

计算机系列教材

新编C语言程序设计

宋威 吴秦 赵芝璞 编著

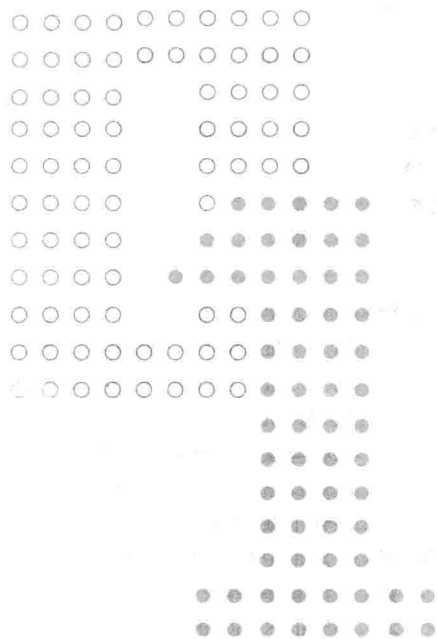


清华大学出版社

计算机系列教材

钱雪忠 宋威 吴秦 赵芝璞 编著

新编C语言程序设计



清华大学出版社

北 京

内 容 简 介

本书是在多年教学实践的基础上编写而成的,在有所创新的同时,希望能做到:概念清晰但不烦琐;例题精选又具有通用性;从实际操作出发,重视应用编程能力;让读者在把握语言知识点的同时敢于面对能力考核。

本书内容全面,重点突出,共 14 章,主要内容包括 C 语言概述、结构化程序设计与算法、数据类型及其运算、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组及其应用、函数及其应用、指针及其应用、自定义类型及其应用、文件及其应用、预处理命令、位运算、C 语言应用案例等。

本书内容通俗易懂、循序渐进、深入浅出,每章都给出了较多的例题与应用实例,各章后有适量的习题以便于读者学习、练习与巩固所学知识。

本书可作为高等院校理工科各专业“C 语言程序设计”类课程的教材,也可供参加自学考试人员、应用系统开发设计人员及其他对程序设计感兴趣的读者参阅。

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

新编 C 语言程序设计/钱雪忠,宋威,吴秦,赵芝璞编著. —北京:清华大学出版社,2014

计算机系列教材

ISBN 978-7-302-35277-8

I. ①新… II. ①钱… ②宋… ③吴… ④赵… 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(2014)第 016177 号

责任编辑:袁勤勇 李 晔

封面设计:常雪影

责任校对:白 蕾

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>, 010-62795954

印 装 者:保定市 中画美凯印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:24.25

字 数:594 千字

版 次:2014 年 2 月第 1 版

印 次:2014 年 2 月第 1 次印刷

印 数:1~2000

定 价:39.50 元

C 语言程序设计是国内外广泛使用的计算机程序设计语言,是高等院校相关专业重要的专业基础课程。C 语言具有功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好等特点。20 世纪 90 年代以来,C 语言迅速在全世界普及推广。目前,C 语言仍然是最优秀的程序设计语言之一。

本书是编者在一线教学实践的基础上,为适应当前本科教育教学改革创新的要求,更好地践行语言类课程注重实践教学与创新能力培养的需要,组织新编而成的教程。本书融合了同类其他教材的优点,并努力求变创新,新编教程力求具有以下特点:

(1) 突出 C 语言中实用的重点概念。在重点概念讲解清晰的情况下,并不求语法概念的详尽与全面,只求轻快明晰、循序渐进、通俗易懂、深入浅出。

(2) 精选例题。引入了趣味性、游戏性应用实例,注重与加强程序阅读、参考、编写和上机实践的能力,重在编程思路的培养与训练。

(3) 从实际操作出发,发现问题解决问题,举一反三,一题多解,增强实用能力。

(4) 能明晰 C 语言各语言成分的意义与价值,以“数据+算法”为核心提高编程能力。

(5) 基本知识学习、上机实验、典型习题与知识点把握等多方面相结合,使读者扎实掌握相关知识,敢于面对 C 语言能力考核。

(6) 语言编程环境以 Visual C++ 6.0 为主,同时能兼顾 Turbo C、Win-TC 等传统简易编程环境,比较不同编程环境程序运行差异可让读者更好地了解语言程序与编译器的依存关系。

