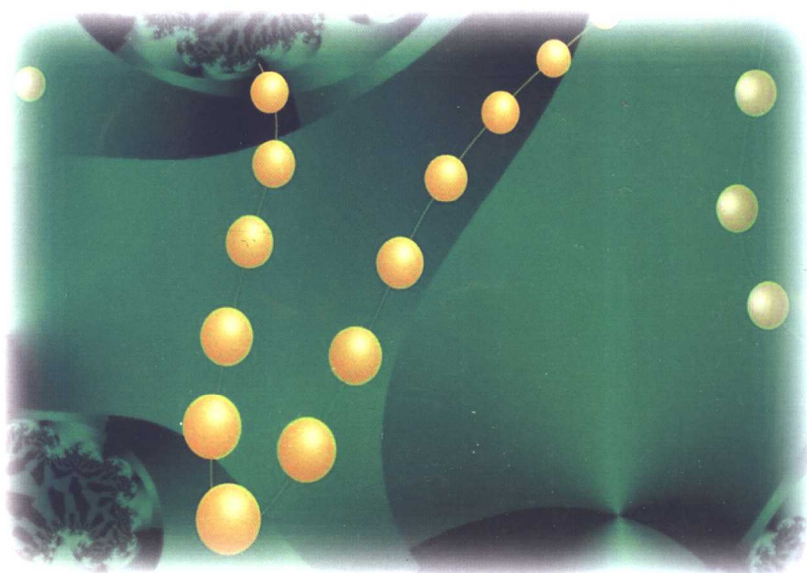




清华大学计算机基础教育课程系列教材

C 语言程序设计

黄维通 马力妮 编著



清华大学出版社



清华大学计算机基础教育课程系列教材

C 语言程序设计

黄维通 马力妮 编著



B1283030

清华大学出版社
北 京

内 容 简 介

本书从 C 语言程序设计的基本原理及程序设计的基本思想出发,贯穿“基础—应用—专题”这一主线,紧扣基础,重点突出,循序渐进,面向应用。

本教材基础部分的主要内容包括程序设计中的基本概念与应用,如变量、数组、控制结构等;在掌握了这些基本概念与应用的基础上引入函数的结构与应用、指针的概念及其应用、图形用户界面的设计与应用、算法设计与实现、结构型数据的应用及文件的操作等面向应用的知识点介绍;然后在上述知识点的基础上进一步介绍数据结构专题,包括链表、栈、队列和二叉树的概念与应用。

本书语言表达严谨、流畅,实例丰富,书中例题的代码都做了详细注释,便于自学。与本书配套的《C 语言程序设计电子教案》和《C 语言程序设计习题解析》将由清华大学出版社出版。

本书适合作为大专院校程序设计课程的入门教材,也可供计算机水平考试培训及各类成人教育等教学使用,还可供计算机爱好者自学。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

C 语言程序设计/黄维通,马力妮编著. —北京:清华大学出版社,2003

(清华大学计算机基础教育课程系列教材)

ISBN 7-302-06483-0

I. C… II. ①黄… ②马… 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(2003)第 022624 号

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

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

责任编辑: 王敏稚

印 刷 者: 北京市人民文学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 22.75 **字数:** 520 千字

版 次: 2003 年 5 月第 1 版 2003 年 5 月第 1 次印刷

书 号: ISBN 7-302-06483-0/TP·4873

印 数: 0001~8000

定 价: 28.00 元

序

计算机科学技术的发展不仅极大地促进了整个科学技术的发展,而且明显地加快了经济信息化和社会信息化的进程。因此,计算机教育在各国备受重视,计算机知识与能力已成为 21 世纪人才素质的基本要素之一。

清华大学自 1990 年开始将计算机教学纳入基础课的范畴,作为校重点课程进行建设和管理,并按照“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次的课程体系组织教学:

第一层次“计算机文化基础”的教学目的是培养学生掌握在未来信息化社会里更好地学习、工作和生活所必须具备的计算机基础知识和基本操作技能,并进行计算机文化道德规范教育。

