

21世纪高等学校计算机规划教材

C语言 程序设计基础

韩忠东 主编 马 华 张贞梅 王 玫 副主编

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



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

21 世纪高等学校计算机规划教材

C 语言程序设计基础

韩忠东 主编

马 华 张贞梅 王 玫 副主编

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书系统全面地介绍了 C 语言的有关知识和技术。全书共 10 章, 主要内容涉及计算机程序设计概述、数据类型、运算符和表达式, C 程序的逻辑结构, 数组, 指针, 结构体、共用体和枚举类型, 函数, 宏定义与文件包含, 文件以及程序设计的原则和风格等内容。

全书内容安排合理, 每章设有“内容概述”、“教学目的”、“学习方法”、“小结”, 并辅以大量的典型例题以及综合应用, 便于学生对知识点融会贯通。章后配有丰富习题, 部分精选于全国计算机等级考试历年的真题, 具有典型性和启发性。第 10 章对程序设计的基本风格和一般规则进行了详细分析和总结, 有利于培养学生良好的编程习惯, 提高其编程能力。

本书可作为高等院校计算机及相关专业程序设计基础课教材, 亦可供编程爱好者、相关专业人员作为自学用书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目(CIP)数据

C 语言程序设计基础/韩忠东主编. —北京: 电子工业出版社, 2007.8

21 世纪高等学校计算机规划教材

ISBN 978-7-121-04799-2

I. C… II. 韩… 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 (2007) 第 117317 号

责任编辑: 陈 虹

印 刷: 北京季峰印刷有限公司

装 订: 三河市万和装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 15.25 字数: 390 千字

印 次: 2007 年 8 月第 1 次印刷

印 数: 4 030 册 定价: 22.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

C 语言功能丰富、使用灵活,目标程序效率高、可移植性好,程序结构性和可读性好,体现了结构化程序设计思想,适合于培养良好的编程风格和训练程序设计技术。C 语言同时也具有许多低级语言的特点,它既可用于开发系统软件,也可用于开发应用软件。另外 C 语言是 C++ 的基础,而 C++ 是学习面向对象编程首选语言,因此学习 C 语言十分必要。

本教材以 C 语言为教学语言,由浅入深、循序渐进,通过大量涉及科学计算、数据处理等方面的例子,充分展示了通过计算机程序设计解决问题的思想和方法,突出了程序设计的基本方法的阐述。希望读者能够重视实践,注重理论联系实际,锻炼动手能力,多思考、多练习,将学习的目标放在独立解决问题上来。

本书适合作为大学计算机及相关专业的程序设计基础课教材。具有以下特点:

(1)“内容概述”、“教学目的”、“学习方法”、“小结”的设立,便于教师组织教学,并有利于学生预习和自学;

(2)实例丰富,除经典例题外,各章还设有综合应用举例,便于学生融会贯通每章的知识点,另外,每章后给出多种类型的高质量习题,便于教师考察学习效果;

(3)以标准的流程书写书中的例题,便于学生掌握规范的程序设计方法;

(4)第 10 章中,在讨论了衡量程序的各主要要素的基础上,对程序设计的基本风格和一般规则进行了详细分析和总结,这对于养成良好的编程习惯,提高程序的可读性、可维护性等方面具有非常重要的意义。

全书组稿和编排工作由韩忠东担任,第 1 章由韩忠东编写,第 2、3 章由孙静、马晓艳、江震、王秀娟共同编写,第 4 章由李玉娟编写,第 5 章由张兰华编写,第 6 章由彭磊编写,第 7 章由马华和张贞梅编写,第 8、9 章由王玫编写,第 10 章由贺宏伟、左风华编写。马华、王玫、李玉娟最后成稿。泰山医学院的张兆臣教授、张西学教授对本书的初稿进行了审阅,并提出了许多宝贵的修改意见。此外,本书的编写还得到了各级领导和电子工业出版社编辑部诸位老师的关心和支持,在此一并表示深深感谢。

