

21世纪高等学校计算机基础教育系列教材

21 SHIJI GAODENG XUEXIAO JISUANJI JICHU JIAOYU XILIE JIAOCAI

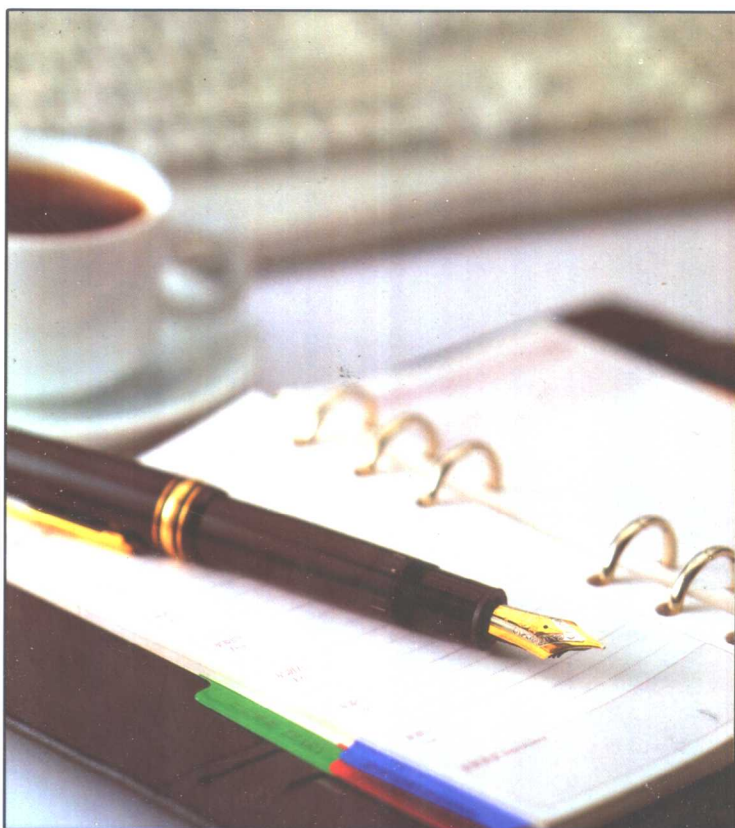
C程序设计

实用教程

■ 罗朝盛 主编
余文芳 副主编

本书提供:

- ※ 配套实验系统
- ※ 电子教案
- ※ 例题源程序
- ※ 习题答案



人民邮电出版社
POSTS & TELECOM PRESS

21 世纪高等学校计算机基础教育系列教材

C 程序设计实用教程

罗朝盛 主编

余文芳 副主编

人民邮电出版社

图书在版编目 (CIP) 数据

C 程序设计实用教程 / 罗朝盛主编. —北京: 人民邮电出版社, 2005.1
(21 世纪高等学校计算机基础教育系列教材)

ISBN 7-115-12810-3

I. C... II. 罗... 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 (2004) 第 131238 号

内容简介

全书共分 11 章, 包括 C 语言程序设计概述、数据类型、运算符和表达式、3 种基本控制结构的程序设计、数组、函数、编译预处理、指针、文件操作、屏幕控制与绘图及 C++ 基础等内容。本书内容丰富、概念清楚、层次分明、通俗易懂, 并配有大量的例题, 适合教师课堂教学和学生自学。另外, 每一章末都附有精选的多种类型的习题, 有助读者复习, 巩固所学知识, 以培养读者的实际编程能力。

本书配有以任务驱动的“C 程序设计实验 CAI 系统”。该系统利用计算机的特点对 C 程序设计实验进行整合。使用本 CAI 系统进行上机实验, 学生上机实验目的明确, 可大大改善实验效果, 减轻教师指导学生实验的工作量。若选用本书作教材, 可获得用于课堂教学的电子课件和实验 CAI 系统网络版。

本书配套的“C 程序设计实验 CAI 系统 (单机版)”、书中例题源程序和习题参考答案, 可直接到作者个人网站 (www.csluo.com) 下载。

本书可以作为各类高等院校、各类高职院校计算机专业学生及理工科非计算机专业学生学习的“C 程序设计”课程的教材, 也可作为广大计算机爱好者学习 C 程序设计语言的参考书。