全书内容共分 14 章,主要内容包括 C 语言概述、结构化程序设计与算法、数据类型及其运算、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组及其应用、函数及其应用、指针及其应用、自定义类型及其应用、文件及其应用、预处理命令、位运算、C 语言应用案例等。

本书内容充实全面,每章除基本知识外,还有章节要点、应用实例、本章小结、适量习题等,以配合对知识点的掌握。本书的配套教材《新编 C 语言程序设计实验与辅导》含有课程实验与辅导内容。课程讲授时可根据学生、专业、课时等情况对内容适当取舍。本教程对教师提供 PPT 演示稿、参考答案、例题程序等。对于第 14 章的一些应用实例,限于篇幅,书中未给出程序,可联系编者索取。

本书可作为高等院校理工科各专业“C 语言程序设计”类课程的教材,也可供参加自学考试人员、应用系统开发设计人员、工程技术人员及其他对程序设计感兴趣的读者参阅。

本书由钱雪忠主编,由江南大学、黑龙江东方学院等相关师生合作编写,参编人员有钱雪忠、宋威、吴秦、赵芝璞、吕莹楠、高婷婷、程建明、王雪茹,参与程序调试的有钱恒、任看看、马亮、施亮、邓杰、孙志鹏等。本书的编写还得到了江南大学物联网工程学院“智能系统与网络计算研究所”同仁们的大力协助与支持,使编者获益良多,谨在此表示衷心的感谢。

