目　录

第 8 章 数 组 / 113

第 9 章 函 数 / 133

第 10 章 指 针 / 155

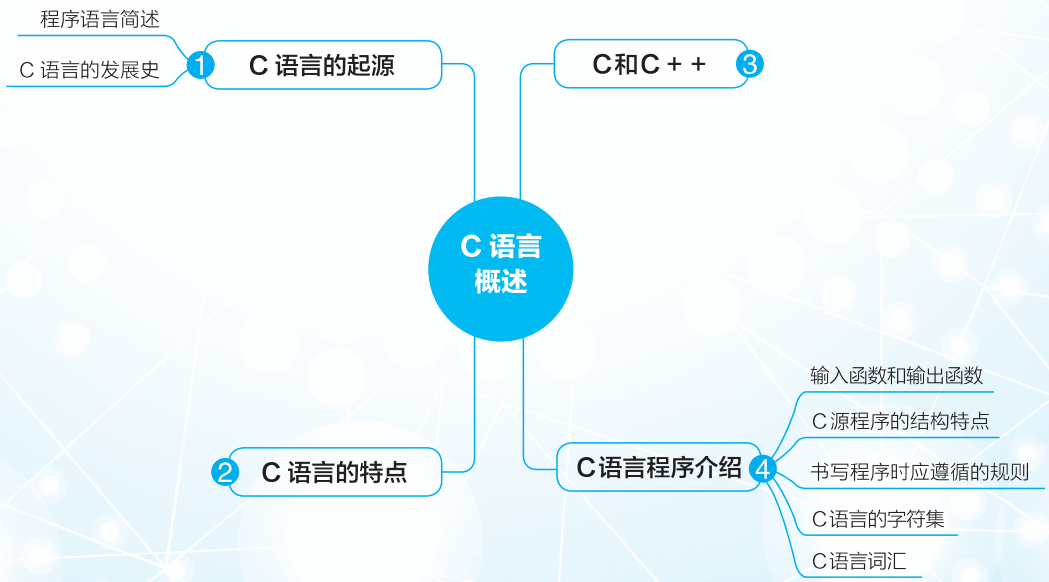
第 11 章 结构体与共用体 / 181

第 1 章 C 语言概述

学习目标

1. 了解 C 语言的发展史。
2. 熟悉 C 语言的基本函数。
3. 熟悉 C 语言的特点。

知识导图



笔记

本章导读

C 语言是一种面向过程、抽象化的通用程序设计语言，广泛应用于底层开发，它能以简易的方式编译、处理低级存储器，是一种仅产生少量机器语言及不需要任何运行环境支持便能运行的高效率程序设计语言。C 语言也是一种编程方式灵活多样、功能强大、应用广泛的程序设计语言。从程序设计语言的发展历程看，尽管后来出现了以 C++、Java、Python 和 C# 等为代表的新语言，但 C 语言的基础地位仍不可撼动。尽管 C 语言提供了许多低级处理的功能，但仍然保持着跨平台的特性，以一个标准规格写出的 C 语言程序可在包括一些类似嵌入式处理器以及超级计算机等作业平台的许多计算机平台上进行编译。

本章旨在使学生了解 C 语言的特点、程序编写和运行流程，掌握常用的开发环境，并通过 Visual C 6.0 编写简单的应用程序。

1.1 C 语言的起源

1.1.1 程序语言简述

在介绍 C 语言的发展历程之前，先对程序语言有一个大概的了解。

1. 机器语言

机器语言是低级语言，也称为二进制代码语言。计算机使用由 0 和 1 组成的二进制数组成的一串指令表达计算机操作。机器语言的特点是计算机可以直接识别，不需要进行任何的翻译。

2. 汇编语言

汇编语言是面向机器的程序设计语言。为了减小使用机器语言编程的麻烦，用英文字母或符号串替代机器语言的二进制码，这样就把不易理解和使用的机器语言变成了汇编语言。因此，汇编语言要比机器语言便于阅读和理解。

3. 高级语言

由于汇编语言依赖于硬件体系，并且该语言中的助记符号数量比较多，所以其运用起来仍然不够方便。为了使程序语言能更贴近人们的自然语言，同时又不依赖于计算机硬件，于是产生了高级语言。这种语言的语法形式类似于英文，并且由于不需要对硬件进行直接操作，因此易于被普通人所理解与使用。其中影响较大、使用普遍的高级语言有 C、C++、VC、VB、Java 等。

1.1.2 C 语言的发展史

从程序语言的发展过程可以看到，以前的操作系统等系统软件主要是用汇编语言编写的。但是，由于汇编语言依赖于计算机硬件，所以程序的可读性和可移植性都不是很好，为了提高可读性和可移植性，人们开始寻找一种语言，这种语言应该既具有高级语言的特性，又不失低级语言的优点，于是 C 语言产生了。

1967 年，剑桥大学的 Martin Richards 对 CPL（Combined Programming Language）进行了简化，于是产生了 BCPL（Basic Combined Programming Language）。

笔记

1970 年，美国贝尔实验室的 Ken Thompson 以 BCPL 为基础设计出很简单且很接近硬件的 B 语言（取 BCPL 的首字母），并且他用 B 语言写了第一个 UNIX 操作系统。

1972 年，美国贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出一种新的语言，他取 BCPL 的第二个字母作为这种语言的名字，这就是 C 语言。

最初，C 语言运行于 AT&T 的多用户、多任务的 UNIX 操作系统上。后来，丹尼斯·里奇用 C 语言改写了 UNIX C 的编译程序，UNIX 操作系统的开发者肯·汤普逊又用 C 语言成功地改写了 UNIX，从此开创了编程史上的新篇章。UNIX 成为第一个不是用汇编语言编写的主流操作系统。

1973 年年初，C 语言的主体完成。Thompson 和 Ritchie 迫不及待地开始用它完全重写了 UNIX。随着 UNIX 的发展，C 语言自身也在不断完善。直到今天，各种版本的 UNIX 内核和周边工具仍然使用 C 语言作为最主要的开发语言，其中还有不少 Thompson 和 Ritchie 编写的代码。

1983 年，美国国家标准学会（ANSI）对 C 语言进行了标准化，于 1983 年颁布了第一个 C 语言草案（83ANSI C），后来于 1987 年又颁布了另一个 C 语言标准草案（87ANSI C），最新的 C 语言标准 C99 于 1999 年颁布，并在 2000 年 3 月被 ANSI 采用。但是，由于未得到主流编译器厂家的支持，C99 并未得到广泛使用。

1989 年，ANSI 发布了第一个完整的 C 语言标准——ANSI X3.159-1989，简称 C89，不过人们也习惯称其为“ANSI C”。C89 在 1990 年被国际标准组织（International Standard Organization，ISO）一字不改地采纳，ISO 官方给予的名称为 ISO/IEC 9899，所以 ISO/IEC 9899：1990 通常被简称为 C90。

