

21世纪高等院校计算机系列教材

C 语言

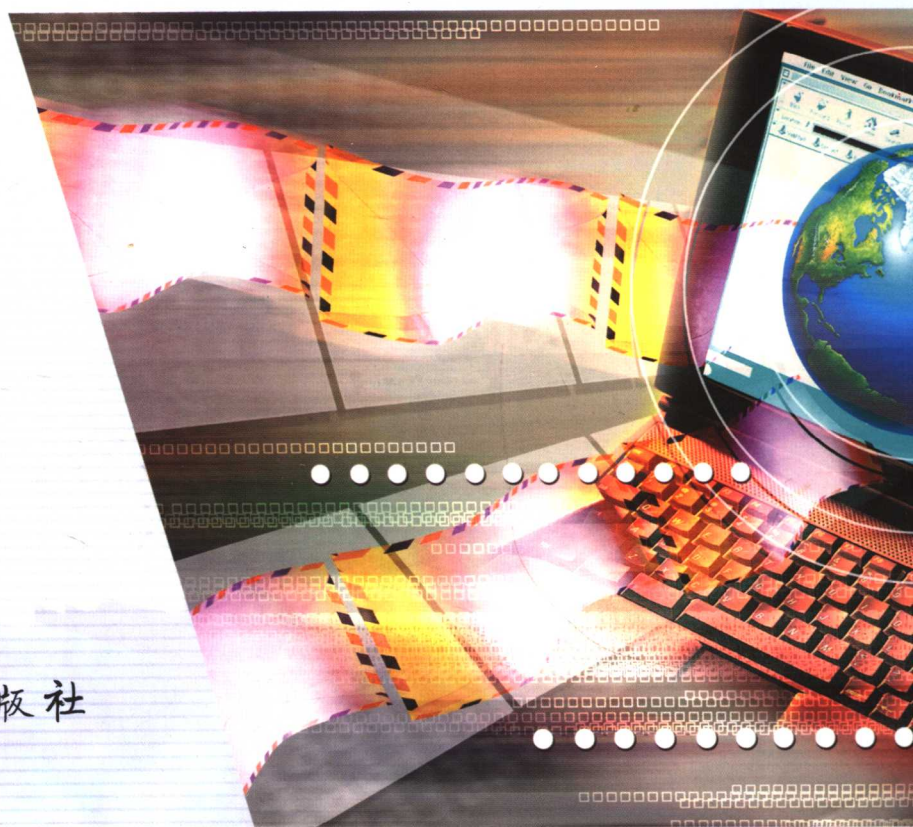
C YUYAN

CHENGXU SHEJI

程序设计

高平昊 杨国诗 吴长勤 李 胜 罗 耀 编著

安徽大学出版社



21世纪高等院校计算机系列教材

C 语言

C YUYAN

CHENGXU SHEJI

程序设计

高平昊 杨国诗 吴长勤 李 胜 罗 罹 编著

安徽大学出版社



内 容 简 介

本书作为高等院校计算机基础教学的教材,全面系统地介绍了C语言的语法规则和程序设计的基本方法、算法和应用,同时兼顾了全国高等院校计算机等级考试的需要,其内容覆盖了二及C语言程序设计教学(大纲)的要求。

本书可以作为高等院校非计算机专业或计算机专业的程序设计教材,也可作为广大软件开发人员和自学人员的参考书,同时也可作为参加全国计算机等级(水平)考试考点学校使用。

本书的电子幻灯片及习题参考答案,可访问 <http://www.ahupress.com.cn> 网站进行下载。

图书在版编目(CIP)数据

C语言程序设计/高平昊等编著.
—合肥:安徽大学出版社,2006.12
ISBN 7-81110-237-4

I. C… II. 高… III. C语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 148257 号

C 语言程序设计

高平昊 杨国诗 吴长勤 李胜 罗雁 编著

出版发行 安徽大学出版社
(合肥市肥西路3号 邮编 230039)
联系电话 发行部 0551-5107784
E-mail ahdxchps@mail.hf.ah.cn
网 址 www.ahupress.com.cn
责任编辑 李镜平
特约编辑 罗季重
封面设计 孟献辉

