

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

21世纪普通高校计算机公共课程规划教材

C 语言程序设计

田丽华 主编

岳俊华 李卓 吕鑫 编著

清华大学出版社



21 世纪普通高校计算机公共课程规划教材

C 语言程序设计

田丽华 主编

岳俊华 李卓 吕鑫 编著

清华大学出版社
北京

内 容 简 介

本书以数据类型为主线,以函数为核心,介绍了C程序的逻辑结构(顺序、选择、循环)、C语言的数据结构(数组、指针、链表等)和文件操作。章节衔接性好,目的明确,将具有代表性和实用性的案例不断扩展和完善,每章所学习的内容是在前面所学知识的基础上进行扩展,体现了结构化程序设计的模块化原则。

本书文字流畅,通俗易懂,深入浅出,实用性强。本书可作为计算机及其他理工科专业的教材以及喜欢编程的爱好者的参考书。与本书配套的《C语言程序设计上机指导与习题解答》(田丽华主编,北京邮电大学出版社出版)提供大部分课后习题解答。

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/田丽华主编;岳俊华,李卓,吕鑫编著. —北京:清华大学出版社,2010.3
(21世纪普通高校计算机公共课程规划教材)

ISBN 978-7-302-22102-9

I. ①C… II. ①田… ②岳… ③李… ④吕… 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(2010)第029118号

责任编辑:付弘宇 王冰飞

责任校对:时翠兰

责任印制:孟凡玉

出版发行:清华大学出版社

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

社 总 机:010-62770175

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:北京国马印刷厂

经 销:全国新华书店

开 本:185×260 印 张:15.75 字 数:381千字

版 次:2010年3月第1版 印 次:2010年3月第1次印刷

印 数:1~3000

定 价:25.00元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系
调换。联系电话:(010)62770177 转 3103 产品编号:034859-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,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,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投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,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,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。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教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程有针对不同层次、面向不同专业的多本具有各自内容特点的教材。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,择优选编。在制定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、评审确定主题。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。建立一支高水平教材编写梯队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 世纪普通高校计算机公共课程规划教材编委会

联系人:梁颖 liangying@tup.tsinghua.edu.cn

前言

C语言作为国内外广泛使用的一种编程语言,虽经历了计算机行业的飞速发展,但一直经久不衰。由于C语言具有功能强大、目标程序效率高、可移植性好、执行速度快等特点,因此特别适合编写各种系统软件。

随着计算机应用的普及,计算机的编程能力已不再是计算机专业人员的强项,许多非计算机的理工类专业人士也广泛地应用计算机进行程序设计,进行专业相关的工作。现在所有的在校大学生基本都学习“C程序设计”这门课程,因为涉及国家等级考试、专业课的基础课以及面向对象程序设计等。可是由于学生的计算机应用能力差异较大,对计算机的工作原理缺乏感性认识,理解起来比较抽象,因此这门课程是学生入门的一大难点,致使学生认为学习计算机编程语言就是背语法,学习起来抽象、难理解。针对以上问题,本书从计算机工作原理出发,使读者在熟悉工作原理的基础上,通过层层递进地编写、运行程序,明白计算机是如何工作的。这样,引入数据类型、变量等概念就不再突兀。

本书总结作者多年的教学经验,以数据类型为主线,以C程序的基本单位——函数为核心,将该门课程难点分散。

本书的具体写作思路是:

(1) 在深入理解计算机工作原理的基础上,引出内存的重要性。CPU只能与内存交换数据,不能直接与外存交换数据。那么数据到底占用多大的内存空间取决于占用该内存空间的数据量的类型,进而引出数据类型的概念,使学生学习数据类型有针对性,不再抽象。

(2) 学习了基本数据类型,会定义某种类型的变量及变量在内存中的存储形式,进而利用三种结构化程序控制流程,实现简单变量的操作。程序对变量进行控制,计算机对相应变量的内存空间进行操作,真正达到人机共同完成一个任务。有了对简单程序的应用能力,进而作为一个功能模块讲解函数。此时函数只能处理简单数据类型的变量。

(3) 接下来,在简单变量的基础上引入数组的概念。数组在数据结构上是顺序存储,数组里的每一个元素与相同数据类型的简单变量具有相同的操作功能,只要关心数组元素的下标即可。利用所学的函数基本理论,将数组元素和数组名作为函数的参数,对发生函数调用时的情形进行扩展。这样,对于函数功能的扩展就变得很容易。

