

21 世纪高等学校计算机教育实用规划教材

C语言程序设计基础教程

纪 陈
钢 媛
金 张
艳 建
勋
等编著



清华大学出版社

21 世纪高等学校计算机教育实用规划教材

C语言程序设计基础教程

陈媛 张建勋 纪钢 金艳 等编著

清华大学出版社
北京

内 容 简 介

本书共分 11 章,内容包括计算机编程及 C 语言概述,基本数据类型、运算符与表达式,顺序结构程序设计,选择结构程序设计,循环结构程序设计,数组,函数,指针,编译预处理,复杂数据类型,文件等。

本书注重教材的可读性和适用性,每章开头均介绍本章内容与前后章节知识点的关系;在“常见编程错误和编译器错误”中给出了该章知识点在编程中可能出现的语法和语义错误;在“小结”中对要求掌握的知识点进行了概要说明;书中附有大量的图表、程序,使读者能正确、直观地理解问题;样例由浅入深,强化知识点、算法、编程方法与技巧,并给出了详细的解释;为了适合计算机等级考试,在内容安排上完全符合计算机等级考试大纲要求;另外,本书还配套提供题型丰富的习题。

本书可作为高等学校本科、高职高专学生“C 程序设计”课程的教学用书,也可作为全国计算机等级考试及各类短训班的培训教材。

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 语言程序设计基础教程/陈媛等编著.--北京:清华大学出版社,2011.6

(21 世纪高等学校计算机教育实用规划教材)

ISBN 978-7-302-24692-3

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

中国版本图书馆 CIP 数据核字(2011)第 018589 号

责任编辑:闫红梅

责任校对:焦丽丽

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京季蜂印刷有限公司

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:185×260 印 张:21 字 数:511 千字

版 次:2011 年 6 月第 1 版 印 次:2011 年 6 月第 1 次印刷

印 数:1~4000

定 价:29.50 元

产品编号:038964-01

出版说明

随着我国高等教育规模的扩大以及产业结构调整的不断深入,社会对高层次应用型人才的需求将更加迫切。各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,合理调整和配置教育资源,在改革和改造传统学科专业的基础上,加强工程型和应用型学科专业建设,积极设置主要面向地方支柱产业、高新技术产业、服务业的工程型和应用型学科专业,积极为地方经济建设输送各类应用型人才。各高校加大了使用信息科学等现代科学技术提升、改造传统学科专业的力度,从而实现传统学科专业向工程型和应用型学科专业的发展与转变。在发挥传统学科专业师资力量强、办学经验丰富、教学资源充裕等优势的同时,不断更新教学内容、改革课程体系,使工程型和应用型学科专业教育与经济建设相适应。计算机课程教学在从传统学科向工程型和应用型学科转变中起着至关重要的作用,工程型和应用型学科专业中的计算机课程设置、内容体系和教学手段及方法等也具有不同于传统学科的鲜明特点。

为了配合高校工程型和应用型学科专业的建设和发展,急需出版一批内容新、体系新、方法新、手段新的高水平计算机课程教材。目前,工程型和应用型学科专业计算机课程教材的建设工作仍滞后于教学改革的实践,如现有的计算机教材中有不少内容陈旧(依然用传统专业计算机教材代替工程型和应用型学科专业教材),重理论、轻实践,不能满足新的教学计划、课程设置的需要;一些课程的教材可供选择的品种太少;一些基础课的教材虽然品种较多,但低水平重复严重;有些教材内容庞杂,书越编越厚;专业课教材、教学辅助教材及教学参考书短缺,等等,都不利于学生能力的提高和素质的培养。为此,在教育部相关教学指导委员会专家的指导和帮助下,清华大学出版社组织出版本系列教材,以满足工程型和应用型学科专业计算机课程教学的需要。本系列教材在规划过程中体现了如下一些基本原则和特点。

(1) 面向工程型与应用型学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映基本理论和原理的综合应用,强调实践和应用环节。

(2) 反映教学需要,促进教学发展。教材规划以新的工程型和应用型专业目录为依据。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材建设仍然把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现工程型和应用型专业教学内容和课程体系改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材要配套,同一门课程可以有多本具有不同内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材,教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机教育实用规划教材编委会