经 销 新华书店
印 刷 中国科学技术大学印刷厂
开 本 787×1092 1/16
印 张 15.375
字 数 380千
版 次 2006年12月第1版
印 次 2006年12月第1次印刷

ISBN 7-81110-237-4/T·100

定价 26.00 元

如有影响阅读的印装质量问题,请与出版社发行部联系调换

前 言

为了方便广大读者特别是初学者能够更容易、更好地学好 C 语言,把握其精髓,我们对目前比较出色的同类教材的特点及优点进行了分析和研究,博采众长,编写了 C 语言程序设计教材;力求实现教材权威性与实用性的完美结合,本教材具有以下特色:

1. 从计算机内存的角度来介绍 C 语言的数据类型。C 语言数据类型极其丰富,初学者往往注重的是对 C 语言语法的学习,而忽视对数据类型的把握,对数据类型的学习感到比较“虚”,不易正确理解和把握,特别是“指针”的概念更是难以理解,本书从计算机内存的角度深入浅出地介绍了 C 语言各种数据类型的特点,并以内存图示的形式直观、形象地反映数据类型在内存中的表示。

2. 从正反两方面来介绍 C 语言语法。为了便于读者对 C 语言语法规则的正确理解和把握,本书不仅从正面介绍了 C 语言的语法规则,而且还列举大量的反例来加深读者对语法规则的正确认识。对 C 语言中易混淆的语法规则还进行了总结和比较。

3. 加深对 C 语言函数的学习。对 C 语言的学习,读者不仅要掌握 C 语言的数据类型和语法规则,而且还应当对 C 语言提供的一些常用库函数做到牢记于心。本书从 C 语言上百个库函数中精心挑选出了一些常用的和实用的库函数,并结合有关章节的内容进行了详细的介绍,而且还应用于实例程序中。

4. 配备大量经典实例程序,对每行语句作详尽的解释。为了帮助读者对 C 语言各章节知识的理解和提高程序设计的应用能力,在各章节都配备有大量的精心设计的实例程序,并尽可能对实例程序中的每一行语句都作了详尽的解释。

5. 突出应用性。各章配备大量的习题,习题类型丰富,难度各异,具有广泛的代表性和实践性。为了帮助读者加深对各章节内容学习的巩固,每章都配备有题型丰富、代表性强的大量习题,并在最后一章给出了几个综合应用实例。

本书作为高等院校计算机基础教学的教材,共分十章,全面、系统地介绍了 C 语言的语法规则和程序设计的基本方法、算法和应用,同时兼顾了全国高等院校计算机等级考试的需要,其内容覆盖了二级 C 语言程序设计教学(大纲)的要求。本书可以作为高等院校非计算机专业或计算机专业的程序设计教材,也可作为广大软件开发人员和自学人员的参考书,同时可作为参加全国计算机等级(水平)考试考点学校使用。

参与本书编写的有:高平旱、杨国涛、吴长勤、李胜、罗罹。全书由高平旱审定。

本书在编写过程中,得到了安徽省高校同行专家们的大力支持和帮助,在此表示衷心的感谢!

编 者

2006 年 12 月

目 录

第 1 章 C 语言程序概述	(1)
1.1 C 语言的发展	(1)
1.2 C 语言的特点	(2)
1.3 简单的 C 语言程序介绍	(3)
1.4 Turbo C 2.0 的上机过程	(5)
1.4.1 编辑、编译、连接、执行及调试程序的概念	(5)
1.4.2 Turbo C 2.0 集成开发环境的使用	(7)
1.5 程序设计算法简介	(10)
1.5.1 算法的概念	(10)
1.5.2 算法的特性	(10)
1.5.3 算法的表示	(11)
习题 1	(13)
第 2 章 数据类型与运算符	(15)
2.1 C 语言的数据类型概述	(15)
2.2 标识符、常量和变量	(16)
2.2.1 标识符	(16)
2.2.2 常量	(16)
2.2.3 变量	(17)
2.3 C 语言的基本数据类型	(18)
2.3.1 整型数据	(18)
2.3.2 实型数据	(19)
2.3.3 字符型数据	(20)





