



普通高等教育“十一五”规划教材

C语言程序设计

(第二版)

王曙燕/主编



科学出版社
www.sciencep.com

普通高等教育“十一五”规划教材

C 语言程序设计

(第二版)

王曙燕 主 编

曹 锰 王春梅 王小银 王 燕 副主编

科学出版社

北 京

内 容 简 介

本书共 13 章, 主要内容包括 C 语言的基本概念、基本语法和基本数据结构、C 和汇编语言的混合编程、编译预处理等知识, 并给出了一个小型超市管理的综合实例, 介绍了 C 语言图形程序设计的基本方法, 还简单介绍了 C++、Visual C++ 和 C# 等面向对象程序设计语言。

本书注重应用性和实践性, 通过一些典型算法的解题分析及其实现给读者一些解题示范和启发。每章后面配有习题, 并提供配套教材《C 程序设计习题与实验指导》。

本书可作为高等院校各专业 C 语言程序设计课程的教材, 也可供准备参加计算机等级考试和考研的读者阅读参考, 同时也可作为工程技术人员和计算机爱好者的参考用书。

图书在版编目 (CIP) 数据

C 语言程序设计/王曙燕主编. —2 版. —北京: 科学出版社, 2008
(普通高等教育“十一五”规划教材)

ISBN 978-7-03-022663-1

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

中国版本图书馆 CIP 数据核字 (2008) 第 116428 号

责任编辑: 陈晓萍 / 责任校对: 刘彦妮

责任印制: 吕春珉 / 封面设计: 耕者设计工作室

科学出版社出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

新蕾印刷厂印刷

科学出版社发行 各地新华书店经销

*

2005 年 2 月第 一 版 开本: 787×1092 1/16

2008 年 8 月第 二 版 印张: 24 3/4

2008 年 8 月第六次印刷 字数: 584 000

印数: 13 001—17 000

定价: 38.00 元

(如有印装质量问题, 我社负责调换〈环伟〉)

销售部电话 010-62134988 编辑部电话 010-62135120-8003

版权所有, 侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303

前 言

C 语言程序设计是理工科各专业计算机应用能力培养的重要技术基础。C 语言既具有低级语言可直接访问内存地址、能进行位操作、程序运行效率高的优点,又具有高级语言运算符和数据类型丰富、结构化控制语句功能强、可移植性好的优点,因此成为程序设计语言的常青树。通过本课程的学习,学生可以掌握 C 语言的程序结构、语法规则和编程方法,达到独立编写常规 C 语言应用程序的能力,同时为设计大型应用程序和系统程序打下坚实的基础。本课程是数据结构、面向对象程序设计、操作系统和软件工程等课程的基础,并可为这些课程提供实践工具。

本书以程序设计为主线,全面介绍 C 语言的基本概念、基本语法和基本数据结构。第 1、2 章介绍了 C 语言的特点和 C 语言的基本概念;第 3~5 章介绍了算法的描述和结构化程序设计的基本方法及三种基本结构(顺序结构、选择结构和循环结构);第 7 章介绍了函数的定义和使用;第 6、8、9 章介绍了数据的构造类型(包括数组、字符串、结构体和共同体等)和指针类型;第 10 和 11 章介绍了 C 和汇编语言的混合编程、编译预处理和文件的概念,并给出了一个小型超市管理的综合实例;第 12 章介绍了 C 语言图形程序设计;第 13 章介绍了面向对象程序设计语言 C++、Visual C++ 和 C# 等。

C 语言是一门实践性很强的课程,对训练学生的编程和调试能力非常重要。在第 3 章和附录 V、附录 VI 中,专门介绍了 C 语言的上机步骤和 C 程序的调试技术,并重点介绍了 Turbo C 2.0 集成环境。在第 13 章中也介绍了 C++ 的上机步骤和环境,便于有能力和兴趣的读者在 Windows 环境下编写 C 程序。本书中的例题均经过 Turbo C 2.0 运行环境的调试。

