|  |
| --- |
| FAST National University |
| **Introduction to DOSBox**  **Lab 7** |
|  |

**Computer Organization and Assembly Language**

|  |  |
| --- | --- |
| **Student Name** | Umamah Hussain |
| **Registration #** | 21L-1858 |
| **Instructor** | Hazoor Ahmad |
| **Class** | CS3A |
| **Section** | A |
| **Semester** | Fall 2022 |

Fast School of Computing

FAST-NU, Lahore, Pakistan

# Activity 1

## **Assembly Language Code**

[org 0x0100]

jmp start

message: db 'hello world'

length: dw 11

message1: db 'sentence1'

length1: dw 13

message2: db 'sentence2'

length2: 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

delay: push cx

mov cx, 0xFFFF

loop1: loop loop1

mov cx, 0xFFFF

loop2: loop loop2

pop cx

ret

printstr: push bp

mov bp, sp

push es

push ax

push cx

push si

push di

mov ax, 0xb800

mov es, ax

mov al, 80

mul byte [bp+12]

add ax, [bp+10]

shl ax, 1

mov di, ax

mov si, [bp+6]

mov cx, [bp+4]

mov ah, [bp+8]

nextchar: mov al, [si]

mov [es:di], ax

add di, 2

add si, 1

call delay

loop nextchar

pop di

pop si

pop cx

pop ax

pop es

pop bp

ret 10

start: call clrscr

mov ax, 10

push ax

mov ax, 30

push ax

mov ax, 1

push ax

mov ax, message

push ax

push word [length]

call printstr

mov ax, 20

push ax

mov ax, 10

push ax

mov ax, 0x04

push ax

mov ax, message1

push ax

push word [length1]

call printstr

mov ax, 20

push ax

mov ax, 30

push ax

mov ax, 0x08

push ax

mov ax, message2

push ax

push word [length2]

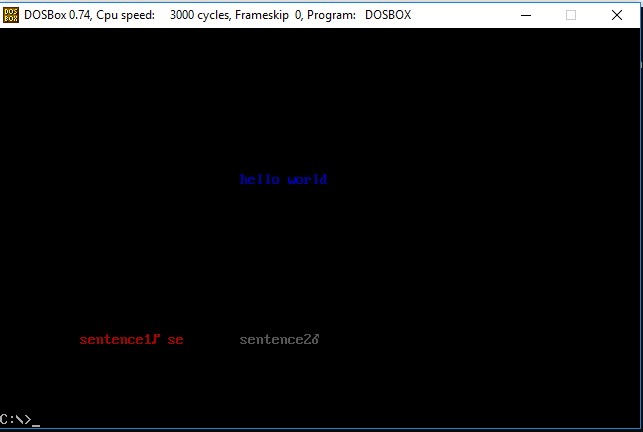
call printstr

mov ax, 0x4c00

int 0x21

-

## **Debugging Screenshots**



# Activity 2

## **Assembly Language Code**

[org 0x0100]

jmp start

top: dw 10

bottom: dw 20

left: dw 30

right: dw 60

start: call clrscr

push word [top]

push word [bottom]

push word [left]

push word [right]

call drawrect

end: mov ax, 0x4c00

int 21h

clrscr: mov ax, 0xb800

mov es, ax

xor di,di

mov ax,0x0720

mov cx,2000

cld

rep stosw

ret

drawrect: push bp

mov bp, sp

pusha

; bp + 4 = right

; bp + 6 = left

; bp + 8 = bottom

; bp + 10 = top

mov al, 80

mul byte [bp + 10]

add ax, [bp + 6]

shl ax, 1

mov di, ax

push di

mov ah, 0x07

a

mov cx, [bp + 4]

sub cx, [bp + 6]

push cx

mov al, '+'

loop1: rep stosw

pop bx

pop di

push bx

dec bx

shl bx, 1

add di, 160

mov cx, [bp + 8]

sub cx, [bp + 10]

sub cx, 2

mov al, '|'

loop2: mov si, di

mov word [es:si], ax

add si, bx

mov word [es:si], ax

sub si, bx

add di, 160

loop loop2

pop cx

mov al, '-'

loop3: rep stosw

return: popa

pop bp

ret 8

## **Debugging Screenshots**