2.4 变量赋初值与类型转换	(24)
2.4.1 变量赋初值	(24)
2.4.2 类型转换	(24)
2.5 运算符和表达式	(26)
2.5.1 算术运算符和算术表达式	(26)
2.5.2 关系运算符和关系表达式	(27)
2.5.3 逻辑运算符和逻辑表达式	(28)
2.5.4 条件运算符和条件表达式	(30)
2.5.5 赋值运算符和赋值表达式	(31)
2.5.6 逗号运算符和逗号表达式	(32)
2.5.7 C语言运算符的优先级与结合性	(33)
习题2	(34)
 第3章 结构化程序设计	(36)
3.1 顺序程序设计	(36)
3.1.1 数据的输入和输出	(36)
3.1.2 printf()函数	(37)
3.1.3 scanf()函数	(41)
3.1.4 getchar()函数与 putchar()函数	(44)
3.1.5 基本语句及复合语句	(45)
3.1.6 程序示例	(46)
3.2 分支程序设计	(48)
3.2.1 if语句的结构及应用	(48)
3.2.2 switch语句的结构及应用	(52)
3.3 循环型程序设计	(54)
3.3.1 while语句的结构及应用	(54)
3.3.2 do...while语句的结构及应用	(56)
3.3.3 for语句的结构及应用	(57)
3.3.4 goto语句	(59)
3.3.5 break语句及continue语句	(60)
习题3	(61)
 第4章 数 组	(65)
4.1 一维数组	(65)



4.1.1 一维数组的定义	(65)
4.1.2 一维数组的初始化	(66)
4.1.3 一维数组的引用	(66)
4.1.4 一维数组的应用举例	(67)
4.2 二维数组	(71)
4.2.1 二维数组的定义和引用	(71)
4.2.2 二维数组的初始化	(71)
4.2.3 二维数组在内存中的存放	(72)
4.2.4 二维数组的应用举例	(72)
4.3 字符数组	(74)
4.3.1 字符数组的定义	(74)
4.3.2 字符数组的初始化	(74)
4.3.3 字符数组的整体操作	(75)
4.3.4 常用的字符串处理函数	(76)
4.4 数组应用举例	(79)
习题 4	(82)

第5章 函 数

5.1 函数的引入	(87)
5.2 函数定义的形式	(90)
5.3 函数的参数和返回值	(92)
5.3.1 形式参数和实际参数	(92)
5.3.2 函数的返回值	(93)
5.4 函数的调用	(94)
5.4.1 函数调用的一般形式	(94)
5.4.2 函数调用的方式	(95)
5.4.3 对被调用函数的说明	(95)
5.5 函数的递归调用	(97)
5.6 数组作为函数的参数	(101)
5.6.1 数组元素作为函数参数	(101)
5.6.2 数组名作为函数的形参和实参	(101)
5.7 数据存储类别	(103)
5.7.1 自动变量	(103)





5.7.2	寄存器变量	(104)
5.7.3	外部变量	(104)
5.7.4	静态变量	(106)
5.8	编译预处理	(108)
5.8.1	概 述	(108)
5.8.2	宏定义	(108)
5.8.3	无参数宏定义	(109)
5.8.4	带参数宏定义	(111)
5.8.5	文件包含	(115)
5.8.6	条件编译	(116)
习题 5		(117)
 第 6 章 指 针		(125)
6.1	指针概述	(125)
6.1.1	地址与指针	(125)
6.1.2	指针变量的定义与引用	(126)
6.2	指针与数组	(130)
6.2.1	指针与一维数组	(130)
6.2.2	指针与二维数组	(133)
6.2.3	指针与字符串	(134)
6.2.4	指针数组与多级指针	(136)
6.3	指针与函数	(138)
6.3.1	指针作为函数参数	(138)
6.3.2	指向函数的指针	(142)
6.3.3	返回指针值的函数	(144)
6.3.4	main 函数中的参数	(145)
习题 6		(146)
 第 7 章 结构体、共用体和枚举类型		(151)
7.1	结构体类型	(151)
7.1.1	结构体类型变量的定义	(151)
7.1.2	结构体类型变量的引用	(154)