第二层次“计算机技术基础”是讲授计算机软硬件的基础知识、基本技术与方法,从而为学生进一步学习计算机的后续课程,并利用计算机解决本专业及相关领域中的问题打下必要的基础。

第三层次“计算机应用基础”则是讲解计算机应用中带有基础性、普遍性的知识,讲解计算机应用与开发中的基本技术、工具与环境。

以上述课程体系为依据,设计了计算机基础教育系列课程。随着计算机技术的飞速发展,计算机教学的内容与方法也在不断更新。近几年来,清华大学不断丰富和完善教学内容,在有关课程中先后引入了面向对象技术、多媒体技术、Internet 与互联网技术等。与此同时,在教材与 CAI 课件建设、网络化的教学环境建设等方面也正在大力开展工作,并积极探索适应 21 世纪人才培养的教学模式。

为进一步加强计算机基础教学工作,适应高校正在开展的课程体系与教学内容的改革,及时反映清华大学计算机基础教学的成果,加强与兄弟院校的交流,清华大学在原有工作的基础上,重新规划了“清华大学计算机基础教育课程系列教材”。

该系列教材有如下几个特色:

1. 自成体系:该系列教材覆盖了计算机基础教学三个层次的教学内容。其中既包括所有大学生都必须掌握的计算机文化基础,也包括适用于各专业的软、硬件基础知识;既包括基本概念、方法与规范,也包括计算机应用开发的工具与环境。

2. 内容先进:该系列教材注重将计算机技术的最新发展适当地引入教学中来,保持了教学内容的先进性。例如,系列教材中包括了面向对象与可视化编程、多媒体技术与应用、Internet 与互联网技术、大型数据库技术等。

3. 适应面广：该系列教材照顾了理、工、文等各种类型专业的教学要求。

4. 立体配套：为适应教学模式、教学方法和手段的改革，该系列教材中多数都配有习题集和实验指导、多媒体电子教案，有的还配有 CAI 课件以及相应的网络教学资源。

本系列教材源于清华大学计算机基础教育的教学实践，凝聚了工作在第一线的任课教师的教学经验与科研成果。我希望本系列教材不断完善，不断更新，为我国高校计算机基础教育做出新的贡献。



1999 年 12 月

注：周远清，曾任教育部副部长，原清华大学副校长、计算机专业教授。

前言

人类已经步入信息化的 21 世纪,信息时代的到来,使社会经济向知识经济发展。为了推进 21 世纪计算机基础教育改革、推进精品课程建设以及与之配套的精品教材建设,从素质教育的理念出发,结合信息化社会对高素质、复合型人才的需求,特出版此教材。此教材的内容及语法均参照目前 C 语言最新的标准 ANSI C99(这是一个向下兼容的 C 语言的新标准)。此教材力求较为全面地介绍 C 语言的基本内容与程序设计思想。

C 语言是目前国内外广泛使用的程序设计语言之一,是国内外大学都在开设的重要的基础课之一。C 语言功能丰富、表达能力强、使用方便灵活、程序执行效率高、可移植性好;既具有高级语言的特点,又具有汇编语言的特点,具有较强的系统处理能力。它支持自顶向下逐步求精的程序设计技术,其函数式结构为实现程序的模块化设计提供了强有力的保障。因此,它被广泛应用于系统软件和应用软件的开发。

本教材从 C 语言程序设计的基本原理及程序设计的基本思想出发,贯穿“基础—应用—专题”这一主线,紧扣基础,面向应用,循序渐进地引导读者学习程序设计的思想和方法。基础部分的主要内容包括程序设计中的基本概念与应用,如变量、数组、控制结构及判断结构等,在掌握了这些基本概念与应用的基础上适时引入函数的结构与应用、指针的概念及其应用、图形用户界面的设计与应用、算法设计与实现、结构型数据的应用及文件的操作等面向应用的知识点介绍,然后在上述知识点的基础上进一步介绍数据结构专题,包括链表、栈、队列和二叉树的概念与应用。为方便授课和学习,本教材还配套有电子教案和习题解析。

