

● 高等学校教材

C语言程序设计

刘达明 聂永萍 陈昌志 王宁 代永亮 编著



高等教育出版社
HIGHER EDUCATION PRESS

高等学校教材

C 语言程序设计

C Yuyan Chengxu Sheji

刘达明 聂永萍 陈昌志 王 宁 代永亮 编著



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书结合计算机等级考试大纲和要求,由浅入深全面介绍 ANSI C,不仅涵盖 C 语言的基本知识,而且注重 C 语言程序的详细讲解。

全书共 13 章,第 1 章主要介绍 C 语言的概貌,第 2~4 章主要介绍 C 语言的基础语法知识和三种基本结构程序设计,第 5 章主要以函数知识为背景介绍模块化程序设计,第 6~9 章主要介绍指针和数组,第 10 章主要介绍结构体和共用体构造数据类型及其指针,第 11 章主要介绍文件处理程序设计,第 12 和 13 章是 C 语言指针的高级应用和位运算。

本书内容全面、层次清晰、逻辑性强、例题丰富、通俗易懂。各章以问题分析与实现入手,重点讲述章节知识点,并通过实例巩固知识点,环环相扣,层层推进,达到通俗易懂的效果,还依据计算机等级考试题型配备了大量习题和模拟练习题。

本书既可以作为高等院校非计算机专业的 C 语言程序设计课程教材,也可以作为计算机类专业的相应课程教材,还可以作为读者自学和 C 语言程序开发人员的极佳参考书。另外本教材还配备有《C 语言程序设计实验与习题解答》辅助教材。

图书在版编目(CIP)数据

C语言程序设计 / 刘达明等编著. -- 北京 : 高等教育出版社, 2013. 1
ISBN 978-7-04-036770-6

I. ①C… II. ①刘… 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(2012)第001485号

策划编辑 刘 茜 责任编辑 刘 茜 封面设计 王 洋 版式设计 余 杨
插图绘制 尹 莉 责任校对 杨凤玲 责任印制 朱学忠

出版发行	高等教育出版社	咨询电话	400-810-0598
社 址	北京市西城区德外大街 4 号	网 址	http://www.hep.edu.cn
邮政编码	100120		http://www.hep.com.cn
印 刷	保定市中国美凯印刷有限公司	网上订购	http://www.landaco.com
开 本	787mm×1092mm 1/16		http://www.landaco.com.cn
印 张	22	版 次	2013 年 1 月第 1 版
字 数	540 千字	印 次	2013 年 1 月第 1 次印刷
购书热线	010-58581118	定 价	32.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换

版权所有 侵权必究
物 料 号 36770-00

前言

C 语言是国内外长期广泛使用的一种计算机语言,它以其功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好等优点深受广大程序开发人员青睐,已经成为硬件产品开发者 and 底层程序开发人员广泛运用的程序开发工具之一。

由于 C 语言具有便于描述算法、方便深入系统底层、语法结构简洁、目标代码高效等特点,诸如 Windows、UNIX、Linux、Mac、os/2 等大多数操作系统都采用 C 语言编写;还有 C++、Java、C#、J#、Perl 等很多新型的语言也都是由 C 语言衍生的。只要掌握了 C 语言,通过简单学习就很容易掌握其他语言了。目前社会招聘软件开发人员,很多单位也将 C 语言作为最基本的知识点进行考核。

本书结合计算机等级考试大纲和要求,由浅入深全面介绍 ANSI C,不仅涵盖 C 语言的基本知识,而且注重 C 语言程序的详细讲解。全书共 13 章,包括:C 语言概述、简单 C 程序设计、选择结构程序设计、循环结构程序设计、模块化程序设计、初识指针、一维数组及其指针、二维数组及其指针、字符数组及其指针、构造数据类型、文件处理程序设计、指针的高级应用和位运算等内容。第 1 章全面介绍 C 语言的概貌,包括 C 语言程序组成、上机步骤等;第 2~4 章介绍 C 语言的基础语法知识和三种基本结构,包括关键字与标识符、常量与变量、基本数据类型、运算符与表达式、C 语句、数据的输入输出、预处理命令等知识点,以及顺序结构、选择结构、循环结构等三大控制结构程序设计;第 5 章模块化程序设计,主要介绍模块化程序设计思想、函数的定义、函数的调用与返回、函数的声明、参数的传递、函数的嵌套调用与递归调用,以及变量的作用域和变量的存储类型、内部函数与外部函数等;第 6~9 章介绍指针和数组,包括变量的指针、一维数组及其指针、二维数组及其指针、字符数组及其指针等;第 10 章主要介绍结构体和共用体构造数据类型及其指针;第 11 章主要介绍文件处理程序设计,包括文件的打开与关闭、文件数据的读与写等;第 12 和 13 章是 C 语言的高级应用,主要包括返回指针的函数、指向函数的指针、指针数组、指向指针的指针、指针数组作 main 函数的形参、动态内存分配函数、链表、动态数组、位运算、位段等。

