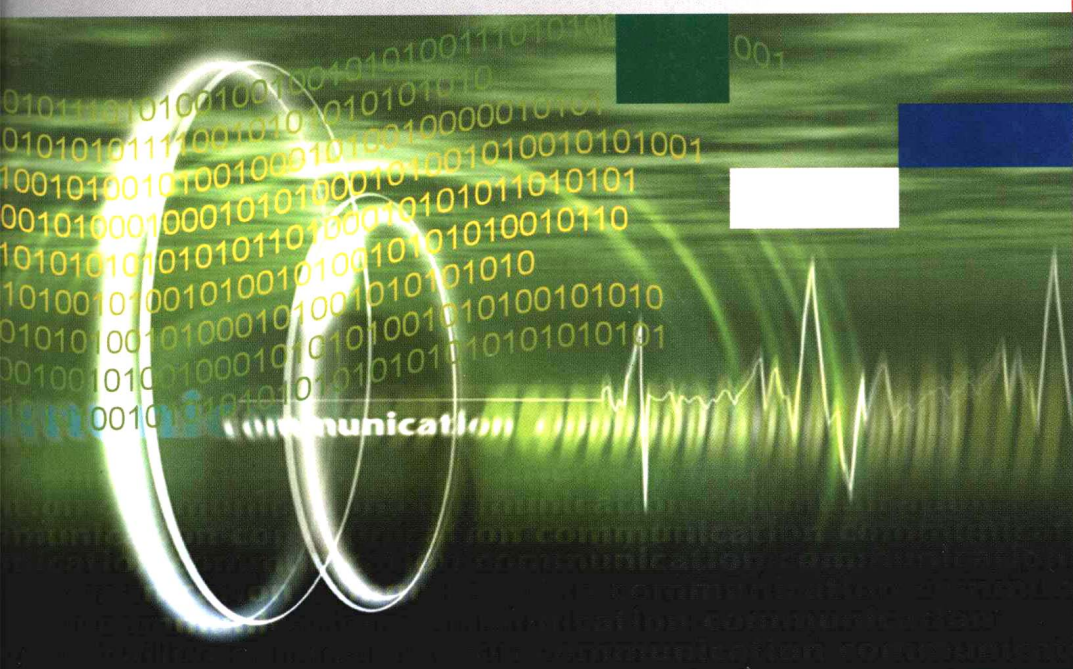




普通高等教育“十一五”规划教材



C语言程序设计

王四万 张郭军 王文东 编著



科学出版社

www.sciencep.com

普通高等教育“十一五”规划教材

C 语言程序设计

王四万 张郭军 王文东 编著

科学出版社

北 京

内 容 简 介

本书主要介绍 C 语言基本概念、基本语法和基本数据结构,还介绍了 C 程序设计中的常见错误和 C 程序的一般上机调试方法,并简单介绍了 C 语言图形程序设计基本技术方法。全书始终贯穿了程序设计基本方法和技术为主线的学习思想;由浅入深、循序渐进的叙述方式,抽象问题形象化、具体化、繁琐问题表格化的编写特点,为读者提供了简洁、逻辑结构清晰的学习内容;各章小结浓缩本章知识要点,为读者提供了掌握学习重点的途径;习题选择重视了实践环节的训练,重在培养读者编程能力。

本书适合作为高等院校计算机程序设计入门基础教材,也可作为理工科各专业技术人员学习 C 语言程序设计的参考资料,还可供参加全国等级考试的读者作为考试复习的参考书。

图书在版编目(CIP)数据

C 语言程序设计/王四万,张郭军,王文东编著. —北京:科学出版社, 2008

(普通高等教育“十一五”规划教材)

ISBN 978-7-03-023255-7

I. C… II. ①王…②张…③王… 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(2008)第 166238 号

责任编辑:陈晓萍/责任校对:耿 耘

责任印制:吕春珉/封面设计:耕者设计工作室

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

骏主印刷厂印刷

科学出版社发行 各地新华书店经销

*

2009 年 1 月第 一 版 开本:787×1092 1/16

2009 年 1 月第一次印刷 印张:20 1/2

印数:1—4 000 字数:480 000

定价:29.00 元

(如有印装质量问题,我社负责调换〈环伟〉)