本书的特点之一是通俗易懂,突出“三基”(基本概念、基本原理与基本应用)的介绍与应用。

本书的特点之二是在介绍“算法设计与实现”这部分内容时,重点介绍经典的排序与查找的算法及其实现,同时通过几种不同算法的比较,讨论算法的效率及代码实现的效率。计算机程序设计,很大一部分工作是为了实现某一特定的模型或算法,通过介绍算法设计并通过计算机程序加以实现,可以使读者更好地学习程序设计的思想、体系结构和方法,尤其是优化的程序设计方法。

本书的特点之三是引入了“图形用户界面的设计与应用”,使得读者在很好地完成功能实现的基础上设计出界面更加友好的程序,体现“易用性、界面友好”等高质量程序的特点。

本书的特点之四是介绍了数据结构的基本内容,如链表、栈、队列和二叉树的“三基”,将这些内容作为综合应用的专题,使读者从实际应用中对 C 语言程序设计的基本知识得以融会贯通和进一步提高。

本书的特点之五是教材中有代码的详细注释,为了用最简洁的语言讲解代码结构及

功能,本教材中的例题代码中给出了详细的代码注释,以利于读者更好地理解代码,同时,教材中的所有代码均在 TC3.0 下,完全调试通过。

本书的特点之六是采用美国最新的 C 语言国家标准 ANSI C99,使学习内容在兼容以前版本的基础上与国际接轨。

本书的特点之七是重点突出、难点分散,这样从学习者对知识的掌握角度来说是有利的,而且这样也有利于自学。本教材标有“*”的内容为选学内容,各学校可以根据自己的实际情况决定教材的学习内容,如果这些内容不作为教学内容学习的话,不影响其他内容在知识点上的承接关系。

本教材中将“编译预处理”内容放到附录中去,由于此部分内容跨越了很多知识点,如果单独作为一章内容,那么只能将这部分内容放到较靠后的位置,但基础知识的学习中也牵涉到编译预处理的内容,因此,将这部分内容放到附录中,可以根据需要及时查询,读者可以根据实际情况安排学习编译预处理内容的进度。

本书可供各高等院校、计算机水平考试培训、各类成人教育学校作为开设程序设计课程的教材,也可供计算机爱好者自学。

本书由黄维通、马力妮、孟威、关继来、邱春风等编写,同时参与编写工作的有成竟峰、马励华、章国利、刘祥、顾飞、谢国华、郑会理,参加资源建设的有关发达、蒋孝蔚、高屹等,参加程序调试、校验的有刘世清、张义、程昆鹏、黄兆文、赵松、宋克宏等,感谢清华大学出版社为编写本书给予的大力支持。

由于作者水平有限,加上时间仓促,书中缺点和错误在所难免,恳请读者批评指正并通过作者电子信箱告知。不胜感激,谢谢!

作者联系信箱: hwt@cic.tsinghua.edu.cn

黄维通

2003 年 2 月于清华园

目 录

第 1 章 C 语言的基本概念	1
1.1 C 语言的发展与特点	1
1.1.1 C 语言的发展	1
1.1.2 C 语言的特点	1
1.2 几个基本概念	2
1.2.1 程序	2
1.2.2 程序设计	3
1.2.3 算法	3
1.2.4 数据结构	3
1.3 软件工程的观念	4
1.4 C 语言程序的基本标识符	4
1.5 C 程序的几个简单实例	6
1.6 C 程序的结构特点	8
1.7 C 语言程序的编译和执行	9
习题	10
第 2 章 C 语言程序的基本数据类型及其运算	11
2.1 C 语言的数据类型	11
2.1.1 数据类型的一般概念	11
2.1.2 常量	12
2.2 数据类型及变量	15
2.2.1 基本数据类型	15
2.2.2 变量的定义	16
2.2.3 变量的初始化	16
2.2.4 数据类型转换	17
2.3 运算符和表达式	19
2.3.1 运算符和表达式概述	19
2.3.2 赋值运算符和赋值表达式	20
2.3.3 算术运算符和算术表达式	21
2.3.4 关系运算符和关系表达式	23
2.3.5 逻辑运算符和逻辑表达式	23
2.3.6 条件运算符	24
2.3.7 其他运算符	25