本书附录还提供了 ASCII 码表、C 库函数、Turbo C 2.0 和 Visual C++ 6.0 编程环境中的常见错误、计算机等级考试模拟练习等内容,供读者查阅和使用。本教材例题程序在 Turboc 2.0 和 Visual C++ 6.0 下调试通过。

本书由刘达明、聂永萍、陈昌志、王宁、代永亮编著。刘达明编写第 6~9 章、第 12 章,聂永萍编写第 2~5 章、第 10 章、附录 D,陈昌志编写第 1 章、第 13 章、附录 A、附录 B,王宁编写第 11 章、附录 C,代永亮编写第 12 章中动态数组、并调试了本教材全部例题程序。全书由刘达明统稿和校稿,代永亮承担了部分校稿工作。在编写过程中,瞿中、甘玲、郑先锋、冯潇、张莉萍、刘亮、唐晓军、薛峙、霍敏霞、向碧群等老师提供了宝贵意见和建议,在此表示感谢。同时,感谢本教材所列参考文献的作者和其他可能遗漏的作者,感谢为出版本教材而付出辛勤劳动的高等教育出版社

工作人员。对使用本教材,以及给本教材提供宝贵意见和建议的全体老师,在此也一并表示感谢。

由于作者水平有限,加之时间仓促,错误与不足之处在所难免,恳请读者批评指正。如需要联系作者,或想索取本教材例题源代码,请发信息至:liudaming@cqupt.edu.cn 或 liudaming88@163.com。

作者于 2012 年夏

目录

第 1 章 C 语言概述	1
1.1 认识程序	1
1.1.1 直观理解程序	1
1.1.2 计算机程序	2
1.2 C 语言程序概述	2
1.2.1 C 语言的发展	2
1.2.2 C 语言的特点	3
1.2.3 C 语言的作用	4
1.2.4 C 语言程序的组成	4
1.2.5 C 语言程序书写规则	6
1.3 C 语言程序上机运行	7
1.3.1 C 语言程序上机步骤	7
1.3.2 TC 环境下的程序运行	8
1.3.3 VC 环境下的程序运行	10
本章小结	13
习题 1	14
第 2 章 简单 C 语言程序设计	15
2.1 计算圆周长和面积	15
2.1.1 分析问题	15
2.1.2 程序实现	15
2.2 常量与变量	16
2.2.1 关键字与标识符	16
2.2.2 常量	16
2.2.3 变量	17
2.3 C 语言数据类型	18
2.3.1 整型数据	18
2.3.2 浮点型数据	19
2.3.3 字符型数据	20
2.3.4 枚举类型	21
2.4 运算符与表达式	22
2.4.1 C 语言运算符与表达式简介	22
2.4.2 算术运算符与算术表达式	23
2.4.3 赋值运算符与赋值表达式	24
2.4.4 条件运算符与条件表达式	25
2.4.5 逗号运算符与逗号表达式	25
2.4.6 sizeof 运算符	26
2.5 C 语句概述	26
2.6 数据的输入输出	27
2.6.1 字符数据的输入与输出	27
2.6.2 格式输入与输出	28
2.7 预处理命令	30
2.7.1 宏定义	31
2.7.2 文件包含	34
2.8 顺序结构程序设计	35
本章小结	36
习题 2	36
第 3 章 选择结构程序设计	40
3.1 判断闰年	40
3.1.1 分析问题	40
3.1.2 程序实现	40
3.2 关系运算符与关系表达式	41
3.3 逻辑运算符与逻辑表达式	42
3.4 if 语句	43
3.5 switch 语句	48
3.6 选择结构的嵌套	50
3.7 应用实例	52
本章小结	54
习题 3	55
第 4 章 循环结构程序设计	59
4.1 计算若干自然数的和	59
4.1.1 分析问题	59
4.1.2 程序实现	59
4.2 自增自减运算符	60

