

C YUYANCHENGXUSHEJI

DEXIANDAIFANGFAYANJIU

# C语言程序设计的 现代方法研究

主编 董军堂 黄顺强 王冬



中国水利水电出版社  
www.waterpub.com.cn

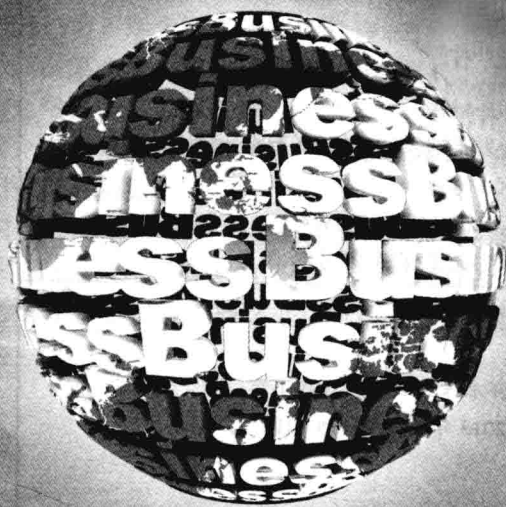
C YUYANCHENGXUSHEJI

DEXIANDAIFANGFAYANJIU

# C语言程序设计的 现代方法研究

主 编 董军堂 黄顺强

副主编 刘 宾 黎才茂 轩兴涛 符 天



中国水利水电出版社  
www.waterpub.com.cn

## 内 容 提 要

本书从应用出发,系统地介绍了C语言程序设计的现代方法,并进行了深入的研究,其主要内容包括绪论、C语言数据类型与运算分析、C语言控制结构编程、利用数组实现批量数据处理、利用函数实现模块化程序设计、利用指针实现动态存储分配、自定义数据类型——结构体、共用体与枚举和C语言中的文件及其操作以及C语言图形程序设计研究、常见错误分析与程序调试和C语言控制结构程序设计等知识。本书可作为高等院校各专业C语言程序设计课程的教材,同时也可作为工程技术人员和计算机爱好者的参考用书。

## 图书在版编目(CIP)数据

C语言程序设计的现代方法研究/董军堂,黄顺强,  
王冬主编.--北京:中国水利水电出版社,2013.8  
ISBN 978-7-5170-1162-0

I. ①C… II. ①董… ②黄… ③王… III. ①  
C语言-程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2013)第189947号

策划编辑:杨庆川 责任编辑:杨元泓 封面设计:崔 蕾

书 名	C语言程序设计的现代方法研究
作 者	主 编 董军堂 黄顺强 王 冬
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座100038) 网址:www.waterpub.com.cn E-mail:mchannel@263.net(万水) sales@waterpub.com.cn
经 售	电话:(010)68367658(发行部)、82562819(万水) 北京科水图书销售中心(零售) 电话:(010)88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排 版	北京鑫海胜蓝数码科技有限公司
印 刷	三河市国源印刷厂
规 格	184mm×260mm 16开本 25印张 640千字
版 次	2013年8月第1版 2013年8月第1次印刷
印 数	0001—3000册
定 价	88.00元

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换

版权所有·侵权必究



## 前 言

C 语言是被人们广泛应用的一种计算机程序设计语言。它于 1972 年由美国贝尔实验室的 D. M. Ritchie 推出。C 语言是一种面向过程的程序设计语言,对于软件工程师来说,它是今后从事面向对象的 C++ 或者 VC++ 程序设计的基础,而对于硬件工程师来说,它是从事以单片机或者嵌入式内核处理器等为核心的硬件产品开发过程中程序设计的必要工具。所以,要想将来从事计算机及其相关工作的技术人员必须学好 C 语言。

C 语言既具备高级语言的特点,又具备直接操作计算机硬件的能力,并且因为其丰富灵活的控制和数据结构、清晰的程序结构、简洁高效的语句表达和良好的可移植性而拥有大量的使用者。

本书在编写方面力求突出以下特点:

- (1) 力求做到难易适当、深入浅出、融会贯通;
- (2) 讲解理论重点、层次分明、通俗易懂;
- (3) 案例丰富,有利于读者掌握所学理论知识;
- (4) 体系完整,内容先进;
- (5) 取舍合理,该有则有,否则便无。