1999 年，在进行了一些必要的修正和完善后，ISO 发布了新的 C 语言标准，命名为 ISO/IEC 9899：1999，简称 C99。

2011 年 12 月 8 日，ISO 又正式发布了新的标准，称为 ISO/IEC 9899：2011，简称为 C11。

在 C 语言基础上发展起来的有支持多种程序设计风格的 C++ 语言，网络上广泛使用的 Java、JavaScript 以及微软的 C# 语言等。也就是说，学好 C 语言之后，再学习其他语言就会比较轻松。

1.2 C 语言的特点

C 语言的特点如下：

（1）C 语言是一种结构化程序设计语言。程序的各个部分除了必要的信息交流外，彼此独立，这使得 C 语言程序层次清晰，便于使用，易于调试和维护。

（2）C 语言是一门中级语言。可以使用 C 语言直接访问内存的物理地址，进行位（bit）操作，实现对硬件的编程操作。因此，C 语言集合了高级语言和低级语言的功能，既可用于系统软件的开发，也适合于应用软件的开发。

（3）语言简洁、紧凑，使用灵活、方便。程序书写形式自由，主要用小写字母表示，代码输入方便。

（4）强大的数据处理能力。ANSI C 一共有 32 个关键字、9 种控制语句和 34 种运算符，使得 C 语言的运算类型极为丰富，表达式类型多样化，可以实现很多其他高级语言

笔记

难以实现的运算。

（5）数据结构丰富。具有整型、实型、字符型、枚举类型、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的运算。

（6）部分变量类型可以转换，例如整型和字符型变量。

（7）生成目标代码质量高，程序执行效率高。一般比用汇编语言编写的程序生成的目标代码效率低 10%~20%。

（8）移植性好。用 C 语言编写的程序，基本上不做修改就可用于各种型号的计算机和操作系统。

1.3 C和C++

在 C 的基础上，1983 年贝尔实验室的 Bjarne Strou-strup 又推出了 C++。C++ 进一步扩充和完善了 C 语言，成为一种面向对象的程序设计语言。C++ 目前流行的最新版本是 Borland C++、Symantec C++ 和 Microsoft Visual C++。

C++ 提出一些更深入的概念，它所支持的这些面向对象的概念容易将问题空间直接映射到程序空间，为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法，因而也增加了整个语言的复杂性，掌握起来有一定难度。

但是，C 是 C++ 的基础，C++ 语言和 C 语言在很多方面是兼容的。因此，掌握了 C 语言，再进一步学习 C++ 就能以一种熟悉的语法学习面向对象的语言，从而达到事半功倍的目的。

1.4 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。虽然有关内容还未介绍，但可从这些例子中了解到组成一个 C 源程序的基本部分和书写格式。

【例 1.1】C 源程序的基本部分。

```
main()
{
   printf(" 世界，您好！ \n");
}
```

main 是主函数的函数名，表示这是一个主函数，每个 C 源程序都必须有，且只能有一个主函数（main( ) 函数）。printf( ) 函数的功能是把要输出的内容送到显示器上显示。printf( ) 函数是一个由系统定义的标准函数，可在程序中直接调用。

【例 1.2】完整的 C 源程序。

```
#include<math.h>
#include<stdio.h>
main()
{
```

```
    double x,s; // 定义两个实数变量，以被后面程序使用
    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("sine of %lf is %lf\n",x,s); // 显示程序运算结果
} //main( ) 函数结束
```

笔记

include 称为文件包含命令，扩展名为 .h 的文件称为头文件。

例 1.2 所示程序的功能是从键盘输入一个数 x[①]，求 x 的正弦值，然后输出结果。在 main( ) 之前的两行称为预处理命令（详见后面）。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中，使其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，在程序中调用一个库函数时，必须包含该函数原型所在的头文件。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，因此程序的主函数前用 include 命令包含了 math.h。由于 scanf( ) 和 printf( ) 是标准输入输出函数，其头文件为 stdio.h，因此在主函数前用 include 命令包含了 stdio.h 文件。

需要说明的是，C 语言规定对 scanf( ) 和 printf( ) 这两个函数可以省去头文件的包含命令。所以，本例也可以删去包含命令 #include<stdio.h>。

同样，例 1.1 中使用了 printf( ) 函数，也省略了包含命令。

例题中的主函数体分为两部分：一部分为说明部分；另一部分为执行部分。说明是指变量的类型说明。例 1.1 中未使用任何变量，因此无说明部分。C 语言规定，源程序中所有用到的变量都必须先说明，后使用，否则将会出错。这一点是编译型高级程序设计语言的一个特点，与解释型的 BASIC 语言是不同的。说明部分是 C 源程序结构中很重要的组成部分。本例中使用两个变量 x、s，表示输入的自变量和 sin( ) 函数值。由于 sin( ) 函数要求这两个量必须是双精度浮点型，故用类型说明符 double 说明这两个变量。说明部分后的四行为执行部分，或称为执行语句部分，用以完成程序的功能。执行部分的第一行是输出语句，调用 printf( ) 函数在显示器上输出提示字符串，请操作人员输入自变量 x 的值。第二行为输入语句，调用 scanf( ) 函数，接受从键盘上输入的数并存入变量 x 中。第三行是调用 sin( ) 函数并把函数值送到变量 s 中。第四行是用 printf( ) 函数输出变量 s 的值，即 x 的正弦值。程序结束。

运行本程序时，首先在显示器屏幕上给出提示串 input number，这是由执行部分的第一行完成的。用户在提示下从键盘上输入某一数，如 5，按下回车键，接着在屏幕上给出计算结果。

1.4.1 输入函数和输出函数

前两个例子都用到了输入函数 scanf( ) 和输出函数 printf( )，后面会详细介绍，这里先

① 为了阅读方便，文中叙述的变量 x、y、s、i 等为与程序中的保持一致，也用正体表示。

笔记

简单介绍一下它们的格式，以便使用。

scanf( ) 和 printf( ) 分别称为格式输入函数和格式输出函数，其意义是按指定的格式输入值和输出值。因此，这两个函数在括号中的参数表都由以下两部分组成：

```
" 格式控制串 "，参数表
```

格式控制串是一个字符串，必须用双引号括起来，它表示了输入输出量的数据类型。各种类型的格式表示法可参阅第 4 章。在 printf( ) 函数中，还可以在格式控制串内出现非格式控制字符，这时在显示屏幕上将原文输出。参数表中给出了输入或输出的量。当有多个量时，用逗号间隔。例如：

```
printf("sine of %lf is %lf\n",x,s);
```

其中 %lf 为格式字符，表示按双精度浮点数处理。它在格式串中出现了两次，对应 x 和 s 两个变量。其余字符为非格式字符，在屏幕上原样输出。

【例 1.3】格式控制串。

```
int max(int a,int b);                /* 函数说明 */
main()                               /* 主函数 */
{
   int x,y,z;                        /* 变量说明 */
   int max(int a,int b);             /* 函数说明 */
   printf("input two numbers:\n");
   scanf("%d%d",&x,&y);              /* 输入 x、y 值 */
   z=max(x,y);                       /* 调用 max( ) 函数 */
   printf("maxmum=%d",z);            /* 输出 */
}
int max(int a,int b)                 /* 定义 max( ) 函数 */
{
   if(a>b)return a;else return b;    /* 把结果返回主调函数 */
}
```

