



新编应用型系列技能丛书

C程序设计 与实例

樊学东 ● 主编 马军红 薛慧芳 ● 副主编

● PPT教学课件

● 配备习题答案

针对性和实用性强

内容丰富、实例翔实

教学资源下载: www.tup.com.cn

清华大学出版社



新编应用型系列技能丛书

C 程序设计与实例

主 编 樊学东

副主编 马军红 薛慧芳

清华大学出版社

北 京

内 容 简 介

本书循序渐进地讲解了 C 语言的基本概念、原理和使用方法,力求为读者打下一个扎实的程序设计基础,培养读者设计程序的能力。主要内容包括:C 语言基本知识、数据类型、运算符和表达式、程序结构、函数、数组、编译预处理、指针、自定义数据类型、位运算、综合实例等。本书采用深入浅出、通俗易懂的讲解方法,本着理论与实践相结合的原则,通过大量经典实例对 C 语言知识进行了重点而系统的讲解,使初学者能够利用 C 语言进行结构化程序设计并掌握编程技术和方法。同时,本书以实例配合教学,通过典型实例实现类比、渐进、深入到综合编程,从而脱离纯理论的枯燥说教,提高教学质量和效率。

本书适用于高等学校各专业的程序设计基础教学,尤其适合应用型本科、高职院校的计算机及非计算机相关专业的学生使用,同时也是一本实用的计算机等级考试备考辅导书,还可作为单片机等各类工程技术人员的学习参考用书。

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 程序设计与实例/樊学东主编. —北京:清华大学出版社,2014
(新编应用型系列技能丛书)

ISBN 978-7-302-34484-1

I. ①C… II. ①樊… III. ①C 语言-程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 274306 号

责任编辑:杜长清

封面设计:刘 超

版式设计:文森时代

责任校对:赵丽杰

责任印制:何 芊

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

印 装 者:三河市吉祥印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:18.5 字 数:477 千字

版 次:2014 年 2 月第 1 版 印 次:2014 年 2 月第 1 次印刷

印 数:1~3800

定 价:32.80 元

产品编号:052629-01

编审委员会

主 任：陈绥阳

副 主 任：谢膺白 李宝敏 樊学东 贺亚茹 张文胜

委 员：（按姓氏笔画排序）

马军红	王元一	王红刚	王征风	王英强	王 栋
王 娜	王艳君	王振辉	王雅静	石永生	任 华
任志宏	汤宏萍	陈 宏	杜晓春	李金良	李校红
李 继	李 梅	杨华勇	苏智华	张小木	张 龙
张首军	张 敏	张 伟	张娓娓	林 青	孟晓丽
范训礼	郑长风	宫 丽	侯亚玲	赵向梅	赵金龙
赵福祥	黄玉蕾	殷亚玲	唐 明	梁计锋	薛慧芳

策划编辑：苏明芳

序

Preface

2012年12月,清华大学出版社在西安组织了两次电子和计算机类的教材讨论会,着重评论了以培养应用型人才为目标的高等院校和高职高专的教材编写问题。本系列丛书,即是针对应用型人才培养的一组教材。

面向培养应用型人才的学校的教师和学生,提供一组电子和计算机类的教材,不仅是市场细分的要求,而且是应用型人才培养目标与培养模式的要求。在以学历文凭为目的的教育中,以应试为目标,其教学是以知识点为主,而不强调应用。高职高专则强调其社会属性,是面向工作岗位的,采用“校企合作、工学结合、顶岗实习”的培养模式,提倡情景教学与面向工作任务的工作过程式教学方法,其教材是技能导向的。以培养应用型人才为目标的高等院校,一方面有别于高职类型的普通本科,其专业既有学科属性,表现为知识的基础性、系统性、完整性和时序性,又有社会分工的社会属性;另一方面,“应用型”则表示该专业的人才培养目标,是为地区经济建设与社会发展服务的。兼顾这两方面的要求,这是本组教材的基本特征。