本书讲授时数为 60~70 学时,其中实验课占 24 学时。学习完本教材后,建议安排两周的“课程设计”,以完成一个小型应用系统的设计与实现。

本书是作者根据多年从事 C 语言的教学经验编写的,在第一版的基础上,根据学生、教师和广大读者使用中提出的要求和意见,进行了精心的修改,增加了总结与提高部分。为配合读者学习,作者另外编写了《C 程序设计习题与实验指导》作为本书的配套教材,已经于 2006 年出版。

本书由王曙燕任主编,并编写了第 1、3 和 5 章,曹锰编写了第 6、8 和 9 章,王春梅编写了第 4、11 章和附录,王小银编写了第 2、7 章,王燕编写了第 12、13 章,王春梅和刘鹏辉编写了第 10 章。王春梅、王小银和王燕老师参与了审校,最后由王曙燕统稿。陈莉君对教材的编写也提出了很好的建议。作者在此一并向他们表示衷心的感谢。

本书可作为大专院校各专业 C 语言程序设计课程的教材,也可供准备参加计算机等级考试和考研的读者阅读参考,同时也可作为工程技术人员和计算机爱好者的参考用书。

由于编者水平有限,书中难免存在缺点和错误,恳请专家和读者批评指正。

作者邮件地址: wsylxj@126.com。

王曙燕

2008 年 5 月

目 录

前言

第 1 章 概述	1
1.1 程序设计语言	1
1.1.1 计算机语言	1
1.1.2 程序设计语言的发展	1
1.2 C 程序设计语言	2
1.2.1 C 语言的发展历史	2
1.2.2 C 语言的标准与版本	3
1.2.3 C 语言的特点	4
1.3 C 语言的基本语法成分	4
1.3.1 字符集	4
1.3.2 标识符	5
1.3.3 运算符	6
1.4 C 语言程序的组成	6
1.4.1 简单的 C 程序介绍	6
1.4.2 C 语言程序的结构	8
1.4.3 C 程序的书写	9
习题 1	11
第 2 章 基本数据类型、运算符及表达式	12
2.1 C 语言的基本数据类型	12
2.1.1 数据类型	12
2.1.2 C 语言数据类型简介	12
2.1.3 C 语言的基本数据类型	13
2.2 常量	14
2.2.1 整型常量	14
2.2.2 实型常量	15
2.2.3 符号常量	15
2.2.4 字符型常量	16
2.3 变量	18
2.3.1 变量名	18
2.3.2 变量类型	18
2.3.3 变量值	19

2.4	运算符及表达式	19
2.4.1	C 运算符简介	19
2.4.2	算术运算符与算术表达式	20
2.4.3	赋值运算符与赋值表达式	23
2.4.4	逗号运算符和逗号表达式	25
2.4.5	不同类型数据间的混合运算与类型转换	25
2.5	总结与提高	27
2.5.1	重点与难点	27
2.5.2	典型题例	27
	习题 2	28
第 3 章	算法和基本程序设计	30
3.1	算法	30
3.1.1	算法的概念	30
3.1.2	算法的评价标准	31
3.1.3	算法的表示	32
3.2	结构化程序设计的方法	34
3.3	程序的基本结构	35
3.3.1	顺序结构	35
3.3.2	选择结构	35
3.3.3	循环结构	36
3.4	顺序结构程序设计	37
3.5	数据的输入与输出	39
3.5.1	C 语言中数据的输入与输出	39
3.5.2	字符数据的输入与输出	39
3.5.3	格式的输入与输出	41
3.6	总结与提高	47
3.6.1	典型题例	47
3.6.2	C 程序的上机步骤	48
3.6.3	Turbo C 2.0 简介	49
3.6.4	C 程序的基本调试技术	51
	习题 3	52
第 4 章	选择分支结构程序设计	53
4.1	关系运算	53
4.1.1	关系运算符	53
4.1.2	关系表达式	53
4.2	逻辑运算	54
4.2.1	逻辑运算符	54

