

C语言及其应用

邱希春 凌云飞 编著
许华虎 薛友义

C语言及其应用

上海科学

312

XC/1

上海科学普及出版社

T1142
LXC/1

C 语言及其应用

邱希春 凌云飞 编著
许华虎 薛友义



上海科学普及出版社

336602

内 容 提 要

本书既是C语言的教科书,又是应用C语言的参考书。作者集多年教学经验,站在学生的角度,由浅入深地用通俗生动的叙述和丰富的应用示例引导读者理解内容、分析程序、学会应用。本书章节后均有习题,供测试和巩固该章节内容。本书结合教学内容给出许多应用示例,如:时钟的利用、DOS软中断的调用、目录控制、用户界面设计、打印机控制、图形、C与汇编的接口、常驻内存程序的设计、C与声音、C与汉字、C与Oracle等等。使读者在开发计算机的某种功能时,有一个参考的样本。

本书可作为大中专院校非计算机专业的教材,也可供C语言自学者和程序员参考。

(沪)新登字第 305 号

责任编辑 郭子安

JS202/24

C 语言 及 其 应 用

邱希春 凌云飞 许华虎 薛友义 编著

上海科学普及出版社出版

(上海曹杨路500号 邮政编码200063)

新华书店上海发行所发行 常熟文化印刷厂印刷

开本 787×1092 1/16 印张 20.5 字数 482000

1996年9月第1版 1996年9月第1次印刷

ISBN 7-5427-1153-9/TP·278 定价:22.00元

前 言

这是一本 C 语言的教科书。作者多年从事 C 语言的教学,深知在学习 C 语言过程中所遇到的一些问题。作者有心想站在学生的角度由浅入深地、用通俗生动的语言引导学生理解内容,学会应用。

本书又是一本应用 C 语言的参考书。作者在该书中给出了许多应用的例子,如计算机时钟的利用、DOS 软中断的调用、目录控制、用户界面设计、打印机控制、图形、C 与汇编语言的接口、常驻内存程序的设计、C 与声音、C 与汉字、C 与 Oracle 等等。使得读者在开发计算机的某种功能时,有一个参考的样本。

C 语言在计算机软件设计中是一种十分受欢迎的语言,之所以如此,是因为得益于“C 语言是一种具有低层功能的高级语言”,把过去只能用汇编语言才能处理的问题从少数人手中解放出来。但是语言作为一种工具总是有针对性的,不可能也不应该苛求一种工具对于计算机的所有应用范围都是方便的、适宜的。比如,对管理信息系统来说,使用数据库管理系统,如 FoxBASE, FoxPro(虽然它们本身是由 C 语言编写的)可能更方便一些;若从事数值计算可能用 FORTRAN 语言更合适一些(不排斥少量子程序用 C 语言编写)。本书在编写过程中注意到了这一点,从而给读者以正确的引导。

学习一种计算机语言总要学习它的语法规则,这当然是必要的。但我们认为还有比语法更重要的东西需要特别加以注意:其一是注意培养自己的抽象思维和计算机得以实现的思维方式。学习语言总是为了运用它写出一个软件来,但这个软件的结构是否清晰、合理,是和人的思维方式和抽象能力有关系的,这类似于懂得中文语法不见得就能写出段落清楚、主题思想明确的好文章一样。其二是注意该语言的编程习惯和风格。培养一种良好的编程习惯和风格可使程序避免许多错误并且容易调试。这实质上是一种科学作风,这种作风只有在不断编写程序、调试程序中才能养成。因此,用心地去实践是学习一门语言的必经之路。

本书由四位教师共同编写的,邱希春编写第二、三、四章,凌云飞编写第五、七章,许华虎编写第六章,薛友义编写第一章,四位作者共同编写了第八章,全书由邱希春整理、审核。在编写出版过程中,范荷英和刘玲帮助做了许多辅助性工作,在此致以衷心的感谢。