专业的培养方案是规范培养过程以达到培养目标的基本文件,其中的教学计划与课程大纲(标准)则建立了教学的课程体系与内容体系。建立这一体系是复杂而细致的工作,通常不是个人能力所及的。目前,市场上现有的教材过于强调教材自身的封闭性,造成教材内容的过度冗余与教材间内容的过度重复,往往也是这一原因而造成了教材的局限性。参加本组教材编写的院校,在多年的专业实践中,为适应市场的需求,不断进行课程体系与内容体系的建设工作,同时进行内容的梳理,在本组教材的编写过程中采用集体讨论、集体编写的方法,有助于体现相应的成果。

本系列教材中,有的课程是相关院校的精品课程、重点课程建设项目,在教学方法上有所探究,也有经验与教训。应当看到,精品课程、重点课程建设首先是针对教师的,是为教师提供一个可供示范的样本和教学资料。但教材首先是面向学生的,本系列教材是面向应用型本科的学生,这有明确的定位。在这一定位上,有的教材在写作风格上追求“让学生能读懂”,这是比较难做到的。事实上,在一本教材中,既有教师指导学生阅读的部分,又有学生自学的部分,这体现了教学过程中教师的主导作用,也是基于建构主义学习理论的。教师编写并出版的仅仅是文本,只有经过教师的使用和学生的阅读,才成为教材。

电子和计算机行业是技术发展很快的行业,反映到教材中要兼顾基础性、工具性和现代性。在该系列教材中,有从基于Windows系统的程序架构到基于Android系统的应用程序设计,有从单片机接口技术与应用实践到嵌入式系统,有从语言类教材到基于数据库的.NET架构,既反映了基于PC(Personal Computer, 个人计算机)体系结构与互联网结构的应用软件系统设计,又反映了基于非PC体系结构的先进计算机系统与移动互联网的应用软件系统设计,既反映了Java中的J2EE技术,又反映了微软C#的技术,以适合B/S结构而满足中小型企业信息化建设的市场需求,读者有较大选择空间。

值得重视的是,在教材的编写中,编者引入了“计算思维”(《计算思维》,卡内基梅隆大学计算机系主任周以真)的观点,即不将计算机科学局限于编写代码,而着重于计算思维的培养。同时,在写作过程中,在涵盖知识点(用知识结构图表述)的基础上,采用基于任务的讲述过程,重视四个层次的实验,即认知性实验、验证性实验、设计性实验和综合性实验,并通过综合实例培养学生的应用能力。同时,积极进行立体化教材编写:含大纲,教学用 PPT、习题、习题答案、模拟试卷、模拟试卷答案、实训指导书等。

在本丛书的成书过程中,编者参考了多本相关书籍,作为附录加以注明,同时又得到清华大学出版社的大力支持,尤其是编辑苏明芳在编辑出版等方面做了大量工作,在此一并感谢。

由于编者学识有限,书中难免挂一漏万,存在不妥之处,敬请读者斧正。

陈安阳

前言

Foreword

目前, C 语言程序设计已不仅是高等院校计算机专业的必修课, 很多高校已经将其作为各专业公共基础课, 在计算机知识日新月异的时代, 可见其重要性。关于 C 语言程序设计的教材有很多, 但是成体系的、有针对性的、应用型的 C 语言教材却不多。具有针对性和实用性、概念清晰、通俗易懂是我们编写本书的目的。

我们知道, 有了 C 语言, 在一定条件下可以不用汇编语言编程, 而且还可以进一步学习 C++ 等面向对象的程序设计语言, 同时 C 语言在片上编程等很多领域也有很大的用处。即使不用 C 语言编写程序, 甚至不从事程序编制工作, C 语言的教育意义仍然存在。比如通过学习 C 语言可以掌握面向过程的编程思想, 掌握如何把实际问题转化为建立流程图、功能模型、拆分等方法, 以提高逻辑思维能力。需要注意的是, 面向对象的语言并不能替代面向过程的语言, 面向过程语言的编程思想是学习计算机编程的基础。

C 语言功能丰富、表达能力强、使用灵活方便、应用广泛、目标程序效率高、可移植性好, 既具有高级语言的优点, 又有低级语言的特性。当然从另一个角度看, C 语言涉及的概念比较复杂, 规则多, 使用灵活, 因此容易出错。本书针对初学者的特点和认知规律, 精选内容, 分散难点, 深入浅出, 以例题习题为引导, 从实用性出发, 提高学生的动手能力。