联系人:魏江江 weijj@tup.tsinghua.edu.cn

前 言

C 语言是当今最流行的程序设计语言之一,它功能丰富,表达力强,使用灵活方便,应用面广,既具有高级语言的特点,又具有低级语言的特点,适合作为系统描述语言,既可以用来编写系统软件,也可以用来编写应用软件。C 语言诞生后,许多原来用汇编语言编写的软件,现在可以用 C 语言编写(例如,著名的 UNIX 操作系统就是用 C 语言编写的),而学习和使用 C 语言要比学习和使用汇编语言容易得多。因此,高校的高级语言程序设计课程,主要以 C 语言作为程序设计语言。

本书注重教材的可读性和适用性,每章开头均介绍本章内容与前后章节知识点的关系、在“常见编程错误和编译器错误”一节给出了在编程中可能出现的语法和语义错误;在“小结”中对要求掌握的知识点进行了概要说明;书中附有大量的图表、程序,使读者能正确、直观地理解问题;样例由浅入深,强化知识点、算法、编程方法与技巧,并给出了详细的解释;为了适合计算机等级考试,在内容安排上完全符合计算机等级考试大纲要求;另外,本书各章还配套提供题型丰富的习题。

本书内容和结构体现了教学改革成果。全书由重庆理工大学“C 语言程序设计”精品课程建设小组的教师集体编写完成。作者都是长期在高校从事“C 语言程序设计”教学的一线教师,有丰富的教学经验和软件开发能力。作者根据多年的教学经验和多项教研课题的研究成果,构建了一个程序设计概念建立和编程思想培养的框架体系,总结提炼了学习本课程的重难点和解决方法,大部分样例都经过整理和组织,以便更好地理解掌握。

本书第 1、2 章由张建勋教授编写,第 3、4 章由金艳编写,第 5 章由李娅编写,第 6 章由洪雄编写,第 7 章由陈媛教授编写,第 8 章由纪钢教授编写,第 9、10 章由陈渝副教授编写,第 11 章由杨继森博士编写。全书由陈媛教授统稿。

本书的读者只要求具有一般计算机基础知识,不要求具有程序设计和算法基础。本书可作为高等学校本科、高职高专学生“C 程序设计”课程的教学用书,也可作为全国计算机等级考试及各类短训班的培训教材。

为了方便教学,本书提供教师用电子教案和精品课程“C 语言程序设计”教学网站材料,使用本书的院校可通过邮件“cy@cqut.edu.cn”向作者索取。

由于我们的水平有限,本书可能会有不尽如人意之处,错漏之处在所难免,敬请读者批评指正,以便我们及时修改。

编 者

2010 年 10 月

目 录

第 1 章 计算机编程及 C 语言概述	1
1.1 程序的基本概念	1
1.1.1 程序设计语言	1
1.1.2 语言实现	1
1.1.3 开发环境	2
1.1.4 程序的工作原理	2
1.2 C 语言的发展及其特点	3
1.2.1 C 语言的发展	3
1.2.2 C 语言的特点	3
1.3 C 语言编程介绍	4
1.3.1 C 语言源程序的基本结构	4
1.3.2 C 语言的字符集	8
1.3.3 C 语言的标识符	8
1.3.4 C 语言的语句	9
1.3.5 编程风格	9
1.4 运行 C 语言程序的步骤与环境	10
1.4.1 运行 C 程序的流程	10
1.4.2 Visual C++6.0 集成开发环境简介	11
1.4.3 Visual C++6.0 下调试运行程序的操作步骤	14
1.5 常见编程错误和编译器错误	16
1.5.1 编程错误	16
1.5.2 编译器错误	17
小结	17
习题	18
第 2 章 基本数据类型、运算符与表达式	21
2.1 C 语言的数据类型	21
2.2 常量、变量	22
2.2.1 常量及符号常量	22
2.2.2 变量及定义	23

