Computer Organization and Assembly Language

Lab 7

Student Name	Muhammad Ali Butt
Registration #	21L-7646
Instructor	Hazoor Ahmad
Class	Lab-CS2A
Section	BCS-3A2
Semester	Fall 2022

Fast School of Computing

FAST-NU, Lahore, Pakistan

Activity 1

Assembly Language Code

```
                [org 0x0100]

        jmp start

m1: db 'Hello world'
l1: dw 11
m2: db 'Second message'
l2: dw 14
m3: db 'Third message'
l3: dw 13

clrscr:
        push ax
        push bx
        push es

        mov ax, 0xb800
        mov es, ax

        mov ah, 0x07
        mov al, 0x20
        mov bx, 0

loop1:
        mov [es:bx], ax
```

```
    add bx, 2
    cmp bx, 4000
    jne loop1

    pop es
    pop bx
    pop ax
    ret

printS:
    push bp
    mov bp, sp
    push es
    push ax
    push bx
    push si
    push di

    mov ax, 0xb800
    mov es, ax

    mov ax, [bp+4]
    mov bl, 80
    mul bl

    add ax, [bp+6]
    shl ax, 1

    mov di, ax
```

```
        mov ax, [bp+8]
        mov si, 0
        mov bx, [bp+12]

        ;di now stores the starting offset address of where the string should be inserted
        ;es stores 0xb800
        ;ah stores the attribute byte
        ;si stores 0
        ;bx stores the starting offset address of the string

printLoop:
        mov al, [bx+si]
        mov [es:di], ax
        add si, 1
        add di, 2
        cmp si, [bp+10]
        jne printLoop

        pop di
        pop si
        pop bx
        pop ax
        pop es
        pop bp
        ret 10


start:
        call clrscr
```

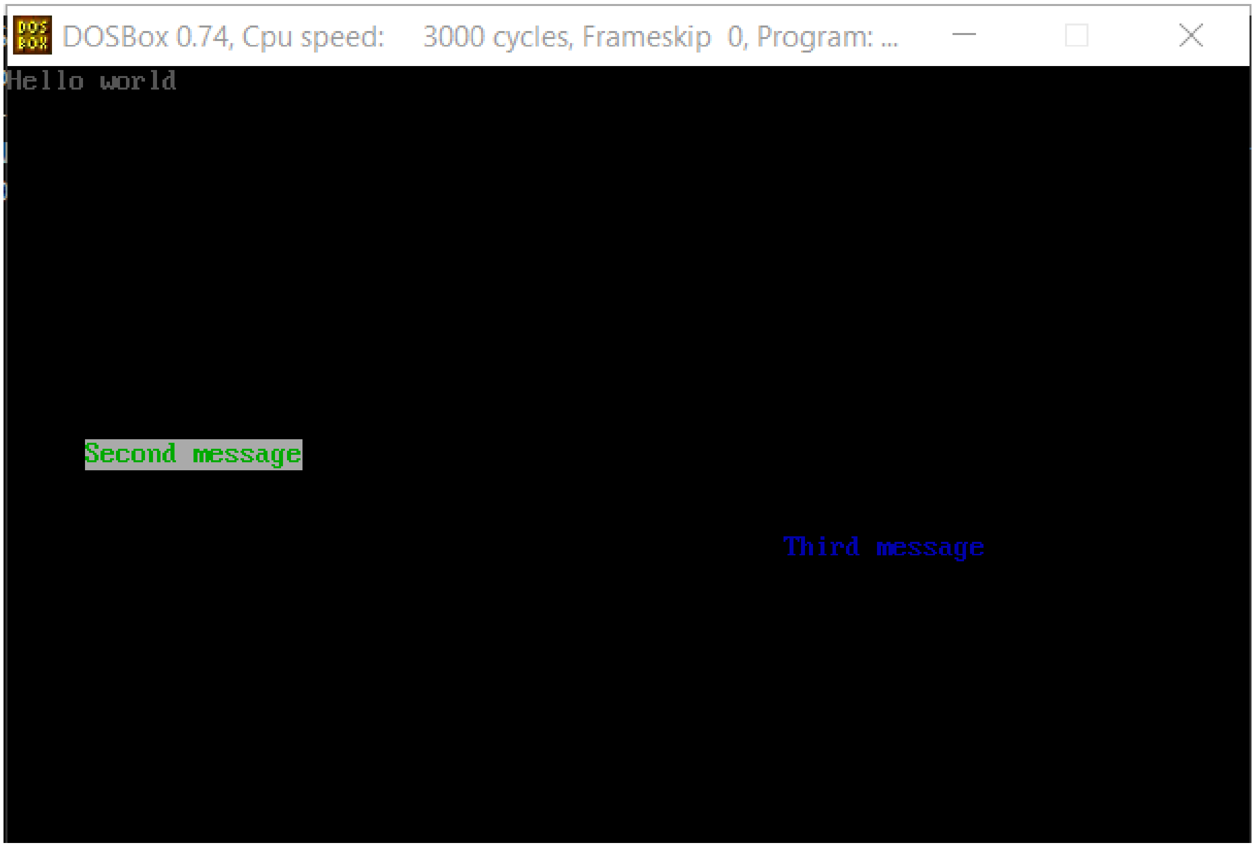
```
push m1
push word [l1]
mov ax, 0x0800
push ax
push 0                  ;Sending column number
push 0                  ;Sending row number
call printS

push m2
push word [l2]
mov ax, 0x7200
push ax
push 5                  ;Sending column number
push 12                 ;Sending row number
call printS

push m3
push word [l3]
mov ax, 0x0100
push ax
push 50                 ;Sending column number
push 15                 ;Sending row number
call printS

mov ah, 0x1
int 0x21
mov ax, 0x4c00
int 0x21
```

