

第 1 章 C 语言概述

本章要点：

- 了解 C 语言的基本特点
- 掌握 C 语言程序的基本结构
- 掌握 C 语言程序的编辑、编译、调试和运行的基本过程

1.1 C 语言的产生与特点

1.1.1 C 语言的产生与特点

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。它由早期的编程语言 BCPL (Bootstrap Combined Programming Language) 发展演变而来。1970 年，AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出比较先进并取名为 B 的语言，最后导致了 C 语言的问世。

随着微型计算机的日益普及，出现了许多的 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。为了改变这种状况，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。

C 语言发展如此迅速，而且成为最受欢迎的语言之一，主要是因为它具有强大的功能。许多著名的系统软件，如 DBASE+、DBASE 都是用 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，像 PC-DOS、WORDSTAR 等就是用这种方法编写的。目前，从微型机到大型计算机，不同类型的计算机以及不同的操作系统，都可以应用 C 语言进行程序设计。由于 C 语言本身具有许多特点，现在它已经成为在微、小、大、巨型计算机上，从系统程序设计到工程应用程序都能使用的一种高级程序设计语言。归纳起来 C 语言具有下列特点：

(1) C 语言是高、低级兼容语言。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。C 语言可以像汇编语言一样对位、字节和地址进行操作，而这三者是计算机最基本的工作单元。

(2) C 语言是结构化程序设计语言。结构化语言的显著特点是代码及数据的分离，即程序的各个部分除了必要的信息交流外，彼此相互独立。这种结构化方式可使程序层次清晰，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，这些函数可方便地调用，并具有多种循环、条件语句来控制程序流向，从而使程序完全结构化。

(3) 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的概念，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，可以用于实现决策目的。

(4) C语言适用范围大。C语言还有一个突出的优点，就是适合于多种操作系统，如DOS、UNIX，同时C语言也适用于多种机型。

1.1.2 C语言的版本、种类和发展

随着计算机的飞速发展，C语言也在不断地变革，不同的C语言种类和版本不断涌现。适用于各种不同操作系统（早期的UNIX、MS-DOS、CP/M86以及目前的Windows等）和不同机种的C语言系统相继出现，早期在MS-DOS下使用的C语言系统程序有Computer Innovation公司的C86，Lattice公司的LC，Microsoft公司推出的MSC（5.0或6.0版）以及Quick C，Turbo公司推出的Turbo C，Borland公司的Borland C等。当前的C语言系统主要是应用在Windows操作系统方面，比较著名的有Microsoft公司的Visual C，Borland公司的C++ Builder等。适用于不同操作系统和不同机种的C语言系统有几十种之多，但不同版本C语言的最主要的语句功能基本是一致的，可以圆满地解决C语言程序在不同机种、不同系统间的移植问题。但是，不同版本之间也存在着某些差异，它主要体现在标准函数库中函数的种类、格式和功能上稍有差别。在使用这些不同种类和版本的C语言时，应注意这些问题。

由于C语言具有众多特点，使之成为计算机应用当中最为广泛普及的语言之一。随着计算机的发展，C语言也得到了极大的发展。继DOS下应用的C语言之后，随着Windows操作系统的出现，C语言有了极大的变革，从C到C++直到当前的C#，从一般的程序设计到面向对象设计，C语言发生了根本性的变化。尽管C语言发生了极大的变革，基于DOS的程序设计也基本上被淘汰，但无论是哪种程序设计语言，其基本的方法和思想都是一致的。C语言作为一种计算机程序设计的基本语言，对于学习和掌握计算机程序设计的基本技能，依然具有不可忽视的作用。

本书以Turbo C 2.0作为教学蓝本，主要介绍C语言的基本特性、结构以及基本的应用方法。作为编程语言的基础，希望通过对本书的学习，能使计算机语言的初学者较好地掌握计算机程序设计的基本概念和基本方法，为今后进一步提高程序设计能力打下基础。

1.2 编写一个简单的C语言程序

本节通过一个简单的C程序例子，可以对C语言有一个初步的认识，进而了解编写C程序的一些基本问题。

C语言像其他语言一样，按其规定的格式和提供的语句，由用户编写应用程序。请看下面一段C源程序。

【例 1.1】已知 $a=1$ ， $b=2$ ，写一个程序，计算并输出和值： $sum=a+b+c$ 。其中 c 的值从键盘输入。

```
#include <stdio.h>          /*包含文件说明*/
main()                      /*主函数定义*/
{
    int a, b, c, sum;
    a=1;
    b=2;
```

```
scanf("%d", &c);           /*从键盘输入一个值 */
sum=a+b+c;
printf("sum=%d\n", sum);
}
```