2.4 位运算符.....	26
2.4.1 按位取反运算符	26
2.4.2 移位运算符	27
2.4.3 按位“与”、按位“或”、按位“异或”	27
2.5 C语言基本输入输出函数	29
2.5.1 字符输入输出函数	29
2.5.2 格式化输入输出函数	30
习题	38
第3章 C语言程序基本控制结构及其应用	41
3.1 算法及结构化程序设计.....	41
3.1.1 算法及其特征	41
3.1.2 算法的类型与结构	43
3.2 顺序结构程序设计.....	44
3.3 分支结构程序设计.....	46
3.3.1 if分支	46
3.3.2 if-else分支	46
3.3.3 多分支 if...else if...else 形式	47
3.3.4 条件分支的嵌套	48
3.3.5 开关分支	50
3.4 循环结构程序设计.....	54
3.4.1 while语句	54
3.4.2 do-while语句	54
3.4.3 for语句.....	55
3.4.4 三种循环的比较	57
3.4.5 多重循环	57
3.4.6 循环和开关分支的中途退出	59
3.4.7 goto语句	61
3.4.8 结构化程序设计注意事项	62
3.5 结构化程序应用举例.....	64
习题	68
第4章 数组及其应用	74
4.1 一维数组.....	74
4.1.1 一维数组的定义	74
4.1.2 一维数组的存储形式	75
4.1.3 一维数组的初始化	75
4.1.4 一维数组的引用	76
4.1.5 一维数组的应用举例	77

4.2 多维数组	79
4.2.1 多维数组的定义	79
4.2.2 多维数组的存储形式	79
4.2.3 多维数组的引用	80
4.2.4 多维数组的初始化	81
4.2.5 多维数组的应用举例	82
4.3 字符型数组与字符串	83
4.3.1 字符型数组的概念	83
4.3.2 字符型数组的初始化	84
4.3.3 字符型数组的输入输出	85
4.3.4 字符型数组的应用举例	86
4.4 综合应用举例	87
习题	90
第5章 函数及其应用	97
5.1 函数的定义与调用	97
5.1.1 C源程序的结构	97
5.1.2 函数的定义	98
5.1.3 函数的调用	100
5.2 函数间的信息传递方式	104
5.2.1 实参-形参之间的信息传递	105
5.2.2 函数调用结果的返回	109
5.3 函数与数组	111
5.3.1 数组元素作实参	111
5.3.2 一维数组名作实参	112
5.3.3 多维数组名作参数	113
5.3.4 字符数组作函数的参数	115
5.4 递归函数与递归调用	116
5.4.1 递归的概念	116
5.4.2 递归程序的设计	117
5.5 变量的存储类型及作用域	118
5.5.1 auto 型变量	119
5.5.2 extern 型变量	119
5.5.3 register 型变量	122
5.5.4 静态变量	123
习题	126
第6章 指针	130
6.1 指针的基本概念及定义方式	130