书中错误在所难免,望读者不吝指正。

编 者

于上海大学

目 录

第 1 章 引论	1
1.1 C 语言简史	1
1.2 C 语言的特点	1
1.3 C 语言程序实例	2
1.4 基本输入输出函数	5
1.4.1 putchar 函数	5
1.4.2 printf 函数	5
1.4.3 getchar 函数	6
1.4.4 scanf 函数	7
1.5 C 语言程序的开发过程	8
1.5.1 启动 Turbo C	8
1.5.2 编辑源程序	9
1.5.3 编译、运行源程序	9
第 2 章 标识符、变量、运算符及表达式	10
2.1 标识符	10
2.2 常量与变量	10
2.2.1 常量	10
2.2.2 变量	13
2.2.3 不同类型变量值的转换	15
2.3 运算符	17
2.3.1 算术运算符	17
2.3.2 关系运算符	19
2.3.3 逻辑运算符	19
2.3.4 按位逻辑运算符	20
2.3.5 条件运算符	21
2.3.6 赋值运算符	22
2.3.7 逗号运算符	23
2.4 表达式	23
2.5 运算符的优先级与结合性	23
习题	25
第 3 章 语句	27
3.1 简单语句与复合语句	27

3.2 条件语句.....	27
3.3 分支语句——switch	31
3.4 循环语句.....	34
3.4.1 while 循环	34
3.4.2 for 循环	35
3.4.3 do-while 循环	37
3.5 break、continue、goto 及 return 语句	38
3.5.1 break 语句	38
3.5.2 continue 语句	40
3.5.3 goto 语句	41
3.5.4 return 语句	41
习题	42
第4章 函数	43
4.1 函数的定义.....	44
4.1.1 函数的定义.....	44
4.1.2 举例.....	45
4.2 函数的调用.....	47
4.2.1 函数调用格式.....	47
4.2.2 函数调用中的一些问题.....	47
4.3 递归.....	50
4.4 变量的存储类、作用域及初始化	54
4.4.1 变量的存储类与作用域.....	54
4.4.2 变量的初始化.....	58
4.5 预处理程序.....	59
4.5.1 无参#define	59
4.5.2 有参数的宏定义.....	61
4.5.3 包含文件.....	62
4.5.4 条件编译命令.....	62
4.5.5 预定义的一些特殊标识符.....	64
4.6 算法之例.....	65
4.6.1 排序.....	65
4.6.2 搜索.....	68
4.6.3 递归与回溯——八皇后问题.....	70
习题	73
第5章 指针与数组	74
5.1 指针变量的定义及引用.....	74
5.1.1 指针的说明与赋值.....	74
5.1.2 指针的引用与运算.....	75
5.2 指向字符串的指针.....	77

5.3 函数与指针.....	82
5.3.1 指针作为函数的参数.....	82
5.3.2 指针作为函数的返回值.....	84
5.3.3 指向函数的指针.....	85
5.4 数组与指针.....	87
5.4.1 数组的概念.....	87
5.4.2 一维数组.....	87
5.4.3 多维数组.....	91
5.5 指针数组和多级指针.....	94
5.6 命令行参数.....	97
习题.....	103
第6章 结构和联合.....	105
6.1 结构的定义和引用	105
6.1.1 结构的定义形式	105
6.1.2 定义结构类型变量的方法	106
6.1.3 结构变量的引用	108
6.1.4 结构类型变量的初始化	109
6.2 结构数组	109
6.2.1 结构数组的定义	109
6.2.2 结构数组的初始化	110
6.2.3 举例	111
6.3 结构指针	111
6.3.1 指向结构变量的指针	112
6.3.2 指向结构数组的指针	113
6.3.3 指向结构的指针作函数参数	114
6.4 结构应用(一)——链表	115
6.4.1 链表的建立	115
6.4.2 链表的输出	118
6.4.3 链表的插入	119
6.4.4 链表的删除	120
6.4.5 双向链表	121
6.4.6 综合举例	131
6.5 结构应用(二)——位域	135
6.5.1 位域的定义	135
6.5.2 位域变量的引用	136
6.6 联合	137
6.6.1 联合的定义	137
6.6.2 联合变量的引用方式	137
6.6.3 联合类型数据的特点	138