Debugging Screenshots


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...

```
Hello world

Second message

Third message
```

Activity 2

Assembly Language Code

```
        [org 0x0100]


        jmp start


top: dw 0

left: dw 0

bottom: dw 10

right: dw 10
```

```
clrscr:
        push ax
        push bx
        push es

        mov ax, 0xb800
        mov es, ax

        mov ah, 0x07
        mov al, 0x20
        mov bx, 0

loop1:
        mov [es:bx], ax
        add bx, 2
        cmp bx, 4000
        jne loop1

        pop es
        pop bx
        pop ax
        ret

drawrect:
        push bp
        mov bp, sp
        push ax
        push es
        push bx
        push cx
        push si
        push di

        mov ax, 0xb800
        mov es, ax

        mov ax, 80
        mov si, [bp+10]
        mul si

        mov si, ax                           ;Address of top row stored in si
        add si, [bp+8]
        shl si, 1                     ;Address of top left coordinate stored in si
        mov word [es:si], 0x012B             ;Store '+' on the top left coordinate

        mov cx, [bp+4]
        sub cx, [bp+8]
        sub cx, 2
        add si, 2

loop2:
        mov word [es:si], 0x012D
        add si, 2
        dec cx
        jnz loop2

        mov word [es:si], 0x012B             ;Store '+' on the top right coordinate
        mov cx, [bp+6]
        sub cx, [bp+10]
        sub cx, 2

        add si, 160

loop3:
        mov word [es:si], 0x017C
        add si, 160
        dec cx
        jnz loop3

        mov word [es:si], 0x012B        ;Store '+' in the bottom right coordinate

        mov cx, [bp+4]
        sub cx, [bp+8]
        sub cx, 2
        sub si, 2

loop4:
        mov word [es:si], 0x012D
        sub si, 2
        dec cx
        jnz loop4

        mov word [es:si], 0x012B        ;Store '+' in the bottom left coordinate

        mov cx, [bp+6]
        sub cx, [bp+10]
        sub cx, 2

        sub si, 160

loop5:
        mov word [es:si], 0x017C
        sub si, 160
        dec cx
        jnz loop5

        pop di
        pop si
        pop cx
        pop bx
        pop es
        pop ax
        pop bp
        ret 8

start:
        call clrscr

        push word [top]
        push word [left]
        push word [bottom]
        push word [right]

        call drawrect
```