编译并运行这个程序得出结果：

```
6✓           (输入 6 给变量 c)
sum=9
```

Turbo C 源程序主要有以下几个特点：

- (1) 程序一般用小写字母书写。
- (2) 大多数语句结尾必须要用“；”作为终止符，否则 C 语言认为该语句没有结束。但预处理命令、函数头和花括号“}”之后不能加分号。
- (3) 每个程序必须有一个而且只能有一个称做主函数的 main()函数。
- (4) 每个程序体，包括主函数和每个子函数，如上例中的 main() 函数 必须用一对花括号“{”和“}”括起来。
- (5) 一个较完整的 C 程序大致由包含文件（一组 #include<*.h> 语句）、用户函数说明部分、全程变量定义、主函数和若干子函数组成。在主函数和子函数中又包括局部变量定义、若干个 C 语言库函数、控制流程语句、用户函数的调用语句等。
- (6) 注释部分包含在“/*”和“*/”之间，在编译时它被 C 语言编译器忽略。
- (7) 源程序中可以有预处理命令（include 命令仅为其中的一种），预处理命令通常应放在源文件或源程序的最前面。
- (8) 标识符、关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符，也可不再加空格来间隔。

编写 C 语言程序时，应该注意以下几个方面：

- (1) 像其他一些语言一样，Turbo C 的变量在使用之前必须先定义其数据类型，未经定义的变量不能使用。定义变量类型应在可执行语句前面，如例 1.1 中 main() 函数的第一条语句：

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

就是变量定义语句，它必须放在第一条执行语句“a=1;”的前面。

- (2) C 语言中的字符有大、小写的区别。C 语言习惯上用小写字母，只有常量、宏定义等用大写。所有的关键字必须小写，如 if-else、while、break 等只能小写。同样，使用变量时也必须注意变量名的大、小写，相同字母的大、小写代表不同的变量，如 A0 与 a0 表示两个不同的变量。

- (3) 在程序中使用好“{ }”。“{ }”括起来的部分通常表示了程序的某一层结构。

- (4) 注意使用程序行的缩格。低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写，以便看起来更加清晰，增加程序的可读性。

- (5) 在程序中使用注释符号“/* */”。对 C 语言程序中的函数模块功能、变量的含义及语句的功能等进行注释，以增加程序的可读性。在对 C 语言程序编译时，源程序中的注释部分不参与编译，因此 C 语言源程序中注释的多少不影响程序的执行效率。所以，应该

说一个好的、有价值的源程序都应当加上必要的注释。

C 语言是一种“自由格式”的语言，书写格式非常灵活，除了一些简单的限制外，没有什么严格限制。写程序时可完全根据自己的爱好去书写，不论用什么格式写程序都不影响程序本身的意义。但如果写程序时格式太随便会使整个程序阅读起来非常不方便，同时也使得程序层次不明确。建议用 Turbo C 编写比较大的程序时，一行一条语句，遇到嵌套语句向后缩进，必要时对程序加上注释行。这样可以使程序结构清楚，易于阅读、维护和修改。

在编程时应力求遵循这些规则，以养成良好的编程风格。

通过以上介绍，可以得出 Turbo C 源程序的一般形式如下：

```
包含文件
子函数类型说明
全程变量定义
main()
{   局部变量定义
    <程序体>
}
```

1.3 Turbo C 开发环境的使用

1.3.1 Turbo C 的产生与发展

Turbo C 是美国 Borland 公司的产品，Borland 公司是一家专门从事软件开发、研制的大公司。该公司相继推出了一套 Turbo 系列软件如 Turbo BASIC, Turbo Pascal, Turbo Prolog, 这些软件很受用户欢迎。该公司在 1987 年首次推出 Turbo C 1.0 产品，其中使用了全然一新的集成开发环境，即使用了一系列下拉式菜单，将文本编辑、程序编译、连接以及程序运行一体化，大大方便了程序的开发。1988 年，Borland 公司又推出 Turbo C 1.5 版本，增加了图形库和文本窗口函数库等，而 Turbo C 2.0 则是该公司 1989 年出版的。Turbo C 2.0 在原来集成开发环境的基础上增加了查错功能，并可以在 Tiny 模式下直接生成 .COM 文件，还可对协处理器（支持 8087/80287/80387 等）进行仿真。Borland 公司后来又推出了面向对象的程序软件包 Turbo C++，它继承和发展了 Turbo C 2.0 的集成开发环境，并包含了面向对象的基本思想和设计方法。1991 年，为了适用 Microsoft 公司的 Windows 3.0 版本，Borland 公司又将 Turbo C++ 做了更新，即 Turbo C 的新一代产品 Borland C++ 问世了。

1.3.2 Turbo C 2.0 的安装和启动

Turbo C 2.0 可运行于 PC 系列微机，可在 DOS 环境中使用。Turbo C 2.0 的安装非常简单，只要将安装盘的 1# 盘插入软驱（如驱动器 A）中，在 DOS 下的“A:\>”下键入：

```
INSTALL
```

即可开始安装，只要在安装过程中按屏幕对盘号的提示，顺序插入各个软盘，就可以顺利地进行安装，安装完毕后将在 C 盘根目录下建立一个 TC 子目录，在 TC 下还建立了两个

子目录 Lib 和 Include ,其中 Lib 子目录中存放库文件 ,Include 子目录中存放所有头文件。本书中所提到的 Turbo C 的操作 是假设 Turbo C 已经安装在 C 盘的 C:\tc 目录中。如果在实际操作中,安装的位置不是在 C:\tc 操作时就应该按 Turbo C 实际安装的位置使用正确的路径。

运行 Turbo C 2.0 时,只要在 TC 子目录下键入 TC 并回车即可进入 Turbo C 集成开发环境。

1.3.3 Turbo C 2.0 集成开发环境的使用

进入 Turbo C 2.0 集成开发环境,是在启动 DOS 操作系统后,在 C 盘进入 TC 目录,键入命令 TC 就可以了:

```
c:\tc\✓
```