销售部电话 010-62134988 编辑部电话 010-62135120-8003

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303

前 言

C 语言自问世以来,由于其功能丰富、表达灵活方便、目标代码效率高、实用性强,特别是可移植性好,既有高级语言的优点,又有低级语言的许多(类似汇编语言的)特性。因而,C 语言已成为当今最为流行的程序设计语言之一,也已成为专家公认的,理工科各专业学生训练程序设计基础、培养计算机应用能力的首选语言。

目前,C 语言已不仅是计算机科学与技术专业必不可少的主要课程,而且也成为广大非计算机专业人员参加全国计算机等级考试、全国计算机应用技术证书考试(NIT)的考试科目之一。显而易见,掌握 C 语言程序设计已成为计算机基础素质培养的重要组成部分。

本书是作者在总结多年来本课程教学经验教训的基础上,听取各方面的意见和建议,进行充分研讨与论证,根据理工科各专业计算机程序设计基础教育要求编写的。它不仅可作为大专院校理工科各专业程序设计的教学用书,还可以作为工程技术人员的自学参考书。

本书由王四万负责全书编写的总体设计、组织、审校和统稿工作。第 1~4、10、11 章和附录由王四万编写,第 5~7 章由张郭军编写,第 8、9、12 章由王文东编写。

本书主要有以下特点:

1. 本书按初学者以掌握程序设计的入门语言安排学习内容,由浅入深、循序渐进,重点和难点分散。只要求读者具有计算机文化基础和一定的数学知识,即可学习并掌握本书内容。

2. 本书内容安排注重了教材的简洁性和条理性,力求做到将繁琐内容通俗化、简单化。如将表达“算法”的各种图形概念分散于相关章节;将各种数据类型、表达式等语法规定“表格化”,以尽可能减少读者对概念内容上的记忆负担。

3. 将“算法设计”基本知识概念渗透到程序设计举例之中,以使读者在具体问题中掌握算法设计的要领与方法,这样一方面可解决抽象概念的难教难学问题,另一方面可使学生更深刻领会“程序=算法+数据结构”的思想。更为重要的是,能够使读者在学习中充分体会到,掌握 C 语言关键在于运用它进行程序设计。

4. 每章最后一节安排为“小结”,以对本章内容进行提要与总结。目的在于帮助读者能够提纲挈领,抓住重点,掌握本章学习内容。

5. 本书取材与选例,力求典型和少而精,目的在于减少读者阅读量的负担。编者认为,程序设计语言学习不单是掌握语言本身,训练学习者的程序设计能力更为重要。因而本教材力求将 C 语言最基本的知识教给读者,突出重点培养读者程序设计基本思想技术方法,而将不多用的内容,如“位运算”写入附录,供读者选学。

6. 图形处理是计算机应用的基本内容。本书专设一章介绍计算机图形处理 C 程序

设计，教给读者计算机处理图形基本技术思想方法，以培养读者图形处理的编程初步能力。

为配合读者学习本书，我们另编写了本书的习题参考答案。欢迎读者与科学出版社联系。

本书编写过程中，我们参考了国内外有关书籍与教材，吸收了各书的编写经验，引用了其中一些材料，在此谨向各书的作者和出版社表示深切感谢。西安交通大学博士生导师冯博琴教授、西安石油大学陈建铎教授、西北大学博士生导师耿国华教授给予了很多有益的指导；陕西师范大学计算机科学学院博士生导师李永明教授、曹函教授和汪西莉教授、李葆华、郭敏等老师给予了很多帮助。冯博琴教授在百忙中审阅了全书并提出宝贵意见，使本书质量得到明显提高。在此一并表示感谢。