由于编者水平有限,加之时间仓促,书中难免有不妥之处,我们衷心希望得到广大读者的批评指正。

编 者

2007 年 4 月于山东泰安

目 录

第 1 章 计算机程序设计概述	1
1.1 计算机程序设计基本概念	1
1.1.1 计算机程序与计算机语言	2
1.1.2 算法与数据结构	2
1.1.3 计算机程序设计发展史	3
1.1.4 程序的纠错与测试	5
1.1.5 程序设计语言简介	5
1.2 C 语言简介	6
1.2.1 初识 C 程序	6
1.2.2 C 程序的基本组成	6
1.2.3 C 程序的基本概念	7
1.3 C 语言上机指南	8
1.3.1 Turbo C 的启动	8
1.3.2 常用菜单项简介	9
小结	14
习题 1	14
第 2 章 数据类型、运算符和表达式	15
2.1 C 语言数据类型	15
2.2 常量与变量	16
2.2.1 常量	16
2.2.2 变量	18
2.3 数值型数据间的混合运算	22
2.4 运算符与表达式	23
2.4.1 运算符与表达式的概念	23
2.4.2 算术运算符和算术表达式	24
2.4.3 自增、自减运算符	24
2.4.4 赋值运算符和赋值表达式	25
2.4.5 逗号运算符和逗号表达式	27
小结	27
习题 2	27
第 3 章 C 程序的逻辑结构	30
3.1 顺序结构	30
3.1.1 C 语句概述	30
3.1.2 赋值语句	31
3.1.3 数据的输入/输出语句	32

3.1.4 综合应用	39
3.2 选择结构	41
3.2.1 关系运算符和关系表达式	41
3.2.2 逻辑运算符和逻辑表达式	42
3.2.3 if 语句	43
3.2.4 switch 语句	48
3.2.5 综合应用	49
3.3 循环结构	51
3.3.1 while 语句(当型循环)	51
3.3.2 do-while 语句(直到型循环)	53
3.3.3 for 语句	54
3.3.4 goto 语句和 if 语句构成循环	57
3.3.5 几种循环结构的比较	57
3.3.6 转移语句	58
3.3.7 循环嵌套	61
3.3.8 综合应用	64
小结	67
习题 3	67
第 4 章 数组	72
4.1 一维数组	72
4.1.1 一维数组的定义	73
4.1.2 一维数组元素的引用	74
4.1.3 一维数组的初始化	75
4.1.4 一维数组的应用	76
4.2 二维数组	84
4.2.1 二维数组的定义	84
4.2.2 二维数组元素的引用	85
4.2.3 二维数组的初始化	86
4.2.4 二维数组应用举例	86
4.3 多维数组	92
4.4 字符数组	92
4.4.1 字符数组的定义	92
4.4.2 字符数组与字符串	93
4.4.3 字符串(字符数组)处理函数	96
4.4.4 字符数组应用举例	100
小结	103
习题 4	104
第 5 章 指针	107
5.1 指针的基本概念	107
5.1.1 地址	107

5.1.2 指针类型和指针变量	108
5.2 指针变量的使用	110
5.2.1 指针变量的定义	110
5.2.2 指针变量的访问	111
5.2.3 指针变量的引用	112
5.3 指针与数组	117
5.3.1 数组的指针和指向数组的指针变量	117
5.3.2 通过指针引用数组元素	118
5.3.3 指向多维数组的指针和指针变量	120
5.4 指针与字符串	122
5.4.1 字符串的表示形式	122
5.4.2 字符数组和字符指针	124
5.5 指针数组与指向指针的指针	125
5.5.1 指针数组	125
5.5.2 指针的指针	128
5.5.3 指针运算举例	129
小结	130
习题 5	130
第 6 章 结构体、共用体和枚举类型	132
6.1 结构体类型	132
6.1.1 结构体类型的定义	133
6.1.2 结构体变量的定义	134
6.1.3 结构体变量的引用	135
6.1.4 结构体变量的初始化	136
6.2 结构体数组	137
6.2.1 结构体数组的定义	138
6.2.2 结构体数组的初始化及引用	139
6.3 结构体指针变量	140
6.3.1 指向结构体的指针变量	140
6.3.2 指向结构体数组的指针变量	141
6.4 共用体	143
6.4.1 共用体类型的定义	143
6.4.2 共用体变量的定义	144
6.4.3 共用体变量的引用	145
6.5 枚举类型	147
6.5.1 枚举类型的定义	148
6.5.2 枚举变量的定义	148
6.5.3 枚举变量的引用	149
6.6 用 typedef 定义类型	149
小结	151