(4) 指针是C语言的一大特点,使操作内存变得很灵活。指针是一个特殊的变量,存储内存地址号。根据指针所指向的内存空间的归属,决定是指向什么的指针变量,进而明确操作。通过将指针变量作为函数的参数,发生函数调用时传址和传值的区别便一目了然。

(5) 结构体变量可以表达许多与生活息息相关的数据结构,把不同数据类型的集合称做结构体类型,与简单变量的定义、引用类似。可以定义结构体类型(用户自定义类型)的变量、数组、指针变量并引用,进而引出在数据结构上的链式存储数据结构——链表。

(6) 上面对各种数据类型变量的操作主要是对内存空间进行的操作,对于存于外存(硬盘等)的数据,可以通过文件操作将外存文件与内存文件指针变量相关联,从而通过操作内存文件指针变量达到操作外存文件的目的。

本书图文并茂、思路清晰,能培养良好的编程风格,使学生知其然也知其所以然。

学生以数据类型为线索,通过标识符的不同形式即“`int a; int a[]; int *p; int f();`”学习各知识点。这样,在学习了基本数据类型应用的基础上,再引入结构体变量就更易理解和掌握。

本书精选大量实例且大部分有详细解释,尤其是前面部分和难理解的程序均给出详细的注释,方便读者学习。本书中所有程序均在 Visual C++ 6.0 下调试通过。通过详细的图解使读者深刻体会变量在机器中的存储形式,能够更好地操作变量。

本书的另一个特点就是作者根据多年教学经验给出了一些基本的编程风格和技巧,帮助学生尽快养成良好的编程风格,积累编程经验和能力。这对初学者非常有好处。

与本书配套的《C 语言程序设计上机指导与习题解答》(田丽华主编)已由北京邮电大学出版社出版,提供大部分课后习题解答和练习题。

全书由田丽华负责统稿,第 1、2、3、4、5、6、8、10 章由田丽华编写,第 9 章由岳俊华编写,第 12 章由李卓编写,第 7、11 章和附录由吕鑫编写。

在本书的写作过程中,侯殿有教授以及其他同事、家人和朋友给予我很大的支持,在此向他们表示感谢。同时要特别感谢我的好友岳俊华(吉林建筑工程学院)和李卓(北华大学),他们不但给予我支持和鼓励,并且提出了许多宝贵意见。

由于作者水平有限,时间仓促,书中难免会有错误,恳请读者指正。作者邮箱:lihua_tian18@sina.com。

本书的配套课件等电子资料可以从清华大学出版社网站 <http://www.tup.tsinghua.edu.cn> 下载,有问题请发邮件到 fuhy@tup.tsinghua.edu.cn。

田丽华

2009 年 11 月

目 录

第 1 章 C 语言概述	1
1.1 程序设计语言的发展及其特点	1
1.2 C 语言的发展过程及其特点	2
1.2.1 C 语言的发展过程	2
1.2.2 当代最优秀的程序设计语言	2
1.2.3 C 语言的版本	2
1.2.4 C 语言的特点	3
1.2.5 C 语言的应用	3
1.3 认识简单的 C 语言程序	4
1.4 Visual C++ 6.0 集成开发环境的使用	8
1.4.1 Visual C++ 6.0 简介和启动	8
1.4.2 利用 Visual C++ 6.0 集成开发环境建立工程	9
1.5 C 程序开发过程	13
本章小结	15
习题 1	16
第 2 章 数据类型、运算符与表达式	17
2.1 计算机的工作原理	17
2.2 C 语言的数据类型	18
2.3 常量、变量和标识符	19
2.3.1 标识符	20
2.3.2 常量和符号常量	20
2.3.3 变量	21
2.4 整型数据	22
2.4.1 整型常量的表示方法	22
2.4.2 整型变量	22
2.4.3 整型常量的分类	26
2.5 实型数据	26
2.5.1 实型常量的表示方法	26
2.5.2 实型变量	27

2.5.3 实型常数的类型	29
2.6 字符型数据	29
2.6.1 字符常量	29
2.6.2 字符变量	30
2.6.3 字符数据在内存中的存储形式及使用方法	31
2.6.4 字符串常量	32
2.7 C 语言的运算符与表达式	33
2.7.1 赋值运算符、赋值表达式	33
2.7.2 不同数据类型间的赋值规则	34
2.7.3 算术运算符和算术表达式	36
2.7.4 逗号运算符和逗号表达式	39
2.7.5 sizeof 运算符和复合赋值运算符	40
本章小结	41
习题 2	42

