|  |
| --- |
| FAST National University |
| **Lab 12** |
|  |

**Computer Organization and Assembly Language**

|  |  |
| --- | --- |
| **Student Name** | Eisha Amir |
| **Registration #** | 21L-5255 |
| **Instructor** | Hazoor Ahmad |
| **Class** | CS3 |
| **Section** | H |
| **Semester** | Fall 2022 |

Fast School of Computing

FAST-NU, Lahore, Pakistan

# Activity 1

## **Assembly Language Code**

[org 0x0100]

jmp main

oldisr: dd 0

buffer: times 2000 dw 0

clrscr:

push es

push ax

push di

mov ax, 0xb800

mov es, ax

mov di, 0

nextchar:

mov word [es:di], 0x720

add di, 2

cmp di, 4000

jne nextchar

pop di

pop ax

pop es

ret

delay:

push cx

push di

mov cx, 0xFF

delay1:

mov di, 0xFFF

delay2:

dec di

jnz delay2

loop delay1

pop di

pop cx

ret

store\_buffer:

push bp

mov bp, sp

push ax

push cx

push si

push di

push es

push ds

mov ax, 0xb800

mov ds, ax

mov si, 0

mov ax, cs

mov es, ax

mov di, buffer

mov cx, 2000

cld

rep movsw

pop ds

pop es

pop di

pop si

pop cx

pop ax

pop bp

ret

load\_buffer:

push bp

mov bp, sp

push ax

push cx

push si

push di

push es

push ds

mov ax, 0xb800

mov es, ax

mov di, 0

mov ax, cs

mov ds, ax

mov si, buffer

mov cx, 2000

cld

rep movsw

pop ds

pop es

pop di

pop si

pop cx

pop ax

pop bp

ret

kbISR:

push ax

in al, 0x60

cmp al, 00011101b

JNE nextCmp

CALL store\_buffer

CALL clrscr

jmp exit

nextCmp:

cmp al, 10011101b

JNE noMatch

CALL delay

CALL load\_buffer

jmp exit

noMatch:

pop ax

jmp far [cs:oldisr]

exit:

mov al, 0x20

out 0x20, al

pop ax

iret

main:

xor ax, ax

mov es, ax

mov ax, [es:9\*4]

mov [oldisr], ax

mov ax, [es:9\*4+2]

mov [oldisr+2], ax

cli

mov word [es:9\*4], kbISR

mov [es:9\*4+2], cs

sti

mov dx, main

add dx, 15

mov cl, 4

shr dx, cl

mov ax, 0x3100

INT 0x21

## **Debugging Screenshots**



# Activity 2

## **Assembly Language Code**

[org 0x0100]

jmp start

ms: dw 0

count: dw 0, 0, 0, 0, 0

tCount: dw -1

iNo : dw 0

location: db 0

clrscr: pusha

push es

mov ax, 0xb800

mov es, ax

xor di,di

mov ax,0x0720

mov cx,2000

cld

rep stosw

pop es

popa

ret

printStars: pusha

push es

mov ax, 0xb800

mov es, ax

mov al, 80

mul byte [cs:location]

add ax, 159

shl ax, 1

mov di, ax

mov cx, [cs:tCount]

cmp cx, 0

jle return

l1: mov byte [es:di], &#39;\*&#39;

inc byte [cs:location]

add di, 160

loop l1

return: pop es

popa

ret

CTS: pusha

cmp word [cs:iNo], 10

jz l2

add word [cs:ms], 55

cmp word [cs:ms], 1000

jl EOI2

mov word [cs:ms], 0

call printStars

mov ax, [cs:tCount]

mov bx, [cs:iNo]

mov word [cs:count + bx], ax

mov word [cs:tCount], 0

add word [cs:iNo], 2

jmp EOI2

l2: add word [cs:ms], 55

cmp word [cs:ms], 1000

jl EOI2

mov word [cs:ms], 0

mov dx, 0

mov ax, [cs:count + 2]

add dx, ax

mov [cs:count], ax

mov ax, [cs:count + 4]

add dx, ax

mov [cs:count + 2], ax

mov ax, [cs:count + 6]

add dx, ax

mov [cs:count + 4], ax

mov ax, [cs:count + 8]

add dx, ax

mov [cs:count + 6], ax

mov ax, [cs:tCount]

add dx, ax

mov [cs:count + 8], ax

jmp a1

EOI2: jmp EOI

a1: mov [cs:tCount], dx

call clrscr

mov byte [cs:location], 0

call printStars

mov word [cs:tCount], 0

EOI: mov al, 0x20

out 0x20, al

exit: popa

iret

kbisr: push ax

in al, 0x60

shl al, 1

jnc EOI1

inc word [cs:tCount]

EOI1:

mov al, 0x20

out 0x20, al

pop ax

iret

start: mov ax, 0

mov es, ax

mov bx, 0

call clrscr

;Hooking the interrupts

cli

mov word [es: 9\*4], kbisr

mov [es:9\*4+2], cs

mov word [es:8\*4], CTS

mov [es:8\*4+2], cs

sti

;Code for making it TSR

mov dx, start

add dx, 15

mov cl, 4

shr dx, cl

end: mov ax, 0x3100

int 21h

## **Debugging Screenshots**