全书共分 11 章,第 1 章简单介绍 C 语言编程所需要的预备知识;第 2 章介绍 C 语言数据类型与运算分析的相关知识;第 3 章介绍 C 语言的顺序、选择和循环流程控制语句;第 4 章介绍利用数组实现批量数据处理;第 5 章介绍利用函数实现模块化程序设计;第 6 章介绍利用指针实现动态存储分配;第 7 章介绍自定义数据类型——结构体、共用体与枚举;第 8 章介绍 C 语言中的文件及其操作;第 9 章介绍 C 语言图形程序设计研究;第 10 章介绍常见错误分析与程序调试;第 11 章介绍面向对象程序设计基础。

本书在编写过程中得到了许多同行专家的支持和帮助,在此表示衷心的感谢;编写时参考了大量的相关著作和文献资料,选用了其中的部分内容,在此向有关作者表示由衷的感谢。

由于编者水平有限,书中错误或不当之处在所难免,热忱欢迎大家批评指正。

编者  
2013 年 6 月

# 目 录

前言

第 1 章 绪论	1
1.1 程序设计语言	1
1.2 C 语言概述	2
1.3 C 语言程序的基本结构	4
1.4 C 语言程序的开发环境	9
1.5 结构化程序设计算法	31
第 2 章 C 语言数据类型与运算分析	34
2.1 C 语言的数据类型	34
2.2 常量与变量	36
2.3 运算符与表达式	43
2.4 数据类型转换	53
第 3 章 C 语言控制结构程序设计	59
3.1 顺序结构程序设计	59
3.2 选择结构程序设计	72
3.3 循环结构程序设计	91
第 4 章 利用数组实现批量数据处理	104
4.1 数组概述	104
4.2 一维数组	104
4.3 二维数组	112
4.4 字符数组	116
第 5 章 利用函数实现模块化程序设计	124
5.1 函数的概念	124
5.2 函数的调用	127
5.3 函数的嵌套调用	139
5.4 函数的递归调用	141
5.5 变量的存储类别与作用域	144
5.6 内部函数与外部函数	146
5.7 编译预处理	149
第 6 章 利用指针实现动态存储分配	162
6.1 指针的概念	162
6.2 指针变量的定义与引用	163
6.3 指针与数组	165
6.4 指针与函数	172

6.5	指针与字符串 .....	176
6.6	指针数组和指向指针的指针 .....	179
6.7	动态内存分配 .....	184
<b>第 7 章</b>	<b>自定义数据类型——结构体、共用体与枚举 .....</b>	<b>188</b>
7.1	结构体类型 .....	188
7.2	结构体变量 .....	189
7.3	结构体数组 .....	196
7.4	结构体指针 .....	205
7.5	结构体与函数 .....	212
7.6	链表 .....	217
7.7	共用体 .....	245
7.8	枚举类型 .....	253
7.9	用 typedef 定义类型 .....	259
<b>第 8 章</b>	<b>C 语言中的文件及其操作 .....</b>	<b>262</b>
8.1	文件概述 .....	262
8.2	文件的打开与关闭 .....	264
8.3	文件的读写 .....	267
8.4	文件的定位 .....	276
8.5	文件的出错检测 .....	279
8.6	位运算 .....	280
<b>第 9 章</b>	<b>C 语言图形程序设计研究 .....</b>	<b>288</b>
9.1	图形程序设计概述 .....	288
9.2	基本图形函数 .....	288
9.3	图形填充 .....	295
9.4	图形模式下的文本输出 .....	301
9.5	视口与视口函数 .....	307
9.6	独立图形运行程序的建立 .....	311
<b>第 10 章</b>	<b>常见错误分析与程序调试 .....</b>	<b>315</b>
10.1	常见错误分析 .....	315
10.2	程序测试与调试 .....	326
<b>第 11 章</b>	<b>面向对象程序设计基础 .....</b>	<b>345</b>
11.1	面向对象程序设计概述 .....	345
11.2	类与对象 .....	351
11.3	继承与派生 .....	363
11.4	重载 .....	372
11.5	虚函数 .....	379
11.6	I/O 流 .....	381

# 第 1 章 绪 论

## 1.1 程序设计语言

### 1. 程序设计语言及其功能

程序设计语言为一种指挥计算机的工具？一种程序员之间交流的方式？一种表述高层设计的媒介？一种算法的记述方式？一种表述观念间关系的途径？一种试验工具？还是一种控制计算机化的设备的途径？实际上，一种通用程序设计语言必须集上述所有这些东西于一身，才能为用户更好地服务。