这里,“ ✓ ”表示为回车符。书中所有的“ ✓ ”均表示回车符 不再另行说明
这时,屏幕出现一个窗口,窗口示意图如图 1.1 所示。

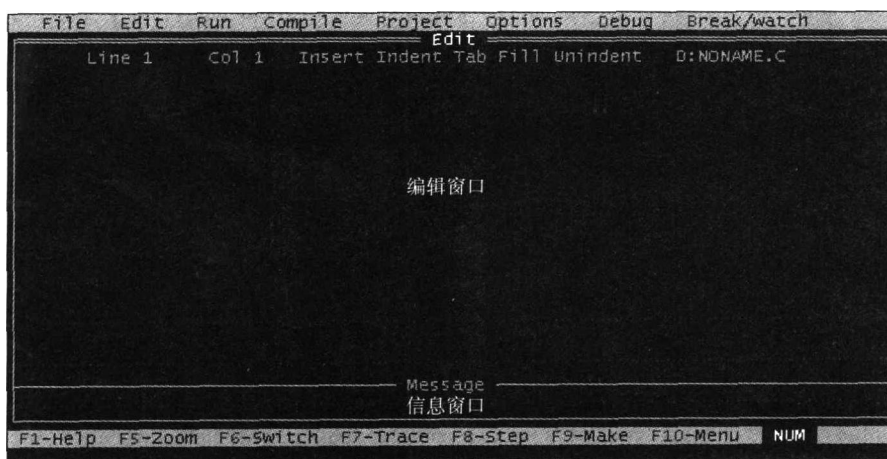


图 1.1 Turbo 2.0 主屏幕

图 1.1 中,顶上一行为 Turbo C 2.0 的主菜单,中间窗口为编辑窗口,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。下面详细介绍主菜单的内容。

在 Turbo C 2.0 主屏幕顶上一行,显示下列内容:

```
File Edit Run Compile Project Options Debug Break/watch
```

除 Edit 外,其他各项均有子菜单,只要用 Alt 键加上某项中第一个字母(即大写字母),就可进入该项的子菜单中。如 Alt+F,就进入 File 菜单项。

以下介绍菜单中主要菜单项的功能,其他不常用的菜单项读者在需要时可查阅相关资料。

1. File (文件) 菜单

按 Alt+F 组合键可进入 File 菜单,该菜单包括以下内容。

(1) Load (加载)。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。该项的快捷键为 F3,即只要在主菜单中按 F3 即可进入该项,而不需要先进入 File 菜单再选此项。

(2) Pick(选择)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。其快捷键为 Alt+F3。

(3) New (新建)。开始编辑一个新的程序文件,默认文件名为 NONAME.C,存盘时可改名。

(4) Save (存盘)。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名。其快捷键为 F2。

(5) Write to (存盘)。可由用户给出文件名将编辑区中的文件存盘,若给出的文件名已存在,则询问要不要覆盖。

(6) Directory (目录)。显示目录及目录中的文件并可由用户选择。

(7) Change dir (改变目录)。显示当前目录,用户可以改变显示的目录。

(8) OS shell(暂时退出)暂时退出 Turbo C 2.0 到 DOS 提示符下,此时可以运行 DOS 命令,若想回到 Turbo C 2.0 中,只要在 DOS 状态下键入 EXIT 即可。

(9) Quit (退出)。退出 Turbo C 2.0,返回到 DOS 操作系统中。其快捷键为 Alt+X。

说明:以上各项可用光标键在菜单中移动到相应位置后,使用回车键选择执行。也可用每一项的第一个大写字母直接选择。若要退到主菜单或从它的下一级菜单列表框退回,均可用 Esc 键,Turbo C 2.0 所有菜单均采用这种方法进行操作,以下不再说明。

2. Edit(编辑)菜单

按 Alt+E 组合键可进入编辑菜单,若再回车,则光标出现在编辑窗口,此时用户可以进行文本编辑。

可用 F1 键获得有关编辑方法的帮助信息。与编辑有关的功能键如下:

F1	获得 Turbo C 2.0 编辑命令的帮助信息
F5	扩大编辑窗口到整个屏幕
F6	在编辑窗口与信息窗口之间进行切换
F10	从编辑窗口转到主菜单

编辑命令简介:

PageUp	向前翻页
PageDn	向后翻页
Home	将光标移到所在行的开始
End	将光标移到所在行的结尾
Ctrl+Y	删除光标所在的一行
Ctrl+T	删除光标所在处的一个词
Ctrl+KB	设置块开始
Ctrl+KK	设置块结尾
Ctrl+KV	块移动
Ctrl+KC	块拷贝
Ctrl+KY	块删除

Ctrl+KR	读文件
Ctrl+KW	存文件
Ctrl+KP	块文件打印
Ctrl+F1	当光标所在处为库函数时，获得有关该函数的帮助信息
Ctrl+Q[查找双界符的后匹配符
Ctrl+Q]	查找双界符的前匹配符