4.3 while 循环	62	6.4 应用实例	117
4.4 do-while 循环	64	本章小结	119
4.5 for 循环	65	习题 6	119
4.6 嵌套循环	69	第 7 章 一维数组及其指针	124
4.7 break 语句和 continue 语句	71	7.1 计算学生的平均成绩	124
4.8 应用实例	73	7.1.1 分析问题	124
本章小结	75	7.1.2 程序实现	124
习题 4	75	7.2 一维数组	127
第 5 章 模块化程序设计	81	7.2.1 一维数组的定义与初始化	127
5.1 求三个数的最大值与 最小值的差	81	7.2.2 一维数组的引用	128
5.1.1 分析问题	81	7.3 指向一维数组的指针	130
5.1.2 程序实现	81	7.3.1 指针的运算	130
5.2 模块化程序设计思想	83	7.3.2 指向一维数组的指针	131
5.3 函数的定义	84	7.4 一维数组及其指针作函数参数	135
5.4 函数的调用	85	7.5 应用实例	138
5.4.1 函数的调用形式	85	本章小结	150
5.4.2 函数的参数传递	86	习题 7	150
5.4.3 函数的返回值	88	第 8 章 二维数组及其指针	154
5.5 函数的声明	89	8.1 打印九九乘法表	154
5.6 函数的嵌套调用	92	8.1.1 分析问题	154
5.7 函数的递归调用	94	8.1.2 程序实现	154
5.8 变量的作用域	97	8.2 二维数组	158
5.9 变量的存储类型	99	8.2.1 二维数组的定义与初始化	158
5.10 内部函数和外部函数	103	8.2.2 二维数组的引用	160
5.11 应用实例	103	8.3 指向二维数组的指针	162
本章小结	106	8.3.1 二维数组的地址	162
习题 5	106	8.3.2 列指针	163
第 6 章 初识指针	111	8.3.3 行指针	164
6.1 交换两个变量的值	111	8.4 二维数组及其指针作函数参数	166
6.1.1 分析问题	111	8.4.1 二维数组作函数参数	166
6.1.2 程序实现	112	8.4.2 指向二维数组的指针作函数 参数	168
6.2 指针的概念	113	8.5 应用实例	170
6.2.1 地址与指针的概念	113	本章小结	176
6.2.2 取地址运算符和指针运算符	114	习题 8	176
6.2.3 指针变量的定义与引用	115	第 9 章 字符数组及其指针	181
6.3 指针变量作函数参数	116	9.1 比较两个字符串的大小	181

9.1.1 分析问题	181	11.1.1 分析问题	242
9.1.2 程序实现	181	11.1.2 程序实现	242
9.2 字符数组	185	11.2 文件的基本概念	243
9.2.1 字符数组的定义与初始化	185	11.3 打开与关闭文件	245
9.2.2 字符串与字符数组	186	11.4 文件的出错检查	247
9.3 指向字符数组的指针	189	11.5 顺序读写文件数据	248
9.4 字符数组及其指针作函数 参数	191	11.5.1 读写一个字符	248
9.5 字符串处理函数	192	11.5.2 读写一个字符串	249
9.6 应用实例	195	11.5.3 格式化方式读写数据	251
本章小结	202	11.5.4 二进制方式读写数据	251
习题 9	203	11.6 随机读写文件数据	252
第 10 章 构造数据类型	208	11.6.1 文件定位	252
10.1 输出学生成绩信息	208	11.6.2 随机文件的读写	253
10.1.1 分析问题	208	11.7 应用实例	254
10.1.2 程序实现	208	本章小结	256
10.2 结构体变量	209	习题 11	256
10.2.1 结构体变量的定义与初始化	209	第 12 章 指针的高级应用	261
10.2.2 结构体变量的引用	213	12.1 函数与指针	261
10.3 结构体数组	215	12.1.1 返回指针的函数	261
10.3.1 结构体数组的定义与初始化	215	12.1.2 指向函数的指针	263
10.3.2 结构体数组的引用	216	12.2 带参数的 main 函数	267
10.4 结构体及其指针	217	12.2.1 指针数组	267
10.4.1 指向结构体变量的指针	217	12.2.2 指向指针的指针	270
10.4.2 指向结构体数组的指针	219	12.2.3 指针数组作 main 函数的形参	272
10.4.3 结构体及其指针作函数参数	221	12.3 动态内存分配函数	274
10.5 共用体类型	223	12.4 链表	275
10.5.1 读取一个整型数据的高字节 数据	223	12.4.1 链表概述	275
10.5.2 共用体变量的定义	224	12.4.2 自引用结构	275
10.5.3 共用体变量的引用	226	12.4.3 链表的建立与输出	276
10.6 使用 typedef 定义数据类型的 别名	228	12.4.4 链表的插入与删除	278
10.7 应用实例	230	12.5 动态数组	281
本章小结	236	本章小结	284
习题 10	236	习题 12	284
第 11 章 文件处理程序设计	242	第 13 章 位运算	288
11.1 复制文件	242	13.1 位运算符与位运算	288
		13.1.1 位运算符	288
		13.1.2 位运算	288
		13.2 位段	291