21 世纪高等学校计算机基础教育系列教材

C 程序设计实用教程

◆ 主 编 罗朝盛

副 主 编 余文芳

责任编辑 邹文波

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

读者热线 010-67129259

北京隆昌伟业印刷有限公司印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 20.5

字数: 495 千字

2005 年 1 月第 1 版

印数: 1-5 000 册

2005 年 1 月北京第 1 次印刷

ISBN 7-115-12810-3/TP · 4297

定价: 28.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

前 言

C 语言是目前世界上最流行、使用最广泛的高级程序设计语言之一。它不仅具有丰富灵活的控制和数据结构、简洁而高效的表达式语句、清晰的程序结构和良好的可移植性等优点,还具有直接操纵计算机硬件的强大功能。因此,C 语言既适合于开发系统程序,又适合于开发应用程序,深受广大计算机应用人员的青睐。目前,高校的计算机专业、绝大多数理工科院校非计算机专业都将 C 语言作为计算机程序设计课程的首选语言。

尽管目前已出版的 C 程序设计书籍很多,其中也包括一些著名的教材,但通过我们多年的教学实践来看,真正适合于教学和计算机爱好者自学的教材并不多。现有的教材一般围绕语言的自身体系,比较注重语法知识讲解,辅以一些编程技巧的介绍,而在教授学生编程方法和训练学生编程能力方面不够突出。

学习程序设计既要熟练地掌握和使用编程语言,又要掌握程序设计方法,后者是程序设计的难点,也是重点,因为程序设计是一项严密的逻辑思维活动,可以完全独立于具体的编程语言,不受它所依托的具体语言的限制。它涉及到很多认识上的技巧,例如,对操作环境和相关开发工具的熟悉,对数据结构、算法的合理运用和测试,对机器内部工作的了解等。因此,如果在介绍程序设计语言时只侧重语法知识介绍就会给读者一个误导,使读者误以为学习程序设计就是记住那些语法规则而已,而忽略了对程序设计方法的训练以及培养良好的程序设计习惯,导致学习完 C 程序设计课程后,仅仅学会了一些 C 语言的语法,而编写不出简单的应用程序,读懂别人编写的简单程序也存在困难。

笔者根据多年在高校从事计算机程序设计语言教学的经验,编写了本教材,与现有的 C 语言教材相比有以下的特点。

(1) 该书吸取了其他众多同类 C 语言教材的优点,章节安排由浅入深、循序渐进。重视编程能力的培养,围绕着结构化与模块化程序设计这个中心,以程序设计为主线,在深入浅出地讲解 C 语言基本语法规则的同时,通过精心设计的例题,着重介绍了 C 语言程序设计的基本方法,加强了结构化程序设计和常用算法的训练。这样可使读者既能掌握 C 语言基础知识,又能掌握程序设计的基本方法。

(2) 突出应用与适用性。本书以简短的篇幅介绍 C 语言中最基本、最常用的内容,同时精心设计了一些 C 语言的编程实例(如分类统计、排序、哥德巴赫猜想、查找、插入、删除、单词数统计、字符加密解密和文件加密保护等),对所讲述的原理、概念加以辅助说明,学生可以通过这些实例加深对 C 语言编程的基本原理、方法的掌握与理解。通过实例分析,并加以编程实现,使学生既掌握了对 C 语言内容的学习,进行了开发实用软件的训练,更激发了探索 C 语言奥妙的兴趣,能达到事半功倍的效果。

(3) 注重改革实践教学。C 语言程序设计是一门实践性很强的课程,上机实验是该课程的重要教学内容,学习程序设计是听不会,看不会的,只有通过大量的编程练习,在实践中掌握语言知识,培养程序设计实践能力。因此本教材专门针对这一特点,开发了一套采用“任务驱动”的 C 语言程序实验 CAI 系统,该系统的每个实验任务都是精心设计的,针对