6.7 枚举类型	140
6.8 用 typedef 语句定义类型	141
习题	143
第7章 输入输出函数与文件	144
7.1 终端输入输出	144
7.1.1 数据流中单个字符的输入输出	145
7.1.2 格式控制输入输出函数 scanf 和 printf	146
7.1.3 字符串的输入输出	150
7.1.4 内存中的格式变换——sprintf 函数与 sscanf 函数	151
7.1.5 控制台输入输出	152
7.2 文件的输入输出	153
7.2.1 文件的概念	153
7.2.2 文本文件的读写	154
7.2.3 二进制文件的读写	157
7.2.4 文件读写错误的处理	161
7.3 低级文件输入输出	162
7.3.1 打开文件	163
7.3.2 当前可读写位置的设定	164
7.3.3 读写数据	164
7.3.4 关闭文件	164
7.3.5 其他常用函数	165
7.4 网络上文件的读写	166
7.4.1 打开网络服务器上的文件	166
7.4.2 加锁与解锁	167
习题	169
第8章 C 语言应用	170
8.1 时钟的设置和读取	170
8.1.1 日期的读取和设置	170
8.1.2 时间的读取和设置	171
8.2 DOS 软中断调用	171
8.2.1 鼠标器控制函数	172
8.2.2 RS-232 异步串行通讯口读写函数	174
8.3 目录控制函数	179
8.4 用户界面设计	182
8.4.1 窗口	182
8.4.2 字符串的显示	184
8.5 C 与图形	185
8.6 C 与汇编语言的接口技术	189
8.6.1 C 调用汇编子程序	189

8.6.2 Turbo C 与汇编语言的接口方法	189
8.6.3 建立汇编语言的框架程序	192
8.6.4 接口程序的编译、链接和运行	198
8.6.5 C 语言行间嵌入汇编语句	199
8.7 C 与驻留内存程序 TSR	200
8.8 声音和音乐	208
8.8.1 引言	208
8.8.2 产生声音	209
8.8.3 延时	210
8.8.4 听力测试程序实例	211
8.8.5 音响模拟程序实例	212
8.8.6 音乐演奏程序实例	214
8.9 西文操作系统状态下的汉字显示技术	216
8.9.1 西文 DOS 状态下显示 16×16 点阵汉字	217
8.9.2 西文 DOS 状态下显示 24×24 点阵汉字	222
8.10 全屏幕编辑器的设计原理及实例	226
8.10.1 全屏幕编辑系统中的一些重要功能和参数	226
8.10.2 全屏幕编辑器的主要函数	227
8.10.3 全屏幕编辑软件源程序	242
8.11 C 与 ORACLE 语言混合编程	263
8.11.1 PRO * C 程序的结构	263
8.11.2 数据更新操作	269
8.12 C 语言控制打印机	272
附录 A Turbo C 库函数	275
A.1 数学函数	275
A.2 输入输出函数	277
A.3 字符函数	283
A.4 字符串函数	284
A.5 动态存储分配函数	287
A.6 时间函数	287
A.7 其他函数	288
A.8 Turbo C 特殊函数	290
A.8.1 接口库函数	291
A.8.2 字符屏幕控制函数	294
A.8.3 图形处理函数	297
附录 B 键盘码表	306

第1章 引 论

1.1 C语言简史

C语言诞生于70年代初期,它的前身是英国剑桥大学的Martin Richards在60年代开发的BCPL语言。该语言是Martin Richards为描述和实现UNIX操作系统,而为自己设计的工作语言。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,并用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语言作了进一步的完善和发展,提出了一种新型的程序设计语言——C语言。1973年,K. Thompson和Dennis M. Ritchie合作用C语言成功地改写了UNIX操作系统。自从C语言问世以来,表现了极强的生命力,从最初为记述UNIX操作系统而开发的语言,到现在已经发展成为广泛应用的系统描述语言和通用的程序设计语言。随着微型计算机软硬件的发展,C语言已经成为微机上用于开发各种软件尤其是系统软件的主要工具之一。

1.2 C语言的特点

C语言的基本特点有:

(1) C语言是介于汇编语言和高级语言之间的一种程序设计语言。C语言与硬件系统比较接近,它有直接访问硬件的功能,并具有汇编语言的大部分功能;C语言又具有高级语言面向用户、容易记忆、便于编程和阅读的优点。所以C语言既是系统描述语言,又是通用的程序设计语言。

(2) 函数是C语言的基本单位。C语言程序是由若干个函数组成,其中必包含一个main()函数,即主函数。main()函数由用户自己定义,程序由main()函数开始执行。其余的函数可由用户自己定义,也可利用C语言函数库中任意一个函数。由于这个特点,C语言便于实现程序的模块化设计。