13.3 应用实例	293	附录 C Turbo C 2.0 和 Visual c++6.0	
本章小结	294	编程环境中的常见错误	303
习题 13	295	附录 D 计算机等级考试模拟	
附录 A ASCII 码表	297	练习	306
附录 B C 库函数	298	参考文献	343

第 1 章 C 语言概述

人与计算机之间的交互可以通过各种各样的语言来实现。C 语言是国际上广泛流行的计算机语言,它既可以用来编写系统软件,也可以用来编写应用软件。

本章从认识程序开始,简单介绍程序的基本概念,C 语言的发展、特点及其作用,C 程序及上机过程。通过本章的学习,能够了解 C 语言的特点,初步认识 C 程序并了解 C 程序的运行过程。

1.1 认识程序

1.1.1 直观理解程序

计算机可以作为延伸人类能力的有力工具,人们通过程序来驾驭计算机。“程序”的概念来自生活,通常指完成某些事务按顺序实施的一个步骤。例如:起床、刷牙、洗脸、吃饭、上课、睡觉等。

在解决数学问题时,程序就是解决数学问题的步骤。如:

1. 求圆的面积和周长步骤

- (1) 开始,输入圆的半径 r ;
- (2) 计算圆的面积 $= 3.14 * r * r$;
- (3) 计算圆的周长 $= 2 * 3.14 * r$;
- (4) 输出得结果;结束。

2. 求 $1+2+\cdots+10$ 的值步骤

- (1) 开始,设置 s 为存储计算结果的存储单元,初值为 0;
- (2) 把 1 加到 s 中,其和仍记为 s ;
- (3) 把 2 加到 s 中,其和仍记为 s ;
-
- (10) 把 9 加到 s 中,其和仍记为 s ;
- (11) 把 10 加到 s 中,其和仍记为 s ;
- (12) 输出 s 的值;结束。

上述例子可以看出,程序有如下特征:

- (1) 开始与结束。

(2) 按一定的步骤完成相关工作。

(3) 用某种形式描述。

计算机程序则是在计算机中解决问题的步骤描述。

1.1.2 计算机程序

计算机程序是人与计算机交流信息的基本方式,人通过程序来控制计算机,程序是计算机系统的灵魂,人们描述(编制)计算机程序的工作称为程序设计或编程,其产品就是软件。

在计算机的整个工作过程中,执行者是硬件,控制者是程序(软件)。程序是机器指令序列的集合,是为完成某项功能所执行的命令序列。早期软件与程序的概念是一样的,随着时间的推移,软件的概念也在发展,软件是程序、数据、文档等的总称。但是,一般仍将程序的集合简称为软件。

软件是由程序员编写的程序源代码,是按照计算机语言的语法规则组成的。由于计算机不能直接识别高级语言代码,因此,需要通过编译器把源代码转换为能被计算机理解的机器代码,并把机器代码以可执行文件的形式保存在磁盘上。