4.2.2 逻辑表达式	55
4.3 二路分支——if 语句	56
4.3.1 if 语句的一般形式	56
4.3.2 if 语句的嵌套	64
4.4 条件运算符与表达式	66
4.5 多路分支——switch 语句	68
4.5.1 switch 语句的一般形式	68
4.5.2 switch 语句的嵌套	70
4.6 选择分支结构程序举例	70
4.7 总结与提高	74
4.7.1 重点与难点	74
4.7.2 典型题例	74
习题 4	77
第 5 章 循环结构程序设计	78
5.1 while 语句	78
5.2 do-while 语句	80
5.3 for 语句	82
5.4 goto 语句	85
5.5 循环的嵌套	86
5.6 break 和 continue 语句	88
5.6.1 continue 语句	88
5.6.2 break 语句	90
5.7 总结与提高	90
5.7.1 重点与难点	90
5.7.2 典型题例	91
习题 5	95
第 6 章 数组	97
6.1 一维数组	97
6.1.1 一维数组的定义和初始化	97
6.1.2 一维数组元素的引用	99
6.1.3 一维数组应用举例	99
6.2 二维数组	104
6.2.1 二维数组的定义和初始化	104
6.2.2 二维数组元素的引用	106
6.3 字符数组	111
6.3.1 字符数组的定义和引用	111
6.3.2 字符数组的输入与输出	112

6.3.3 字符串处理函数	114
6.4 总结与提高	120
6.4.1 重点与难点	120
6.4.2 典型题例	121
习题 6	126
第 7 章 函数	130
7.1 概述	130
7.1.1 C 程序的基本结构	130
7.1.2 函数分类	132
7.2 函数的定义	133
7.3 函数的参数和函数的返回值	134
7.3.1 形式参数和实际参数	135
7.3.2 函数与数组	136
7.3.3 函数的返回值	139
7.4 函数的调用	141
7.4.1 函数调用的一般形式	141
7.4.2 函数调用的方式	141
7.5 函数声明和函数原型	142
7.6 函数的嵌套调用	144
7.7 函数的递归调用	145
7.8 变量的作用域	149
7.8.1 局部变量	149
7.8.2 全局变量	150
7.9 变量的存储类型	152
7.9.1 静态存储方式和动态存储方式	152
7.9.2 静态局部变量	152
7.9.3 自动变量	153
7.9.4 寄存器变量	154
7.9.5 静态全局变量和非静态全局变量	155
7.9.6 存储类型小结	156
7.10 内部函数和外部函数	157
7.10.1 内部函数	157
7.10.2 外部函数	157
7.11 多文件程序的运行	158
7.12 总结与提高	159
7.12.1 重点与难点	159
7.12.2 典型题例	160

习题 7	164
第 8 章 结构体与共用体	168
8.1 结构体类型定义	168
8.2 结构体变量	169
8.2.1 结构体变量的定义与初始化	169
8.2.2 结构体变量的引用	171
8.3 结构体数组	173
8.3.1 结构体数组的定义与初始化	173
8.3.2 结构体数组元素的引用	174
8.4 结构体和函数	176
8.4.1 结构体作函数参数	176
8.4.2 返回值为结构体类型的函数	177
8.5 共用体	179
8.5.1 共用体类型定义	179
8.5.2 共用体变量定义与引用	179
8.5.3 共用体应用举例	181
8.6 枚举类型	183
8.7 typedef 语句	185
8.8 总结与提高	186
习题 8	188
第 9 章 指针	191
9.1 地址和指针的概念	191
9.1.1 变量的内容和变量的地址	191
9.1.2 直接访问和间接访问	192
9.1.3 指针的概念	193
9.2 指针变量	193
9.2.1 指针变量的定义	193
9.2.2 指针变量的引用	194
9.2.3 实现引用传递	196
9.3 指针与数组	199
9.3.1 指向数组元素的指针	199
9.3.2 字符指针、字符数组和字符串	204
9.3.3 地址越界问题	207
9.3.4 指针数组	208
9.3.5 多维数组和指向分数组的指针	210
9.4 结构体与指针	215
9.4.1 指向结构体的指针变量	215

