

# 第 1 章 C 语言概述

## 本章要点

- ◆ 了解程序和程序设计的基本概念
- ◆ 了解 C 语言的发展及其特点
- ◆ 掌握 C 语言的构成
- ◆ 掌握 Turbo C 的作用及使用方法
- ◆ 掌握 C 程序的编译、连接和运行步骤

## 第 1 节 程序和程序设计

### 1. 程序

当今使用的计算机高级语言有上百种，如 BASIC、PASCAL 及本书介绍的 C 语言等。这些计算机语言的表达形式都接近人们习惯使用的自然语言和数学语言，人们学习和操作起来都感到十分方便。但是，计算机本身并不能直接识别由高级语言编写的程序，它只能处理由 0、1 代码构成的二进制指令或数据，这种由 0、1 代码构成的二进制指令的集合称为机器语言，不同的机器，汇编语言也不同。

所谓程序指的是为解决某一问题而使用计算机语言编排的处理步骤。通常把由高级语言编写的程序称为“源程序”，把由二进制代码构成的程序称为“目标程序”。为了能使计算机处理高级语言源程序，需要把源程序转换成机器能够接受的目标程序，这种具有翻译功能的软件称为“编译程序”或“解释程序”。每一种高级语言都有与它对应的翻译程序。

由 C 语言构成的指令序列称为 C 源程序，每条 C 语句，经过编译（Compile）后都将转换成二进制的机器指令。C 源程序经过编译程序编译后生成的目标文件扩展名为 .OBJ。目标程序还不能运行，它还需要通过连接（Link）程序把 .OBJ 文件与 C 语言提供的各种库函数连接起来，生成一个扩展名为 .EXE 的可执行文件。在 DOS 状态下，只需键入该文件名就可运行这个 C 程序。

### 2. 程序设计

程序设计就是编写程序的过程，一般包括以下几个部分：

- (1) 确定数据结构。根据用户提出的要求、原始数据以及输出形式，确定数据结构。
- (2) 确定算法。针对数据结构，确定解决问题、完成任务的步骤。
- (3) 编写程序。根据确定的数据结构和算法，使用计算机语言编写程序代码，输入到计算机中，简称编程。
- (4) 调试程序。调试过程主要是检查程序的语法错误或逻辑错误，并输入各种可能的少量数据对程序进行测试，使它能得到正确的结果，同时对非法数据能进行容错处理。
- (5) 整理并写出文档资料。

### 3. 结构化和模块化程序

#### 3.1 结构化程序

结构化程序是由顺序结构、选择结构和循环结构这三种基本结构组成。其算法结构可以解决任何复杂的问题，控制这些结构中的各种程序流程的语句就是流程控制语句。

C 语言是一种典型的结构化程序设计语言，它提供了实现结构化程序所需的多种流程

控制语句。选择结构的流程控制语句有：if~else 条件分支 和 switch~case( 开关语句 ); 循环结构的流程控制语句有：while、do~while 和 for 辅助控制语句有：break、continue、goto 和 return。

### 3.2 模块化程序设计方法

现在大多数程序都比较复杂，一般都由许多人员共同协作开发完成，这时可采用“自顶向下，逐步求精”的模块化程序设计方法，即开始时并不涉及问题的实质和具体解决的步骤，而只是从问题的全局出发，给出一个概要的抽象描述。处理的方法是：将这个复杂的大任务分解为若干个较小的、容易处理的子任务，如果分解好的子任务本身还很复杂，可以再细分为许多更小的子任务，一个子任务只完成一项简单的任务。一旦这些较小的任务解决了，那么原来的问题也就迎刃而解了，这些较小的任务称为“模块”。在程序设计时，分别用各个小模块来实现总体功能。程序设计人员分别完成一个或多个小模块，这样的程序设计方法称为模块化程序设计方法，由一个个功能模块构成的程序结构为模块化结构。模块化结构程序设计方法使得同一任务可由一组人员同时进行分工编写，并分别调试，这就大大提高了程序编制的效率。

软件编制人员在进行程序设计的时候，首先应该考虑主程序的算法，写出主程序后再动手逐步完成子程序调用。对于这些“子”程序，也可利用调试主程序的方法逐步完成其下一层子程序的调用。

## 第 2 节 C 语言的发展和特点

### 1. C 语言的发展

C 语言是当今广泛流行的编译型程序设计语言，它面向用户，面向解题的过程，C 语言易于编程，又可以像汇编语言一样，直接对硬件内存单元的位、字节进行操作，程序运行效率高。因此，C 语言既可以编写系统软件，也可编写应用软件。它的发展过程是一个充实和完善的过程。

