**Computer Organization and Assembly Language**

|  |  |
| --- | --- |
| **Lab 13** | |
| **Topic** | 1. Interrupts |

**PART 1**

**Example 1: Printing Character Using Interrupt**

start:  mov ah, 0

int 16h ; wait for any key....

cmp al, 27  ; if key is 'esc' then exit.

je stop

;al contains ascii of pressed key

mov ah, 0Eh ; print it.

int 10h

jmp start

stop:

.exit

**Example 2: Printing String Using Interrupt**

mov al, 1;update curser after every character printing

mov bh, 0;page 0, means first page

mov bl, 00111011b;attribules

mov cx, 15 ; message size

mov dl, 10 ;row

mov dh, 7 ;col

push cs

pop es

mov bp, msg1

mov ah, 13h

int 10h

mov ax,0x4c00

int 21h

msg1 db " hello, world! "

**Example 3: (Taking Input from User and display)**

MOV AX, 0xB800

MOV ES, AX ; Initializing ES with video memory address

MOV AH, 0 ; service number

INT 0x16 ; calling interrupt number 16h

; When you call interrupt 16h with service number 0, processor waits for keyboard input. When a key is pressed, its ASCII value is stored in AL register.

; Printing the character on screen.

MOV DI, 0 ; screen location di=0 top left.

MOV AH,07h ; attribute byte

STOSW ; displaying on screen

Mov ax,0x4c00

Int 21h

**Example 4: (Take input from User until they press Esc)**

MOV AX, 0xB800

MOV ES, AX ; Initializing ES with video memory address

XOR DI, DI ; screen location di=0 top left.

again:

MOV AH, 0 ; service number

INT 0x16 ; calling interrupt number 16h

; When you call interrupt 16h with service number 0, processor waits for keyboard input. When a key is pressed, its ASCII value is stored in AL register.

; Printing the character on screen.

MOV AH,07h ; attribute byte

STOSW ; displaying on screen

cmp al, 0x1b

jne again

mov ax,0x4c00

int 21h

**Example 5: (Taking Input from User setting cursor position display the character)**

;total 3 interrupts are used to perform the task.

[org 100h]

;input interrupt

mov ah,0

int 16h

;setting cursor position interrupt

mov dh, 12

mov dl, 40

mov bh, 0

mov ah, 2

int 10h

;display character interrupt

;mov al, '\*'

mov ah, 0eh

int 10h

mov ax,0x4c00

int 21h

**Interrupt vector table-address mapping**

* Offset: n\*4 ; offset address of nth interrupt
* Segment: n\*4+2 ; base address of nth interrupt