习题 6	151
第 7 章 函数	153
7.1 函数概述	153
7.1.1 C 语言的函数	153
7.1.2 C 语言程序的组成	153
7.1.3 使用函数的意义	154
7.2 函数的一般形式	155
7.2.1 函数声明	155
7.2.2 函数定义	155
7.3 函数的参数和返回值	157
7.3.1 形式参数与实际参数	157
7.3.2 函数的返回值	159
7.4 函数的调用	160
7.4.1 函数调用的一般形式	160
7.4.2 函数调用的方法	160
7.5 函数的嵌套调用和递归调用	161
7.5.1 函数的嵌套调用	161
7.5.2 函数的递归调用	162
7.6 变量的作用域	163
7.6.1 局部变量	164
7.6.2 全局变量	165
7.7 变量的存储类别	167
7.7.1 动态存储方式	168
7.7.2 静态存储方式	168
7.8 数组、指针与函数	169
7.8.1 数组、指针作为函数参数	170
7.8.2 利用指针调用函数	175
7.8.3 指针型函数	176
7.9 内部函数和外部函数	177
7.9.1 内部函数	177
7.9.2 外部函数	178
7.10 综合应用	178
小结	180
习题 7	181
第 8 章 宏定义与文件包含	185
8.1 宏定义	186
8.1.1 不带参数的宏定义	186
8.1.2 带参数的宏定义	187
8.2 文件包含	189
8.3 条件编译	191

小结	194
习题 8	194
第 9 章 文件	195
9.1 文件概述	195
9.2 文件类型指针	196
9.3 文件的打开与关闭	196
9.3.1 文件打开库函数 fopen()	196
9.3.2 文件关闭函数 fclose()	197
9.4 文件的读、写	198
9.4.1 读文件函数	198
9.4.2 写文件函数	201
9.5 文件的定位	204
9.5.1 重返文件头函数 rewind()	205
9.5.2 位置指针移动函数 fseek()	205
9.5.3 获取当前位置指针函数 ftell()	206
9.6 出错检测和处理	206
小结	207
习题 9	207
第 10 章 程序设计的原则和风格	211
10.1 程序的质量要素	211
10.2 程序代码的风格	213
10.2.1 程序的版式	213
10.2.2 命名规则	215
10.2.3 表达式和基本语句	216
10.2.4 函数设计	217
10.2.5 文件组织	218
小结	218
习题 10	218
附录 A 阿兰·图灵与冯·诺依曼简介	220
附录 B ASCII 码表	223
附录 C C 语言中的关键字	224
附录 D 运算符、运算符优先级及结合性一览表	225
附录 E Turbo C 2.0 常用库函数及其头文件	227
附录 F Turbo C 2.0 编译错误信息	231
参考文献	233

第 1 章 计算机程序设计概述

【内容概述】

本章介绍计算机程序设计的基本概念，以及 C 语言的相关知识。主要包括计算机软件、计算机程序、计算机语言、算法与数据结构、程序设计、程序的纠错与测试，以及 C 语言的基本结构和相关概念。本章的最后一部分，详细介绍了 Turbo C 集成编译环境下的使用方法。

【教学目的】

- 掌握计算机程序设计及其相关概念。
- 了解计算机程序设计基本方法和程序设计语言的发展历史。
- 了解计算机软件的发展史。
- 理解软件、程序与算法之间的内在联系。
- 掌握 Turbo C 的基本上机操作。