语言是一种交流工具，发明程序设计语言的目的就是为了使计算机更容易使用。程序设计语言是计算机软件中非常独特的一部分，属于系统软件，但又和应用软件息息相关。其功能是使人类能够用某些命令、指令让计算机为人类进行数值、逻辑运算。

### 2. 程序设计语言按对硬件依赖的程度分类

人们将程序设计语言按其对于计算机硬件依赖的程度分为低级语言和高级语言。低级语言包括机器语言和汇编语言。

机器语言为一台机器本身的语言，使此台机器可直接响应的指令的记述形式，是唯一可被计算机直接执行的语言。面向机器程序语言的指令由许多的 0 和 1 组成，一条计算机指令指示计算机一次完成一个最基本的操作。因为该种语言编写的程序不但冗长，而且可读性差、移植性差、容易出错、晦涩难懂，是一般人类所无法接受的。然而这样的程序执行效率高，节省内存，运行速度快，通常用于直接控制计算机的硬件。

为了克服机器语言的缺点，产生了汇编语言。汇编语言是机器语言的一种变形，是将机器指令助记成为可读易懂的符号，然而仍只能完成计算机层次的操作；用汇编语言编写的程序要在计算机上执行，先要将用汇编语言编写的源程序转换成机器语言程序，称完成这个转换功能的程序为“汇编程序”。

低级语言的缺点是指令功能简单，即使完成一个算术表达式的运算也需编写大段程序，而且需要对硬件进行了解，完成的程序不具可移植性。

低级语言的优点是，执行速度快，可直接控制硬件，适用于实时性要求较高的自动控制系统。为了克服以上两种语言的缺陷，从而产生了许多高级语言。

高级语言的共同特点是完全不依赖于硬件，接近于自然语言。高级语言的基本构成是语句，语句的功能要比机器语言指令的功能强大得多，另外还提供了丰富的函数或类库。

### 3. 程序设计语言按其采用的范型分类

#### (1) 面向过程的语言。

面向过程的语言虽可独立于计算机编写程序,然而采用该类语言编写程序时,程序不仅要说明做什么,更重要的是非常详细地告诉计算机如何做,程序需要详细描述解题的过程和细节。

#### (2) 面向问题的语言。

不必关心问题的求解算法和求解的过程,只需指出问题是做什么以及数据的输入和输出形式,就能得到所需结果。

面向问题语言又称为非过程化语言或陈述性语言,如报表语言、SQL(Structured Query Language)语言等。SQL语言是数据库查询和操纵语言,能直接使用数据库管理系统。因为使用面向问题语言解题只要告诉计算机做什么,不必告诉计算机如何做。

#### (3) 面向对象的语言。

为克服面向过程语言过分强调求解过程的细节,程序不易复用的缺点,产生了面向对象程序设计方法和面向对象语言。面向对象语言引入了对象、消息、类、继承、封装、抽象、多态性等机制和概念。用面向对象语言进行程序设计时,以问题域中的对象为基础,将具有类似性质的对象抽象成类,并利用继承机制,仅对差异进行程序设计,可以提高软件开发效率。

### 4. 其他分类方式

按应用领域分类,有人工智能程序设计语言、逻辑推理程序设计语言、系统程序设计语言。

命令式程序设计语言,因为其适用性强、应用范围广、语句简捷、灵活而得到广泛的使用,如Pascal、Basic、Fortran、C都属于这类语言。程序设计初学者一般首先学习的就是命令式程序设计语言。

### 5. 关于 C 语言

C语言是在20世纪70年代初问世的,1978年由美国电话电报公司(AT&T)贝尔实验室正式发表了C语言。C语言属于面向过程的高级语言。在C语言的基础上,1983年又由贝尔实验室的Bjarne Stroustrup推出了C++语言。C++语言进一步扩充和完善了C语言,是一种面向对象的高级语言。程序设计语言只是程序设计的工具,通过对一种程序设计语言的学习,掌握了程序设计的思想方法后是可以触类旁通的。

## 1.2 C 语言概述

### 1. C 语言的发展概况

C语言是当今社会应用广泛并受到众多用户欢迎的一种计算机算法语言。既可以作为系统软件描述语言,也可用来开发应用程序。

C语言的前身是BCPL语言,用于描述和实现UNIX操作系统。1970年,美国贝尔实验室的



Ken. Thompson 继承和发展了 BCPL 语言,提出了 B 语言,在当时最新型的小型机 PDP-7 上实现了第一个 UNIX 操作系统。1972 年,美国贝尔实验室的 Dennis M. Ritchie 和 Brian. W. Kernighan 对 B 语言作了完善和发展,提出了一种新型的程序设计语言——C 语言。1973 年,他们用 C 语言改写了 UNIX 操作系统,从此 C 语言名闻天下。

