



21世纪高职高专规划教材·计算机系列



C语言程序设计

冯艳茹 主 编

冯高峰 张小娜 副主编

国防科技大学出版社

21 世纪高职高专规划教材

计算机系列

C 语言程序设计

冯艳茹 主 编

冯高峰 张小娜 副主编

国防科技大学出版社

【内容简介】C语言是目前最为流行的通用程序设计语言之一,是许多计算机专业人员和计算机爱好者学习程序设计语言的首选。本书共8章,内容包括C语言基础、C语言中的数据、结构化程序设计、模块化程序设计、数组、指针、用户自定义数据类型和文件。

本书注重可读性和实用性,安排了大量有代表性的例题,这些例题程序由浅入深,强化了知识点、算法、编程方法与技巧,并给出了详细的解释。书中的全部例题程序均已在 Visual C++ 6.0 中调试通过。另外,本书还配套有《C语言程序设计习题解答与上机指导》教材。

本书适合高职高专学生使用,也可以作为全国计算机等级考试参考书,还可供自学者参考。

图书在版编目(CIP)数据

C语言程序设计/冯艳茹主编. —长沙:国防科技大学出版社,2010.3

ISBN 978-7-81099-740-9

I. C… II. 冯… III. C语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2010)第 030935 号

出版发行:国防科技大学出版社

网 址: <http://www.gfkdcbs.com>

责任编辑:耿 筠 特约编辑:朱秀英

印 刷 者:北京振兴源印务有限公司

开 本:787mm×1 092mm 1/16

印 张:18

字 数:449 千字

版 次:2010 年 3 月第 1 版

印 次:2010 年 3 月第 1 次印刷

定 价:28.00 元

21 世纪高职高专规划教材·计算机系列

编审委员会

顾 问 郑启华 清华大学教授
计算机教育资深专家

主 任 黄维通 清华大学计算机科学与技术系
全国计算机基础教育研究会副秘书长

副 主 任 李 俊 清华大学信息科学技术学院
骆海峰 北京大学软件与微电子学院
梁振方 上海交通大学电子信息与电气工程学院

委 员 (以姓氏笔画为序)

卫世浩	王玉芬	王军号	王建平	卢云宏
付俊辉	朱广丽	刘庆杰	刘春霞	江 枫
李永波	李光杰	李克东	李学勇	张春飞
张 岩	郑 义	姚海军	高国红	徐桂保
殷晓波	程华安	谢广彬	詹 林	

课程审定 张 歆 清华大学信息科学技术学院
战 扬 北京大学软件与微电子学院

内容审定 倪铭辰 清华大学信息科学技术学院
谢力军 北京大学软件与微电子学院
李振华 北京航空航天大学计算机学院

出版说明

高职高专教育作为我国高等教育的重要组成部分,承担着培养高素质技术、技能型人才的重任。近年来,在国家和社会的支持下,我国的高职高专教育取得了不小的成就,但随着我国经济的腾飞,高技能人才的缺乏越来越成为影响我国经济进一步快速健康发展的瓶颈。这一现状对于我国高职高专教育的改革和发展而言,既是挑战,更是机遇。

要加快高职高专教育改革的步伐,就必须对课程体系和教学模式等问题进行探索。在这个过程中,教材的建设与改革无疑起着至关重要的基础性作用,高质量的教材是培养高素质人才的保证。高职高专教材作为体现高职高专教育特色的知识载体和教学的基本工具,直接关系到高职高专教育能否为社会培养并输送符合要求的高技能人才。

为促进高职高专教育的发展,加强教材建设,教育部在《关于全面提高高等职业教育教学质量的若干意见》中,提出了“重点建设好3 000种左右国家规划教材”的建议和要求,并对高职高专教材的修订提出了一定的标准。为了顺应当前我国高职高专教育的发展潮流,推动高职高专教材的建设,我们精心组织了一批具有丰富教学和科研经验的人员成立了21世纪高职高专规划教材编审委员会。