对于参与本书讨论和提出宝贵意见的同志致以谢意。

由于我们水平有限，编写经验不足，加上时间仓促，书中缺点、错误在所难免，欢迎读者批评指正。

王四万

2008 年 10 月于西安

目 录

前言

第 1 章 概述	1
1.1 程序与程序设计语言	1
1.1.1 程序	1
1.1.2 程序设计语言	1
1.2 C 语言的历史与特点	3
1.2.1 C 语言的发展历史	4
1.2.2 C 语言的特点	4
1.3 C 程序简介	5
1.4 C 程序的运行环境及其编译与执行	7
习题	11
第 2 章 数据类型、运算符与表达式	12
2.1 常量与变量	12
2.1.1 常量与符号常量	12
2.1.2 变量	13
2.2 整型数据	15
2.2.1 整型常量	15
2.2.2 整型变量	16
2.3 实型数据	16
2.3.1 实型常量	16
2.3.2 实型变量	17
2.4 字符型数据	18
2.4.1 字符常量	18
2.4.2 字符变量	19
2.4.3 字符串常量	20
2.5 各种数值型数据间的混合运算	21
2.5.1 数据类型转换规则	21
2.5.2 数据类型转换方法	22
2.6 算术运算符和算术表达式	23
2.6.1 基本的算术运算符	23
2.6.2 算术表达式	24
2.7 赋值运算和赋值表达式	24

2.7.1 简单赋值运算	24
2.7.2 复合赋值运算	26
2.7.3 自增运算符和自减运算符	26
2.8 逗号运算符和逗号表达式	27
2.9 小结	28
习题	30
第3章 顺序程序设计	33
3.1 概述	33
3.1.1 C 语句分类	33
3.1.2 顺序程序设计概念	34
3.2 赋值语句、变量赋初值	35
3.2.1 赋值语句	35
3.2.2 变量赋初值	35
3.3 C 程序处理数据的输入与输出	36
3.3.1 数据输入与输出概念	36
3.3.2 字符数据的输入与输出	36
3.4 格式输入与输出	37
3.4.1 格式输出函数——printf 函数	37
3.4.2 格式输入函数——scanf 函数	41
3.5 程序举例	42
3.6 小结	44
习题	46
第4章 选择结构程序设计	49
4.1 关系运算符和关系表达式	49
4.1.1 关系运算符	49
4.1.2 关系表达式	50
4.2 逻辑运算符和逻辑表达式	51
4.2.1 逻辑运算符	51
4.2.2 逻辑表达式	52
4.3 条件运算符和条件表达式	53
4.4 选择语句	55
4.4.1 流程图和 N-S 图	55
4.4.2 if 语句的语法格式	55
4.4.3 if 语句的嵌套	57
4.5 switch 和 break 语句	59
4.5.1 switch 语句	59
4.5.2 break 语句	61

4.6	程序举例	61
4.7	小结	64
	习题	66
第 5 章	循环结构程序设计	68
5.1	概述	68
5.2	goto 语句与循环	68
5.3	while 语句	69
5.4	do-while 语句	70
5.5	for 语句	71
5.6	几种循环结构的比较	73
5.7	循环的嵌套	73
5.8	循环控制语句	74
5.8.1	用 break 语句终止循环	74
5.8.2	continue 语句	75
5.9	算法及其表示方法简介	77
5.9.1	算法及其特性	77
5.9.2	算法表示方法	78
5.9.3	用程序设计语言实现算法	81
5.10	循环算法的设计与实现	82
5.10.1	穷举算法	83
5.10.2	迭代算法	85
5.11	小结	87
	习题	88
第 6 章	数组	92
6.1	概述	92
6.2	一维数组	94
6.2.1	一维数组的定义	94
6.2.2	一维数组元素的引用	95
6.2.3	一维数组的初始化	96
6.2.4	一维数组程序举例	97
6.3	二维数组	105
6.3.1	二维数组的定义	105
6.3.2	二维数组的引用	106
6.3.3	二维数组的初始化	106
6.3.4	二维数组程序举例	107
6.4	字符数组	110
6.4.1	字符数组的定义	110