性强,学生可在实验任务的驱动下进行操作,并能将完成结果与系统要求达到的结果进行比较,从而训练、提高学生的实际编程能力。

(4) 多元化“立体化”教材建设。本书为了方便教学与学习,不仅提供了教师使用的课堂教学课件,还配套了采用“任务驱动”的实验 CAI 系统。另外,作者建立的自己个人教学网站(www.csluo.com),为广大读者提供了一个良好的学习平台,其中提供教材的例题源程序、习题及相关辅导资料的下载等。

(5) 为开阔学生的视野,本书还介绍了屏幕控制与绘图(第 10 章)和 C++基础(第 11 章),这部分内容是为学生应用 C 语言编写适用程序和进一步学习 C++或 VC++打下基础。这些内容为选学内容,教师可根据实际情况选讲或指导学生自学。

另外,每一章末附有精心挑选和设计的多种类型的习题,有助读者复习、巩固所学知识。

本书由在教学第一线并具有丰富计算机程序设计教学经验的多位教师共同编写。其中第 1~3 章及附录由罗朝盛编写,第 4 章由余文芳编写,第 5 章、第 7 章由魏英编写,第 6 章、第 9 章由马杨琿编写,第 8 章由郭招娣编写,第 10 章、第 11 章由孙奕鸣编写。本书由浙江科技学院罗朝盛教授任主编,浙江工商大学余文芳副教授任副主编。

本书在编写过程中得到了浙江大学余瑞钊教授、杭州电子工业学院的胡维华教授等人的帮助和支持,他们对本书提出了不少有益的建议。另外,本书还得到了浙江科技学院教育信息技术学系计算机基础教研室全体教师的支持,浙江科技学院信电系 2000 级电气工程专业的余匡辉、孙燕同学参与实验 CAI 系统的开发工作,在此表示衷心的感谢。

本书可作为各类高等院校计算机专业及理工科类非计算机专业学生学习 C 程序设计的教材,还可作为有关工程技术人员和计算机爱好者学习 C 程序设计的参考书。

本书虽经多次讨论并反复修改,但限于作者水平,不当之处仍在所难免,谨请广大读者指正。

编著者

2005 年 1 月

目 录

第 1 章 C 程序设计概述	1
1.1 计算机程序设计概述	1
1.1.1 程序设计语言的发展	1
1.1.2 程序设计方法概述	3
1.2 C 语言简介	5
1.2.1 C 语言的发展	5
1.2.2 C 语言的特点	5
1.3 C 语言的字符集与标识符	6
1.3.1 C 语言的字符集	6
1.3.2 C 语言标识符	7
1.4 C 语言的程序结构	8
1.4.1 几个简单的 C 程序实例	8
1.4.2 C 程序的结构特点	10
1.4.3 C 函数的结构	11
1.4.4 C 程序的书写风格	11
1.5 运行一个 C 程序	11
1.5.1 C 程序运行的一般步骤	11
1.5.2 在 TC 中运行一个 C 程序	12
1.6 小结	15
习题	15
第 2 章 数据类型、运算符和表达式	18
2.1 C 语言的数据类型	18
2.1.1 C 语言数据类型概述	18
2.1.2 基本数据类型	19
2.2 常量	20
2.2.1 数值常量	20
2.2.2 字符常量	21
2.2.3 字符串常量	21
2.2.4 转义字符	21
2.2.5 符号常量	22
2.3 变量	23
2.3.1 变量的定义	23

