

## USO DEL MOUSE

### INT 33H

Esta es una interrupción de DOS y ayuda a controlar el uso del mouse.

Se puede establecer el uso del ratón en modo texto y en modo gráfico.

Para hacer uso del mouse se debe establecer el modo de video, inicializar mouse, mostrar el puntero, realizar acciones de desplazamiento, selección, etc., y al final ocultar el puntero.

### Iniciación del mouse

El servicio utilizado es el 00, este servicio inicializa el controlador del mouse como sigue:

- Establece el puntero en el centro de la pantalla
- Si esta visible el apuntador lo oculta
- Fija la pagina de despliegue del apuntador del ratón en cero
- Establece el puntero de acuerdo al modo de video
  - Modo texto: rectángulo, color inverso
  - Modo grafico: forma de flecha
- Establece la razón mickey a píxel
  - Razón hor: 8 a 8
  - Razón Vert: 16 a 8
- Establece los limites horizontal y vertical para el puntero máximo y mínimo
- Establece el umbral de velocidad doble a 64 mickey por segundo

Descripción del servicio:

Registro	Valor	Retorna	
AX	Servicio:00	Registro	Valor
		AX	0000h, si no está disponible controlador FFFFFh, si esta disponible
		BX	Número de botones del ratón

### **Despliegue del puntero del mouse**

Servicio utilizado 01, para mostrar el puntero del mouse

Descripción del servicio:

Registro	Valor
AX	Servicio:01

### **Ocultar el puntero del mouse**

Servicio utilizado 02, se utiliza al final de la ejecución del programa.

Descripción del servicio:

Registro	Valor
AX	Servicio:02

### **Obtiene el estado del botón y la posición del puntero del mouse**

Servicio utilizado 03, retorna información útil para el manejo del mouse.

Descripción del servicio:

Registro	Valor	Retorna	
AX	Servicio:03	<b>Registro</b>	<b>Valor</b>
		BX	Estado de los botones de acuerdo a la posición del bit, como sigue: Bit 0: botón izquierdo, donde 0= arriba, 1= abajo Bit 1: botón derecho, donde 0= arriba, 1= abajo Bit 2: botón central, donde 0= arriba, 1= abajo Bits 3-15: reservados
		CX	Coordenada horizontal: x
		DX	Coordenada vertical: y

*Nota: Las coordenadas son expresadas en términos de píxeles, aun en modo texto (8 x byte para modo video 03)*

#### **Establece la posición del puntero del mouse**

Servicio utilizado 04, coloca el puntero en la nueva posición, ajustando como sea necesario si está fuera de los límites mínimo y máximo.

Descripción del servicio:

Registro	Valor
AX	Servicio:04
CX	Posición horizontal
DX	Posición vertical

#### **Obtiene información de la pulsación de los botones del mouse**

Servicio utilizado 05, retorna el estado arriba-debajo de todos los botones, el conteo de las presiones y posición del botón requerido.

Descripción del servicio:

Registro	Valor	Retorna	
<b>AX</b>	Servicio:05		
<b>BX</b>	Número del botón: 0 = izquierdo 1 = derecho 2 = central	<b>Registro</b>	<b>Valor</b>
		<b>AX</b>	Estado de los botones de acuerdo a la posición del bit como sigue: Bit 0: botón izquierdo, donde 0= arriba, 1= abajo Bit 1: botón derecho, donde 0= arriba, 1= abajo Bit 2: botón central, donde 0= arriba, 1= abajo Bits 3-15: reservados
		<b>BX</b>	Contador de presiones del botón
		<b>CX</b>	Coordenada horizontal de la última presión del botón
		<b>DX</b>	Coordenada vertical de la última presión del botón

*Nota: La operación restablece el contador de presiones del botón en cero.*

### **Fija los límites horizontales para el puntero del mouse**

Servicio utilizado 07, fija los límites horizontales mínimo y máximo para el puntero.