6.4.2	字符数组的初始化	111
6.4.3	字符数组的输入和输出	112
6.4.4	字符串处理函数	115
6.5	字符数组应用	118
6.6	小结	121
	习题	121
第 7 章	函数	126
7.1	概述	126
7.2	函数的定义	127
7.2.1	函数定义的一般形式	127
7.2.2	函数的参数与函数的返回值	129
7.3	函数的调用	132
7.3.1	函数调用的方式	132
7.3.2	对被调用函数的声明	132
7.4	函数的嵌套调用和递归调用	134
7.4.1	函数的嵌套调用	134
7.4.2	函数的递归调用	137
7.5	局部变量和全局变量	141
7.5.1	局部变量	141
7.5.2	全局变量	142
7.6	变量的存储类别	143
7.6.1	自动变量	144
7.6.2	局部静态变量	145
7.6.3	register (寄存器) 变量	146
7.6.4	用 extern 关键字声明的外部变量	146
7.6.5	用 static 关键字声明的外部变量	147
7.7	内部函数和外部函数	147
7.7.1	内部函数	147
7.7.2	外部函数	148
7.8	结构化程序设计方法简介	148
7.8.1	自顶向下程序设计	148
7.8.2	运行一个多源文件组成的 C 程序	149
7.9	小结	150
	习题	151
第 8 章	指针	160
8.1	概述	160
8.2	指针的概念	160

8.3 指针变量	162
8.3.1 指针变量的定义	162
8.3.2 指针变量的应用	163
8.3.3 指针变量作函数参数	166
8.4 指针与数组	168
8.4.1 指向一维数组元素的指针	168
8.4.2 一维数组作函数参数	171
8.4.3 指向数组元素的指针变量作函数参数	172
8.4.4 指向多维数组元素的指针	173
8.4.5 数组指针	175
8.4.6 指向多维数组元素的指针作函数参数	178
8.4.7 动态内存分配	179
8.5 指针与字符串	182
8.5.1 指向字符串的指针	182
8.5.2 字符串指针作函数参数	185
8.6 指针数组与多级指针	187
8.6.1 指针数组	187
8.6.2 指针数组的应用	188
8.6.3 指向指针的指针变量	190
8.6.4 main 函数的参数	192
8.7 函数指针与指针函数	194
8.7.1 函数指针	194
8.7.2 指针函数	196
8.8 小结	198
习题	198
第9章 结构体与共用体	204
9.1 概述	204
9.2 结构体	204
9.2.1 结构体类型的定义	204
9.2.2 结构体类型变量的定义及初始化	206
9.2.3 结构体类型变量及其成员的引用	208
9.2.4 结构体数组	209
9.2.5 指向结构体类型数据的指针	212
9.2.6 用结构体变量及其成员作函数参数	215
9.3 共用体	217
9.3.1 共用体类型及变量的定义	217
9.3.2 共用体变量的引用方式	218

9.4	枚举类型	222
9.5	用 typedef 定义类型	223
9.6	编译预处理命令	224
9.6.1	宏定义	224
9.6.2	文件包含	227
9.6.3	条件编译	227
9.7	小结	229
	习题	231
第 10 章	文件	236
10.1	C 文件概念	236
10.2	文件类型与文件指针	237
10.3	文件的打开与关闭	238
10.3.1	文件打开与 fopen 函数	238
10.3.2	文件关闭与 fclose 函数	240
10.4	文件的读写	241
10.4.1	单字符(字节)方式读写	241
10.4.2	字节块数据项读写与 fread 和 fwrite 函数	244
10.4.3	文件的格式读写	248
10.4.4	文件处理的其他读写函数	250
10.5	文件读写指针定位	251
10.5.1	rewind 函数	251
10.5.2	随机读写和 fseek 函数	252
10.5.3	文件位置指针的当前位置与 ftell 函数	253
10.6	文件检测	253
10.6.1	文件操作出错检测	253
10.6.2	文件是否结束检测	254
10.7	小结	254
	习题	256
第 11 章	常见错误和程序调试	260
11.1	常见错误分析	260
11.2	程序调试	271
11.3	检查和分析错误原因	272
11.4	Turbo C 环境下编辑、调试和运行 C 程序的方法	273
第 12 章	图形程序设计	279
12.1	概述	279
12.2	图形模式的初始化	279
12.3	独立图形运行程序的建立	281