编译器的翻译方式分为解释方式和编译方式。解释方式是通过运行“解释器”一条一条地解释源程序语句并执行的,速度慢,如 BASIC、JavaScript、VBScript、ASP 等是属于解释语言。编译方式是最经典而高效的运行方式,它将源程序翻译成目标代码并链接相关代码生成可执行文件,如 C 语言就是一种编译语言。

1.2 C语言程序概述

1.2.1 C语言的发展

C 语言是 1972 年由美国的 Dennis Ritchie 设计发明的,并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。

C 语言的原型是 1960 年出现的 ALGOL 60 语言,它是面向问题的高级语言。1963 年,英国剑桥大学在 ALGOL 60 的基础上推出了能够访问硬件的 CPL(Combined Programming Language)语言,但它规模较大,难以实现。1967 年,剑桥大学又对 CPL 语言做了简化,推出了 BCPL(Basic Combined Programming Language)语言。1970 年,AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言又一次做了简化,设计出较先进的并取名为 B 的语言,并且他用 B 语言写了 UNIX 操作系统。1972 年,AT&T 贝尔实验室的 Dennis 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 的第二字母作为这种语言的名字,这就是 C 语言。C 语言的发展历程示意图如图 1.1 所示。

C 语言问世后,1978 年由 AT&T 贝尔实验室正式发表了 C 语言。同时由 B. W. Kernighan 和 D. M. Ritchie 合著了《The C Programming Language》(称为《K&R》,也称之为《K&R》标准)。1983

年美国国家标准协会在此基础上制定了一个 C 语言标准,通常称之为 ANSI C。

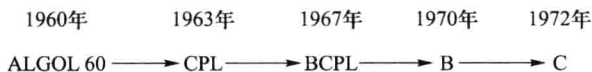


图 1.1 C 语言的发展历程示意图

1988 年 Brian Kernighan 和 Dennis Ritchie 修改了《K&R》,命名《The C Programming Language》(第二版),第二版涵盖了 ANSI C 语言标准。从此该书成为大学计算机教育有关 C 语言的经典教材。

在 ANSI C 标准确立之后,C 语言的规范在很长一段时期内都没有大的变动。1995 年 WG14 小组对 C 语言做了一些修改,1999 年发布了 ISO/IEC 9899:1999 标准,即 C99。但各个商业公司对 C99 的支持各不相同。

目前,几乎所有的开发工具都支持 ANSI C 标准,它是 C 语言用得最广泛的一个标准版本。目前 GCC 等一些商业编译器能够支持 C99,但仍有部分编译器不支持 C99。

目前最流行的 C 语言版本有: Borland Turbo C(或称 Turbo C)、Microsoft C(或称 MS C)和 AT&T C。这些 C 语言版本不仅实现了 ANSI C 标准,而且在此基础上各自作了一些扩充,使之更加方便、完美。

1.2.2 C 语言的特点

C 语言之所以发展迅速,成为最受欢迎的语言之一,主要因为它具有强大的功能。归纳起来,C 语言具有下列特点:

1. C 语言简洁、紧凑,使用方便、灵活

ANSI C 一共只有 32 个关键字、9 种控制语句,程序书写自由,主要用小写字母表示。

C 语言运算符丰富,共有 34 种,C 语言把括号、赋值、逗号等都作为运算符来处理,C 语言的各种运算符可以实现在其他高级语言中难以实现的运算。C 语言数据结构丰富,数据类型包括:整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等,能实现各种复杂的数据结构(如链表、树、栈等)的运算。

2. C 语言是中级语言

C 语言把高级语言的基本结构和语句与低级语言的实用性结合起来。它允许直接访问物理地址,可以像汇编语言一样对位、字节和地址进行操作,实现汇编语言的大部分功能,因此有人把它称为中级语言。

3. C 语言是结构式语言

结构式语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式使程序层次清晰,便于使用、维护以及调试。C 语言是以函数形式提供给用户的,这些函数可方便地被调用,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 语言功能齐全