2.3	基本数据类型与表示范围	25
2.3.1	整型数据	26
2.3.2	实型数据	29
2.3.3	字符型数据和字符串常量	31
2.3.4	各类数值型数据间的混合运算	36
2.4	C 语言的运算符与表达式	37
2.4.1	C 语言运算符与表达式简介	37
2.4.2	算术运算符和算术表达式	39
2.4.3	赋值运算符和赋值表达式	41
2.4.4	强制类型转换符	46
2.4.5	自增自减运算符	47
2.4.6	位运算符和位运算表达式	49
2.4.7	逗号运算符和逗号表达式	51
2.4.8	指针运算符、sizeof 运算符	53
2.5	常见编程错误和编译器错误	53
2.5.1	编程错误	53
2.5.2	编译器错误	54
小结		54
习题		54
第 3 章	顺序结构程序设计	59
3.1	结构化程序设计	59
3.1.1	结构化程序设计概述	59
3.1.2	结构化程序设计的基本结构及其特点	59
3.2	算法	60
3.2.1	算法的基本概念	60
3.2.2	算法的特性	61
3.2.3	算法的流程图表示法	61
3.2.4	基本算法	64
3.3	C 语句概述	64
3.4	输入输出介绍	66
3.4.1	格式化输出函数 printf	67
3.4.2	格式化输入函数 scanf	70
3.4.3	字符输出函数 putchar	72
3.4.4	字符输入函数 getchar	72
3.5	顺序结构程序设计举例	74
3.6	常见编程错误和编译器错误	75
3.6.1	编程错误	76
3.6.2	编译器错误	76

小结	76
习题	76
第 4 章 选择结构程序设计	80
4.1 关系运算符、逻辑运算符、条件运算符	80
4.1.1 关系运算符和关系表达式	80
4.1.2 逻辑运算符和逻辑表达式	81
4.1.3 条件运算符和条件表达式	82
4.2 if 语句	83
4.2.1 if 语句的一般形式	83
4.2.2 if 语句的嵌套形式	84
4.3 switch 语句	86
4.3.1 switch 语句的一般形式	86
4.3.2 switch 语句的嵌套形式	88
4.4 选择结构程序设计举例	88
4.5 常见编程错误和编译器错误	92
4.5.1 编程错误	92
4.5.2 编译器错误	92
小结	93
习题	93
第 5 章 循环结构程序设计	96
5.1 基本循环结构	96
5.2 while 语句	96
5.3 do...while 语句	99
5.4 for 语句	102
5.5 goto、break、continue 语句	104
5.5.1 goto 语句	104
5.5.2 break 语句	105
5.5.3 continue 语句	106
5.6 循环的嵌套	107
5.7 几种循环的比较	107
5.8 循环结构程序设计举例	108
5.9 常见编程错误和编译器错误	110
5.9.1 编程错误	110
5.9.2 编译器错误	111
小结	112
习题	112

第 6 章 数组	122
6.1 一维数组	123
6.1.1 一维数组的定义	123
6.1.2 一维数组的引用	124
6.1.3 一维数组元素的初始化	126
6.1.4 一维数组程序举例	127
6.2 二维数组	132
6.2.1 二维数组的定义	132
6.2.2 二维数组的引用	133
6.2.3 二维数组元素的初始化	134
6.2.4 二维数组程序举例	135
6.3 字符数组	139
6.3.1 字符数组的定义	139
6.3.2 字符数组的初始化	140
6.3.3 字符数组的输入输出	141
6.3.4 字符串基础知识	142
6.3.5 字符串的输入输出	145
6.3.6 字符串常用操作函数	148
6.3.7 字符数组应用举例	150
6.4 常见编程错误和编译器错误	153
6.4.1 编程错误	153
6.4.2 编译器错误	154
小结	155
习题	155
第 7 章 函数	158
7.1 函数的概念	158
7.1.1 标准库函数的特点	158
7.1.2 用户自定义函数的特点	159
7.1.3 函数运行中的参数传递及返回值	160
7.2 函数的定义	160
7.2.1 函数定义的一般形式	160
7.2.2 函数参数的传递方式	162
7.2.3 函数的返回值	164
7.3 函数的调用	165
7.3.1 函数调用的形式	165
7.3.2 函数调用的方式	165
7.3.3 函数声明	166

7.4	数组在函数参数传递中的应用	168
7.4.1	数组元素作函数参数	168
7.4.2	数组名作函数参数	168
7.5	函数的嵌套调用与函数的递归调用	171
7.5.1	函数的嵌套调用	171
7.5.2	函数的递归调用	172
7.6	主函数 main 带参数	175
7.6.1	main 函数的带参数格式定义	176
7.6.2	main 函数的调用	176
7.7	函数的作用域	177
7.7.1	局部变量	177
7.7.2	全局变量	179
7.8	变量的存储类别	181
7.8.1	自动变量	182
7.8.2	静态变量	182
7.8.3	寄存器变量	183
7.8.4	外部变量	184
7.9	文件程序	185
7.9.1	内部函数	185
7.9.2	外部函数	186
7.10	常见编程错误和编译器错误	187
7.10.1	编程错误	187
7.10.2	编译器错误	187
	小结	188
	习题	188