2.3.2 变量的赋初值	24
2.4 运算符与表达式	25
2.4.1 C 语言中的运算符	25
2.4.2 运算符的优先级和结合性	25
2.4.3 算术运算符与算术表达式	26
2.4.4 关系运算符与关系表达式	28
2.4.5 逻辑运算符与逻辑表达式	28
2.4.6 赋值运算符和赋值表达式	30
2.4.7 位运算符	31
2.4.8 逗号运算符和逗号表达式	34
2.4.9 表达式中的类型转换	34
2.5 常用库函数	35
2.6 小结	36
习题	37
第 3 章 3 种基本结构程序设计	39
3.1 算法及算法的表示	39
3.1.1 算法概述	39
3.1.2 算法的特性	40
3.1.3 算法的表示	41
3.1.4 用 N-S 流程图表示算法	43
3.2 顺序结构程序设计	45
3.2.1 赋值语句	45
3.2.2 数据格式输入——scanf()函数	46
3.2.3 数据格式输出——printf()函数	48
3.2.4 字符数据非格式输入/输出函数	50
3.2.5 顺序结构程序举例	52
3.3 选择结构	53
3.3.1 if 条件语句	53
3.3.2 条件运算符和条件表达式	58
3.3.3 switch 语句(开关语句)	59
3.3.4 选择结构程序举例	60
3.4 循环结构	61
3.4.1 while 语句	62
3.4.2 do...while 语句	63
3.4.3 for 循环语句	65
3.4.4 循环的嵌套——多重循环结构	67
3.4.5 3 种循环语句比较	68
3.5 其他控制语句	68

3.5.1 break 语句	68
3.5.2 continue 语句	69
3.5.3 goto 语句	69
3.6 应用程序举例	70
3.6.1 素数与哥德巴赫猜想	70
3.6.2 穷举法	72
3.6.3 迭代法	73
3.7 小结	74
习题	75
第4章 数组	82
4.1 概述	82
4.2 一维数组	83
4.2.1 一维数组的声明	83
4.2.2 一维数组的初始化	84
4.2.3 一维数组元素的引用	85
4.2.4 一维数组的基本操作	85
4.2.5 一维数组的应用举例	87
4.3 二维数组与多维数组	88
4.3.1 二维数组的声明	88
4.3.2 二维数组元素的引用	89
4.3.3 二维数组的初始化	91
4.3.4 二维数组的基本操作	92
4.3.5 二维数组应用举例	95
4.3.6 多维数组的声明和引用	96
4.4 字符数组与字符串	96
4.4.1 字符数组与初值化	96
4.4.2 字符数组的引用	97
4.4.3 字符串与字符数组	98
4.4.4 字符数组的输入/输出	99
4.4.5 字符串处理函数	100
4.5 应用程序举例	104
4.5.1 排序问题	104
4.5.2 数据查找	106
4.5.3 插入法	108
4.5.4 字符串的处理	109
4.6 小结	112
习题	113

第 5 章 函数	119
5.1 函数概述	119
5.1.1 模块化程序设计方法	119
5.1.2 C 函数的分类	120
5.2 函数的定义	121
5.2.1 函数的定义形式	121
5.2.2 函数返回值	122
5.2.3 形式参数的设计	123
5.2.4 函数原型	124
5.3 函数调用与参数传递	125
5.3.1 函数的调用方式	125
5.3.2 参数传递	126
5.3.3 数组参数	127
5.3.4 函数的嵌套调用	131
5.3.5 函数的递归调用	132
5.4 函数与变量的作用域	133
5.4.1 局部变量与全局变量	133
5.4.2 动态存储变量与静态存储变量	139
5.4.3 内部函数与外部函数	141
5.5 程序应用举例	142
5.6 小结	146
习题	147
第 6 章 编译预处理	151
6.1 宏定义#define	151
6.1.1 无参宏定义	151
6.1.2 有参宏定义	153
6.2 文件包含命令 #include	156
6.3 条件编译命令	157
6.4 小结	160
习题	160
第 7 章 指针	163
7.1 指针概述	163
7.1.1 地址的概念	163
7.1.2 变量的存储与访问	164
7.2 指针变量	165
7.2.1 指针变量的定义	165

