

C 语言教程

大家好，我是桂圆，C 语言程序设计教程，由我来跟大家一起学习，自我介绍就不过多的介绍了

当大家看完此课程后有什么疑惑，意见或建议，请大家到论坛上发表，关于技术的问题，我会为大家耐心的解答

1 C 语言概述

1.1 C 语言的发展过程

C 语言是在 70 年代初问世的。一九七八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。

当代最优秀的程序设计语言

早期的 C 语言主要是用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们认识，到了八十年代，C 开始进入其它操作系统，并很快在各类大、中、小和微型计算机上得到了广泛的使用，成为当代最优秀的程序设计语言之一。

1.2 C 语言版本

目前最流行的 C 语言有以下几种：

- Microsoft C 或称 MS C
- Borland Turbo C 或称 Turbo C

这些 C 语言版本不仅实现了 ANSI C 标准，而且在此基础上各自作了一些扩充，使之更加方便、完美。

1.1 C 语言的特点

1 • C 语言简洁、紧凑，使用方便、灵活。ANSI C 一共只有 32 个关键字：

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

9 种控制语句，程序书写自由，主要用小写字母表示，压缩了一切不必要的成分。

注意：在 C 语言中，关键字都是小写的。

2• 运算符丰富。共有 34 种。C 把括号、赋值、逗号等都作为运算符处理。

从而使 C 的运算类型极为丰富，可以实现其他高级语言难以实现的运算。

3• 数据结构类型丰富。

4• 具有结构化的控制语句。

5• 语法限制不太严格，程序设计自由度大。

6• C 语言允许直接访问物理地址，能进行位（bit）操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。因此有人把它称为中级语言。

7• 生成目标代码质量高，程序执行效率高。

8• 与汇编语言相比，用 C 语言写的程序可移植性好。

但是，C 语言对程序员要求也高，程序员用 C 写程序会感到限制少、灵活性大，功能强，但较其他高级语言在学习上要困难一些。

1.2 面向对象的程序设计语言

在 C 的基础上，一九八三年又由贝尔实验室推出了 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++ 和 Microsoft VisualC++。

C++ 提出了一些更为深入的概念，它所支持的这些面向对象的概念容易将问题空间直接地映射到程序空间，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。

1.3 C 和 C++

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来学习面向对象的语言，从而达到事半功倍的目的。

1.4 简单的 C 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】

```
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。
- 每一个 C 源程序都必须有，且只能有一个主函数(main 函数)。
- 函数调用语句，printf 函数的功能是把要输出的内容送到显示器去显示。

- `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】

```
#include<math.h>
#include<stdio.h>
main()
{
    double x,s;
    printf("input number:\n");
    scanf("%lf",&x);
    s=sin(x);
    printf("sine of %lf is %lf\n",x,s);
}
```

- `include` 称为文件包含命令
- 扩展名为 `.h` 的文件称为头文件
- 定义两个实数变量，以被后面程序使用
- 显示提示信息
- 从键盘获得一个实数 `x`
- 求 `x` 的正弦,并把它赋给变量 `s`
- 显示程序运算结果
- `main` 函数结束

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。

在 `main()` 之前的两行称为预处理命令(详见后面)。

预处理命令还有其它几种，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。

需要说明的是，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>`。

同样，在例 1.1 中使用了 `printf` 函数，也省略了包含命令。

在例题中的主函数体中又分为两部分，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。例题 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。本例中使用了两个变量 `x`，`s`，用来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` 来说明这两个变量。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键入某一数，如 `5`，按下回车键，接着在屏幕上给出计算结果。

1.5 输入和输出函数

在前两个例子中用到了输入和输出函数 `scanf` 和 `printf`，在以后要详细介绍。这里我们先简单介绍一下它们的格式，以便下面使用。

`scanf` 和 `printf` 这两个函数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三章。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照印。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。它在格式串中两次出现，对应了 `x` 和 `s` 两个变量。其余字符为非格式字符则照原样输出在屏幕上。

【例 1.3】