该程序的功能是：由用户输入两个整数，程序执行后输出其中较大的数。本程序由两个函数组成：主函数 main( ) 和 max( ) 函数。函数之间是并列关系。可从主函数中调用其他函数。max( ) 函数的功能是比较两个数，然后把较大的数返回给主函数。max( ) 函数是一个用户自定义函数，因此在主函数中要给出说明（程序第三行）。可见，在程序的说明部分中，不仅可以有变量说明，还可以有函数说明。关于函数的内容将在第 5 章介绍。在程序的每行后用 /* 和 */ 括起来的内容为注释部分，程序不执行注释部分。

例 1.3 所示程序的执行过程是：首先在屏幕上显示提示串，请用户输入两个数，按回车键后由 scanf( ) 函数接收这两个数送入变量 x、y 中，然后调用 max( ) 函数，并把 x、y 的值传送给 max( ) 函数的参数 a、b。在 max( ) 函数中比较 a、b 的大小，把大者返回给主函数的变量 z，最后在屏幕上输出 z 的值。

1.4.2 C源程序的结构特点

笔记

C 源程序的结构特点如下：

（1）一个 C 语言源程序可以由一个或多个源文件组成。

（2）每个源文件可由一个或多个函数组成。

（3）一个源程序不论由多少个文件组成，都有一个且只能有一个 main( ) 函数，即主函数。

（4）源程序中可以有预处理命令（include 命令仅为其中一种）。预处理命令通常放在源文件或源程序的最前面。

（5）每个说明、每条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。

（6）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。

1.4.3 书写程序时应遵循的规则

从书写清晰，便于阅读、理解、维护的角度出发，在书写程序时应遵循以下规则：

（1）一个说明或一个语句占一行。

（2）用 {} 括起来的部分通常表示了程序的某一层次结构。{} 一般与该结构语句的第一个字母对齐，并单独占一行。

（3）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。

1.4.4 C语言的字符集

字符是组成语言的最基本的元素。C 语言字符集由字母、数字、空格、标点和特殊字符组成。在字符常量、字符串常量和注释中还可以使用汉字或其他可表示的图形符号。

（1）字母：小写字母 a~z 共 26 个，大写字母 A~Z 共 26 个。

（2）数字：0~9 共 10 个。

（3）空白符：空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用，在其他地方出现时，只起间隔作用，编译程序对它们忽略不计。因此，在程序中使用空白符与否，对程序的编译不产生影响，但在程序中适当的地方使用空白符将增加程序的清晰性和可读性。

（4）标点和特殊字符。

1.4.5 C语言词汇

在 C 语言中使用的词汇分为六类：标识符、关键字、运算符、分隔符、常量、注释符。

1. 标识符

在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外，其余都由用户自定义。C 规定，标识符只能是字母（A~Z，a~z）、数字（0~9）、下画线（_）组成的字符串，并且其第一个字符必须是字母或下画线。

笔记

以下标识符是合法的：

a，x，x3，BOOK_1，sum5。

以下标识符是非法的：

3s　以数字开头

s*T　出现非法字符 *

-3x　以减号开头

bowy-1　出现非法字符 -（减号）

在使用标识符时还必须注意以下几点：

（1）标准 C 不限制标识符的长度，但它受各种版本的 C 语言编译系统限制，同时也受到具体机器的限制。例如，在某版本 C 中规定标识符前八位有效，当两个标识符前八位相同时，则被认为是同一个标识符。

（2）在标识符中，大小写是有区别的。例如，BOOK 和 book 是两个不同的标识符。

（3）标识符虽然可由程序员随意定义，但标识符是用于标识某个量的符号。因此，命名应尽量有相应的意义，以便于阅读、理解，做到“顾名思义”。

2. 关键字

关键字是由 C 语言规定的具有特定意义的字符串，通常也称为保留字。用户定义的标识符不应与关键字相同。C 语言的关键字分为以下三类。

（1）类型说明符：用于定义和说明变量、函数或其他数据结构的类型，如前面例题中用到的 int、double 等

（2）语句定义符：用于表示一个语句的功能。如例 1.3 中用到的 if…else 就是条件语句的语句定义符。

（3）预处理命令字：用于表示一个预处理命令，如前面各例中用到的 include。

3. 运算符

C 语言中含有相当丰富的运算符。运算符与变量、函数一起组成表达式，表示各种运算功能。运算符由一个或多个字符组成。

4. 分隔符

在 C 语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中，用于分隔各个变量。空格多用于语句各单词之间，做间隔符。在关键字与标识符之间必须有一个以上的空格符作为间隔，否则将会出现语法错误，例如，把 int a ; 写成 inta ; C 编译器会把 inta 当成一个标识符处理，其结果必然出错。

5. 常量

C 语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种。后面章节会专门介绍。

6. 注释符

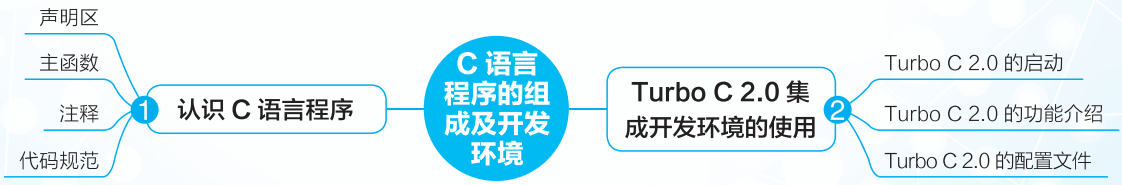
C 语言的注释符是以“/*”开头并以“*/”结尾的串。在“/*”和“*/”之间的内容为注释。程序编译时，不对注释做任何处理。注释可出现在程序中的任何位置。注释用来向用户提示或解释程序的意义。在调试程序中，暂不使用的语句也可用注释符括起来，使编译时跳过，待调试结束后再去掉注释符。

第 2 章 C 语言程序的组成及开发环境

学习目标

1. 认识 C 语言程序。
2. 熟悉 C 语言代码规范。
3. 熟悉 C 语言开发环境。

知识导图



笔记

本章导读

本章主要学习 C 语言程序的基本构成。一个完整的 C 语言程序，无论多么简单或者多么复杂，都是由一些基本的内容构成的，如头文件、变量声明、主函数、注释等。在学习 C 语言程序基本构成的基础上，了解和掌握代码编写的一些基本规范。

2.1 认识 C 语言程序

在第 1 章中，我们认识了简单的 C 语言程序。接下来通过实例进一步认识 C 语言程序的一些基本组成部分。例 2.1 所示的程序包含头文件、变量声明、主函数、注释等部分。

【例 2.1】定义 C 语言程序。

```
#include“stdio.h”
int a,b,sum=0;// 定义 a、b、sum 三个变量
main()
{
   a=10;// 给变量 a 赋初值
   b=20;// 给变量 b 赋初值
   sum=a+b; // 将 a 和 b 的值相加，再赋给 sum
   printf(“a+b=%d\n”,sum);// 输出结果
}
```