9.4.2 用指向结构体的指针作函数参数	216
9.5 返回值为指针类型的函数	217
9.6 指针与链表	219
9.6.1 存储空间的分配和释放	219
9.6.2 动态数据结构——链表	220
9.7 总结与提高	226
9.7.1 重点与难点	226
9.7.2 典型题例	227
9.7.3 指向函数的指针和函数参数	228
习题 9	231
第 10 章 编译预处理和位运算及混合编程	234
10.1 文件包含处理——#include	234
10.2 宏定义——#define	236
10.2.1 不带参数的宏定义	236
10.2.2 带参数的宏定义	238
10.3 条件编译	239
10.4 位运算符和位运算	241
10.5 位段	243
10.6 位运算举例	245
10.7 C 语言与汇编语言的混合编程	246
10.7.1 内嵌汇编代码	247
10.7.2 模块化连接方法	249
习题 10	254
第 11 章 文件	257
11.1 文件的概述	257
11.1.1 数据流	257
11.1.2 文件	257
11.1.3 文件的操作流程	259
11.1.4 文件和内存的交互处理	260
11.2 文件类型的指针	261
11.3 标准输入/输出函数	261
11.3.1 打开文件	262
11.3.2 关闭文件	264
11.3.3 获取文件的属性	264
11.3.4 文件的顺序读写	266
11.3.5 文件的随机读写	277
11.3.6 出错检查	280

11.4 系统输入/输出函数	282
11.5 总结与提高	282
11.5.1 重点与难点	282
11.5.2 典型题例	283
习题 11	293
第 12 章 C 语言图形程序设计	295
12.1 图形显示的坐标和像素	295
12.1.1 图形显示的坐标	295
12.1.2 像素	295
12.2 Turbo C 支持的图形适配器和图形模式	296
12.3 图形模式的初始化	298
12.3.1 图形系统的初始化函数	298
12.3.2 图形系统的自动检测函数	299
12.3.3 关闭图形模式函数	300
12.3.4 屏幕颜色的设置	300
12.3.5 清屏函数	301
12.4 基本绘画	301
12.4.1 画点	301
12.4.2 画直线函数	302
12.4.3 画矩形和多边形函数	304
12.4.4 画圆和椭圆函数	305
12.4.5 设定线形函数	306
12.5 图形填充	307
12.5.1 画填充图函数	308
12.5.2 设定填充方式函数	309
12.5.3 可对任意封闭图形填充的函数	311
12.6 图形模式下的文本输出	312
12.6.1 文本输出函数	312
12.6.2 字体格式和输出方式的设置	313
12.7 视口和视口函数	315
12.8 屏幕操作	318
12.8.1 设置显示页函数	318
12.8.2 屏幕图像处理存储和显示函数	319
12.8.3 键盘对屏幕的控制操作	321
12.9 总结与提高	323
12.9.1 重点与难点	323
12.9.2 典型题例	324

习题 12	327
第 13 章 C++面向对象程序设计	330
13.1 面向对象程序设计	330
13.1.1 面向对象程序设计的产生背景	330
13.1.2 类和对象	331
13.1.3 构造函数和析构函数	333
13.1.4 继承	333
13.1.5 多态性	334
13.2 C++	336
13.2.1 C++语言的特点	336
13.2.2 输出流、输入流	336
13.2.3 函数内联	340
13.2.4 函数重载	341
13.2.5 引用	344
13.2.6 C++增加的运算符、数据类型、注释	346
13.2.7 C++程序的集成开发环境	347
13.3 C#语言	348
13.3.1 C#简介	348
13.3.2 C#的特点	348
13.4 总结与提高	349
13.4.1 重点与难点	349
13.4.2 典型题例	349
习题 13	352
附录	354
附录 I 常用字符与 ASCII 码对照表	354
附录 II C 语言中的关键字表	357
附录 III C 语言中的运算符的优先级与结合性一览表	357
附录 IV C 库函数	358
附录 V Turbo C 2.0 菜单介绍	367
附录 VI Turbo C 2.0 编译错误信息介绍	375
主要参考文献	382