本书采用 Turbo C 2.0 编译系统, 读者在学习过程中也可以使用 Visual C++ 6.0 对程序进行编译, 因为 C++ 语言是兼容 C 语言的。另外, Turbo C 3.0 是基于 Windows 操作系统的, 支持鼠标操作, 读者如果有兴趣, 可以自行查看相关资料。

学习程序设计课程的目的是掌握程序设计的思路, 学会用计算机语言编写程序, 实现处理问题和任务。所以在学习过程中, 一定要把重点放在应用计算机语言解决问题的方法和思路, 而不是死记硬背语法规则。

学习任何语言都需要实践, 编写好的程序需要上机运行, 反复修改调试, 才能得到正确结果。因此, 掌握好一门高级语言, 需要坚持算法简洁、思路清晰、举一反三、善于总结的原则。衡量对 C 语言知识的掌握程度, 不在于“知不知道”, 而在于“会不会干”, 即不是看选择题、填空题能否答对, 而是看所编写的程序能不能正确地通过调试运行, 运行结果是不是正确。

本书针对知识点配备有相应的例题、习题、综合实例, 读者在学习过程中应掌握最基本、最常用的内容, 打好基本功。本书第 14 章为课程设计环节, 按其可完成有一定规模和实践性的程序设计, 可以提高程序设计的综合能力。

本书的第 1~6 章及附录由樊学东编写, 第 7~10 章由马军红编写, 第 11~14 章由薛慧芳编写。樊学东负责全书的统稿和审阅工作。由于编者水平有限, 书中难免存在一些缺点和错误, 殷切希望广大读者批评指正。

编 者

目 录

Contents

第 1 章 C 语言基础.....	1
1.1 计算机系统.....	1
1.2 C 语言的特点	2
1.3 从实例开发第一个 C 语言程序	3
1.3.1 C 语言程序的结构	4
1.3.2 C 语言程序设计风格	6
1.4 上机环境.....	8
1.4.1 安装 Turbo C 2.0.....	9
1.4.2 调用 Turbo C	9
1.4.3 编辑、编译、连接、运行的基本操作	10
1.5 结构化程序设计方法.....	12
本章自测题.....	13
第 2 章 数据类型	15
2.1 C 语言的数据类型	15
2.2 常量和变量.....	16
2.2.1 常量和符号常量.....	16
2.2.2 标识符与变量.....	17
2.3 整型数据.....	18
2.3.1 整型常量.....	18
2.3.2 整型变量.....	19
2.4 实型数据.....	21
2.4.1 实型常量的表示方法.....	21
2.4.2 实型变量.....	22
2.4.3 双精度型数据.....	22
2.5 字符型数据.....	23
2.5.1 字符常量.....	23
2.5.2 字符变量.....	24
2.5.3 字符串常量.....	26
2.6 变量初始化.....	26
本章自测题.....	27

第 3 章 运算符和表达式	29
3.1 C 语言运算符简介	29
3.2 算术运算符和算术表达式	30
3.2.1 算术运算符	30
3.2.2 算术表达式和运算符的优先级与结合性	30
3.3 自增、自减运算符	31
3.4 强制类型转换运算符及类型转换	32
3.5 赋值运算符和赋值表达式	33
3.6 逗号运算符和逗号表达式	35
3.7 优先级和结合性	36
本章自测题	37
第 4 章 顺序结构程序设计	39
4.1 C 语句概述	39
4.2 赋值语句	40
4.3 格式化输入/输出	41
4.3.1 printf 函数	42
4.3.2 scanf 函数	45
4.4 字符数据的输入/输出	48
4.4.1 putchar 函数	48
4.4.2 getchar 函数	49
4.5 顺序结构程序举例	49
本章自测题	51
第 5 章 选择结构程序设计	54
5.1 关系运算符和关系表达式	54
5.1.1 关系运算符	54
5.1.2 关系表达式	54
5.2 逻辑运算符和逻辑表达式	55
5.2.1 逻辑运算符	55
5.2.2 逻辑表达式	55
5.3 if 语句和条件运算符	57
5.3.1 if 语句	57
5.3.2 if 语句的嵌套	59
5.3.3 条件运算符	60
5.4 switch 语句	61
5.5 程序举例	63
本章自测题	65