6.1.1 指针的基本概念	130
6.1.2 指针的定义	131
6.1.3 指针的初始化	131
6.1.4 指针运算符	133
6.2 指针的运算	133
6.2.1 指针的算术运算	133
6.2.2 关系运算	135
6.2.3 指针的赋值运算	135
6.3 指针与数组	136
6.3.1 指向一维数组的指针	136
6.3.2 指向多维数组的指针	138
6.4 字符指针和字符串	140
6.5 指针数组	142
6.5.1 指针数组的概念	142
6.5.2 指针数组的应用	143
6.5.3 指针数组在带形参的 main 函数中的应用	145
6.6 指针在函数参数传递中的应用	147
6.7 指针型函数	148
6.7.1 指针型函数的定义和引用	148
6.7.2 指针型函数的应用举例	148
6.8 多级指针	150
6.8.1 多级指针的概念及定义	150
6.8.2 多级指针应用举例	151
6.9 指向函数的指针	152
6.9.1 指向函数的指针的概念	152
6.9.2 指向函数的指针的应用	153
6.10 动态指针	155
6.10.1 动态内存分配的概念	155
6.10.2 动态内存分配的应用	158
习题	160
* 第 7 章 图形设计与应用	165
7.1 用 C 语言图形库函数绘图	165
7.1.1 显示方式及图形方式的初始化	165
7.1.2 常用图形函数	169
7.2 复杂图形的绘制和图形程序设计方法	179
7.2.1 复杂图形的绘制	179
7.2.2 图形程序设计中的递归算法及其应用	184
习题	186

第 8 章 结构体、联合体和枚举	187
8.1 结构体的说明和定义	187
8.1.1 什么叫结构体	187
8.1.2 结构体的说明及结构体变量的定义	187
8.2 结构体成员的引用与结构体变量的初始化	190
8.2.1 结构体成员的引用	190
8.2.2 结构体变量的初始化	191
8.3 结构体数组	192
8.3.1 结构体数组的定义及初始化	192
8.3.2 结构体数组的应用举例	193
8.4 结构体指针	195
8.4.1 结构体指针及其定义	195
8.4.2 通过指针引用结构体成员	196
8.4.3 结构体指针的应用举例	196
8.5 结构体在函数间的传递	199
8.5.1 结构体变量的传递	199
8.5.2 结构体数组在函数间的传递	202
8.6 结构体指针型和结构体型函数	204
8.6.1 结构体指针型函数	204
8.6.2 结构体型函数	206
8.7 结构体嵌套	207
8.7.1 什么是结构体嵌套	207
8.7.2 嵌套结构体类型变量的引用	208
8.7.3 结构体嵌套应用举例	210
8.8 联合体	211
8.8.1 联合体的说明及联合体变量的定义	211
8.8.2 使用联合体变量应注意的问题	215
8.9 枚举类型	217
8.9.1 枚举类型数据的概念及其定义	217
8.9.2 枚举型变量的使用	217
8.10 自定义类型	219
8.10.1 自定义类型及其表示形式	219
8.10.2 自定义类型的优点	220
8.11 位字段结构体	221
习题	221
* 第 9 章 排序及查找算法及其实现	226
9.1 排序概述	226
9.1.1 排序的概念	226

9.1.2	排序的定义	226
9.1.3	排序的方法	227
9.1.4	排序效率	227
9.2	冒泡排序法的设计及其实现	229
9.2.1	冒泡算法设计思想	229
9.2.2	冒泡算法的实现	230
9.3	选择排序法的设计及其实现	231
9.3.1	选择排序法设计思想	231
9.3.2	选择排序法的实现	232
9.4	插入排序法的设计及其实现	233
9.4.1	插入排序法设计思想	233
9.4.2	插入排序法的实现	234
9.5	SHELL 排序法的设计及其实现	235
9.5.1	SHELL 排序法设计思想	235
9.5.2	SHELL 排序法的实现	237
9.6	字符串数组的排序设计及其实现	238
9.6.1	字符串数组的排序算法设计思想	238
9.6.2	字符串数组排序算法的实现	239
9.7	查找概述	240
9.8	顺序查找及其应用	241
9.8.1	顺序查找算法的设计思想	241
9.8.2	顺序查找算法的实现	241
9.9	折半查找及其应用	242
9.9.1	折半查找算法的设计思想	242
9.9.2	折半查找算法的实现	243
	习题	244
第 10 章	文件操作	245
10.1	文件的概念	245
10.2	文件的基本操作	248
10.2.1	文件的打开	248
10.2.2	文件的关闭	249
10.2.3	文件的删除	250
10.2.4	文件的重命名	250
10.2.5	临时文件的创建	250
10.3	文件的读写操作	251
10.3.1	文件的非格式化读写	251
10.3.2	文件的格式化写操作	257
10.3.3	文件的格式化读操作	259