7.1.3 结构体数组	(156)
7.1.4 结构体变量与指针	(158)
7.1.5 结构体变量与函数	(160)
7.1.6 动态分配与链表	(161)
7.2 共用体类型	(175)
7.2.1 共用体的定义及引用	(175)
7.2.2 共用体变量的应用	(176)
7.3 枚举类型数据	(178)
7.3.1 枚举变量的定义与声明	(178)
7.3.2 枚举变量的赋值与使用	(178)
7.4 用 TYPEDEF 定义类型	(179)
习题 7	(180)

第8章 位运算..... (184)

8.1 位运算概念	(184)
8.2 位运算符使用方法	(184)
8.2.1 按位与运算符(&)	(185)
8.2.2 按位或运算符()	(185)
8.2.3 按位异或运算符(^)	(185)
8.2.4 按位取反运算符(~)	(186)
8.2.5 左移运算符(<<)	(187)
8.2.6 右移运算符(>>)	(187)
8.3 位运算示例	(188)
习题 8	(189)

第9章 文件..... (191)

9.1 文件概述	(191)
9.1.1 文件的概念	(191)
9.1.2 缓冲文件系统(标准 I/O)和非缓冲文件系统(系统 I/O)	(192)
9.1.3 文件(FILE)类型指针	(193)
9.2 文件的打开与关闭	(193)
9.2.1 文件的打开(fopen 函数)	(193)





9.2.2 文件的关闭(fclose 函数)	(195)
9.3 文件的顺序读写	(195)
9.3.1 输入和输出一个字符	(195)
9.3.2 输入和输出一个字符串	(199)
9.3.3 按“记录”的方式输入和输出	(201)
9.3.4 格式化的输入和输出	(202)
9.4 文件的定位与随机读写	(204)
9.4.1 文件的定位	(204)
9.4.2 随机读写	(205)
9.5 文件操作的检测	(206)
9.6 非缓冲文件系统与 C 语言库文件	(207)
9.6.1 非缓冲文件系统	(207)
9.6.2 C 语言库文件	(208)
习题 9	(209)
第 10 章 综合应用程序实例	(212)
10.1 文件加密技术	(212)
10.2 学生成绩管理系统	(214)
10.3 简单的接口程序	(222)
10.4 图形模式下的文本输出及屏幕操作	(228)
附录 1 常用字符和 ASCII 代码对照表	(233)
附录 2 运算符优先级和结合方向	(234)
附录 3 常用库函数	(235)





第1章 C语言程序概述

1.1 C语言的发展

计算机要完成某一特定的任务,必须执行一系列计算机指令。程序就是由这样的一系列计算机指令组成的。程序设计就是针对某一要处理的问题,设计出解决该问题的计算机指令序列。因此程序设计是一项创造性的工作。进行程序设计必须借助于语言来描述,这就是程序设计语言。只有严格按照程序设计语言的语法规则来书写程序,才能让计算机正确执行指令序列,完成指定的任务。程序设计语言分为低级语言和高级语言两大类。低级语言直接面向机器,如机器语言和汇编语言;高级语言独立于机器,用高级语言编写的程序在不同的机器上必须使用不同的翻译程序。C语言程序是一种高级语言程序,它必须被翻译成计算机能识别的语言,即机器语言,才能在计算机上运行。

计算机程序设计语言从诞生至今已经历了4代:

1. 机器语言

由一组计算机可识别的0和1所组成的序列构成。

2. 汇编语言

把上述的一组0和1的序列用一条助记指令来代替。

3. 面向过程的高级语言

上述两种语言都是面向机器的,而不是面向解决问题的过程。使用高级语言来编写程序时,人们不必去熟悉计算机的内部结构和工作原理,而把主要的精力放在算法的描述上,因此这种语言又称为算法语言。从1954年出现FORTRAN语言后,至今世界上已出现了上千种高级语言。

4. 非过程化的高级语言

面向过程的高级语言编写成程序实际上是提出问题“做什么”,然后再去构造“怎么做”的解题过程,而用非过程化的高级语言时,人们只要提出“做什么”即可,而“怎么做”的过程则由计算机去解决。

C语言是1972年由美国的Dennis Ritchie设计发明的,它由早期的编程语言发展演变而来,可用于编写系统软件,还可用于编写应用软件,是广泛流行的高级语言。

