**PRACTICAL # 11**

**OBJECT:**

Graphics in assembly.

**THEORY:**

Working with graphics is a fun. In assembly we can directly read/write to some of the graphics devices. However to make things easier some of the prebuilt interrupts provide many useful functions to play with the graphics.

The interrupt 10h provides some basic graphics functionalities. Services include setting the video mode, character and string output, and graphics primitives (reading and writing pixels in graphics mode).

To use this call, load AH with the number of the desired subfunction, load other required parameters in other registers, and make the call.

The function number 0 initializes and sets the video mode. The video mode to be set is specified in AL register.

**Program:**

.model small

.stack 100h

.data

xPoint dw 10 ; x coordinate of line start

yPoint dw 50 ; y coordinate of line start

length dw 25 ; length of line

.code

main proc

mov ax, @data

mov ds, ax

; set the video mode 320x200, 256 colors

mov al, 13h

mov ah, 0

int 10h

; to draw a pixel, set color in al

mov al, 45

; P(x, y) x in cx, y in dx

mov cx, xPoint

mov dx, yPoint

; set function value in ah to draw a pixel

mov ah, 0ch ; 0ch is function or interrupt number to draw a pixel

; loop from xStart to (xStart+length) to draw a horizontal line

LoopStart:

; invoke the graphics interrupt

int 10h ; draws a pixel at x,y

inc cx ; x coordinate is incremented

; Test to see if x coordinate has reached terminal value

cmp cx, length

; continue loop if cx <= length

jle LoopStart

mov ah, 4ch

int 21h

main endp

end main

After initializing the video mode, a pixel drawing color is set in the AL register. Drawing a point requires x and y coordinates to be set in CX and DX registers respectively. To actually draw a pixel, we need to set the pixel drawing function value in the AH register which is 0CH and color of the pixel is set in AL register. Calling int 10H in loop and changing the coordinate values draws a line using individual pixels.

**ACTIVITIES**

**Activity 1**

Draw a diagonal line using the video interrupt.

**Activity 2**

Draw a dotted line of the pattern,

. . . . . . .

Hint: draw alternate pixels, skipping the other alternate or draw only even pixels leaving odd ones undrawn.

**Activity 3**

Draw underscore and dotted line of the pattern,

. \_ . \_

Hint: for drawing \_ draw two or three consecutive pixels to form \_ like shape.

**REVIEW QUESTIONS**

1. What is a pixel?
2. Which interrupt is used for graphics in assembly?
3. Which interrupt number is used to draw a single pixel?
4. What is RGB value?
5. Which register is used to set pixel color?