National University of Computer and Emerging Sciences 

**Laboratory Manual**

*for*

**Computer Organization and Assembly Language Programming**

| Course Instructor | Aleena Ahmed |
| --- | --- |
| Lab Instructor | Sana Ejaz |
| Semester | Fall 2024 |

Department of Computer Science

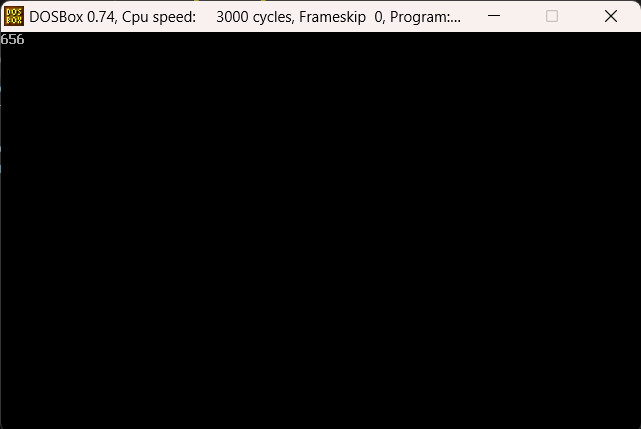
FAST-NU, Lahore, Pakistan

Page 1

**OBJECTIVES:**

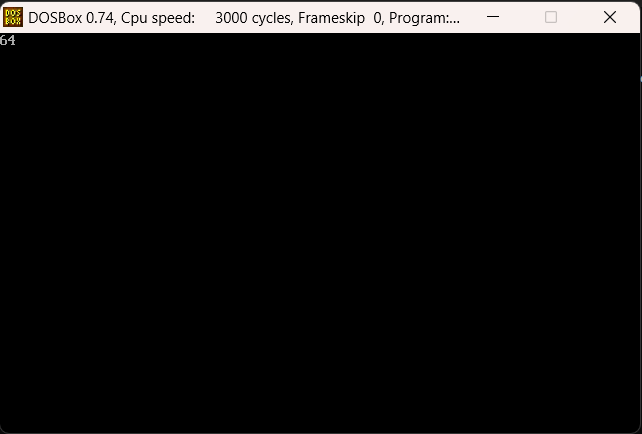
* Understand the use of IN and OUT instructions for direct hardware communication.
* Learn to manipulate and handle Programmable Interrupt Controller (PIC) ports.
* Experiment with interrupt chaining and unhooking interrupts for custom handling.
* Explore the basics of the Programmable Interval Timer (PIT) and its integration with interrupts.
* Gain insight into terminating and staying resident (TSR) programs and their applications.

**Task 1: Basic Hardware Communication with IN and OUT Instructions**

****

| [org 0x0100]  JMP start  tick: dw 0  oldISR: dd 0  printnum:  PUSH BP  MOV BP, SP    PUSH ES  PUSH AX  PUSH BX  PUSH CX  PUSH DX  PUSH DI    MOV AX, 0xB800  MOV ES, AX    MOV AX, [BP + 4]  MOV BX, 10  MOV CX, 0  nextdigit:  MOV DX, 0  DIV BX  ADD DX, 0x0730  PUSH DX  INC CX    CMP AX, 0  JNZ nextdigit    MOV DI, 0  nextpos:  POP DX  MOV [ES:DI], DX  ADD DI, 2  LOOP nextpos    POP DI  POP DX  POP CX  POP BX  POP AX  POP ES  POP BP  RET 2  timerISR:  INC word [CS:tick]  PUSH word [CS:tick]  CALL printnum  JMP FAR [oldISR]  hookTimer:  PUSH AX  PUSH ES  PUSH word 0  POP ES  MOV AX, [ES:8 \* 4]  MOV [oldISR], AX  MOV AX, [ES:8 \* 4 + 2]  MOV [oldISR + 2], AX  cli  MOV word [ES:8 \* 4], timerISR  MOV [ES:8 \* 4 + 2], CS  sti  POP ES  POP AX  RET  unhookTimer:  PUSH AX  PUSH DX  PUSH DS  MOV AX, 0x2508  MOV DX, [oldISR]  MOV DS, [oldISR + 2]  INT 0x21  POP DS  POP DX  POP AX    RET  rwPort:  PUSH BP  MOV BP, SP  PUSH AX  CMP AL, 0x31  JNE nm1  MOV AL, 0x36  OUT 0x43, AL  MOV AX, 11931  OUT 0x40, AL  MOV AL, AH  OUT 0x40, AL  nm1:  CMP AL, 0x32  JNE nm2  MOV AL, 0x36  OUT 0x43, AL  MOV AX, 5966  OUT 0x40, AL  MOV AL, AH  OUT 0x40, AL  nm2:  CMP AL, 0x33  JNE nm3  MOV AL, 0x36  OUT 0x43, AL  MOV AX, 3977  OUT 0x40, AL  MOV AL, AH  OUT 0x40, AL    nm3:  CMP AL, 0x34  JNE nm4  MOV AL, 0x36  OUT 0x43, AL  MOV AX, 2983  OUT 0x40, AL  MOV AL, AH  OUT 0x40, AL  nm4:  CMP AL, 0x35  JNE term  MOV AL, 0x36  OUT 0x43, AL  MOV AX, 0  OUT 0x40, AL  MOV AL, AH  OUT 0x40, AL  term:  POP AX  POP BP    RET 2  start:  MOV AX, 0x0003  INT 0x10  CALL hookTimer  l1:  MOV AX, 0  INT 0x16  PUSH AX  CALL rwPort  CMP AL, 27  JE terminate    JMP l1  terminate:  CALL unhookTimer    MOV AX, 0x4C00  INT 0x21 |
| --- |

**Task 2: Interrupt Chaining and Unhooking an Interrupt**

****

| [org 0x0100]  JMP start  tick: dw 0  oldISR: dd 0  printnum:  PUSH BP  MOV BP, SP    PUSH ES  PUSH AX  PUSH BX  PUSH CX  PUSH DX  PUSH DI    MOV AX, 0xB800  MOV ES, AX    MOV AX, [BP + 4]  MOV BX, 10  MOV CX, 0  nextdigit:  MOV DX, 0  DIV BX  ADD DX, 0x0730  PUSH DX  INC CX    CMP AX, 0  JNZ nextdigit    MOV DI, 0  nextpos:  POP DX  MOV [ES:DI], DX  ADD DI, 2  LOOP nextpos    POP DI  POP DX  POP CX  POP BX  POP AX  POP ES  POP BP  RET 2  timerISR:  INC word [CS:tick]  PUSH word [CS:tick]  CALL printnum  JMP FAR [oldISR]  hookTimer:  PUSH AX  PUSH BX  PUSH DX  PUSH ES  MOV AX, 0x3508  INT 0x21  MOV [oldISR], BX  MOV [oldISR + 2], ES  MOV AX, 0x2508  MOV DX, timerISR  INT 0x21  POP ES  POP DX  POP BX  POP AX  RET  unhookTimer:  PUSH AX  PUSH DX  PUSH DS  MOV AX, 0x2508  MOV DX, [oldISR]  MOV DS, [oldISR + 2]  INT 0x21  POP DS  POP DX  POP AX    RET  start:  MOV AX, 0x0003  INT 0x10  CALL hookTimer  l1:  MOV AX, 0  INT 0x16  CMP AL, 27  JE terminate  CMP AL, 117  JNE nextmatch  CALL unhookTimer  nextmatch:  CMP AL, 104  JNE l1  CALL hookTimer  JMP l1  terminate:  MOV AX, 0x4C00  INT 0x21 |
| --- |