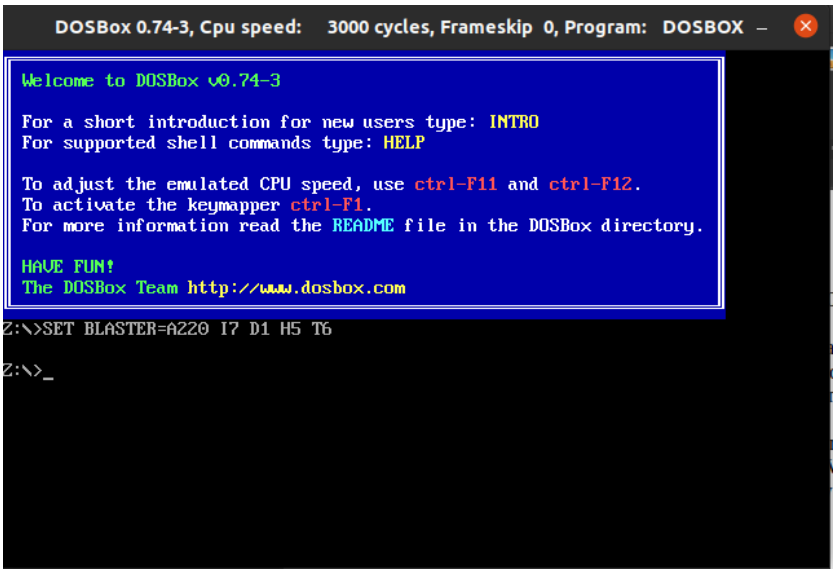
COAL_A_p200165_R10

- **Introduction:**

In Lab NO 10 we have studied about Code Segment(CS), Data Segment(DS) and Printing. Whatever we have done now we have not printed anything screen. Now In Lab 10 we have learned about clear screen and printing some string on the dosbox console.

- **Dosbox:**

Dosbox have total 25 rows and 80 columns in the screen and total 80*25 = 2000 cells. The starting location of video memory is **b800.** Whatever we want to show on the screen we will write that data on to the video memory and that will be shown on the screen.



- **Code No 1:**

First of all we have learned that how will we clear our whole screen. The value of blank space is **20** and the value black background is **07** so what we have done is that we have filled all the video memory with these values and when we runed our program all screen will be cleared and all the existing data is removed from it.

c07-01.asm

```
1   [org 0x0100]
2
3   mov  ax, 0xb800         ; video memory base
4   mov  es, ax             ; cannot move to es through IMM
5   mov  di, 0              ; top left location
6
7   nextpos:
8       mov  word [es:di], 0x0720       ; 0x07 -- full white  (try 41)
9                                       ; 0x20 is the space character
10      add  di, 2
11      cmp  di, 4000
12      jne  nextpos
13
14      mov  ax, 0x4c00
15      int  0x21
```

Here is the code **[es:di]** → mean that take es value add with it di value and go to given location and put **0x0720** to that location, 07 goes to first byte that is black background and 20 goes to another byte and in this way all screen of dosbox will be cleared.

- **Code No 2:**

c07-02.asm

```
[org 0x0100]
    jmp start
message:     db   'hello world'
length:      dw   11
clrscr:
    push es
    push ax
    push di
    mov  ax, 0xb800
    mov  es, ax
    mov  di, 0
    nextloc:
        mov  word [es:di], 0x0720
        add  di, 2
        cmp  di, 4000
        jne  nextloc
    pop  di
    pop  ax
    pop  es
    ret
printstr:
    push bp
    mov  bp, sp
    push es
    push ax
    push cx
    push si
    push di
    mov ax, 0xb800
    mov es, ax
    mov di, 0
    mov si, [bp + 6]
    mov cx, [bp + 4]
    mov ah, 0x07 ; only need to do this once
    nextchar:
        mov al, [si]
        mov [es:di], ax
        add di, 2
        add si, 1
        loop nextchar
    pop di
    pop si
    pop cx
    pop ax
    pop es
    pop bp
    ret 4
start:
    call clrscr
    mov ax, message
    push ax
    push word [length]
    call printstr
    mov ah, 0x1        ; input char is 0x1 in ah
    int 0x21
    mov ax, 0x4c00
    int 0x21
```

The next thing we have learned is printing string on screen. What we did is that we move all the character ascii value into the video-memory location where we want to print the string and that string will be shown on screen when we run our program. To Print anything on screen we need to move to video-memory location that data and that will be shown on screen.

- 1Kbyte = 1024bits
- CS = 2bytes = 16bits
- DS = 2bytes = 16bits

Name: Jawad Ahmed Roll No: 20P-0165 Section: 3A