第1章 概述

1946年,世界上第一台电子数字式计算机在美国宾夕法尼亚大学诞生,这就是ENIAC(Electronic Numerical Integrator and Calculator),电子数值积分计算机。ENIAC奠定了电子计算机的发展基础,开辟了一个计算机科学技术的新纪元。在此后短短的几十年间,计算机技术发展突飞猛进。伴随着硬件的发展,计算机软件、计算机网络技术也得到了迅速的发展,计算机的应用领域也由最初的数值计算扩展到人类生活的各个领域,现在计算机已成为最基本的信息处理工具。

人们使用计算机解决问题时,必须用某种“语言”来和计算机进行交流。具体地说,就是利用某种计算机语言提供的命令来编制程序,并把程序存储在计算机的存储器中,然后利用这个程序来控制计算机的运行,以达到解决问题的目的。这种用于编写计算机可执行程序的语言称为程序设计语言。目前已发明的计算机程序设计语言有上千种,无论什么样的计算机语言,其程序设计的基本方法都是相同的。本书将以国际上广泛流行的C程序设计语言为例,介绍程序设计的基本概念和基本方法。

1.1 程序设计语言

1.1.1 计算机语言

计算机语言是人们描述计算过程(即程序)的规范书写语言。程序是对计算机处理对象和计算规则的描述。语言的基础是一组记号和语法规则。根据语法规则由记号构成记号串的全体就是语言。

人类是使用像英语、汉语这样的自然语言相互交流和表达思想的。人与计算机如何“交流”呢?人和计算机交流也要用人和计算机都容易接受和理解的语言,这就是计算机语言。计算机语言是根据计算机的特点而编制的,是计算机能够“理解”的语言,它是有限规则的集合。计算机语言不像自然语言那样包含复杂的语义和语境,而是用语法来表达程序员的思想,所以编写程序时必须严格遵守语法规则。

1.1.2 程序设计语言的发展

计算机是一种具有内部存储能力、由程序自动控制的电子设备。人们将需要计算机做的工作写成一定形式的指令,并把它们存储在计算机的内部存储器中。当人们需要结果时就向计算机发出一条简单的命令,计算机就按指令顺序自动进行操作。人们把这种可以连续执行的一条条指令的集合称为程序。也就是说,程序是计算机指令的序列,编制程序的工作就是为计算机安排指令序列。

程序设计语言伴随着计算机技术的发展层出不穷,从机器语言到高级语言,从面向

过程的语言到面向对象的语言。至目前为止, 计算机语言的发展大致经历了五代。

第一代也称机器语言, 它将计算机指令中的操作码和操作数均以二进制代码形式表示, 是计算机能直接识别和执行的语言。它在形式上是由“0”和“1”构成的一串二进制代码, 每种计算机都有自己的一套机器指令。机器语言的优点是无需翻译、占用内存少、执行速度快。缺点是随机而异, 通用性差, 并且因指令和数据都是二进制代码形式, 难于阅读和记忆, 编码工作量大, 难以维护。

第二语言也叫汇编语言, 是用助记符号来表示机器指令的符号语言。如用 ADD 表示加法, 用 SUB 表示减法, 用变量取代各类地址, 这样构成的计算机符号语言称为汇编语言。用汇编语言编写的程序称为汇编语言源程序。这种程序必须经过翻译(称为汇编), 变成机器语言程序才能被计算机识别和执行。汇编语言在一定程度上克服了机器语言难于辨认和记忆的缺点, 但对于大多数用户来说, 仍然不便于理解和使用。