2.1.1 声明区

在 C 语言中，声明区主要包含头文件、函数声明、变量声明等内容。函数的声明和使用将在第 9 章中介绍，这里主要介绍头文件和变量声明。

1. 头文件

在程序的第一行有一个 #include，叫作“文件包含”。include 语句包含并运行指定文件。其后面要包含的文件要么使用 < >，要么使用 " " 标识。< > 表示包含的是编译器的类库路径里面的头文件；" " 表示包含的是当前程序目录的相对路径中的头文件。如果使用 " "，它会先在你项目的当前目录查找是否有对应的头文件，如果没有，它还会在编译器的类库路径里面查找对应的头文件。

#include 后面包含了“ stdio.h”文件。这里的 stdio.h 就叫作头文件。常见的头文件有两个：stdio.h 和 stdafx.h。其中，stdio（Standard Input & Output），即标准输入输出。stdio.h 是 C 语言标准库文件的头文件，包含基本的输入输出语句，以及文件操作语句等。Stdafx（Standard Application Framework Extensions），即标准应用程序框架的扩展，是 MFC 的编译向导自动生成的。stdafx.h 中包含 stdio.h，且 stdafx.h 仅适用于支持 MFC 的平台。MFC（Microsoft Foundation Classes），即微软基础类库，是微软公司提供的一个类库（class libraries），以 C++ 类的形式封装了 Windows API，并且包含一个应用程序框架，以减少应用程序开发人员的工作量。其中包含大量的 Windows 句柄封装类和很多 Windows 的内建控件和组件的封装类。MFC 除了是一个类库以外，还是一个框架。在 VC++ 里新建一个 MFC 的工程，开发环境会自动产生许多文件。

笔记

所以，只要是文件后缀名为 .h 的都叫作头文件。在 C 语言家族程序中，头文件被大量使用。一般而言，每个 C 程序通常由头文件（header files）和定义文件（definition files）组成。头文件作为一种包含功能函数、数据接口声明的载体文件，主要用于保存程序的声明（declaration），而定义文件，也就是你自己编写的程序，主要用于保存程序的实现（implementation）。

2. 变量声明

C 语言是一种强类型的编程语言。这就意味着凡是在程序中要使用到的变量，都需要事先声明（定义），即“先定义，后使用。”一次可以定义一个变量，也可以定义多个变量，多个变量之间通过逗号分隔。可以在定义变量的同时就给变量赋一个初始值，也可以在定义之后再对变量进行赋值。给变量赋初始值，也叫作对变量进行初始化。关于变量的类型及其他相关知识，后续章节会讲到。

2.1.2 主函数

每个 C 语言程序都有一个主函数，也叫作入口函数，是程序执行的起点。无论一个程序多么复杂，总是从主函数的位置开始执行。一个程序有且只有一个主函数。主函数的一般形式为 main( )。

在 C 语言中，一个程序无论是复杂或是简单，总体上都是一个“函数”；这个函数就称为“ main( ) 函数”，也就是“主函数”。main( ) 函数在程序中大多数是必须存在的，但是依然有例外情况，比如 Windows 编程中可以编写一个动态链接库（Dynamic Link Library，DLL）模块，简称 dll，这是其他 Windows 程序可以使用的代码。由于 DLL 模块不是独立的程序，因此不需要 main( ) 函数。再如，用于专业环境的程序——如机器人中的控制芯片——可能不需要 main( ) 函数。

除了主函数之外，在上面的程序中还有一个函数 printf( )。printf( ) 函数是格式化输出函数，一般用于向标准输出设备（显示器、控制台等）按规定格式输出信息。要输出的文字除了可以使用字母、数字、空格和一些数字符号外，还可以使用一些转义字符表示特殊的含义。实例中的 \n 为换行符。

2.1.3 注释

除了声明区和主函数之外，上面的程序中还有一些“ //”及其后面出现的一个中文描述，这部分叫作注释。任何一门程序设计语言都有注释。注释的内容主要是便于代码编写人员对程序的某些部分进行说明，只是给人看的，对编译和运行不起作用，即“供人阅读”而非“供机器编译”。所以，编译器遇到注释符号的时候会自动跳过，不会编译其后的内容。

常见的注释符号有 // 单行注释符和 /*…*/ 多行注释符。注释部分可以使用汉字，也可以使用英语或者汉语拼音等。

2.1.4 代码规范

要写好 C 语言程序，掌握一些程序书写规则非常必要。一般来说，C 语言的代码编写有如下一些注意事项和约定俗成的规则。

笔记

（1）一个 C 语言源程序可以由一个或多个函数组成。

（2）每个源程序可由一个或多个函数组成。

（3）一个源程序无论由多少个文件组成，都有且只有一个 main( ) 函数。

（4）源程序中可以有预处理命令（include 命令仅为其中一种，相关知识将在第 12 章中讲到），预处理命令通常应放在源程序的最前面。

（5）每个语句都必须以分号结尾，但预处理命令、函数头和大括号之后不能加分号。

（6）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，例如小括号，也可不添加空格。

（7）为了使程序结构更为清晰，尽量一个语句占一行。

（8）用 {} 括起来的部分，通常表示程序的某一层次结构。{} 一般与该结构的第一个字母对齐，并单独占一行。

（9）低一层次的语句可比高一层次的语句缩进若干空格后书写，以便程序结构更加清晰，增强程序的可读性。书写时，可以使用空格，也可以按“Tab”键控制缩进。

2.2 Turbo C 2.0 集成开发环境的使用

2.2.1 Turbo C 2.0 的启动

我们上机实习和将来考试使用的都是 Borland Turbo C 2.0 这个版本。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。

我们机房是在 D 盘根目录下建立一个 TC 子目录下安装 Turbo C 2.0 系统的。TC 下还建立了两个子目录 LIB 和 INCLUDE，LIB 子目录中存放库文件，INCLUDE 子目录中存放所有头文件。

在 DOS 环境下或在 Windows 98 的 DOS 窗口下运行 Turbo C 2.0 时，只要在 TC 子目录下输入 TC 并按回车键即可进入 Turbo C 2. 0 集成开发环境。

在 Windows 98 环境下，可以选择运行菜单，然后输入 d：\tc\tc，也可以在 tc 文件夹中找到 tc.exe 文件，然后双击该文件名进入 Turbo C 2. 0 集成开发环境。

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。该公司相继推出一套 Turbo 系列软件，如 Turbo Basic、TurboPascal、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（数据、代码、堆栈处在同一 64KB 内存中）文件。还可对数学协处理器（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++，它继承发展 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 Borlandc C++ 也问世了。

2.2.2 Turbo C 2.0 的功能介绍

笔记

进入 Turbo C 2.0 集成开发环境后，屏幕显示如图 2-1 所示。

图 2-1　Turbo C 2.0 集成开发环境

其中顶行为 Turbo C 2.0 主菜单，中间窗口为编辑区，接下来是信息窗口，底行为参考行。这四部分构成了 Turbo C 2.0 的主屏幕，以后的编程、编译、调试以及运行都在这个主屏幕中进行。

