



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



工程图学(2)



内 容

- C语言先修知识
- C语言图形屏幕设置
- C语言图形函数及应用

内 容

- C语言先修知识
- C语言图形屏幕设置
- C语言图形函数及应用

C语言先修知识

- C语言的符号与规定
 - 命名规则
 - 符号—控制字符、注释符号、变量类型、运算符、增1和减1算符
- 输入与输出
 - 字符的输入与输出 **getch()**, **putch()**
 - 格式化输入与输出 **scanf()**, **printf()**

格式可以baidu或者查看相关*.h头文件
conio.h、stdio.h

C语言先修知识

- C语言控制语句

- 循环语句—while、do~while 、 for

- if条件句

- switch语句

- 函数、数组与指针

- 文件的读写

- fopen(), fclose();
 - fscanf(), fprintf();

} stdio.h

Basic Graphics Interface

图形接口文件

头文件

ATT.BGI
BGIDEMO.C
BOLD.CHR
CGA.BGI
EGAVGA.BGI
EURO.CHR
GOTH.CHR
HERC.BGI
IBM8514.BGI
LCOM.CHR
LITT.CHR
PC3270.BGI
SANS.CHR
SCRI.CHR
SIMP.CHR
TRIP.CHR
TSCR.CHR

_DEFS.H
_NULL.H
ABSTARRY.H
Alloc.h
ARRAY.H
ARRAYS.H
Assert.h
ASSOC.H
BAG.H
BAGS.H
BCD.H
Bios.h
BTREE.H
CHECKS.H
CLSDEFS.H
CLSTYPES.H
COLLECT.H
COMPLEX.H
Conio.h
CONSTREA.H
CONTAIN.H
Ctype.h
DBLLIST.H
DEF.H
DEF2.H
DEQUE.H
DEQUES.H
DICT.H
DICTION.H
Dir.h
DIRECT.H
DIRECTRY.H
DIRENT.H
DLISTIMP.H
Dos.h
Errno.h
Fcntl.h
FIGURES.H
FILEDATA.H
FILTER.H
Float.h
FSTREAM.H
GENERIC.H
Graphics.h
HASHTBL.H
Io.h
IOMANIP.H
IOSTREAM.H
LDATE.H
Limits.h
LIST.H
LIST2.H
LISTIMP.H
LOCALE.H
LOCKING.H
LTIME.H
Malloc.h
Math.h
Mem.h
MEMMGR.H
MEMORY.H
NBMP.H
NCOLOR.H
NCOMMON.H
NDRAW.H
NEFFECT.H
NEMS.H
NEO.H
NERROR.H
NEW.H
NGUI.H
NICON.H
NINPUT.H
NOTHER.H
NSCROLL.H
NSOUND.H
NTIMER.H
NVESA.H
NWORD.H
NXMS.H
OBJECT.H
POINT.H
PRIORTYQ.H
Process.h
QUEUE.H
QUEUES.H
RESOURCE.H
SEARCH.H
semLib.h
SET.H
Setjmp.h
SETS.H
Share.h
SHDDEL.H
Signal.h
SORTABLE.H
SORTARRY.H
STACK.H
STACK2.H
STACKS.H
STAT.H
Stdarg.h
Stddef.h
Stdio.h
STDIOSTR.H
Stdlib.h
STDTEMPL.H
String.h
STRNG.H
STRSTREA.H
Svgacc.h
TCALC.H
Time.h
TIMEB.H
TIMER.H
TYPES.H
UTIME.H
Values.h
VARARGS.H
VECTIMP.H
VPOINT.H
vxWorks.h
编译开关.txt
读我.txt

内 容

- C语言先修知识
- C语言图形屏幕设置
- C语言图形函数及应用

C语言图形屏幕设置

一小段程序：

```
#include "graphics.h"
main()
{int drive,mode;
drive=VGA;
mode=VGAMED;
initgraph(&drive,&mode,"c:\\tc");
setbkcolor(4);
setcolor(3);

circle(100,100,50);
line(100,100,200,200);
getch();
closegraph();
}
```


C语言图形屏幕设置

各部分含义：

头文件

```
#include "graphics.h"
```

程序主函数

```
main()
```

```
{
```

```
int drive,mode;
```

```
drive=VGA;
```

```
mode=VGAMED;
```

```
initgraph(&drive,&mode, " c:\\tc");
```

```
....
```

```
}
```

程序主体

C语言图形屏幕设置

各部分含义：

#include “graphics.h”

先到文件所在目录寻找graphics.h，找不到时再按系统指定的路径进行检索

#include <graphics.h>

直接按系统指定的路径寻找graphics.h

C语言图形屏幕设置

图形系统初始化-函数

`initgraph(&drive,&mode,path);`

作用： 对图形系统进行初始化。

- 装入相应的图形显示器的驱动程序，
- 选择显示的模式，
- 指明驱动程序存放的路径。

C语言图形屏幕设置

图形系统初始化-参数含义

drive: 图形驱动程序（程序）
扩展名为BGI

赋值

➤ 图形适配器符号；
或

➤ 相应的数值代码

例如： `drive=VGA;`

或者 `drive=9;`

注意：
字母全部大写

C语言图形屏幕设置

图形系统初始化-参数含义

mode: 显示模式,
即相应的显示分辨率

赋值

➤ 图形显示模式代号;
或

➤ 相应的数值代码

例如: `mode=VGAHI;`

或者 `mode=2;`

注意:
字母全部大写

C语言图形屏幕设置

图形系统初始化-参数含义

path: 驱动程序所在的目录路径
赋值

➤完整的文件路径

例如: `initgraph(&drive,&mode, "c:\\tc");`
或 `initgraph(&drive,&mode, "d:\\turbo");`

若路径为“c:\\tc”,则可简写成“ ”

注意:
两个反斜杠
以及双引号

C语言图形屏幕设置

图形系统初始化-硬件测试

`detectgraph(&drive,&mode);`

作用： 检测当前图形适配器的类型和最佳显示模式

➤ 参数的含义同函数 `initgraph()`

C语言图形屏幕设置

图形系统初始化-硬件测试

应用
示例

```
#include "graphics.h"
main()
{
int drive,mode;
detectgraph(&drive,&mode);
initgraph(&drive,&mode, " c:\\tc");
....
}
```

变量drive本质表示一个存储单元
C语言中，drive表示单元里的数据
&drive表示单元地址

C语言图形屏幕设置VGA示例

若用VGA图形驱动程序,图形显示模式为VGAHI,
则调用方式如下:

```
int gdriver,gmode;
```

```
gdriver=VGA
```

```
gmode=VGAHI
```

```
initgraph(&gdriver,&gmode,"c:\\TC");
```

关闭图形方式函数为 closegraph()

C语言图形屏幕设置

屏幕颜色设置

屏幕背景色

`setbkcolor(color);`

屏幕前景色

`setcolor(color);`

参数color的赋值:

颜色的符号名（全部大写）或相应的数值

C语言图形屏幕设置

颜色代码

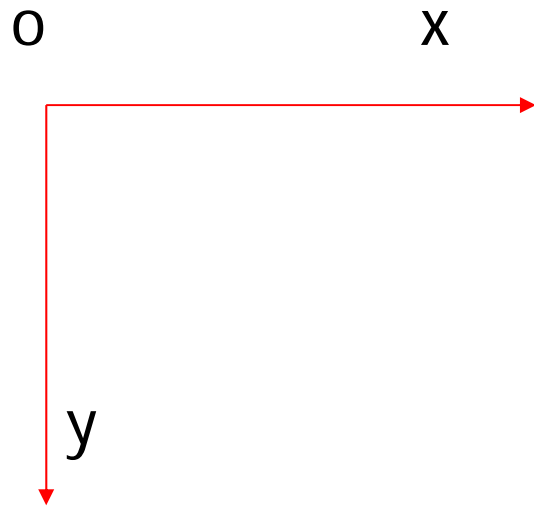
颜色	符号名	数值码
黑	BLACK	0
蓝	BLUE	1
绿	GREEN	2
青	CYAN	3
红	RED	4
紫红	MAGENTA	5
棕	BROWN	6
浅灰	LIGHTGRAY	7

颜色	符号名	数值码
深灰	DARKGRAY	8
浅蓝	LIGHTBLUE	9
浅绿	LIGHTGREEN	10
浅青	LIGHTCYAN	11
浅红	LIGHTRED	12
浅紫红	LIGHTMAGENTA	13
黄	YELLOW	14
白	WHITE	15

C语言图形屏幕设置

屏幕坐标及屏幕视口的设置—坐标

屏幕坐标架的原点在屏幕左上角



分辨率为**640x350**,
则

x坐标最大值为**639**

y坐标最大值为**349**

C语言图形屏幕设置

屏幕坐标及屏幕视口的设置—坐标函数

`getmaxx();`
`getmaxy();` } 获取屏幕的最
大X和Y坐标值

例句:

```
printf("xmax=%d",getmaxx());
```

```
printf("ymax=%d",getmaxy());
```

C语言图形屏幕设置

屏幕坐标及屏幕视口的设置—坐标函数

`getx();`
`gety();`

获取当前位置
的X和Y坐标值

例句:

```
printf("current position: x=%d,y=%d",getx(),gety());
```

C语言图形屏幕设置

屏幕坐标及屏幕视口的设置—屏幕视口

视口：屏幕上的一个矩形窗口

视口函数的**作用**：

- 建立一个视口；
- 视口左上角的坐标变成 $(0, 0)$ ；
- 控制视口外的图形是否显示

C语言图形屏幕设置

屏幕坐标及屏幕视口的设置—屏幕视口

`setviewport(xlu, ylu, xrd, yrd, k);`

`(xlu, ylu)`: 视口左上角坐标; left upper

`(xrd, yrd)`: 视口右下角坐标; right down

`k`: 裁剪控制参数

`k`为1时, 裁剪; `k`为0时, 不裁剪

`K=1`则超出视口的输出图形自动被裁剪掉, 即所有作图限制于当前图形视口之内, 如果`k`为0, 则不做裁剪, 即作图将无限制地扩展于视口周界之外, 直到屏幕边界。

C语言图形屏幕设置

屏幕坐标及屏幕视口的设置—屏幕视口

例句：

```
setviewport (20, 30, 180, 220, 1);
```

```
clearviewport();
```

- 此函数用于清除视口
- 把当前光标位置重置于屏幕左上角 (0, 0)

C语言图形屏幕设置

图形输出的活动页、可见页的设置

基本概念

- ❖ 图形屏幕可以看成一页显示图形的纸张；
- ❖ 显卡内存容量大时，可以容纳两幅或两幅以上的屏幕信息；即有多图形页；
- ❖ 在图形模式中，只有EGA和VGA图形卡的模型模式支持多图形页；

C语言图形屏幕设置

图形输出的活动页、可见页的设置

- ❖ 某页处于工作状态时，称作输出活动页；
- ❖ 当某页处于显示状态时，称为可见图形页；

C语言图形屏幕设置

图形输出的活动页、可见页的设置

❖ 设置图形输出活动页函数:

`setactivepage(page);`

❖ 设置可见图形页函数:

`setvisualpage(page);`

参数page的赋值:

0, 1, 2等页码数值

C语言图形屏幕设置

图形输出的活动页、可见页-示例

示例：画两幅图，然后交互显示

C语言图形屏幕设置

图形输出的活动页、可见页-示例

```
#include "graphics.h"
main()
{
    int drive,mode;
    drive=VGA;
    mode=VGAMED;
    initgraph(&drive,&mode,
    " c:\\tc");
    cleardevice();
```

C语言图形屏幕设置

图形输出的活动页、可见页-示例

```
setbkcolor(CYAN);  
setcolor(RED);  
setactivepage(0);  
    bar(50,50,200,200);  
setactivepage(1);  
    circle(200,200,50);  
loop:setvisualpage(0);  
    sleep(5);
```

C语言图形屏幕设置

图形输出的活动页、可见页-示例

```
    if(kbhit()) exit(0);  
    setvisualpage(1);  
    sleep(5);  
    if(kbhit()) exit(0);  
goto loop;  
    closegraph();  
}
```


C语言图形屏幕设置

关闭图形

`closegraph();`

作用：

- 使屏幕由图形显示模式返回文本模式；
- 释放图形模式下所占用的内存。

内 容

- C语言先修知识
- C语言图形屏幕设置
- C语言图形函数及应用

C语言图形函数及应用

- 线型设置函数;
- 直线函数;
- 圆、圆弧、椭圆弧;
- 多边形、点;
- 填充函数;
- 文本函数;
- 动画函数;

C语言图形函数及应用

线型设置函数

setlinestyle(L,M,W)

作用： 设置当前的线型类型和宽度

C语言图形函数及应用

线型设置函数参数含义

L: linestyle, 线型

下划线

名称	代号	数值代码
实线	SOLID_LINE	0
点线	DOTTEN_LINE	1
中心线	CENTER_LINE	2
虚线	DASHED_LINE	3
用户定义的类型	USERBIT_LINE	4

C语言图形函数及应用

线型设置函数参数含义

W: Width, 线宽

代号	数值代码	含义
NORM_WIDTH	0	一个像素宽
THICK_WIDTH	1	三个像素宽

C语言图形函数及应用

线型设置函数参数含义

M: Mode, 线型模式参数

注:

- 只有当L的值为**USERBIT_LINE**时才起作用
- 否则, 可取任意数值, 但不起作用

C语言图形函数及应用

线型设置函数参数含义

例句：设置宽度为3个像素的实线

```
setlinestyle(SOLID_LINE,0,THICK_WIDTH);
```

或

```
setlinestyle(0,0,1);
```


C语言图形函数及应用

直线函数

函数	画线否	当前位置移动否
<code>moveto(x,y);</code>	NO	YES
<code>lineto(x,y);</code>	YES	YES
<code>line(x1,y1,x2,y2);</code>	YES	NO
<code>moverel(dx,dy);</code>	NO	YES
<code>linereel(dx,dy);</code>	YES	YES