7.2.2 指针变量的访问	165
7.2.3 指针变量的运算	167
7.2.4 指针变量的初始化	168
7.2.5 指针变量做参数	168
7.3 数组与指针	171
7.3.1 数组的存储与访问	171
7.3.2 指向一维数组的指针	173
7.3.3 指向多维数组的指针	175
7.3.4 指向数组的指针做参数	178
7.4 字符串与指针	180
7.4.1 字符串的存储与访问	180
7.4.2 指向字符串的指针	181
7.4.3 指向字符串的指针做参数	184
7.5 指针数组	186
7.6 指向指针的指针	187
7.7 main()函数的参数	189
7.8 函数与指针	190
7.8.1 返回指针值的函数	190
7.8.2 指向函数的指针变量	191
7.9 小结	194
习题	194
第8章 结构体、共用体与枚举类型	200
8.1 结构体类型与结构体变量	200
8.1.1 什么是结构体类型	200
8.1.2 结构体的声明	201
8.1.3 结构体变量的定义	202
8.1.4 结构体变量的引用	203
8.1.5 结构体变量的初始化	205
8.1.6 指向结构体的指针	206
8.2 结构体数组	207
8.2.1 结构体数组的定义	207
8.2.2 结构体数组的初始化	208
8.2.3 结构体数组与指针	208
8.2.4 结构体数组的应用实例	209
8.3 结构体与函数	210
8.3.1 结构体变量做函数参数	210
8.3.2 结构体数组做函数参数	211
8.3.3 返回结构体的函数	212

8.4 动态数据结构——链表	213
8.4.1 问题的提出	213
8.4.2 链表的基本结构	213
8.4.3 链表结点的定义	214
8.4.4 单向链表的访问	214
8.4.5 动态存储空间的建立和释放	215
8.4.6 动态链表的建立	216
8.4.7 链表的删除操作	218
8.4.8 链表的插入操作	220
8.5 共用体	223
8.6 枚举类型	225
8.7 自定义类型标识符 (typedef)	227
8.8 小结	227
习题	228
第9章 文件操作	237
9.1 文件系统的概念	237
9.1.1 C 文件概述	237
9.1.2 缓冲文件系统	238
9.1.3 非缓冲文件系统	238
9.2 文件的打开与关闭	238
9.2.1 文件类型指针	239
9.2.2 文件的打开	239
9.2.3 文件的关闭 (fclose()函数)	241
9.3 文件的读写	241
9.3.1 文件的写函数	242
9.3.2 文件的读函数	246
9.3.3 随机读写文件	251
*9.4 文件操作函数	252
9.5 应用举例	254
9.5.1 文件的加密和解密	254
9.5.2 文件的连接	256
9.6 小结	257
习题	257
*第10章 屏幕控制与绘图	262
10.1 文本模式	262
10.1.1 文本窗口 (字符屏幕) 的创建	262
10.1.2 文本窗口屏幕处理函数	263

10.2 图形模式.....	264
10.2.1 设置图形模式.....	264
10.2.2 图形窗口.....	267
10.3 图形函数及其使用.....	269
10.3.1 颜色函数.....	269
10.3.2 屏幕坐标位置的函数.....	270
10.3.3 基本图形函数.....	271
10.4 小结.....	278
习题.....	278
第 11 章 C++简介.....	280
11.1 面向对象的概念.....	280
11.2 C++的输入与输出.....	282
11.2.1 使用 cout 进行输出.....	282
11.2.2 使用 cin 进行输入.....	283
11.3 函数重载.....	284
11.4 类与对象.....	285
11.4.1 类与对象的概念.....	285
11.4.2 对象的创建.....	287
11.4.3 构造函数.....	289
11.4.4 析构函数.....	291
11.5 继承与派生.....	292
11.5.1 继承的基本概念.....	292
11.5.2 派生类.....	293
11.6 小结.....	296
习题.....	299
附录 1 ASCII 字符集.....	301
附录 2 运算符的优先级和结合性.....	302
附录 3 标准 C 语言系统的库函数.....	303
附录 4 C 程序设计实验 CAI 系统.....	308
参考文献.....	315

第 1 章

C 程序设计概述

本章介绍计算机程序设计的概念, 计算机语言的发展, C 语言的特点, C 程序设计的一些基本知识和编制开发计算机程序的基本步骤。

本章主要任务:

- (1) 了解计算机程序设计语言及程序设计方法的发展, 用高级语言编写的程序的执行过程;
- (2) 了解 C 语言的特点, 正确使用 C 语言中的标识符;
- (3) 掌握 C 程序的基本组成结构;
- (4) 掌握开发和运行一个 C 程序的过程。