编审委员会依据教育部高教司制定的《高职高专教育基础课程教学基本要求》和《高职高专教育专业人才培养目标及规格》,调研了百余所具有代表性的高等职业技术学院和高等专科学校,广泛而深入地了解了高职高专的专业和课程设置,系统地研究了课程的体系结构,同时充分汲取各院校在探索培养应用型人才方面取得的成功经验,并在教材出版的各个环节设置专业的审定人员进行严格审查,从而确保了整套教材“突出行业需求,突出职业的核心能力”的特色。

本套教材的编写遵循以下原则:

- (1) 成立教材编审委员会,由编审委员会进行教材的规划与评审。
- (2) 按照人才培养方案以及教学大纲的需要,严格遵循高职高专院校各学科的专业规范,同时最大程度地体现高职高专教育的特点及时代发展的要求。因此,本套教材非常注重培养学生的实践技能,力避传统教材“全而深”的教学模式,将“教、学、做”有机地融为一体,在教给学生知识的同时,强化了对学生实际操作能力的培养。
- (3) 教材的定位更加强调“以就业为导向”,因此也更为科学。教育部对我国的高职高专教育提出了“以应用为目的,以必需、够用为度”的原则。根据这一原则,本套教材在编写过程中,力求从实际应用的需要出发,尽量减少枯燥、实用性不强的理论灌输,充分体现“以行业为向导,以能力为本,以学生为中心”的风格,从而使本套教材更具实用性和前瞻性,与就业市场结合也更为紧密。
- (4) 采用“以案例导入教学”的编写模式。本套教材力图突破陈旧的教育理念,在讲解的过程中,援引大量鲜明实用的案例进行分析,紧密结合实际,以达到编写实训教材的

目标。这些精心设计的案例不但可以方便教师授课,同时又可以启发学生思考,加快对学生实践能力的培养,改革人才的培养模式。

本套教材涵盖了公共基础课系列、财经管理系列、物流管理系列、电子商务系列、计算机系列、电子信息系列、机械系列、汽车系列和化学化工系列的主要课程。目前已经规划的教材系列名称如下:

财经管理系列

- 财经管理基础课
- 工商管理类
- 财务会计类
- 经济贸易类
- 财政金融类
- 市场营销类

机械系列

- 机械基础课
- 机械设计与制造类
- 数控技术类
- 模具设计与制造类
- 机电一体化类

计算机系列

- 公共基础课
- 计算机专业基础课
- 计算机网络技术类
- 计算机软件技术类
- 计算机应用技术类

公共基础课系列

物流管理系列

电子商务系列

电子信息系列

化学化工系列

汽车系列

对于教材出版及使用过程中遇到的各种问题,欢迎您通过电子邮件及时与我们取得联系(联系方式详见“教师服务登记表”)。同时,我们希望有更多经验丰富的教师加入到我们的行列当中,编写出更多符合高职高专教学需要的高质量教材,为我国的高职高专教育做出积极的贡献。

21 世纪高职高专规划教材编审委员会

序

21 世纪是科技和经济高速发展的重要时期。随着我国经济的持续快速健康发展, 各行各业对高技能专业型人才的需求量迅速增加, 对人才素质的要求也越来越高。高职高专教育作为我国高等教育的重要组成部分, 在加快培养高技能专业型人才方面发挥着重要的作用。

与国外相比, 我国高职高专教育起步时间短, 这种状况与我国经济发展对人才大量需求的现状是很不协调的。因此, 必须加快高职高专教育的发展步伐, 提高应用型人才的培养水平。

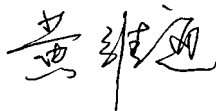
高职高专教育水平的提高, 离不开课程体系的完善。相关领域人才的培养需要一批兼具前瞻性和实践性的优秀教材。教育部高教司针对高职高专教育人才培养模式提出了“以就业为导向”的指导思想, 这也正是本套高职高专教材的编写宗旨和依据。

如何使高职高专教材既突出行业的需求特点, 又突出职业的核心能力? 这是教材编写的过程中必须首先解决的问题。本系列教材编委会深入研究了高职高专教育的课程和专业设置, 并对以往的教材进行了详细分析和认真考察, 力图在不破坏教材系统性的前提下, 加强教材的创新和实践性内容, 从而确保学生在学习专业知识的同时多动手, 增强自己的实践能力, 以加强“知”与“行”的结合。

