

实验八 图形处理程序设计

一、实验目的

1. 了解文本方式和图形方式处理图形的方法。
2. 掌握应用 BIOS 显示功能进行彩色图形处理的方式。
3. 正确编写图形处理程序。

二、实验内容

1. 复习教材中有关图形处理的程序设计内容。
2. 阅读程序 1。

```
DATA    SEGMENT
A        DB 'How do you do?',0AH,0DH
          DB '$'
B        DB 20,?,20 DUP(?)
DATA    ENDS
CODE    SEGMENT
          ASSUME  CS:CODE,DS:DATA
START:  MOV     AX,DATA
          MOV     DS,AX
          MOV     AH,09H
          LEA     DX,A
          INT     21H
          LEA     DX,B
          MOV     AH,0AH
          INT     21H
          MOV     DL,0AH
          MOV     AH,02H
          INT     21H
          MOV     AL,B+1
          MOV     AH,0
          MOV     SI,AX
          MOV     DX,OFFSET B+2
          MOV     BX,DX
          MOV     BYTE PTR [BX+SI+1], '$'
          MOV     AH,09H
          INT     21H
          MOV     AH,4CH
```

```
        INT    21H
CODE    ENDS
        END    START
```

3. 阅读程序 2

```
CODE    SEGMENT
        ASSUME CS:CODE
START:  MOV    AX,0002H
        INT    10H
        MOV    AH,02H
        MOV    BH,00H
        MOV    DX,0524H
        INT    10H
        MOV    AH,09H
        MOV    BH,00H
        MOV    BL,2CH
        MOV    AL,'G'
        MOV    CX,8
        INT    10H
        MOV    AH,02H
        MOV    BH,00H
        MOV    DX,0A24H
        INT    10H
        MOV    AH,09H
        MOV    BH,00H
        MOV    BL,1EH
        MOV    AL,'O'
        MOV    CX,8
        INT    10H
        MOV    AH,02H
        MOV    BH,00H
        MOV    DX,0F24H
        INT    10H
        MOV    AH,09H
        MOV    BH,00H
        MOV    BL,4FH
        MOV    AL,'O'
        MOV    CX,8
```

```
        INT    10H
        MOV    AH, 02H
        MOV    BH, 00H
        MOV    DX, 1424H
        INT    10H
        MOV    AH, 09H
        MOV    BH, 00H
        MOV    BL, 02H
        MOV    AL, 'D'
        MOV    CX, 8
        INT    10H
        MOV    AH, 4CH
        INT    21H
CODE    ENDS
        END    START
```

4. 阅读程序 3

```
WINWIDTH = 40
WINTOP   = 8
WINLEFT  = 20
WINBOTTOM = 17
WINRIGHT = WINLEFT+WINWIDTH-1
COLOR    = 74H
PAGEN    = 0
CTRL_C   = 03H
CODE     SEGMENT
        ASSUME CS:CODE
START:   MOV AL, PAGEN
        MOV AH, 5
        INT 10H
        ;
        MOV CH, WINTOP
        MOV CL, WINLEFT
        MOV DH, WINBOTTOM
        MOV DL, WINRIGHT
        MOV BH, COLOR
        MOV AL, 0
        MOV AH, 6
```

```
        INT 10H
        ;
        MOV BH, PAGEN
        MOVDH, WINBOTTOM
        MOVDL, WINLEFT
        MOV BH, COLOR
        MOVAH, 2
        INT 10H
        ;
NEXT:    MOVAH, 0
        INT 16H
        CMP AL, CTRL_C
        JE  EXIT
        ;
        MOV BH, PAGEN
        MOV CX, 1
        MOVAH, 0AH
        INT 10H
        ;
        INCDL
        CMP DL, WINRIGHT+1
        JNE SETCUR
        MOV CH, WINTOP
        MOV CL, WINLEFT
        MOVDH, WINBOTTOM
        MOVDL, WINRIGHT
        MOV BH, COLOR
        MOVAL, 1
        MOVAH, 6
        INT 10H
        MOVDL, WINLEFT
SETCUR:  MOV BH, PAGEN
        MOVAH, 02H
        INT 10H
        JMP NEXT
EXIT:    MOV     AH, 4CH
        INT 21H
```