第 6 章 循环结构程序设计	68
6.1 概述	68
6.2 while 语句	68
6.2.1 循环的引出	68
6.2.2 while 语句的一般形式及执行过程	70
6.3 do-while 语句	72
6.4 for 语句	73
6.4.1 for 语句的一般形式	73
6.4.2 for 语句形式的多样性	75
6.4.3 几种循环的比较	76
6.5 循环的嵌套	77
6.6 goto、break 和 continue 语句	78
6.6.1 goto 语句	78
6.6.2 break 语句	79
6.6.3 continue 语句	80
6.7 程序举例	80
本章自测题	82
第 7 章 函数	85
7.1 函数的定义	85
7.1.1 函数定义的一般方式	86
7.1.2 形式参数与实际参数	89
7.2 函数的调用	90
7.2.1 函数调用的形式	90
7.2.2 函数调用的方式	90
7.3 函数返回值和函数类型说明	91
7.3.1 函数的返回值	91
7.3.2 函数的类型声明	93
7.4 函数的嵌套调用和递归调用	94
7.4.1 函数的嵌套调用	94
7.4.2 递归调用的形式	95
7.4.3 递归函数的使用	97
7.4.4 消去递归	98
7.5 变量存储类别	99
7.5.1 局部变量与全局变量	99
7.5.2 自动变量	101
7.5.3 寄存器变量	102
7.5.4 外部变量	102
7.5.5 静态变量	104

7.6 内部函数与外部函数.....	105
7.6.1 内部函数.....	105
7.6.2 外部函数.....	106
7.7 程序举例.....	106
本章自测题.....	108
第 8 章 数组	113
8.1 一维数组的定义和引用.....	113
8.1.1 一维数组的定义.....	113
8.1.2 一维数组元素的引用.....	115
8.1.3 一维数组的初始化.....	115
8.1.4 应用举例.....	116
8.2 二维数组的定义和引用.....	119
8.2.1 二维数组元素的定义.....	119
8.2.2 二维数组元素的引用.....	120
8.2.3 二维数组元素的初始化.....	120
8.2.4 二维数组的应用.....	122
8.3 字符数组的定义和引用.....	124
8.3.1 字符数组元素的定义.....	124
8.3.2 字符数组的引用.....	125
8.3.3 字符数组的初始化.....	125
8.3.4 字符数组与字符串.....	126
8.3.5 字符串处理函数.....	128
8.3.6 应用举例.....	131
8.4 数组作为参数.....	132
本章自测题.....	136
第 9 章 编译预处理	140
9.1 宏定义.....	140
9.1.1 不带参数的宏定义.....	140
9.1.2 带参数的宏定义.....	143
9.2 文件包含.....	145
9.3 条件编译.....	146
本章自测题.....	149
第 10 章 指针	154
10.1 地址和指针.....	154
10.2 变量的指针.....	155
10.2.1 指针变量的定义.....	156
10.2.2 指针变量的引用.....	156

