**Descripción**: MDIO.H utiliza la Interrupción 0x33 – 33h para controlar el driver del mouse este opera en la arquitectura x86.

Las funciones se introducen en el registro AX, parra ello debemos vincularlo con los registros de la CPU

Para escribir un datos debemos hacer un union de los registros (REGS).

unión REGS in, out; || unión REGS registros;

in.x.ax = 0x0084 🡪 Queremos almacenar en la instrucción ax el valor 0x0084

Para lanzar la interrupción, usaremos la función de la librería dos.h int86, que se define en Turbo C como:

int int86 (int numero\_interrupcion, union REGS \*registros\_entrada, union REGS \*registros\_salida);

**La estructura interna**.

El lenguaje C nos proporciona una serie de estructuras en la librería dos.h que nos

Permiten escribir en los registros de la CPU. Estas son:

struct WORDREGS {

unsigned int ax;

unsigned int bx;

unsigned int cx;

unsigned int dx;

unsigned int si;

unsigned int di;

unsigned int cflag;

unsigned int flags;

}

struct BYTEREGS {

unsigned char al;

unsigned char ah;

unsigned char bl;

unsigned char bh;

unsigned char cl;

unsigned char ch;

unsigned cha.r d1;

unsigned cha.r dh;

}

union REGS {

struct WORDREGS x;

struct BYTEREGS h;

}

|  |  |  |  |
| --- | --- | --- | --- |
| **Función hexadecimal** | **Que realiza** | **interrupción** | **Return** |
| 00h | Resetea el mouse | AX = 0 | AX > 0 ‘si se ha detectado mouse’  AX = 0 ‘no se ha detectado el mouse’ |
| 01h | Muestra el mouse | AX = 1 | N/A |
| 02h | Oculta el mouse | AX = 2 | N/A |
| 03h | Lee el driver del mouse | AX = 3 | BX = 0 ‘no se ha pulsado ninguna tecla’  BX = 1 ‘se ha pulsado el botón izquierdo’  BX = 2 ‘se ha pulsado el botón derecho’  BX = 4 ‘se ha pulsado el botón centra’  CX = H ‘posición horizontal’  CX = V ‘posición vertical’ |
| 04h | Establece la posición x e y del mouse | AX = 4  DX = X ‘posición x’  CX = Y ‘posición y’ | N/A |
| 07h | Rango de desplazamiento de X – Horizontal | AX = 7  DX = H ‘mínimo’  CX = H ‘máximo’ | N/A |
| 08h | Rango de desplazamiento de Y – Vertical | AX = 8  DX = V ‘mínimo’  CX = V ‘máximo’ | N/A |

**FUNCIONES DE LA LIBRERÍA MDIO.H**

* **Funciones Principales**
  + startmd();
  + showmd();
  + hidemd();
* **Obtener atributos del mouse**
  + getmd\_all();
  + getmd\_pos();
  + getmd\_press();
  + getmd\_x();
  + getmd\_y();
* **Introducir atributos del mouse**
  + setmd\_pos();
  + setmd\_x();
  + setmd\_y();
* **Otros atributos**
  + delmc\_x();
  + delmc\_y();

**Funcionamiento de las funciones.**

**FUNCIONES PRINCIPALES**

* **startmd**

Inicializa el ratón, devuelve el estado del mouse.

SI existe el mouse devolverá un 1 si no existe devolver un 0;

int startmd();

* **showmd**

Muestra de forma gráfica o no grafica el mouse;

void showmd();

* **hidemd**

Oculta el mouse de la pantalla;

void hidemd();

**FUNCIONES DE EXTRACION DE DATOS**

* **getmd\_all**

Devuelve el click del mouse, posición X y posición Y;

void getmd\_all(int\*,int\*,int\*);

* **getmd\_pos**

Devuelve la posición X y posición Y;

void getmd\_pos(int\*,int\*);

* **getmd\_press**

Devuelve el click del mouse;

void getmd\_press(int\*);

* **getmd\_x**

Devuelve solo la posición X;

void getmd\_x(int\*);

* **getmd\_y**

Devuelve solo la posición X;

void getmd\_y(int\*);

**FUNCIONES DE INSERCION DE DATOS**

* **setmd\_pos**

Se introduce la nueva posición X e Y del mouse;

void setmd\_pos(int\*,int\*);

* **setmd\_x**

Se introduce la nueva posicion X;

void setmd\_x(int\*);

* **setmd\_y**

Se introduce la nueva posicion Y;

void setmd\_y(int\*);

**FUNCIONES DE DELIMITACION DEL MOUSE**

* **delmd\_x**

Limita posicion X con minimo y maximo;

void delmd\_x(int\*,int\*);

* **delmd\_y**

Limita posicion X con minimo y maximo;

void delmd\_y(int\*,int\*);