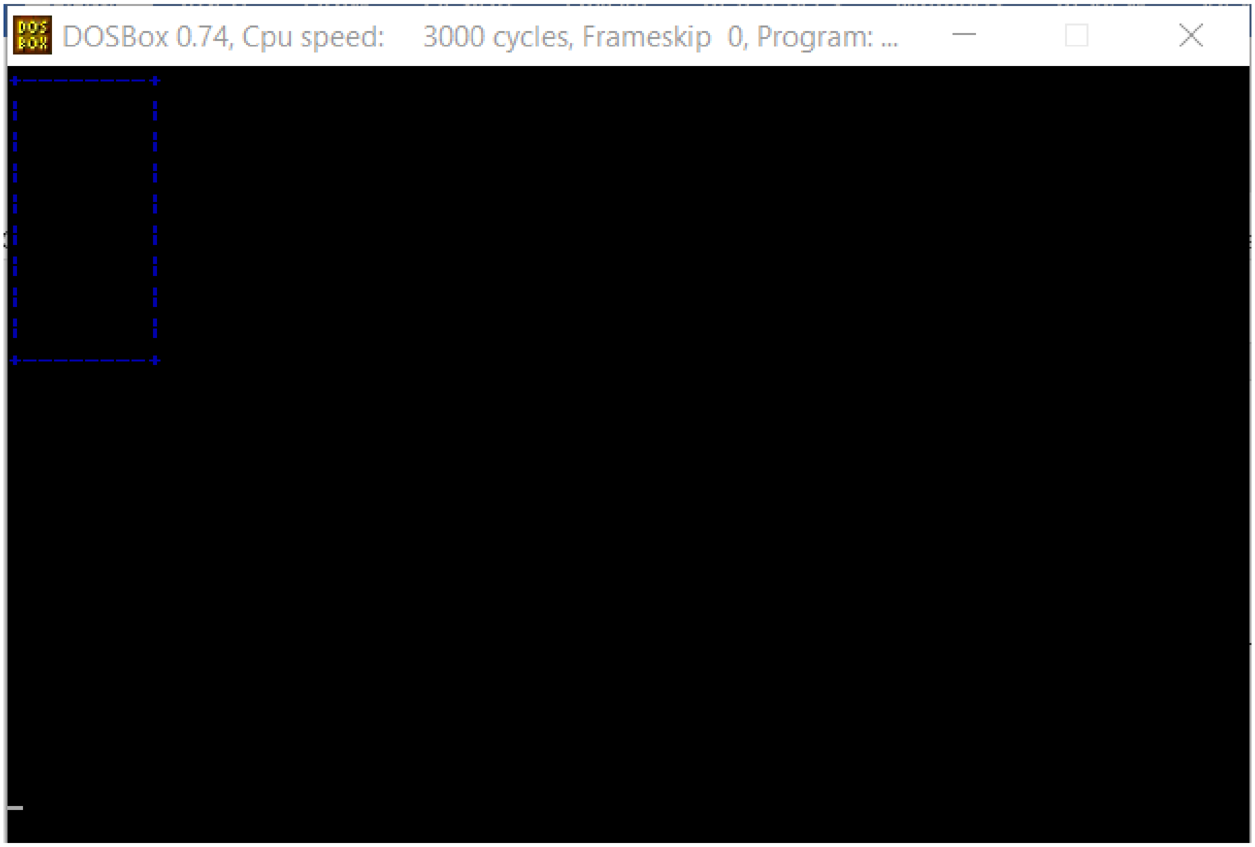
```
mov ah, 0x1

int 0x21

mov ax, 0x4c00

int 0x21
```

Debugging Screenshots


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...