C语言诞生以前,系统软件主要是用汇编语言编写的。汇编语言可以实现对计算机硬件的直接操作,但是它依赖于计算机硬件,其可读性和可移植性都很差。但一般的高级语言却难以实现对计算机硬件的直接操作,所以人们希望有一种计算机语言能有高级语言的优





点,同时又有低级语言的功能,C语言就是在这种背景下产生的。

1970年,AT&T贝尔实验室的Ken Thompson根据BCPL(basic combined programming language)设计出较先进的并取名为B的语言,最后导致了C语言的问世。随着微型计算机的普及,出现了许多C语言版本。由于没有统一的标准,使得这些C语言之间出现了一些不一致的地方。为了改变这种情况,美国国家标准研究所(ANSI)为C语言制定了一套ANSI标准,成为现行的C语言标准。目前,在微机上广泛使用的C语言编译系统有Microsoft C、Turbo C、Borland C等。虽然它们的基本部分都是相同的,但还是有一些差异,所以还须注意所使用的C编译系统的特点和规定。

1.2 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,主要因为它具有强大的功能。许多著名的系统软件和应用软件,如UNIX、LINUX、PLUS、FOXBASE等都是由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语言的优势了,如PC-DOS、WORDSTAR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. 语言简洁、紧凑,使用方便、灵活

C语言一共只有32个关键字,9种控制语句,压缩了一切不必要的成分。程序书写形式自由,主要使用小写字母。

2. 运算符丰富

C语言的运算符包含的范围很广,共有34种运算符。C语言把括号、赋值、强制类型转换等都作为运算符处理,从而使C语言的运算类型极为丰富,表达式类型多样化。

3. 数据结构丰富,具有现代化语言的各种数据结构

C语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等,能用来实现各种复杂的数据结构。

4. 具有结构化的控制语句

用函数作为程序的模块单位,便于实现程序的模块化。C语言是理想的结构化程序设计语言,符合现代编程风格要求。

5. 语法限制不太严格,程序设计自由度大

数据类型可相互通用(整型和字符型),对下标越界不作检查,由程序员自己保证程序的正确性。

6. C语言允许直接访问物理地址,能进行位操作,可以直接对硬件进行操作

C语言可与机器硬件打交道,直接访问内存地址,具有“高”、“低”级语言之功能。

7. 生成目标代码质量高,程序执行效率高

仅比汇编语言目标代码效率低10%~20%。

8. 用C语言写的程序可移植性好

程序基本不作修改就能用于各种计算机和各种操作系统。





1.3 简单的 C 语言程序介绍

为了说明 C 语言源程序结构的特点,先看以下几个程序。这几个程序由简到难,表现了 C 语言源程序在组成结构上的特点。可从这些例子中了解到编写一个 C 语言源程序的基本部分和书写格式。

【例 1.1】 第一个 C 语言程序。

```
main()
{
    printf("This is my first C program.\n");
}
```

程序的功能是输出一行信息:This is my first C program.

① 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”main 的函数头。

② 用 { } 括起来的是“主函数”main 的函数体。main 函数中的所有语句都在这一对 { } 之间,也就是说 main 函数的所有操作都在 main 函数体中。

③ “主函数”main 中只有一条语句,它是 C 语言的库函数,功能是用于程序的输出(显示在屏幕上),本例用于将一个字符串“This is my first C program.\n”的内容输出。即在屏幕上显示:

This is my first C program.

④ 每条语句用“;”号结束。

【例 1.2】 计算两数之和并输出。

```
main()    /* 计算两数之和 */
{
    int m,n,sum;    /* 这是定义变量 */
    m=123;n=78;    /* 以下 3 行为 C 语句 */
    sum=m+n;
    printf("sum=%d\n",sum);
}
```

程序的功能是计算两数之和,并输出结果。

同样此程序也必须包含一个 main 函数作为程序执行的起点。{ } 之间为 main 函数的函数体,main 函数所有操作均在 main 函数体中。

① /* */ 括起来的部分是一段注释,注释只是为了改善程序的可读性,在编译、运行时不起作用(事实上编译时会跳过注释,目标代码中不会包含注释)。注释可以放在程序任何位置,并允许占用多行,只是需要注意“/*”、“*/”匹配,不允许嵌套注释。

