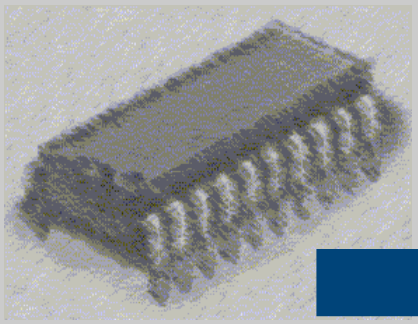
Enunciados de prácticas

Práctica 5. Registros y posiciones de memoria



Laboratorio de Estructura de Computadores

I. T. Informática de Gestión / Sistemas

Curso 2008-2009

PRÁCTICA 5: Registros y posiciones de memoria

Objetivos:

La práctica consiste en mostrar al alumno como se almacenan los datos en memoria y la forma de acceder a estas posiciones de memoria.

Medios:

Para la programación se emplea el Microsoft Assembler 5.1. El software viene acompañado de un programa ensamblador (MASM) y su correspondiente enlazador (LINK) que generará un fichero ejecutable (.EXE) que será el que podrá ser ejecutado paso a paso mediante el simulador o ejecutado de golpe como se hace habitualmente.

ACTIVIDADES PARA LA PRÁCTICA 5

En alguna de las actividades siguientes puede haber alguna errata en el código con el fin de que se aprendan a interpretar los mensajes de error del ensamblador y cómo corregir un programa en ensamblador.

Núm Ejer.	Ejercicio

1	Escribe, ensambla y ejecuta con el Code View el siguiente código:
	dosseg
	.model small
	.stack 100h
	.data
	  Terminar EQU 4Ch
	.code
	Inicio:
	  mov ax, @data
	  mov ds, ax
	
	  mov al, 10h
	  mov cl, 4
	  shl al, cl
	
	  mov ah, 4Ch
	  int 21h
	end Inicio

2	¿Qué efecto produce el programa anterior en el registro AX?. Prueba a cambiar la instrucción shl al, cl por la instrucción shr al, cl y realiza la actividad de nuevo. ¿Qué ocurre?

Núm Ejer.	Ejercicio

3	Introduce y ensambla el código siguiente:
	dosseg
	.model small
	.stack 100h
	.data
	  Texto DB "Introduce una frase de como mucho 8 caracteres:
	",13,10,'$'
	  MaximoMas1 DB 9
	  CaracteresLeidos DB 0
	  Cadena DB 9 DUP (0)
	  Otralinea DB 13,10,'$'
	
	.code
	Inicio:
	  mov ax, @data
	  mov ds, ax
	
	  mov ah, 9
	  lea dx, Texto
	  int 21h
	
	  mov ah, 0Ah
	  lea dx, MaximoMas1
	  int 21h
	
	  mov ah, 9
	  lea dx, Otralinea
	  int 21h
	
	  mov cl, MaximoMas1
	  xor SI, SI
	  mov ah, 2
	  bucle:
	  mov dl, Cadena[SI]
	  int 21h
	  inc SI
	  loop bucle
	
	  mov ah, 4Ch
	  int 21h
	END Inicio

4	Modifica el programa anterior para que la frase pueda almacenar 20 caracteres.

Núm Ejer.	Ejercicio
5	Introduce el código siguiente SUB DL, 20h dentro del bucle justamente detrás de la instrucción mov dl, Cadena[SI] ¿Qué función realiza el programa?

6	Introduce y ensambla el código siguiente:
	dosseg
	.model small
	.stack 100h
	.data
	Dir1 DB 10h, 20h, 30h, 40h
	Dir2 DB 50h, 60h, 70h, 80h
	..Resul DB ?, ?, ?, ?
	.code
	Inicio:
	mov ax, @data
	mov ds, ax
	
	mov cx, 4
	mov bx, 0
	clc
	Otro:
	mov al, Dir1[bx]
	adc al, Dir2[bx]
	mov Resul[bx], al
	inc bx
	loop Otro
	
	mov ah, 4Ch
	int 21h
	END Inicio

7	¿Qué hace el código anterior? Visualiza las variables Dir1, Dir2 y Resul (Ej. En la línea de comandos del Code View **> Wb Dir1 l 4**) Cambia el contenido de memoria directamente en el Code View y modifica los datos a sumar (Ej. En la línea de comandos del Code View >**Eb Dir1** y pulsar enter, ir modificando los datos dando a la barra espaciadora)
8	Modifica el programa anterior para que sume palabras de 16 bits.
9	Introduce, ensambla y ejecuta con el Code View el código siguiente, presta atención especial al valor que tendrá el registro BL al final del programa: dosseg .model small.

(Continúa)

Núm Ejer.	Ejercicio

9	Introduce, ensambla y ejecuta con el Code View el código siguiente, presta atención especial al valor que tendrá el registro BL al final del programa:
	dosseg
	.model small. **(Continúa)**
	(Continúa)
	.stack 100h
	.data
	Texto DB 'Introduce un número hexadecimal de cómo mucho dos cifras entre 0 y 9$'
	.code
	Inicio:
	mov ax, @data
	mov ds, ax
	
	mov ah, 9
	lea dx, Texto
	int 21h
	xor bl, bl
	mov ah, 1
	int 21h
	
	mov cl, 4
	mov bl, al
	sub bl, 30h
	shl bl, cl
	
	int 21h
	sub al, 30h
	add bl, al
	
	mov ah, 4Ch
	int 21h
	END Inicio

10	Modifica la actividad anterior para que lea un número de 4 cifras hexadecimales.
11	Escribe, ensambla y ejecuta el código siguiente: dosseg .model small .stack 100h .data Numero DB 17h .code Inicio: mov ax, @data

Núm Ejer.	Ejercicio

11	Escribe, ensambla y ejecuta el código siguiente:
	dosseg
	.model small
	.stack 100h
	.data
	Numero DB 17h
	.code
	Inicio:
	mov ax, @data
	mov ds, ax
	
	mov bh, Numero
	mov cl, 4
	shr bh, cl **(Continúa)**
	mov ah, 2 **(Continúa)**
	mov dl, bh
	add dl, 30h
	int 21h
	
	(Continúa)
	(Continúa)
	mov dl, Numero
	and dl, 0Fh
	add dl, 30h
	int 21h
	
	mov ah, 4Ch
	int 21h
	END Inicio

12	¿Que hace el programa anterior?
13	Cambia el número 17h por los números 34h y 0Fh. ¿Qué ocurre?

PRÁCTICA 5:
Registros y posiciones de memoria

Realizar un programa en ensamblador que pida introducir por teclado un número binario de 16 bits y que
- Suponiendo que el nº introducido está en complemento a 1 almacenar en una variable llamada varc1 dicho número cambiado de signo.
- Suponiendo que el nº introducido está en signo-magnitud almacenar en una variable llamada varsm dicho número cambiado de signo.