10.4 文件的定位.....	260
10.4.1 fgetpos 函数	260
10.4.2 fsetpos 函数	260
10.4.3 ftell 函数.....	261
10.4.4 fseek 函数	261
10.4.5 feof 函数.....	262
10.5 错误处理.....	262
10.5.1 ferror 函数	263
10.5.2 perror 函数.....	263
习题.....	263
* 第 11 章 链表及其应用	269
11.1 链表的基本概念.....	270
11.1.1 链表的定义及基本结构.....	270
11.1.2 动态内存分配.....	271
11.2 链表的基本操作方法.....	271
11.2.1 链表的建立.....	271
11.2.2 链表结点的访问.....	273
11.2.3 同结构链表的连接.....	274
11.2.4 链表结点的插入.....	274
11.2.5 结点的删除.....	277
11.2.6 释放链表存储空间.....	279
11.3 循环链表.....	280
11.4 双向链表.....	280
11.5 链表的应用.....	281
习题.....	284
* 第 12 章 栈及其应用	287
12.1 栈的定义及其基本操作.....	287
12.2 栈的实现.....	288
12.2.1 栈的数组实现.....	288
12.2.2 栈的链式存储实现.....	290
12.3 进制转换——栈的应用实例.....	292
12.3.1 进制转换的算法.....	292
12.3.2 算法的实现.....	293
习题.....	295
* 第 13 章 队列及其应用	296
13.1 队列的定义及基本操作.....	296
13.2 队列的实现.....	296

13.2.1 用循环数组实现队列·····	297
13.2.2 用线性表实现队列操作·····	301
13.3 队列的应用·····	303
习题·····	306
* 第 14 章 二叉树及其应用 ·····	307
14.1 树的概念·····	307
14.2 关于树的一些术语及特性·····	308
14.3 二叉树的特点与数学性质·····	309
14.3.1 二叉树的特点·····	309
14.3.2 两种特殊形态的二叉树·····	310
14.3.3 二叉树的数学性质·····	311
14.4 二叉树的基本操作及其实现·····	312
14.4.1 二叉树的基本操作·····	312
14.4.2 二叉树基本操作的实现·····	312
14.5 二叉树的应用·····	317
习题·····	320
附录 A 预处理命令的应用 ·····	322
附录 B Turbo C 3.0 的上机过程 ·····	333
附录 C Turbo C 3.0 程序的调试 ·····	338
附录 D ASCII 码表 ·····	346
参考文献 ·····	347

第 1 章

C 语言的基本概念

1.1 C 语言的发展与特点

1.1.1 C 语言的发展

C 语言是目前国际上广泛流行的一种结构化的程序设计语言,它不仅是开发系统软件的很好的工具,而且也是开发应用程序的很好的程序设计语言。因此,它深受广大程序设计者的欢迎。