```
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*/
}
```

上面例中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，主函数和 `max` 函数。函数之间是并列关系。可从主函数中调用其它函数。`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。关于函数的详细内容将在以后第五章介绍。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。

上例中程序的执行过程是，首先在屏幕上显示提示串，请用户输入两个数，回车后由

scanf 函数语句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。

1.6 C 源程序的结构特点

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 命令仅为其中的一种), 预处理命令通常应放在源文件或源程序的最前面。
5. 每一个说明, 每一个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, 也可不再加空格来间隔。

1.7 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。

1.8 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它可表示的图形符号。

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。在其它地方出现时, 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, 对程序的编译不发生影响, 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

1.9 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，注释符等。

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)、下划线(_)组成的字符串，并且其第一个字符必须是字母或下划线。

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，作到“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几类：

(1)类型说明符

用于定义、说明变量、函数或其它数据结构的类型。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，分隔各个变量。空格多用于语句各单词之间，作间隔符。在关键字，标识符之间必须要有一个以上的空格符作间隔，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。在后面章节中将专门给予介绍。

6. 注释符

C 语言的注释符是以 “/*” 开头并以 “*/” 结尾的串。在 “/*” 和 “*/” 之间的即为注释。程序编译时，不对注释作任何处理。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。在调试程序中对暂不使用的语句也可用注释符括起来，使翻译跳过不作处理，待调试结束后再去掉注释符。

1.10 Turbo C 2.0 集成开发环境的使用

1.10.1 Turbo C 2.0 简介和启动

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, TurboPascal, 这些软件很受用户欢迎。

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++ 作了更新, 即 Turbo C 的新一代产品 Borlandc C++ 也已经问世了。