(3) C语言运算符丰富,数据结构类型广泛,同时具有结构化控制语句,是一种结构化程序设计语言,即具有顺序、分支、循环三种基本结构。

(4) C语言语言简洁、紧凑,使用方便灵活,语法限制不太严格,程序设计自由度大。

(5) C语言生成的目标代码质量高,程序执行效率高。而且用C语言写的程序移植性好。

1.3 C 语言程序实例

本节用三个简单的例子,介绍 C 语言程序的概貌、程序中使用到的一些语句和函数等,读者不必追求细节,只需从整体上了解即可。在以后的各个章节中,将会详细介绍。

[例 1.1] 从键盘上输入两个整数,并以 $a+b=c$ 的形式输出该两数的和。程序如下:

```
#include <stdio.h>
#include <conio.h>
void main()                                /* 函数说明 */
{
    int a,b,c;                             /* 变量 a,b,c 说明 */
    printf("input two number a,b :");     /* 输出" "内的一句话 */
    scanf("%d,%d",&a,&b);                 /* 输入 a,b 两个数 */
    c=a+b;                                 /* 计算 a+b 的和 */
    printf("%d+%d=%d\n",a,b,c);           /* 输出形如 a+b=c 的结果 */
    getch();                               /* 等待从键盘上键入字符 */
}
```

其中,main()是主函数。每一个 C 语言程序必须有一个 main 函数,程序从 main 函数开始执行。函数由函数说明和函数体组成。函数体由一对大括号{ }括起来,一般包含变量说明和若干语句。printf(),scanf(),getch()是 C 语言的三个库函数。该函数包含在 stdio.h 和 conio.h 中,它必须在使用前用#include 语句加以说明。关于函数和#include 语句的使用,详见后面各章节的说明。

该程序运行后,首先在屏幕上显示:

input two number a,b :

然后可键入两个数,例如 12,16,按回车后,屏幕显示:

12+16=28

然后等待,按任一键后程序运行结束。

[例 1.2] DOS 系统调用,给出当前驱动器的盘符。程序如下:

```
#include <stdio.h>
#include <dos.h>
#include <conio.h>
char current_drive(void);
void main(void)
{
    printf("The current drive is %c:\n",current_drive());
    getch();
}

/* Get current drive as 'A','B',... */
char current_drive(void)
```

```

{
    char curdrive;
    /* Get current disk as 0,1,... */
    curdrive = bdos(0x19,0,0);
    return('A'+curdrive);
}

```

此程序包含了两个函数,主函数 main 和被调用的函数 current_drive。current_drive 函数给出当前驱动器的盘符 'A', 'B' ..., 主函数打印出当前驱动器的盘符。设当前驱动器为 C 盘,则该程序运行后在屏幕上显示的结果为:

The current drive is c;

此程序中的函数 bdos(0x19,0,0),用来调用 DOS 系统的功能,返回当前驱动器的数字编号,而返回语句 return('A'+curdrive),则将数字编号转换为盘符。

[例 1.3] 变幻线程序。此程序在屏幕上画出一个大小、形状、颜色不断变化的四边形。稍作改动,即是 Windows 下的一个屏幕保护程序。程序如下:

```

#include <stdlib.h>
#include <graphics.h>
#include <io.h>
void scale_xy(double *,double *,double *,double *,int *);
main()
{
    int i,j,k,l;
    long now,s;
    double x[4],y[4],step_x[4],step_y[4];
    i=DETECT;
    initgraph(&i,&j,"e:\\tc"); /* 初始化图形系统 */
    srand(time(&now)%37);      /* 取随机数 */
    for (i=0;i<4;i++)
    {
        /* 随机取四边形四个顶点和边长 */
        x[i]=random(500)+50;
        y[i]=random(300)+50;
        step_x[i]=random(20);
        step_y[i]=random(20);
    }
    k=2;
    setbkcolor(BLACK);          /* 取背景色 */
    setcolor(k);                /* 取前景色 */
    do {
        for (i=0;i<4;i++)
            scale_xy(x+i,y+i,step_x+i,step_y+i,&k); /* 改变顶点坐标 */
    }
}