1.1 计算机程序设计概述

1.1.1 程序设计语言的发展

计算机系统是由硬件系统和软件系统两大部分构成的, 硬件是物质基础, 而软件可以说是计算机的灵魂, 没有软件, 计算机只是一台“裸机”, 是什么也不能干的, 有了软件, 才能成为一台真正的“电脑”。所有的软件, 都是用计算机语言编写的。

人和计算机交流信息使用的语言称为计算机语言或程序设计语言。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 (Machine Language)

机器语言是一种用二进制代码“0”和“1”形式表示的, 能被计算机直接识别和执行的语言。用机器语言编写的程序, 称为计算机机器语言程序。机器语言程序不便于记忆、阅读和书写。但由于使用的是针对特定型号计算机的语言, 所以运算效率是所有语言中最高的。机器语言, 是第一代计算机程序设计语言。

2. 汇编语言 (Assemble Language)

为了克服使用机器语言编程的难记、难读等缺点, 人们进行了有益的

改进，用一些简洁的英文字母、符号串（称为注记符）来替代一个特定的指令的二进制串，例如，用“ADD”代表加法，“MOV”代表数据传递等，这样就很容易读懂和理解程序在干什么，程序维护也就变得方便了，这种程序设计语言就称为汇编语言。用汇编语言编制的程序称为汇编语言程序，机器不能直接识别和执行，必须由“汇编程序”（或汇编系统）将这些符号翻译成二进制数的机器语言才能运行。这种“汇编程序”就是汇编语言的翻译程序。汇编语言程序的执行过程如图 1-1 所示。

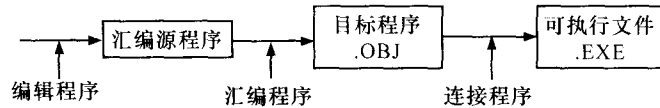


图 1-1 汇编语言程序的执行过程

汇编语言属于第二代计算机程序设计语言。汇编语言适用于编写直接控制机器操作的底层程序，它与机器密切相关，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，程序精炼而质量高，所以至今仍是一种常用而强有力的软件开发工具。

汇编语言和机器语言都是面向机器的程序设计语言，一般称为低级语言。

3. 高级语言（High Level Language）

高级语言是一种与硬件结构及指令系统无关，表达方式比较接近自然语言和数学表达式的计算机程序设计语言。其优点是：描述问题能力强，通用性、可读性及可维护性都较好。其缺点是：执行速度较慢，编制访问硬件资源的系统软件较难。

高级语言的发明是计算机发展史上最惊人的成就。1954 年，第一个完全脱离机器硬件的高级语言——FORTRAN 问世，五十多年来，共有几百种高级语言出现，目前在计算机中广泛使用的高级语言有几十种，影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、LISP、SNOBOL、PL/1、Pascal、C、PROLOG、Ada、C++、Visual C++、Visual Basic、Delphi 和 Java 等。

（1）BASIC 语言是一种简单易学的计算机高级语言。尤其是 Visual Basic 语言，具有很强的可视化设计功能，给用户在 Windows 环境下开发软件带来了方便，是重要的多媒体编程工具语言。

（2）FORTRAN 语言是一种适合科学和工程设计计算的语言，它具有大量的工程设计计算程序库。

（3）Pascal 语言是结构化程序设计语言，适用于教学、科学计算、数据处理和系统软件的开发。

（4）C 语言是一种具有很高灵活性的高级语言，适用于系统软件、数值计算和数据处理等，使用非常广泛。面向对象的 C++ 及 Visual C++，Borland C++ 等集成开发工具很受程序开发者的青睐。