第 3 章 C 程序设计——顺序程序设计

3.1 程序的控制结构	43
3.1.1 算法的基本概念	43
3.1.2 算法的特性	44
3.1.3 算法的表示	44
3.1.4 用 N-S 流程图表示算法	46
3.1.5 用计算机语言表示算法	46
3.1.6 结构化程序设计方法	47
3.2 C 语句概述	47
3.3 数据输入输出的概念及在 C 语言中的实现	48
3.4 字符数据的输入与输出	49
3.4.1 字符输出函数 putchar	49
3.4.2 键盘输入函数 getchar	49
3.5 格式输出函数 printf	50
3.5.1 printf 函数调用的一般形式	50
3.5.2 使用 printf 函数注意事项	52
3.6 格式输入函数 scanf	53
3.7 顺序结构程序设计举例	57
本章小结	59
习题 3	59

第 4 章 选择结构程序设计

4.1 关系运算符和表达式	61
4.1.1 关系运算符及其优先级	61

4.1.2	关系表达式	62
4.2	逻辑运算符和逻辑表达式	63
4.2.1	逻辑运算符	63
4.2.2	逻辑表达式	63
4.3	if 语句	65
4.3.1	if 语句的三种形式	65
4.3.2	if 语句的嵌套	69
4.3.3	条件运算符和条件表达式	70
4.4	switch 语句	71
4.5	程序举例	75
	本章小结	76
	习题 4	77
第 5 章	循环控制	78
5.1	概述	78
5.2	goto 语句以及用 goto 语句构成循环	78
5.3	while 语句	79
5.4	do-while 语句	80
5.5	for 语句	82
5.6	循环的嵌套	84
5.7	几种循环的比较	85
5.8	break 和 continue 语句	85
5.8.1	break 语句	85
5.8.2	continue 语句	86
5.9	程序举例	87
	本章小结	89
	习题 5	89
第 6 章	函数	90
6.1	函数概述	90
6.2	函数的定义与调用	91
6.2.1	无参函数的定义形式	91
6.2.2	有参函数定义的一般形式	93
6.3	函数的返回值	95
6.4	函数参数的传递方式	96
6.5	函数的调用	98
6.5.1	函数调用的方式	98
6.5.2	函数的嵌套调用	98
6.5.3	函数的递归调用	99

6.6 变量的作用域和生存期	103
6.6.1 局部变量的作用域和生存期	103
6.6.2 全局变量	105
6.7 变量的存储类别	107
6.7.1 动态存储方式与静态存储方式	107
6.7.2 auto 变量	108
6.7.3 用 static 声明局部变量	108
6.7.4 register 变量	109
本章小结	110
习题 6	110

第 7 章 预处理命令	111
7.1 概述	111
7.2 宏定义	112
7.2.1 无参宏定义	114
7.2.2 带参宏定义	117
7.3 文件包含	118
7.4 * 条件编译	120
本章小结	121
习题 7	121

第 8 章 数组	122
8.1 一维数组	122
8.1.1 一维数组的定义方式	124
8.1.2 一维数组在内存中的存放形式	124
8.1.3 一维数组元素的引用	125
8.1.4 一维数组的初始化	126
8.1.5 一维数组程序举例	128
8.2 二维数组的定义和引用	128
8.2.1 二维数组的定义	129
8.2.2 二维数组在内存中的存放形式	129
8.2.3 二维数组元素的引用	131
8.2.4 二维数组的初始化	132
8.2.5 二维数组程序举例	133
8.3 字符数组和字符串	133
8.3.1 字符数组的定义	133
8.3.2 字符数组在内存中的存放形式	134
8.3.3 字符数组的初始化	134
8.3.4 字符数组的应用	134

8.3.5	字符串和字符串结束标志	134
8.3.6	字符数组的输入输出	135
8.3.7	字符串处理函数	137
8.4	数组名或数组元素作为函数的参数	139
8.4.1	数组元素作为函数的参数	139
8.4.2	数组名作为函数的参数	141
8.5	程序举例	142
	本章小结	146
	习题 8	146
第 9 章	指针	148
9.1	地址和指针的基本概念	148
9.2	变量的指针和指向变量的指针变量	149
9.2.1	定义一个指针变量	149
9.2.2	指针变量在内存中的表示	150
9.2.3	指针变量的引用	150
9.2.4	指针变量作为函数参数	153
9.3	数组的指针和指向数组的指针变量	155
9.3.1	指向数组元素的指针	155
9.3.2	通过指针引用数组元素	156
9.3.3	数组名作为函数参数	159
9.3.4	* 指向多维数组的指针和指针变量	163
9.4	字符串的指针和指向字符串的指针变量	166
9.4.1	字符串的表示形式	166
9.4.2	使用字符串指针变量与字符数组的区别	169
9.5	* 函数指针变量	170
9.6	* 指针型函数	170
9.7	指针数组和指向指针的指针	171
9.7.1	* 指针数组的概念	171
9.7.2	指向指针的指针	172
9.7.3	main 函数的参数	174
9.8	有关指针的数据类型和指针运算的小结	176
9.8.1	有关指针的数据类型的小结	176
9.8.2	指针运算的小结	176
9.8.3	void 指针类型	177
	本章小结	177
	习题 9	177
第 10 章	结构体与共用体	178
10.1	定义一个结构体的一般形式	178