说明：

(1) Turbo C 2.0的定界符包括以下几种符号：

花括号	{ 和 }
尖括号	< 和 >
圆括号	(和)
方括号	[和]
注释符	/* 和 */
双引号	
单引号	

(2) Turbo C 2.0 在编辑文件时还有一种功能，就是能够自动缩进，即光标定位和上一个非空字符对齐。在编辑窗口中， Ctrl+OL 为自动缩进开关的控制键。

3. Run (运行)菜单

按 Alt+R 组合键可进入 Run 菜单，该菜单包括以下各项。

(1) Run (运行程序)。运行由 Project/Project name 项指定的文件名或当前编辑区的文件。如果对上一次编译后的源代码未做过修改，则直接运行到下一个断点（没有断点则运行到结束）。要是源代码已经被修改，则自动先进行编译、连接，然后再运行。其快捷键为 Ctrl+F9。

(2) Program reset (程序重启)。中止当前的调试，释放分配给程序的空间。其快捷键为 Ctrl+F2。

(3) Go to cursor (运行到光标处)。调试程序时使用，选择该项可使程序运行到光标所在行。光标所在行必须为一条可执行语句，否则提示错误。其快捷键为 F4。

(4) Trace into(跟踪进入)。使用 Trace into 项，是在执行一条调用其他用户定义的子函数的程序行时，跟踪执行到该子函数内部。其快捷键为 F7。

(5) Step over (单步执行)。执行当前函数的下一条语句，即使用户函数调用，执行时不会跟踪进函数内部。其快捷键为 F8。

(6) User screen(用户屏幕)显示程序运行时在屏幕上显示的结果。其快捷键为 Alt+F5。

4. Compile(编译)菜单

按 Alt+C 组合键可进入 Compile 菜单，该菜单包括以下内容。

(1) Compile to OBJ (编译生成目标文件)。将一个 C 源文件编译生成 .obj 目标文件，同时显示生成的文件名。其快捷键为 Alt+F9。

(2) Make EXE file (生成可执行文件)此命令生成一个 .exe 的文件并显示生成的 .exe 文件名。

(3) Link EXE file (连接生成可执行文件)。把当前 .obj 文件及库文件连接在一起生成 .exe 文件。

(4) Build all (建立所有文件)。重新编译项目里的所有文件并进行装配,生成 .exe 文件。该命令不做过时检查(上面的几条命令要做过时检查,即如果目前项目里源文件的日期和时间与目标文件相同或更早,则不再对源文件进行编译)。

(5) Primary C file (主 C 文件)。使用这项命令可以指定主文件。指定主文件后,在以后的编译中如没有项目文件名,则编译此项中规定的主 C 文件;如果编译中有错误,则将此文件调入编辑窗口,而不管目前窗口中是不是主 C 文件。

(6) Get info (获得有关信息)。使用这项命令可以获得包括当前路径、源文件名、源文件字节大小、编译中的错误数目、可用空间等有关的信息。

5. Project (项目) 菜单

按 Alt+P 组合键可进入 Project 菜单。当源程序被编辑成若干个文件时,通过这个菜单将所有的源程序组成一个项目,此后就可以对这个项目进行编译和连接。该菜单包括以下内容。

(1) Project name (项目名)。项目名具有 .prj 的扩展名,其中包括将要编译、连接的文件名。例如,有一个程序分别由 file1.c、file2.c 和 file3.c 组成,要将这 3 个文件编译装配成一个 file.exe 的执行文件,可以先建立一个 file.prj 的项目文件,其内容如下:

```
file1.c
file2.c
file3.c
```

此时将 file.prj 放入 Project name 项中,以后进行编译时将自动对项目文件中规定的 3 个源文件分别进行编译,然后连接成 file.exe 文件。

如果其中有些文件已经编译成 .obj 文件而又没有修改过,可直接写上 .obj 扩展名。此时将不再编译而只进行连接。例如:

```
file1.obj
file2.c
file3.c
```

将不对 file1.c 进行编译,而直接连接。

说明:当项目文件中的每个文件无扩展名时,均按源文件对待,另外,其中的文件也可以是 .obj 目标文件,但必须写上扩展名 .lib。

(2) Break make on (中止编译)。由用户选择是否有 Warning(警告)、Errors(错误)、Fatal errors(致命错误)时或 Link(连接)之前退出 Make 编译。

(3) Auto dependencies(自动依赖)。当开关置为 On 时,编译时将检查源文件与对应的 .obj 文件的日期和时间;若开关置为 Off,则不进行检查。

(4) Clear project (清除项目文件)。清除 Project/Project name 中的项目文件名。

(5) Remove messages (删除信息)。把错误信息从信息窗口中清除掉。

6. Options(选择菜单)

按 Alt+O 组合键可进入 Options 菜单,该菜单对初学者来说要谨慎使用。

(1) **Compiler (编译器)**。本项选择又有许多子菜单,可以让用户选择硬件配置、存储模型、调试技术、代码优化、对话信息控制和宏定义等内容。

(2) **Linker (连接器)**。本项菜单设置有关连接的选择项。

(3) **Environment (环境)**。本菜单规定是否对某些文件自动存盘以及制表键和屏幕大小的设置,其中:

Config auto save 是否在 **Run**、**Shell** 或退出集成开发环境之前,将所做的配置改动存入配置文件中

Edit auto save 是否在 **Run** 或 **Shell** 之前,自动存储编辑的源文件

Backup file 是否在源文件存盘时产生后备文件 (.bak 文件)

Tab size 设置制表键大小,默认为 8

(4) **Directories (路径)**。本菜单规定编译、连接所需文件的路径,其中:

Include directories 包含文件的路径,多个子目录用“;”分开

Library directories 库文件路径,多个子目录用“;”分开

Output directory 输出文件 (.obj、.exe、.map 文件)的目录

Turbo C directory Turbo C 所在的目录

Pick file name 定义加载的 pick 文件名,如不定义则从 **current pick file** 中读取

(5) **Arguments (命令行参数)**。允许用户使用命令行参数。

(6) **Save options (存储配置)**。保存所有选择的编译、连接、调试和项目到配置文件中,默认的配置文件为 **tcconfig.tc**。

(7) **Retrieve options**。装入一个配置文件到 **TC** 中,**TC** 将使用该文件的选择项。

7. Debug (调试) 菜单

按 **Alt+D** 组合键可选择 **Debug** 菜单,该菜单主要用于程序调试时的查错。

8. Break/watch (断点及监视表达式)

按 **Alt+B** 组合键可进入 **Break/watch** 菜单,该菜单进行程序断点控制和监视表达式的操作。

上面只对常用的菜单操作进行了介绍,如果要进一步了解选择菜单的详细内容以及更多的功能(如高级编辑功能、项目管理、调试功能),请参阅有关资料。

1.3.4 Turbo C 2.0 的配置文件

所谓配置文件是包含 **Turbo C 2.0** 有关信息的文件,其中保存有编译、连接的选择和路径等信息。可以用下述方法建立 **Turbo C 2.0** 的配置文件。

(1) 建立用户自己命名的配置文件。可以从 **Options** 菜单中选择 **Options/Save options** 命令,将当前集成开发环境的所有配置存入一个由用户命名的配置文件中。下次启动 **TC** 时只要在 **DOS** 下键入:

```
tc/c<用户命名的配置文件名>
```

就会以这个配置文件中的内容作为 **Turbo C 2.0** 的选择。

(2) 设置 **Options/Environment/Config auto save** 为 **On**,则退出集成开发环境时,当前的设置会自动存放到 **Turbo C 2.0** 的配置文件 **tcconfig.tc** 中。**Turbo C** 在启动时会自动寻找

这个配置文件。

(3)用 `tcinst` 设置 Turbo C 的有关配置并将结果存入 `tc.exe` 中。Turbo C 在再次启动时，若没有找到配置文件，则取 `tc.exe` 中的默认值。

应该在进行程序编辑、编译之前，先进行一些必要的配置。主要的配置内容有以下几项。下面的配置假设 Turbo C 2.0 安装在 C 盘的 TC 目录下，通常设定如下：

Include directories	包含文件的文件目录，设定为 <code>c:\tc\include</code>
Library directories	库文件的文件目录，设定为 <code>c:\tc\lib</code>
Turbo C directory	Turbo C 目录，设定为 <code>c:\tc</code>

设定后，再使用 Options 菜单中的 Save options 菜单项，将设定保存在 `c:\tc` 目录下，所保存的文件名默认为 `tcconfig.tc`。

1.4 C 程序的编译与运行

C 语言程序必须事先经过编译、连接后才能运行，它不像运行解释型的 BASIC 程序那样直接启动解释程序即可边解释边执行程序。C 语言程序的基本开发过程如图 1.2 所示。

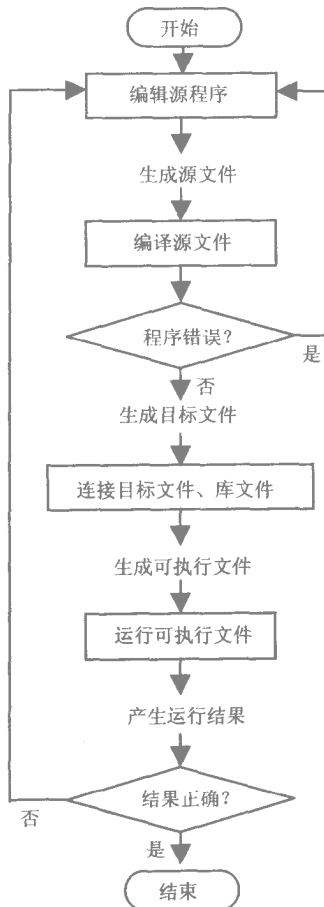


图 1.2 C 语言程序的开发过程

1.4.1 编辑源程序

大部分 C 语言编译系统带有独立的编辑程序，可用于输入或修改源程序。也可使用操作系统提供的编辑程序，如 MS-DOS 的 `edit` 命令，或 Windows 操作系统的记事本，或其他专门的文本编辑器。源程序经编辑程序由键盘输入编辑形成源程序，源程序以文本文件的形式存储在软盘或硬盘中。源程序文件的名字由用户选定，但扩展名均为 `.c`，也就是 C 语言程序源文件均以 `c` 作为文件的扩展名。