1.10.2 Turbo C 2.0 集成开发环境

进入 Turbo C 2.0 集成开发环境中后, 屏幕上显示:



其中顶上一行为 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。

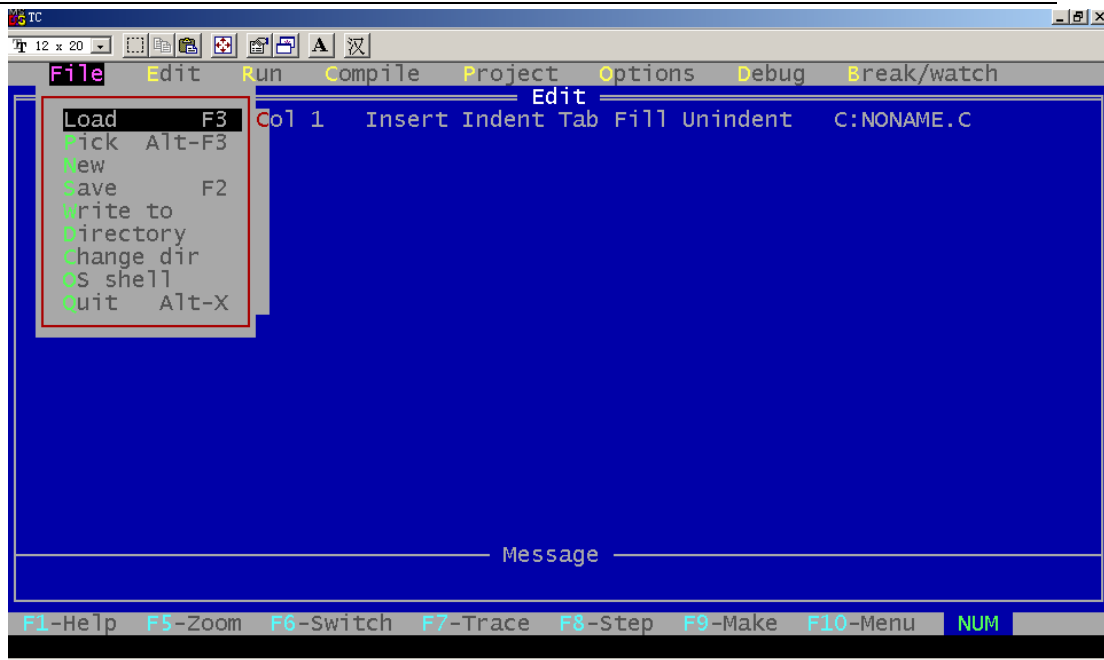
主菜单在 Turbo C 2.0 主屏幕顶上一行, 显示下列内容:

File Edit Run Compile Project Options Debug Break/watch

除 Edit 外, 其它各项均有子菜单, 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。

1.10.3 File 菜单

按 Alt+F 可进入 File 菜单, 如图:



File 菜单的子菜单共有 9 项, 分别叙述如下:

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。也可装入其它扩展名的文件, 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 状态下键入 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, 回车则执行。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。

1.10.4 Edit 菜单

按 Alt+E 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。编辑方法基本与 wordstar 相同, 可用 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+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