C语言图形函数及应用

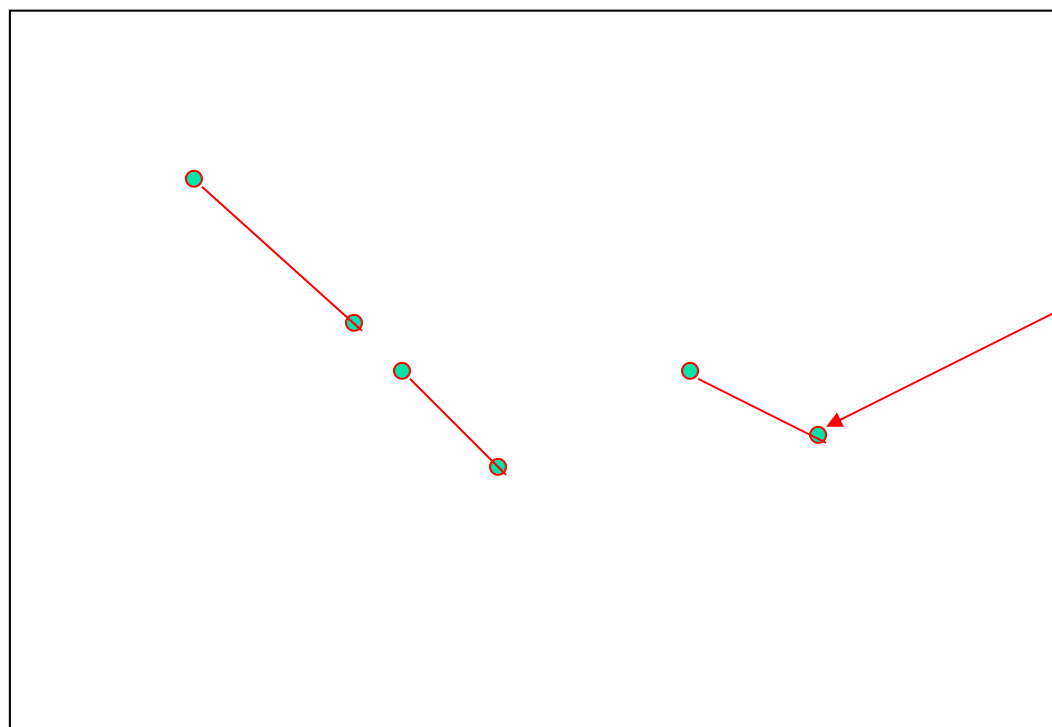
直线函数-示例

例1：由以下程序段，绘出图形结果并指出当前位置

```
...  
moveto(30,40);  
lineto(60,60);  
line(70,70,90,90);  
moverel(50,10);  
linerel(30,10);  
...
```

C语言图形函数及应用

直线函数-示例



当前位置: (140,80)

C语言图形函数及应用

直线函数-应用

用直线函数可以绘制任意参数曲线

$$\begin{cases} x = x(t) \\ y = y(t) \end{cases} \quad a \leq t \leq b$$

思路和方法

- 对 t 进行细分，即对区间 $[a,b]$ 进行划分；
- 依次连接前后两点，即用直线函数绘图。

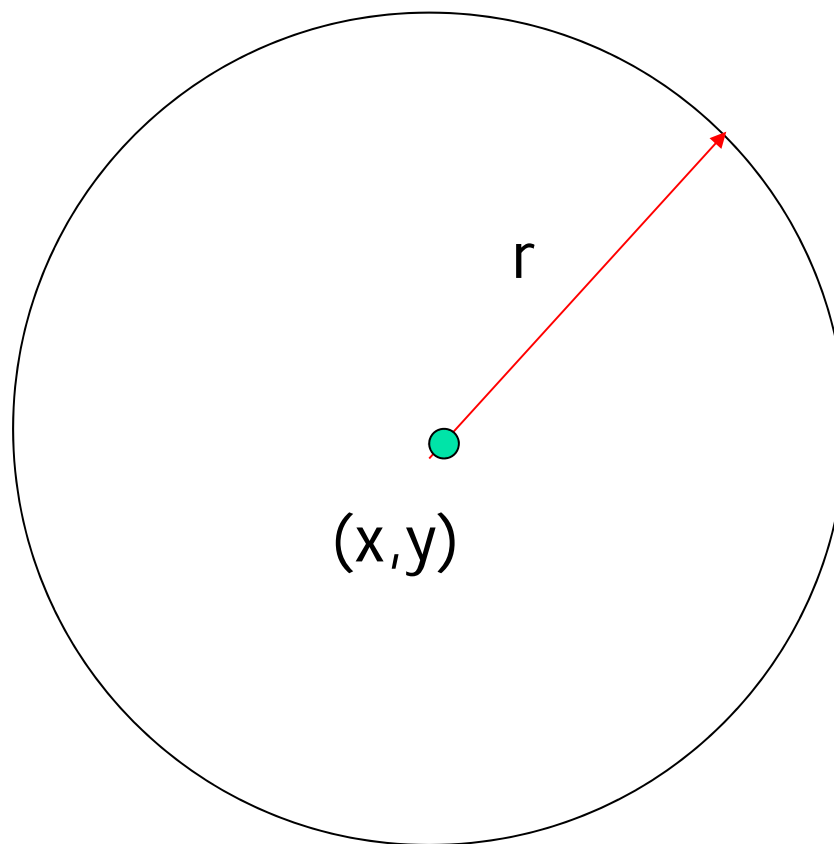
❖用折线代替曲线；

❖划分越细，效果越好。

C语言图形函数及应用

圆

`circle(x,y,r);`



C语言图形函数及应用

圆弧

$\text{arc}(x,y,\alpha1,\alpha2,r);$

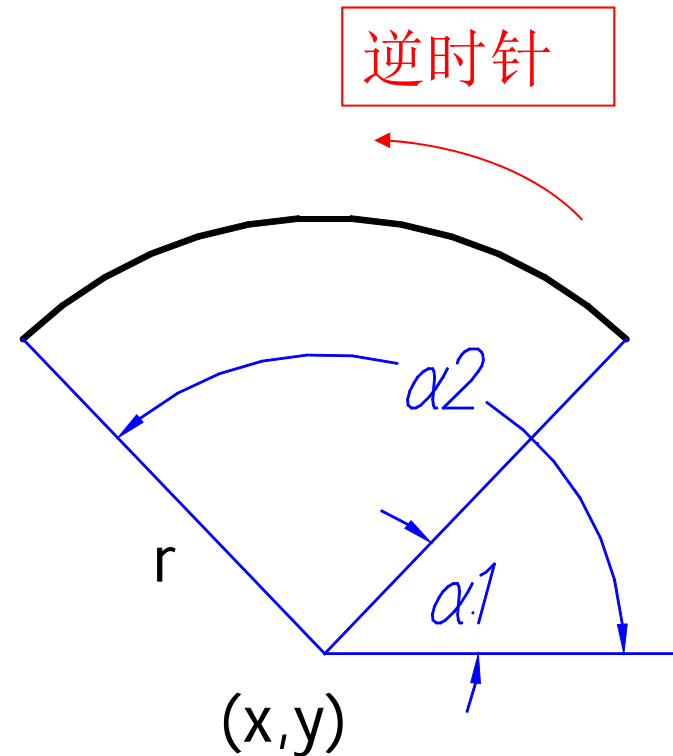
x,y : 圆心坐标;