由于时间仓促,编者水平有限,书中难免有错误、疏漏和欠妥之处,敬请广大读者与同行专家批评指正。

联系方式(E-mail):qxzvb@163.com 或 xzqian@jiangnan.edu.cn。

编者于江南大学蠡湖校区

2013 年 10 月

第 1 章 C 语言概述 /1

1.1 程序设计语言简介 /1

1.2 C 语言发展过程 /3

1.3 C 语言的特点 /4

1.3.1 C 语言是优秀的程序语言 /4

1.3.2 C 语言与 C++ 的关系 /6

1.4 初识简单的 C 程序 /7

1.5 C 程序语法概述 /10

1.5.1 C 程序的结构特点 /10

1.5.2 C 程序的书写规则 /11

1.5.3 C 语言字符集与词汇 /11

1.6 如何运行 C 程序 /13

1.7 本章小结 /15

1.8 习题 /15

第 2 章 结构化程序设计与算法 /18

2.1 初识算法 /18

2.1.1 算法的概念 /18

2.1.2 算法举例 /19

2.1.3 算法的特征 /20

2.2 结构化程序设计 /21

2.2.1 结构化程序设计方法简介 /21

2.2.2 结构化程序设计方法的原则 /21

2.2.3 结构化程序设计的三种基本结构 /22

2.3 表示算法的多种方法 /23

2.3.1 用自然语言表示算法 /23

2.3.2 用流程图表示算法 /23

2.3.3 用 N-S 流程图表示算法 /26

2.3.4 用伪代码表示算法 /27

2.3.5 用计算机语言表示算法 /27

2.4 结构化程序设计应用举例 /28

2.5 本章小结 /30

2.6 习题 /30

第 3 章 数据类型及其运算 /32

- 3.1 数据类型 /32
- 3.2 常量与变量 /33
 - 3.2.1 常量 /33
 - 3.2.2 变量 /34
- 3.3 整型数据 /35
 - 3.3.1 整型常量 /35
 - 3.3.2 整型变量 /36
- 3.4 实型数据 /40
 - 3.4.1 实型常量 /40
 - 3.4.2 实型变量 /41
- 3.5 字符型数据 /45
 - 3.5.1 字符常量 /45
 - 3.5.2 转义字符 /45
 - 3.5.3 字符变量 /46
 - 3.5.4 字符数据的存储与使用 /46
 - 3.5.5 字符串常量 /47
- 3.6 变量赋初值 /48
- 3.7 算术运算符和表达式 /48
 - 3.7.1 C 运算符简介 /48
 - 3.7.2 算术运算符和算术表达式 /49
- 3.8 不同类型数据混合运算 /52
- 3.9 赋值运算符和表达式 /54
- 3.10 逗号运算符和逗号表达式 /56
- 3.11 应用实例 /57
- 3.12 本章小结 /60
- 3.13 习题 /60

第 4 章 顺序结构程序设计 /63

- 4.1 C 语言语句概述 /63
- 4.2 C 语言赋值语句 /64
- 4.3 数据输入输出的概念 /65
- 4.4 字符数据的输入输出 /66

4.4.1	putchar 函数	/66
4.4.2	getchar 函数	/67
4.5	格式数据的输入输出	/68
4.5.1	格式输出函数 printf	/68
4.5.2	格式输入函数 scanf	/75
4.6	应用实例	/79
4.7	本章小结	/85
4.8	习题	/85
第 5 章 选择结构程序设计 /89		
5.1	关系运算符和表达式	/89
5.1.1	关系运算符及其优先级	/89
5.1.2	关系表达式	/90
5.2	逻辑运算符和表达式	/90
5.2.1	逻辑运算符及其优先级	/90
5.2.2	逻辑运算及其取值	/91
5.2.3	逻辑表达式	/92
5.3	if 语句的用法	/93
5.3.1	if 语句的三种形式	/93
5.3.2	if 语句的嵌套	/96
5.3.3	条件运算符和条件表达式	/98
5.4	switch 语句的用法	/99
5.5	应用实例	/100
5.6	本章小结	/108
5.7	习题	/108
第 6 章 循环结构程序设计 /112		
6.1	概述	/112
6.2	goto 语句简介	/112
6.3	while 语句的用法	/113
6.4	do-while 语句的用法	/114
6.5	for 语句的用法	/116
6.6	循环的比较及其嵌套	/118

- 6.7 break 和 continue 语句用法 /119
 - 6.7.1 break 语句 /119
 - 6.7.2 continue 语句 /119
- 6.8 应用实例 /120
- 6.9 本章小结 /135
- 6.10 习题 /135

第 7 章 数组及其应用 /143

- 7.1 一维数组的定义和引用 /143
 - 7.1.1 一维数组的定义方式 /143
 - 7.1.2 一维数组元素的引用 /145
 - 7.1.3 一维数组的初始化 /146
 - 7.1.4 一维数组程序举例 /146
- 7.2 二维数组的定义和引用 /149
 - 7.2.1 二维数组的定义 /149
 - 7.2.2 二维数组元素的引用 /150
 - 7.2.3 二维数组的初始化 /151
- 7.3 字符数组 /153
 - 7.3.1 字符数组的定义 /153
 - 7.3.2 字符数组的初始化 /153
 - 7.3.3 字符数组的引用 /154
 - 7.3.4 字符串和字符串结束标志 /154
 - 7.3.5 字符数组的输入输出 /155
 - 7.3.6 字符串处理函数 /156
- 7.4 应用实例 /162
- 7.5 本章小结 /171
- 7.6 习题 /171

第 8 章 函数及其应用 /176

- 8.1 C 语言函数概述 /176
- 8.2 函数定义的一般形式 /177
- 8.3 函数的参数和函数的值 /179
 - 8.3.1 形式参数和实际参数 /179