C 语言具有各种各样的数据类型,并引入了指针概念,可使程序效率更高。另外 C 语言也具

有强大的图形功能,支持多种显示器和驱动器。而且计算功能、逻辑判断功能也比较强大,可以实现决策目的。

5. C语言适用范围广

C语言适合多种操作系统,如DOS、Windows、UNIX,也适用于多种机型。

6. C语言程序执行速度快

C语言的程序生成目标代码质量高,程序执行效率高,只比汇编生成的目标代码率低10%~20%。

7. C语法限制不太严格,程序设计自由度大

一般的高级语言语法检查比较严格,能检查出几乎所有的语法错误。而C语言放宽了对语法的检查,完全需要程序员仔细检查程序,确保其正确。“限制”与“灵活”是一对矛盾。限制严格就失去灵活性;而强调灵活就必然放松限制。对使用C语言编程的人,要求更加熟练程序设计。

1.2.3 C语言的作用

C语言是一种通用语言,几乎任何一种计算机上都有至少一种能用的C编译器。并且它的语法和函数库在不同的平台上都是统一的,这个特性对开发者来说很有吸引力。相对而言,C语言比其他语言难一些,主要是语法较复杂、灵活多样。从另一方面来看,提供的语法越多,其功能越强大,这也正是我们选择学习C语言的重要原因之一。

C语言的作用主要体现在以下几个方面:

1. 计算机系统底层开发

C语言主要用于计算机系统底层开发,如操作系统、驱动程序等,用C语言编写的操作系统有UNIX系列,包括Linux、FreeBSD等,基于Linux的嵌入式操作系统也是用C语言编写的,如uClinux;Windows内核大约有50%的C程序,X Window系统也是用C语言编写的;运行在UNIX字符环境中的很多应用软件都是用C语言编写的,如著名的Apache等。

2. 功能齐全,无所不能

几乎没有不能用C语言写出来的软件,没有不支持C语言的系统。C语言是一种通用语言,任何想让计算机做的事情,都可以用它编程实现。C语言既可以编写系统软件,又可以编写应用软件。使用C语言嵌入汇编语言,既可以完成C语言的功能,又可以完成汇编的功能。C语言还提供了与汇编语言、数据库等的接口。

3. 流行语言,借鉴于它

很多流行语言、新生语言都借鉴了C语言思想和语法,如:C++、Java、C#等。另外,学好C语言后,还可以很快地学会其他语言。

1.2.4 C语言程序的组成

1. C语言的基本组成

(1) 主函数 main()

main()表示C语言的一个特殊函数,称之为主函数。每个C语言程序都有且只有一个主函

数,整个程序从主函数开始执行,最后结束于主函数。

(2) 标识符与关键字

关键字是 C 语言预先规定的具有固定含义的一些单词,通常也称为保留字。

在程序中使用的常量、变量、语句标号以及函数的名称等统称为标识符。除库函数的函数名、头文件名和预处理命令由系统定义外,其余都由用户自定义。C 语言规定,标识符是以字母或下划线开头的并由字母(A~Z,a~z)、数字(0~9)、下划线(_)组成的字符序列。

(3) 基本数据类型

在 C 语言中,每个变量在使用之前必须定义其数据类型。所谓数据类型(Data Type)是按被定义变量的性质、表示形式、占据存储的空间和构造特点来划分的。在 C 语言中,数据类型可分为:基本数据类型、构造数据类型、指针类型、空类型四大类。如 int、char、float 等都属于基本数据类型,不同数据类型进行运算时,将自动转换或强制转换成相同的数据类型。

(4) 常量与变量

数据在程序中有常量和变量形式。在程序运行过程中,其值不能被改变的量称为常量,其值可以改变的量称为变量。常量可以直接使用,而变量则必须先定义后使用。

常量是可以不经声明而直接引用的不同的类型数据,还可以使用#define 宏定义符号常量。

变量代表具有特定属性的内存空间,它用来存放不同类型的数据。每个变量都有一个名字,变量的命名按照标识符的命名规则。变量代表相应的存储单元,由变量名、变量值、存储地址三要素组成。

(5) 基本表达式

表达式是用运算符将运算量(常量、变量、函数)连接起来的有意义的式子,每个表达式有一个值及其类型,表达式求值按运算符的优先级和结合性规定的顺序进行。

C 语言的运算符非常丰富,除了提供一般高级语言的算术、关系、逻辑运算符外,还提供赋值运算符、位操作运算符、自增自减运算符等,甚至数组下标、函数调用都可以作为运算符。