今天 C 语言仍然有广泛的应用,除了当初为 UNIX 操作系统开发使用外,现在各行各业的应用软件,许多都是用 C 语言进行系统描述和开发的,如大量工程设计、工程测绘和科学计算软件是以 C 语言为基础开发的,许多工业领域,如机电控制、军事、建筑、冶金、纺织等都直接使用 C 语言开发工业控制软件。

## 2. C 语言的特点

C 语言之所以具有强大的生命力,是因为它有不同于(或优于)其他语言的特点。

(1) 语言简洁、紧凑。

C 语言只有 32 个关键字,使编写的程序短小精练。

(2) 数据类型丰富,运算功能强大。

C 语言既提供简单常用的数据表示方法,还提供复杂的数据描述方法。其中,指针类型是它最鲜明的特色,特别适合于开发需要复杂数据表示的系统软件。C 语言提供多达 34 种运算符,表达式灵活、多样,功能大大强于其他高级语言。

(3) 完全模块化和结构化设计功能。

提供了用于 3 种标准化程序设计使用的 9 种控制语句,以函数作为模块化程序设计的基本单位。

(4) 具备低级语言和高级语言的双重功能。

C 语言又称中间语言,除了强大的高级语言功能外,还可以直接访问硬件,具备低级语言的大部分功能。

(5) 语法灵活,设计自由度大。

C 语言语法限制少,为程序设计提供了最大自由度,受到编程高手的热烈欢迎。对于初学者来说,需要提高程序设计熟练度,逐渐适应 C 语言语法。

(6) 程序高效,可移植性好。

C 语言编写的程序代码短,执行速度快。一次编写成的程序,可以不用修改或稍做修改,就可应用于不同的硬件和不同的操作系统中。

## 3. 怎样学好 C 语言

C 语言是各学科计算机应用的基础,也是进一步提高计算机能力的基础,还是当前计算机等级考试过级的必备,因此必须从开始就掌握正确的方法,通过认真学习,达到掌握和运用 C 语言的目标。

学好 C 语言要注意以下几个方面:

(1) 掌握重点,消化难点。

C 语言灵活,内容丰富。所以一开始学习就要目标明确,抓住重点的内容学习,在巩固重点的情况下,再丰富所学内容,提高能力。

(2) 多看多动手。

初学者起步时容易出现的问题有不理解概念、看不懂程序、自己没有编程思路。通过阅读程

序,认真体会各种概念的具体应用。遇到看不懂程序的情况,可以通过多动手解决。多动手是指在初学每个知识点的前几个程序时,不能光动脑,在动脑的同时,还要动手。记下变量的初始值,按程序的执行流程,记录变量的不同变化,通过精读彻底明白程序。练习几次以后再看程序,便能很快理解程序。编程没有思路,可以通过举一反三的方法解决,按照已学程序的方法,先模仿、再改造,最后组合各种功能,就会自己动手编写程序了。

### (3) 多上机练习。

C 语言有很强的实践性,一个没有经过上机验证的程序,始终不能算是真正正确的程序。何况 C 语言灵活性很强,在上机中会发现很多问题。通过上机可以学会程序调试的方法,不同计算机环境下的编程环境设置,找到程序调试的技巧等。

### (4) 扩展视野。

在掌握了重点和基础的内容后,还可以结合自己的需要多看多学一些知识。

## 1.3 C 语言程序的基本结构

### 1.3.1 C 语言程序的基本结构

#### 1. C 语言程序的一般形式

C 语言程序的一般形式:

#include 语句

全局变量说明

```
main( )           /* 主函数 main ( ) */
{
    局部变量      /* main 定义的变量,在 main 中有效 */
    程序段        /* 完成特定功能的一系列语句 */
}

fun1( )           /* 用户自行定义的名为 fun1 的函数 */
{
    局部变量      /* fun1 定义的变量,在 fun1 中有效 */
    程序段        /* 完成 fun1 特定功能的一系列语句 */
}

fun2( ) {……}    /* 用户可以根据需要自定义多个函数 */
```

说明:

(1) C 语言程序的开头一般都使用一个或多个 include 语句,用来引入某些标准库文件。这类标准库文件通常称为头文件,头文件的扩展名为 .h。

(2) 程序的前部有时可以定义一些全局变量,这些变量被后面的各个函数共享,但要注意对全局变量的使用不能互相干扰。在某个函数体内定义的变量,称为局部变量,它只在这个函数体

