

高等院校“十三五”应用技能培养规划教材

• 移动应用开发系列

C语言 程序设计

杨连贺 赵玉玲 丁刚 主 编
池明文 李俊荣 副主编

- 本书全面涵盖C语言知识，内容新颖，按现代编程风格设计
- 集C语言经典教材之特点，内容上发展为“现代模式”体系
- 天津市高校教师讲课大赛冠军获得者联合其他专家共同编写



清华大学出版社

高等院校“十三五”应用技能培养规划教材·移动应用开发系列

C 语言

杨连贺 赵玉玲 丁 刚 主 编

池明文 李俊荣 副主编

清华大学出版社
北 京

内 容 简 介

C 语言是国内外使用最广泛的程序设计语言之一,更是高等院校工科学生应该具备的基础知识。学会使用 C 语言进行程序设计,也是计算机工作者必备的一项基本功。

本书系统而全面地介绍了 C 语言的知识,适合零基础的读者自学。在内容方面,由于位运算多用于检测和控制领域,常规编程较少触及,各类考试也很少涉及,故本书未将此部分内容纳入。

与一般的 C 语言教材相比,本书增加了“预备知识”和“VC 环境下 C 语言的运行与调试”两部分内容。前者主要解决晦涩难懂的补码问题,有助于读者理解 C 语言整型数据的存储;后者则有助于读者上机考试。书中每道例题的输出结果,均以屏幕截图的方式提供,原滋原味地呈现给读者,便于分析程序。

本书可作为普通高等院校及大专院校各专业的 C 语言教材,亦可供读者自学 C 语言使用,尤其适合作为各类 C 语言考试的培训教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计/杨连贺,赵玉玲,丁刚主编. —北京:清华大学出版社,2017

(高等院校“十三五”应用技能培养规划教材·移动应用开发系列)

ISBN 978-7-302-46211-8

I. ①C… II. ①杨… ②赵… ③丁… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 020029 号

责任编辑:汤涌涛

装帧设计:杨玉兰

责任校对:宋延清

责任印制:杨 艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62791865

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:23.25 插 页:1 字 数:590 千字

版 次:2017 年 3 月第 1 版 印 次:2017 年 3 月第 1 次印刷

印 数:1~3000

定 价:48.00 元

产品编号:072230-01

本书特色

本书根据应用的需要，合理取舍，精选内容，采用读者容易理解的方法，力求深入浅出、通俗易懂。

本书的主要特色如下。

- (1) 讲解预备知识，以飧基础薄弱者，利于无障碍地学会编程。
- (2) 增加了运行环境介绍，使用 VC 编译工具，接轨国家等级考试。
- (3) 详解程序调试，通过真题进行实战讲解，专门针对上机考试。
- (4) 例题都给出了程序运行的真实截屏效果，以保障所讲解的程序是能够运行的。
- (5) 习题经过细心琢磨，专为应试做准备，部分习题选自历年的考题。
- (6) 各章末尾的案例实训是精选的综合性案例，具有一定的实用性。
- (7) 篇末的项目实践贯穿主要的知识点，以加强系统性和实践性。
- (8) 精选了网上的一些内容，博采众家之长，集众多优质案例于一书。

前 言

计算机程序设计语言现今已有数百种，常用的也有数十种，但很少有哪一种语言像 C 语言那样充满活力，流行近半个世纪经久不衰；也很少有哪一种语言像 C 语言那样在中国如此普及。国家计算机等级考试中，各级考试均离不开 C 语言的身影；国内高等院校几乎都把计算机文化基础和 C 语言作为理工科学生步入计算机领域的“敲门砖”，C 语言程序设计已成为理工科学生必备的技能之一。

C 语言是 20 世纪 70 年代初期推出的编译型程序设计语言。经过近半个世纪的发展，C 语言表现出了强大的生命力，已经从早期在少数小型机机种上使用的算法语言，发展成了当今在各类平台上普遍使用的通用程序设计语言。因为 C 语言具有强大的处理功能，特别是在某些方面可以替代汇编语言，所以熟练掌握 C 语言是现代计算机工作者应具备的基本条件，也是学习更高层次面向对象的 C++ 语言的必备基础。

在近半个世纪的发展过程中，C 语言的体系日臻完善，功能不断增强。与此相应，各种新型的 C 编译系统相继问世，C 语言已从早期的“传统模式”发展为当前的“现代模式”。