同时, 本系列教材在编写过程中还充分重视群体和类别的差异性, 面对不同学校 and 不同专业方向的定位差异, 精心设计了与其相配套的辅助实验指南及相关的习题解答等。这些栏目的设计使本系列教材内容更加丰富, 条理更为清晰, 为老师的讲授和学生的学习都提供了很大的便利。

经过编委会的辛勤努力, 本套教材终于顺利出版了, 相信本套教材一定能够很好地适应现代高职高专教育的教学需求, 也一定能够在高职高专教育计算机课程的改革中发挥积极的推动作用, 为社会培养更多优秀的应用型人才。

全国计算机基础教育研究会副秘书长



前 言

C 语言是一种结构化程序设计语言,具有编程方便、语句简练、功能强、移植性好等特点,是软件开发中最常用的计算机语言。它既具有高级语言的特点,又具有汇编语言的特点;既适用于系统软件的开发,也适合于应用软件的开发,而且其编写不依赖于计算机硬件。因此,C 语言以其自身的特色和优势,在工业控制、嵌入式系统开发、单片机编程、系统底层开发等领域应用非常广泛。许多高校将 C 语言作为入门的程序设计教学语言。

全书共有 8 章和 5 个附录,第 1 章主要介绍 C 语言的基础知识和 C 语言的运行步骤;第 2 章主要介绍 C 语言中的基本数据类型,输入/输出函数的格式与应用;第 3 章主要介绍算法的概念以及结构化程序设计的 3 种基本结构——顺序结构、选择结构和循环结构;第 4 章主要介绍函数的定义、调用的方法以及变量的使用和 C 语言中预处理命令的使用;第 5 章主要介绍数值型数组、字符串以及常用的字符串处理函数;第 6 章主要介绍指针变量的定义、引用方法以及数组的指针和数组元素的指针表示方法等;第 7 章主要介绍结构体和共用体的定义和引用方法;第 8 章主要介绍文件操作的基本步骤和方法以及常用的读写操作函数;附录 A 列举了 C 语言运算符的优先级与结合性;附录 B 列举了 C 语言中的关键字;附录 C 列举了常用字符与 ASCII 码表;附录 D 列举了 C 语言常用的标准库函数;附录 E 简单介绍了 Turbo C++ 3.0 集成环境。

本书内容合理、实例丰富,讲解深入浅出,既注重培养学习者的程序设计能力,又提倡良好的程序设计风格。本书中所有的案例程序代码均在 Visual C++ 6.0 环境下调试通过。为了便于教学,本书配有教学资料包,包括教学课件、源代码、课后习题参考答案、期末测试卷等。另外,本书还配套有《C 语言程序设计习题解答与上机指导》教材。

本书由冯艳茹任主编,冯高峰、张小娜任副主编,参加编写工作的还有孙建国和张晓利。其中,张晓利编写了第 2 章,孙建国编写了第 3 章,张小娜编写了第 4 章、第 6 章和第 7 章,其余章节的编写和全书的统稿工作由冯艳茹完成。

在编写本书的过程中,参考了大量书籍,得到了许多同志的支持,在此向广大同仁和所参考书籍的作者表示衷心的感谢。

由于时间仓促,编者水平有限,书中难免存在一些不妥之处,恳请读者提出宝贵意见。

编 者

目 录

第 1 章 C 语言基础	1
1.1 程序与程序设计语言	1
1.1.1 计算机与程序	1
1.1.2 计算机语言	1
1.1.3 高级语言程序的开发过程	3
1.2 C 语言概述	4
1.2.1 C 语言的发展.....	4
1.2.2 C 语言的标准.....	4
1.2.3 C 语言的特点.....	5
1.3 C 程序的基本结构	5
1.3.1 一个简单的 C 程序	5
1.3.2 C 程序的组成.....	6
1.4 运行 C 程序的步骤和方法	8
1.4.1 C 程序的编译环境介绍	8
1.4.2 运行一个简单的 C 语言程序	9
本章小结	13
习题 1	13
第 2 章 C 语言中的数据	14
2.1 概述	14
2.1.1 数据及数据处理	14
2.1.2 数据处理的步骤	14
2.2 数据类型	14
2.3 常量	15
2.4 变量	18
2.4.1 变量的定义	19
2.4.2 变量的分类	19
2.4.3 变量的初始化	22
2.5 数据的输出	23
2.5.1 printf 函数	23
2.5.2 putchar 函数	27
2.6 数据的输入	27
2.6.1 scanf 函数	28