内有效。

(3) C 语言程序都是由若干个函数为单位组成,如 `main( )`、`fun1( )`、`fun2( )`,每个函数又由若干个 C 语言基本语句组成,每个语句必须以分号(;)结束。

(4) C 语言程序中必须有一个 `main` 函数(又称为主函数),作为程序执行的起点,但是 `main( )` 位置不限。`main( )` 表示是一个函数,其中 `main` 是函数的名称,( ) 内列出函数执行需要的  $n$  个参数,{} 内为完成特定功能的一系列语句。

(5) 为了实现程序设计的模块化,如果需要,用户可以定义若干个自定义的函数,如 `fun1( )`、`fun2( )`……自定义函数的作用是使程序结构清晰,便于实现某个功能的重复使用。

(6) 每个函数中的程序段可以由一系列语句组成,如完成输入数据、计算、输出数据和显示数据等功能。

## 2. 简单的 C 语言程序

例 1.1 编写一个 C 语言程序,其功能是在显示屏上输出 English is fun!。

```
#include <stdio.h>    /* 头文件 stdio.h 是标准输入输出头文件,指定输入 */
                    /* 为键盘,输出为显示屏 */
main( )              /* 主函数 */
{
    printf("English is fun ! \n"); /* 使用 stdio.h 中已经定义的 printf 函数 */
                                /* 输出显示内容 */
}
```

程序说明:

(1) 只用一个 `main( )` 就可以完成这个功能。不需要定义全局变量、局部变量,也不需要用户自定义函数。`main( )` 函数中的程序段只有一个语句。

(2) 程序开头使用了 `#include <stdio.h>`,可以直接使用标准输入输出头文件中定义的 `printf` 函数。

(3) `printf( )` 的双引号内是可以直接输出显示的内容。其中“\n”是特殊符号,表示输出结束后光标转到下一行。

运行结果:

English is fun!

如果要在显示屏上输出“Hello, World!”或类似的一句话,并且在输出后光标换行,程序应当怎么写? 留给读者思考。

例 1.2 从键盘上输入两个实数,计算并显示这两个实数平方之和的平方根。

```
#include <stdio.h>
#include <math.h>    /* 引入 math.h 头文件,以便使用开平方函数 sqrt( ) */
main( )
{
    float x,y,z;    /* 用 float 定义表示实数的局部变量,x 和 y 存放输入 */
                    /* 的数据,s 存放计算结果 */
    scanf("%f,%f",&x,&y); /* 从键盘上输入两个实数,保存在 x 和 y 中 */
}
```

```

    s = sqrt(x * x + y * y);          /* 计算 */
    printf("z = %f\n", z);          /* 显示结果 */
}

```

程序说明:

(1) 使用了数学头文件 `math.h`, 可以直接使用 `sqrt()` 函数实现开平方计算。

(2) 程序中一般用“\*”代替“×”。

(3) `float` 用来表示实数, `int` 用来表示整数。

(4) `scanf()` 也是 `stdio.h` 文件中的函数, 用于从键盘上读入数据。引号中的“%f”表示按实数形式读入, 两个“%f”之间用逗号相隔, 则输入数据时也要用逗号相隔。&x, &y 表示输入的数据要放到 x 和 y 所在的内存单元中去, & 是取地址符号, &x 表示的是 x 在内存中的地址。

(5) `printf()` 中出现的“%f”是用来指定输出数据的格式, 用后面的 z 替换“%f”的位置, 并按小数的形式输出。

(6) 常用的控制符号, “%d”是按十进制整数格式输出; “%c”是按单个字符输出; “%s”是按字符串格式输出。

输入数据:

1,2 ↵

程序运行结果:

z = 2.236068

试编写一个程序, 要求从键盘上输入 4 个数, 表示一个同学 4 门课的成绩, 求总成绩和平均成绩。

例 1.3 由 `main()` 函数和 1 个求和函数 `sum()` 构成的 C 语言程序示例。

```

#include <stdio.h>

int sum(int x, int y)          /* 计算两数之和 */
{
    int z;                    /* sum 函数体开始 */
    z = x + y;                /* 声明部分, 定义变量 */
    return(z);                /* 将 z 值返回, 通过 sum 带回调用处 */
}

main()                        /* sum 函数体结束 */
{
    int a, b, c;              /* 主函数 */
    /* main 函数体开始 */
    printf("Input the first number:");
    scanf("%d", &a);
    printf("Input the second number:");
    scanf("%d", &b);
    c = sum(a, b);            /* 调用 max, 将调用结果赋给 c */
    printf("sum = %d\n", c);
}
/* main 函数体结束 */