鉴于目前流行的 C 语言编译系统大多数是以 ANSI C89 为基础开发的，所以本书以 C89 为基础并结合当前最常用的 VC 集成开发环境进行介绍，同时兼顾过去 20 年非常流行的 Turbo C 集成环境。总而言之，本书的宗旨在于向读者呈现一本内容新颖、系统全面、按现代编程风格介绍 C 语言程序设计的书籍。

本书共分为 11 章，各章的主要内容说明如下。

第 1 章介绍 C 语言的基础。考虑到计算机零基础读者的需要，本章适量增加了一些 C 语言预备知识。

第 2 章以 VC 为背景，介绍运行 C 语言程序的步骤与方法，并通过典型案例，介绍 C 语言程序的调试方法。

第 3 章介绍 C 语言的基本数据类型、常量、变量以及 C 语言的各种运算符与表达式。

第 4 章从结构化程序设计的角度，介绍 C 程序的流程设计，同时融入算法的概念。

第 5 章介绍最简单的构造类型——数组。

第 6 章介绍各种类型的函数，以及函数间数据传递的方法。

第 7 章介绍编译预处理。

第 8 章讨论 C 语言最重要的概念——指针及其特性。

第 9 章介绍 C 语言的另一种构造类型——结构体与共用体，同时，还介绍枚举类型和用 typedef 定义既有类型的方法。

第 10 章从实用出发，介绍用 C 语言处理文件的方法。

第 11 章为项目实践，用一个大案例贯穿本书的主要知识点。书后的附录列出了 ASCII 码表、C 语言关键字、常用库函数以及运算符的优先级与结合性。

此处提请读者注意：全书中紧随程序代码后的是程序的运行结果截图。在各截图标题栏中显示的文件名及路径仅是作者调试程序时使用的临时文件名和路径。

本书作者在多年从事 C 语言教学和软件开发的基础上,参考了国内外有关的 C 语言著作,参考了因特网上诸多的优秀程序,并参照了近几年来一直延用的《2013 年计算机二级 C 语言程序设计考试大纲》,完成了编著工作。因此,本书既可作为普通高等院校及大专院校的教材使用,又可供参加各类计算机考试的人员学习和参考。

本书由天津工业大学的杨连贺教授、天津市信息安全测评中心的赵玉玲、天津广播电视大学的丁刚副教授主编,池明文、李俊荣为副主编。在本书的编写过程中,还得到了天津大学软件学院的刘石英副研究员、高翔副研究员、曹军老师、沈士强老师的热情帮助和指导。天津农学院计算机与信息工程学院的靳润昭教授、烟台大学计算机与控制工程学院的贺利坚副教授、西北工业大学计算机学院的张彦春副教授提供了不少案例,天津工业大学计算机科学与软件学院的硕士研究生董禹龙和房超在程序调试方面也做了不少工作,在此一并向他们表示衷心的感谢。

限于作者的经验和水平,书中的错误与不足之处在所难免,希望得到专家和读者的批评指正。

作 者

2017 年 2 月于天津工业大学



杨连贺

博士，教授，博士生导师。1965年生于天津，1986年毕业于西北电讯工程学院（现西安电子科技大学），获工学学士学位；1989年、1998年毕业于天津纺织工学院（现天津工业大学），分别获工学硕士、工学博士学位。2000年破格晋升为教授。2004年遴选为博士生导师。2007—2008年在美国Texas A&M University做高级访问学者。从事复合材料力学性能、计算机仿真与辅助设计、数据库应用、信息系统开发、可视化计算、数据挖掘等领域的教学与研究工作，主持完成原中国纺织总会、天津市科委、天津市教委、教育部等项目十余项，获得两项天津市科研成果；近年来在国内外学术刊物上公开发表论文百余篇，其中多篇被EI收录；主编著作十余部。

1996年以讲授《C语言程序设计》课程在天津市第三届高校青年教师讲课大赛中夺冠。荣获国家教育部霍英东教育基金会青年教师奖，获校“青年骨干教师”称号，获桑麻奖教金。兼任天津市电子信息教育专家指导委员会委员，以及天津市科技进步奖、科技发明奖评审专家。