12.4 屏幕颜色的设置和清屏函数.....	281
12.5 基本图形绘制函数.....	283
12.5.1 点的函数.....	283
12.5.2 画线函数.....	284
12.6 封闭图形的填充.....	286
12.6.1 先画轮廓再填充.....	286
12.6.2 设定填充方式.....	287
12.6.3 任意封闭图形的填充.....	289
12.7 图形模式下的文本输出.....	289
12.7.1 文本输出函数.....	289
12.7.2 有关文本字体、字型和输出方式的设置.....	290
12.7.3 用户对文本字符大小的设置.....	292
12.8 有关图形窗口和图形屏幕操作函数.....	293
12.8.1 图形窗口操作.....	293
12.8.2 屏幕操作.....	293
12.9 小结.....	294
习题.....	294
附录 A 常用字符与 ASCII 代码对照表.....	295
附录 B C 语言位运算.....	297
附录 C 运算符和结合性.....	301
附录 D Turbo C 常用库函数.....	303
附录 E 常见的编译出错信息.....	310
主要参考文献.....	314

第 1 章 概 述

伴随人类进入信息化社会, 计算机技术日新月异迅猛发展, 广泛应用于社会的各个方面, 越来越显示出了计算机对于人类社会经济、文化以及国防的巨大作用和深刻意义。人们愈来愈明显地发现, 一切高新的科学技术无一没有依赖计算机技术!

然而计算机本身并不具备解决实际问题的能力, 人们应用计算机“解算”实际问题时, 必须要有与计算机交流信息的“语言”, 还要设计给出计算机解算问题的“程序”; 本章介绍计算机语言及其发展, 以及程序设计语言等概念, 并通过对 C 语言发展简要介绍, 以使读者理解计算机语言、程序等概念, 并对 C 程序设计语言有一个基本了解。

1.1 程序与程序设计语言

1.1.1 程序

通常, 人们把数字电子计算机 (Digital Electronic Computer) 简称为计算机 (Computer), 典型的数字电子计算机由中央处理器 (CPU)、计算机存储系统和计算机输入/输出系统组成, 并统称它们为计算机“硬件”。计算机仅有硬件还不能工作, 还必须为计算机配备一定的“软件”, 这样计算机才可以为人们服务, 完成人们既定的工作任务。因此通常有计算机系统之称:

计算机系统 = 计算机硬件 + 计算机软件

所谓计算机软件, 是指为计算机所配备的一套解算特定问题的“程序”及其相关 (维护、使用) 资料的集合; 所谓计算机程序 (简称程序), 是指计算机解算特定问题时赖以确定信息处理的规则和次序, 换句话说, 程序就是解算特定问题所需计算机指令构成的规则序列。如果把解算每一个问题的“解决方案及其实施方法”看成为一篇文章, 那么这篇文章的计算机语言表达就是程序。

同样的硬件配以不同的程序, 就可以解决不同类型的问题。由此可见, 计算机的程序和相应的数据及文档合称为计算机软件, 它包括计算机系统软件、计算机应用软件和计算机支持软件。计算机硬件和软件组成的有机整体称为计算机系统。

1.1.2 程序设计语言

在计算机问世以来的几十年中, 随着电子技术和计算机技术的发展, 计算机硬件不断升级换代并使计算机应用领域迅速扩大, 计算机程序设计语言也随之有很大的发展。在程序设计语言的发展过程中, 先后产生了机器语言、汇编语言、高级语言、直到目前流行的面向对象语言和可视化编程语言, 经历了从低级向高级的发展。