C 语言的表达式分为算术表达式、赋值表达式、逻辑表达式和逗号表达式。基本表达式是指算术表达式和赋值表达式。

(6) C 语句

C 语言程序的执行部分是由语句组成的,用“;”作为语句终止符。程序的功能也是由执行语句实现的。C 语句分为表达式语句、函数调用语句、控制语句、复合语句、空语句等。

(7) 格式化输入输出

计算机由主机(CPU、内存)、外部设备(输入输出设备)、I/O 接口组成。所谓输入输出是对计算机主机而言,从计算机向外部输出设备(如显示器)输出数据称为输出,从输入设备(如键盘)向计算机输入数据称为输入。

在 C 语言中,没有专门的输入输出语句,所有的输入输出操作都是通过调用库函数来实现。最常用的有格式输入函数 scanf() 和格式输出函数 printf()。

(8) 预处理命令

所谓预处理,是指在源程序文件中,加入预处理命令,以改进程序设计环境,扩大程序功能,提高编程效率。预处理命令是由 ANSI C 统一规定的,不是 C 语言本身的组成部分,不能直接对它们进行编译,因为编译程序不能识别它们,所以对源程序编译之前,先对这些命令进行预处理,

然后将预处理的结果和源程序一起进行编译处理,从而得到目标代码。编译预处理是C语言区别于其他高级语言的一个重要标志。

在程序中,凡是以“#”开头的均为预处理命令。常用预处理命令有三种:宏定义、文件包含、条件编译命令。

(9) 注释

C语言的注释符是以“/*”开头并以“*/”结尾的字符序列。在“/*”和“*/”之间的即为注释。注释在程序中起提示、解释程序的作用。注释可出现在程序中的任何位置。程序编译时,不对注释作任何处理。

2. C程序结构及特点

C语言是模块化程序设计语言,每个模块都是由函数完成的,C语言是函数式的语言,函数就是模块。使用顺序结构、分支结构、循环结构三种基本结构设计的程序必然就是结构化程序。C语言程序结构如图1.2所示。

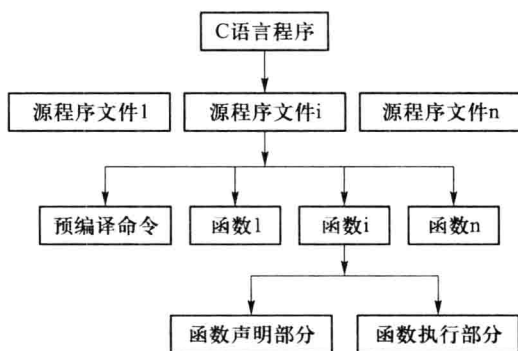


图 1.2 C语言程序结构

C语言源程序结构具有如下特点:

- (1) 一个C语言源程序有且只有一个主函数(main()函数)。
- (2) 函数由函数头和函数体构成,函数体由声明部分和执行部分构成。
- (3) 源程序中可以有预处理命令,预处理命令通常应放在源程序的最前面。
- (4) 每一个声明或语句都必须以分号结尾。但预处理命令、函数头和花括号“{”之后不能加分号。

1.2.5 C语言程序书写规则

从书写清晰,便于阅读、理解和维护的角度出发,在书写程序时应遵循以下规则:

- (1) 一个声明或一个语句占一行。
- (2) 用一对花括号“{}”括起来的部分,通常表示程序的某一层结构。“{}”一般与该结构语句的第一个字母对齐,并单独占一行。第一个花括号“{”后面也可以用空格分隔后直接书写语句,但花括号“}”原则上都是单独占一行。
- (3) 低一层次的语句或声明可比高一层次的语句或声明缩进若干空格后书写。以便看起来

更加清晰,增加程序的可读性。在编程时应力求遵循这些规则,以养成良好的编程风格。

(4) 在 C 语言程序中的所有符号均为英文符号(即半角符号)。

1.3 C 语言程序上机运行

1.3.1 C 语言程序上机步骤

【例 1.1】 第一个 C 语言程序。上机实现在屏幕上输出字符序列“Hello,World!”。

分析:

最简单的 C 语言程序包括预处理命令行、主函数行、函数体,函数体中使用 `printf()` 输出信息。源程序命名为 `LT1_1.c`,然后再使用 C 编程环境调试该程序。

程序源代码:

```
/* LT1_1.c */  
#include <stdio.h>           /* 预处理命令 */  
int main()                   /* 主函数 */  
{  
    printf("hello,world! \n"); /* 库函数,输出字符序列 */  
    return(0);  
}
```

程序的运行结果如下:

hello,world!