说明:

1) Turbo C 2.0 的双界符包括以下几种符号:

- a) 花括号 {和}
- b) 尖括号 <和>
- c) 圆括号 (和)
- d) 方括号 [和]
- e) 注释符 /*和*/
- f) 双引号 "
- g) 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。

1.10.5 Run 菜单

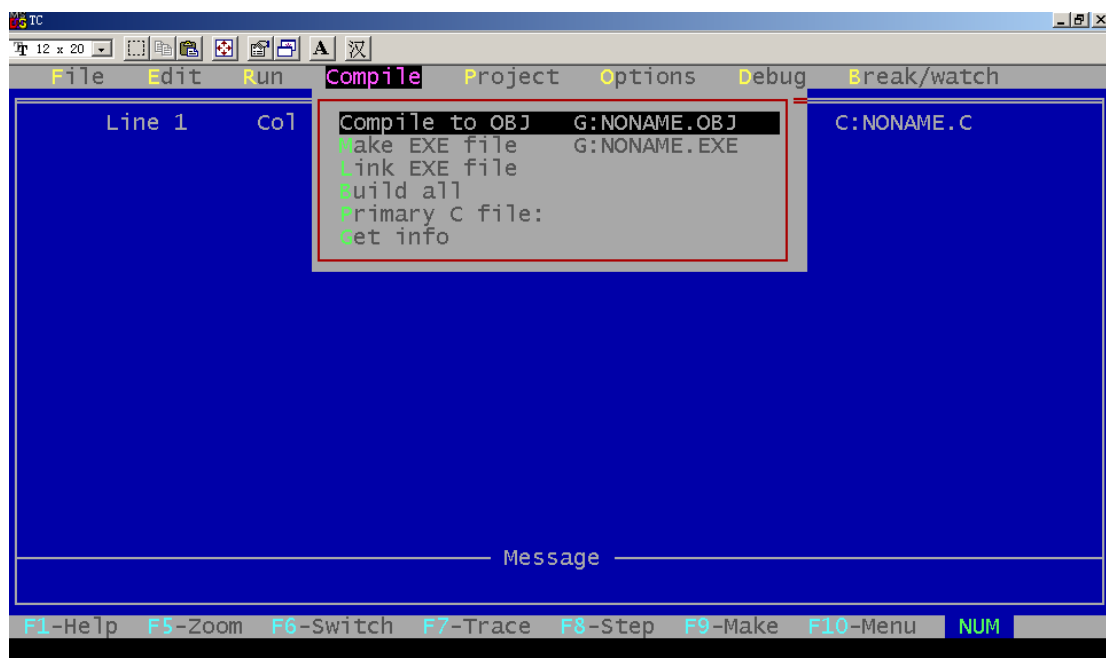
按 Alt+R 可进入 Run 菜单, 该菜单有以下各项, 如图所示:



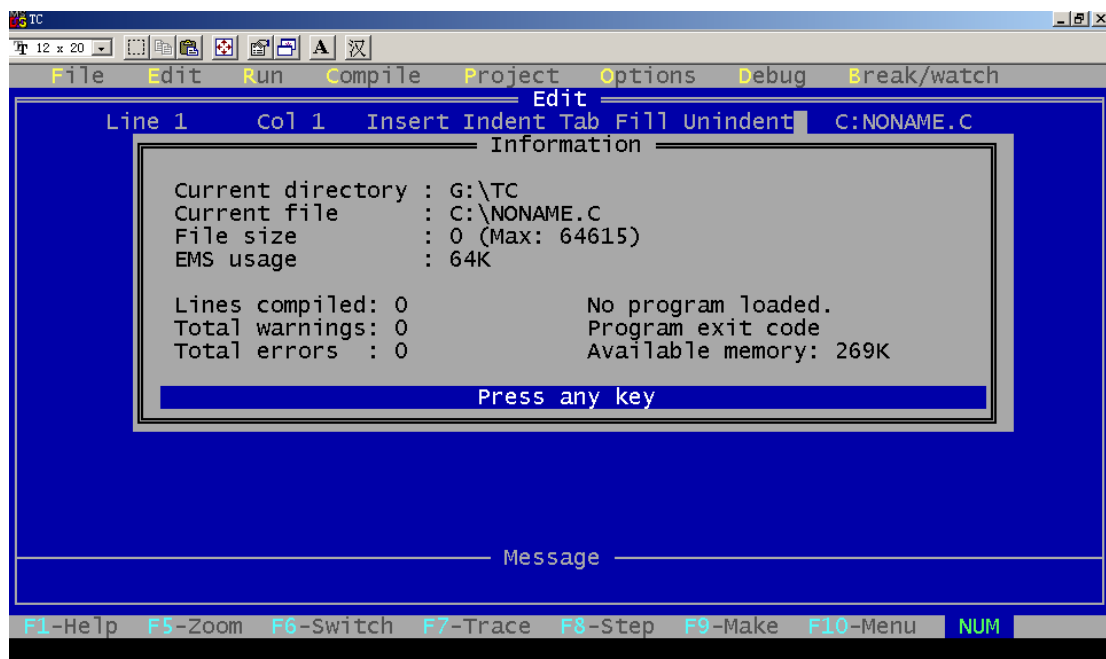
1. **Run:** 运行由 Project/Project name 项指定的文件名或当前编辑区的文件。如果对上次编译后的源代码未做过修改, 则直接运行到下一个断点(没有断点则运行到结束)。否则先进行编译、连接后才运行, 其热键为 **Ctrl+F9**。
2. **Program reset:** 中止当前的调试, 释放分给程序的空间, 其热键为 **Ctrl+F2**。
3. **Go to cursor:** :调试程序时使用, 选择该项可使程序运行到光标所在行。光标所在行必须为一条可执行语句, 否则提示错误。其热键为 **F4**。
4. **Trace into:** 在执行一条调用其它用户定义的子函数时, 若用 **Trace into** 项, 则执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**。

1.10.6 Compile 菜单

按 **Alt+C** 可进入 **Compile** 菜单, 该菜单有以下几个内容, 如图所示:

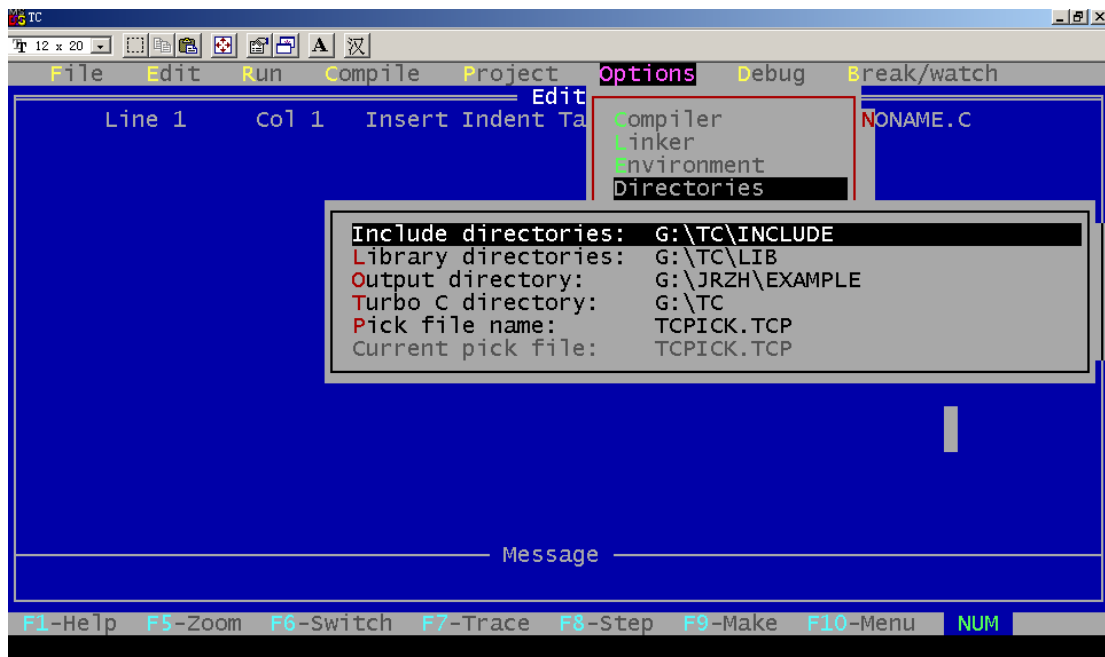


1. Compile to OBJ: 将一个 C 源文件编译生成.OBJ 目标文件, 同时显示生成的文件名。其热键为 Alt+F9。