主菜单在 Turbo C 2.0 主屏幕顶上一行，显示了下列内容：

File Edit Run Compile Project Options Debug Break/watch

除 Edit 外，其他各项均有子菜单，只要按“Alt”键加上某项中的第一个字母，就可进入该项的子菜单中。

1. File 菜单

按组合键“Alt+F”可进入 File 菜单，如图 2-2 所示。

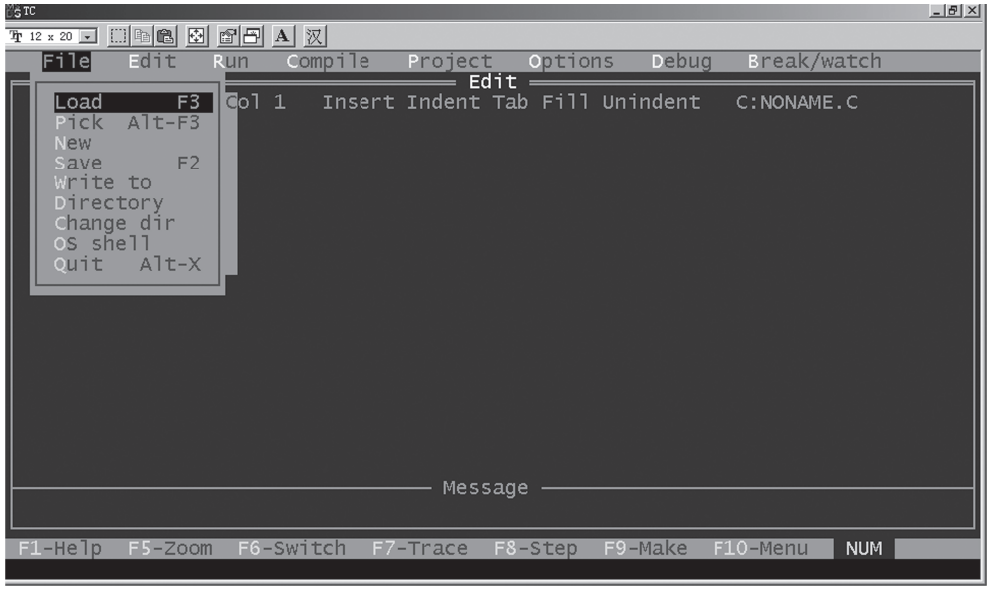


图 2-2　File 菜单

笔记

File 菜单的子菜单共有 9 项，分别如下。

（1）Load：装入一个文件，可用类似 DOS 的通配符（如 *.C）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 键即可进入该项，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名，其热键为“F2”。

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

（6）Directory：显示目录及目录中的文件，并可由用户选择。

（7）Change dir：显示当前默认目录，用户可以改变默认目录。

（8）Os shell：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 状态下输入 EXIT 即可。

（9）Quit：退出 Turbo C 2.0，返回到 DOS 操作系统中，其热键为“Alt+X”。

说明

以上各项可用光标键移动色棒进行选择，按回车键则执行。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，均可按“Esc”键，Turbo C 2.0 的所有菜单均采用这种方法进行操作，以下不再说明。

2. Edit 菜单

按组合键“Alt+E”可进入编辑菜单，若再按回车键，则光标出现在编辑窗口，此时用户可以进行文本编辑。编辑方法基本与文字处理器软件 WordStar 相同，可通过“F1”键获得有关编辑方法的帮助信息。

（1）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

（2）编辑命令如下：

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	块复制
Ctrl+KY	块删除
Ctrl+KR	读文件
Ctrl+KW	存文件
Ctrl+KP	块文件打印

Ctrl+F1　　如果光标所在处为 Turbo C 2.0 库函数，则获得有关该函数的帮助信息

Ctrl+Q[　　查找 Turbo C 2.0 双界符的后匹配符

Ctrl+Q]　　查找 Turbo C 2.0 双界符的前匹配符

笔记

知识链接

Turbo C 2.0 的双界符包括以下几种符号：

（1）花括符　{ 和 }

（2）尖括符　< 和 >

（3）圆括符　（和）

（4）方括符　[和]

（5）注释符　/* 和 */

（6）双引号　"

（7）单引号　'

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 菜单

按组合键 Alt+R 可进入 Run 菜单，该菜单如图 2-3 所示。

图 2-3　Run 菜单

（1）Run：运行由 Project/Project name 项指定的文件名或当前编辑区的文件。如果对上次编译后的源代码未做过修改，则直接运行到下一个断点（若没有断点，则运行到结束），否则先进行编译、连接，之后才运行，其热键为“Ctrl+F9”。

（2）Program reset：中止当前的调试，释放分给程序的空间，其热键为“Ctrl+F2”。

（3）Go to cursor：：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，其热键为“F4”。

（4）Trace into：在执行一条调用其他用户定义的子函数时，若用 Trace into 项，则执

笔记

行长条将跟踪到该子函数内部去执行，其热键为“F7”。

（5）Step over：执行当前函数的下一条语句，即使用户函数调用，执行长条也不会跟踪进函数内部，其热键为“F8”。

（6）User screen：显示程序运行时在屏幕上显示的结果，其热键为“Alt+F5”。

4. Compile 菜单

按组合键“Alt+C”可进入 Compile 菜单，该菜单如图 2-4 所示。

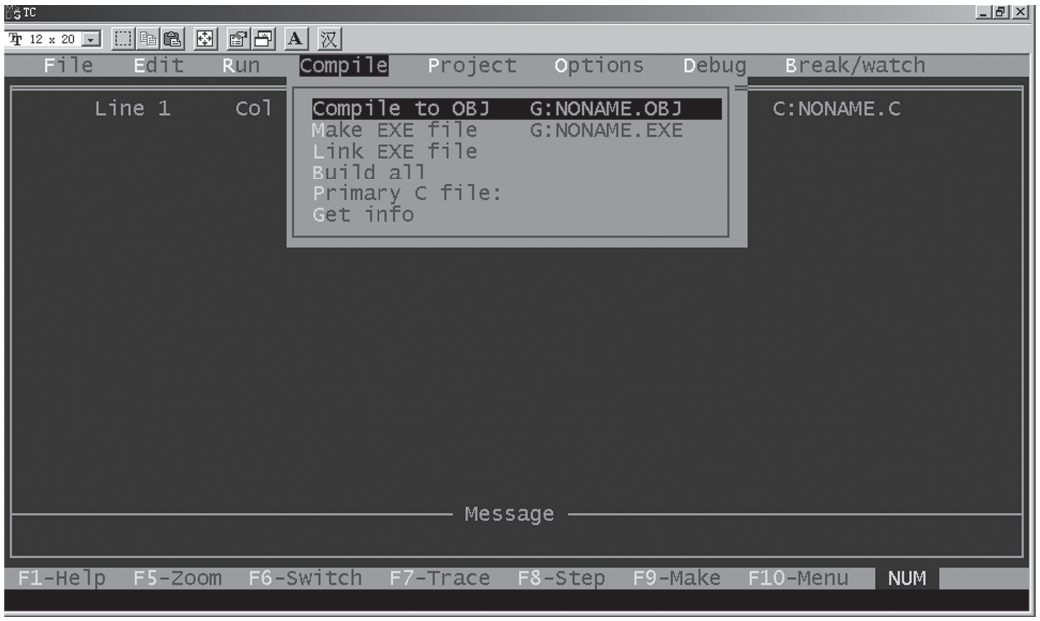


图 2-4　Compile 菜单

（1）Compile to OBJ：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，其热键为“Alt+F9”。