CODE	ENDS
	ENDSTART

5. 阅读程序 4

DATA	SEGMENT
ROW	DW 0
COL	DW 0
COLOR	DB 1
DATA	ENDS
CODE	SEGMENT
	ASSUME CS:CODE,DS:DATA
START:	MOV AX, DATA
	MOV DS, AX
	MOV AH, 15
	INT 10H
	PUSH AX
	MOV AX, 0004H
	INT 10H
	MOV AH, 0BH
	MOV BH, 01H
	MOV BL, 00H
	INT 10H
	MOV CX, 3
@1:	PUSH AX
	PUSH CX
	PUSH DX
	PUSH SI
	PUSH DI
	MOV AH, 0CH
	MOV AL, COLOR
	MOV DI, 100
	MOV DX, ROW
@@1:	MOV SI, 20
	MOV CX, COL
@@2:	INT 10H
	INC CX
	DEC SI
	JNZ @@2

```
        INC    DX
        DEC    DI
        JNZ    @@1
        POP    DI
        POP    SI
        POP    DX
        POP    CX
        POP    AX
        mov    ah,01h
        int    21h

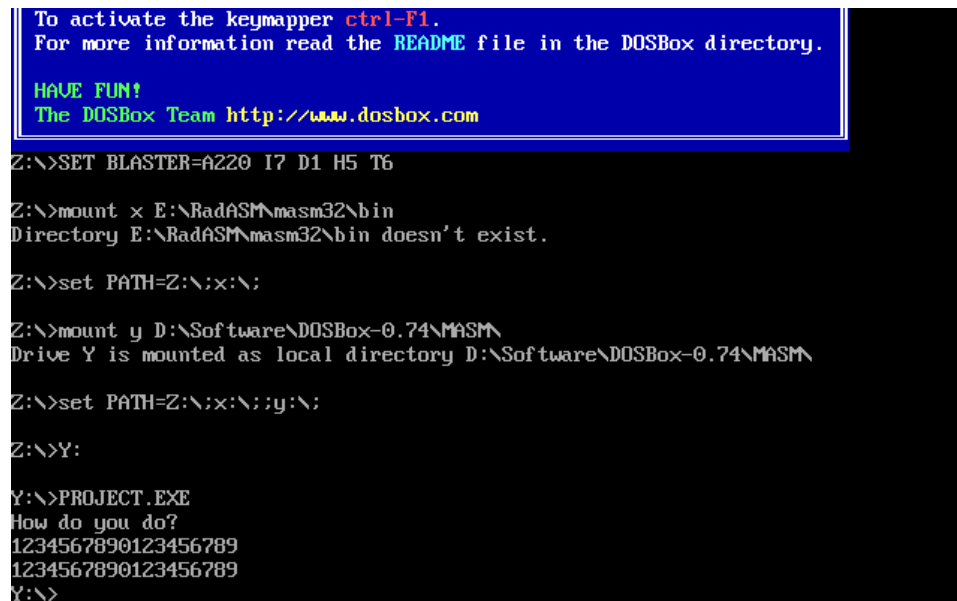
        INC    COLOR
        ADD    COL,    40
        LOOP   @1
        POP    AX
        MOV    AH, 0H
        INT    10H
        MOV    AH, 4CH
        INT    21H
CODE    ENDS
        END    START
```