Descripción del servicio:

Registro	Valor
<b>AX</b>	Servicio:07
<b>CX</b>	Límite inferior
<b>DX</b>	Límite superior

*Nota: Si el número mínimo es mayor que el máximo, la operación intercambia los números, también la operación mueve al puntero al área nueva.*

### **Fija los límites verticales para el puntero del mouse**

Servicio utilizado 08, fija los límites verticales mínimo y máximo para el puntero.

Descripción del servicio:

Registro	Valor
<b>AX</b>	Servicio:08
<b>CX</b>	Límite inferior
<b>DX</b>	Límite superior

*Nota: Si el número mínimo es mayor que el máximo, la operación intercambia los números, también la operación mueve al puntero al área nueva.*

```
;-----  
MANEJO DEL MOUSE EN MODO TEXTO  
;-----  
  
.MODEL SMALL  
.STACK 100H  
.DATA  
MEN1 DB ' MOUSE NO DETECTADO$'  
  
.CODE  
MAIN PROC FAR  
    ; Protocolo  
    PUSH DS  
    SUB AX, AX  
    PUSH AX  
    MOV AX, @DATA  
    MOV DS, AX  
    ; Protocolo (End)  
  
    ; Inicializacion del mouse (begin)  
    CALL INIT_MOUSE  
    CMP AX, 00  
    JE ERROR  
    CALL SHOW_CURSOR  
    MOV CX, 320  
    MOV DX, 240  
    CALL PUT_CURSOR  
    JMP EXIT  
    ; Inicializacion del mouse (end)  
ERROR: LEA DX, MEN1  
    CALL MENSAJE  
    CALL LEE  
    JMP SALTE  
  
EXIT:    CALL LEE  
    CALL OCULTA_MOUSE  
    CALL RESTAURA  
SALTE:  CALL SAL_DOS  
    RET  
MAIN ENDP  
  
RESTAURA PROC  
    MOV AH, 00H  
    MOV AL, 03H  
    INT 10H
```

```

RET
RESTAURA ENDP

SAL_DOS PROC
    MOV AH,4CH
    INT 21H
    RET
SAL_DOS ENDP

INIT_MOUSE PROC NEAR
    MOV AX, 00H
    INT 33H
    RET
INIT_MOUSE ENDP

OCULTA_MOUSE PROC NEAR
    MOV AX, 02H

    INT 33H
    RET
OCULTA_MOUSE ENDP

SHOW_CURSOR PROC NEAR
    PUSH AX
    MOV AX, 01H
    INT 33H
    POP AX
    RET
SHOW_CURSOR ENDP

PUT_CURSOR PROC NEAR
    PUSH AX
    MOV AX, 04H
    INT 33H
    POP AX
    RET
PUT_CURSOR ENDP

LEE PROC
    PUSH AX
    MOV AH,01
    INT 21H
    POP AX
    RET
LEE ENDP

MENSAJE PROC
    PUSH AX
    MOV AH,09H
    INT 21H
    POP AX
    RET
MENSAJE ENDP

END MAIN

```

```

;-----
MANEJO DEL MOUSE EN MODO GRAFICO
;-----

.MODEL SMALL
.STACK 100H
.DATA
MEN1 DB 'NO MOUSE DETECTADO$'

.CODE
MAIN PROC FAR
    ; Protocolo
    PUSH DS
    SUB AX, AX
    PUSH AX
    MOV AX, @DATA
    MOV DS, AX
    ; Protocolo (End)

    CALL GRAFICO

    ; Inicializacion del mouse
    CALL INIT_MOUSE
    CMP AX, 0000
    JE ERROR
    CALL SHOW_CURSOR
    MOV CX, 620
    MOV DX, 410
    CALL PUT_CURSOR
    JMP EXIT
ERROR: LEA DX, MEN1
    CALL MENSAJE
    CALL LEE
    JMP SALTE

EXIT:  CALL LEE
      CALL OCULTA_MOUSE
SALTE: CALL RESTAURA
      CALL SAL_DOS
      RET
MAIN ENDP

    RESTAURA PROC
        MOV AH, 00H
        MOV AL, 03H
        INT 10H
        RET
    RESTAURA ENDP

    SAL_DOS PROC
        MOV AH, 4CH
        INT 21H
        RET
    SAL_DOS ENDP

GRAFICO PROC NEAR
    ; Inicializa modo grafico 640x480x16 (Begin)