C 语言是在 B 语言的基础上发展起来的，它的发展与操作系统 UNIX 密切相关，UNIX 的早期版本是用汇编语言编写的。用汇编语言编写的 UNIX 可移植性差、效率低、编程困难，因此，1970 年美国贝尔实验室的 Ken Thompson 在 BCPL 语言（计算机软件人员在开发系统时，作为记述语言）的基础上，设计出了简单且又接近硬件的 B 语言（取自 BCPL 的第一个字母）并用 B 语言编写了 UNIX 操作系统。BCPL 语言基于 CPL 语言，而 CPL 语言又是基于 ALGOL 60，ALGOL 60 是一种面向问题的高级语言。由于 B 语言依赖于机器，过于简单，功能有限，又无数据类型，所以没有流行开来。1972 年到 1973 年间，贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出了 C 语言（取 BCPL 的第二个字母），C 语言既保持了 BCPL 和 B 语言的特点（精练、接近硬件），又克服了它们的缺点（过于简单、无数据类型等）。1973 年，K.Thompson 和 D.M.Ritchie 合作，把 UNIX 的 90% 以上用 C 语言改写。后来，C 语言多次进行了改进，1977 年出现了不依赖于具体机器的 C 语言编译文本，C 语言也迅速得到推广，成为当今应用最广泛的计算机语言之一。

Brian W.Kernighan 和 Dennis M.Ritchie (简称 K&R) 合著的《The C Programming Language》成为后来广泛使用的 C 语言版本的基础,称为标准 C。1983 年美国国家标准化协会(ANSI)根据 C 问世以来的各种版本进行了发展和扩充,并制定了新的标准,称为 ANSI C。1988 年 K&R 修改了他们的经典著作《The C Programming Language》,按照 ANSI C 的标准重写了该书。1987 年,ANSI 又公布了 87 ANSI C (新标准),目前流行的 C 语言版本都是以它为基础的,各种版本基本部分相同,但也有一些不同。在 PC 机上使用的有 DOS 下的 Microsoft C、Turbo C 和 Quick C 等,它们的不同版本又有差异。随着面向对象编程技术(OOP)的发展,又出现了 Turbo C++、Visual C 等。本书的例题和习题都是在 Turbo C 2.0 编译环境下运行。

## 精 讲 与 必 读

## 2. C 语言的特点

C 语言的主要特点如下:

(1) C 语言是处于汇编语言和高级语言之间的一种记述性程序设计语言,它的发展历史漫长。

(2) 语言简洁紧凑。C 语言一共只有 32 个关键字(见附录 3),9 种控制语句,Turbo C 增加了 11 个关键字(用于各种增强和扩展功能)。C 与 PASCAL 相比,语言简练,源程序短,如用 { } 代替了 PASCAL 中的 Begin...End,以 ++ 表示自增 1,运算符省略等等。

(3) 运算符丰富,表达式能力强。C 语言共有 32 种运算符(见附录 4),它把括号、赋值、强制类型转换等都作为运算符处理,因此 C 的运算类型丰富,表达式类型多样,可以进行四则运算、位运算、单项运行运算、逻辑运算和复合运算,能实现其他高级语言中难以实现的运算。另外,C 语言允许直接访问物理地址,可直接处理字符、数字、地址,能进行位处理,能实现汇编语言的大部分功能。所以,C 语言兼顾了高级语言和汇编语言的优点,最适宜用来开发系统软件。

(4) 数据结构丰富。C 语言具有丰富的数据结构,其数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等,因此能实现复杂的数据结构的运算。

(5) C 语言是结构化、模块化的编程语言。C 语言具有结构化控制语句,可以通过多种结构语句组成程序的逻辑结构,因此功能强大足以描述各种结构的程序。

C 程序的主要结构成分是函数,用函数作为程序模块来实现模块化,因此可将整个程序分成若干模块,以便多人共同开发。C 的函数库相当丰富,标准 C 提供了一百多个库函数,Turbo C 提供了三百多个库函数(见附录 5)每个编程者还可以根据需要建立自己的函数库。编程时可直接调用相应的函数,从而节省编程时间。

(6) C 程序中,可使用宏定义编译预处理语句、条件编译预处理语句,为编程提供了方便。

(7) 可移植性好。与汇编语言相比,C 程序基本上不作修改就可以运行于各种型号的计算机和各种操作系统。

由于 C 语言具有上述优点,因此 C 语言得到了迅速推广,成为人们编写大型软件的首选语言之一。许多原来用汇编语言处理的问题可以用 C 语言来处理了。编程者在使用 C 语言

编写程序时会感到限制少、灵活性大、功能强。

## 第 3 节 C 语言的构成

### 1. C 程序简介

下面通过简单的 C 程序，介绍 C 程序的基本构成和格式，并了解 C 程序的特性。

【例 1-1】在计算机屏幕上输出 “ How are you!”。

```
main()
{
    printf("How are you!\n");
}
```

程序运行结果：

How are you!