6. 编写程序 5，在屏幕中部画一条垂直的黄色垂线。

三、实验结果（截图）

1. 程序 1

输出 “How do you do?” 后读入用户输入，再重新输出到命令行。



```
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount x E:\RadASM\masm32\bin
Directory E:\RadASM\masm32\bin doesn't exist.

Z:\>set PATH=Z:\x:\;

Z:\>mount y D:\Software\DOSBox-0.74\MASM\
Drive Y is mounted as local directory D:\Software\DOSBox-0.74\MASM\

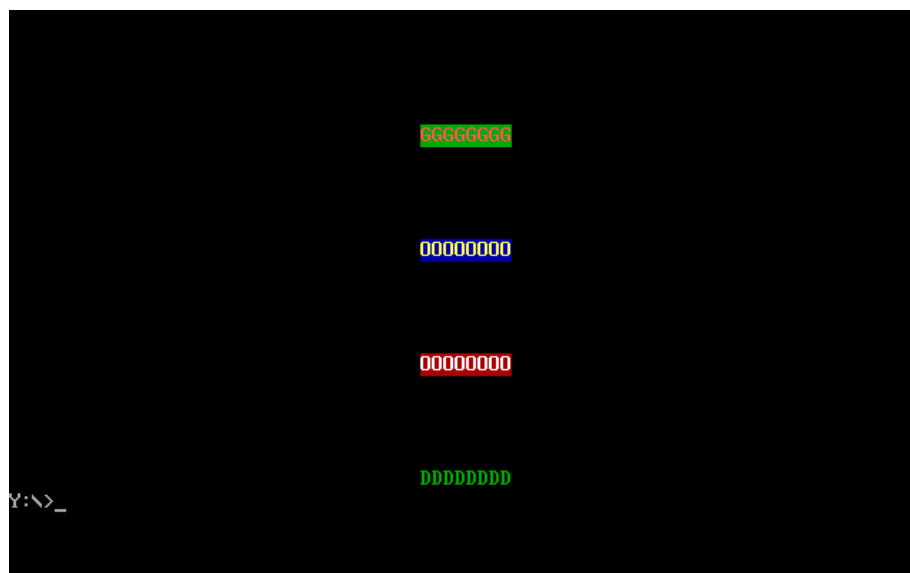
Z:\>set PATH=Z:\x\::y:\;

Z:\>Y:

Y:\>PROJECT.EXE
How do you do?
1234567890123456789
1234567890123456789
Y:\>
```