例如：

```
filename.c
```

```
fl.c
```

1.4.2 编译源文件

在程序运行之前，必须用系统提供的编译程序对源程序文件进行编译。编译程序要对源程序进行语法检查，若没有发现错误，就对源程序进行编译，生成目标文件。目标文件是由目标代码组成的，目标代码是二进制机器码，在 MS-DOS 下目标文件带有 `.obj` 文件扩展名。如果在编译过程中发现有错误，则显示错误信息，这时，程序员应使用编辑工具对程序进行编辑修正，排除程序错误后重新进行编译，直到编译正确为止。

1.4.3 连接目标文件及库文件

源程序经编译程序编译后产生的目标文件虽然是由机器指令代码组成，但是它还是不能直接在机器上运行，它是一个浮动的程序模块。也就是说，编译后程序的机器码指令在内存中的地址并未绝对地确定，因此，应该将程序重定位在确定的绝对地址上，这部分工作由连接程序完成。另一方面，所有的 C 语言编译程序都是和标准函数库文件一起提供的，保存在函数库文件中的函数也都是可重定位的。因此，当所编写的 C 语言程序调用了标准库函数或者其他入编写的函数时，必须通过连接程序（`Link`）将用户编写的程序产生的目标代码同标准函数库文件中找到的目标码结合起来，这个过程称为连接。在连接过程中，对程序和函数进行重新定位。连接后产生的文件称为可执行文件，可执行文件以 `.exe` 作为文件扩展名。以 `.exe` 为文件扩展名的文件可以在机器上直接运行。

对一个程序编译的方法是：在文件存盘后，按 `F10` 键返回到主菜单，然后将亮条移至 `Compile` 菜单处，选择使用 `Compile to OBJ` 命令将程序编译生成目标码，或使用 `Make EXE file` 命令，生成可执行文件；也可以使用 `Link EXE file` 命令将当前 `.obj` 文件及库文件连接在一起生成 `.exe` 文件；或者使用 `Build all` 命令将所有源程序编译并连接，建立可执行文件。

在编译连接过程中，若程序有错，则在屏幕底部的 `Message` 窗口显示出错及警告信息。这时可以再进行编辑修改，修改完成后再重新进行编译、连接。

1.4.4 运行程序

经过编译和连接后产生的可执行文件是 `.exe` 文件，只需在操作系统下键入这个可执行文件的文件名，程序即可启动运行。也可以利用主菜单中的 `Run` 命令或直接按 `Ctrl+F9` 来运行程序。

C 语言允许进行分割编译，即可将较大的程序分成若干块，形成若干个文件，每一个文件单独编辑成为 .c 源文件并可单独进行编译。这种特性对于由多人共同开发同一个软件是很有用的。当所有的文件编译完成后，再对这些编译过的文件进行连接。连接程序分别将编译后产生的所有目标文件和库函数中的函数进行连接，形成一个完整的可执行文件。

使用 Run 命令运行程序后，仍会回到 Turbo C 集成环境。这时若想看运行结果，可以按 Alt+F5 组合键转到用户屏幕，程序运行的结果显示在用户屏幕上。看完后可按任意键返回到 Turbo C 集成环境。

本章小结

本章主要介绍了 C 语言的来源及特点，C 语言程序的基本结构，并且还详细介绍了 Turbo C 集成环境的应用，装入 Turbo C 源程序的方法及新建源程序的方法，在 Turbo C 集成环境中输入、编译、调试和运行 C 程序的基本方法和步骤。

在 C 语言编写和开发过程中要注意以下几个问题：

- (1) 编写程序应该规范，培养良好的程序设计风格。
- (2) 注意 C 语言的语法要求，注意关键字、标识符、常量、运算符、分隔符以及注释符号的使用。
- (3) 标准 C 语言有 32 个关键字，不能被重新定义，并有一些标识符，如 printf、scanf 等被系统标准库函数占用，不能再定义为其他变量名。
- (4) C 语言程序以一个或多个函数组成，至少有一个 main() 函数。
- (5) C 语言程序由语句组成，通常每一行为一条基本语句，由 “ {} ” 括起来的部分为一组复合语句。
- (6) C 语言程序要先输入、编辑源程序，再对源程序进行编译、连接，最后才可以运行。
- (7) Turbo C 是 C 语言的编辑、编译和连接的一个环境，要熟练掌握源程序的编辑、保存、编译、连接及运行的操作方法，了解程序调试的基本概念。

习 题

- 1.1 C 语言程序由哪几部分组成？C 语言程序结构有什么特点？
- 1.2 C 语言的主要特点是什么？
- 1.3 C 语言的主要用途是什么？C 语言与其他高级语言比较有什么特点？
- 1.4 请指出下列 C 语言程序中的错误，并改正。

```
/* File name: error.c*/
/* The program is error program!
main();
{   INT a, b, c, sum;
    a:=1; b:=2; c:=3;
    SUM=A+B+C;
    printf("SUM=%d\n", sum)
}
```

1.5 理解和记忆 C 语言操作的过程。

1.6 编写一个 C 程序，输入变量 a、b 和 c，计算并输出其平均值 $ave=(a+b+c)/3$ 。

上机实验

1. 目的和要求

(1) 通过上机，使学生掌握如何使用 Turbo C 在微机上录入、编辑、编译、连接、运行一个 C 语言程序。

(2) 熟悉 Turbo C 的窗口和菜单，掌握本章中介绍的几个菜单项的意义和操作使用方法，熟悉编辑命令的使用。

2. 实验内容

(1) 输入以下程序，熟悉 Turbo C 系统的上机方法与步骤。

```
/* Hello.c is an example of program */
#include <stdio.h>
main()
{   printf("Hello C!\n");
}
```