这个源程序只由一个函数组成，“main”表示主函数，C 语言规定必须用 `main` 作为主函数名，函数名后的一对圆括号不能省略，圆括号中内容可以是空的。一个 C 程序可以包含任意多个函数，但必须有一个且仅有一个主函数，一个 C 程序总是从主函数开始执行。函数体需用花括号括起来，其间可以有定义（说明）部分和执行语句部分，语句的数量不限，程序中由这些语句向计算机系统发出指令，本程序函数体内只有一个输出语句，双引号内的内容原样输出，“`\n`”表示输出字符后换行。C 程序中的每一条语句都必须用分号“`;`”结束，分号是 C 语句的一部分，不是语句之间的分隔符。

【例 1-2】已知圆的半径为 6，求圆的周长和面积。

```
main()
{ int r;                /* 圆半径 r 为整型变量 */
  float l,s;            /* 周长 l、面积 s 为实型变量 */
  r=6;                 /* r 赋初值 */
  l=2*3.14159*r; s=3.14159*r*r; /* 计算 l,s 的值 */
  printf("r=%d,l=%f,s=%f\n",r,l,s); /* 输出圆的半径、周长和面积 */
}
```

程序的运行结果为：

r=6,l=37.699081,s=113.097237

程序中第二、三行定义了三个变量，说明了 `r` 为整型变量，`l,s` 为实型变量。第四、五行完成输入已知值 `r` 并根据 `r` 的值计算圆周长和面积。输出语句中的“`%d,%f`”为输出格式符，分别表示十进制整型和实型，它指定输出结果时的数据类型和格式，程序在执行时，该位置由具体数据替代。程序中的“`/*.....*/`”表示注释部分，作用是帮助用户阅读程序，它对程序的运行不起作用，在对源程序进行编译时，注释会被忽略。“`/*`”和“`*/`”必须成对出现，且“`/`”和“`*`”之间不能有空格，注释内容可以是西文，也可以是中文，注释中可以说明变量的含义、程序段的功能。注释可以放在程序中任意合适位置，一个好的程序应该有详细的注释。

【例 1-3】 输入矩形的两个边长，求矩形的面积。

程序如下：

```
#include "stdio.h"
main()
{
    int x,y,z;
    scanf("%d,%d",&x,&y);          /* 输入矩形的两条边长 */
    z=area(x,y);                   /* 调用函数 area */
    printf("area is %d\n",z);      /* 输出矩形的面积 */
}

int area(a,b)
int a,b;
{int c;
  c=a*b;
  return(c);
}
```

程序运行结果为：

```
6,8          ( 键盘输入 6,8 给 x,y)
area=48
```

#include "stdio.h"称为命令行，命令行必须用"#"号开头，后面不能加"."号，因为它不是C程序中的语句，“stdio.h”是系统提供的库函数名，其中包含有关输入输出函数的信息。本程序由主函数 main 和被调用函数 area 组成，在主函数中输入两个边长 x,y，然后通过语句 z=area(x,y)调用函数 area，计算结果由 return 语句返回给主函数。scanf 和 printf 是 C 语言提供的标准输入输出函数，&a 和 &b 中的“&”的含义是“取地址”，程序中 scanf 函数的作用是从键盘上键入的两个数，输入到变量 x 和 y 所标志的内存单元中，亦即输入给变量 x 和 y。

## 2. C 语言的构成

通过上面几个例子，可以看到：

### (1) C 语言是由函数构成的

C 语言中，一个具有独立功能的程序段可组织成函数，一个 C 源程序必须有一个主函数 main，还可以包含若干个其他函数，函数是 C 程序的基本单位。被调用函数可以是系统库函数，或是用户根据需要自己编写的函数。正由于这个特点，C 程序很容易实现模块化。

### (2) 每个函数由函数说明部分和函数体组成

函数说明部分由函数名、函数类型、函数参数名、形式参数类型组成，如：

```
int area(a,b);
```

```
int a,b;
```

函数体包括数据说明部分和执行部分，当然可以没有数据说明部分或没有执行部分，

甚至可以是一个空函数。

(3) C程序从 `main` 函数开始执行

`main` 函数可以在程序的任意位置，但一个 C 程序总是从 `main` 函数开始执行的。其他函数都是通过嵌套调用而得以执行的。

(4) 书写格式自由

C 程序没有行号，源程序书写（输入）的时候，可以在一行内写几个语句，也可以把一个语句分成多行书写，只要每个语句间用分号分隔就可以。多个语句可以用“{ }”括起来成为复合语句。