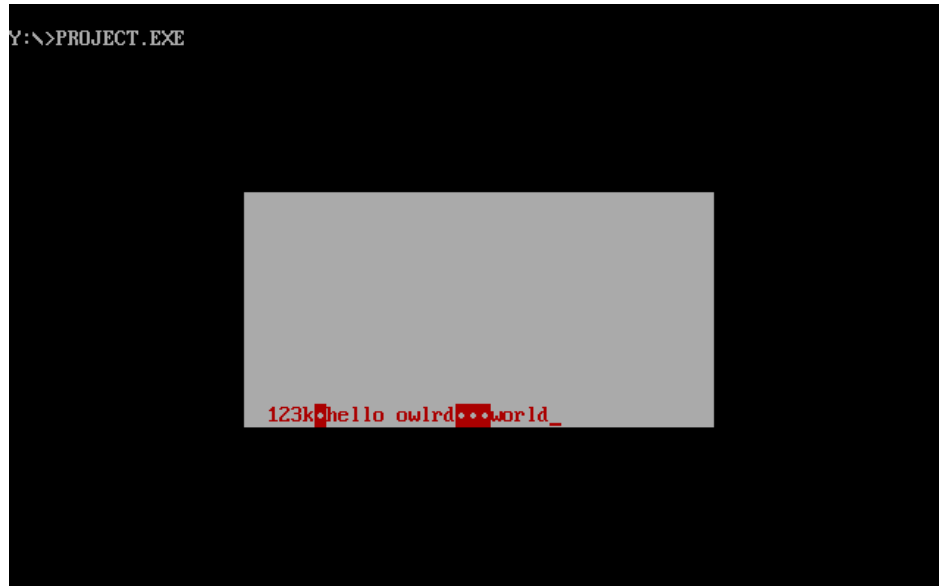
2. 程序 2

以分辨率 80*25 的 16 色文本显示模式显示了一些字符，也进行了一些光标控制操作。



3. 程序 3

此程序在命令行中间创建了一个前景色为红色、背景色为银色的窗口，保持在最下面一行输入，输入的内容不断上卷。输入 `ctrl+C` 退出程序。



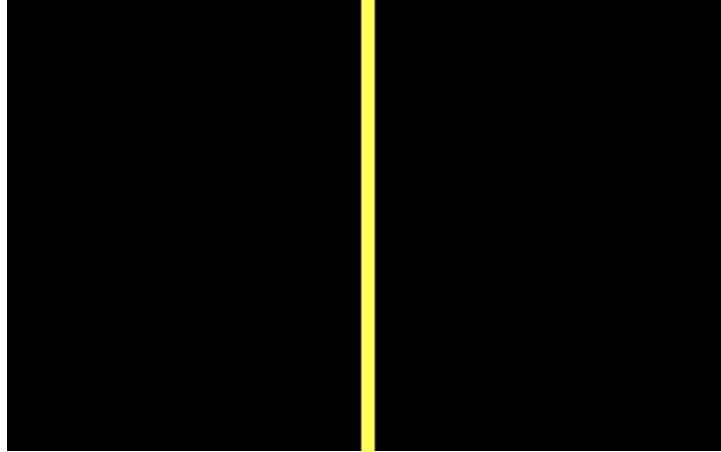
4. 程序 4

此程序循环输出了三个不同颜色的纵向颜色条带，每条宽为 20 像素、高为 100 行，间隔 40 像素。由于代码中使用的是有回显的键盘输入，因此会将一开始的一些像素替换为输入的字符，有黑色空缺或替换成了一些字符。



4. 程序 5

使用的是 320x200 分辨率的 256 色 VGA 显示模式，利用两层嵌套循环完成有宽度竖线的显示，相当于显示一个矩形。



四、实验报告要求（习题）

1. 程序 1

这段程序先输出字符串 “How do you do? ”，然后等待用户输入一个不超过 20 字符长度的字符串（包含回车），最后再将用户输入的字符串输出一遍到命令行。

```
DATA    SEGMENT
    A    DB  'How do you do?',0AH,0DH
          DB  '$'
    ; B 作为输入缓冲区，最多输入 20 个字符
    B    DB  20,?,20 DUP(?)
DATA    ENDS
CODE    SEGMENT
    ASSUME CS:CODE,DS:DATA
START:
    ; 设置段寄存器
    MOV AX, DATA
    MOV DS, AX
    ; 显示字符串 A
    MOV AH, 09H
    LEA DX, A
    INT 21H
    ; 键盘输入到缓冲区，需要输入回车键结束输入
    ; 也就是说实际能够看到的输入只能 19 个字符
```

```
LEA DX, B
MOV AH, 0AH
INT 21H
; 输出个换行符将输入和输出分开到两行
MOV DL, 0AH
MOV AH, 02H
INT 21H
; 获得实际输入的字符串长度
MOV AL, B+1
MOV AH, 0
; 输入字符串实际长度放在 SI 中
MOV SI, AX
; 输入字符串的开始位置存在 DX 中
MOV DX, OFFSET B+2
MOV BX, DX
; 在输入字符串的末尾加上字符串结束标记 '$'
MOV BYTE PTR [BX+SI+1], '$'
; 调用 21H 中断显示 DS:DX 处的字符串
MOV AH, 09H
INT 21H
; 退出代码
MOV AH, 4CH
INT 21H
CODE    ENDS
END START
```

2. 程序 2——计算和式直到 S 超过 200

相当于如下的高级语言代码

```
sum = 1, n = 2, ax = 0;
while (sum <= 200) {
    ax = n*n + n;
    sum += ax;
    n++;
}
```

