Graphics mode

Mode number **CGA Graphics**

4 320 \* 200 4 color

5 320 \* 200 4 color

6 640 \* 200 2 color

**EGA Graphics**

D 320 \* 200 16 Color

E 640 \* 200 16 color

F 640 350 Monocrome

10 640\* 350 16 color

**VGA Color**

11 640 \* 480 2 color

12 640 \* 480 16 color

13 320\* 200 256 color

**Set screen Mode**

Mov ah,0 0 function

Mov al, mode number

Int 10h

**Set background color**

Mov ah, OBH ; OBH is background function

Mov Bh,0 ; mode

Movbl , color number (0-15)

**Write graphics pixel**

Mov ah 0CH ; 0CH function is used to write pixel on screen

Moval , pixel color value ; 0-15

Mov cx, column number

Mov , dx, row number

**Read graphics pixel**

Mov ah 0DH ; 0DH function is used to Read pixel on screen

Moval , pixel color value ; 0-15

Mov cx, column number

Mov , dx, row number

**Program#1 : Create a program to print a line on screen**

.model small

.stack 100h

.code

mainproc

;set graphic mode

Mov ax,6

int 10h ; int 10h used for screen manipulation

;draw line

Mov ah,0ch

Mov al,1

Mov cx,301

mov dx,100

L1:

int 10h

inc cx

cmp cx,600

jle l1

; read keyboard

Mov ah,0

int 16h ; int 16h will communicate with keyboard

; set text mode

Mov ax,3

int 10h

return dos

Mov ah,4ch

int 21h

mainendp

end main

**Program 2: Create a program to print a character in color on screen**

.model small

.stack 100h

.code

start:

Mov ah, 0 ; set mode

mov al,04h ; mode 4

int 10h

mov ah,0bh ; function 0BH for background

mov bh,00h ; select background color

mov bl,3 ; blue

int 10h

mov ah,02 ; set cursor

mov bh,0 ; page 0

mov dh,0 ; row 0

mov dl,30 col 39

int 10h

mov ah,09 ; write character function

moval,'A'

mov bl,2 ; red color

mov cx,1 ; write 1 character

int 10h

mov ah,4ch

int 21h

end start

**program #3 create a program which display 256 color**

.model small

.stack 100h

.code

start:

; set mode

mov ah,0

mov al, 13h

int 10h

; display 256 pixel in row 100

mov ah ,0ch ; write pixel function

mov al, 0 ; start with pixel color 0

mov bh, 0 ; page 0

mov cx, 0 ; column 0

mov dx, 100h ; row 100

l1:

int 10h ; write pixel

inc al ; next color

inc cx ; next column

cmp cx, 256 ; finished?

jl l1 ; no repeat

mov ah, 4ch

int 21h

end start

**;TITLE: "Program to clear screen"**

.model small ; small memory model

.stack 100h ; 256 Byte memory is allocated for stack memory

.code ; start of code memory

main proc

mov ah,06h ;window scroll up

mov al,0 ;scroll entire window

mov ch,00 ;upper left row window

mov cl,00 ;upper left column window

mov dh,24 ;lower right row window

mov dl,79 ;lower right column window

mov bh,07h ;normal video attribute (bh=attribute for blank lines)

int 10h ;clear screen

mov ah,4ch

int 21h

main endp

end main

;**TITLE: "Program to set cursor size"**

.model small ; small memory model

.stack 100h ; 256 Byte memory is allocated for stack memory

.code ; start of code memory

main proc

mov ah,01h ;set cursor size

mov ch,00h ;set cursor top line

mov cl,07h ;set cursor bottom line

int 10h ;display cursor

mov ah,4ch

int 21h

main endp

end main

**program : Set the position of the cursor.**

* input:
* DH = row.
* DL = column.
* BH = page number (0..7).

.model small

.stack 100h

.code

start:

mov dh, 10

mov dl, 20

mov bh, 0

mov ah, 2

int 10h

end start

The following program clears the screen and positions the cursor in the middle of the screen. Two memory locations ‘row’ and ‘col’ are used to keep track of the cursor position.