$\alpha1$: 起始角

$\alpha2$: 终止角

度数

r : 半径



C语言图形函数及应用

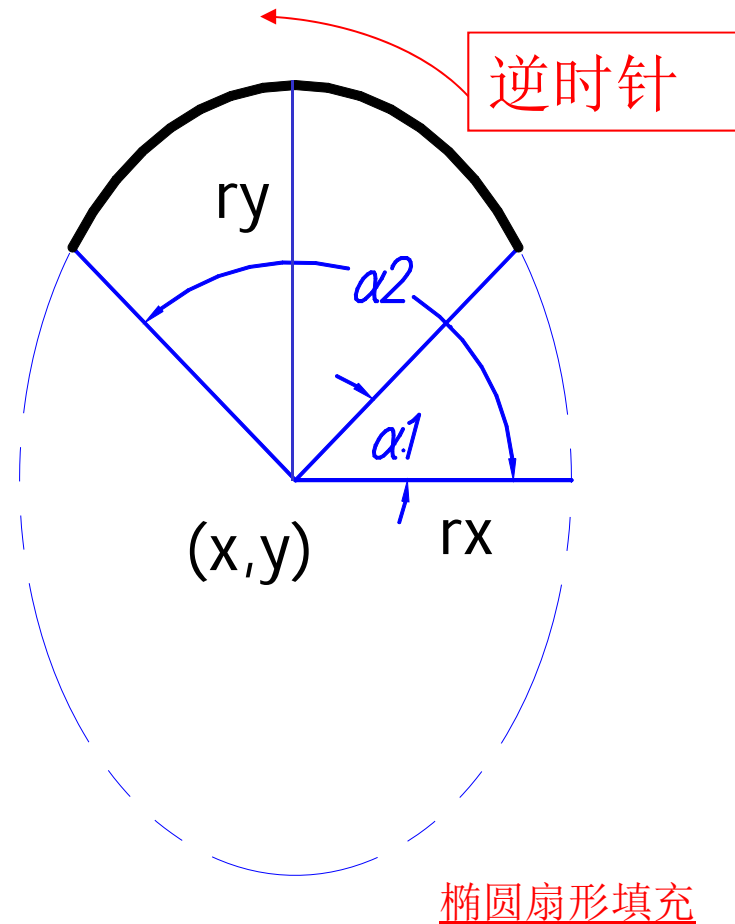
椭圆弧

`ellipse(x,y, $\alpha 1$, $\alpha 2$,rx,ry);`

$x,y,\alpha 1,\alpha 2$: 同圆弧

rx: x轴半径

ry: y轴半径

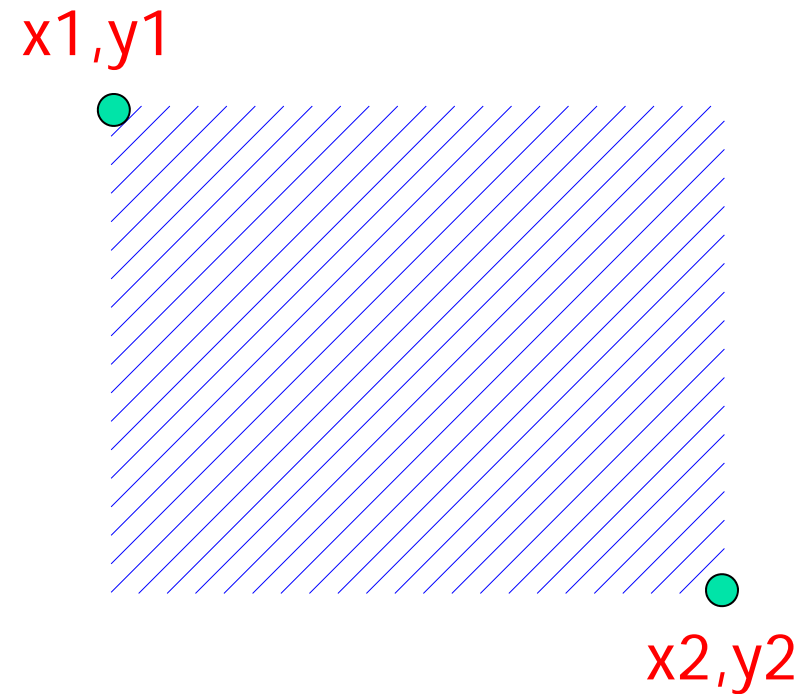


C语言图形函数及应用

无框线矩形

`bar(x1,y1,x2,y2);`

- 以当前的填充方式和颜色进行填充;
- 不画矩形边框

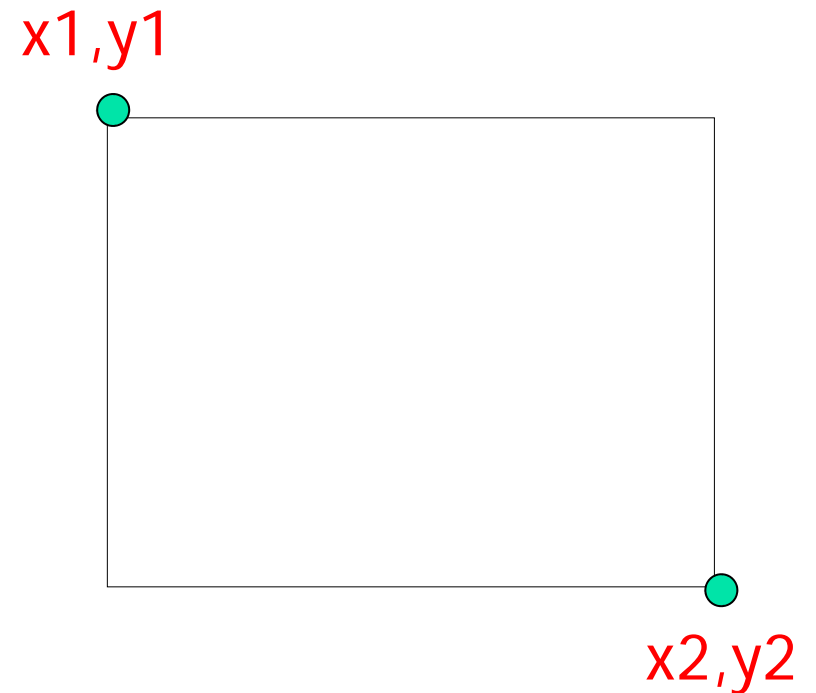


C语言图形函数及应用

矩形

`rectangle(x1,y1,x2,y2);`

- 画矩形边框



C语言图形函数及应用

长方体

```
bar3d(x1,y1,x2,y2,d,k);
```

作用

- 画长方体;
- 用当前的填充方式和颜色对正面进行填充

C语言图形函数及应用

长方体

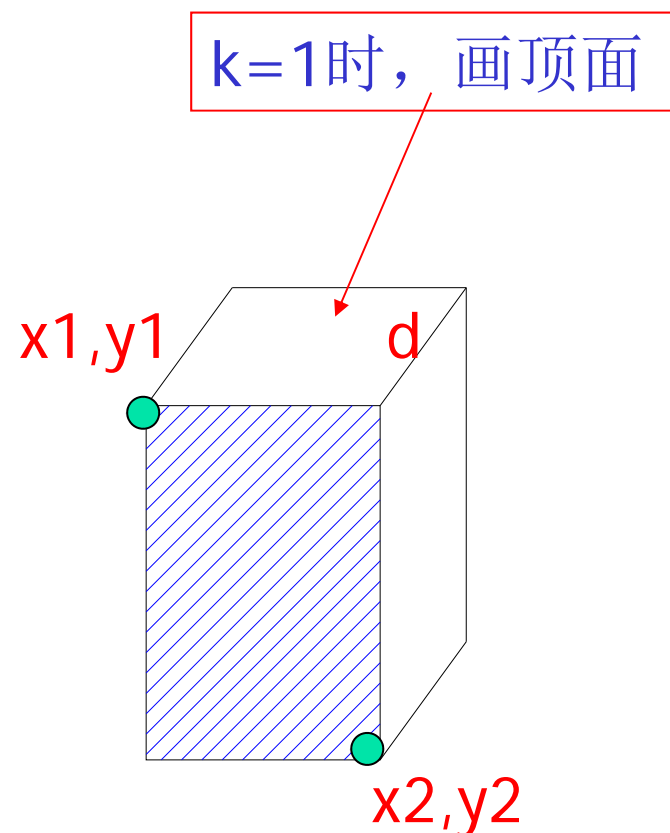
`bar3d(x1,y1,x2,y2,d,k);`

`x1,y1`:正面左上角坐标;