目 录

第 1 章 C 语言基础..... 1	第 3 章 数据描述..... 31
1.1 计算机语言的概念..... 2	3.1 C 语言的数据类型..... 32
1.2 C 语言的发展和特点..... 3	3.2 常量和变量..... 33
1.2.1 概述..... 3	3.2.1 常量和符号常量..... 33
1.2.2 C 语言的发展历史..... 4	3.2.2 变量..... 34
1.2.3 C 语言的特点..... 5	3.3 整型数据..... 34
1.3 计算机中数的表示及进制转换..... 6	3.3.1 整型常量的表示方法..... 34
1.3.1 数制的概念..... 6	3.3.2 整型变量..... 36
1.3.2 数码、基数与位权..... 6	3.4 实型数据..... 40
1.3.3 各种进制之间的转换..... 7	3.4.1 实型常量的表示方法..... 40
1.4 字节和位的概念..... 9	3.4.2 实型变量..... 41
1.5 整数的表示方法——原码、反码 和补码..... 10	3.4.3 实型常数的类型..... 42
1.5.1 整数的原码表示..... 10	3.5 字符型数据..... 42
1.5.2 整数的反码表示..... 10	3.5.1 字符常量..... 42
1.5.3 整数的补码表示..... 10	3.5.2 转义字符..... 43
1.6 书写程序时应遵循的规则..... 11	3.5.3 字符变量..... 44
1.7 C 语言程序的结构..... 11	3.5.4 字符数据在内存中的存储形式 及使用方法..... 44
1.7.1 源程序结构..... 11	3.5.5 字符串常量..... 45
1.7.2 C 语言的字符集..... 13	3.6 为变量赋初值..... 46
1.7.3 C 语言的词汇..... 14	3.7 算术运算符和算术表达式..... 46
1.8 简单的 C 程序举例..... 15	3.7.1 C 语言的运算符简介..... 47
1.9 运行 C 语言程序的步骤和方法..... 17	3.7.2 算术运算符和算术表达式..... 47
第 2 章 VC 环境下 C 程序的运行 和调试..... 19	3.8 赋值运算符和赋值表达式..... 50
2.1 在 VC 下运行一个 C 程序..... 20	3.9 逗号运算符和逗号表达式..... 52
2.1.1 启动 VC..... 20	3.10 各类数值型数据之间的混合运算..... 53
2.1.2 编辑程序..... 20	3.11 本章小结..... 56
2.1.3 编译程序..... 21	习题..... 57
2.1.4 运行程序..... 22	第 4 章 C 程序的流程设计..... 59
2.2 在 VC 下调试 C 程序的方法..... 23	4.1 C 语句概述..... 60
2.2.1 VC 调试功能简介..... 23	4.2 赋值语句..... 62
2.2.2 使用 VC 调试 C 程序的方法..... 26	4.3 数据输入输出的概念及在 C 语言中 的实现..... 63

4.4	字符数据的输入输出	63
4.4.1	putchar 函数(字符输出函数).....	63
4.4.2	getchar 函数(键盘输入函数).....	64
4.5	格式输入与输出	65
4.5.1	printf 函数(格式输出函数).....	65
4.5.2	scanf 函数(格式输入函数)	68
4.6	结构化程序设计的方法	74
4.7	顺序结构程序设计	76
4.7.1	顺序结构的程序	76
4.7.2	顺序结构程序的案例实训	76
4.8	选择结构程序设计	77
4.8.1	关系运算符和关系表达式	77
4.8.2	逻辑运算符和逻辑表达式	79
4.8.3	if 语句	81
4.8.4	switch 语句	89
4.8.5	分支结构程序的案例实训	91
4.9	循环结构程序的设计	94
4.9.1	概述	94
4.9.2	goto 语句以及用 goto 语句 构成的循环	95
4.9.3	while 语句	95
4.9.4	do-while 语句	98
4.9.5	for 语句	100
4.9.6	循环的嵌套	103
4.9.7	几种循环的比较	103
4.9.8	break 语句和 continue 语句	104
4.9.9	循环结构程序的案例实训	106
4.10	本章小结	110
	习题	113

第 5 章 数组

5.1	一维数组的定义和引用	118
5.1.1	一维数组的定义方式	118
5.1.2	一维数组元素的引用	119
5.1.3	一维数组的初始化	121
5.1.4	一维数组应用举例	122
5.2	二维数组的定义和引用	123
5.2.1	二维数组的定义	123
5.2.2	二维数组元素的引用	124
5.2.3	二维数组的初始化	126