10.3	指针运算.....	159
10.4	指针变量作为函数的参数.....	161
10.5	指针和数组.....	164
10.5.1	指向数组元素的指针变量.....	164
10.5.2	通过指针引用数组元素.....	165
10.5.3	数组名作函数参数.....	168
10.5.4	多维数组的指针.....	171
10.6	字符串的指针.....	175
10.6.1	字符串的表示和应用.....	176
10.6.2	字符串指针作函数参数.....	180
10.7	指针数组和指向指针的指针.....	181
10.7.1	指针数组.....	181
10.7.2	多级指针.....	184
10.7.3	命令行参数.....	186
10.8	函数的指针和返回指针值的函数.....	187
10.8.1	函数的指针.....	187
10.8.2	返回指针的函数.....	190
10.9	指针应用举例.....	191
	本章自测题.....	194
第 11 章	结构体、共同体和枚举.....	198
11.1	结构体类型和结构体变量的定义.....	198
11.1.1	结构体类型的定义.....	198
11.1.2	结构体类型变量的定义.....	199
11.2	结构体变量的引用和初始化.....	201
11.2.1	结构体变量的引用.....	201
11.2.2	结构体变量的初始化.....	202
11.3	结构体数组.....	202
11.3.1	结构体数组的定义.....	202
11.3.2	结构体数组的初始化.....	203
11.3.3	结构体数组的应用实例.....	203
11.4	结构体和指针.....	205
11.4.1	指向结构体变量的指针.....	205
11.4.2	用结构体指针变量访问结构体成员.....	205
11.4.3	结构体指针的应用——链表.....	207
11.5	共用体.....	211
11.5.1	共用体的概念.....	211
11.5.2	共用体变量及其引用.....	211
11.6	枚举类型.....	215

11.7 用 typedef 声明新类型名	216
本章自测题	217
第 12 章 位运算	224
12.1 位运算符和位运算	224
12.1.1 按位取反运算	224
12.1.2 按位“与”运算	225
12.1.3 按位“或”运算	226
12.1.4 按位“异或”运算	226
12.1.5 左移运算	227
12.1.6 右移运算	228
12.2 位运算举例	229
12.3 位段简介	230
本章自测题	232
第 13 章 文件	235
13.1 文件概述	235
13.1.1 使用文件的目的	235
13.1.2 文件的概念与分类	235
13.1.3 流式文件的概念	236
13.1.4 文件指针	236
13.2 文件的打开与关闭	237
13.2.1 文件的打开 (fopen 函数)	237
13.2.2 文件的关闭 (fclose 函数)	239
13.3 文件的读/写	239
13.3.1 以字符方式读/写文件	239
13.3.2 fgets 和 fputs 函数	241
13.3.3 fread 和 fwrite 函数	243
13.3.4 fscanf 和 fprintf 函数	245
13.4 文件的定位	246
13.4.1 rewind 函数	246
13.4.2 fseek 函数	247
13.4.3 ftell 函数	248
13.5 出错的检测	249
13.5.1 ferror 函数	249
13.5.2 clearerr 函数	250
本章自测题	250
第 14 章 综合程序案例	254
14.1 模块化程序设计概述	254

14.1.1 模块化程序设计的基本思想	254
14.1.2 模块化程序设计的一般步骤	254
14.1.3 模块化程序设计的基本要求	255
14.2 综合实例	255
参考文献	264
附录 A 常用字符与 ASCII 代码对照表	265
附录 B C 语言中的关键字	267
附录 C 运算符和结合性	268
附录 D 库函数	270
附录 E C 语言常用语法摘要	275

第 1 章

C 语言基础

【本章内容简介】

C 语言是一种计算机程序设计语言。从 1972 年问世以后, C 语言已成为国际上应用广泛的高级语言, 特别适合于开发系统软件、单片机控制编程、片上控制编程, 同时它又是学习 C++、Java 等面向对象程序设计语言的基础。

- 函数: C 程序的基本单位是函数。一个 C 程序由一个主函数和若干个自定义函数构成, 每个函数都由函数首部和函数体组成。
- 结构化程序设计: 由 3 种基本结构组成, 每种结构都只有一个入口、一个出口; 没有死循环和死语句; 遵循自顶向下、逐步求精, 模块化设计和结构化编码的原则。
- Turbo C 集成环境: 集编辑、编译、连接和运行于一体的环境。

【本章重点难点】

本章是学习 C 语言程序设计的基础, 重点要掌握 C 程序的总体结构和编程环境。本章无难点。

1.1 计算机系统

计算机系统由硬件系统和软件系统组成。

计算机硬件是组成计算机的物理设备的总称, 由中央处理器 (运算器和控制器)、内部存储器、外部设备和接口、总线等部件和电子线路组成, 是计算机完成计算工作的物质基础。