`x2,y2`:正面右下角坐标

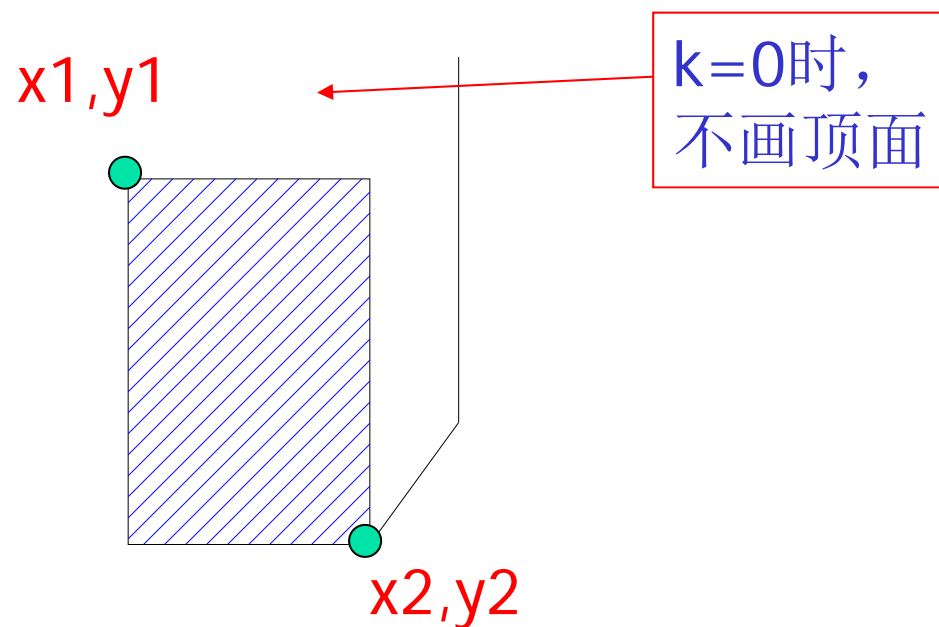
`d`:长方体的厚度

`k`:控制是否画顶面



C语言图形函数及应用

长方体



C语言图形函数及应用

多边形

`drawpoly(n+1,ps);`

参数含义

`n`: 多边形边数;

`ps`: 多边形顶点($n+1$ 个)构成的数组, 共有 $2(n+1)$ 个数。

C语言图形函数及应用

多边形-示例

例：已知六边形六个顶点的坐标，绘制六边形

```
#include "graphics.h"
#include "stdio.h"
main()
{
    int drive,mode;
    int ps[7*2]={140,60,150,70,170,70,180,60,170,50,150,50,140,60} ;
    drive=VGA;
    mode=VGAMED;
```

C语言图形函数及应用

多边形-示例

```
initgraph(&drive,&mode," ");  
cleardevice();  
setbkcolor(1);  
setcolor(4);  
    drawpoly(7,ps);  
getch();  
closegraph();  
}
```