```

程序说明:

(1)本程序包括两个函数。其中主函数 main 仍然是整个程序执行的起点。函数 sum 计算两数之和。

(2)主函数 main 调用 scanf 函数获得两个整数,存入 a,b 两个变量,然后调用函数 sum 获得两个数之和,并赋给变量 c。最后输出变量 c 的值(结果)。

(3)int sum(int x,int y)是函数 sum 的函数头。函数 sum 的函数头表明此函数获得两个整数,返回一个整数。

(4)函数 sum 同样也用{}将函数体括起来。sum 的函数体是函数 sum 的具体实现。从参数表获得数据,处理后得到结果 z,然后将 z 返回调用函数 main。

(5)本例还表明函数除了调用库函数外,还可以调回用户自己定义、编制的函数。

程序运行结果:

Input the first number:3 ↵

Input the second number:5 ↵

sum = 8

### 3. C 语言程序的编写要求

(1)C 程序是由函数构成的,一个 C 源程序至少要有 main 函数。

(2)一个 C 程序总是从 main 函数开始执行的,不论 main 函数在整个程序中的位置如何(可以在程序最前头,也可以在最后头,或在程序中间)。

(3)C 程序书写格式自由,一行内可以写几个语句,一个语句也可以分写在多行上(在行末加 <\回车>,在下一行接着书写)。不要把一个关键词、标识符、常量、运算符和字符串拆分为两行。最好每个语句占用一个书写行,每个函数都按语句的层次关系形成缩进形式。

(4)使用/\*……\*/进行注释。为了增加程序的可读性,可在程序的任何需要的地方加上注解。/\*……\*/中的内容是给阅读源程序的人看的,计算机执行程序时会忽略这些注释。

### 4. C 语言程序的执行过程

输入与编辑源程序→编译源程序,产生目标程序→连接各个目标代码、库函数,产生可执行程序→运行程序。

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”。事实上编写的程序,不管采用什么计算机语言,都是源程序,因为不会有人用机器语言去编程。源程序不能直接在计算机上执行,需要用“编译程序”将源程序翻译为二进制形式的代码。

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”。目标代码尽管已经是机器指令,但是还不能运行,因为目标程序还没有解决函数调用问题,需要将各个目标程序与库函数连接,才能形成完整的可执行的程序。

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/Windows 环境下)。

#### 1.3.2 关键字和标识符

C 语言程序中包含着许多单词和符号,它们在程序中有着不同的性质和作用。



## 1. 关键字

关键字又称保留字,是 C 语言中具有固定含义的特殊单词,如: `int` 表示整型数据, `double` 表示双精度实型数据等。用户在程序中引用它们时,只能利用它们本身的特定意义,而不能重新定义以做他用。C 语言的关键字,如表 1-1 所示。

表 1-1 C 语言的关键字

<code>auto</code>	<code>break</code>	<code>case</code>	<code>char</code>	<code>const</code>	<code>continue</code>	<code>default</code>	<code>do</code>
<code>double</code>	<code>else</code>	<code>enum</code>	<code>extern</code>	<code>float</code>	<code>for</code>	<code>goto</code>	<code>if</code>
<code>int</code>	<code>long</code>	<code>register</code>	<code>short</code>	<code>signed</code>	<code>sizeof</code>	<code>static</code>	<code>return</code>
<code>asm</code>	<code>far</code>	<code>huge</code>	<code>interrupt</code>	<code>near</code>	<code>pascal</code>	<code>cdecl</code>	<code>while</code>
<code>struct</code>	<code>switch</code>	<code>typedef</code>	<code>union</code>	<code>unsigned</code>	<code>void</code>	<code>volatile</code>	

## 2. 标识符

标识符分为系统预定义标识符和用户自定义标识符两种。

(1) 系统预定义标识符是由一些单词组成,它们的功能和含义是由系统预先定义的,如 `main` 代表主函数名, `scanf` 代表输入函数名等。但是它们允许被用户赋予新的含义,这是它们区别于关键字的地方。不过这样做会引起误解,C 语言编程时不提倡该做法。