```

```

for (l=0;l<=10;l+=1)
{
    /* 以前景色画四边形 */
    line(x[0]+l, y[0]-l, x[1]-2 * l, y[1]+2 * l);
    line(x[1]-2 * l, y[1]+2 * l, x[2]+3 * l, y[2]-3 * l);
    line(x[2]+3 * l, y[2]-3 * l, x[3]-4 * l, y[3]+4 * l);
    line(x[3]-4 * l, y[3]+4 * l, x[0]+l, y[0]-l); }
delay(200);
setcolor(BLACK);
for (l=0;l<=10;l+=1)
{
    /* 以背景色画四边形 */
    line(x[0]+l, y[0]-l, x[1]-2 * l, y[1]+2 * l);
    line(x[1]-2 * l, y[1]+2 * l, x[2]+3 * l, y[2]-3 * l);
    line(x[2]+3 * l, y[2]-3 * l, x[3]-4 * l, y[3]+4 * l);
    line(x[3]-4 * l, y[3]+4 * l, x[0]+l, y[0]-l); }
setcolor(k);
}while(! kbhit());
}

void scale_xy(double * x,double * y,double * step_x,double * step_y,int * color)
{
    /* 改变顶点坐标和颜色函数 */
    if (* x>=600 || * x<=40)
    {
        * step_x * = -1;
        if(* color==7) * color+=1;
        * color = * color%15+1;
    }
    if (* y>=440 || * y<=40)
    {
        * step_y * = -1;
        if(* color==7) * color+=1;
        * color = * color%15+1;
    }
    * x += * step_x;
    * y += * step_y;
}

```

读者可上机实践一下,暂时不必理会程序的细节。程序中的注解行是用中文写成,上机时可不必输入。

从上述例子中可以看到:

(1) C 程序是由若干个函数组成的,其中至少包含一个 main 函数,其他的函数可由用户根据需要自己编制,例如 current_drive,也可利用 C 语言系统提供的库函数;例如 scanf, printf。函数在程序中的先后次序是无关紧要的,程序从 main 函数开始运行。

(2) C 语言程序的书写格式比较自由,一行内可以写几个语句,例如 setbkcolor

(BLUE); getch(); 一个语句也可分几行书写。语句没有行号。

(3) 每个语句和数据定义的最后必须有一个分号,分号是语句的必要组成部分,不可省略。

(4) 由/*...*/组成的部分是C语言的注释部分。一个好的程序,都应当加上必要的注释,以增加程序的可读性。

1.4 基本输入输出函数

C语言本身不提供输入输出语句,其输入输出是由C语言标准函数库提供的函数完成的。为了后续章节的使用,本节先简单地介绍四个基本输入输出函数,细节请参见第7章。

1.4.1 putchar 函数

格式: putchar(c);

作用: 输出字符变量的值,也可以输出控制字符,如 putchar('\n'),则输出一个换行符。

[例 1.4]

```
#include <stdio.h>
main()
{ char a,b,c;
  a='Y' ; b='e' ; c='s' ;
  putchar(a); putchar(b); putchar(c);putchar('\n');
  putchar(a); putchar('\n');putchar(b);putchar('\n'); putchar(c);
}
```

运行的结果在屏幕上输出:

Yes

Y

e

s

1.4.2 printf 函数

格式: printf(格式控制,输出表);

作用: 向终端输出若干个任意类型的数据。

其中:

(1) 格式控制是用双引号括起来的字符串,它由格式说明和普通字符组成。格式说明是由%和格式字符组成,它的作用是将输出的数据转换为指定的格式输出,并总是以%开始。

格式字符为:

d 用来输出十进制整数。

- o 以八进制形式输出整数。
- x 以十六进制形式输出整数。
- u 用来输出无符号数,以十进制形式输出。
- c 用来输出一个字符。
- s 用来输出一个字符串。
- f 以小数形式输出单、双精度数,数字部分小数位数为 6 位。

普通字符是需要按字符原样输出的字符。

(2) 输出表是需要输出的数据,可以是表达式。各输出项之间用逗号分隔。

例如:

```
printf ("%d %c \n",a,b);
```

↑
↑
 格式说明 输出表

```
printf ("a=%d b=%c \n",a,b)
```

↑
↑
 格式说明 输出表

[例 1.5]

```
#include <stdio.h>
main()
{
    int a;
    char b;
    a=12 ; b='y' ;
    printf ("%d %c \n", a, b) ;
    printf ("a=%d b=%c \n", a, b);
}
```

运行的结果为:

```
12 y
a=12 b=y
```

1.4.3 getchar 函数

格式: getchar()

作用: 从终端输入一个字符。

[例 1.6]

```
#include <stdio.h>
main()
{
    int a;
```

```

a=getchar();
putchar(a);
printf(" a=%c \n", a);
}