C 源程序习惯上使用小写字母书写，但在一些宏定义中，将常量名用大写字母表示，而对某些特殊含义的变量，偶尔也用大写字母表示。需要注意，C 语言区分大小写字母。

## 第 4 节 Turbo C 中运行 C 程序的步骤

### 1. C 程序编译、连接过程

一个 C 程序必须要经过编辑、编译、连接的过程，才能生成一个可执行程序。编译和连接过程如图 1-1 所示。

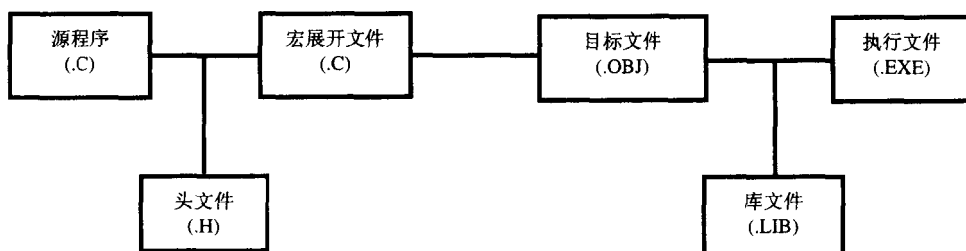


图 1-1 C 程序编译、连接过程

### 2. Turbo C 2.0 运行 C 程序的步骤

Turbo C 2.0 是美国 Borland 公司 1989 年继 Turbo C 1.0 版、Turbo C 1.5 版之后又一集编辑、编译、连接、执行为一体的 C 语言开发软件，它具有友好的用户界面和丰富的库函数，是目前较为流行的版本之一。用户可将 Turbo C 2.0 按默认的路径安装到硬盘上，安装程序将自动在硬盘上创建 TC 子目录，在 TC 子目录下又创建 INCLUDE 和 LIB 子目录。其中 TC 是 Turbo C 系统的主目录，INCLUDE 是 Turbo C 系统包含文件所在目录，LIB 是 Turbo C 系统所在目录。

Turbo C 提供了 TC 和 TCC 两种方法实现编辑、编译、连接和执行程序。TC 是一种集编辑、编译、连接、调试和执行为一体的集成开发环境，它将整个过程一气呵成，具有速度快、效率高、功能强等优点；另一种是命令行编译方式，它在 DOS 提示符下运行 TCC 命令，是一个传统的编译程序，它可以弥补 TC 命令的一些不足。

## 2.1 Turbo C 集成开发环境

在 DOS 下只需键入 TC 并回车后，即可进入 Turbo C 集成开发环境，如图 1-2 所示，它由主菜单行、编辑窗、信息窗和快速参考行等四部分组成。Turbo C 定义了两种屏幕状态，程序开发环境和程序运行时控制的屏幕，它们是相互独立的，通常 Turbo C 处于开发环境下。用户可在 Turbo C 环境下进行全屏幕编辑，利用窗口进行编辑、编译、连接、执行和环境设置等工作。下面简单介绍一下在 Turbo C 集成开发环境下运行 C 程序的步骤。

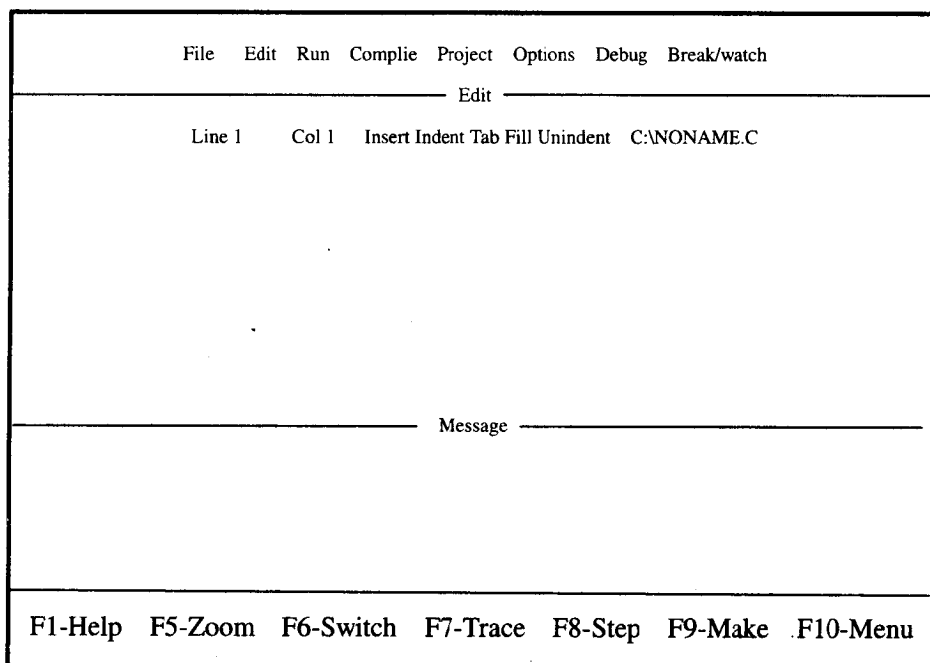


图 1-2 Turbo C 2.0主屏幕

### 2.1.1 集成进行法

#### (1) 输入源程序文件名

按“F3”键或在 File 菜单中选择 Load 项，在输入框中输入（或选择）源文件名，如果该文件存在，则调入内存并显示在屏幕上，如果不存在此文件名，则建立一个新文件。也可以在 File 菜单中选择 New 菜单项，直接建立一个默认名为“NONAME.C”的新文件。调入或建立一个文件后，自动转为编辑（Edit）状态。