2.6.2	getchar 函数	31
2.7	运算符与表达式	32
2.7.1	赋值运算	32
2.7.2	算术运算	33
2.7.3	自增、自减运算符	35
2.7.4	复合赋值运算符	36
2.7.5	逗号运算符和逗号表达式	37
2.8	不同数据类型间的转换	39
2.8.1	数据类型的隐式转换	39
2.8.2	数据类型的显式转换	39
	本章小结	40
	习题 2	41
第 3 章	结构化程序设计	42
3.1	算法	42
3.1.1	算法概述	42
3.1.2	算法的描述	43
3.1.3	算法实现的过程	45
3.1.4	C 语言程序设计的基本步骤	45
3.2	顺序结构程序设计	46
3.3	选择结构程序设计	48
3.3.1	条件的描述和条件表达式	49
3.3.2	if 语句	53
3.3.3	if 语句的嵌套	59
3.3.4	条件运算符	62
3.3.5	switch 语句	64
3.4	循环结构程序设计	68
3.4.1	迭代与穷举算法	68
3.4.2	while 语句	71
3.4.3	do...while 语句	74
3.4.4	for 语句	76
3.4.5	循环中断控制语句	80
3.4.6	循环的嵌套	82
	本章小结	85
	习题 3	86
第 4 章	模块化程序设计	88
4.1	函数	88
4.1.1	概述	88

4.1.2	函数的分类、定义与调用	89
4.1.3	函数的返回值	95
4.1.4	形实结合与传值调用和传址调用	97
4.1.5	函数的嵌套调用与递归调用	100
4.2	变量的使用范围	105
4.2.1	变量的作用域与生存期	105
4.2.2	C语言中变量的存储类型	109
4.3	C语言预处理	114
4.3.1	宏定义	114
4.3.2	文件包含	122
4.3.3	条件编译	122
	本章小结	125
	习题4	125
第5章	数组	129
5.1	数值型数组	129
5.1.1	一维数组	129
5.1.2	二维数组	140
5.2	字符数组	148
5.2.1	一维字符数组和二维字符数组	148
5.2.2	字符串的输入/输出	150
5.2.3	字符串处理函数	152
	本章小结	156
	习题5	156
第6章	指针	160
6.1	指针基础	160
6.1.1	地址与指针	160
6.1.2	指针变量及其定义	161
6.1.3	指针变量的引用	161
6.1.4	指针变量作为函数参数	165
6.2	指针与数组	168
6.2.1	指向数组元素的指针	168
6.2.2	通过指针引用数组元素	169
6.2.3	指向多维数组的指针和指针变量	172
6.2.4	字符串指针与指向字符串的指针变量	177
6.2.5	指针数组和指向指针的指针	183
6.3	指针与函数	191
6.3.1	用指针实现函数的地址传递	191

6.3.2	返回指针值的函数	192
6.3.3	指向函数的指针	193
6.4	有关指针的数据类型和指针变量的运算	194
6.4.1	有关指针的各种定义及其含义	194
6.4.2	指针变量的运算	195
6.4.3	void 指针类型	196
本章小结		196
习题 6		197
第 7 章	用户自定义数据类型	200
7.1	结构体	200
7.1.1	结构体类型及其定义	200
7.1.2	结构体变量的定义与引用	201
7.1.3	结构体数组	205
7.1.4	结构体类型的指针	209
7.2	共用体	212
7.2.1	共用体的定义	212
7.2.2	共用体变量的应用	213
7.3	动态内存的分配	215
7.4	链表	217
7.5	枚举类型	220
7.5.1	枚举类型的定义	220
7.5.2	枚举类型变量的应用	221
7.6	用 typedef 定义数据类型	223
本章小结		226
习题 7		226
第 8 章	文件	230
8.1	概述	230
8.1.1	文件的概念	230
8.1.2	文件操作的基本方法和步骤	230
8.2	文件的打开和关闭	231
8.2.1	文件指针	231
8.2.2	文件的打开	232
8.2.3	文件的关闭	234
8.3	顺序读写文件	235
8.3.1	fputc 函数和 fgetc 函数	235
8.3.2	fputs 函数和 fgets 函数	241
8.3.3	fwrite 函数和 fread 函数	243