C语言图形函数及应用

点

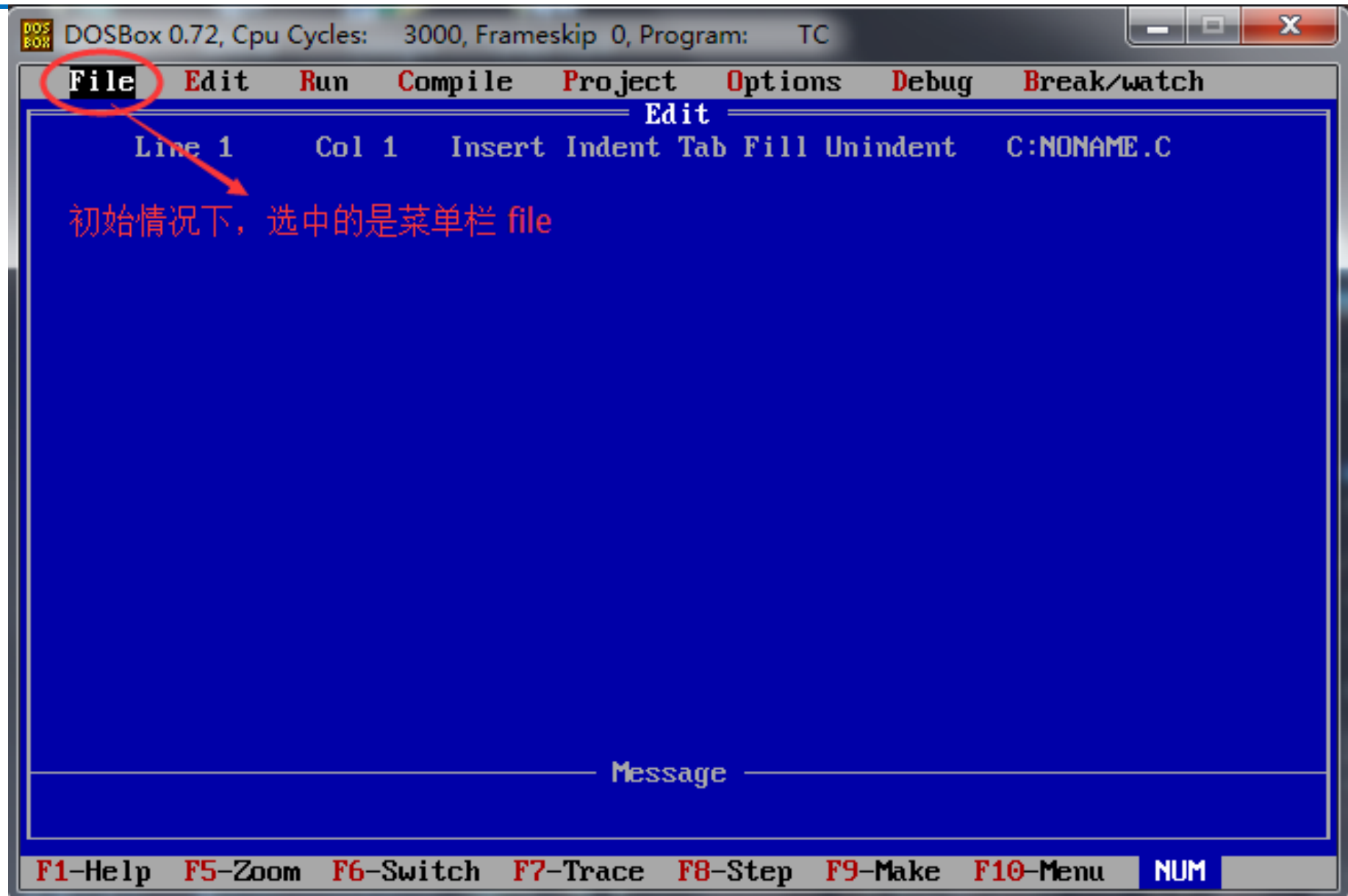
```
putpixel(x,y,c);
```

在点(x,y)处，以颜色c显示一个点

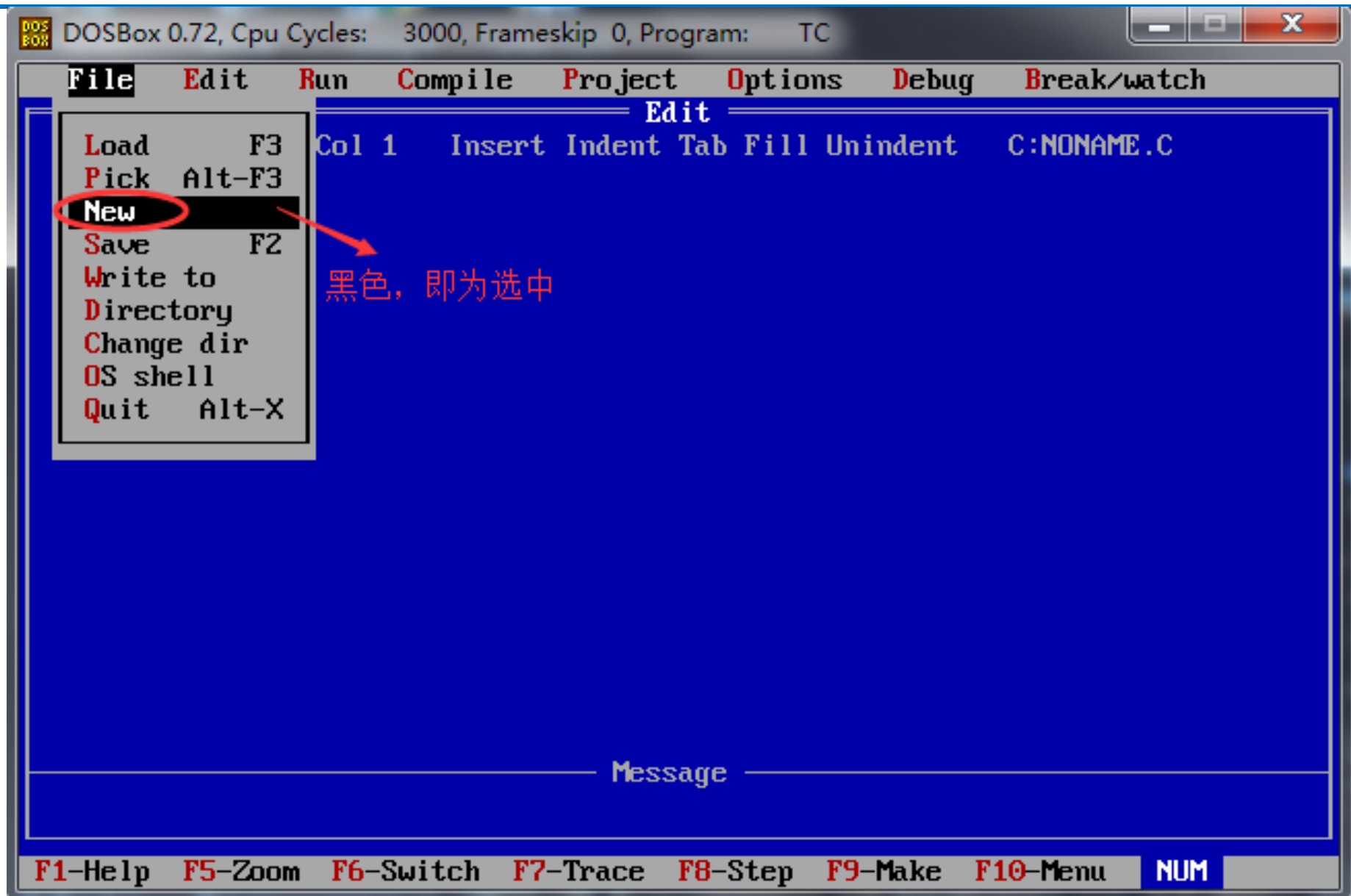
内容回顾

- C语言先修知识
- C语言图形屏幕设置
 - 初始化、屏幕颜色、屏幕坐标
 - 图形输出活动页、可见页
 - 关闭图形系统
- C语言图形函数及应用
 - 线型设置、直线函数
 - 圆、圆弧、椭圆弧；
 - 多边形、点；

