National University of Computer and Emerging Sciences 

**Laboratory Manual**

*for*

**Computer Organization and Assembly Language Programming**

| Course Instructor | Salman Mubarak |
| --- | --- |
| Lab Instructor | Sana Ejaz |
| Semester | Fall 2024 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

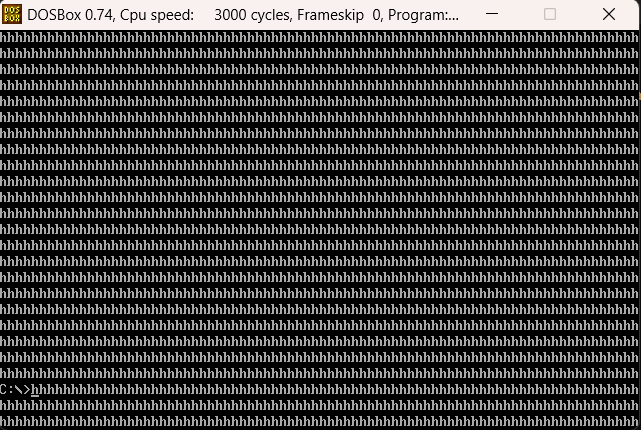
Page 1

**OBJECTIVES:**

∙ Perform Hooking and

∙ Interrupts

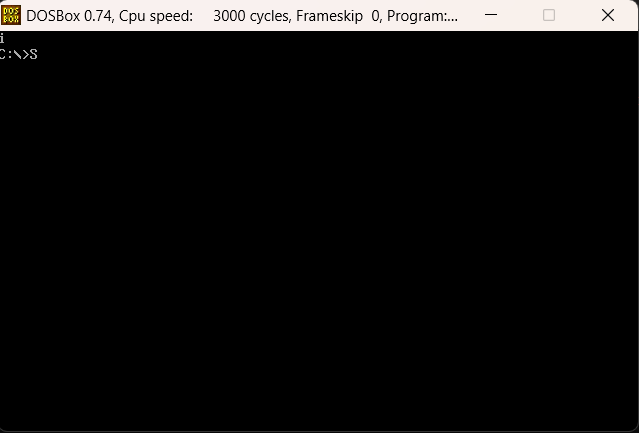
**Task 1: The following program keeps taking a key from the user and filling the screen with this key. Fix the code such that it exits when the user presses ESC (Escape).**

****

| ; Infinite Key Printing  [org 0x0100]  jmp start  printKey:  push ax  pop bx ; bx=ax  push es  push ax  push cx  push di  mov ax, 0xb800  mov es, ax ; point es to video base  xor di, di ; point di to top left column  mov al, bl  mov ah, 0x07 ; normal attribute  mov cx, 2000 ; number of screen locations  cld ; auto increment mode  rep stosw ; clear the whole screen  pop di  pop cx  pop ax  pop es  ret  start:  mov ah, 0 ; service 0 – get keystroke  int 0x16 ; call BIOS keyboard service  cmp al, 27  je terminate  call printKey ; clear the screen  jmp start  terminate:  mov ax, 0x4c00 ; terminate program  int 0x21 |
| --- |

**Task 2: Write a code to read a key from the keyboard and displays the next character on screen. For example, if ‘e’ is pressed then ‘f’ is displayed.**

| [org 0x0100]  JMP start  printnext:  PUSH BP  MOV BP, SP  PUSHA  MOV AH, 0x02  MOV DX, [BP + 4]  INC DL  INT 0x21  POPA  MOV SP, BP  POP BP  RET 2  start:  MOV AX, 0x0003  INT 0x10  MOV AX, 0  INT 0x16    PUSH AX  CALL printnext  MOV AX, 0x4C00  INT 0x21 |
| --- |

****

**Task 3: Hook int 80h such that whenever Left, Up, Down, and right keys are pressed the asterisk on- screen will mow Left, Up, down, and right respectively.**

| [org 0x0100]  JMP start  pos: dw 40, 12  customISRforINT80h:  PUSHA  PUSH DS  PUSH ES  PUSH CS  POP DS  MOV AX, 0xB800  MOV ES, AX    MOV AX, 80  MOV BL, [pos + 2]  MUL BL  ADD AX, [pos]  SHL AX, 1  MOV DI, AX  MOV AX, 0  INT 0x16    CMP AH, 0x4B ; leftarrow  JNE check\_right  MOV word [ES:DI], 0x0720  MOV word [ES:DI - 2], 0x072A    DEC word [pos]  JMP terminate  check\_right:  CMP AH, 0x4D ; rightarrow  JNE check\_up  MOV word [ES:DI], 0x0720  MOV word [ES:DI + 2], 0x072A    INC word [pos]  JMP terminate  check\_up:  CMP AH, 0x48 ; uparrow  JNE check\_down  MOV word [ES:DI], 0x0720  MOV word [ES:DI - 160], 0x072A    DEC word [pos + 2]  JMP terminate  check\_down:  CMP AH, 0x50 ; downarrow  JNE terminate  MOV word [ES:DI], 0x0720  MOV word [ES:DI + 160], 0x072A    INC word [pos + 2]  terminate:  POP ES  POP DS  POPA  IRET  start:  MOV AX, 0x0003  INT 0x10  MOV AX, 0xB800  MOV ES, AX  MOV word [ES:2000], 0x072A  XOR AX, AX  MOV ES, AX  MOV AX, customISRforINT80h  CLI  MOV word [ES:128 \* 4], customISRforINT80h  MOV [ES:128 \* 4 + 2], CS  STI  l1:  INT 0x80  JMP l1 |
| --- |