8.3.4	fprintf 函数和 fscanf 函数	247
8.4	随机读写文件	249
8.4.1	文件位置指针的定位	249
8.4.2	文件随机读写程序举例	251
8.5	文件操作的出错检测	252
8.6	常用文件操作函数小结	253
	本章小结	253
	习题 8	254
附录		256
附录 A	C 语言运算符的优先级与结合性	256
附录 B	C 语言中的关键字	257
附录 C	常用字符与 ASCII 码表	257
附录 D	C 语言常用的标准库函数	258
附录 E	Turbo C++ 3.0 的集成环境简介	265
参考文献		269

第 1 章 C 语言基础

掌握程序设计的前提是掌握程序设计语言,在众多程序设计语言中,C 语言以其灵活性和实用性受到了广大计算机应用人员的喜爱。C 语言是一种既得到美国国家标准化协会(ANSI)标准化又得到工业界广泛支持的计算机语言。几乎任何一种机型、任何一种操作系统都支持 C 语言开发。C 语言在巩固其原有应用领域的同时,又在拓展新的应用领域。C 语言支持大型数据库开发和 Internet 应用,一旦掌握了 C 语言,就可以较为轻松地学习其他任何一种程序设计语言,并为后续的面向对象程序设计、Windows 程序设计、Java 程序设计等程序设计语言的学习打下基础。

1.1 程序与程序设计语言

1.1.1 计算机与程序

1946 年,世界上第一台电子计算机问世。随着计算机科学及其应用的迅猛发展,计算机被广泛地应用于人类生产、生活的各个领域,推动了社会的进步与发展。特别是随着国际互联网(Internet)日益深入千家万户,传统的信息收集、传输及交换方式正在发生革命性的改变,计算机已将人类带入了一个新的时代——信息时代。

计算机是一种能快速、高效地对各种信息进行存储和处理的电子设备。它按照人们事先编写的程序对输入的原始数据进行加工处理、存储或传送,以得到预期的输出信息,并利用这些信息来提高社会效率,改善人们的生活质量。

计算机是怎样工作的呢?它又是如何懂得人类语言的呢?计算机是“聪明傻瓜”,不告诉它怎样干它什么也不会干,告诉它怎样干它会干得很好。使计算机明白一个个的命令,它就会按照你的命令去干。这种给计算机的命令,称为指令。指令是计算机要执行的一种基本操作命令,是对计算机进行程序控制的最小单位。指令由操作码和操作数构成,它们分别表示何种操作和存储地址。一连串计算机指令的集合,叫做程序。

所谓程序,就是要计算机完成某一任务所规定的一系列动作或步骤。没有程序和程序设计,计算机将不能做任何事情,即程序(软件)是计算机的必要组成部分。

1.1.2 计算机语言

要使计算机能够很好地为人类服务,人类与计算机之间必须通过一种语言来相互沟通和交流。这种既能够表达人类的思想,又能够被计算机所识别、接受的语言就叫做计算机语言。

计算机程序设计语言的发展,经历了从机器语言、汇编语言、高级语言到面向对象的程序设计语言的历程。

1. 机器语言

机器语言被称为第一代计算机语言。机器语言是最底层的计算机语言,是用二进制代码指令表达的计算机语言。它能被计算机硬件直接识别并执行,由操作码和操作数组成,如指令 00111110,00000111。机器语言程序编写的难度较大且不容易移植,即针对一种计算机编写的机器语言程序不能在另一种计算机上运行。