（5）Java 语言是近几年发展起来的一种新型的高级语言，它简单、安全、可移植性强。Java 语言适用于网络环境的编程，多用于交互式多媒体应用。

用高级语言编写的程序称为“源程序”，计算机不能识别和执行，要把用高级语言编写

的源程序翻译成机器指令，通常有编译和解释两种方式。

编译方式是将源程序整个编译成等价的、独立的目标程序，然后通过连接程序将目标程序连接成可执行程序。执行过程如图 1-2 所示。

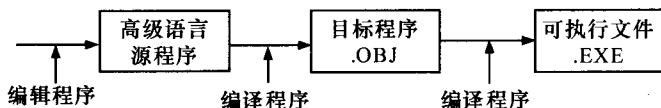


图 1-2 高级语言编译执行方式

解释方式是将源程序逐句翻译，翻译一句执行一句，边翻译边执行，不产生目标程序。整个执行过程，解释程序都一直在内存中，如 Basic 语言。解释方式执行过程如图 1-3 所示。

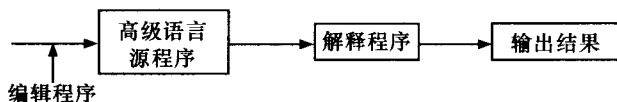


图 1-3 高级语言解释执行方式

目前，一般的高级语言都提供了集成开发环境，它集源程序编辑、编译（解释）和执行为一体，非常方便用户使用，如 Turbo C、Visual Basic、Visual C++ 和 Delphi 等。

1.1.2 程序设计方法概述

程序设计方法的发展可以划分以下 3 个阶段。

1. 早期的程序设计

计算机发展的初期，由于 CPU 运行速度慢、内存容量小，因此衡量程序质量优劣的标准是占用内存的大小和运行时间的长短，这就导致了程序设计人员不得不把大量的精力耗费在程序设计技巧上。反映在程序结构上，对程序的流程没有严格的限制，程序员可以随心所欲地使流程转来转去，程序流程变得毫无规律，读者要花费很大的精力去追踪流程，使得程序很难修改和维护。

随着计算机技术的发展，特别是到了 20 世纪 50 年代中期到 60 年代，晶体管和集成电路相继问世，使得计算机的性价比不断提高，各种程序设计语言不断推出。另一方面由于计算机应用领域的不断扩大，软件的规模也越来越大，软件产品供不应求。传统方法研制软件的时间长、成本高、可靠性低、难以修改和不易维护等问题日益突出，这就造成了当时出现的“软件危机”。软件危机引起了人们的高度注意，不少计算机专家开始着手研究探讨产生软件危机的原因以及解决软件危机的途径，于是也就出现了对软件工程、软件管理、软件可靠性及程序设计方法（设计、编制、调试及维护程序的方法）等问题的研究。

2. 结构化程序设计

20 世纪 60 年代末，著名学者 E.W.Dijkstra 首先提出了“结构化程序设计”的思想。这种方法要求程序设计者按照一定的结构形式来设计和编写程序，使程序易阅读、易理解、易修改和易维护。这种结构形式主要包括以下两方面的内容。

(1) 在程序设计中, 采用自上向下、逐步细化的原则。

按照这个原则, 整个程序设计过程应分成若干层次, 逐步加以解决: 每一步是在前一步的基础上, 对前一步设计的细化。这样, 一个较复杂的大问题, 就被层层分解成为多个相对独立的、易于解决的小模块, 有利于程序设计工作的分工和组织, 也使调试工作比较容易进行。

(2) 在程序设计中, 编写程序的控制结构仅由 3 种基本的控制结构(顺序结构、选择结构和循环结构, 它们的算法表示将在第 3 章中介绍)组成, 避免使用可能造成程序结构混乱的 `goto` 转向语句。

所谓程序的控制结构是指用于规定程序流程的方法和手段。它是一种逻辑结构, 描述程序执行的顺序, 也是一种形式结构, 描述程序的编写规则。按照结构化程序设计方法, 使设计编写的程序的控制结构由上述 3 种结构组成, 这样的程序就是结构化程序。C 语言是结构化程序设计语言。

结构化程序设计技术虽已使用了几十年, 但软件质量的问题仍未得到很好的解决。这是因为面向过程的程序设计方法, 仍然存在与人的思维方式不协调的地方, 所以很难自然、准确地反映真实世界, 因而用此方法开发出来的软件, 还是很难保证质量, 甚至需要进行重新开发。另外, 该方法在实现中只突出了实现功能的操作方法(模块), 而被操作的数据(变量)处于实现功能的从属地位, 即程序模块和数据结构是松散地结合在一起的。因此, 当应用程序比较复杂时, 容易出错, 难以维护。