（2）Make EXE file：此命令生成一个 .exe 的文件，并显示生成的 .exe 文件名。其中 .exe 文件名是下面几项之一：

①由 Project/Project name 说明的项目文件名。

②若没有项目文件名，则是由 Primary C file 说明的源文件。

③若以上两项都没有文件名，则为当前窗口的文件名。

（3）Link EXE file：把当前 .obj 文件及库文件连接在一起生成 .exe 文件。

（4）Build all：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，则拒绝对源文件进行编译）。

（5）Primary C file：当在该项中指定主文件后，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，不管目前窗口中是不是主 C 文件。

（6）Get info：获得有关当前路径、源文件名、源文件字节大小、编译中的错误数目、可用空间等信息，如图 2-5 所示。

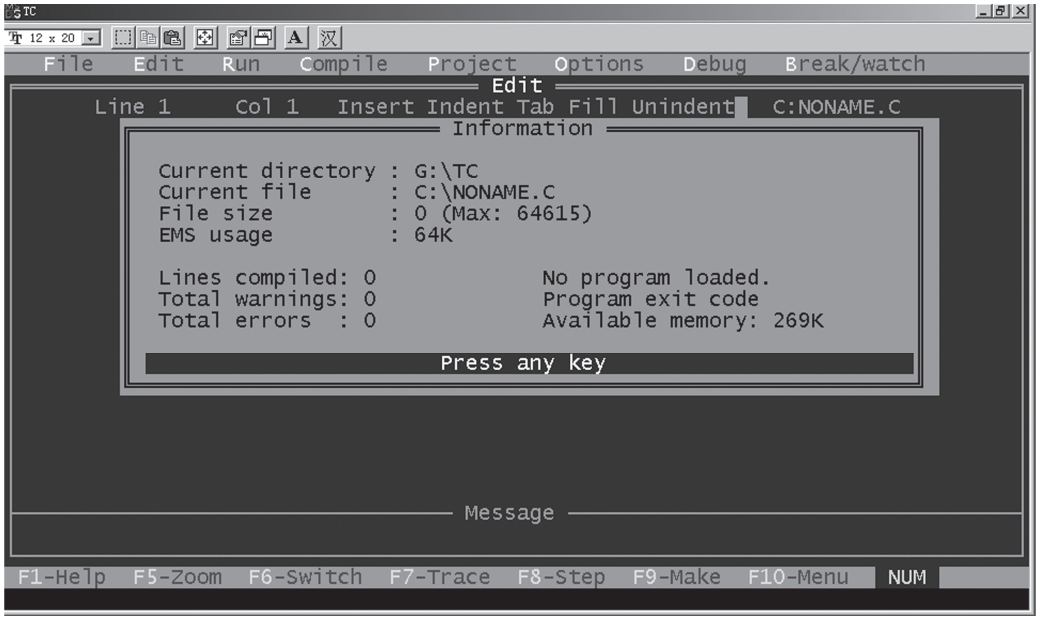


图 2-5　Get info

5. Project 菜单

按组合键“Alt+P”可进入 Project 菜单，该菜单如图 2-6 所示。

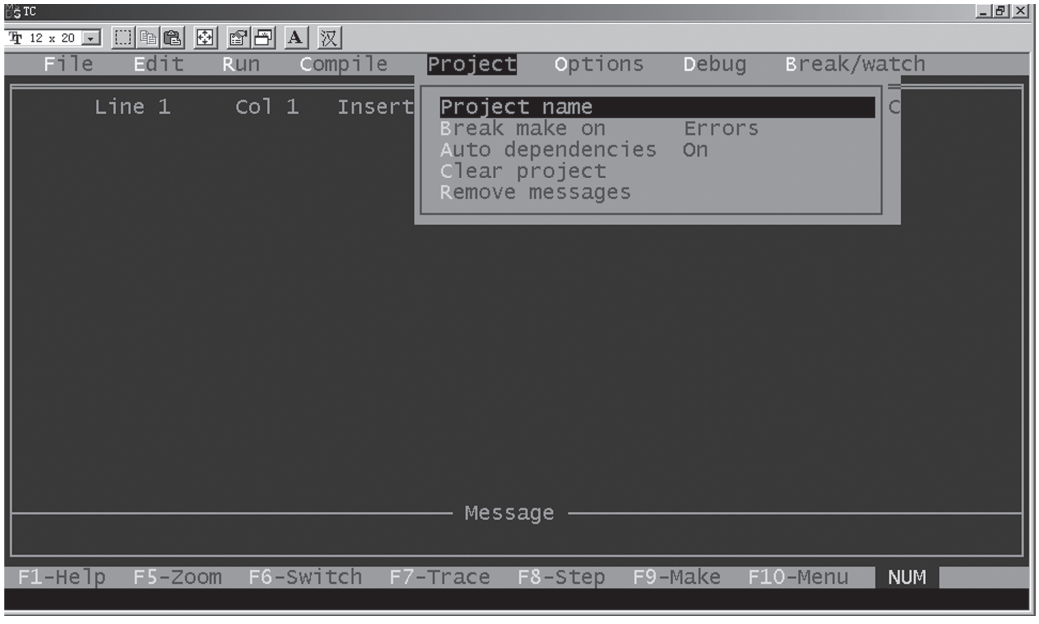


图 2-6　Project 菜单

（1）Project name：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，有一个程序由 file1.c、file2.c、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
```

笔记

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三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，则可直接写上 .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库文件，但必须写上扩展名 .lib。

（2）Break make on：由用户选择是否在有 Warining、Errors、Fatal Errors 时或在 Link 之前退出 Make 编译。

（3）Auto dependencies：当开关置为 On，编译时将检查源文件与对应的 .obj 文件日期和时间，否则不进行检查。

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

（5）Remove messages：把错误信息从信息窗口中清除。

6. Options 菜单

按组合键“Alt+O”可进入 Options 菜单，该菜单对初学者来说要谨慎使用。Options 菜单如图 2-7 所示。

图 2-7　Options 菜单

（1）Compiler：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。这些子菜单如图 2-8 所示。

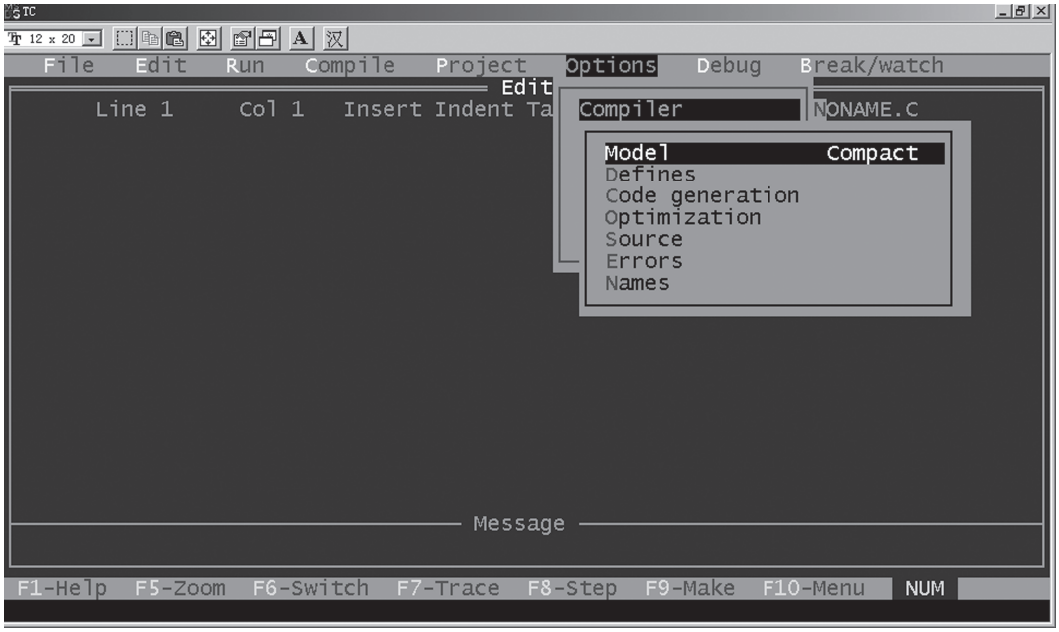


图 2-8　Compiler

① Model：有 Tiny、small、medium、compact、large、huge 六种模式可由用户选择。

② Defines ：打开一个宏定义框，用户可输入宏定义。多重定义可用分号，赋值可用等号。

③ Code generation ：它又有许多任选项，这些任选项告诉编译器产生什么样的目标代码。

- Calling convention　可选择 C 或 Pascal 方式传递参数。
- Instruction set　可选择 8088/8086 或 80186/80286 指令系列。
- Floating point　可选择仿真浮点、数学协处理器浮点或无浮点运算。
