

Práctica N°3
PROGRAMAS RESIDENTES EN MEMORIA
ARQUITECTURA BASICA DE UNA COMPUTADORA

OBJETIVOS:

- Utilizar los mecanismos para generar un programa residente en memoria (TSR)
- Identificar el manejo de Interrupciones con el 8086
- Diseñar un programa utilizando interrupciones por software y por hardware.

TRABAJO A DESARROLLAR:

PARTE 1:

Escribir un programa que realice la función de un reloj.

Para lo cual se debe realizar un procedimiento o subrutina que muestre en la pantalla inicialmente la hora de la computadora (el valor de horas, minutos y segundos) de la siguiente forma 05:22:00, a medida que transcurra el tiempo se deberán ir incrementando los segundos y los minutos.

Primero se deberá desarrollar una subrutina que se encargue de determinar cuando haya transcurrido un segundo y actualizar los registros correspondientes a segundos, minutos y horas.

Posteriormente se deberá transformar la subrutina para ser accedida como una interrupción por software (se puede utilizar una interrupción entre la 4BH y la 7FH) y finalmente se creará otra versión en la que la subrutina sea accedida por hardware.

Para esta etapa se recomienda usar la INT 1CH del BIOS la cual es la interrupción de un timer que automáticamente interrumpe cada 54.9254 ms (casi 18.2 veces cada segundo). De tal manera que se puede inicializar un contador que por cada interrupción se incremente en 1, y cada que el contador llegue a 18 se deberá incrementar los segundos (y los minutos cada que el contador de segundos llegue a 60), e inicializar el contador a ceros.

También se puede utilizar la interrupción 8, en cuyo caso se deberá realizar un cambio de vectores (en la tabla de vectores de interrupción), en donde se sustituirá el vector actual por la dirección de la rutina del reloj.

Los valores originales de la interrupción 8 deberán relocarse en el vector correspondiente a la interrupción 61H (en lugar de la 61H se puede utilizar otra interrupción entre la 4BH y la 7FH), y al final de la interrupción del reloj se deberá llamar a la interrupción 61H para permitir que el sistema no se altere, ya que esta interrupción se utiliza para el reloj de tiempo real de la PC.

PARTE 2

Realice los cambios apropiados para dejar residente el programa, utilizando el servicio 31H de la Interrupción 21H.

Nota: Puede encontrar útiles las siguientes llamadas a Funciones de DOS

INT 21H con servicio (AH):

25H. Establecer la dirección de la tabla de interrupciones

AL: Tipo de Interrupción

DS: DX Dirección de la rutina de interrupción

35H. Obtener dirección del vector de interrupción

AL: tipo de interrupción

ES: BX Dirección del vector de interrupción presente.

31H. Termina proceso y permanece residente.

AL: Código de salida (puede ignorarse por el momento)

DX: cantidad de memoria que será conservada residente.

**CUESTIONARIO
PRACTICA 3**

- 1.- En la primera versión del programa, ¿Cómo funciona una subrutina (procedimiento)? ¿Qué tipos de instrucciones se deben utilizar para acceder y regresar al punto donde fue invocada?
- 2.- ¿Cómo funciona una interrupción por software? Explique ampliamente.
- 3.- ¿Qué cambios se realizaron en el programa para cambiar una subrutina llamada con CALL por una subrutina de atención a interrupción por software?
- 4.- ¿Por qué se requiere un vector de interrupción?
- 5.- ¿Cómo se escribe un vector de interrupción? Explique e indique fragmento de código.
- 6.- ¿Para qué se utilizaron los contadores anidados en la práctica?
- 7.- ¿Cómo funcionan los contadores anidados? Explique.
- 8.- Explique detalladamente cómo funciona una interrupción por hardware
- 9.- ¿Cuál es la diferencia de una interrupción por software y una interrupción por hardware? (Explique e Indique los fragmentos de código con sus respectivos comentarios)
- 10.- ¿Qué se requiere para dejar residente un programa? Explique e indique fragmento de código.
- 11.- ¿Qué significa que un programa quede residente?
- 12.- De los cuatro programas realizados en la práctica ¿Cuáles de estos, cree que es el más preciso y cual no lo es? Explique.