5.2.4	二维数组程序举例	127
5.3	字符数组	128
5.3.1	字符数组的定义	128
5.3.2	字符数组的初始化	128
5.3.3	字符数组的引用	129
5.3.4	字符串及其结束标志	129
5.3.5	字符数组的输入输出	130
5.3.6	字符串处理函数	132
5.4	案例实训	135
5.5	本章小结	138
	习题	139

第 6 章 函数

6.1	概述	144
6.2	函数定义的一般形式	145
6.3	函数的参数和函数的值	147
6.3.1	形参和实参	147
6.3.2	函数的返回值	149
6.4	函数的调用与被调用函数的声明	150
6.4.1	函数调用的一般形式	150
6.4.2	函数调用的方式	150
6.4.3	被调用函数的声明和函数 原型	152
6.5	函数的嵌套调用	153
6.6	函数的递归调用	155
6.7	数组作为函数参数	157
6.8	局部变量和全局变量	163
6.8.1	局部变量	163
6.8.2	全局变量	165
6.9	变量的存储类别	167
6.9.1	动态存储方式与静态存储 方式	167
6.9.2	auto 变量	168
6.9.3	用 static 声明局部变量	168
6.9.4	register 变量	170
6.9.5	用 extern 声明外部变量	170
6.10	内部函数和外部函数	172
6.10.1	内部函数	172
6.10.2	外部函数	172
6.10.3	建立包含多个文件的程序	174

6.11 案例实训	177	8.7.1 指针数组的概念	247
6.12 本章小结	181	8.7.2 指向指针的指针	251
习题	183	8.7.3 main 函数的参数	253
第 7 章 编译预处理	189	8.8 有关指针的数据类型和指针运算	
7.1 概述	190	小结	254
7.2 宏定义	190	8.8.1 有关指针的数据类型小结	254
7.2.1 无参宏的定义	190	8.8.2 指针运算小结	255
7.2.2 带参宏定义	193	8.8.3 void 指针类型	256
7.3 文件包含	199	8.9 案例实训	256
7.4 条件编译	200	8.10 本章小结	258
7.5 案例实训	202	习题	259
7.6 本章小结	203	第 9 章 结构体与共用体	263
习题	204	9.1 结构体类型的定义	264
第 8 章 指针	207	9.2 结构体变量的说明	265
8.1 指针的基本概念	208	9.3 结构体变量成员的引用方法	267
8.2 变量的指针和指向变量的指针		9.4 结构体变量的赋值	268
变量	209	9.5 结构体变量的初始化	268
8.2.1 定义一个指针变量	209	9.6 结构体数组的定义	269
8.2.2 指针变量的引用	210	9.7 结构体指针变量的说明和使用	271
8.2.3 指针变量作为函数参数	214	9.7.1 指向结构体变量的指针	271
8.2.4 关于指针变量几个问题的		9.7.2 指向结构体数组的指针	273
进一步说明	218	9.7.3 结构体指针变量作为函数	
8.3 数组的指针和指向数组的指针		参数	274
变量	221	9.8 动态存储分配	276
8.3.1 指向数组元素的指针	222	9.9 用指针处理链表	278
8.3.2 通过指针引用数组元素	223	9.9.1 链表概述	278
8.3.3 数组名与指针变量作为函数的		9.9.2 链表的建立和输出	279
参数	227	9.10 共用体	288
8.3.4 指向多维数组的指针和指针		9.10.1 共用体的定义	288
变量	235	9.10.2 共用体的引用	289
8.4 字符串的指针和指向字符串的指针		9.11 枚举类型	290
变量	239	9.11.1 枚举类型的定义和枚举变量	
8.4.1 字符串的表示形式	239	的说明	290
8.4.2 字符串指针变量与字符数组的		9.11.2 枚举类型变量的赋值	
区别	244	和使用	290
8.5 函数指针变量	244	9.12 用 typedef 定义已有类型	292
8.6 指针型函数	246	9.13 案例实训	293
8.7 指针数组和指向指针的指针	247	9.14 本章小结	295
		习题	298