```

```

        MOV AH, 00H
        MOV AL, 12H
        INT 10H
        ; Inicializa modo grafico 640x480x16 (End)
        RET
GRAFICO ENDP

INIT_MOUSE PROC NEAR
        MOV AX, 00H
        INT 33H
        RET
INIT_MOUSE ENDP

OCULTA_MOUSE PROC NEAR
        MOV AX, 02H
        INT 33H
        RET
OCULTA_MOUSE ENDP

SHOW_CURSOR PROC NEAR
        PUSH AX
        MOV AX, 01H
        INT 33H
        POP AX
        RET
SHOW_CURSOR ENDP

PUT_CURSOR PROC NEAR
        PUSH AX
        MOV AX, 04H
        INT 33H
        POP AX
        RET
PUT_CURSOR ENDP

LEE PROC
        PUSH AX
        MOV AH, 01
        INT 21H
        POP AX
        RET
LEE ENDP

MENSAJE PROC
        PUSH AX
        MOV AH, 09H
        INT 21H
        POP AX
        RET
MENSAJE ENDP

END MAIN

```



```

;-----
MUESTRA LAS COORDENADAS DE LA POSICIÓN ACTUAL DEL PUNTERO DEL MOUSE
USO DE INT 21, SERVICIO 40
;-----

```

```

.MODEL SMALL
.STACK 100H
.DATA
XB      DW  00
YB      DW  00
VALASC  DW  00
MEN1    DB  'Mouse no disponible...$'
DESPDATO LABEL BYTE
        XMEN  DB  'x= '
        XASCII DW ?
        YMEN  DB  ' y= '
        YASCII DW ?
YXI     DW  00
YXF     DW  00
FPP     DB  00
.CODE
MAIN    PROC  FAR
        ;PROTOCOLO
        PUSH  DS
        SUB   AX,AX
        PUSH  AX
        MOV   AX,@DATA
        MOV   DS,AX
        MOV   ES,AX
        ;FIN  PROTOCOLO

        CALL  INITTEXT
        MOV   FPP,3EH
        MOV   YXI,00H
        MOV   YXF,184FH
        CALL  TEXTBACKGROUND
        CALL  I_MOUSE
        CMP   AX,00
        JE    ERROR
        CALL  M_MOUSE
        MOV   YXI,1843H
OTRO:    CALL  P_MOUSE ; SERVICIO 03
        CMP   BX,01
        JE    EXIT
        CALL  GOTOXY
        MOV   AX,XB
        CALL  CONV
        MOV   AX,VALASC
        MOV   XASCII,AX
        MOV   AX,YB
        CALL  CONV
        MOV   AX,VALASC
        MOV   YASCII,AX
        CALL  DESPLIEGA ;SERVICIO 40 DE LA INT 21
        JMP   OTRO
ERROR:   LEA   DX,MEN1
        CALL  PRINTF
EXIT:    CALL  F_MOUSE
        MOV   FPP,07H