如何运行这个程序取决于所使用的系统,但一般过程包括:编辑、编译、链接、运行,如图 1.3 所示。

下面以 Turbo C 2.0(简称 TC 2.0)为例说明 C 语言程序的上机过程:

1. 编辑

编辑 C 语言源程序文件包括:建立或修改 C 语言源程序文件。利用 TC 集成环境或 Windows 提供的记事本等编辑工具,通过键盘输入例 1.1 的源代码,输入完成后保存源程序文件并命名为 `LT1_1.c`。

2. 编译

编译工作由专门的编译程序完成,编译后得到的目标代码文件的扩展名为 `.obj`。编译程序(如 TC 2.0 等)对 `LT1_1.c` 进行语法分析查错,翻译生成目标程序 `LT1_1.obj`。

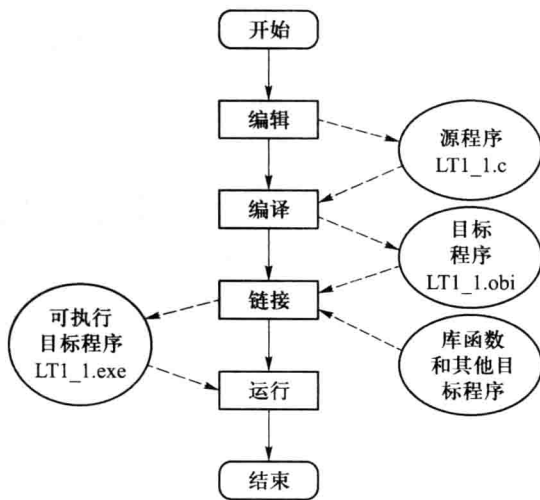


图 1.3 C 语言程序的上机过程示意图

3. 链接

编译后得到的目标代码虽然计算机能直接认识,但还不能直接执行。因为目标模块可能只是整个程序中的一个模块,并不是整个程序的完整模块;另外,在目标模块中往往使用了一些未在本模块中定义的函数等。因此,编译后还必须把各目标模块组合起来,同时把有关的各种代码装配在一起产生一个完整的可执行文件后,才能直接执行。即“链接”后得到可执行文件 LT1_1.exe。

4. 运行

经过编译和链接,最后得到了扩展名为 .exe 的可执行文件,就可以直接运行了。即在 DOS 下执行 LT1_1.exe 文件,也可以在集成环境中执行相应的命令。

1.3.2 TC 环境下的程序运行

以例 1.1 的程序 LT1_1.c 说明 TC 2.0 系统的调试运行。在 TC 2.0 环境中编辑、编译、链接和运行 C 源程序的步骤如下:

(1) 从任务栏开始,依次单击“开始”→“运行”。随后会弹出如图 1.4 所示的对话框。在对话框中输入 TC 编译环境所在的安装目录及其运行程序 TC.exe,或者单击“浏览(B)...”按钮直接在浏览窗口中查找 TC.exe 所在安装目录即可。

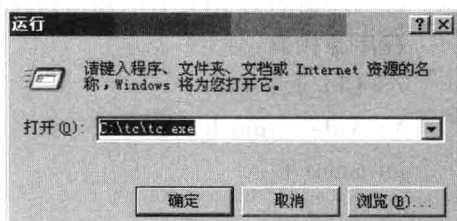


图 1.4 运行对话框

(2) 打开 TC 2.0 后,进入其界面,按 F10 键将光标移向菜单栏的“File”菜单(也可以使用 Alt + F 快捷键,快速定位到“File”菜单),按回车键,弹出 File 下拉菜单。用上下光标键将光标移向“New”菜单项,按回车键后新建一个编辑窗口,即可在编辑区编写源代码,如图 1.5 所示。



图 1.5 “File”下拉菜单

(3) 在编辑区输入源代码。在编辑过程中可以使用光标键上下左右移动光标,完成对程序