1. 机器语言

计算机程序设计语言是人与计算机进行信息交流的一种工具，它用于描述控制计算机工作的各种指令和数据。所谓机器语言，就是把这些指令（如加法指令、减法指令、数据传输指令等）和参加运算的数据以及数据存放地址等，按照一定规则形式组织为 0、1 代码序列。由于机器语言指令的表示形式与它在计算机中存放的形式一致，计算机可以直接识别，因而用机器语言编写的程序，计算机不仅可以直接执行，而且运行速度很快。它的缺点是，不同机型的指令代码不同，因而用机器语言编写的程序通用性差，另外，不难想象使用机器语言编写程序很繁琐，它不仅难学、难记、难理解、难维护，而且容易出错。

2. 汇编语言

由于用机器语言编写程序非常困难，随之而来的就出现了符号汇编语言。汇编语言把用二进制数表示的机器指令，用一些“助记符”表示，如用英文缩写 ADD 表示加法运算、SUB 表示减法运算，用一些其他形式的数字和符号表示数值、存储单元的地址等。汇编语言指令与机器指令基本上是一一对应的，但它便于记忆和使用，因而编写程序比较容易。但是由于汇编指令计算机不能直接识别与执行，所以用汇编语言编写的程序，执行前需要先翻译成机器指令，然后才能执行。这个翻译过程称为“汇编”，专门用于进行这种翻译工作的程序称为“汇编程序”。

汇编语言虽较机器语言前进了一大步，但仍保留了机器语言的一些特点。例如，它仍是面向机器的语言（因此汇编语言和机器语言被称为低级语言），用它编写程序不仅要了解计算机内部结构有一定的了解，而且不同类型机器使用的汇编语言也不相同。尽管如此，汇编语言仍具有程序短、运行速度快的特点，因此在某些特殊应用（如实时控制）中仍然使用。

3. 高级语言

从 20 世纪 50 年代中期以来陆续产生了许多“高级算法语言”，这些高级语言不依赖于具体的机器，可以用接近于数学语言和自然语言的方式描述解决问题的方法和步骤。高级语言由于独立于机器，编程者在不了解机器内部构造和特点的情况下，也可以编写出解决实际问题的程序；高级语言由于不受具体机器的限制，用高级语言编写的程序，可以在不同类型的计算机上运行，从而提高程序的通用性；高级语言由于易学易用，因此迅速得以推广和使用。显然，使用高级语言可以提高编程效率，极大地推动计算机的普及与应用。

尽管高级语言具有诸多优点，但是用高级语言编写的程序，计算机仍然不能直接识别和执行，需要将它翻译成机器指令后才能在计算机上执行。一般将用高级语言编写的程序称为“源程序”，把翻译后生成的机器语言程序称为“目标程序”。这种翻译通常有两种做法，即“编译”方式和“解释”方式。所谓编译方式是先用编译程序把高级语言源程序翻译成目标程序，然后再执行生成的目标程序；而解释方式是将源程序由解释程

序逐句翻译,解释一句执行一句,边解释边执行。

迄今为止虽然已有上千种高级语言,但新的语言仍在不断出现。尤其是面向对象语言和可视化编程语言,以其显著的优点和强大的功能,已经成为当前主要的软件开发工具。

面向对象语言是比面向过程语言更高级的程序设计语言,面向对象语言的出现改变了编程者的思维方式,使程序设计的出发点由着眼于解决问题的过程,转向着眼于问题中的对象及其相互关系,面向对象语言更接近于自然语言,是人们对于客观事物更高层次的抽象。

可视化编程语言具有强大的图形用户界面(GUI)开发功能,它不需要编写大量的程序代码,只需使用一些预先建立的控件,就可以很容易地设计出各种美观、实用的图形用户界面,是 Windows 环境下理想的软件开发工具。

目前在各领域中经常使用的高级语言,主要有以下几种:

