.1 C 语言的继承、产生与发展历程

1.3 C 语言的体系结构及特征

人类语言是一个具有层次结构的逻辑体系,如图 1.2 所示。程序语言与人类语言一样,具有本身的体系结构以及描述数据和处理数据的基本功能。C 语言作为一种结构化程序设计语言,除有着高级程序设计语言的共性外,它的成功,还得益于其在语法、语义和语言定义过程中的系统适应能力和灵活性,也就是 C 语言与其他高级语言相比具有更多的特殊性。

1.3.1 C 语言的体系结构

C 语言的基本体系结构如图 1.3 所示。从上往下看,顶层是程序,它描述的是应用,是一个完整的执行单位。一个程序通常是由若干个函数组成的,函数有自己的数据,具有独立功能。函数是由语句组成的,而组成语句的成分则是各种类型的表达式。表达式既是描述数据运算的基本结构,又是由常量、变量、运算符和关键字等组成,其中最基本的是字符集,它由有限的基本符号组成。从下往上看,每一个层次中各成分的组成和所具有的意义以及向上一层次的扩展,都是由 C 语言的语法和语义两方面定义的。语法一般指词法规则和语法规则,其中词法规则规定了 C 语言基本符号集上的字符串是一个单词符号,语法规则规定了如何由单

词符号组成更大的结构(语法范畴),如表达式、语句、函数和程序,等等。例如,字符串 $0.5 * X1 + A$,通常被看成是由常数 0.5、标识符 $X1$ 和 A 以及算符 $*$ 和 $+$ 所组成的一个表达式,其中常数 0.5、标识符 $X1$ 和 A 、算符“ $*$ ”和“ $+$ ”称为单词符号,而表达式 $0.5 * X1 + A$ 称为语言的范畴。

语义规则是指对语言体系中的每种组成成分所具有功能的规定。语言的每种成分都有逻辑含义和计算机实现两方面的功能(或意义)。其中,当从问题的表达和处理考虑时,需要了解它的逻辑意义;当从计算机角度来看时,需要知道在机内的表示和实现。例如,赋值语句“ $X = 0.5 * X1 + A;$ ”,它表达的语义是首先计算语句右边表达式的值,然后把所得结果赋值给左边变量 X 。从逻辑上看,这是计算和保存表达式的值;而从计算机实现上看,则是对存储单元中的数据进行存储和访问,或称为读和写。

C 语言体系结构每部分的详细组成将在后面的章节中介绍,本节只是从语法角度介绍其体系架构,以便于读者对 C 语言的初步理解。

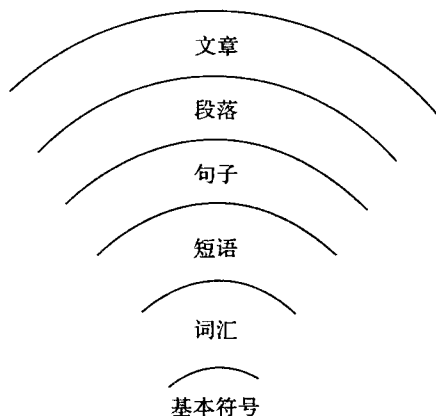


图 1.2 人类语言的基本体系结构

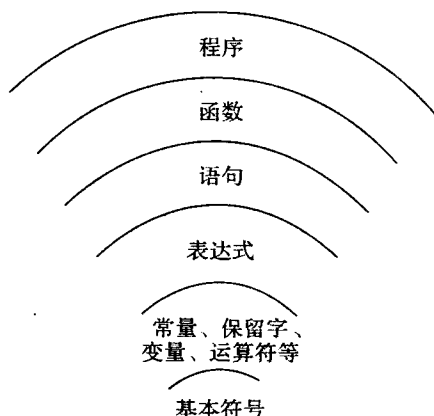


图 1.3 C 语言的基本体系结构

1.3.2 C 语言的语言特征

C 语言既具有高级语言的特点,也具有汇编语言的某些特点,因此可用于应用软件和系统软件的编程。尤其是它的表达力强、效率高、简练、紧凑、对机器的最终控制以及跨平台的高度移植性等特点,使其在系统编程、嵌入式编程等领域一直占据着统治地位。而 C99 标准的制订则再一次使 C 语言焕发出新的活力。综合起来,C 语言主要有以下一些与众不同的特点:

1) 语言简洁紧凑

C 语言词法记号中关键字集小,只有 37 个关键字(keywords)。用以构造程序的语句集简洁,将声明语句包括在内全部只有 12 种语句。即:声明语句,表达式语句,复合句,if—else 语句,switch 语句,while 语句,for 语句,do—while 语句,break 语句,continue 语句,以及 goto 语句和标号语句。除声明语句外,其余 11 种语句称为可执行语句。可执行语句中前面 7 种为常用语句,这 7 种常用语句往往被称为基本语句。简洁的语句集和小关键字集可以最大限度地减少编程需要记忆的成分,有利于人们对语言的学习与使用。

2) 语言表达能力强

一种语言的表达能力是由该语言提供的类型集、操作集以及语句集对应用描述的支持程度来度量的。