- Default char type　规定 char 的类型。
- Alignonent　规定地址对准原则。
- Merge duplicate strings　做优化用，将重复的字符串合并在一起。
- Standard stack frame　产生一个标准的栈结构。
- Test stack overflow　产生一段程序运行时检测堆栈溢出的代码。
- Line number　在 .obj 文件中放进行号以供调试时用。
- OBJ debug information　在 .obj 文件中产生调试信息。

④ Optimization：它又有许多任选项。

- Optimize for　选择是对程序小型化，还是对程序速度进行优化处理。
- Use register variable　用来选择是否允许使用寄存器变量。
- Register optimization　尽可能使用寄存器变量，以减少过多的取数操作。
- Jump optimization　通过去除多余的跳转和调整循环与开关语句的办法压缩代码。

⑤ Source：它又有许多任选项。

- Indentifier length　说明标识符有效字符的个数，默认为 32 个。
- Nested comments　是否允许嵌套注释。
- ANSI keywords only　是只允许 ANSI 关键字，还是也允许 Turbo C 2.0 关键字。

笔记

笔记

⑥ Errors：

- Error stop after　多少个错误时停止编译，默认为 25 个。
- Warning stop after　多少个警告错误时停止编译，默认为 100 个。
- Display warning　显示警告错误。
- Portability warning　移植性警告错误。
- ANSI Violations　侵犯了 ANSI 关键字的警告错误。
- Common error　常见的警告错误。
- Less common error　少见的警告错误。

⑦ Names：用于改变段（segment）、组（group）和类（class）的名字，默认值为 CODE、DATA、BSS。

（2）Linker：本菜单用于设置有关连接的选择项，它有以下内容（见图 2-9）：

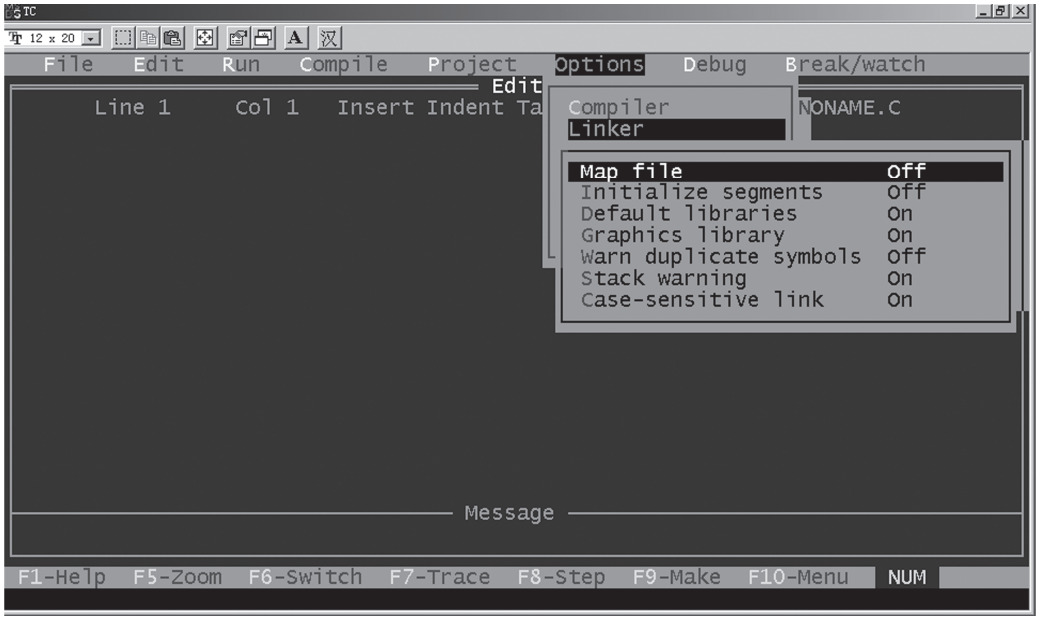


图 2-9　Linker

① Map file　选择是否产生 .map 文件。

② Initialize segments　是否在连接时初始化没有初始化的段。

③ Default libraries　是否在连接其他编译程序产生的目标文件时寻找其缺省库。

④ Graphics library　是否连接 graphics 库中的函数。

⑤ Warn duplicate symbols　当有重复符号时产生警告信息。

⑥ Stack warning　是否让连接程序产生 No stack 的警告信息。

⑦ Case-sensitive link　是否区分大小写。

（3）Environment：该菜单规定是否对某些文件自动存盘，以及进行制表键和屏幕大小的设置，它有以下内容，如图 2-10 所示。

图 2-10　Environment

① Message Tracking：

- Current file　跟踪在编辑窗口中的文件错误。
- All files　跟踪所有文件错误。
- Off　不跟踪。

② Keep messages：编译前是否清除 Message 窗口中的信息。

③ Config auto save ：选 On 时，在 Run、Shell 或退出集成开发环境之前，如果 Turbo C 2.0 的配置被改过，则所做的改动将存入配置文件中。选 Off 时不存。

④ Edit auto save：是否在 Run 或 Shell 之前自动存储编辑的源文件。

⑤ Backup files：是否在源文件存盘时产生后备文件（.bak 文件）。

⑥ Tab size：设置制表键大小，默认为 8。

