

## **Archivo KEYBOARD.ASM**

### **Subrutina getScanCode**

*Recibe el Scancode del puerto del teclado.*

LLAMADA:

call getScanCode

DEVUELVE:

AL = ScanCode

EFFECTO: Devuelve el código escaneado en el puerto 60h de la interfaz de E/S, perteneciente al teclado.

### **Subrutina SCtoASCII**

*Convierte de SCANCODE del teclado a ASCII.*

LLAMADA:

AL = ScanCode

call SCtoASCII

DEVUELVE:

AL = código ASCII

AH =  $\overline{\text{Pressed}}$  /Released

EFFECTO: Se realiza una conversión del SCANCODE al código ASCII y se devuelve este mismo código junto con un valor que representa si la tecla esta presionada o se ha soltado.

## **Archivo TIMER.ASM**

### **Subrutina TimRst**

*Resetea el contador de clocks del sistema.*

LLAMADA:

call TimRst

DEVUELVE:

Nada.

EFFECTO: Pone en cero al contador de clocks del sistema, el cual comenzará a contar nuevamente sin necesidad de enviarle otra instrucción.

### **Subrutina TimSet**

*Setea el contador de clocks del sistema utilizando el sistema sexagesimal.*

LLAMADA:

PUSH1 = Horas

PUSH2 = Minutos

PUSH3 = Segundos

PUSH4 = Centésimos

call TimSet

DEVUELVE:

Nada.

EFFECTO: Setea el valor del contador de clocks del sistema según horas, minutos, segundos y centésimas.

### **Subrutina TimSetC**

***Setea el contador de clocks del sistema utilizando cantidad de clocks.***

**LLAMADA:**

PUSH1 = Parte más significativa del contador

PUSH2 = Parte menos significativa del contador

call TimSetC

**DEVUELVE:**

Nada.

**EFFECTO:** Setea el valor del contador de clocks del sistema según cantidad de clocks. En cada segundo hay 18,2 ciclos.

### **Subrutina getTim**

***Recibe el valor del contador de clocks del sistema utilizando el sistema sexagesimal.***

**LLAMADA:**

call getTim

**DEVUELVE:**

CH = Horas

CL = Minutos

DH = Segundos

DL = Centésimas

**EFFECTO:** Devuelve el tiempo actual del contador en horas, minutos, segundos y centésimas.

### **Subrutina getTimC**

***Recibe el Scancode del puerto del teclado utilizando cantidad de clocks.***

**LLAMADA:**

call getTimC

**DEVUELVE:**

CX = Parte más significativa del contador

DX = Parte menos significativa del contador

**EFFECTO:** Devuelve el tiempo actual del contador en ciclos del clock. Se producen 18,2 ciclos por segundo.

### **Subrutina cmpMicroSec**

***Realiza una comparación de cumplimiento del tiempo transcurrido.***

**LLAMADA:**

AX = Microsegundos

call cmpMicroSec

**DEVUELVE:**

Una comparación lista para hacer un salto condicional.

**EFFECTO:** Si el tiempo pasado en el contador de clocks del sistema superó el tiempo recibido en el registro en AX entonces podrá utilizar un JE para saltar a otra parte del programa, en caso contrario el JE será salteado. Para realizar esto el proceso modifica los flags directamente.