10.2	结构体变量的说明	179
10.3	结构体成员变量的表示方法	182
10.4	结构体变量的赋值	182
10.5	结构体变量的初始化	183
10.6	结构体数组	184
10.6.1	结构体数组的定义	184
10.6.2	结构体数组的初始化	185
10.6.3	结构体数组元素的引用	185
10.6.4	结构体数组的应用	185
10.7	结构体指针变量的说明和使用	187
10.7.1	指向结构体变量的指针变量	187
10.7.2	指向结构体数组的指针	189
10.7.3	结构体变量和指针变量作为函数参数	190
10.8	动态存储分配	191
10.9	链表的概念	193
10.9.1	创建动态链表	194
10.9.2	链表结点的查找与输出	195
10.9.3	链表的插入操作	196
10.9.4	删除结点操作	198
10.10	联合体	199
10.10.1	联合体类型的定义	199
10.10.2	联合体变量的定义和引用	200
10.10.3	联合体变量的赋值	200
10.11	枚举类型	202
10.11.1	枚举类型的定义和枚举变量的说明	202
10.11.2	枚举类型变量的赋值和使用	202
	本章小结	203
	习题 10	204
第 11 章	位运算	205
11.1	按位与运算	205
11.2	按位或运算	207
11.3	按位异或运算	207
11.4	取反运算	209
11.5	左移运算	209
11.6	右移运算	210
	本章小结	210
	习题 11	210

第 12 章 文件	211
12.1 文件概述	211
12.2 文件指针	212
12.3 文件的打开与关闭	213
12.3.1 文件打开函数 fopen	213
12.3.2 文件关闭函数 fclose	214
12.4 文件的读写	215
12.4.1 字符读写函数 fgetc 和 fputc	215
12.4.2 字符串读写函数 fgets 和 fputs	218
12.4.3 数据块读写函数 fread 和 fwrite	219
12.4.4 格式化读写函数 fscanf 和 fprintf	221
12.5 文件的随机读写	222
12.5.1 文件定位	222
12.5.2 文件的随机读写	223
12.6 文件检测函数	224
12.6.1 文件结束检测函数 feof	224
12.6.2 读写文件出错检测函数 ferror	224
本章小结	224
习题 12	224
附录 A 关键字	225
附录 B ASCII 码表	227
附录 C 运算符	229
附录 D 常用函数	231
参考文献	235

第 1 章

C 语言概述

本章导读：

了解 C 语言的发展简史、C 语言的特点及其广泛应用,进而知道学习 C 语言的必要性。认识 C 语言的组成结构,本书本着难点分散的思路,各章依次介绍 C 语言的各个组成部分。C 语言是一种高级语言,需要经过相应的编译器解释成机器语言,可以利用 Visual C++ 6.0 编译器对 C 语言程序进行编辑、编译和运行。熟悉 Visual C++ 6.0 的工作环境,理解 C 语言的开发过程。

本章学习重点：

- ① 认识 C 程序的各组成部分;
- ② 熟悉 VC++ 6.0 集成开发环境;
- ③ 掌握一个 C 程序的开发过程。

1.1 程序设计语言的发展及其特点

计算机程序设计语言的发展,经历了从机器语言、汇编语言到高级语言的历程。

1. 机器语言

机器语言是计算机能唯一识别的语言。机器语言是一串串由 0 和 1 组成的指令序列。使用机器语言是十分痛苦的,特别是在程序有错需要修改时,更是如此。而且,由于每台计算机的指令系统往往各不相同,因此,在一台计算机上执行的程序,要想在另一台计算机上执行,必须另编程序,会造成重复工作。但由于使用的针对特定型号计算机的语言,故而运算效率是所有语言中最高的。机器语言是第一代计算机语言。