第三代语言即高级语言, 也称为面向过程的语言。高级语言是具有国际标准的, 描述形式接近自然语言的计算机语言。它采用了完全符号化的描述形式, 用类似自然语言的形式描述对问题的处理过程, 用数学表达式的形式描述对数据的计算过程。常用的计算机高级语言有 Basic、Fortran、Cobol、Pascal 和 C 语言等。由于高级语言只是要求人们关心计算机描述问题的求解过程, 而不关心计算机的内部结构, 所以把高级语言称为面向过程的语言。使用面向过程的语言编程时, 编程者的主要精力放在算法过程的设计和描述上。

第四代语言又叫非过程化语言, 是一种功能更强的高级语言。主要特点是: 非过程性、采用图形窗口和人机对话形式、基于数据库和“面向对象”技术。易编程、易理解、易使用、易维护。但是, 程序的运行效率和语言的灵活性不如过程化语言高。常用的非过程化语言有 Visual Basic、Java、C++ 和 Delphi 等。

如果说第三代语言要求人们告诉计算机“怎么做”, 那么第四代语言只要求人们告诉计算机“做什么”。

第五代语言也称智能化语言, 主要使用在人工智能领域, 帮助人们编写推理、演绎程序。

目前, 国内外大多数计算机上运行的程序, 大多是用第三代或第四代计算机语言编写的。因此, 应当熟练地掌握用高级语言编写程序的方法和技巧。

由于面向过程的语言是程序设计的基础, 所以, 本书将以面向过程的 C 语言为背景, 介绍程序设计的基本概念和基本方法。在本书最后一章, 将简单介绍面向对象程序设计的基本方法及流行的面向对象程序设计语言 C++ 和 C#。

1.2 C 程序设计语言

1.2.1 C 语言的发展历史

C 语言是国际上流行的, 使用最广泛的高级程序设计语言。它既可用于写系统软件, 也可用来写应用软件。C 语言具有语言简洁紧凑, 使用方便灵活及运算符丰富等特点,

它具有现代化语言的各种数据结构,有结构化的控制语句,并且语法限制不太严格,程序设计自由度大,能实现汇编语言的大部分功能。另外,C语言生成目标代码质量高,不仅程序执行效率高,而且程序可移植性好。

C语言的产生基于两个方面的需要。一是为满足UNIX操作系统开发的需要。UNIX操作系统是一个通用的、复杂的计算机管理系统。二是为接近硬件的需要,即直接访问物理地址、直接对硬件进行操作的需要。C语言集高级语言与汇编语言优点于一身。C语言面对实际应用的需要而产生,直至今日仍不改初衷。C语言是从BCPL(Basic Combined Programming Language)语言和B语言演化而来的。1960年出现的ALGOL语言是一种面向问题的高级语言,远离硬件,但不适于开发系统软件。1963年,英国剑桥大学推出CPL语言,CPL语言比ALGOL语言接近硬件一些,但规模较大,难以实现。1969年,剑桥大学的M.Richards对CPL语言进行简化,推出BCPL语言。1970年,贝尔实验室的K.Thompson为DEC公司PDP-7计算机上运行的一种早期UNIX操作系统设计了一种类BCPL语言,称为B语言。B语言规模小,接近硬件,1971年在PDP-11计算机上实现,并用B语言编写了UNIX操作系统和绝大多数实用程序。由于B语言面向字存取、功能过于简单、数据无类型和描述问题的能力有限,而且编译程序产生的是解释执行的代码,运行速度慢,因此没有流行起来。1972~1973年,贝尔实验室的D.M.Ritchie在保留B语言优点的基础上,设计出一种新的语言。这种新语言克服了B语言功能过于简单、数据无类型和描述问题的能力有限的缺点,扩充了很多适合于系统设计和应用开发的功能。因为这种新语言是在B语言的基础上开发出来的,不管是在英文字母序列中也好,还是在BCPL这个名字中也好,排在B后面的均为C,因此将这种语言命名为C语言。1973年,UNIX操作系统被用C语言改写,称为UNIX第五版。最初的C语言只是一种UNIX操作系统的工作语言,依附于UNIX系统,主要在贝尔实验室内部使用。