第 10 章 文件的输入输出	303
10.1 文件概述.....	304
10.2 文件指针.....	305
10.3 文件的打开与关闭.....	305
10.3.1 文件的打开(fopen 函数).....	305
10.3.2 文件的关闭(fclose 函数).....	307
10.4 文件的顺序读写.....	307
10.4.1 字符读写函数 fgetc 和 fputc.....	307
10.4.2 字符串读写函数 fgets 和 fputs.....	311
10.4.3 数据块读写函数 fread 和 fwrite.....	313
10.4.4 格式化读写函数 fscanf 和 fprintf.....	314
10.4.5 文件结束检测函数 feof.....	316
10.5 文件的随机读写.....	316
10.5.1 文件定位函数.....	317
10.5.2 文件随机读写.....	317

10.6 文件读写的出错检测.....	318
10.7 案例实训.....	319
10.8 本章小结.....	321
习题.....	321

第 11 章 项目实践	325
11.1 题目要求.....	326
11.2 系统分析.....	326
11.3 系统设计.....	326
11.3.1 总体设计.....	326
11.3.2 详细设计.....	327
11.4 系统实施.....	334
11.5 关于项目实践的几点说明.....	345
附录 A ASCII 代码表	347
附录 B C 语言中的关键字	349
附录 C C 语言的常用库函数	351
附录 D 运算符的优先级与结合性	359
参考文献	361

第

1

章

C 语言基础

本章要点

- (1) 计算机语言的基本概念，机器语言、汇编语言、高级语言。
- (2) C 语言的发展历史，C 语言的特点。
- (3) 预备知识：计算机中数的表示及进制转换；字节和位的概念；整数的表示方法——原码、反码和补码。
- (4) 书写 C 语言程序时应遵循的规则。
- (5) C 语言程序的结构。
- (6) 运行 C 语言程序的步骤与方法。

学习目标

- (1) 了解计算机语言的基本概念，了解机器语言、汇编语言、高级语言之间的区别。
- (2) 了解 C 语言的发展历史，熟悉 C 语言的特点。
- (3) 掌握计算机中数的表示及进制转换，了解字节和位的概念，熟悉整数的原码、反码表示方法；掌握整数的补码表示方法。
- (4) 熟悉书写 C 语言程序时应遵循的规则。
- (5) 熟悉 C 语言程序的结构。
- (6) 掌握运行 C 语言程序的上机步骤与方法。

1.1 计算机语言的概念

计算机语言(Computer Language)是指用于人与计算机之间通信的语言,它是人机之间传递信息的媒介。计算机系统最显著的特征,就是通过一种语言把指令传达给机器,机器自动地、高速地完成指定的工作。为了使计算机能够完成各种工作,就需要有一套用以编写计算机程序的数字、字符和语法规则,由这些字符和语法规则组成各种指令(或各种语句),就是计算机所能接受的语言。

为什么不能使用人类的自然语言作为计算机语言呢?这是因为,自然语言具有二义性甚至多义性。例如,某人说:“小红和小兰的妈妈来了。”到底来的一个人(两个孩子的共同妈妈)还是两个人(小红、小兰各自的妈妈),语法上是说不清的。正因如此,自然语言不能为计算机所用。换言之,计算机语言必须是一种人造的、无二义的、专门用于计算机的语言。

计算机语言的种类是非常多的,但总体来说,可以分成机器语言、汇编语言、高级语言三大类。

1. 机器语言——第一代计算机语言

计算机发明之初,人们只能吃力地用很低级的语言去命令计算机做事情。换言之,就是必须写出一串串由“0”和“1”组成的指令序列,交由计算机去执行。这种计算机能够直接识别的语言,就是机器语言。使用机器语言编程是十分痛苦的,阅读程序如读天书一般,当程序有错,需要修改时,就更是这样了。

一条机器语言语句通常称为一条指令。指令是不可分割的最小功能单元。而且,由于每台计算机的指令系统往往各不相同,所以,在一台计算机上执行的程序,想移植到另一台计算机上执行时,必须另编程序。但由于使用的是针对特定型号计算机的语言,故而运算效率也是所有语言中最高的。

2. 汇编语言——第二代计算机语言

为了减轻使用机器语言编程的痛苦,人们进行了一种有益的改进:用一些简洁的助记符来替代一个特定指令的二进制串。比如,用 ADD 代表加法,用 MOV 代表数据传递等。这样一来,人们就很容易读懂并理解程序的功能,纠错及维护都变得方便多了。这种编程语言就称为汇编语言,即第二代计算机语言。然而,计算机是不能直接认识这些助记符的,这就需要有一个专门的程序,负责将这些符号翻译成二进制的机器语言。这类翻译程序被称为汇编语言程序。