8.3.2	函数的返回值	/180
8.4	函数的调用	/181
8.4.1	函数调用的一般形式	/181
8.4.2	函数调用的方式	/181
8.4.3	被调用函数的声明	/182
8.5	函数的嵌套调用	/183
8.6	函数的递归调用	/184
8.7	数组作为函数参数	/187
8.8	局部变量和全局变量	/191
8.8.1	局部变量	/191
8.8.2	全局变量	/193
8.9	变量的存储类别	/194
8.9.1	auto 变量	/195
8.9.2	用 static 声明局部变量	/195
8.9.3	register 变量	/197
8.9.4	用 static 声明全局变量	/197
8.10	内部函数和外部函数	/199
8.11	函数与模块化程序设计	/200
8.12	应用实例	/200
8.13	本章小结	/213
8.14	习题	/213
第 9 章	指针及其应用	/219
9.1	指针的基本概念	/219
9.2	指针变量	/220
9.2.1	定义指针变量	/220
9.2.2	指针变量的引用	/221
9.2.3	指针变量作为函数参数	/225
9.2.4	指针变量的几个问题	/228
9.3	指针与数组	/231
9.3.1	指向数组元素的指针	/231
9.3.2	通过指针引用数组	/232
9.3.3	数组指针变量作函数参数	/233

9.3.4	指向多维数组的指针	/239
9.4	指针与字符串	/241
9.4.1	字符串的表示形式	/241
9.4.2	字符(串)指针变量与字符数组	/244
9.5	指针与函数	/245
9.5.1	函数指针变量	/246
9.5.2	指针型函数	/247
9.6	多级指针	/248
9.6.1	指针数组概念	/248
9.6.2	指向指针的指针	/253
9.6.3	main 函数参数	/254
9.7	应用实例	/255
9.8	本章小结	/259
9.9	习题	/260
第 10 章	自定义类型及其应用	/266
10.1	结构体与结构体变量	/266
10.1.1	如何定义结构体	/266
10.1.2	如何说明结构体变量	/267
10.1.3	结构体变量成员的表示方法	/269
10.1.4	结构体变量的赋值	/269
10.1.5	结构体变量的初始化	/270
10.2	结构体数组与结构体指针	/270
10.2.1	结构体数组的定义与使用	/270
10.2.2	指向结构体变量的指针	/272
10.2.3	指向结构体数组的指针	/273
10.2.4	结构体指针变量作函数参数	/274
10.3	C 语言动态存储分配	/275
10.4	C 语言链表的概念	/277
10.5	共用体	/280
10.5.1	共用体类型的定义	/280
10.5.2	共用体变量的说明	/281
10.5.3	共用体变量的赋值和使用	/281

10.6	C 语言枚举类型	/282
10.6.1	枚举类型的定义及其变量说明	/283
10.6.2	枚举类型变量的赋值和使用	/283
10.7	C 语言类型定义符 typedef 详解	/285
10.8	应用实例	/286
10.9	本章小结	/296
10.10	习题	/296
第 11 章	文件及其应用	/301
11.1	C 语言文件概述	/301
11.1.1	文件的分类	/301
11.1.2	库文件	/302
11.2	文件指针	/303
11.3	文件的打开与关闭	/304
11.3.1	文件打开函数 fopen	/304
11.3.2	文件关闭函数 fclose	/306
11.4	文件的读写	/306
11.4.1	字符读写函数 fgetc 和 fputc	/306
11.4.2	字符串读写函数 fgets 和 fputs	/309
11.4.3	数据块读写函数 fread 和 fwrite	/310
11.4.4	格式化读写函数 fscanf 和 fprintf	/311
11.5	文件的随机读写	/312
11.5.1	文件定位	/312
11.5.2	文件的随机读写	/313
11.6	文件检测函数	/314
11.7	应用实例	/315
11.8	本章小结	/317
11.9	习题	/317
第 12 章	预处理命令	/322
12.1	C 语言预处理概述	/322
12.2	宏定义	/322
12.2.1	无参宏定义	/323