计算机软件是在计算机硬件设备上运行的各种程序和相关资料的总称。

计算机硬件和软件是相互依存、相互促进的关系, 在某些功能上是等价的, 例如常常听说的“硬件软化, 软件硬化”等, 一些软件功能可以用硬件实现, 一些硬件功能可以由软件完成。没有软件的计算机通常称为“裸机”, 裸机是无法工作的。系统软件中的操作系统是最贴近硬件的系统软件, 是用户和计算机的接口, 统一控制和管理计算机软件和硬件资源, 保证计算机各个部分协调有序地工作。

计算机语言是面向计算机的语言, 是进行程序设计的工具, 因此也称程序设计语言。程序设计语言可以分为机器语言、汇编语言、高级语言。高级语言中一种是面向问题的程序设计语言, 它要对具体的算法和过程进行描述, 也可以称为面向过程的算法语言, 例如早期的 FORTRAN 语言、BASIC 语言、Pascal 语言和 C 语言等。另一种属于人工智能语言, 例如 PROLOG 语言。还有一种是当前流行的面向对象的程序设计语言, 例如 C++、Java、C# 语言等, 它们都是以 C 语

言为基础的,学了 C 语言再学习其他语言就方便多了。

计算机能直接执行的是机器语言程序。汇编语言也称符号语言,用汇编语言编写的程序称汇编语言程序。计算机不能识别和直接运行汇编语言程序,必须由汇编程序翻译成机器语言程序后才能识别和运行。同样,高级语言程序也不能被计算机直接识别和执行,也必须把高级语言程序翻译成机器语言程序才能执行。语言处理程序就是完成这个翻译过程的,按照处理方式的不同,分为解释型程序和编译型程序两大类。C 语言采用编译程序,把源程序编译成目标程序,再通过连接程序,生成可执行程序才能运行。具体过程将在 1.4 节详细说明。

1.2 C 语言的特点

C 语言是当前广为流行的程序设计语言,适合于编写系统软件,也可以编写应用软件。

以前的操作系统等系统软件主要是用汇编语言编写的(包括 UNIX 操作系统)。由于汇编语言依赖于计算机硬件,程序的可读性和可移植性都比较差。为了提高可读性和可移植性,最好使用高级语言,但一般的高级语言难以实现对计算机硬件的直接操作,例如对内存地址的操作、位操作等,于是人们期望找到一种既具有高级语言特性,又具有汇编语言特性的新语言。C 语言就是在这种情况下应运而生的。

C 语言和 UNIX 操作系统是一对孪生兄弟,在发展过程中相辅相成,紧密地联系在一起。当初 D. M. 里奇(D.Ritchie)开发 C 语言主要是为了更好地描述 UNIX 操作系统,自从有了 C 语言之后,UNIX 系统上的所有命令解释程序、各种实用程序(包括 C 语言编译程序)和后来在 UNIX 上开发的所有应用系统(数据库管理系统、网络通信系统等)都是用 C 语言编写的。UNIX 系统的许多特点,特别是易于剪裁和易于移植等主要受益于 C 语言。C 语言之所以有今天,正是由于它植根于 UNIX 系统之上,在 UNIX 系统上开发程序,可得到 UNIX 系统的强有力的支持。在编写 C 语言程序时可以直接使用 UNIX 操作系统所提供的许多功能,特别是输入/输出库函数和系统调用等。C 语言现在已独立于 UNIX 系统和开发时使用的 PDP 计算机,移植到大、中、小和微型计算机上,风靡全球,成为世界上应用最广泛的计算机语言之一。