(2) 用户自定义标识符是由用户根据需要自己定义的名称,例如变量名、数组名、自定义函数名等。建立的规则是要求标识符的头一个字符必须是字母或下划线,标识符中的字符可由英文字母、数字和下划线组成,且相同字母的大写和小写是有区别的,被认为是不同的表示形式。标识符的长度视不同的编译器而不同,一般可以识别前 8 个字符。

例如, `count`、`score33`、`_above`、`AUI` 是 4 个合法的标识符。`M.d`、`99yiohn`、`$ 12` 是 3 个非法的标识符。

为使程序有较好的可读性,程序设计中提倡建立具有一定含义的标识符。例如,建立 `pi` 变量名表示圆周率,用 `count` 代表计数, `score` 可以表示分数值等。

### 1.3.3 常量和变量

程序的运行过程实际上是在处理各种各样的数据,一般情况下,程序中处理的数据是以常量或变量的形式表示的。

#### 1. 常量

在程序执行过程中值不变的数据叫做常量。例如,圆周率的值 3.14159 就属于常量,它的值在程序执行中不需改变。如果给常量建立一个标识符,则此常量被称做符号常量。C 程序是通过 `#define` 预处理命令来定义符号常量的。

例如,如果在程序的开始处带有一行预处理命令:

```
#define PI 3.14159
```

那么该命令将 3.14159 命名为符号常量 `PI`,从此, `PI` 这个名称在程序中将只代表数值

3. 14159,且不允许对 PI 重新赋值。

常量有多种类型,如整型常量、实型常量、字符型常量、字符串常量等。

## 2. 变量

在程序执行过程中其内部的值可以发生改变的量叫做变量。程序中的变量实际上是存储单元,它对应于计算机内存的某个存储空间。变量也包括多种类型,如整型变量、实型变量、字符型变量等。

程序中的变量由用户标识符来表示,在 C 语言程序中,所有变量必须遵循“先定义,后使用”的原则。变量的定义一般放在函数体的开始处,它属于程序的声明部分。

变量定义语句的形式是:

类型名 变量名 1,变量名 2,……;

其中,类型名为关键字,不同的关键字表示不同的数据类型。每一个变量都有唯一的名称叫做变量名,是由用户自己定义的。例如:

```
int a,b,c;
```

此语句表示定义了名为 a,b,c 的 3 个整型变量。

通常,在定义一个变量而没有给变量赋初值时,变量中存放的就是一个随机值。因此,为定义的变量有一个确切的数值,就需要给定义的变量赋一个初值。另外,C 语言也允许在定义变量的同时为其赋初值,其形式是:

类型名 变量名 1 = 常量 1,变量名 2 = 常量 2,……;

每个变量代表着内存中一个具体的存储单元,变量名是存储单元的标识,存储单元中存放的数据是变量的值,变量的值可以通过赋值的方法获得和发生改变。

# 1.4 C 语言程序的开发环境

## 1.4.1 Turbo2.0

C 语言有多种不同的编译器,其中 Turbo C(以下简称 TC)是深受 C 语言初学者喜爱的编译器之一。Turbo C 2.0 是美国 Borland 公司在 1988 年推出的为 PC 实现的一个快速、高效和优化的 C 语言集成环境,它不仅提供了基于 ANSIC 的编译器和链接器,还提供了一个供程序员修改程序的方便的编辑器,此外还包括了一些混合模式程序设计的扩充,具有图形库和文本窗口函数库,以及查错和协处理器的仿真等功能,形成了一个集成化的 C 语言程序开发和运行的环境。TC 2.0 每分钟可编译 12000 行源代码,这是因为 TC 2.0 编译时利用了内存存放中间数据结构,一次性产生内部代码,只需一次即可读出盘上的源文件和写出目标代码。

Turbo C 最显著的优点是提供了一个完整的集成开发环境,包括:

(1)一个功能强大的交互式全屏幕文本编辑程序,并具有源程序级的调试功能,当编译程序发生错误时,它会查出错误,对于每一个错误信息自动把光标设置在源程序相应的位置,极大地方便了排错和调试。

(2)操作全自动,易学易用。使用 TC,不需要另外使用单独的编辑程序、编译程序、连接程序等软件工具去建立文件、执行程序,所有这些特性都已完整地集成在 TC 内,加之全屏幕的显示方式使用应用变得更加直观、简明。

(3)TC 支持 6 种存储模式:tiny(极小),small(小),medium(中),compact(紧凑),large(大)和 huge(巨大)。可混合使用远和近指针。此外,还包括了一些混合语言和混合模式程序设计的选择,从而更进一步挖掘了 PC 的能力。