为适应现代化软件开发的需要, 一种全新的软件开发技术应运而生, 这就是面向对象的程序设计(Object Oriented Programming, OOP)。

3. 面向对象的程序设计

面向对象的程序设计在 20 世纪 80 年代初就提出了, 它起源于 Smalltalk 语言。用面向对象的方法解决问题, 不再将问题分解为过程, 而是将问题分解为对象。对象是现实世界中可以独立存在、可以被区分的实体, 也可以是一些概念上的实体, 世界是由众多对象组成的。对象有自己的数据(属性), 也包括作用于数据的操作(方法), 对象将自己的属性和方法封装成一个整体, 供程序设计者使用。对象之间的相互作用通过消息传送来实现。这种“对象+消息”的面向对象的程序设计模式将取代“数据结构+算法”的面向过程的设计模式。

但要注意到, 面向对象的程序设计并不是要抛弃结构化程序设计方法, 而是站在比结构化程序设计更高、更抽象的层次上去解决问题。当它被分解为低级代码模块时, 仍需要结构化编程的方法和技巧, 只是它分解一个大问题为小问题时采取的思路与结构化方法是不同的。结构化的分解突出过程, 强调的是如何做(How to do?), 代码的功能如何完成; 面向对象的分解突出现实世界和抽象的对象, 强调的是做什么(What to do?), 它将大量的工作由相应的对象来完成, 程序员在应用程序中只需说明要求对象完成的任务。

目前常用的面向对象的程序设计语言有: Borland C++、Visual C++、Visual FoxPro、Visual Basic、Java、Delphi 和 Visual Fortran 等。它们虽然风格各异, 但都有共同的概念和编程模式。

1.2 C语言简介

1.2.1 C语言的发展

C语言是在20世纪70年代初问世的。1973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,10年后(即1983年)由美国国家标准学会在此基础上制定了一个C语言标准,通常称之为ANSI C。

早期的C语言主要是用于UNIX操作系统。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语言开始进入其他操作系统,并很快在各类大、中、小和微型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。

在C语言的基础上,1983年贝尔实验室的Bjarne Stroustrup又推出了C++语言。C++语言进一步扩充和完善了C语言,成为一种面向对象的程序设计语言。C++语言目前流行的是Borland C++、Symantec C++和Microsoft Visual C++。C++语言提出了一些更为深入的概念,它所支持的这些面向对象的概念容易将问题空间直接地映射到程序空间,为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法,因而也增加了整个语言的复杂性,掌握起来有一定难度。

但是,C语言是C++语言的基础,C++语言和C语言在很多方面是兼容的。因此,掌握了C语言,再进一步学习C++语言就能以一种熟悉的语法来学习面向对象的语言,从而达到事半功倍的目的。在本书的第11章,将介绍有关C++语言的基本知识,使读者在学习C程序设计后,了解C++程序设计的基本概念,为进一步学习C++程序设计打下坚实的基础。

1.2.2 C语言的特点

C语言发展如此迅速,而且成为最受欢迎的语言之一,主要是因为它具有强大的功能。许多著名的系统软件,如FoxBASE、DBASE III都是由C语言编写的。用C语言加上一些汇编语言子程序,就更能显示C语言的优势,例如PC-DOS、Word等就是用这种方法编写的。归纳起来C语言具有下列特点。

(1) 简洁紧凑、灵活方便。

C语言一共只有32个关键字,9种控制语句,程序书写自由。一个C语句可以写在一行上,也可分多行书写,主要用小写字母表示。

(2) 运算符丰富。

C语言的运算符包含的范围很广泛,共有34个运算符。C语言把括号、赋值和强制类型转换等都作为运算符处理。从而使C语言的运算类型极其丰富,表达式类型多样化,可以灵活使用各种运算符实现在其他高级语言中难以实现的运算。