#### (2) 编辑、保存源文件

在全屏幕编辑方式下，可以完成对源程序的输入、修改、删除等编辑操作，源程序编辑完成后，可以按“F2”键或在 File 菜单中选择 Save 项在磁盘上保存源程序或在 File 菜单中选择 Write to 项，把源程序改名保存。文件名由用户自己定义。

注意，在保存文件时系统默认的 C 源程序扩展名为“.C”

源程序的编辑也可使用其他的字处理软件，如 EDIT、WPS 等字处理软件。

#### (3) 编译、连接、运行源程序

按“Ctrl-F9”或在 Run 菜单中选择 Run 项后，即可完成编译、连接、运行三个过程。这时屏幕显示信息，表示此时系统正对源程序进行编译，编译完后系统对“.OBJ”文件进行连接，这个过程如没有严重的语法错误（error），系统自动将生成的.OBJ 与.EXE 文件存盘，并在屏幕上显示运行结果。

有时运行结果在屏幕上一闪而过，用户来不及看到时，可按“Ctrl-F5”键或在 Run 菜单中选择 User screen 项，则屏幕转换成用户屏幕，在此可看到程序运行结果。

Turbo C 中源程序经过编译、连接之后，生成的扩展名为“.EXE”的可执行文件。以后在 DOS 提示符下可键入该文件名执行这个文件。

#### (4)退出 Turbo C

按“Alt-X”或在 File 菜单中选择 Quit 项，就可退出 Turbo C 环境，返回到 DOS 提示符下。

### 选 学 与 提 高

#### 2.1.2 分步进行法

在 Turbo C 环境下，也可用传统的方法，将编译、连接过程分步进行，方法如下：

(1) Compile 菜单中选择 Compile to OBJ 项，将源程序.C 文件编译成“.OBJ”文件。当源程序有错时将显示错误信息，按任意键，显示源程序并将光标停留在出错处，同时在信息窗中显示相应的错误行和出错原因，可根据这些信息修改源程序，再进行编译，直到源程序正确为止。

(2)选择 Compile 菜单中的 Link EXE file 项，就可在指定的输出目录下，生成可执行文件“.EXE”

(3)按“Ctrl-F9”或在 Run 菜单中选择 Run，即可运行程序。按“Alt-F5”键或在 Run 菜单中选择 User screen 项，即可看到程序运行结果。

#### 2.1.3 工作环境设置

设置工作环境主要设置 Turbo C 集成环境是如何工作的，如指定包含文件和库文件的所在位置，输出文件存放于何处等。设置方法如下：

在图 1-2 的主菜单中选择“Options”选项后，在下拉菜单中有编译器、连接器等选项，其中最常用的是设置目录选项。这时可将光标移到“Directories”处按回车键，屏幕上又弹出一个窗口，如图 1-3 所示。在这个窗口中可以更改各种设置，如设置为如图 1-4 所示的情形。它表示 Turbo C 集成环境包含文件存放在 C:\TC\INCLUDE 子目录和 C:\TURBO\_C\INCLUDE2 子目录两个目录下；而库函数的路径在 C:\TC\LIB 子目录下编译连接后的目标文件和可执行文件指定存放在 D:\HOME 子目录中等。

为了保存所作的更改，还应该“Options”选项中选择“Save Options”选项，当出现一个会话框，要求输入配置文件名，一般为方便起见，可以取默认的“TCCONFIG.TC”，按回车键后，就在“TCCONFIG.TC”中更改了设置。