② int m,n,sum 是变量声明。声明了 3 个具有整数类型的变量 m,n,sum。C 语言的变量必须先声明再使用。

③ m=123;n=78;是两条赋值语句。将整数 123 赋给整型变量 m,将整数 78 赋给整型变量 n。m,n 两个变量值分别为 123,78。注意这是两条赋值语句,每条语句均用“;”结束。





也可以将两条语句写成两行,即:

```
m=123;
```

```
n=78;
```

由此可见 C 语言程序的书写可以相当随意,但是为了保证容易阅读要遵循一定的规范。

④ `sum=m+n`;是将 `m,n` 两变量内容相加,然后将结果赋值给整型变量 `sum`。此时 `sum` 的值为 201。

⑤ `printf("sum=%d\n",sum)`;是调用库函数输出 `sum` 的结果。`%d` 为格式控制符,表示 `sum` 的值以十进制整数形式输出。程序运行后,输出(显示):`sum=201`

【例 1.3】 求 `x` 的正弦值并输出。

```
#include<math.h>           /* include 称为文件包含命令,扩展名为.h 的文件称为头文件 */
#include<stdio.h>
main()
{
    double x,s;              /* 定义两个实数变量 */
    printf("input number:\n"); /* 显示提示信息 */
    scanf("%lf",&x);          /* 从键盘获得一个实数 x */
    s=sin(x);                /* 求 x 的正弦,并把它赋给变量 s */
    printf("sin of %lf is %lf\n",x,s); /* 输出程序运算结果 */
}
```

程序的功能是从键盘输入一个数 `x`,求 `x` 的正弦值,然后输出结果。

① 在 `main()` 之前的两行称为预处理命令。预处理命令还有其他几种,这里的 `include` 称为文件包含命令,其意义是把尖括号 `<>` 或引号 `"` 内指定的文件包含到本程序来,成为本程序的一部分。被包含的文件通常是由系统提供的,其扩展名为 `.h`。因此也称为头文件或首部文件。C 语言的头文件中包括了各个标准库函数的函数原型。因此,凡是在程序中调用一个库函数时,都必须包含该函数原型所在的头文件。在本例中,使用了三个库函数:输入函数 `scanf`,正弦函数 `sin`,输出函数 `printf`。`sin` 函数是数学函数,其头文件为 `math.h` 文件,因此在程序的主函数前用 `include` 命令包含了 `math.h`。`scanf` 和 `printf` 是标准输入输出函数,其头文件为 `stdio.h`,在主函数前也用 `include` 命令包含了 `stdio.h` 文件。

② C 语言规定对 `scanf` 和 `printf` 这两个函数可以省去对其头文件的包含命令。所以在本例中也可以删去第二行的包含命令 `#include<stdio.h>`。

综合上述 3 个例子,我们对 C 语言程序的基本组成和形式有了一个初步了解:

1. C 语言程序由函数构成(C 语言是函数式的语言,函数是 C 语言程序的基本单位)

一个 C 语言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 语言程序的基本单位。

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。C 语言是函数式的语言,程序的全部工作都是由各个函数完成。编写 C 语言程序就是编写一个个函数。

C 语言函数库非常丰富,ANSI C 提供 100 多个库函数,Turbo C 提供 300 多个库函数。





2. main 函数(主函数)是每个程序执行的起始点

一个 C 程序总是从 main 函数开始执行,而不论 main 函数在程序中的位置。可以将 main 函数放在整个程序的最前面,也可以放在整个程序的最后,或者放在其他函数之间。

3. 一个函数由函数首部和函数体两部分组成

(1) 函数首部:返回值类型 函数名(函数参数)

注意:

函数可以没有参数,但是后面的一对()不能省略,这是格式的规定。

(2) 函数体:函数首部下用一对{}括起来的部分。如果函数体内有多个{},最外层是函数体的范围。函数体一般包括声明部分和执行部分。

{

[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)。

}

4. C 语言程序书写格式自由

一行可以写几个语句,一个语句也可以写在多行上。

C 语言程序没有行号,也没有 FORTRAN, COBOL 那样严格规定书写格式(如语句必须从某一列开始)。每条语句的最后必须有一个分号“;”表示语句的结束。

5. 可以使用“/* */”对 C 语言程序中的任何部分作注释

注释可以提高程序可读性,使用注释是编程人员的良好习惯。

编写好的程序往往需要修改、完善,事实上没有一个应用系统是不需要修改、完善的。很多人会发现自己编写的程序在经历了一段时间以后,由于缺乏必要的文档、注释,最后连自己都很难再读懂。需要花费大量时间重新思考、理解原来的程序,这浪费了大量的时间。如果一开始编程就对程序进行注释,刚开始麻烦一些,但日后可以节省大量的时间。

一个实际的系统往往是多人合作开发,程序文档、注释是其中重要的交流工具。

6. C 语言本身不提供输入/输出语句,输入/输出的操作是通过调用库函数 (scanf, printf) 完成

输入/输出操作涉及具体计算机硬件,把输入/输出操作放在函数中处理,可以简化 C 语言和 C 语言的编译系统,便于 C 语言在各种计算机上实现。不同的计算机系统需要对函数库中的函数做不同的处理,以便实现同样或类似的功能。

不同的计算机系统除了提供函数库中的标准函数外,还按照硬件的情况提供一些专门的函数。因此不同计算机系统提供的函数数量、功能会有一定差异。

1.4 Turboc 2.0 的上机过程

1.4.1 编辑、编译、连接、执行及调试程序的概念

1. 编 辑

程序员用 C 语言编写的程序称为 C 的源程序(一般为 *.C 的文件)。编辑就是编写源程序的过程,它包括新建一个源程序文件或修改已有的源程序文件,它的操作有插入、删除、





修改源程序。除了 Turbo C 2.0 集成开发环境能够编辑源程序外,还可使用 DOS 环境中的 EDIT、CCED、WPS 或 Windows 环境中的 WORD、记事本、写字板等常用的编辑软件来编辑 C 的源程序,存盘时应采用纯文本方式保存文件。

2. 编译

源程序是以纯文本形式存储的,必须翻译成机器语言才能被计算机识别。完成这一翻译工作的就是所谓的编译程序。源程序经过编译程序翻译成等价的机器语言程序——目标程序(一般为 *.OBJ 的文件),这一翻译过程称为编译。Turbo C 2.0 集成开发环境带有编译程序。

3. 连接

如果编译成功,还应将目标程序和 C 的库函数连接成可执行程序(一般为 *.EXE 的文件),并存储在计算机的存储设备中,以便执行。负责目标程序和库函数连接工作的程序称为连接程序。Turbo C 2.0 集成开发环境带有连接程序。

4. 执行

源程序经过编译、连接成为可执行文件(扩展名为 .exe 或 .com)后,一般存于计算机的外存中。所谓执行程序就是把一可执行文件从外存调入计算机内存,并由计算机完成该程序预定的功能,如完成输入数据、处理数据及输出结果等任务。执行程序又称为运行程序。

5. 调试

程序调试是指对程序进行查错和排错。最常见的错误是编译错误和逻辑错误。

源程序中难免会存在错误,错误一般可分为 4 类:

(1) 编译错误:程序不符合 C 语言语法规则,在编译时将出错,编译错误包括语法错误(error)和警告错误(warning)。例如某一变量未定义先使用,则会出现语法错误。又如某变量未赋初值就用来求和,则会出现警告错误。

(2) 逻辑错误:一个程序在编译时没有出现错误,执行后仍然得不到正确结果,这是由于在算法的设计过程或程序的表达式中存在错误,如表达式书写错误、程序控制流程错误等。

(3) 运行错误:程序执行时在某些特殊情况发生的错误,如变量越界、除零错误等。

(4) 连接错误:把目标程序连接成可执行程序时出现错误。如找不到库文件错误等。

上述几个步骤在 Turbo C 2.0 集成开发环境中可以很方便地实现。上机操作的整个过程如图 1.1 所示。

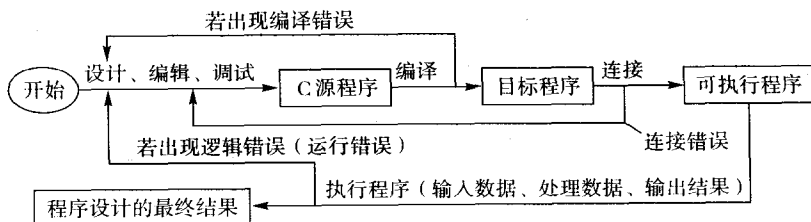


图 1.1 C 语言上机操作过程

运行一个 C 语言程序的一般过程:

