

LABORATORIO DE SISTEMAS DIGITALES CON
MICROPROCESADORES

PRACTICA 4

TIEMPO

Arturo Reyes Santana

UNIVERSIDAD AUTÓNOMA METROPOLITANA

Objetivos:

- Administrar eventos con bases en el tiempo mediante el acceso a la localidad 0040:006C del BIOS.
- Emplear las interrupciones 1Ah – función 2 para obtener la hora, minutos y segundos de la PC.
- Emplear memoria de video para desplegar información.

Trabajo para desarrollar:

Mostrar la hora, minutos y segundos que tenga registrado reloj de su equipo de computo de acuerdo con las siguientes consideraciones.

- El programa deberá mostrar, en la parte central de la pantalla y durante 60 segundos , la hora en el formato hh:mm:ss dentro de un recuadro con líneas dobles.
- Los caracteres estarán en color amarillo con fondo azul.
- Se debe emplear memoria de video para la impresión en pantalla.
- El temporizador debe estar con base en el acceso a la localidad 0040:006C del BIOS.
- La “hora actual” debe ser obtenido por medio de la función 2 INT 1Ah.

Entrega:

Los archivos serán enviados al mail del profesor: InicialesApellidosInicialNombre_P4.ASM para el código y InicialesApellidosInicialNombre_P4.pdf (Por ejemplo, Valle Alba Juan Luis: VAJL_P4.ASM VAJL_P4.pdf)

Desarrollo:

```
title RSA_P4.ASM
.model small
.stack 100h

.code
main proc
    call reloj
    mov ah, 4ch
    int 21h
main endp
;###Procesos###
end
```

En el proceso de lo usaremos para inicializar el reloj, también nos servirá para poder adornar la división de las horas, minutos y segundos.

Algo que nos logramos comprender es que al momento de ejecutar el programa no se puede mostrar en una sola celda de video si es sucesiva a un numero por lo que fue necesario tener que colocar un espacio al principio de cada separador para darle prioridad a la apariencia.

Para estos procesos se establecerán en un orden específico y también es un proceso infinito, dado que no tenemos la intención de que el reloj se detenga.

Por lo que las horas seguirán hasta que cerremos la máquina virtual.

Para el desarrollo de la practica usamos un total de 4 procesos iniciación con el proceso al que llamamos "reloj".

```
reloj proc
    push ax
    push bx
    push cx
    push dx
    push es

    ;Modo texto y modo grafico
    mov ax, 0b800h
    mov es, ax

    ;Separadores
    mov di,7C8h           ;Posicion central
    mov es:[di],20h       ;Character
    mov es:[di+1],00011110b ;Atributos

    ;Separadores
    mov di,7CAh           ;Posicion central
    mov es:[di],3Ah       ;Character
    mov es:[di+1],00011110b ;Atributos

    ;Separadores
    mov di,7D0h           ;Posicion central:
    mov es:[di],20h       ;Character
    mov es:[di+1], 00011110b ;Atributos

    ;Separadores
    mov di,7D2h           ;Posicion central:
    mov es:[di],3Ah       ;Character
    mov es:[di+1], 00011110b ;Atributos

    horaVid:
        call hora
    loop horaVid

    pop es
    pop dx
    pop cx
    pop bx
    pop ax
    ret
reloj endp
```

```

hora proc
    push ax
    push bx
    push cx
    push dx
    push es

    ;Modo texto y modo grafico
    mov ax, 0b800h
    mov es, ax

    ;Obtenemos hora
    mov ah,2
    int 1ah
    ;ch= [h]

    mov al,ch
    shr al,01
    shr al,01
    shr al,01
    shr al,01
    add al,30h

    mov di,7C4h          ;Posicion central:
    mov es:[di], al      ;Caracter
    mov es:[di+1], 00011110b;Atributos

    mov al, ch
    and al,0Fh
    add al,30h

    mov di,7C6h          ;Posicion central:
    mov es:[di], al      ;Caracter
    mov es:[di+1], 00011110b;Atributos

minVid:
    call minutos
    ;Obtenemos minutos
    mov ah,2
    int 1ah
    ;cl = [m]
    cmp cl,00h
    je sigHora
    loop minVid

sigHora:
    pop es
    pop dx
    pop cx
    pop bx
    pop ax
    ret
hora endp

```

De igual forma se realiza lo mismo con los minutos este retornara el proceso siguiente al momento de detectar que los segundo muestran el valor de 00 en los segundos en lugar de los minutos como en el proceso anterior.

La única diferencia entre el proceso anterior y este que cambiaria el registro a analizar u los las posicione de video a mostrar .

En la función hora lo que vamos a hacer es manipular las instrucciones que se nos otorgaron en clase para hacer un recorrido de bits y poder obtener la parte decimal y la parte de unidades.