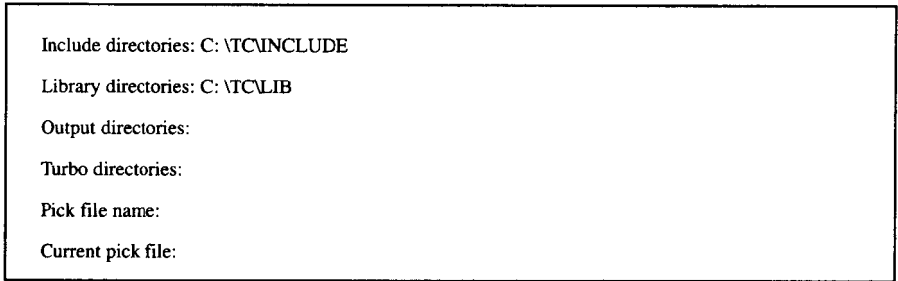


图 1-3 “Directories” 选项弹出菜单

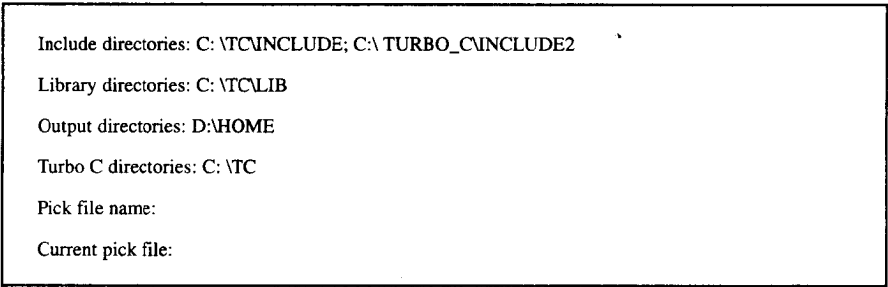


图 1-4 “Directories” 选项更改后的设置

## 2.2 Turbo C 的命令行编译连接

命令行编译连接是指在 DOS 提示符下，调用 Turbo C 下的 TCC.EXE 程序，完成对源程序的编译连接工作，它是一个传统方式的编译程序。如果要采用 C 程序和汇编语言混合编程，也要使用 TCC.EXE 程序进行编译连接。

关于 TCC.EXE 程序的使用方法，读者可参阅有关手册，这里只介绍 TCC.EXE 程序的简单使用方法。

TCC.EXE 的调用格式：

TCC [ 选择项 1 选择项 2 ... ] 文件名 1 文件名 2 ...

说明：

TCC.EXE 程序安装在 TC 目录下。

选择项是指定编译或连接时的选择项，常用的选择项如表 1-1 所示。每个选择项前均带有“-”，字母的大小写也有区别。文件名是指源程序“.C”或目标文件“.OBJ”或库文件“.LIB”。

表 1-1 命令行选择项

| 选择项 | 含 义            | 选择项   | 含 义              |
|-----|----------------|-------|------------------|
| -C  | 只编译.OBJ 文件，不连接 | -mT   | 极小内存编译模式         |
| -B  | 编译带有内嵌汇编指令的程序  | -mm   | 中内存编译模式          |
| -f  | 使用深入浮点仿真       | -mL   | 大内存编译模式          |
| -L  | 指定库文件路径        | -W    | 显示警告错误           |
| -I  | 指定包含文件路径       | -W-   | 不显示警告错误          |
| -S  | 输出一个汇编模块格式     | -n*** | 指定输出的目录 (***)    |
| -mS | 小内存编译模式        | -e*** | 指定生成的执行文件名 (***) |

(1) TCC.EXE 程序安装有 TC 目录下。

(2) 选择项是指定编译或连接时的选择项，常用的选择项如表 1-1 所示。每个选择项前均带有“-”，字母的大小写也有区别。文件名是指源程序“.C”或目标文件“.OBJ”或库文件“.LIB”。

(3) 如果不指定只编译不连接时，TCC 将完成编译和连接两步，对.LIB 只进行形式上的连接，对于标准库用户不用进行连接。

例如：

```
TCC -IC:\INCLUDE -LC:\LIB -eEXAMPLE ABC.C DEF.OBJ GHI
```

执行这个命令时，将源文件 ABC.C、目标文件 DEF.OBJ 和 GHI.C（命令中该文件没有后缀）这三个文件，分别进行编译（实际对 DEF.OBJ 不再进行编译），然后进行连接并生成一个名为 EXAMPLE.EXE 文件。

## 习 题 1

- 1-1 简述 C 语言的主要特点。
- 1-2 请叙述 C 语言的一般构成。
- 1-3 编写程序，在屏幕上输出如下信息：

```
=====
= I am a student. =
=====
```

- 1-4 简述在 Turbo C 环境中运行一个 C 程序的步骤。
- 1-5 C 语言源程序文件、目标文件和可执行文件的扩展名是什么？
- 1-6 Turbo C 下的 TC 和 TCC 有什么区别？
- 1-7 Turbo C 的环境设置一般需要设置哪些内容？设置完成后保存在哪个文件中？



## 第 2 章 数据类型及其运算

### 本章要点

- ◆ 了解 C 语言的数据类型
- ◆ 掌握标识符、常量的基本概念
- ◆ 掌握变量概念、变量的分类及变量的初始化方法
- ◆ 掌握整型、实型和字符型数据的形式
- ◆ 掌握各类数值型数据间的混合运算规则
- ◆ 掌握常见的各种运算符及表达式
- ◆ 了解运算符优先级与结合性