- ① 启动 TC, 进入 TC 集成环境。
- ② 输入与编辑源程序。编辑后文件的扩展名为:.c。



③ 对源程序进行编译。如果编译成功,则可进行下一步操作;否则,返回②修改源程序,再重新编译,直至编译成功。编译成功后文件的扩展名为: .obj。

④ 与库函数进行连接。如果连接成功,则可进行下一步操作;否则,根据系统的错误提示,进行相应修改,再重新连接,直至连接成功。连接成功后的可执行文件的扩展名为: .exe。

⑤ 运行可执行的目标程序。通过观察程序运行结果,验证程序的正确性。如果出现逻辑错误,则必须返回②修改源程序,再重新编译、连接和运行,直至程序正确。

1.4.2 Turbo C 2.0 集成开发环境的使用

Turbo C 2.0 是 Borland 公司开发的一个 C 语言集成开发环境。可以在 Turbo C 中完成 C 语言程序的编辑、编译、连接、运行、调试。该系统是 DOS 操作系统支持下的软件,在 Windows 环境下,可以在 DOS 窗口下运行。

进入 Turbo C 2.0 集成开发环境中后,屏幕上显示如图 1.2 所示的界面。

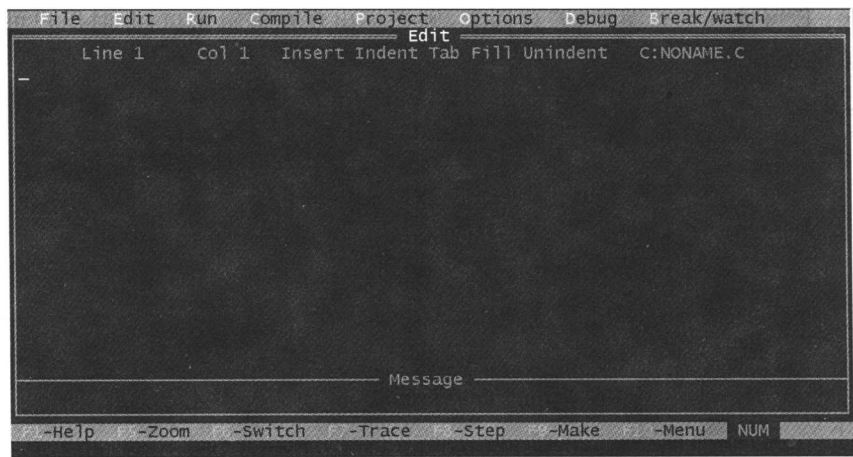


图 1.2 Turbo C 2.0 集成开发环境

其中顶端一行为 Turbo C 2.0 主菜单,中间窗口为编辑区,接下来是信息窗口,最底下一行为参考行。这四个窗口构成了 Turbo C 2.0 的主屏幕,以后的编程、编译、调试以及运行都将在这个主屏幕中进行。

除 Edit 外,其他各项均有子菜单,只要用 Alt 加上某项中第一个字母,就可进入该项的子菜单中。常用的菜单及其功能简介如下:

1. File 菜单

按 Alt+F 可进入 File 菜单,File 菜单的子菜单共有 9 项。

(1) Load: 装入一个文件,可用类似 DOS 的通配符(如 *.C)来进行列表选择。也可装入其他扩展名的文件,只要给出文件名(或只给路径)即可。该项的热键为 F3,即只要按 F3 即可进入该项,而不需要先进入 File 菜单再选此项。

(2) Pick: 将最近装入编辑窗口的 8 个文件列成一个表让用户选择,选择后将该程序装入编辑区,并将光标置在上次修改过的地方。其热键为 Alt+F3。

(3) New: 新建文件,缺省文件名为 NONAME.C,存盘时可改名。

(4) Save: 将编辑区中的文件存盘,若文件名是 NONAME.C 时,将询问是否更改文件