12.2.2	有参宏定义	/325
12.3	文件包含命令	/329
12.4	C 语言条件编译	/330
12.5	应用实例	/333
12.6	本章小结	/334
12.7	习题	/334
第 13 章 位运算 /338		
13.1	C 语言位运算符	/338
13.2	C 语言位域(位段)	/340
13.3	应用实例	/342
13.4	本章小结	/344
13.5	习题	/344
第 14 章 C 语言应用案例 /347		
14.1	应用实例	/347
14.1.1	简单的接口程序	/347
14.1.2	大整数四则运算	/348
14.1.3	学生成绩管理系统	/350
14.1.4	模拟时钟的实现	/352
14.2	习题	/353
附录 A ASCII 与扩展 ASCII 编码表 /354		
附录 B C 语言关键字大全 /355		
附录 C C 语言运算符及其优先级 /357		
附录 D C 语言编程时常见错误汇编 /358		
附录 E VC++ 6.0 程序调试常见错误信息 /366		
附录 F C 语言库函数 /369		
参考文献 /376		

第 1 章 C 语言概述

C 语言功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好,既具有高级语言的优点,又具有低级语言的许多特点;既适于编写系统软件,又能方便地用来编写应用软件。20 世纪 90 年代以来,C 语言迅速在全世界普及推广。目前,C 语言仍然是最优秀的程序设计语言之一。

学习重点和难点:

- C 语言简介与语言的特点。
- C 语言程序结构。
- 程序运行环境。

学习本章后将对 C 语言及 C 语言程序有初步认识,并能开展 C 语言程序的运行与实践。

1.1 程序设计语言简介

语言? 程序? 程序设计?

自从第一台计算机诞生以来,程序设计语言和程序设计方法在不断发展着。

语言是思维的载体。人和计算机打交道,必须要解决“语言”沟通的问题。计算机并不能理解和执行人们使用的自然语言,而只能接受和执行二进制的指令。计算机能够直接识别和执行的这种指令,称为机器指令。这种机器指令的集合就是机器语言指令系统,简称为机器语言。为了解决某一特定问题,需要选用指令系统中的某一些指令,这些指令按要求选取并组织起来就组成一个“程序”。如下程序是 8086 指令系统对应的二进制代码程序,能完成两个十六进制数相加的功能:

```
10111000 00111111100001011
10001110 11011000
10100001 0000000000000010
00000001 00000110 0000000000000000
10110100 01001100
11001101 00100001
```

换言之,一个程序是完成某一特定任务的一组指令序列,或者说,为实现某一算法的指令序列称为“程序”,机器世界中真正存在的就是这样的二进制程序。

用机器语言编制的程序虽然能够直接被计算机识别、直接执行,但是机器语言本身是随不同类型的机器而异,所以可移植性差,而且机器语言本身难学、难记、难懂、难修改,给使用者带来极大的不便。于是,为了绕开机器指令,克服机器指令程序的缺陷,人们提出了程序设计语言的构想,即使用人们熟悉、习惯的语言符号来编写程序,最好是直接使用人们间交流的自然语言来编程。在过去的几十年中,人们创造了许多介于自然语言和机器指令之间

