

```

***** M O U S E P A *****
;*
;*-----
;* Task : Create mouse called event handler for use with
;* a Turbo Pascal program.
;*-----
;*
;* Author : MICHAEL TISCHER
;* Developed on : 04/24/1989
;* Last update : 04/24/1989
;*-----
;* assembly : MASM /MX MOUSEPA; or
;* TASM -MX MOUSEPA
;* ... add to MOUSEP program code
;*****
;

;== Data segment =====
DATA segment word public
DATA ends ;note--no variables in this program

;== Program =====
CODE segment byte public ;Program segment

assume CS:CODE ;CS points to the code segment whose
;contents are unknown to DS, SS & ES

public AssmHand ;Allows the TP program to read
;the address of the assembler handlers

extrn MouEventHandler : near ;TP event handler to be called

active db 0 ;points to whether a call can occur

;-----
;-- AssmHand: The event handler which first calls the mouse driver, then
;-- calls the TP MouEventHandler procedure
;-- Direct call from TP not allowed

AssmHand proc far

;-- First save all processor registers on stack ---

cmp active,0 ;Call done yet?
jne ende ;NO --> Don't exit call

mov active,1 ;No more calls, please

push ax
push bx
push cx
push dx
push di
push si
push bp
push es
push ds

;-- Push arguments for TP function call onto stack -----
;-- Call:
;-- MouEventHandler (EvFlags, ButStatus, x , y : integer );

push ax ;Push event flags onto stack
push bx ;Push mouse button status onto stack

mov di,cx ;Move horizontal ordinate onto DI
mov cl,3 ;Counter for coordinate number

shr di,cl ;Divide DI (horizontal ord.) by 8 and
push di ;push onto stack

shr dx,cl ;Divide DX (vertical ord.) by 8 and
push dx ;push onto stack

mov ax,DATA ;Segment address of data segment AX
mov ds,ax ;Move data from AX to DS register

call MouEventHandler ;Call TP procedure

;-- Get reserved registers from stack -----

pop ds
pop es
pop bp
pop si
pop di

```

```

        pop    dx
        pop    cx
        pop    bx
        pop    ax

        mov    active,0          ;Re-enable call

ende:    ret                    ;Return to mouse driver

AssmHand endp

;-----
CODE     ends                  ;End of code segment
        end                ;End of program
```