【学习方法】

计算机语言是与人类自然语言近似的一种符号体系，是人们用于描述信息、处理信息的一种基本工具。像学习其他语言一样，模仿是一种最好的入门方法。

1.1 计算机程序设计基本概念

众所周知，计算机是 20 世纪人类最伟大的发明之一。通过计算机语言编制计算机程序的方式来控制计算机，是人类将自身的智慧天赋拓展到机器设备的巧夺天工之举。于是就有了阿兰·图灵（如图 1-1 所示）、冯·诺依曼（如图 1-2 所示）等各位先贤，因为他们巨大的贡献而永载史册的美谈；也有了比尔·盖茨等一大批因为推广计算机软件技术而一跃进入



图 1-1 计算机科学之父阿兰·图灵



图 1-2 现代计算机之父冯·诺依曼

世界首富之列的 IT 精英。在计算机程序设计理论的建立方面，阿兰·图灵的贡献功不可没，被后人誉为计算机程序设计的理论奠基人、“计算机科学之父”，计算机领域的最高奖也被 ACM 命名为“图灵奖”。冯·诺依曼因为对计算机体系结构提出重大改进意见，使得计算机得以批量生产，成为人类社会发展的强大动力，被誉为“现代计算机之父”，为计算机程序设计的发展奠定了坚实的基础。直至今日，现代计算机仍然被称为“冯·诺依曼型计算机”。近几年来，以世界首富比尔·盖茨等为代表的计算机软、硬件市场的开拓者们，在为自身创造财富的同时，为计算机应用技术向人类社会的各个领域、各个角落快速渗透做出了非凡的贡献。

1.1.1 计算机程序与计算机语言

计算机程序是操作指令的集合，对于人们使用和控制计算机非常重要。计算机语言是使用人们生活中习惯的方式来描述计算机程序或操作指令的基本工具。

理解计算机软件和计算机程序的关系非常重要，通常前者对后者完全包含，即计算机程序是计算机软件的重要组成部分，是应用计算机指令系统完成计算任务的实体。计算机程序是计算机指令集中的指令调用数据集中的数据进行运算，或计算机指令集中的指令对计算机进行控制的操作集合。

1.1.2 算法与数据结构

应用计算机语言进行程序设计的时候，一个关键的、事先需要解决的问题是，用什么样的思路和步骤去描述解决问题的过程，而且该步骤可以方便转化为计算机语言。

先来看一个小问题：

同学聚会时，一位先生问已经成为数学家的他当年的同室好友：“今天见到这么多老同学真是太高兴了。而且对我来说，今天更是非同寻常，因为我的 3 个儿子同时在庆祝他们的生日！你能告诉我他们的具体年龄吗？”

数学家思考后回答：“我需要些提示。”

这位先生说：“他们的年龄之积是 36。”

数学家进一步思考后又说：“还不够。”

这位先生又说：“他们的年龄总数与那幢房子的窗户数相等。”

数学家沉思后说：“再来点提示。”

这位先生想要故意难为数学家：“大儿子的眼睛是蓝色的。”

没想到数学家说：“谢谢！稍微懂些算法的人都会告诉你答案的。你的 3 个儿子年龄分别是……，对吗？”

先生愕然！

这里给出解决问题的思路：

(1) 首先，根据条件 1，即 3 个孩子的年龄之积等于 36，有 8 种可能结果。

$$36 = 36 \times 1 \times 1$$

$$36 = 18 \times 2 \times 1$$

$$36 = 12 \times 3 \times 1$$

$$36 = 9 \times 4 \times 1$$

$$36 = 9 \times 2 \times 2$$

$$36 = 6 \times 6 \times 1$$

$$36 = 6 \times 3 \times 2$$

$$36 = 4 \times 3 \times 3$$

(2) 其次, 根据条件 2, 即不知道那幢楼的窗户数量。但是, 可以将所有可能的和列出来:

$$36 + 1 + 1 = 38$$

$$18 + 2 + 1 = 21$$

$$12 + 3 + 1 = 16$$

$$9 + 4 + 1 = 14$$

$$9 + 2 + 2 = 13 \quad /* 1 */$$

$$6 + 6 + 1 = 13 \quad /* 2 */$$

$$6 + 3 + 2 = 11$$

$$4 + 3 + 3 = 10$$

因为, 数学家知道窗户的数目, 依此判断, 肯定是有两种以上的选择使得他不能决断。结果只会在/* 1 */和/* 2 */之间。

(3) 最后, 那位先生的第 3 个提示为答案的最终确定提供了条件, “大儿子”的提示说明情况/* 2 */不合条件, 因为不存在大儿子。所以, 答案是: 3 个孩子的年龄分别是: 9 岁、2 岁、2 岁。

经过上述推理, 最终选择出了正确的答案。在选择的过程中, 经历了 (1), (2), (3) 等具体步骤。这些用来解决问题的具体步骤, 被称之为问题的求解算法, 简称算法。

而应用计算机解决问题的时候, 仅仅有算法是不够的, 还要组织好进行计算时所需的参与运算的数据。数据在计算机内部的逻辑和物理组织形式称之为数据结构。可以认为, 数据结构是将人所理解的算法变成计算机所理解的程序的基本工具。

1.1.3 计算机程序设计发展史

计算机程序设计是以计算机语言为工具编制计算机程序的一个计算机软件生产过程中的重要阶段。软件工程 (一门将计算机程序设计或软件开发看成一项工程技术项目的学科, 是开展软件项目施工规范、施工过程、技术方法、施工工具与平台等领域研究的计算机科学与技术这一一级学科下的二级子学科) 中, 计算机软件开发与设计的过程主要包括: 可行性分析、需求分析、设计、编码实现、测试、维护等。而计算机程序设计是由软件开发与设计过程中设计工作的一部分、编码实现工作及测试工作的全部组成的。

下面以时间为主线, 简要介绍计算机程序设计从孕育、诞生, 到成熟、壮大整个时期内的重大事件, 以便程序设计的初学者在这条轨迹中发现和总结一些可以给人们带来启示的东西。

1. 孕育时期

(1) Charles Babbage 巴贝奇 (英国人), 1822 年, 发明机械式差分机, 可用穿孔卡片输入工作程序和数据; 1834 年, 发明机械式分析机, 使用穿孔卡片 (ROM) 存储程序和数

据, 出现了控制中心 (CPU) 和存储程序 (存储器) 协调工作的雏形。

(2) 1848 年, George Boole 创立布尔代数, 即二进制代数学, 未来的量子计算机可能

会采用三进制。

(3) 1880 年, 美国为人口普查数据进行统计的时间达到 7 年, 1890 年再次普查, 统计时间可能会增加到 10 年以上, Herman Hollerith 模仿巴贝奇的机器, 设计了机械式计算机, 使人口普查数据计算 6 周就完成了。

(4) 1896 年, Herman Hollerith 创立了制造穿孔卡片计算机的公司, 即 IBM 的前身。

(5) 1906 年, 发明电子管, 为电子计算机的出现奠定了技术基础。

(6) 1924 年 2 月, IBM 诞生。

(7) 1937 年, Alan M. Turing 发表了一篇论文, 在论文的注释中提出了一种模型, 即“图灵机”, 至今所有的软件都没有突破该模型。

(8) 1937 年, 贝尔实验室用继电器模拟二进制成功。

(9) 1939 年, 机电式计算机在 HP 公司诞生, David Hewlet 和 William Packard 用投硬币方式决定了谁的名字在工资名称中靠前。