的程序设计语言。按语言的级别来分,则大致可分为汇编语言(低级)和高级语言(第三代、第四代、……)。

汇编语言的特点是使用一些“助记符号”来替代那些难懂难记的二进制代码,所以汇编语言相对于机器指令便于理解和记忆,但它和机器语言的指令基本上是一一对应,两者都是针对特定的计算机硬件系统的,可移植性差,因此称它们都是“面向机器的低级语言”。为了直观地了解汇编语言程序,下面给出一段实现 X、Y 两个 16 位二进制数相加的 8086 汇编程序:

```
;X,Y 分别为 16 位二进制数,程序实现 X=X+Y(不考虑 溢出)
DATA SEGMENT                ;定义数据段开始
X DW 123H                    ;定义一个字变量(16 位)X
Y DW 987H                    ;定义一个字变量(16 位)Y
DATA ENDS                    ;定义数据段结束
CODE SEGMENT                  ;定义代码段开始
ASSUME CS:CODE,DS:DATA       ;建立段寄存器与段名之间的映射关系
START:MOV AX,DATA             ;取 DATA 段地址送 AX 寄存器
MOV DS,AX                    ;将数据段段地址送数据段寄存器 DS
MOV AX,Y                      ;取变量 Y 值送给寄存器 AX
ADD X,AX                      ;将 X 的值与 AX 的内容相加,结果送给 X,实现 X=X+Y
MOV AH,4CH                   ;将 DOS 调用的 4CH 功能号送 8 位寄存器 AH
INT 21H                       ;执行 DOS 功能调用,退出程序,回到 DOS
CODE ENDS                     ;定义代码段结束
END START                     ;源程序结束,主程序从标号 START 开始
```

高级语言与自然语言(主要是英语)类似,由专门的符号根据词汇规则构成单词,由单词根据句法规则构成语句,每种语句有确切的语义并能由计算机解释。高级语言包含许多英语单词,有“自然化”的特点;高级语言书编写计算公式接近于大家熟知的数学公式的规则。高级语言与机器指令完全分离,具有通用性,一条高级语言语句常常相当于几条或几十条机器指令。所以高级语言的出现,给程序设计从形式和内容上都带来了重大的变革,大大方便了程序的编写,提高了可读性。例如,BASIC、C、Visual Basic(简称 VB)、Visual C++(简称 VC++)、VB.NET、C#.NET、Java 等都是高级语言。高级语言一般能细分为第三代高级语言、第四代高级语言、……,分类依据是高级语言的逻辑级别、表达能力、接近自然语言的程度等。如 Turbo C 2.0(简称 TC)为第三代高级语言,而 VB 6.0、VC++ 6.0、C#、VB.NET、Java 等可认为是第四代高级语言。第四代高级语言一般是具有面向对象特性、具有快速或自动生成部分应用程序能力的高级语言,它表达能力强,编写程序效率高,更接近人的自然语言,高一级别的语言一般具有低一级别语言的语言表达能力。如下是输入两个整数并随即显示两整数之和的 Turbo C 2.0 语言程序:

```
#include<stdio.h>                /* Turbo C2.0 在 DOS 环境运行的 */
void main()                       /* main 主函数 */
{ int num1,num2;                  /* 定义两个整型变量 */
    printf("Input two numbers: "); /* 屏幕上显示输入提示 */
    scanf("%d %d",&num1,&num2);    /* 通过键盘读两个整数 */
```

```
printf("The sum is %d\n", num1+num2);    /* 屏幕上显示两整数之和 */  
}
```

显然,高级语言程序要比面向机器的低级语言要易懂、明了,并且简短得多。

应该看到的是:高级语言是不断发展变化的,不断有新的更好的语言产生,同时也有旧且功能差而不再实用的语言消亡。而C语言自产生以来,已历经40余年,依然具有强大的生命力与活力,该语言依然是当今最热门、最实用的高级语言之一。

1.2 C语言发展过程

在学习C语言之前,我们先了解一下C语言的历史。

C语言是一门通用的、模块化、程序化的编程语言,被广泛应用于操作系统和应用软件的开发。由于其高效和可移植性,适应于不同硬件和软件平台,深受开发员的青睐。

1. C语言早期发展

1969—1973年,在AT&T贝尔实验室开始了C语言的最初研发。据C语言的发明者丹尼斯·里奇(Dennis Ritchie)说,C语言最重要的研发时期是在1972年。

1970年,美国贝尔实验室的Ken Thompson以BCPL(Basic Combined Programming Language)语言为基础,设计出很简单且很接近硬件的B语言(取BCPL的首字母),并且用B语言写了第一个UNIX操作系统。

在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的第二个字母作为这种语言的名字,这就是C语言。因此,可以说C语言的祖先是BCPL语言。