⑦ Zoomed windows：将现行活动窗口放大到整个屏幕，其热键为“F5”。

⑧ Screen size：设置屏幕文本大小。

（4）Directories：规定编译、连接所需文件的路径，有下列各项，如图 2-11 所示。

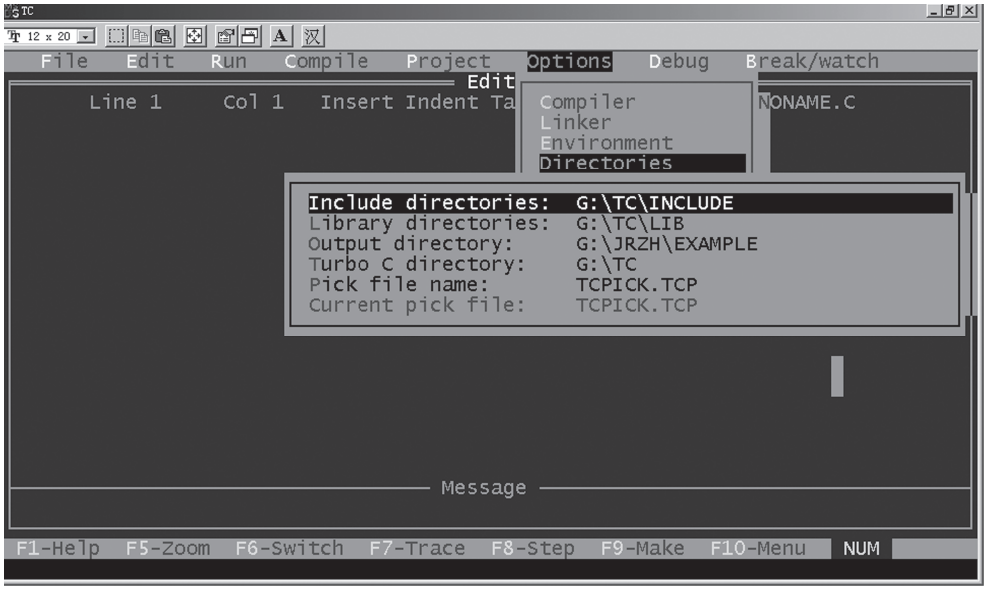


图 2-11　Directories

笔记

① Include directories：包含文件的路径，多个子目录用 "；" 分开。

② Library directories：库文件路径，多个子目录用 "；" 分开。

③ Output directory：输出文件（.obj，.exe，.map 文件）的目录。

④ Turbo C directory：Turbo C 所在的目录。

⑤ Pick file name：定义加载的 pick 文件名，如不定义，则从 currentpick file 中取。

⑥ Current pick file：当前 pick 文件。

（5）Arguments：允许用户使用命令行参数。

（6）Save options：保存所有选择的编译、连接、调试和项目到配置文件中，缺省的配置文件为 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 菜单

按组合键“Alt+D”可选择 Debug 菜单，该菜单主要用于查错，它包括以下内容，如图 2-12 所示。

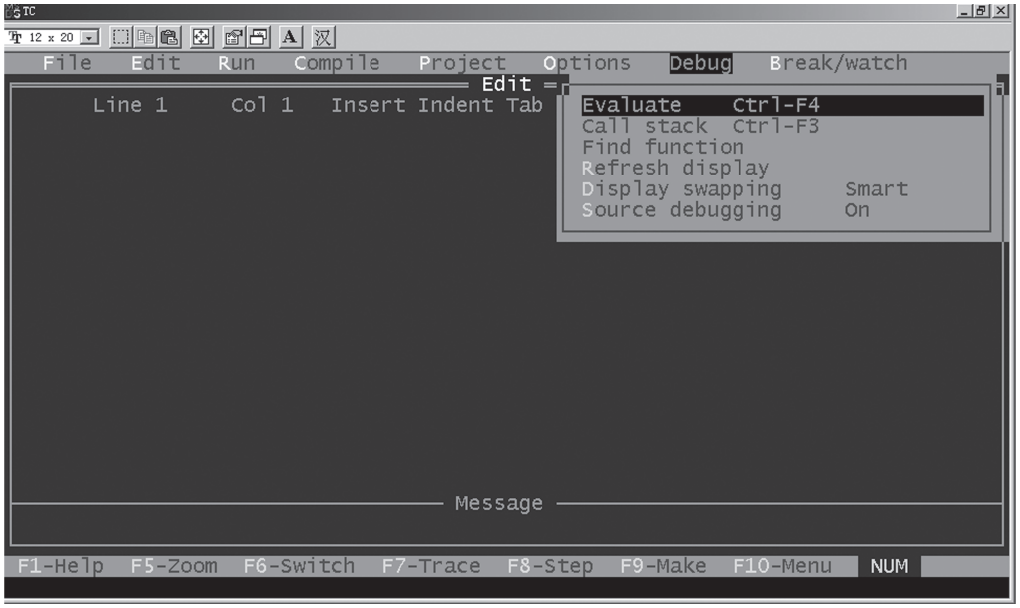


图 2-12 Debug 菜单

（1）Evaluate：

① Expression：要计算结果的表达式。

② Result：显示表达式的计算结果。

③ New value：赋给新值。

（2）Call stack：该项不可接触，而在 Turbo C debuger 时用于检查堆栈情况。

（3）Find function：在运行 Turbo C debugger 时用于显示规定的函数。

（4）Refresh display：如果编辑窗口偶然被用户窗口重写了，则可用此恢复编辑窗口的内容。

（5）Display swapping：用于显示屏幕切换方式。

（6）Source debugging：用于设置是否允许源代码调试。

8. Break/watch 菜单

按组合键“Alt+B”可进入 Break/watch 菜单，该菜单有以下内容，如图 2-13 所示。

图 2-13　Break/watch 菜单

（1）Add watch：向监视窗口插入一监视表达式。

（2）Delete watch：从监视窗口中删除当前的监视表达式。

（3）Edit watch：在监视窗口中编辑一个监视表达式。

（4）Remove all watches：从监视窗口中删除所有的监视表达式。

（5）Toggle breakpoint：对光标所在的行设置或清除断点。

（6）Clear all breakpoints：清除所有断点。

（7）View next breakpoint：将光标移动到下一个断点处。

2.2.3 Turbo C 2.0 的配置文件

所谓配置文件，是指包含 Turbo C 2.0 有关信息的文件，其中存有编译、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。

（1）建立用户自命名的配置文件：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 下输入：

```
tc/c< 用户命名的配置文件名 >
```

就会将这个配置文件中的内容作为 Turbo C 2.0 的选择。

（2）若设置 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 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 在启动时若没有找到配置文件，则取 TC.exe 中的默认值。

笔记