National University of Computer and Emerging Sciences 

**Laboratory Manual**

*for*

**Computer Organization and Assembly Language Programming**

Lab Instructor Sana Ejaz

| Semester | Fall 2024 |
| --- | --- |

Department of Computer Science

FAST-NU, Lahore, Pakistan

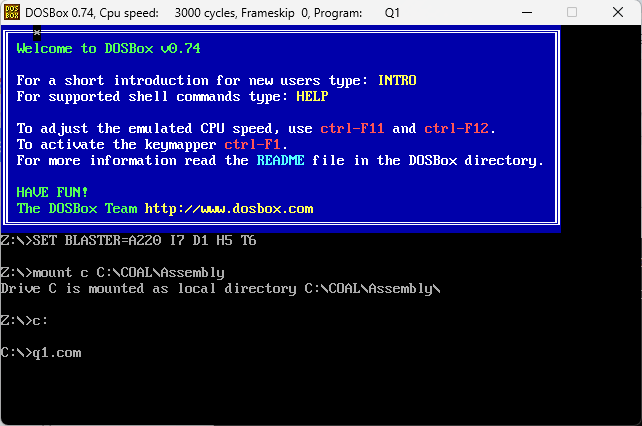
**OBJECTIVES:**

∙ 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: Timer-Controlled Asterisk Movement**

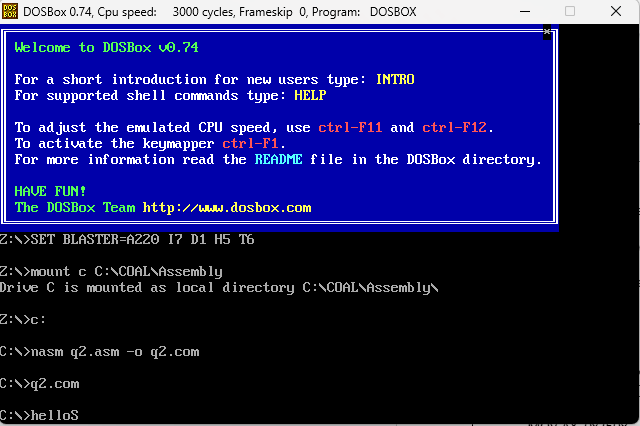
Write a program to make an asterisk travel the border of the screen, from upper left to upper right to lower right to lower left and back to upper left indefinitely, making each movement after one second. You are not allowed to write any loop in your program. The Timer will move the star by one cell only. Do not lose the previous content of the cell.



| [org 0x0100]  JMP start  oldtimer: dd 0  tickcount: dw 0  movement: dw 0  position: dw -1, 0  prevValue: dw 0  timer:  PUSH ES  PUSH AX  PUSH DI  INC word [CS:tickcount]  CMP word [CS:tickcount], 18  JE updatePosition  terminateTimer:  POP DI  POP AX  POP ES  JMP FAR [CS:oldtimer]  updatePosition:  PUSH 0xB800  POP ES  MOV AX, 80  MUL byte [CS:position + 2]  ADD AX, [CS:position]  SHL AX, 1  MOV DI, AX  MOV AX, [CS:prevValue]  MOV word [ES:DI], AX  CMP word [CS:movement], 0  JNE downMove  INC word [CS:position]  ADD DI, 2  CMP DI, 158  JNE endPosition    INC word [CS:movement]  JMP endPosition  downMove:  CMP word [CS:movement], 1  JNE leftMove  INC word [CS:position + 2]  ADD DI, 160  CMP DI, 3998  JNE endPosition  INC word [CS:movement]  JMP endPosition  leftMove:  CMP word [CS:movement], 2  JNE upMove  DEC word [CS:position]  SUB DI, 2  CMP DI, 3840  JNE endPosition  INC word [CS:movement]  JMP endPosition  upMove:  DEC word [CS:position + 2]  SUB DI, 160  CMP DI, 0  JNE endPosition  MOV word [CS:movement], 0  endPosition:  MOV AX, [ES:DI]  MOV [CS:prevValue], AX  MOV word [ES:DI], 0x072A  MOV word [CS:tickcount], 0  JMP terminateTimer  start:  MOV AX, 0x3508  INT 0x21  MOV [oldtimer], BX  MOV [oldtimer + 2], ES  MOV AX, 0x2508  MOV DX, timer  INT 0x21  JMP $ |
| --- |

**Task 2: Adding User Control to Start/Stop the Animation and Keypad Filtering**

Update previous program, star should start moving (from its previous position, initial position will be top left of screen) if user enters Left Shift key and it should stop on Right Shift Key. Other applications should work fine along with this functionality. Both shifts should work properly on command prompt. Number keys at the right side of keypad should not work on command prompt but number keys on top of the keypad should work properly.



| [org 0x0100]  JMP start  oldtimer: dd 0  oldkb: dd 0  tickcount: dw 0  movement: dw 0  position: dw -1, 0  prevValue: dw 0  timer:  PUSH ES  PUSH AX  PUSH DI  INC word [CS:tickcount]  CMP word [CS:tickcount], 18  JE updatePosition  terminateTimer:  POP DI  POP AX  POP ES  JMP FAR [CS:oldtimer]  updatePosition:  PUSH 0xB800  POP ES  MOV AX, 80  MUL byte [CS:position + 2]  ADD AX, [CS:position]  SHL AX, 1  MOV DI, AX  MOV AX, [CS:prevValue]  MOV word [ES:DI], AX  CMP word [CS:movement], 0  JNE downMove  INC word [CS:position]  ADD DI, 2  CMP DI, 158  JNE endPosition    INC word [CS:movement]  JMP endPosition  downMove:  CMP word [CS:movement], 1  JNE leftMove  INC word [CS:position + 2]  ADD DI, 160  CMP DI, 3998  JNE endPosition  INC word [CS:movement]  JMP endPosition  leftMove:  CMP word [CS:movement], 2  JNE upMove  DEC word [CS:position]  SUB DI, 2  CMP DI, 3840  JNE endPosition  INC word [CS:movement]  JMP endPosition  upMove:  DEC word [CS:position + 2]  SUB DI, 160  CMP DI, 0  JNE endPosition  MOV word [CS:movement], 0  endPosition:  MOV AX, [ES:DI]  MOV [CS:prevValue], AX  MOV word [ES:DI], 0x072A  MOV word [CS:tickcount], 0  JMP terminateTimer  kb:  PUSH AX  PUSH DX  PUSH DS  IN AL, 0x60    ; NUMPAD 7  CMP AL, 71  JL otherKey  ; NUMPAD 3  CMP AL, 81  JG otherKey  MOV AL, 0x20  OUT 0x20, AL  POP DS  POP DX  POP AX  IRET  otherKey:  CMP AL, 0x2A  JNE checkRight  PUSH CS  POP DS  MOV AX, 0x2508  MOV DX, timer  INT 0x21  JMP nomatch  checkRight:  CMP AL, 0x36  JNE nomatch  MOV DX, [CS:oldtimer + 2]  MOV DS, DX  MOV DX, [CS:oldtimer]  MOV AX, 0x2508  INT 0x21    nomatch:  POP DS  POP DX  POP AX  JMP FAR [CS:oldkb]  start:  MOV AX, 0x3508  INT 0x21  MOV [oldtimer], BX  MOV [oldtimer + 2], ES  MOV AX, 0x3509  INT 0x21  MOV [oldkb], BX  MOV [oldkb + 2], ES  MOV AX, 0x2509  MOV DX, kb  INT 0x21  MOV DX, start  ADD DX, 15  SHR DX, 4  MOV AX, 0x3100  INT 0x21 |
| --- |