C语言的诞生是和UNIX操作系统的开发密不可分的,原先的UNIX操作系统都是用汇编语言写的,1973年UNIX操作系统的核心用C语言改写,从此以后,C语言成为编写操作系统的主要语言。

2. ANSI C标准

20世纪70~80年代,C语言被广泛应用,从大型主机到小型微机,衍生了C语言的很多不同版本。为统一C语言版本,1983年美国国家标准局(American National Standards Institute, ANSI)成立了一个委员会,来制定C语言标准。1989年C语言标准被批准,被称为ANSI X3.159-1989“Programming Language C”。这个版本的C语言标准通常被称为ANSI C(C89)。目前,几乎所有的开发工具都支持ANSI C标准,是C语言使用最广泛的一个标准版本。

3. C99标准

在ANSI C标准确立之后,C语言的规范在很长一段时间内都没有大的变动。1995年WG14小组对C语言进行了一些修改,成为后来的1999年发布的ISO/IEC 9899:1999标准,通常被称为C99。但是各个公司对C99的支持所表现出来的兴趣不同。当GCC和其他一些商业编译器支持C99的大部分特性的时候,微软和Borland公司却似乎对此不感兴趣。

4. ISO发布C语言标准新版本

ISO(International Organization for Standardization)于2011年4月正式公布C语言新的国际标准草案。之前被命名为C1X的新标准将被称为ISO/IEC 9899:2011(C11版)。新

的标准修订了 C11 版本,提高了对 C++ 的兼容性,并将新的特性增加到 C 语言中。

新功能包括支持多线程,基于 ISO/IEC TR 19769:2004 规范下支持 Unicode,提供更多用于查询浮点数类型特性的宏定义和静态声明功能。根据草案规定,最新发布的标准草案修订了许多特性,支持当前的编译器。

5. C 语言对其他语言的影响

很多编程语言都深受 C 语言的影响,比如 C++ (原先是 C 语言的一个扩展)、C#、Java、PHP、JavaScript、Perl、LPC 和 UNIX 的 C Shell。也正因为 C 语言的影响力,掌握 C 语言的人,再学其他编程语言,大多能很快上手,触类旁通。

6. 目前 C 语言的商用版本

目前最流行的 C 语言有以下几种:

- (1) Microsoft C 或称 MS C。
- (2) Borland C(简称 BC)或 Turbo C。
- (3) Win-TC。
- (4) AT&T C。
- (5) Objective-C。

这些 C 语言版本不仅实现了 ANSI C 标准(C89 版),而且在此基础上各自做了一些扩充,使之更加方便实用。

1.3 C 语言的特点

C 语言是非常优秀的程序语言之一,C 语言是进一步学习面向对象语言 C++ 的基础。

1.3.1 C 语言是优秀的程序语言

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. C 语言的优秀之处

1) 语言简洁,使用方便灵活

C 语言是现有程序设计语言中规模最小的语言之一,而小的语言体系往往能设计出较好的程序。C 语言的关键字很少,ANSI C 标准一共只有 37 个关键字(详见附录 B)。9 种控制语句,压缩了一切不必要的成分。C 语言的书写形式比较自由,表达方法简洁,使用一些简单的方法就可以构造出相当复杂的数据类型和程序结构。

2) 可移植性好

用过汇编语言的读者都知道,即使是功能完全相同的程序,对于不同的单片机(或 CPU),必须采用不同的汇编语言来编写。这是因为汇编语言完全依赖于单片机硬件。而现代社会中新器件的更新换代速度非常快,也许我们每年都要跟新的单片机打交道。如果每接触一种新的单片机就要学习一次新的汇编语言,那么也许我们将一事无成,因为每学一种新的汇编语言,少则几月,多则上年,那么还有多少时间真正用于产品开发呢?

C 语言是通过编译来得到可执行代码的,统计资料表明,不同机器上的 C 语言编译程序