TC2.0运行界面简介



TC2.0运行界面简介



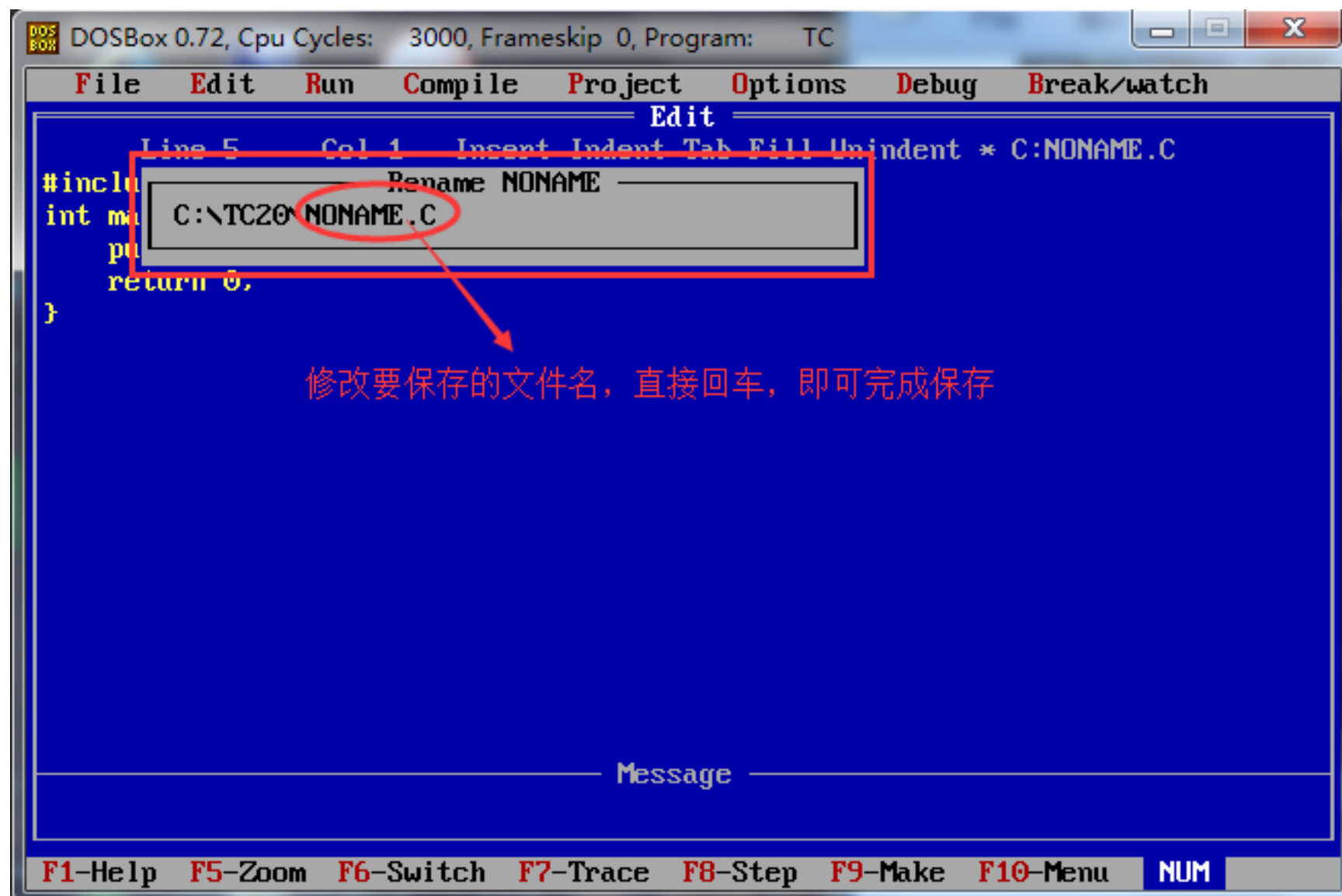
TC2.0运行界面简介



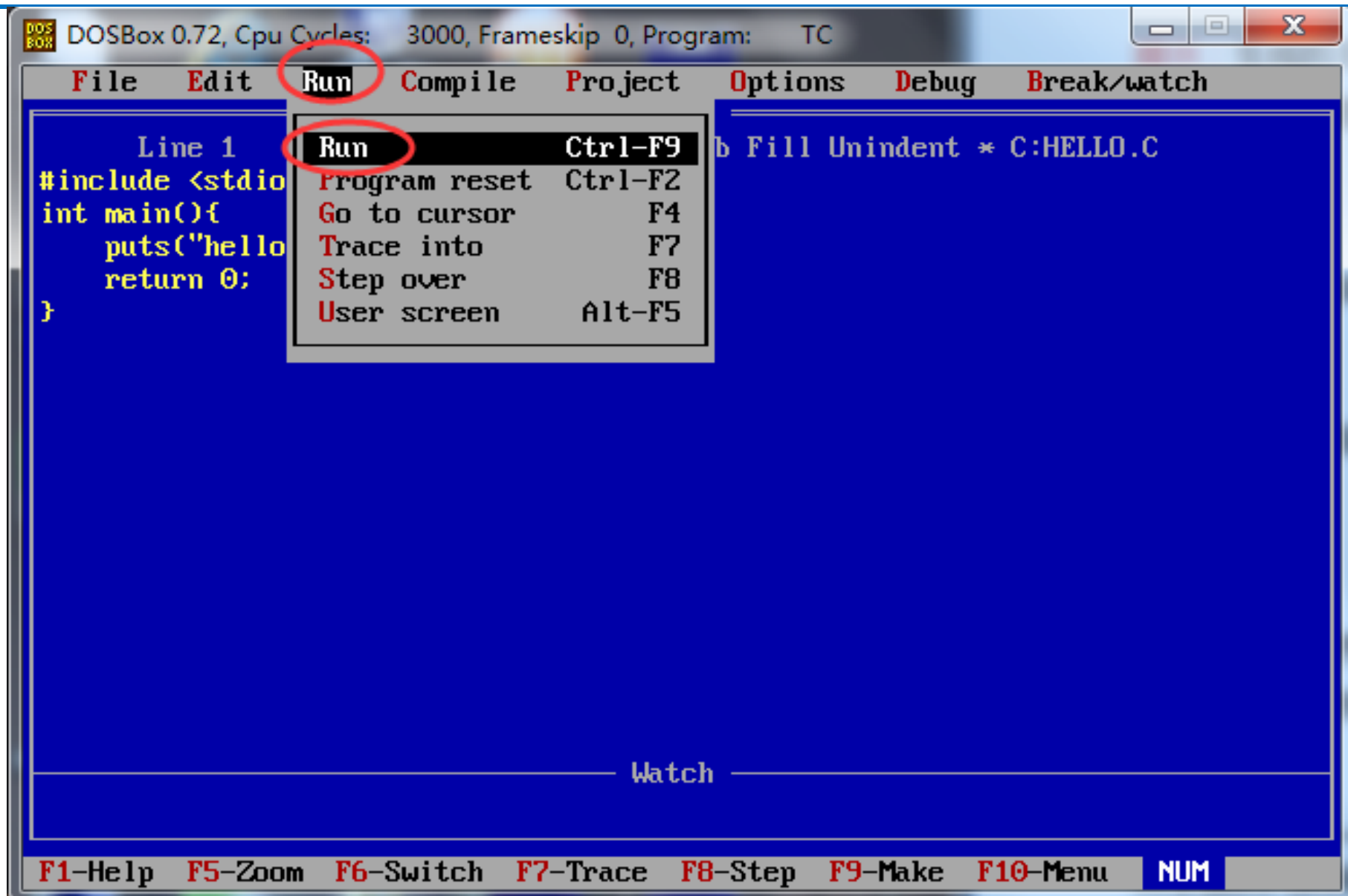
TC2.0运行界面简介



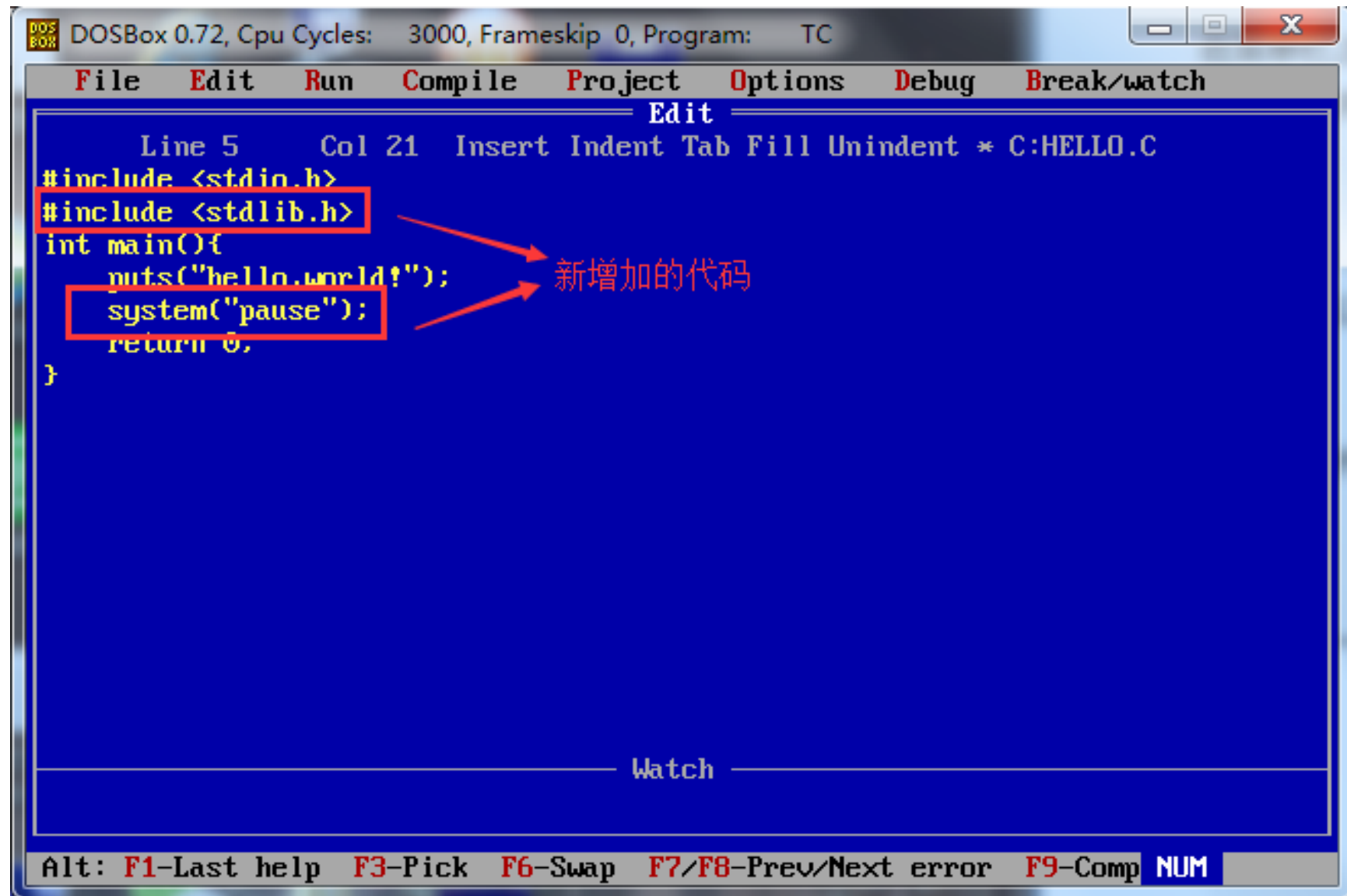
TC2.0运行界面简介



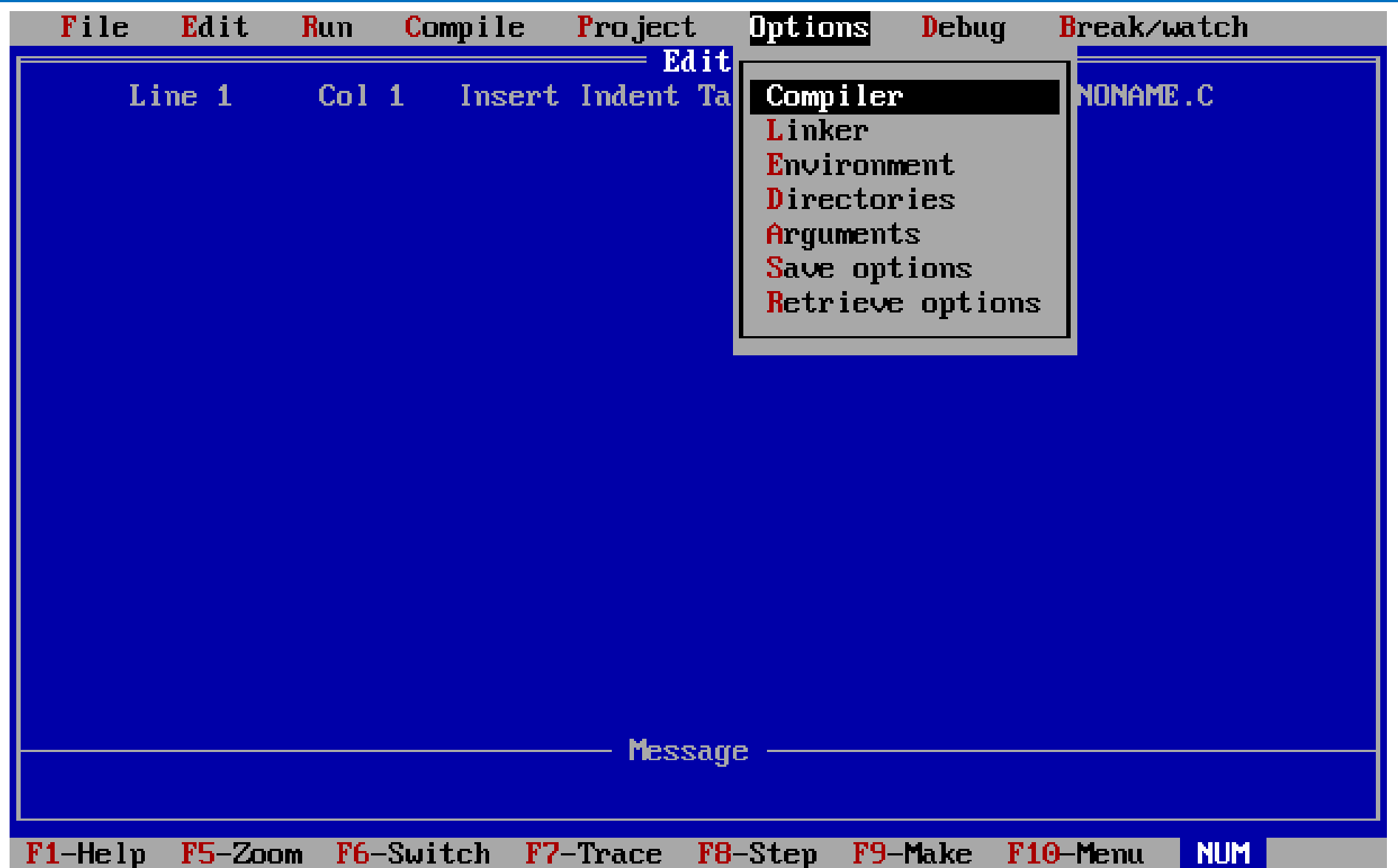
TC2.0运行界面简介



TC2.0运行界面简介



TC2.0运行界面简介



TC2.0运行界面简介

使用图形初始化函数设置VGA高分辨率图形模式:

```
#include  
int main()  
{  
    int gdriver, gmode;  
    gdriver=VGA;  
    gmode=VGAHI;  
    initgraph(&gdriver, &gmode, "c://caic//bgi");  
    bar3d(100, 100, 300, 250, 50, 1); /*画一长方体*/  
    getch();  
    closegraph();  
    return 0;  
}
```

TC2.0运行界面简介

void far detectgraph(int *gdriver, *gmode); 其中gdriver和gmode的意义与上面相同

自动进行硬件测试后进行图形初始化:

```
#include
int main()
{
int gdriver, gmode;
detectgraph(&gdriver, &gmode); /*自动测试硬件*/
printf("the graphics driver is %d, mode is %d/n",
gdriver,gmode); /*输出测试结果*/
getch();
initgraph(&gdriver, &gmode, "c://caic//bgi");