2. 诞生时期

(1) 1943—1959 年, 电子管计算机大量诞生, 30 吨的 ENIAC 被认为是第一台数字电子计算机。

(2) 1949 年, EDVAC 首次使用磁带为存储介质, 科学家预言未来的计算机将不会超过 1.5 吨。

(3) 1952 年, 在 MIT (麻省理工学院) 诞生了汇编语言。

(4) 1954 年, 在 IBM 诞生了 FORTRAN (FORmula TRANslation) 语言。

(5) 1958 年, Intel 的创始人 Robert Noyce 发明了集成电路。

(6) 1959—1961 年, COBOL (Common Business Oriented Language) 语言诞生, 风靡银行及大型企业。

(7) 1960 年, Algol60 语言出现, 它是今天的 C, BASIC, Pascal 的共同祖先。

(8) 1961 年, 专门用于矩阵运算的 APL 语言诞生。

(9) 1965 年, BASIC 诞生, 用于初学者。

(10) 1967 年, Pascal 出现, 用于教学。

3. 发展时期

(1) 1968 年, 软件工程, 面向对象思想开始萌芽。

(2) 1969 年, ARPANET 计划启动。

(3) 1970 年, UNIX 开始研制。

(4) 1971 年, 微处理器诞生。

(5) 1972 年, C 语言诞生, 系统级主力语言, 低级语言终结者。

(6) 1974 年, 微型计算机诞生, Bil Gates 和 Paul Allen 开始用 BASIC 写程序。

(7) 1975 年, Microsoft 诞生。

(8) 1976 年, Z80 处理器和 CP/M 操作系统 (Zilog 公司), APPLE 独占个人机市场。

(9) 1979 年, ADA 语言诞生, IBM 介入个人机并迅速取得优势, 微软开始了在 IBM 的寄生生涯 (操作系统、开发工具)。

(10) 1981 年, IBM_PC 上市, 微软为之提供操作系统, 但未被 IBM 买断, 这是微软独霸 PC 操作系统的开始。

(11) 1983 年, BORLAND 公司成立, 开发了当时最流行的语言 Turbo Pascal 和 Turbo C。

4. 壮大时期

(1) 1985 年, Microsoft 开发 Windows 1.0; 1993 年, Windows 3.1 上市, 由于盗取 APPLE 技术取得成功; 1997 年, 微软向处于财政危机的 APPLE 注入 1.5 亿作为回报。

(2) 1989 年, WWW 和 HTML 推动了 Internet 的繁荣。

(3) 1994 年, Netscape1.0 发布。

(4) 1995 年, 微软公司的 Windows95 面世, 而后推出 Windows NT, Windows2000, Windows2003 等, 迅速成为个人计算软件开发平台的主要垄断者, 因此, 微软公司多次站到反垄断诉讼的被告席上。2007 年微软又推出新的操作系统 Vista, 因为新的操作系统包含诸多垄断因素, 以欧洲为首的国际软件联盟再次对其起诉。

(5) 号称“微软终结者”的 JAVA 面世, JSP 出现。

(6) 1996 年, 基于 JSP 的 Netscape Navigator2.0 面世, 微软岌岌可危, 但靠垄断力量迅速扭转局面。

(7) 2002 年, 微软经过卧薪尝胆和已经占领市场的优势, .NET 的诞生是微软度过其危机四伏时期的标志。

(8) 自 2003 年至今, 计算机程序设计的开发平台进入“J2EE 和 .NET 并存, Java 与 C# 对垒”的时代。

1.1.4 程序的纠错与测试

在计算机程序设计过程中, 编写代码是必须存在的重要工作。但是, 还有很多实际的工作在等着人们去完成。如程序代码完成之后的两项非常重要的工作: 程序的纠错和测试。

程序的纠错是指将程序中包含的语法和语义错误找出来, 以便人们能够得到正确运行的计算机软件的过程。

一个成熟的软件产品, 还需要在程序正确运行之后, 对程序中可能隐含的错误全面剔除, 这项工作叫做程序的测试。

1.1.5 程序设计语言简介

计算机程序设计语言包括高级语言、低级语言。

高级语言是指接近人类语言的计算机语言, 目前流行的高级语言包括: FORTRAN 语言、Pascal 语言、C 语言、BASIC 语言、Java 等。

低级语言是指与计算机硬件指令系统密切对应的计算机语言。包括机器语言和汇编语言两种。其中机器语言是能够被计算机直接调用的、与计算机指令系统一致的纯粹二进制语言; 而汇编语言则是人们为了摆脱记忆二进制指令的麻烦, 而给出机器指令的助记符号的计算机语言。