Una vez imprimimos nuevamente entramos a otro ciclo dende a partir del valor que se tenga en los minutos de en “cl” retornara si se detecta un 00 en los minutos.

```

minutos proc
    push ax
    push bx
    push cx
    push dx
    push es

    ;Modo texto y modo grafico
    mov ax, 0b800h
    mov es, ax

    ;Obtenemos minutos
    mov ah,2
    int 1ah
    ;cl= [m]

    mov al,cl
    shr al,01
    shr al,01
    shr al,01
    shr al,01
    add al,30h

    mov di,7CC          ;Posicion central:
    mov es:[di], al     ;Caracter
    mov es:[di+1], 00011110b;Atributos

    mov al,cl
    and al,0Fh
    add al,30h

    mov di,7CEh          ;Posicion central:
    mov es:[di], al     ;Caracter
    mov es:[di+1], 00011110b;Atributos

segVid:
    call segundos
    ;Obtenemos minutos
    mov ah,2
    int 1ah
    ;dh = [s]
    cmp dh,00h
    je sigMin
    loop segVid

sigMin:

    pop es
    pop dx
    pop cx
    pop bx
    pop ax
    ret
minutos endp

```

```

segundos proc
    push ax
    push bx
    push cx
    push dx
    push es

    ;Modo texto y modo grafico
    mov ax, 0b800h
    mov es, ax

    ;Obtenemos minutos
    mov ah, 2
    int 1Ah
    ;dh= [s]

    mov al, dh
    shr al, 01
    shr al, 01
    shr al, 01
    shr al, 01
    add al, 30h

    mov di, 7D4h          ;Posicion central:
    mov es:[di], al       ;Caracter
    mov es:[di+1], 00011110b;Atributos

    mov al, dh
    and al, 0Fh
    add al, 30h

    mov di, 7D6h          ;Posicion central:
    mov es:[di], al       ;Caracter
    mov es:[di+1], 00011110b;Atributos

    call retardo

    pop es
    pop dx
    pop cx
    pop bx
    pop ax
    ret
segundos endp

```

Para poder interpretar el timer chip lo que hacemos es definir un contador que constantemente ira registrando los 18 pulsos y que al momento de registrarlo lo que nos dará será el retorno para poder seguir actualizando la fecha en los siguientes segundos y no tener que ir realizando peticiones de fecha en los registros.

Para el proceso de los segundos lo que hicimos fue hacer lo mismo que en los procesos anteriores, le asignamos las direcciones de video que ocuparíamos, el ir constantemente verificando que los segundos se estén actualizados antes de mostrarlo en pantalla.

Lo que cambiaria es que ya no es necesario tener que implementar un ciclo si no lo que hacemos es establecer un contador que fue mostrado en clase, donde según la teoría lo que hacemos es interpretar un timer chip.

En contexto todas las PC's tiene un timer chip, el timer chip se inicializa para pulsar aproximadamente 18.2 veces por segundo, los pulsos son contados por el sistema operativo y forman el contador hora-minutos-segundos del BIOS, un valor de cuatro bytes manteniendo en el área de datos en el BIOS en la posición 0040:006C.

Los timer chips pulsan al mismo ritmo en todas las maquinas, no importa que tan rápido corra el procesador de la máquina. Para ello, emplean un divisor de frecuencia. La tasa se desvía solamente cuando el software reprograma el timer chip, o cuando alguien cambia el cristal del reloj de la computadora para hacer que la maquina corra más rápido además de que siempre el timer chip pulsa y se invoca una interrupción de tiempo.

```

retardo proc
    push ax
    push bx
    push cx
    push dx

    ;Definimos el valor de es
    mov ax, 40h          ;(60)_10
    mov es, ax

    mov bl, es:[6Ch]
    ;Sumamos el valor que tenemos 18 pulsos es decir un segundo
    add bl, 18

    tryagain:
        ;Se comparamos hasta realizar 18 pulsos
        cmp bl, es:[6Ch]
        jne tryagain

    pop dx
    pop cx
    pop bx
    pop ax
    ret
retardo endp

```

Conclusión:

Se logro administrar eventos con bases en el tiempo mediante el acceso a la localidad 0040:006C del BIOS para poder establecer los tiempos que damos por segundo para ir intercambiando la actualización de la fecha usando la interrupcio1Ah – función 2 para obtener la hora, minutos y segundos de la PC además de que empleamos fue memoria de video para desplegar información.

Los únicos inconvenientes que tuvimos fueron que al momento de mostrar los detalles como los separadores tuvimos que separarlos con dos casillas, además de que el reloj es infinito no se puede detener el reloj al menos de que detengamos toda la máquina.