1) Fortran 语言:它是世界上第一个出现的高级语言,1954 年提出并于 1956 年实现。该语言特别适用于数值计算,是科学计算的主流语言,目前还有使用。

2) Basic 语言:是一种适于初学者使用的语言,简单易学。现今 Basic 已有许多高级版本,尤其 Windows 环境下的 Visual Basic 是一个功能强大的可视化软件开发工具。

3) Pascal 语言:它是结构化程序设计语言,适用于教学、数值计算、数据处理和系统软件开发等。目前十分流行的 Delphi 就是由它发展而来的。

4) C 语言:C 程序设计语言以简练、功能强大而著称全球,它具有丰富的数据类型和灵活的运算表达式,不仅具备较好的数值计算功能,而且还具备较好的符号处理功能,从而被公认为是培养程序设计能力的最好高级语言。适用于系统软件开发、数值计算、数据处理等,目前已成为高级语言中使用最多的语言之一。C++是在 C 语言的基础上发展起来的,是对 C 语言的扩充。它是一个功能强大的面向对象程序设计语言,是目前主要的软件开发工具之一,其中具有可视化编程环境、功能强大的 Visual C++目前使用十分流行。

5) Java 语言:它是一种跨平台分布式程序设计语言,是一种基于 C++的新语言,具有简单易学、安全稳定、面向对象、与平台无关、支持多线程等特点。它也是目前流行的主要软件开发工具之一。

1.2 C 语言的历史与特点

C 语言是国际上广泛流行的计算机高级语言。它具有丰富的数据类型和灵活的运算表达式,不仅具备较好的数值计算功能,而且还具备较好的符号处理功能,因而被很多大专院校理工科各个专业作为培养学生程序设计能力的首选高级语言。由于它既具有高级语言的特性,又具有某些低级语言的特性,因而被称为中级语言。它不仅可用于编写系统软件,也可编写应用软件,以 C 语言作为理工科各专业学生程序设计能力培养的工具,对于读者今后在各自领域开展应用软件开发练好基本功,将具有十分重要的现实意义

义。所以 C 语言仍然是目前最流行、最受欢迎的计算机语言之一。

1.2.1 C 语言的发展历史

C 语言最初是作为设计操作系统而研制的高级程序设计语言,它是以 1967 年英国剑桥大学 Martin Richards 推出的 BCPL 语言等为基础发展而来的。1970 年美国贝尔实验室的肯·汤普森 (Ken Thompson) 在开发 UNIX 操作系统的过程中,以 BCPL 语言为基础设计出了 B 语言。但由于 B 语言过于简单,功能有限,于是 1972 年贝尔实验室的丹尼斯·里奇 (Dennis Ritchie) 又在 B 语言的基础上编写出了 C 语言。C 语言不仅保持了 BCPL 和 B 语言的优点,同时还克服了它们的缺点。1973 年肯·汤普森和丹尼斯·里奇合作,重新用 C 语言编写了 UNIX 操作系统。其后,随着 UNIX 操作系统开发获得成功,并且被日益广泛地应用,C 语言也被越来越多的人接受。于是,C 语言被逐渐发展成为一种通用的程序设计语言,风靡全世界。

C 语言在产生的初期并没有确定统一的标准,直到 Brian W.Kernighan 和 Dennis M.Ritchie 以 UNIX 第七版中的 C 编译程序为基础,合著了影响深远的名著《The C Programming Language》。该书中介绍的 C 语言成为后来各种 C 语言版本的基础,即现在所谓的“经典 C”或 K&R C。1983 年,美国国家标准化协会 (ANSI) 根据 C 语言问世以来各种版本对 C 的发展和扩充,制定了新的标准,称为 ANSI C。其后,于 1987 年 ANSI 又公布了 ANSI C 的新标准,1990 年 ANSI 又与国际标准化组织 (ISO) 合作在全球范围内将 C 语言标准化。目前流行的各种 C 编译系统虽有一些不同,但都以这个标准为基础,其基本部分是相同的。本书的叙述也基本上以 ANSI C 为基础。