(2) 将本章例 1.1 程序输入并编译运行，查看运行结果。

3. 思考题

(1) 在运行 Turbo C 系统前先运行中文系统，在 hello.c 程序中的一对双引号之间写一句中文，看看程序运行结果如何。

(2) 在 hello.c 程序中加入一些空格、制表符、换行符，看看在什么情况下会出现问题，什么情况下不会出现问题。

第 2 章 C 语言程序基础

本章要点：

- 掌握 C 语言中的基本数据类型及其使用方法
- 掌握 C 语言中变量的使用方法
- 掌握 C 语言中各种运算符的运算规则、优先级等
- 掌握顺序程序设计的编程方法和技巧

程序设计的目的是通过程序来解决问题，通过对信息的获取、操作、处理来完成。在程序设计中必须掌握的内容是数据类型、数据输入、数据处理、数据输出；在对程序处理过程中必须掌握的方法是程序的顺序执行、条件执行、循环执行和函数（子程序）等处理方法。

本章介绍的是 C 语言基本的数据类型、常用运算符、表达式、赋值操作以及标准输入 / 输出函数等。通过本章的学习，可以掌握计算机中数据类型的基本概念，建立一般的表达式及编写简单的程序。

2.1 基本的数据类型

在 Turbo C 语言中，每个变量在使用之前必须定义其数据类型。Turbo C 的数据类型可分为基本数据类型、构造数据类型、指针类型和空类型四大类。基本数据类型最主要的特点是，其值不可以再分解为其他类型。基本数据类型有以下几种，它们是：整型（int）、字符型（char）、实型（float）。本节只介绍 Turbo C 的基本数据类型，其他几种数据类型将在以后的章节中介绍。

2.1.1 整型（int）

1. 整型数说明

整型可分为四种，分别是基本型（int）、短整型（short）、长整型（long）和无符号型（unsigned）：

signed short int	有符号短整型数说明，简称为 short 或 int，字长为 2 字节共 16 位二进制数，数的范围是 -32768~32767
signed long int	有符号长整型数说明，简称为 long，字长为 4 字节共 32 位二进制数，数的范围是 -2147483648~2147483647
unsigned short int	无符号短整型数说明，简称为 unsigned int，字长为 2 字节共 16 位二进制数，数的范围是 0~65535

`unsigned long int` 无符号长整型数说明，简称为 `unsigned long`，字长为 4 字节共 32 位二进制数，数的范围是 0~4294967295。

2. 整型变量定义

可以用下列语句定义整型变量：

```
int a, b;           /* a、b 被定义为有符号短整型变量 */
unsigned long c;   /* c 被定义为无符号长整型变量 */
```

3. 整型常数表示

按不同的进制区分，整型常数有以下三种表示方法：

(1) 十进制数。十进制常数没有前缀以非 0 数字开始，其数码为 0~9，如 220、-560、45900。

(2) 八进制数。八进制整常数必须以 0 开头，即以 0 作为八进制数的前缀，数码取值为 0~7，如 06、0106、057。

(3) 十六进制数。十六进制整常数的前缀为 0X 或 0x，其数码取值为 0~9、A~F 或 a~f，如 0X0D、0XFF、0x4e。

另外，可在整型常数后添加一个“L”或“l”字母表示该数为长整型数，如 22L、0773L、0Xae4l。

值得注意的是，在数学中，整数是一个无限集合，但在特定的机器系统中，所表示数的范围是有限的。如在 16 位字长的机器上，基本整型的长度也为 16 位，因此表示的数的范围也是有限定的。用十进制无符号整常数的范围应为 0~65535，用有符号数表示的十进制数的范围为 -32768~+32767。十六进制无符号数的表示范围为 0X0~0XFFFF 或 0x0~0xFFFF。如果使用的数超过了上述范围，就必须用长整型数来表示。

假设长整数值为 158L，它和基本整常数 158 在数值上并无区别。但对于 158L 来说，因为是长整型数，C 编译系统将为它分配 4 个字节存储空间。而对于基本整常数 158，只分配 2 个字节的存储空间。因此，在运算和输出格式上要予以注意，避免出错。

2.1.2 浮点型 (float)

实型数也就是浮点数，使用浮点表示方法可以表示较大范围的实型数。

1. 浮点数说明

浮点数有两种类型：

<code>float</code>	单精度浮点数，字长为 4 个字节共 32 位二进制数，数的范围是 $3.4 \times 10^{-38} \sim 3.4 \times 10^{+38}$ ，有效数字 6~7 位
<code>double</code>	双精度浮点数，字长为 8 个字节共 64 位二进制数，数的范围是 $1.7 \times 10^{-308} \sim 1.7 \times 10^{+308}$ ，有效数字 15~16 位

浮点数均为有符号浮点数，没有无符号浮点数。浮点数只以十进制表示。

2. 浮点型变量定义

可以用下列语句定义浮点型变量：

```
float a, f;      /* a、f被定义为单精度浮点型变量 */
double b;      /* b被定义为双精度浮点型变量 */
```