```

运行时,若从键盘输入字符 y,则结果为:

```

y          输入 y 后,要按回车键
y a=y      按两种格式输出变量 a 的值

```

1.4.4 scanf 函数

格式: scanf(格式控制,地址表)

作用: 输入任何类型的多个数据。

其中,格式控制的含义同 printf 函数,地址表是由若干地址组成的表,可以是变量的地址,或字符串的首地址。例如:

```

scanf("%d %d", &a,&b);
      ↑      ↑
      格式说明 地址表

```

其中,&a,&b 分别表示变量 a,b 在内存中的地址,此 scanf 函数的作用是将 a,b 的值分别存到 a,b 所对应的内存单元中去。

格式字符有:

- d 用来输入十进制整数。
- o 用来输入八进制整数。
- x 用来输入十六进制整数。
- c 用来输入一个字符。
- s 用来输入字符串。
- f 用来输入小数形式的单、双精度数。

其他的格式字符,请见第 7 章。

[例 1.7]

```

#include <stdio.h>
main()
{
    int a,b;
    scanf("%d %d",&a,&b);
    printf("a=%d b=%d \n", a,b);
}

```

运行时,若输入 12 34 则输出为:

```
a=12 b=34
```


1.5 C 语言程序的开发过程

用 C 语言开发一个程序的过程如下：

(1) 编制源程序,即根据问题的实际情况,用 C 语言编写程序,这个程序称为 C 语言的源程序。

(2) 编辑源程序,用计算机系统提供的编辑程序或用 C 语言编译系统自身提供的编辑系统,将源程序输入计算机,并通过修改、编辑后存入文件中。该源程序以文本文件的形式存储在计算机中。源文件的名字由用户自己定义,扩展名一般定为“.C”(在 Turbo 系列中),或“.CPP”(在 Borland C++ 中)。

(3) 编译源程序,用 C 编译程序对源程序进行编译生成目标程序,扩展名为“.OBJ”。编译时还对源程序的语法和程序的逻辑结构等进行检查,当发现错误时,在显示器上列出错误的类型和位置,此时要对源文件再次进行编辑。

(4) 链接,将目标程序和库函数连接成可执行的程序,扩展名为“.EXE”

(5) 运行生成的执行程序。

下面以 Turbo C 为例简单介绍 C 程序编辑、运行的过程。详细请查看 Turbo C 使用 and 参考手册。

1.5.1 启动 Turbo C

在 DOS 提示符下,键入 TC,回车后立即出现 Turbo C 的主屏幕菜单,如图 1.1 所示,该主屏幕分成四个部分:主菜单、编辑窗口、消息窗口和功能键提示行。

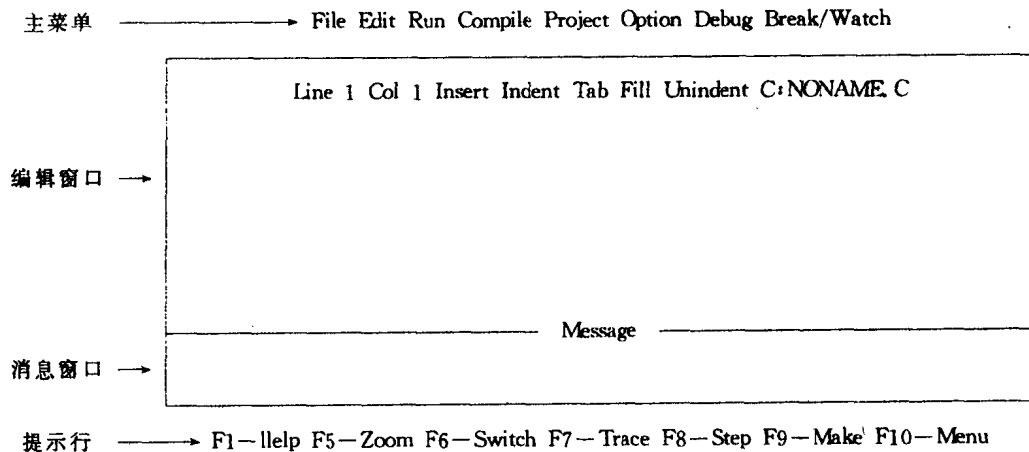


图 1.1 TC 主屏幕

用 F10 键激活主菜单。用光标移动键可以在主菜单中移动,并下拉出一个菜单,选中某一菜单选项后,按回车键。用 Esc 键退出主菜单。