2. 汇编语言

汇编语言是用助记符代替操作码,用地址符代替操作数的一种面向机器的低级语言,一条汇编指令对应一条机器指令。例如,用“ADD”代表加法,“MOV”代表数据传递等。这样,人们很容易理解程序在做什么,纠错及维护就变得方便了。这种程序设计语言也称为编译语言,即第二代计算机语言。由于汇编语言采用了助记符,它比机器语言易于修改、编写、阅读,但用汇编语言编写的程序(称为汇编语言源程序)机器不能直接执行,必须使用编译程序把它翻译成机器语言即目标程序后,才能被机器理解和执行,这个编译的过程称为汇编。

汇编语言同样依赖于计算机硬件,移植性不好,但效率仍十分高。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,程序精练而质量高。所以,它至今仍是一种常用且强有力的软件开发工具。

汇编语言和机器语言都是面向机器的语言,称为低级语言。

3. 高级语言

从最初与计算机交流的痛苦经历中,人们意识到,应该设计一种语言,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,编写出的程序能在所有机器上通用。经过不断努力,1954年,第一个完全脱离计算机硬件的高级语言——FORTRAN 语言问世了。50多年来,共有几百种高级语言出现,其中,有重要意义的有几十种。

高级语言是直接面向过程的程序设计语言,它与具体的计算机硬件无关。用高级语言编写的源程序可以直接运行在不同机型上,因而具有通用性。但是,计算机不能直接识别和运行高级语言程序,必须经过“翻译”。所谓“翻译”,就是由一种特殊程序把源程序转换成目标程序,这种特殊程序被称为语言处理程序。高级语言的翻译方式有两种:一种是“编译方式”,另一种是“解释方式”。编译方式是通过编译程序将整个高级语言源程序翻译成目标程序,再经过连接程序生成可执行程序(.EXE);解释方式是通过解释程序边解释边执行,不产生目标程序。

最常用的高级语言有 BASIC,C,PASCAL,FORTRAN 等。

20 世纪 60 年代中后期,软件越来越多,规模越来越大,而软件的生产基本上是各自为战,缺乏科学规范的系统规划与测试和评估标准。其后果是大批耗费巨资建立起来的软件系统,由于错误而无法使用,从而带来巨大损失。软件给人的感觉越来越不可靠,几乎没有不出错的软件。这一切,极大地震动了计算机界,史称“软件危机”。人们认识到:大型程序的编制不同于写小程序,它应该是一项新的技术,应该像处理工程一样处理软件研制的全过程。程序的设计应易于保证正确性,也便于验证正确性。1969 年,结构化程序设计方法被提出,1970 年,第一个结构化程序设计语言——PASCAL 语言出现,标志着结构化程序设计时期的开始。

20 世纪 80 年代初开始,在软件设计思想上,又产生了一次革命,其成果就是面向对象的程序设计。

高级语言的下一个发展目标是面向应用,即只需要告诉程序要干什么,程序就能自动生成算法,自动进行处理,这就是非过程化的程序语言。

4. 面向对象的程序设计语言

面向对象的程序设计语言,一般具有可视化、网络化、多媒体等功能。目前比较流行的有 Visual Basic, Visual C++, Java, Delphi, PowerBuilder 等。

1.1.3 高级语言程序的开发过程

程序是为解决某一问题而编写的语句的集合,编写程序的过程就是程序设计。概括地说,程序设计就是分析问题、设计算法、编写程序、调试程序的过程。在最初的程序设计步骤中,把解决问题的过程看成是数据被加工的过程,基于这种方法的程序设计称为面向过程的程序设计。

C语言是面向过程的结构化程序设计语言,它适合用自顶向下的方法进行软件开发。该方法采用逐步求精的设计过程,主要有6个步骤:确定问题、分析问题、设计算法、实现算法、程序的测试与调试和程序的维护。本书的案例分析都采用该方法。

1. 确定问题

确定问题就是要明白程序要解决的问题。一般情况下,一个具体的问题要涉及诸多方面,这是问题的复杂性所在。为了便于解决这个问题,往往需要忽略一些次要的方面,而把注意力集中在问题的实质上。

2. 分析问题

分析问题就是要确定问题中的数据以及数据之间的联系。例如,求解两个数之和的问题,需要输入的数据是两个数,要输出的结果是两个数的和,如何由两个数计算出它们的和,则是算法要解决的问题。