UNIX以后的第六版、第七版、SYSTEM III和SYSTEM V都是在第五版的基础上发展起来的,C语言也做了多次改进。到1975年,随着UNIX第六版的公布,C语言越来越受到人们的普遍注意。

UNIX操作系统的广泛使用,促进了C语言的迅速发展及普及。同时,C语言的发展与普及也促进了UNIX操作系统的推广。到1978年出现了独立于UNIX和PDP计算机的C语言,从而,C语言被迅速移植到大、中、小与微型机上。当年,B.W.Kernighan和D.M.Ritchie以UNIX第七版的C编译程序为基础,出版了影响深远的名著《C程序设计语言》。如今,C语言已经风靡全球,成为世界上应用最广泛的程序设计语言之一。

1.2.2 C语言的标准与版本

随着C语言的普及,各机构分别推出了自己的C语言版本。某些执行过程的微小差别不时引起C程序之间的不兼容,给程序的移植带来很大的困难。美国国家标准协会(ANSI)从1983年开始,经过长达5年的努力,制定了C语言的新标准——ANSI C。现在提及C语言的标准就是指该标准。ANSI C比原标准C有很大的发展,解决了经典定义中的二义性,给出了C语言的新特点。任何C程序都必须遵循ANSI C标准,本教

材也以 ANSI C 作为基础。尽管这样,各种版本的 C 编译系统还是略有差异,因此,读者在使用具体的 C 语言编译系统时,还应参考相关的手册以了解具体的规定。

C 语言有多种版本: TC 2.0, BC 3.1, BC 5.0 和 VC 6.0 等,最新的是为 Windows 和 Web 应用程序提供动态开发环境的 Visual C++、.NET (C#)。

1.2.3 C 语言的特点

C 语言的主要特点有如下几方面。

(1) C 语言简洁、紧凑,编写的程序短小精悍

C 编译程序的代码量较小,便于在微型机上应用。

(2) 运算符丰富,数据结构丰富

C 的数据类型有:整型、实型、字符型、数组类型、指针类型、结构体类型和共用体类型等,能实现各种复杂数据类型的数据进行运算,并引入了指针概念,使程序效率更高。另外 C 语言具有强大的图形功能。

(3) C 语言是一种结构化程序设计语言,具有结构化语言所要求的三种基本结构

这种结构化方式可使程序层次清晰,便于使用、维护和调试。C 语言是以函数形式提供给用户的,这些函数可方便地调用,并具有多种循环、条件语句控制程序流向,从而使程序完全结构化。

(4) C 语言允许直接访问物理地址

C 语言能进行位运算,能实现汇编语言的大部分功能,能直接对硬件进行操作。这使 C 语言既具有高级语言的功能,又具有低级语言的许多特点,可以用来写系统程序。

(5) C 语言预处理机制

C 语言提供预处理机制,有利于大程序的编写和调试。

(6) C 语言可移植性好

编写的程序不需要做很多改动就可从一种机型上移到另一种机型上运行。C 语言有一个突出的优点就是适合于多种操作系统,如 DOS、UNIX,也适用于多种机型。

(7) C 语言语法限制不太严格,程序设计自由度大,对程序员要求不高

一般的高级语言语法检查比较严格,能够检查出几乎所有的语法错误。而 C 语言允许程序编写者有较大的自由度。

(8) C 语言程序生成代码质量高,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%。

C 语言也存在某些缺点,例如,运算符较多,某些运算符优先顺序与习惯不完全一致,类型转换比较随便等。

1.3 C 语言的基本语法成分

1.3.1 字符集

字符是可以区分的最小符号,是构成程序的基础。C 语言字符集是 ASCII 字符集的