汇编语言同样依赖于机器硬件,可移植性不好,但效率仍十分高(与机器语言相同)。针对计算机特定硬件而编制的汇编语言程序,能准确发挥计算机硬件的功能和特长,程序精炼而且质量高,所以汇编语言至今仍是一种常用的强有力的软件开发工具。

汇编语言的实质与机器语言是一样的,都是直接对硬件进行操作,只不过指令采用了助记符,更容易识别和记忆。它同样需要编程者将每一步具体的操作以命令的形式写出来。

汇编语言程序的每一条指令只能对应实际操作过程中一个很细微的动作,例如加减、移动、自增等。因此,汇编语言源程序一般比较冗长、复杂、容易出错,而且使用汇编语

言编程需要有更多的计算机专业知识。但汇编语言的优点也是显而易见的,用汇编语言所能完成的操作不是一般高级语言所能实现的,而且源程序经汇编生成的可执行文件不仅比较小,而且执行速度很快。

3. 高级语言——第三代计算机语言

从最初与计算机交流的艰难经历中,人们意识到,应该设计一种这样的语言,这种语言接近于数学语言或人类的自然语言,同时又不依赖于计算机硬件,编出的程序能在所有计算机上通用。

经过计算机科学家的努力,1954年,世界上第一种完全脱离机器硬件的高级语言 FORTRAN 问世了。从那以后,半个多世纪以来,已有数百种高级语言出现,有重要意义的有几十种,影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、Basic、LISP、SNOBOL、PL/1、Pascal、C、Prolog、Ada、C++、Delphi、Java 等。

特别要强调的是,在 C 语言诞生以前,系统软件主要是用汇编语言编写的。由于汇编语言程序高度依赖于计算机硬件,所以其可读性和可移植性都很差;但一般的高级语言又难以实现对计算机硬件的直接操作(而这正是汇编语言的优势),于是人们期盼有一种兼有汇编语言和高级语言特性的新语言出现,C 语言由此应运而生。

高级语言是绝大多数编程者的选择。与汇编语言相比,它不但将许多相关的机器指令合成为单条指令,而且去掉了与具体操作有关但与完成工作无关的细节,例如使用堆栈、寄存器等,这样就大大简化了编程。同时,由于屏蔽了很多细节,所以编程者也无需太多的专业知识。

与汇编语言一样,计算机不能直接识别这些高级语言的指令(语句),需要有一个专门的程序负责将这些语句转换为二进制的机器语言,按转换方式的不同,可将它们分为两类:解释程序和编译程序。

在解释执行的语言中,程序是随着每条指令的执行而逐条语句地翻译成目标代码的。解释执行的语言的优点是,它们可以立即开始执行程序,而不需要等到所有的源代码都得到编译。对程序的更改也可以相当快地进行,无须等到程序重新编译完。

解释执行的语言的缺点在于,它们执行起来慢,因为每次运行程序,都必须对整个程序一次一条指令地翻译。

编译执行的语言与此完全不同,它是将使用高级编程语言编写的源代码一次性地翻译成目标代码,不仅只需编译一次,而且执行起来也比解释执行的语言快得多。由于这个原因,编译执行的语言比解释执行的语言更常用,而且,在专业和科学领域中,几乎总是应用编译执行的语言。

C 语言属于编译执行的语言,而且拥有众多的编译程序,几乎遍及所有平台。

1.2 C 语言的发展和特点

1.2.1 概述

虽然 C 语言是一种高级语言,但它兼有汇编语言的特点。有鉴于此,曾有人将其称为“中级语言”。它可以作为系统程序的编程语言,编写系统软件(如操作系统、编译程序等),

也可以作为应用程序的编程语言，编写不依赖于计算机硬件的应用程序。因此，它的应用范围十分广泛，在要求运行速度快、占用内存小的大多数场合中，C 语言都是非常理想的一种选择。

在对操作系统进行编程以及需要对硬件进行操作的场合中，C 语言明显优于其他解释型高级语言，很多大型应用软件都是用 C 语言编写的。

C 语言绘图能力强，可移植性好，并具备很强的数据处理能力，因此适用于编写系统软件以及二维、三维图形和动画程序，在数值计算领域的应用也十分广泛。在单片机以及嵌入式系统开发方面，C 语言更是功不可没。