2. 汇编语言

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:用一些简洁的英文字母、符号串来替代一个特定指令的二进制串,比如,用“ADD”代表加法等,这样,人们根据英文意思很容易读懂并理解程序所执行的任务,使调试和维护都变得方便,这种程序设计语言就称为汇编语言,即第二代计算机语言。然而,计算机是不认识这些符号的,这就需要有一个专门的程序,专门负责将这些符号翻译成二进制的机器语言,这种翻译程序被称为汇编程序。汇编语言同样十分依赖于机器硬件,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,所以至今仍是一种强有力的软件开发工具。

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,应该设计一种这样的语言:接近于数

学语言或人的自然语言,同时又不依赖于计算机硬件,编出的程序能在所有机器上执行。经过努力,1954年,第一个完全脱离机器硬件的高级语言——FORTRAN问世了,50多年来,共有几百种高级语言出现,具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++、VC、VB、Delphi、Java等。

高级语言的发展也经历了从早期语言到结构化程序设计语言,从面向过程到面向对象程序语言的过程。相应地,软件的开发也由最初的个体手工作坊式的封闭式生产,发展为产业化、流水线式的工业化生产。

1969年,提出了结构化程序设计方法,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。

从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程序设计语言的出现。在此之前的高级语言,几乎都是面向过程的,程序的执行是流水线式的,在一个模块被执行完成前,人们不能干别的事,也无法动态地改变程序的执行方向。这不同于人们日常事务的处理方式,对人而言是希望发生一件事就处理一件事,也就是说,不能面向过程,而应是面向具体的应用功能,也就是对象(object)。其方法就是软件的集成化,如同硬件的集成电路一样,生产一些通用的、封装紧密的功能模块,称之为软件集成块,它与具体应用无关,但能相互组合,完成具体的应用功能,同时又能重复使用。对使用者来说,只关心它的接口(输入量、输出量)及能实现的功能,至于如何实现的,那是它内部的事,使用者完全不用关心,C++、VB、Delphi就是典型代表。

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。

1.2 C语言的发展过程及其特点

1.2.1 C语言的发展过程

C语言是在20世纪70年代初问世的。1978年由美国电话电报公司(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。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)在此基础上制定了一个C语言标准,于1983年发表。通常称之为ANSI C。

1.2.2 当代最优秀的程序设计语言

早期的C语言主要是用于UNIX系统。由于C语言的强大功能和各方面的优点逐渐为人们认识,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了广泛的使用,成为当代最优秀的程序设计语言之一。

1.2.3 C语言的版本

目前最流行、常用的C语言编译环境主要有以下几种:

- Microsoft Visual C++;
- Borland Turbo C 或称 Turbo C;
- Borland C++ Builder。

这些 C 语言版本不仅实现了 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。本书所列举的实例均是基于 ANSI C 标准利用 Visual C++ 6.0 编译器实现的。

1.2.4 C 语言的特点

① C 语言简洁、紧凑,使用方便、灵活。

ANSI C 一共只有 32 个关键字。9 种控制语句,程序书写自由,主要用小写字母表示。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。

② 运算符丰富。

共有 34 种运算符。C 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运算。

③ 数据结构类型丰富。

④ 具有结构化的控制语句。这种结构化方式可使程序层次清晰,便于使用、维护和调试。

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(bit)操作,能实现汇编语言的大部分功能,可以直接对硬件进行操作。

⑦ 生成目标代码质量高,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%。

⑧ 与汇编语言相比,用 C 语言写的程序可移植性好。

但是,C 语言对程序员要求也高,程序员用 C 写程序会感到限制少、灵活性大,功能强,但较其他高级语言在学习上要困难一些。

1.2.5 C 语言的应用

C 语言应用领域广泛,下面列举了一些 C 语言的应用。

① 许多系统软件和大型应用软件都是用 C 语言编写的,如 UNIX、Linux 等操作系统。

② 在软件需要对硬件进行操作的场合,用 C 语言明显高于其他语言。例如计算机的显卡驱动程序、打印机驱动程序等一般都是用 C 语言编写的。

③ 在图形、图像及动画处理方面,C 语言具有绝对优势,游戏软件的开发主要就是用 C 语言。

④ 在 Internet 中,通信程序的编制首选就是 C 语言。

⑤ C 语言适用于多种操作系统,像 Windows、UNIX、Linux 等绝大多数操作系统都支持 C 语言,其他高级语言未必能得到支持,所以在某个特定操作系统下运行的软件用 C 语言编写是最佳选择。