DATA SEGMENT

N DW 2

SUM DW 1

DATA ENDS

STACK SEGMENT STACK

DW 128 DUP(0)

STACK ENDS

CODE SEGMENT

ASSUME cs:CODE, ds:DATA, ss:STACK

START:

mov ax, DATA

mov ds, ax

; 初始化变量

mov SUM, 1

mov N, 2

mov ax, 0

AGAIN:

; while(sum <= 200)

cmp sum, 200

jg EXIT

; 减少指令，将 $n*(n+1)$ 转换为 n^2+n

; n^2+n 存在 ax 中

; $ax = n*n+n$

mov ax, N

imul ax

add ax, N

; 往 sum 上加

; $sum += ax$

add SUM, ax

; $n++$

inc N

jmp AGAIN

EXIT: ; 退出代码

mov ax, 4C00H

int 21H

CODE ENDS

END START

3. 程序 3

通过 int 10H、AH=6 的 BIOS 功能调用初始化和卷动窗口，实现在中间的小窗口进行键盘字符的键入与显示。

```
; 一些宏定义
WINWIDTH = 40
WINTOP = 8
WINLEFT = 20
WINBOTTOM = 17
WINRIGHT= WINLEFT+WINWIDTH-1
COLOR = 74H
PAGEN = 0
CTRL_C = 03H

CODE    SEGMENT
    ASSUME CS:CODE
START:
    ; 置当前显示页为 pageN
    MOV AL, PAGEN
    MOV AH, 5
    INT 10H
    ; 初始化窗口
    ; 左上角为 wintop 行 winleft 列，左上角为 winbottom 行 winright 列
    ; 行属性为 color（即设置前景色与背景色）
    MOV CH, WINTOP
    MOV CL, WINLEFT
    MOV DH, WINBOTTOM
    MOV DL, WINRIGHT
    MOV BH, COLOR
    MOV AL, 0
    MOV AH, 6
    INT 10H
    ; 置光标位置于当前页窗口的左下角
    MOV BH, PAGEN
    MOV DH, WINBOTTOM
    MOV DL, WINLEFT
```

```
MOV BH, COLOR
MOV AH, 2
INT 10H
NEXT:
    ; 从键盘读入字符，若为 ctrl+c，跳转到退出代码
MOV AH, 0
INT 16H
CMP AL, CTRL_C
JE EXIT
    ; 若输入的字符不为 ctrl+c，显示一次输入的字符
MOV BH, PAGEN
MOV CX, 1
MOV AH, 0AH
INT 10H
    ; 准备向下一列移动光标
INC DL
    ; 若右侧要超出窗口右边界，向上卷动一行并将光标置于最左侧
CMP DL, WINRIGHT+1
JNE SETCUR
MOV CH, WINTOP
MOV CL, WINLEFT
MOV DH, WINBOTTOM
MOV DL, WINRIGHT
MOV BH, COLOR
MOV AL, 1
MOV AH, 6
INT 10H
MOV DL, WINLEFT
SETCUR:    ; 置光标位置到最新位置
MOV BH, PAGEN
MOV AH, 02H
INT 10H
    ; 继续输入
JMP NEXT
EXIT:    ; 退出代码
MOV AH, 4CH
INT 21H
```

```
CODE ENDS  
END START
```