C 语言是在 20 世纪 70 年代初由美国贝尔实验室的 Dennis M. Ritchie 设计的,并首先在安装 UNIX 操作系统的 DEC PDP-11 计算机上实现,因此说最初的 C 语言是为描述和实现 UNIX 操作系统提供一种工作语言而设计的。到了 1973 年,K. Thompson 和 Dennis M. Ritchie 两个人合作把 UNIX 的 90% 以上内容用 C 语言进行了改写,即大家熟知的 UNIX 第五版。多年来,UNIX V 系统配备的 C 语言一直是 C 的公认标准,在 Brian W. Kernighan 和 Dennis M. Ritchie 合著的《C Programming Language》中对此也进行了介绍。因此,当初的 C 语言是为开发 UNIX 操作系统而研制的,它随着 UNIX 的出名而闻名。随着微型计算机的普及,出现了大批 C 语言系统,其中多数系统接受的源程序都能高度兼容,然而,没有统一的标准,再怎么兼容,总存在一定的差异,这种差异对于计算机应用技术的发展,显然不利。为了克服此不利局面,ANSI 于 1983 年成立了专门定义 C 语言标准的委员会,花了 6 年时间使 C 语言迈向了标准化。随着 C 语言的广泛应用又不断推出新的 C 语言版本,其性能也越来越强。到了 1975 年,随着 UNIX 第六版的推出和面向对象程序设计技术的出现,C 语言的突出优点引起了人们的普遍关注,ANSI C 标准于 1989 年被采用,该标准一般称为 ANSI/ISO Standard C,于是,1989 年定义的 C 标准定义为 C89,到了 1995 年,出现了 C 语言的修订版,其中增加了一些库函数,出现了初步的 C++,在此基础上,C89 成为 C++ 的子集。此后,C 语言不断发展,在 1999 年又推出了 C99,C99 在基本保留了 C 语言特性的基础上增加了一系列新的特性,随后又几经修改和完善,它也从面向过程的编程语言发展到面向对象的程序设计语言,目前可在微机上运行的 C 语言版本主要有 Microsoft C/C++, Turbo C, Quick C, Visual C/C++ 等版本。

1.1.2 C 语言的特点

C 语言之所以能被推广并被广泛使用,概括地说主要有如下特点:

(1) 中级语言

C 语言通常被称为中级语言,这并不意味着 C 语言的功能不如高级语言,而是因为它把高级语言的先进思想与汇编语言的控制和灵活性有机地结合了起来。作为中级语言, C 语言允许对位、字节和地址这些计算机功能中的基本成分进行操作。

(2) 结构化语言

C 语言是便于进行模块化程序设计的语言, C 语言程序是由一系列函数所组成,这种结构便于把一个大型程序划分为若干相对独立的模块,模块间通过函数调用来实现相互连接,函数允许一个程序中的各个任务被分别定义和编码,从而使程序模块化。一个设计良好的函数可以在各种情况下正常工作,不应该对程序的其他部分产生不良影响。因此,对于大型程序的设计来说,函数的设计至关重要,因为函数是 C 语言的独立的子程序,是一种构件,程序的所有操作都在其中发生,能够设计出独立的函数在大型的项目中是至关重要的。在这个项目中,一个程序员的代码不得意外地影响他人的程序。

C 语言能够把执行某个特殊任务所需要的指令和数据从程序的其余部分分离出去,隐藏起来,实现代码和数据的封装。结构化语言还提供了大量的程序设计功能,直接支持顺序、分支和循环三种典型的基本结构(这三种结构将在第 3 章进行详细介绍),使程序设计人员便于使用“自顶向下逐步求精”的结构化程序设计技术。

(3) 可移植性

C 语言程序具有较高的移植性,在这一点上 C 语言不同于 FORTRAN 等程序设计语言。可移植性指的是,可以把为某种计算机编写的软件运行在另一种机器或操作系统上,如在 DOS 下写的程序,能够方便地在 Windows 2000 下运行,这个程序就是一个可移植的程序。C 语言不包含依赖硬件的输入输出机制,其输入输出功能是由独立于 C 语言的库函数来实现。这样就使 C 语言程序本身不依赖于硬件系统,也便于在不同的机器系统间移植。

1.2 几个基本概念

在介绍 C 语言程序设计之前,我们先来认识几个有关程序设计的基本概念。

1.2.1 程序

所谓程序,就是一系列遵循一定规则和思想并能正确完成指定工作的代码(也称为指令序列)。通常,一个计算机程序主要描述两部分的内容,其一是描述问题的每个对象及他们之间的关系,其二是描述对这些对象进行处理的处理规则。其中关于对象及它们之间的关系涉及到数据结构的内容,而处理规则却是求解某个问题的算法。因此,对程序的描述,经常有如下等式:

$$\text{程序} = \text{数据结构} + \text{算法}$$

一个设计合理的数据结构往往可以简化算法,而且一个好的程序有可靠性、易读性、可维护性等良好特性。