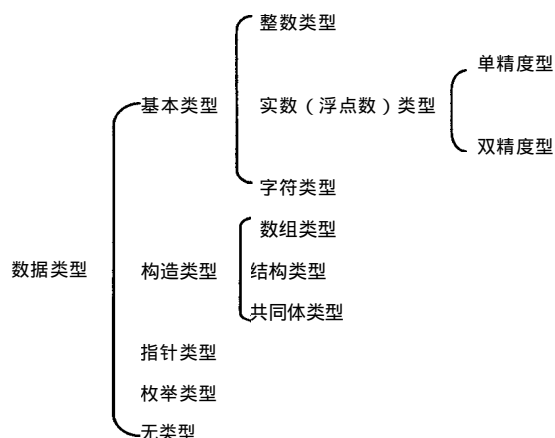
数据和运算符是构成表达式的两个最基本要素。本章将介绍 C 语言中的数据类型及表达式。

## 第 1 节 C 语言的数据类型

计算机学家 Nikiklaus.wirth 对计算机程序给出以下定义：

数据结构 + 算法 = 程序

这个定义说明了数据结构在计算机程序设计中起着特别重要的作用。C 语言中的数据结构在程序设计中是以数据类型的形式出现的，它可分为：基本类型、构造类型、指针类型、枚举类型和无类型，具体如下：



本章主要介绍基本类型，在基本类型之前加上相应的限定词：

short          短型  
long          长型  
unsigned      无符号型

各种组合的具体取值范围及位数如表 2-1。

表 2-1 基本数据组合类型(IBM-PC 机)

| 类 型               | 位 数 | 取 值 范 围                  |
|-------------------|-----|--------------------------|
| Char              | 8   | -128 ~ 127               |
| Unsigned char     | 8   | 0 ~ 255                  |
| Int               | 16  | -32768 ~ 32767           |
| short int         | 16  | -32768 ~ 32767           |
| long int          | 32  | -2147483648 ~ 2147483647 |
| Unsigned long int | 32  | 0 ~ 4294967295           |
| float             | 32  | 3.4E-38 ~ 3.4E+38        |
| Double            | 64  | 1.7E-308 ~ 1.7E+308      |
| long double       | 80  | 3.4E-4932 ~ 1.1E+4932    |

由表 2-1 可知：

(1) 加了限定词 **unsigned** 的数据类型比相应的不加限定词 **unsigned** 的数据类型的最大值的范围要扩大一倍，原因是无符号数的最高位也作为数值位。

(2) 未加限定词 **unsigned** 的数据类型的最高位是做为符号位的标志：最高位为 1 表示负数，为 0 表示正数。

## 第 2 节 常 量

### 1. 标识符

标识符是一个字符序列，它用来标识变量名、符号常量名、函数名、数组名、类型名和文件名等所起的名字。C 语言中标识符通常可分为以下几种。

#### 1.1 保留字 (或称保留关键字)

关键字在 C 程序中都代表着某个固定含义，在程序中不能另作它用，关键字都用小写字母表示 共有以下 32 个：

|          |        |        |          |          |          |         |
|----------|--------|--------|----------|----------|----------|---------|
| auto     | break  | case   | char     | const    | continue | default |
| do       | double | else   | enum     | extern   | float    | for     |
| goto     | if     | int    | long     | register | return   | short   |
| singed   | sizeof | static | struct   | switch   | typedef  | union   |
| unsigned | void   | while  | volatile |          |          |         |

#### 1.2 预定义标识符

C 语言中提供的库函数的名字和预编译处理命令都属于预定义标识符（见附录 3 和附录 5），C 语言把这类标识符作为统一的库函数名或预编译处理中的专用命令名使用，用户不要把这些预定义标识符另作它用。

#### 1.3 用户标识符

用户标识符是用户根据需要给程序中的变量、符号常量、函数、数组、指针等所起的名字，其命名规则如下：

- (1) 由英文字母、数字和下划线组成；
- (2) 第一个字符不能是数字；
- (3) 标识符最短为一个字符，最长可达 32 个字符，一般系统只识别前 8 个字符；
- (4) 英文字母大小写含义不同；
- (5) 不能使用保留关键字和预定义标识符。