计算机高级语言的诞生与低级语言的诞生间隔不久。在 20 世纪 80 年代至 90 年代, 高级语言又经历了面向对象思想的冲击产生了许多语言的面向对象设计的版本, 有代表性的包括 C++ 语言、面向对象的 Pascal 语言 (Delphi 为代表)、ADA 语言、Java 语言等。进入 21 世纪之后, 面向构件与软件重用思想的冲击又产生了一些面向构件的程序设计架构, 如微软的 COM+, SUN 公司的 J2EE (Java 2 for Enterprise Edition, 另含 J2SE 和 J2ME), 还有企业

界公认的有良好口碑的 CORBA 等。

1.2 C 语言简介

自从 C 语言诞生以来,就因为它的灵活性、功能强大,备受程序员的喜欢。它以其简洁的语法和较高的执行效率将过程语言带到了最高峰,可以说 C 语言是世界上使用最多的一种编程语言。后来在给 C 语言增加面向对象功能的目的下, Bjarne Stroustrup 开发了 C with class, 后来发展成 C++。然后, C++ 开始取代 C 语言的霸主地位。但是直到今天,在修改早期程序,进行某些平台的嵌入开发, Linux 开发等很多方面,仍有 C 语言的一席之地。从计算机语言发展的历史看, C 语言是最伟大的编程语言之一。

C 语言作为国际上流行的、颇具发展前途的计算机高级语言,适合作为系统描述语言。它既可以用来编写系统软件,也可以用来编写应用程序。最初,各种操作系统以及其他系统软件主要采用汇编语言编写。汇编语言依赖于计算机硬件,程序的可读性、可移植性都比较差。为了提高可读性和可移植性,人们希望采用高级语言编写这些软件,但是一般的高级语言难以实现汇编语言的某些操作,无法直接面对硬件编程,如内存地址的读写、直接硬件、二进制位的操作等。人们设法寻找一种既具有一般高级语言特性,又具有低级语言特性的语言, C 语言就在这种情况下生存并发展起来。目前流行的各种操作系统,除了内核部分,绝大多数都是由 C 语言编程实现的,从这一角度来看,操作系统是一种特殊的计算机程序。

1.2.1 初识 C 程序

下面就是一个非常简单的 C 语言源程序,它完成了计算 1~99 的所有整数求和的功能。

【例 1-1】计算 1~99 间所有整数之和。

```
main()
{
    int i, sum=0;                                /*说明性语句*/

    for (i = 0; i < 100; i++)                    /*功能性语句*/
    {
        sum = sum + i;
    }
    printf("The sum of 1 to 100 is: %d\n",sum);
}
```

1.2.2 C 程序的基本组成

分析上述程序,可以看出:

(1) 语法简洁清晰、易于理解。稍懂些英语的人都能看懂语言的含义。

(2) 有些运算符与传统的符号相同或近似,如等号、加号;有些则存在较大差别,如“++”表示增 1 运算,“i++”表示 i 的值增加 1。

(3) 程序结构完整。main 表示“主函数”。每个 C 语言程序都必须有一个 main 函数,它是每一个 C 语言程序的执行起始点(入口点)。main()表示“主函数”。一个 C 源程序至少

包含一个 main 函数，也可以包含一个 main 函数和若干个其他函数。函数是 C 语言的基本组成单位。

(4) 用 { } 括起来的是“主函数”main 的函数体。main 函数中的所有操作都在这一对 {} 之间。也就是说 main 函数的所有操作都在 main 函数体中。