2. Make EXE file: 此命令生成一个.EXE 的文件, 并显示生成的.EXE 文件名



Options 菜单

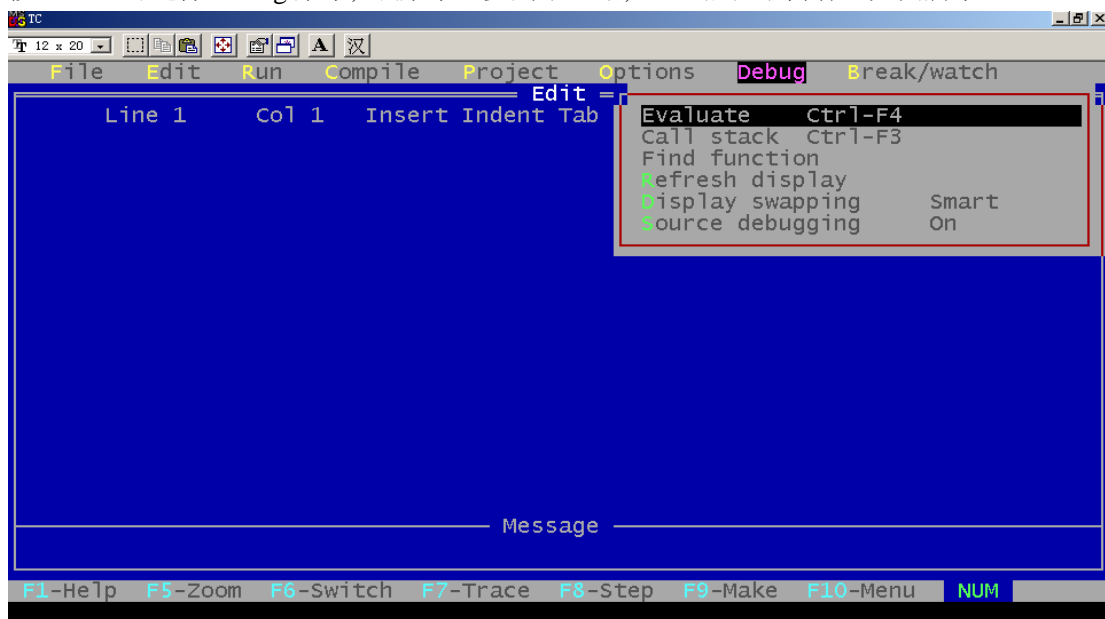
1. Directories: 规定编译、连接所需文件的路径, 有下列各项, 如图所示:



- (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 directoried: 输出文件(.OBJ, .EXE, .MAP 文件)的目录。
 - (4) Turbo C directoried: 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。
2. Arguments: 允许用户使用命令行参数。
 3. 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。
 4. Retrive options 装入一个配置文件到 TC 中, TC 将使用该文件的选择项。

1.10.7 Debug 菜单

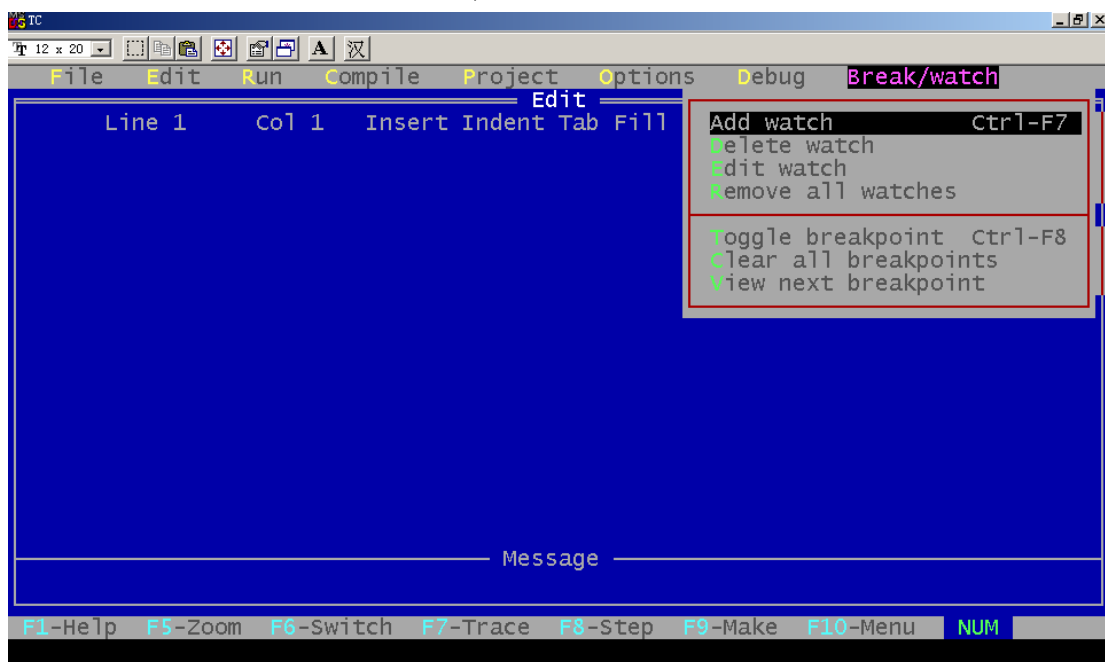
按 Alt+D 可选择 Debug 菜单, 该菜单主要用于查错, 它包括以下内容, 如图所示:



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 debugger 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可用此恢复编辑窗口的内容。

1.10.8 Break/watch 菜单

按 Alt+B 可进入 Break/watch 菜单, 该菜单有以下内容, 如图所示:



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。

本课总结：

C 语言概述

C 语言特点

C 语言的词汇

TC 的简单使用