3. 设计算法

设计算法就是要用一种符号或语言来描述解决问题的具体步骤和方法。一般采用自顶向下、逐步求精的设计原则和方法。先从最原始的问题入手,按照功能把问题逐步分解为若干子问题,子问题还可以再分解为若干下一级子问题,直到不能再分解为止。

4. 实现算法

实现算法就是要用计算机语言来实现具体的算法。在这个环节中,把描述算法的每个步骤转变为一种程序设计语言中的一条或多条语句,最后把算法转变为计算机程序。

5. 程序的测试与调试

程序的测试是在程序通过编译、不存在语法错误和连接错误的前提下,找出程序中可能存在的错误并对其进行改正,因此,应该测试程序的运行情况。输入不同的数据可以检测出程序在不同情况下运行的状况,看程序是否完成预期的功能,是否能够顺利通过和实现程序的设计目的。若有错误必须对程序进行修改,直到正确为止。

6. 程序的维护

程序在交付使用之后对其进行的修改称为程序的维护。程序维护的目的是更正程序设计中的漏洞,升级和更新程序。

1.2 C 语言概述

C 语言是国际上广泛流行的、很有发展前途的计算机高级语言,它集高级语言和低级语言的功能于一体。它适合作系统描述语言,既可用于编写系统软件,也可用来开发应用软件。本节介绍 C 语言的发展历程,C 语言的标准及特点。

1.2.1 C 语言的发展

以前的系统软件主要是用汇编语言编写的。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。为了提高程序的可读性和可移植性,最好改用高级语言,但一般高级语言难以实现汇编语言的某些功能。人们设想能否找到一种既具有一般高级语言特性,又具有低级语言特性的语言。C 语言就在这种情况下应运而生了。

在 20 世纪 60 年代,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程序语言。1970 年,美国贝尔实验室的 Ken Thompson 在软件开发工作中,继承和发展了 BCPL 语言的特点,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 UNIX 操作系统就是使用 B 语言记述和开发的。但 B 语言过于简单,功能有限。1972 年至 1973 年间,美国贝尔实验室的 Dennis M. Ritchie 在 B 语言的基础上设计出了 C 语言。

早期的 C 语言主要用于 UNIX 系统。后来,C 语言进行了多次改进,但主要还是在贝尔实验室内部使用。直到 1975 年 UNIX 第 6 版公布后,C 语言的突出优点才引起人们的普遍注意。由于 C 语言的强大功能和各方面的优点逐渐为人们所认识,到了 20 世纪 80 年代,C 开始进入其他操作系统,并很快在各类大、中、小和微型计算机上得到广泛的使用。现在 C 语言已经风靡全世界,成为世界上应用最广泛的几种计算机语言之一。

1.2.2 C 语言的标准

1978 年,Brian W. Kernighan 和 Dennis M. Ritchie(合称 K&R)合著了影响深远的名著 *The C Programming Language* 一书。这本书中介绍的 C 语言成为广泛使用的 C 语言版本的基础,它被称为标准 C,又称为 K&R 标准。1983 年,美国国家标准化协会根据 C 语言的各种版本对 C 的发展和扩充,制定了一个新的标准,称为 ANSI C。ANSI C 是 C 语言的标准,任何 C 语言的编译器都在 ANSI C 的基础上扩充。

1990 年,国际标准化组织 ISO(International Organization for Standardization)接受 87 ANSI C 为 ISO C 的标准(ISO 9899:1990)。目前流行的 C 编译系统都是以它为基础的,但不同版本的 C 编译系统所实现的语言功能和语法规则略有差别。本书的叙述基本上以 ANSI C 为基础。

在 ANSI 标准化后,C 语言的标准在相当一段时间内都保持不变,尽管 C++ 继续在改进(实际上,Normative Amendment 1 在 1995 年已经开发了一个新的 C 语言版本。但是这个版本很少为人所知)。C 语言标准在 20 世纪 90 年代才真正经历了改进,这就是 ISO 9899:1999(1999 年出版)。这个版本就是通常提及的 C99,它于 2000 年 3 月被 ANSI 采用。