1960 年出现的 Algol 60 是一种比较复杂的语言,它的结构非常严谨,其设计者非常注重语法分析、分程序结构,对后来的程序设计语言,特别是 Pascal、PL/1 等都有极大的影响。但由于它是面向过程的语言,不宜用来编写系统软件。1963 年英国剑桥大学仿照 Algol 60 推出了 CPL (Combined Programming Language),虽然它接近硬件一些,但其规模仍然过于庞大和复杂,难以实现和使用。1967 年剑桥大学的 M. 里查德(Martin Richard)对 CPL 做了简化,推出了 BCPL(Basic CPL),保持了 CPL 的基本特点。1970 年美国贝尔实验室的 K. 汤普逊(Ken Thompson)对 BCPL 做了进一步简化,设计出了非常简单而又接近于硬件的 B 语言(取 BCPL 的第一个字母),并用 B 语言编写了第一个 UNIX 操作系统,在 PDP-7 计算机上得以实现。但 BCPL 和 B 语言在简化语言上能力有限,只能用于特殊用途而不能解决一般常见的许多问题。1972—1973 年间贝尔实验室的 D. M. 里奇(D.M.Ritchie)把一些缺少的功能加入 B 语言并作了整体规划和整理,设计出了既能描述计算机硬件,又能适应数值计算、正文处理和数据处理需要的通用 C 语言(取 BCPL 的第二个字母)。C 语言既保持了 BCPL 和 B 语言接近硬件、精炼的优点,又克服了无数据类型、过于简单的缺点。最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工作语言而设计的,只在贝

尔实验室内部和大学研究机构中使用。后来又对 C 语言做了多次改进,直到 1975 年 UNIX 第 6 版公布以后,C 语言的突出优点才引起人们的普遍注意。1977 年出现了不依赖于具体机器的 C 语言编译文本《可移植 C 语言编译程序》,使将 C 语言移植到其他机器时所需做的工作大大简化,这也推进了 UNIX 操作系统迅速在各种机器上实现,C 语言进而得到广泛推广,成为人们进行程序设计特别是软件开发、资料准备、正文处理中得心应手的工具。

C 语言之所以能存在、发展并具有强大生命力是因为它有如下特点:

(1) 简洁、紧凑,使用方便、灵活。C 语言共有 32 个关键字,9 条控制语句,而且有功能很强、非常丰富的运算符(34 种)。例如自增(++)运算符、自减(--)运算符、复合赋值运算符、条件运算符,并且把括号、赋值、强制类型转换等都作为运算符处理,从而使 C 语言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。对软件生产率的研究表明,一个程序员在一个工作日里只能编制少量的工作程序,采用简洁的语言可以明显地提高程序设计者的生产效率。

(2) 表达能力强。C 语言可以直接访问地址,能进行位(bit)操作,直接操纵硬件,完成通常由硬件实现的普通的算术运算、逻辑运算,甚至足以取代汇编语言编写各种系统软件和应用软件,UNIX 操作系统的 90% 以上是用 C 语言编写的就是一个明证。当然,C 语言同样可以表述数值处理、文字处理和数据处理等,所以它又是一种通用语言。

(3) 数据结构丰富。C 语言的数据类型包括整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型、枚举类型等,基本上具有现代化语言的各种数据结构,能实现各种复杂的数据结构(如链表、树、堆栈)的运算。

(4) 模块化结构。C 语言具有结构化的控制语句(如分支、循环)。以函数作为程序的模块单位,便于程序的分别编写、分别调试,加快软件的开发过程,是理想的结构化语言。

(5) 可移植性好。所谓可移植性是指一个程序可以不加修改或稍加修改就能在不同的机器或不同操作系统环境下运行。C 语言程序具有很好的可移植性。

(6) C 语言语法限制不太严格,书写格式比较自由。

(7) 生成的目标代码质量高,程序执行效率高。C 语言生成的目标代码效率一般只比汇编语言低 10%~20%。

任何事物都是一分为二的。C 语言有众多的优点,当然也存在一定的不足,主要是:

(1) C 语言的数据类型区分严格,类型转换分为自动转换和强制转换两种。

(2) 没有动态数组界限检查。

(3) 允许编译程序在表达式中重新安排计算顺序和参数表,容易产生副作用。

C 语言灵活性高、限制少、功能强的特点对程序员的要求更高,初学者会感到困难,但是只要认真掌握它的实质,多读程序,多动手编程序,多上机调试程序,注意总结学习中的经验和体会,是一定可以学好的。

1.3 从实例开发第一个 C 语言程序

按一定顺序安排的工作——操作序列,称为程序。用来描述完成某项功能所涉及的对象和动作规则。计算机程序就是要计算机完成某项功能所涉及的对象和动作规则,即由计算机进行解释