第8章 指针

8.1	指针与指针变量的概念	193
8.1.1	指针的概念	193
8.1.2	指针变量的概念	193
8.2	指针变量的定义和引用	195
8.2.1	指针变量的定义	195
8.2.2	指针变量的引用	196
8.3	指针运算	198
8.3.1	单个指针变量的运算	198
8.3.2	两个指针变量之间的运算	199
8.3.3	空指针的操作	199
8.4	指针与数组	200
8.4.1	指针变量与数组的关系	200

8.4.2	数组中的指针操作	200
8.4.3	指针变量对一维数组的操作方式	201
8.4.4	指针变量在多维数组中的应用	205
8.4.5	指向由 m 个元素构成的一维数组的指针变量	208
8.5	指针与字符串	211
8.5.1	字符串操作的特点及字符指针变量的引入	211
8.5.2	指向字符串的指针变量	211
8.6	指针数组与多级指针	214
8.6.1	指针数组的定义	214
8.6.2	指针数组的使用	215
8.6.3	多级指针	216
8.7	指针变量与函数	218
8.7.1	函数的操作方式与指针变量	218
8.7.2	指针型函数的定义与使用	218
8.7.3	函数指针的定义与使用	220
8.7.4	与指针有关的函数参数传递方式	222
8.8	指针与动态内存分配	226
8.8.1	动态存储的概念	226
8.8.2	C 语言的动态存储管理方式	227
8.9	常见编程错误和编译器错误	229
8.9.1	编程错误	229
8.9.2	编译器错误	230
小结		230
习题		230
第 9 章	编译预处理	237
9.1	宏定义	237
9.1.1	不带参数的宏定义	237
9.1.2	带参数的宏定义	239
9.2	文件包含处理	241
9.3	条件编译	243
9.4	常见编程错误和编译器错误	244
9.4.1	编程错误	244
9.4.2	编译器错误	244
小结		245
习题		245
第 10 章	复杂数据类型	248
10.1	复杂数据类型概述	248

10.2	结构体	249
10.2.1	结构体类型的概念及定义	249
10.2.2	结构体变量的概念及定义	250
10.2.3	结构体变量的初始化和引用	252
10.3	结构体与数组、函数、指针	253
10.3.1	结构体与数组	253
10.3.2	结构体与函数	257
10.3.3	结构体与指针	258
10.4	联合体	262
10.4.1	联合体类型的概念及定义	262
10.4.2	联合体变量的概念及定义	262
10.4.3	联合体变量的初始化和引用	263
10.5	线性链表	264
10.5.1	链表的概念	265
10.5.2	线性链表的基本操作	267
10.6	自定义类型	270
10.7	枚举类型	271
10.7.1	枚举类型的定义	271
10.7.2	枚举变量的定义	272
10.7.3	枚举变量的初始化和引用	273
10.7.4	枚举类型与 sizeof 运算符	276
10.8	复杂数据类型应用综合举例	277
10.9	常见编程错误和编译器错误	280
10.9.1	编程错误	280
10.9.2	编译器错误	280
小结		281
习题		282

第 11 章 文件 285

11.1	C 文件概述	285
11.1.1	文件的基本概念	285
11.1.2	文件的类别	285
11.1.3	流与缓冲文件系统	286
11.2	文件类型指针	287
11.2.1	文件结构体类型	287
11.2.2	文件指针	287
11.2.3	相关说明	287
11.3	文件操作概述	288
11.3.1	文件处理的一般过程	288

11.3.2	文件处理的一般算法	289
11.4	文件的打开和关闭	289
11.4.1	文件的打开(fopen 函数)	289
11.4.2	文件的关闭(fclose 函数)	290
11.5	文件的读/写	291
11.5.1	单个字符读/写函数	291
11.5.2	字符串读/写函数	293
11.5.3	数据块及格式化读/写函数	294
11.6	文件的随机读写	297
11.7	常见编程错误和编译器错误	299
11.7.1	编程错误	299
11.7.2	编译器错误	299
小结		299
习题		300
附录 1	C 语言的字符集——ASCII 字符表	302
附录 2	C 语言的库函数	303
附录 3	常见错误提示的中文解释	313
参考文献		320