```

```

        MOV     YXI,00H
        MOV     YXF,184FH
        CALL    TEXTBACKGROUND
        MOV     AX,4C00H
        INT     21H
        RET
MAIN     ENDP

I_MOUSE  PROC    NEAR
        MOV     AX,00
        INT     33H
        RET
I_MOUSE  ENDP

M_MOUSE  PROC    NEAR
        PUSH    AX
        MOV     AX,01H
        INT     33H
        POP     AX
        RET
M_MOUSE  ENDP

C_MOUSE  PROC    NEAR
        PUSH    AX
        MOV     AX,04H
        INT     33H
        POP     AX
        RET
C_MOUSE  ENDP

F_MOUSE  PROC    NEAR
        PUSH    AX
        MOV     AX,02H
        INT     33H

        POP     AX
        RET
F_MOUSE  ENDP

C_CURSOR PROC    NEAR
        MOV     AH,02
        MOV     BH,00
        MOV     DH,00
        MOV     DL,25
        INT     10H
        RET
C_CURSOR ENDP

P_MOUSE  PROC    NEAR
        ;La subrutina devuelve XB,YB

SAL3:    MOV     AX,03 ;RETORNA BX = Edo DE LOS BOTONES Y
        INT     33H ;RETORNA CX= COORD X, DX= COORD Y
        CMP     BX,01 ; SI SE PULSO BOTON IZQ.
        JE      SAL1
        MOV     AX,CX ; AUXILIAR DE LA COORDENADA HOR
        MOV     CL,3
        SHR     AX,CL ;DIVIDE COORDENDA X / 8

```

```

                SHR    DX,CL          ;DIVIDE COORDENADA Y / 8
                MOV    CX,AX          ;RECUPERANDO LA COORDENADA PARA MODO
TEXTO
                CMP    CX,XB          ; SI EXISTE MOVIMIENTO ?
                JNE    SAL2           ; SI
                CMP    DX,YB
                JE     SAL3           ;NO
SAL2:          MOV    XB,CX          ;REFresco DE COORDENADAS
                MOV    YB,DX
SAL1:          RET
P_MOUSE       ENDP

CONV  PROC  NEAR
        MOV    VALASC,2020H        ;LIMPIANDO VALASC
        MOV    CX,10               ; FACTOR DE DIVISIÓN
        LEA    SI,VALASC+1
        CMP    AX,CX               ;AX CONTIENE LA COORDENADA
        JB     C1                  ; SI ES MENOR
        DIV    CL                   ; SI MAYOR DIVIDE
        OR     AH,30H              ;RESIDUO
        MOV    [SI],AH
        DEC    SI
C1:      OR     AL,30H              ;COCIENTE
        MOV    [SI],AL
        RET
CONV  ENDP

DESPLIEGA  PROC  NEAR
                ; Servicio 40h :manejador de archivo para despliegue en pantalla
                MOV    AH,40H
                MOV    BX,01        ;Número de dispositivo: 1 = pantalla
                MOV    CX,11        ;Total de caracteres a desplegar
                LEA    DX,DESPDATO  ; Area a desplegar
                INT    21H
                RET
DESPLIEGA  ENDP

INITTEXT  PROC  NEAR
        PUSH    AX
        MOV     AH,00H
        MOV     AL,03H
        INT     10H
        POP     AX
        RET
INITTEXT  ENDP

PRINTF  PROC  NEAR
        PUSH    AX
        MOV     AH,09H
        INT     21H
        POP     AX
        RET
PRINTF  ENDP

TEXTBACKGROUND  PROC  NEAR
                PUSH    AX
                PUSH    BX

```

```

        PUSH    CX
        PUSH    DX
        MOV     AX,0600H
        MOV     BH,FPP
        MOV     CX,YXI
        MOV     DX,YXF
        INT     10H
        POP     DX
        POP     CX
        POP     BX
        POP     AX
        RET
TEXTBACKGROUND ENDP

GOTOXY  PROC NEAR
        PUSH    AX
        PUSH    BX
        PUSH    DX
        MOV     AH,02H
        MOV     BH,00H
        MOV     DX,YXI
        INT     10H
        POP     DX
        POP     BX
        POP     AX
        RET
GOTOXY  ENDP

END      MAIN

```