(4)TC 除提供集成开发环境版本外,还提供了常规命令行版本。

要安装 TC 2.0 版本,可运行安装盘中的 INSTALL.EXE 文件,这时可以改变系统安装所需配置的一些参数,然后按屏幕提示操作即可。INSTALL 运行完之后,便可以运行 TC 了。

若用户将 TC 2.0 按系统默认的路径安装到硬盘上,则安装程序将会创建如下目录:

C:\TC 系统主目录

C:\TC\INCLUDE 系统包含文件目录

C:\TC\LIB 系统库文件目录

若用户不按默认的路径安装,可自行创建一个目录。首先在根目录下建立子目录,然后进入此子目录并安装所有程序文件,如在 E 盘建立一个子目录\TC,接着在\TC 下建立两个子目录 LIB 和 INCLUDE,在子目录\TC\LIB 中存放所有库文件,在子目录\TC\INCLUDE 中存放所有头文件。

几个常用的头文件如下。

alloc.h:存储管理函数说明。

bios.h:调用 BIOS(基本输入输出系统)的函数说明。

math.h:数学函数说明。

string.h:串函数说明。

conio.h:调用控制台 I/O 例行程序的函数说明。

stdio.h:标准 I/O 函数说明。

常用的库函数有如下几个(以小模式为例)。

cos.obj:小模式启动代码。

cs.lib:小模式运行时库。

Maths.lib:小模式数学库。

xxx.bgi:某一图形卡的图形驱动程序。

graphics.lib:图形库。

## 1.4.2 Turbo C 开发环境介绍

### 1. TC 开发环境概述

Turbo C 定义了两种屏幕状态:程序开发状态和程序运行状态。两个状态相互独立,互不干涉。通常,Turbo C 位于开发环境状态,只有程序运行时才进入运行状态(也称用户屏幕)。

在安装有 Turbo C 的 Windows 平台上,打开含有 Turbo C 的文件夹后,点击可执行文件 TC;或着通过在桌面上为 TC 建立的快捷方式,直接双击该快捷方式,均可进入 Turbo C 2.0 集成开发环境。Turbo C 的界面如图 1-1 所示。



图 1-1 TC 开发环境

### (1) 标题栏。

位于窗口的第一行,显示当前程序名 TC。标题栏左端的图标一方面表明它是一个 Turbo C 集成环境系统,另一方面点击它可以下拉一个控制菜单,用于 Turbo C 集成环境系统的关闭及窗口大小的控制,如图 1-2 所示。这些控制功能也可以用标题栏右端的三个按钮执行。

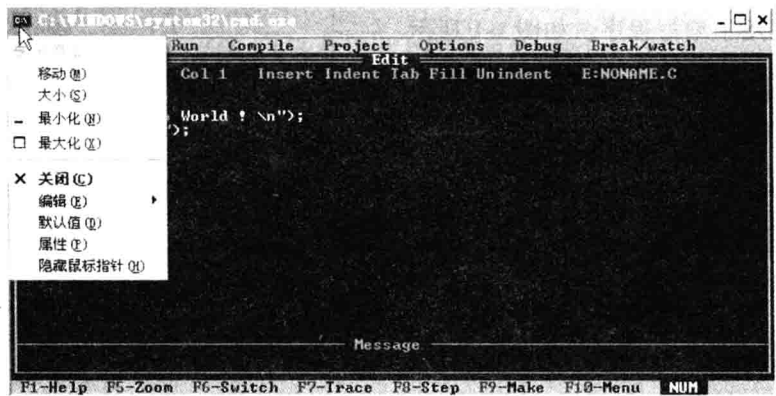


图 1-2 TC 标题栏示意图

### (2) 菜单栏。

位于窗口的第二行,如图 1-3 所示,共有 8 个菜单项:

- |                      |                             |
|----------------------|-----------------------------|
| File(Alt + f)        | 进行文件和目录操作。                  |
| Edit(Alt + e)        | 编辑当前编辑窗口中显示的程序。             |
| Run(Alt + r)         | 控制程序的运行状态。                  |
| Compile(Alt + c)     | 进行程序的编译和链接。                 |
| Project(Alt + p)     | 对工程文件(由多个 C 语言文件组成的程序)进行管理。 |
| Options(Alt + o)     | 设置选项。                       |
| Debug(Alt + d)       | 调试程序,包括:显示变量值,查找函数及查看调用栈状态。 |
| Break/Match(Alt + b) | 调试程序;设置清除断点,监测变量值变化。        |