常用的 C 语言集成开发环境(Integrated Development Environment, IDE)有 Microsoft Visual C++、Borland C++、Watcom C++、Borland C++ Builder、Borland C++ 3.1 for DOS、GNU DJGPP C++、LCC Win32 C Compiler 3.1、Microsoft C、High C、Turbo C 等。

1.2.2 C 语言的发展历史

C 语言的原型为 ALGOL 60 语言(也称为 A 语言)。

1963 年，剑桥大学将 ALGOL 60 发展成为组合编程语言(Combined Programming Language, CPL)。1967 年，剑桥大学的 Martin Richards 对 CPL 语言进行了简化，产生了 BCPL 语言。1970 年，美国贝尔实验室的 Ken Thompson 对 BCPL 进行了修改，并为它起了一个有趣的名字“B 语言”。其意思是，将 BCPL 语言“煮干”，提炼出它的精华。Ken Thompson 用 B 语言写了第一个 Unix 操作系统。

而在 1973 年，B 语言也被人“煮”了一下。美国贝尔实验室的 Dennis M. Ritchie 在 B 语言的基础上最终设计出了一种新的语言，他取了 BCPL 的第二字母作为这种语言的名字，这就是 C 语言。

为了使 Unix 操作系统得以推广，Dennis M. Ritchie 于 1977 年发表了不依赖于具体计算机系统的 C 语言编译文本“可移植的 C 语言编译程序”，即是著名的 ANSI C 的最初版本。

1978 年，Brian W. Kernighan 和 Dennis M. Ritchie 合著了著名的《The C Programming Language》一书，通常简称为 K&R，也有人称为 K&R 标准。但是，在 K&R 中，并没有定义一个完整的标准 C 语言，后来由美国国家标准协会(American National Standards Institute, ANSI)在此基础上制定了一个 C 语言标准，于 1983 年发表，通常称为 ANSI C。

K&R 在很多语言细节上还不够精确，甚至没有很好地表达它所描述的语言，把后续扩展扔到了一边。最后，C 在早期项目中的使用受商业和政府合同支配，意味着一个认可的正式标准是很重要的。因此，在 M.D. Mcilroy 的催促下，ANSI 于 1983 年夏天，在 CBEMA 的领导下建立了 X3J11 委员会，目的是产生一个 C 标准。X3J11 在 1989 年末提出了一个他们的报告(ANSI 89)。

1990 年，ISO 接受了 89 ANSI C 为 ISO C 的标准(ISO 9899:1990)。

1994 年，ISO 修订了 C 语言的标准。

1995 年，ISO 又做了一些修订，即“1995 基准增补 1(ISO/IEC/9899/AMD1:1995)”。

1999 年，ISO 又对 C 语言标准进行了修订，在基本保留原来 C 语言特征的基础上，根据需要，增加了一些功能，尤其是针对 C++ 中一些功能的扩展，命名为 ISO/IEC9899:1999。

2001 年和 2004 年，ISO 先后又对 C 标准进行了两次技术修正。

C 语言的发展非常迅速,而且成为最受欢迎的编程语言之一。许多著名的软件,如 Oracle、dBASE III Plus、dBASE IV、Apache、PHP、Python 等,都是用 C 语言编写的。

C 语言与一些汇编语言子程序相结合,就更能显示 C 语言的优势了,像个人计算机第一代操作系统 PC-DOS 和 20 世纪末风靡全球的 WordStar 等,就是用这种方式开发的。当然,最著名的当属 Unix 操作系统,其中大约 95%的代码是用 C 编写的,只有内核是用汇编语言编写的,仅占全部工作量的 5%。

当今的很多高级语言(包括一些脚本语言)都是从 C 语言发展而来的,典型的当属时下最为热门的 Java 语言。

尽管 C 语言的体系日臻完善,功能不断增强,而且各种新型的 C 编译系统层出不穷,但目前流行的 C 语言编译系统大多数还是以 89 ANSI C(简称 C89)为基础开发的,所以本书仍以 C89 为基础,并结合国家计算机等级考试专用的 VC 集成开发环境进行介绍,但兼顾 Turbo C 集成环境。