函数体一般包括声明部分、执行部分两部分，即分别为例 1-1 中的说明性语句和功能性语句。

```
{  
    [声明部分]: 在这部分定义本函数所使用的变量。  
    [执行部分]: 由若干条语句组成命令序列 (可以在其中调用其他函数)。  
}
```

(5) C 是函数式的语言，程序的全部工作都是由各个函数完成。被调用的函数可以是系统提供的库函数，也可以是用户根据需要自己编写设计的函数。在 Turbo C 开发环境中，库函数存放在 INCLUDE 和 LIB 两个文件夹下面。编写 C 程序就是编写各个函数。“主函数”printf 是 C 语言的库函数，功能是用于程序运行结果的输出（显示在屏幕上），本例用于将 $1+2+3+\cdots+99$ 的和计算出来并显示在屏幕上，程序的运行结果为：

The sum of 1 to 100 is: 4950

1.2.3 C 程序的基本概念

C 程序的基本概念除了上面介绍的函数之外，还有以下几个。

1. 语句

语句是指计算机程序中完成相对完整功能的一句话。计算机语句包括说明性语句和功能性语句。例 1-1 中除了一行说明性语句外，下面的均可看为功能性语句。构成语句的基本要素包括关键字、变量、常量、运算符、表达式，语句都应该符合约定的语法规则。后面章节还将介绍语句的其他分类法。

2. 版本

在 C 语言的不同发展阶段，各软件公司推出彼此间有一定差别的各种版本。各版本虽然遵循相同的 C 语言的语法规则和标准，但又存在没有涉及的细节差异。新版本是对老版本的更新换代。下面就是一个关于 C 语言版本的简单介绍。

- Borland 公司: Turbo C, Turbo C++, Borland C++, C++ Builder (Windows 版本)。
- Microsoft 公司: Microsoft C, Visual C++ (Windows 版本), Visual C++.Net, C#。

3. 程序

程序可以分为源程序、目标程序和可执行程序。

(1) 源程序: 程序可以用高级语言或汇编语言编写，用高级语言或汇编语言编写的程序称为源程序。C 程序源程序的扩展名为“.c”。

(2) 目标程序: 源程序经过“编译程序”翻译所得到的二进制代码称为目标程序。目标程序的扩展名为“.obj”。

(3) 可执行程序: 目标程序与库函数连接，形成的完整的可在操作系统下独立执行的程序称为可执行程序。可执行程序的扩展名为“.exe”（在 DOS 或 Windows 环境下）。

其他相关概念将在今后的学习中逐步引入和介绍。

1.3 C 语言上机指南

以下以 Turbo C 为例介绍使用 C 语言进行程序设计的基本上机操作步骤。与使用其他任何语言编写程序相似，C 语言程序的上机步骤如图 1-3 所示。

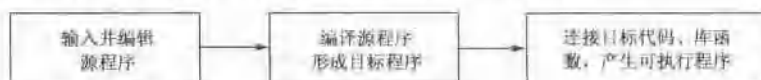


图 1-3 使用 C 语言上机操作的主要步骤

1.3.1 Turbo C 的启动

Turbo C2.0 是 Borland 公司开发的微机上一个 C 语言集成开发环境，启动后的初始界面如图 1-4 所示。可以在 Turbo C 中完成 C 语言程序的编辑、编译、连接、运行、调试。初次使用需要重点练习：启动、退出，以及重要菜单项的功能。启动方法主要有两种：①DOS 环境下键入命令和回车启动，如图 1-5 所示；②Windows 下双击主程序 TC.EXE 启动，如图 1-6 所示。启动之后的状态均如图 1-4 所示，操作者按下 F10 或者“ALT+首字母”完成对菜单项的访问。从图 1-4 可以看出，在 Turbo C2.0 启动后共有 File 等 8 个菜单项，下面只对常用的菜单项功能做介绍。



图 1-4 Turbo C2.0 初始界面



图 1-5 DOS 环境下启动 Turbo C2.0 的过程