例如：有下列标识符：

①abc1\_    ②\_abc1    ③1abc    ④ABC1\_  
⑤else    ⑥OH!    ⑦Abc1\_

其中 ①、②、③和⑦是合法的用户标识符，而 ④、⑤和⑥是不合法的用户标识符。

因为 违反第 2)条规定, 违反第 5)条规定, ⑥违反第(1)条规定。注意: 、 和 ⑦是不同的标识符。

## 2. 常量

常量是在程序运行过程中保持不变的量, 它可分为整型常量、实型常量、字符常量、字符串常量。它以直接常量和符号常量两种形式出现。直接常量就是通常所说的常数, 如: 21, -7, 3.5, 'B' 等等都是直接常量。符号常量是指定一个标识符代表一个常量, 一般用 `#define` 命令来定义 (在第 8 章中介绍), 习惯上符号常量名用大写字母来表示。

### 2.1 整型常量

整型常量也就是整型常数, 通常可分为十进制、八进制和十六进制三种。

(1) 十进制整型常量。可由 0 ~ 9 十个数字组成, 注意不能以数字 0 开头来表示一个十进制整型常量, 如: 918, -77, 0。

(2) 八进制整型常量。以数字 0 开头 后面数字可由 0~7 八个数字组成 如 0624, -0513 等, 而 083, 0912 是错误的, 因为八进制中不使用 8 和 9。

(3) 十六进制整型常量。以 0x 或 0X 开头 后面由数字 0 ~ 9 中和字母 A ~ F 共 16 个符号组成, 字母不区分大小写, 如 0X52, -0X5F, -0xCDE 等。

### 2.2 实型常量

实型常量即实数, 它有十进制小数形式和指数形式两种表示方法。

(1) 十进制小数形式。由整数部分、小数点和小数部分组成, 如 2.37, -5.244, 0.5 等。

(2) 指数形式。由有效数字、E (或 e) 和指数三部分组成。有效数字可用整型数表示, 也可用小数形式的实型数来表示, 有效数字前的正 (负) 号表示整个数的正 (负)。E (或 e) 是分隔有效数字与后面的指数的标志。指数部分必须是整型数, 表示 10 的整数次幂, 而且 E (或 e) 前面必须有数字。如 :1e2, 2.3E-4, -5e+6 等 而 E4, 7e3.2 是不合法的指数形式。有效数字与指数共同表示一个数的大小, 例如 1e2 表示  $1 \times 10^2$ , 3.2E7 表示  $3.2 \times 10^7$ , 2.3e-4 表示  $2.3 \times 10^{-4}$ 。

### 2.3 字符常量

字符常量是指括在单引号内的一个字符。如: 'X'、'B'、'6'、'&' 等等。

字符常量也是整型量, 它的值就是该字符在 ASCII 码表中的十进制编码值。如: 'X' 为 88, 'B' 为 66, '?' 为 63。所以字符常量也能参加数值运算。

【例 2-1】 写出下面程序的运行结果。

```
#include "stdio.h"
main()
{
    int x,y;
    x=36+'a';
    y=36-'a';
    printf("x=%d,x=%d\n",x,y);
```

}

运行结果如下：

x=133,y=-61

请大家思考一下：x=36+'0' 与 x=36+0 结果是否相同？

C 语言中还有特殊的字符常量——转义字符，它以反斜杠 (\) 开头 后面可以跟一个字符，或者是一个八进制常数或十六进制常数，其功能是将反斜杠 (\) 后面的字符转换成另一种含义，如表 2-2 所示。

表 2-2 转义字符

| 码  | 功 能  | 码    | 功 能          |
|----|------|------|--------------|
| \a | 报警   | \"   | 双引号          |
| \b | 退格   | \'   | 单引号          |
| \f | 走纸换页 | \\   | 反斜杠          |
| \n | 换行   | \v   | 纵向跳格         |
| \r | 回车   | \ddd | 1 到 3 位八进制数  |
| \t | 横向跳格 | \xhh | 1 到 2 位十六进制数 |

例如：'\102' 代表字符 'B'，'\107' 代表字符 'G'。

## 2.4 字符串常量

字符串常量是用一对双引号括起来的字符序列，如：

“this is a string” “ggftj ”，“5467.8FFG”

其中双引号是定界符，它不是串的一部分，不要把字符常量与字符串常量相混淆，字符串常量存储时 自动在字符串尾部加 '\0' 作为字符串结束的标志。

请大家思考一下：“a ”与'a'在存储时是否相同？

# 第 3 节 变量及其初始化

## 1. 变量

在程序运行过程中其值可以改变的量称为变量。 C 语言中的变量在使用前必须先定义。

变量定义的一般形式如下：

数据类型 变量表；

其中 数据类型必须是 C 语言中有效的数据类型，如整型、实型、指针型等等，变量表可以由一个或多个用逗号分隔的变量名构成，如：

float m,n;

其中，float 是数据类型 m、n 是变量名。

## 2. 整型变量

整型变量可分为普通整型(int)、短整型(short int 或 short)、长整型(long int 或 long)、无符号整型 包括 unsigned int 和 unsigned short int 及 unsigned long int)，它们不同之处表