第 1 章

计算机编程及 C 语言概述

计算机系统由硬件和软件组成,计算机硬件由控制器、运算器、存储器、输入和输出设备五大部件组成,软件是程序及相关文档的集合,包括系统软件和应用软件。

为了使计算机能够按照人们的意志进行工作,必须根据问题的要求,编写相应的程序。计算机程序是一组计算机能识别和执行的指令和数据的结构化组合,每条指令使计算机执行特定的操作。计算机程序设计是用计算机能够响应、其他程序员能够理解的语言编写这些指令的过程。程序的表达手段就是程序设计语言,它由能够构造程序的指令集组成。

1.1 程序的基本概念

1.1.1 程序设计语言

程序设计语言也叫计算机语言,是指根据预先制定的规则(语法)而写出的语句集合或指令集合。可用的计算机程序设计语言以各种形式和类型出现,一般分为机器语言、汇编语言和高级语言,C 语言就是一种高级程序设计语言。

机器语言是由二进制指令构成的编程语言,它与具体计算机硬件有关。

汇编语言是以机器语言为基础引入助记符(操作代码、变量名)的编程语言,它也与具体计算机硬件有关。

高级语言结合了数学表达式和英语符号,因而接近自然语言,是用于编写与 CPU 类型无关程序的编程语言。但高级语言编写的程序不能被计算机直接识别,它必须通过编译、连接等过程才能被执行。

程序设计语言实际上就是一套规范的集合,一个程序只有严格按照语言规定的语法编写,才能保证编写的程序在计算机中能正确地执行,同时也便于阅读和理解。学习一门程序设计语言,关键是要学习使用语言来解决实际问题的方法,如果掌握的语法和程序设计方法能够高效解决实际工作中的各种问题,那就表明已经掌握了这门语言。

1.1.2 语言实现

语言实现是具体地实现一种语言的各种特征并支持特定编程模式的技术和工具。一般来讲,编程语言的实现就是编译器(compiler)和连接器(linker)(编译-连接模式)或者解释器(interpreter)(解释模式)的实现,即用来分析源代码并生成最终的可执行机器指令集合的技术和工具,以及一套标准库实现。语言最终要表现为某个具体的实现版本,Turbo C、Borland C、Microsoft C、GNU C 等都是 C 语言不同的实现版本。

1.1.3 开发环境

编程就像是在写文章。写文章要求你首先会一门语言(如同程序设计语言),要有内容(如同代码),还要有各种工具,如桌子、笔墨纸砚等(如同你的工作平台)。文章刚写出来时不能马上交给读者看,还需要排版、校对、印刷和发行(如同代码调试、编译连接和发布)。

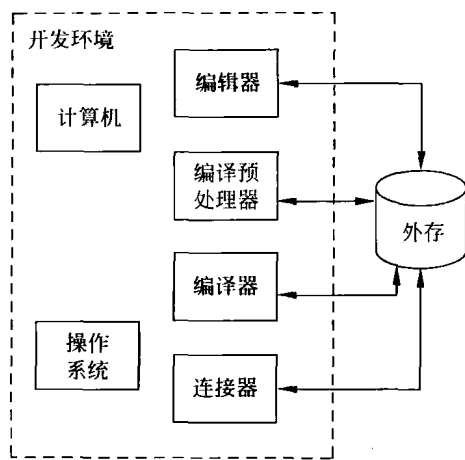


图 1.1 C 开发环境

开发环境泛指支持软件开发的一切工具,例如操作系统、代码编辑器、编译器、连接器、调试器等。典型的 C 开发环境如图 1.1 所示。编辑器是帮助程序员把源程序录入到计算机中的软件;编译预处理处理器是处理 C 语言源程序中的预处理命令的软件;编译器是将高级语言编写的源程序翻译成机器语言的软件;连接器是把多个目标程序和函数连接成一个二进制的可执行程序的软件。

集成开发环境(IDE)就是把编辑器、编译器、连接器、调试器等各种工具集成到一个工作空间中,以方便程序员开发程序。

1.1.4 程序的工作原理

程序既可以指开发完成的可执行文件及其相关文件和数据,也可以指正处于开发阶段的源代码及其相关文件和数据(IDE 称之为程序工程)。把一个程序工程转变为一个可执行程序要经历编译、连接等过程。