4. 程序 4

```
DATA    SEGMENT  
    ROW DW  0  
    COL DW  0  
    COLOR DB  1  
DATA    ENDS  
CODE    SEGMENT  
    ASSUME CS:CODE,DS:DATA  
START:  
    ; 设置段寄存器  
    MOV AX, DATA  
    MOV DS, AX  
    ; 取当前页号、字符列数、显示方式  
    MOV AH, 15  
    INT 10H  
    ; 将原显示方式信息保护  
    PUSH AX  
    ; 设置显示模式为 4 色图形（分辨率为 320x200）  
    MOV AX, 0004H  
    INT 10H  
    ; 设置调色板 ID  
    MOV AH, 0BH  
    MOV BH, 01H  
    MOV BL, 00H  
    INT 10H  
    ; 总共要画三种颜色  
    MOV CX, 3  
@1:  
    ; 保护寄存器  
    PUSH  AX  
    PUSH  CX  
    PUSH  DX  
    PUSH  SI  
    PUSH  DI  
    ; 准备写像素，AL 为颜色值
```

```
    ; 写的位置在 DX 行、CX 列
    MOV AH, 0CH
    MOV AL, COLOR
    MOV DI, 100
    MOV DX, ROW
@@1:
    MOV SI, 20
    MOV CX, COL
@@2:
    INT 10H
    ; 写一行宽为 20 的像素
    INC CX
    DEC SI
    JNZ @@2
    ; 反复写 100 行
    INC DX
    DEC DI
    JNZ @@1
    ; 恢复寄存器
    POP DI
    POP SI
    POP DX
    POP CX
    POP AX
    ; 等待用户输入
    mov ah, 01h
    int 21h
    ; 用户有输入后，隔 40 列，画下一种颜色，总共三种
    INC COLOR
    ADD COL, 40
    LOOP @1
    ; 恢复出原来的显示方式并将显示方式还原
    POP AX
    MOV AH, 0H
    INT 10H
    ; 退出代码
    MOV AH, 4CH
```

```
INT 21H
CODE    ENDS
END START
```

5. 程序 5

```
; 宏定义绘制的行列起点和终点，以及颜色
COL_START = 155
COL_END   = 165
ROW_START = 0
ROW_END   = 200
COLOR     = 1110b

DATA SEGMENT
    ROW DW 0
    COL DW 0
DATA ENDS

STACK SEGMENT STACK
    DW 128 DUP(0)
STACK ENDS

CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
START:
    ; 设置段寄存器
    MOV AX, DATA
    MOV DS, AX
    ; 获取当前显示模式并进行保护
    mov ah, 0FH
    int 10H
    push ax
    push bx
    ; 设置显示模式 256 色图形（分辨率 320x200）
    mov ax, 0013H
    INT 10H
    ; 初始化变量
```



```
    mov ROW, ROW_START
    mov COL, COL_START
    mov cx, ROW_END-ROW_START
```

DRAW_ROW:

```
    mov COL, COL_START
    push ax
    push bx
    push cx
    push dx
```

DRAW_LINE:

```
    cmp COL, COL_END
    jae NEXT
    ; 绘制点
    mov dx, ROW
    mov cx, COL
    mov al, COLOR
    mov ah, 0CH
    mov bh, 0
    int 10H
    inc COL
    jmp DRAW_LINE
```

NEXT:

```
    pop dx
    pop cx
    pop bx
    pop ax
    inc ROW
    loop DRAW_ROW
    ; 等待用户输入
    mov ah, 01h
    int 21h
```

EXIT: ; 退出代码

```
    ; 还原最初的显示模式
    pop bx
    pop ax
    mov ah, 0
    int 10H
```

```
mov ah, 4CH  
int 21H  
CODE ENDS  
END START
```

6. 遇到的问题及解决

书上附录中的显示方式表上有几个和 DOSBox 中的不太一致，经过测试后在程序 5 中选择使用 256 色 VGA 显示方式。

五、个人体会与总结

本次实验，主要是进行各种显示方式的尝试。DOS 下总的来分，有两大显示方式，一种是文本显示、另一种是图形显示。需要注意的是，调用 BIOS 的 10H 中断绘制图像，颜色来源于内置的颜色索引表，使用时选择颜色的编号而不是输入 RGB 值。事实上，这个颜色索引表通过 3C8H 与 3C9H 端口可以进行改变，即修改显存中颜色索引表的内容。