1.2.3 C 语言的特点

C 语言自 20 世纪 70 年代问世以后,历经 40 余年,经久不衰,这与其特点是分不开的。C 语言的主要特点如下。

(1) C 语言简洁、紧凑,使用方便、灵活。不同的 C 编译系统使用的关键字不尽相同,ANSI C 一共只有 32 个关键字(如表 1-1 所示),9 条控制语句(如表 1-2 所示),程序书写自由,主要用小写字母表示,压缩了一切不必要的成分。

表 1-1 ANSI C 的关键字

auto	break	case	char	const	continue	default
do	double	else	enum	extern	float	for
goto	if	int	long	register	return	short
signed	static	sizeof	struct	switch	typedef	union
unsigned	void	volatile	while			

表 1-2 C 语言的 9 条控制语句

类 别	语 句
条件判断语句	if 语句, switch 语句
循环执行语句	do while 语句, while 语句, for 语句
转向语句	break 语句, goto 语句, continue 语句, return 语句

(2) 运算符丰富。C 语言共有 34 种运算符,它把括号、赋值、逗号等都作为运算符处理。从而使 C 的运算类型极为丰富,可以实现其他高级语言难以实现的各种运算。

(3) 数据结构类型丰富。

(4) 具有结构化的控制语句。

(5) 语法限制不太严格,程序设计自由度大。

(6) C 语言允许直接访问物理地址,能进行位操作,能实现汇编语言的大部分功能,可

以直接对硬件设备的端口进行操作。

(7) 生成的目标代码质量高, 程序执行效率高。

(8) C 语言程序的可移植性好。

然而, C 语言也并非十全十美。例如, 输入输出跟很多其他语言相比都较为复杂, 语法限制不太严格, 类型检查机制相对较弱, 对数组下标越界不做检查, 指针操作带来了许多不安全因素等。此外, C 语言对程序员的要求也比较高, 学习时比 Basic、Pascal、FORTRAN 等高级语言的难度要大。

1.3 计算机中数的表示及进制转换

本节及随后的两节内容不在 C 语言之列, 属于 C 语言的预备知识, 但对于初学 C 语言的读者而言, 很有必要熟悉这些内容。如果读者曾系统地学过计算机文化基础之类的课程, 则建议跳过这三节内容, 直接进入第 1.6 节的学习。

1.3.1 数制的概念

所谓“数制”, 又称“计数制”, 是指用一组固定的计数符号(称为数符或数码)和一套统一的规则表示数值的方法。

按进位的规则进行计数, 称为进位计数制。在不引起混淆的情况下, 我们将进位计数制简称为“数制”或“进制”。在日常生活中经常要用到数制, 通常以十进制进行计数。除了十进制计数以外, 还有许多非十进制的计数方法。例如, 60 分钟为 1 小时, 用的是六十进制计数法; 1 星期有 7 天, 是七进制计数法; 1 年有 12 个月, 是十二进制计数法。当然, 在生活中, 还有许多其他各种各样的进制计数法。不过, 计算机系统中采用的是二进制, 其主要原因是由于电路设计简易、运算简单、工作可靠、逻辑性强。无论是哪一种数制, 其计数和运算都有共同的规律和特点。

在进位计数法中, 数码序列中相同的一个数码所表示的数值大小, 与它在该数码序列中的位置有关。进位计数制有三个要素: 数码、基数与位权。

1.3.2 数码、基数与位权

(1) 数码: 用不同的数字符号来表示一种数制的数值, 这些数字符号称为“数码”。例如, 十进制所用的数码是 0、1、2、3、4、5、6、7、8、9, 二进制所使用的数码为 0 和 1, 十六进制则使用 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 作为数码。

(2) 基数: 数制所使用的数码个数称为“基数”, 亦称为“基”。例如, 十进制的基数为 10, 二进制的基数为 2。数制的进位遵循“逢 N 进 1”的规则, 其中 N 就是数制中的基数。

(3) 位权: 某数制中每一位所具有的权重称为“位权”, 亦称“权”。位权与基数的关系是: 各进位制中, 位权的值是基数的若干次幂。例如, 十进制数 256.78 中, 2 的位权为 $10^2=100$, 而 5 的位权为 $10^1=10$, 个位上 6 的位权为 $10^0=1$, 十分位上 7 的位权为 $10^{-1}=0.1$, 百分位上 8 的位权为 $10^{-2}=0.01$ 。