程序在执行前必须先由辅助存储器中传输到内存中,才能被 CPU 访问执行,这个过程由加载器完成,加载器还引导 CPU 从第一条指令开始执行。内存是存储单元的有序序列,每个内存单元都有一个唯一的地址,内存单元的大小一般为一个字节(byte),一个字节由八个位(bit)组成。存储在内存单元中的数据称为内存单元的内容,内存单元的内容永不为空,但其初始值可能对程序毫无意义。内存是易失性存储介质,无论其中存储的是什么内容,一旦电源切断就会丢失。

现在的计算机仍然遵循冯·诺依曼的“存储程序控制”原理。本质上,任何一个程序都是由待处理的数据和一系列处理它们的指令(操作)组成的,这些指令通过内存地址来访问待处理的数据。典型的 C 运行环境如图 1.2 所示。

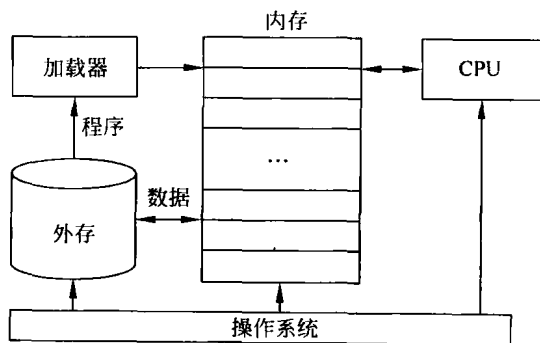


图 1.2 C 程序运行环境

1.2 C语言的发展及其特点

1.2.1 C语言的发展

C语言是当今最流行的程序设计语言之一,既可以用来编写系统软件,也可以用来编写应用软件。它由美国贝尔实验室的 D. M. Ritchie 于 1972 年推出。1978 年后,C语言已先后被移植到大、中、小及微型机上。

- C语言的原型是 ALGOL 60 语言(也称为 A 语言)。
- 1963 年,剑桥大学将 ALGOL 60 语言发展成为 CPL(Combined Programming Language)语言。
- 1967 年,剑桥大学的 Martin Richards 对 CPL 语言进行了简化,于是产生了 BCPL 语言。
- 1970 年,美国贝尔实验室的 Ken Thompson 将 BCPL 进行修改,产生了 B 语言,并用 B 语言写了第一个 UNIX 操作系统。
- 1972 年,美国贝尔实验室的 D. 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。
- 1977 年,D. M. Ritchie 为了使 UNIX 操作系统推广,发表了不依赖于具体机器系统的 C 语言编译文本——可移植的 C 语言编译程序。
- 1978 年,美国电话电报公司(AT&T)贝尔实验室正式发表了 C 语言。同时由 B. W. Kernighan 和 D. M. Ritchie 合著了著名的 *The C Programming Language* 一书。通常简称为 K&R,也有人称之为 K&R 标准。
- 1983 年,由美国国家标准化协会(American National Standards Institute, ANSI)在 K&R 标准基础上制定了一个新的标准 ANSI C。

目前最流行的 C 语言版本有 Microsoft C(或称 MS C)、Borland Turbo C(或称 Turbo C)、AT&T C 等,它们不仅实现了 ANSI C 标准,而且在此基础上各自作了一些扩充,使之更加方便、完美,不同版本的 C 编译系统所实现的语言功能和语法规则又略有差别。

目前常用的 C 语言 IDE(集成开发环境)有 Microsoft Visual C++、Dev-C++、Code::Blocks、Borland C++、Watcom C++、Borland C++ Builder、GNU DJGPP C++、Lccwin32 C Compiler 3.1、High C、Turbo C、C-Free、win-tc 等等。对于初学者,Microsoft Visual C++ 是一个比较好的开发环境,界面友好,功能强大,程序调试也很方便。

本书所有示例均在 Visual C++ 6.0 系统下调试通过。

1.2.2 C语言的特点

C语言的主要特点如下:

(1) C语言既具有高级语言的功能,又具有汇编语言的许多功能,C语言实际上属中级语言。它把高级语言的基本结构和语句与低级语言的实用性结合起来。C语言允许直接访问位、字节和地址,能实现汇编语言的大部分功能,可以直接对硬件进行操作。对于编写需要对硬件进行操作的应用程序,明显优于其他解释型高级语言。