1.2.2 C 语言的特点

C 语言能够成为目前应用最广泛的高级语言之一,是由其语言特点决定的。C 语言的特点可大致归纳如下。

1) 语言简洁、紧凑,使用方便、灵活。C 语言一共只有 32 个关键字,9 种控制语句,程序书写形式自由。

2) 运算符丰富,表达能力强。C 运算符包含的范围很广泛,共有 34 种运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理,从而使 C 的运算类型极其丰富,表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

3) 数据结构丰富,具有现代语言的各种数据结构。C 的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。C 语言丰富的数据结构能用来描述实际应用中的各种复杂数据结构及其运算,尤其是指针类型数据使用起来更为灵活、多样。

4) 具有结构化的控制语句 (如 if ... else 语句、while 语句、do ... while 语句、switch 语句、for 语句),用函数作为程序模块以实现程序的模块化。因此是理想的结构化语言,符合现代编程风格的基本要求。

5) 语法限制不太严格,程序设计自由度大。例如,对数组下标越界不做检查,由编

程者自己保证程序的正确性。对变量类型使用比较灵活，如整型数据与字符型数据以及逻辑型数据可以通用。一般高级语言的语法检查比较严格，几乎能检查出所有的语法错误。而 C 语言允许编程者有较大的自由度，因此放宽了语法检查。这样程序设计人员就有了更大的自主性，就能设计出更加灵活的程序。但是它对程序设计人员的要求却更高了，因此初学者一般较难掌握。

6) C 语言允许直接访问物理地址，能进行位运算，能实现汇编语言的大部分功能，可以直接对硬件进行操作。因此 C 语言既有高级语言的功能，又具有低级语言的许多功能，可以用来写系统软件。C 语言的这种双重性，使它既是成功的系统描述语言，又是通用的程序设计语言，因而有人把它称为“中级语言”。

7) 生成的目标代码质量高，因此程序执行效率高。一般只比汇编程序生成的目标代码效率低 10%~20%，因此可代替汇编语言编写系统软件。不过在某些对程序执行效率有特殊要求的编程中（如操作系统的核心部分），有时还是需要使用汇编语言编程。

8) 用 C 语言写的程序可移植性好（与汇编语言相比），基本不做修改就能用于各种型号的计算机和各种操作系统。

C 语言的以上特点，读者现在也许还不能深刻理解，待学完以后再回顾一下就会有比较深刻的体会。

1.3 C 程序简介

C 程序是由函数组成的，一个简单的 C 程序可以只由一个主函数组成，不过一般 C 程序中经常还包括一些用户自己定义的函数。下面通过几个简单的例子使大家对 C 程序及其特点有一个概括的了解。

【例 1.1】 简单的输入与输出。

```
main()                                /* 主函数 */
{ char name[20];                      /* 定义字符数组 */
  printf("What's your name ?");       /* 屏幕显示询问姓名 */
  scanf("%s",name);                  /* 通过键盘输入姓名 */
  printf("Hello %s,you are welcome !\n",name); /* 屏幕输出欢迎的句子 */
}
```

上述程序运行后，先在屏幕上输出英文句子“What's your name ?”询问您的姓名。当通过键盘输入姓名后，程序接着输出一个表示欢迎的句子。程序的执行情况如下：

```
What's your name ? ZhangLin ✓      (输入姓名)
Hello ZhangLin ,you are welcome !   (输出欢迎)
```

程序中由 main 开始的是主函数，在每个 C 程序中都必须包含一个并且只能有一个主函数，因为每个程序都是从主函数开始执行的。主函数的函数体是由一对花括号括起来的部分，程序中执行各种操作的语句都包括在函数体中。函数体中一般可包括多条语句，但也可以只包括一条甚至没有语句。程序中写在“/* ... */”内的不是语句，而是用于提高程序可读性的注释部分，它可以出现在程序中的任何位置。此外，C 语句的书写一律采用小写字母，每条语句都必须用“;”结束。