3. 浮点常数表示

在 C 语言中，实数有以下两种形式：

(1) 十进制小数形式。它由数字和小数点组成，如 12.3、0.345、456.00 均为十进制小数形式。

(2) 指数形式。由十进制数加阶码标志“e”或“E”以及阶码（只能为整数，可以带符号）组成。其一般形式为 aEn （ a 为十进制数， n 为十进制整数），其值为 $a \times 10^n$ 。例如，2.1E5 表示 2.1×10^5 。注意：字母“e”或“E”之前必须有数字，之后必须为整数。

一个实型数据一般在内存中占 4 个字节。与整型数据的存储方式不同，实型数据是按照指数形式存储的。系统把一个实型数据分成小数部分和指数部分分别存放，指数部分采用规范化的指数形式。例如，+29.56、-56.33、-6.8e-18、6.365。

浮点数在计算机中是用二进制数来表示的。每个浮点数所占的 4 个字节中，究竟用多少位来分别表示小数部分和指数部分，标准 C 并无具体规定，要由各 C 编译系统来决定。但多数编译系统是用 24 位表示小数部分，用 8 位表示指数部分。

由于实型变量是由有限的单元组成，因此能够提供的有效数字总是有限的。这样一来，实数的存储总是有误差的。

使用浮点数时要注意以下几个问题：

- (1) 浮点常数只有一种进制（十进制）。
- (2) 所有浮点常数都被默认为 double。
- (3) 绝对值小于 1 的浮点数，其小数点前面的零可以省略。例如，0.22 可写为 .22，-0.0015E-3 可写为 -.0015E-3。
- (4) Turbo C 默认格式输出浮点数时，最多只保留小数点后 6 位。

2.1.3 字符型 (char)

1. 字符型数说明

字符型用 char 表示，一般用 1 个字节（8 位二进制数）来存储一个字符，实际上存放的是该字符的 ASCII 码值。因此，字符型和整型关系密切，可把字符型看做一种特殊的整型。事实上，字符型数据和整型数据之间经常混合使用。

加上不同的修饰符，可以定义有符号和无符号两种类型的字符型变量，例如：

```
char a;          /*a 被定义为有符号字符变量*/
unsigned char i; /*i 被定义为无符号字符变量*/
```

有符号字符型数取值范围为 -128~127，无符号字符型数取值范围是 0~255。因此在 Turbo C 语言中，字符型数据在操作时将按整型数处理。如果某个变量定义成 char，则表

明该变量是有符号的，即它将转换成有符号的整型数。

C 语言中规定对 ASCII 码值大于 0x80 的字符将被认为是负数。例如，ASCII 值为 0x8c 的字符，定义成 char 时，被转换成十六进制的整数 0xff8c。这是因为，当 ASCII 码值大于 0x80 时，该字节的最高位为 1，计算机会认为该数为负数，对于 0x8c 表示的数实际上是 -74（8c 的各位取反再加 1），而 -74 转换成 2 个字节整型数并在计算机中表示时就是 0xff8c 对 0074 各位取反再加 1）。因此，只有定义为 unsigned char 时，0x8c 转换成整型数时才是 8c。这一点在处理大于 0x80 的 ASCII 码字符时（如汉字码）要特别注意。一般汉字均定义为 unsigned char（在以后的程序中会经常碰到）

另外，也可以定义一个字符型数组（关于数组后面将做详细介绍），此时该数组表示一个字符串。例如：

```
char str[10];
```

表示为一个名为 str 的字符串。计算机在编译时，将留出连续 10 个字符的空间，即 str[0] 到 str[9] 共 10 个变量，但只有前 9 个供用户使用，第 10 个字符 str[9] 用来存放字符串终止符 NULL，即“\0”。终止符是编译程序自动加上的，这一点应特别注意。

2. 字符常量表示

能用符号表示的字符可直接用单引号括起来表示，如 'a', '9', 'Z'，也可用该字符的 ASCII 码值表示，例如，十进制数 85 表示大写字母“U”，十六进制数 0x5d 表示“J”，八进制数 0102 表示大写字母“B”。一些不能用符号表示的控制符，只能用 ASCII 码值来表示，如十进制数 10 表示换行，十六进制数 0x0d 表示回车，八进制数 033 表示 Esc；也可以使用另外一种表示方法，如用“\033”表示 Esc，这里“\0”符号后面的数字表示十六进制的 ASCII 值，当然，这种表示方法也适用于可直接用符号表示的字符。

另外，有些常用的字符用以下特殊规定来表示：

规定符	等价于	含义
\f	\X0C	换页
\r	\X0D	回车
\t	\X09	制表键
\n	\X0A	换行
\\	\X5C	\ 符
'\'	\X27	' 符（单引号）
'\"'	\X22	" 符（双引号）

对于字符串常量，一般用双引号括起来表示，如 "Hello Turbo C2.0"。

2.1.4 无值型 (void)

无值型字节长度为 0，主要有两个用途：一是明确地表示一个函数不返回任何值；二是产生一个同一类型指针（可根据需要为其动态分配内存）。例如：

```
void *buffer; /* buffer 被定义为无值型指针 */
```