/* 根据测试结果初始化图形*/
bar3d(10, 10, 130, 250, 20, 1);
getch();
closegraph();
return 0;
}
```

gdriver=DETECT 语句后再跟initgraph()函数就行
采用这种方法后, 上例可改为:

```
#include
int main()
{
int gdriver=DETECT, gmode;
initgraph(&gdriver, &gmode, "c://caic//bgi");
bar3d(50, 50, 150, 30, 50, 1);
getch();
closegraph();
return 0;
}
```

上机作业

作业：用C语言编程绘制以下5条曲线

要求：

■ 包含：

- ✓ 题目
- ✓ 程序
- ✓ 运行结果
- ✓ 总结（心得）

$$\begin{cases} x = a \sin 3t \cos t \\ y = a \cos 3t \sin t \end{cases} \quad (0 \leq t \leq 2\pi)$$

$$r = a\theta \quad (a > 0) \quad (\text{阿基米德螺旋线})$$

$$r = a(1 - \cos\theta) \quad (a > 0) \quad (\text{心形线})$$

$$\begin{cases} x = a(t - \sin t) \\ y = a(1 - \cos t) \end{cases} \quad (\text{摆线, } t \text{ 单位为弧度})$$

$$\begin{cases} x = a(\cos t + t \sin t) \\ y = a(\sin t - t \cos t) \end{cases} \quad (\text{圆的渐伸线})$$



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



问题？
谢谢！