**;TITLE "Program to move the cursor on the screen"**

.MODEL SMALL ; this defines the memory model

.STACK 100 ; define a stack segment of 100 bytes

.DATA ; this is the data segment

row DB 12 ; define initial row number

col DB 39 ; define initial column number

.CODE

MAIN PROC

MOV AX,@DATA ; get the address of the data segment

MOV DS, AX ; and store it in register DS

CALL CLEARSCREEN ; clear the screen

CALL SETCURSOR ; set the cursor position

MOV AX, 4C00H ; exit to DOS

INT 21H

MAIN ENDP

CLEARSCREEN PROC

MOV AH, 00 ; set video mode

MOV AL, 03 ; for text 80 x 25

INT 10H ; call the DOS interrupt

RET ; return to main procedure

CLEARSCREEN ENDP

SETCURSOR PROC

MOV DH, row ; load row number

MOV DL, col ; load column number

MOV AH, 2 ; use DOS interrupt service for positioning screen

MOV BH, 0 ; video page (usually 0)

INT 10H ; call the DOS interrupt

RET ; return to main procedure

SETCURSOR ENDP

END MAIN

**Lab Task:**

A program clears the screen and positions the cursor at a specified location on the screen using INT 10H functions. The program also displays a message string on the screen using function 09h of INT 21H. Run the program after assembling and linking.

**TITLE "Program to enter characters from keyboard"**

.MODEL SMALL ; this defines the memory model

.STACK 100 ; define a stack segment of 100 bytes

.DATA ; this is the data segment

LF EQU 10 ; Line Feed character (0A in Hex)

CR EQU 13 ; Carriage Return character (0D in Hex)

msg1 DB "WELCOME !!! ", LF, CR, "$"

msg2 DB " CS 203 Lab, CS Department, SMIU ", LF, CR, "$"

.CODE

MAIN PROC

MOV AX,@DATA ; get the address of the data segment

MOV DS, AX ; and store it in register DS

CALL CLEARSCREEN ; clear the screen

MOV DH, 10 ; row 10

MOV DL, 13 ; column 13

CALL SETCURSOR ; set cursor position

LEA DX, msg1 ; load the address offset of message to be displayed

MOV AH, 09h ; use DOS interrupt service for string display

INT 21H ; call the DOS interrupt

MOV DH, 20 ; row 20

MOV DL, 13 ; column 13

CALL SETCURSOR ; set cursor position

LEA DX, msg2 ; load the address offset of message to be displayed

MOV AH, 09h ; use DOS interrupt service for string display

INT 21H ; call the DOS interrupt

MOV AX, 4C00H ; exit to DOS

INT 21H

MAIN ENDP

CLEARSCREEN PROC

MOV AH, 00 ; set video mode

MOV AL, 03 ; for text 80 x 25

INT 10H ; call the DOS interrupt

RET ; return to main procedure

CLEARSCREEN ENDP

SETCURSOR PROC

MOV AH, 2 ; use DOS interrupt service for positioning screen

MOV BH, 0 ; video page (usually 0)

INT 10H ;call the DOS interrupt

RET ; return to main procedure

SETCURSOR ENDP

END MAIN

**Notes**:

1. The above program uses three procedures – MAIN, SETCURSOR, and CLEARSCREEN. The SETCURSOR and CLEARSCREEN procedures are called from the MAIN procedure using the CALL instruction.

2. The SETCURSOR procedure sets the cursor at a specified location on the screen whereas the CLEARSCREEN procedure uses the SET MODE function 00H of INT 10H to set the video mode to 80 x 25 text which automatically clears the screen.

3. You can display a string of characters on the screen, without using a loop, by using MOV AH, 09 with INT 21h. But the string must end with ‘$’ character. You must also load the effective address of the string in register DX.

4. To display a string on a new line, you need to put CR after your string and LF and '$' at the end. CR stands for Carriage Return (or Enter key) and LF stands for Line Feed. You can also put 0Dh or 13 instead of CR (or cr), and 0Ah or 10 instead of LF (or lf).

**Task:**

**Change position of character to center.**

**Change background color with changing palate.**

**Change font color.**

**Display your name**

**Create line with change pixel from 0 to 100.**