

José Daniel Molina Galindo

Carnet: 1007420

MICROPROGRAMACIÓN



## LABORATORIO 5

### Ejercicio 1: Conteste las siguientes preguntas:

1. Cuando se genera el ejecutable del archivo "Ejemplo2.asm", ¿cuál es la advertencia que se muestra en pantalla? ¿Por qué muestra esa advertencia?

El problema o la advertencia es del WARNING: no stack y se muestra porque el código no tiene el punto stack ósea que no esta definido en el programa.

2. Modificado el programa para que no muestre la advertencia, ¿cuál es el resultado del programa, es decir, por qué se imprime ese carácter y no un 30?

En términos 5A se representa o se muestra en hexadecimales y que se corresponde con el símbolo de la letra Z.

La motivación para realizar la suma del contenido de los registros, tanto el AL con el valor 30h original como los 30h adicionales introducidos en la nueva entrada, conduce a un total de 60h como resultado.

### Ejercicio 2: Utilización del Modo "DEBUG"

Comandos del Modo "DEBUG":

- N Nombrar un programa.
- L Se encarga de cargar el programa.
- U "Desensamblar" código máquina y pasarlo a código simbólico.
- A Ensamblar instrucciones simbólicas y pasarlas a código máquina.
- D Mostrar el contenido de un área de memoria.
- E Introducir datos en memoria, iniciando en una localidad específica.
- G Correr el programa ejecutable que se encuentra en memoria.
- P Proceder o ejecutar un conjunto de instrucciones relacionadas.
- Q Salir de la sesión con DEBUG.
- R Mostrar el contenido de uno o más registros.
- T Rastrear la ejecución de una instrucción.
- W Escribir o grabar un programa en disco.

Utilizando el Modo “DEBUG” de DOS cargue el programa “Ejemplo2.exe” y responda las siguientes preguntas:

1. ¿En qué dirección de memoria inicia el código del programa?

0B6E:0000 B8700B MOV AX,0B70

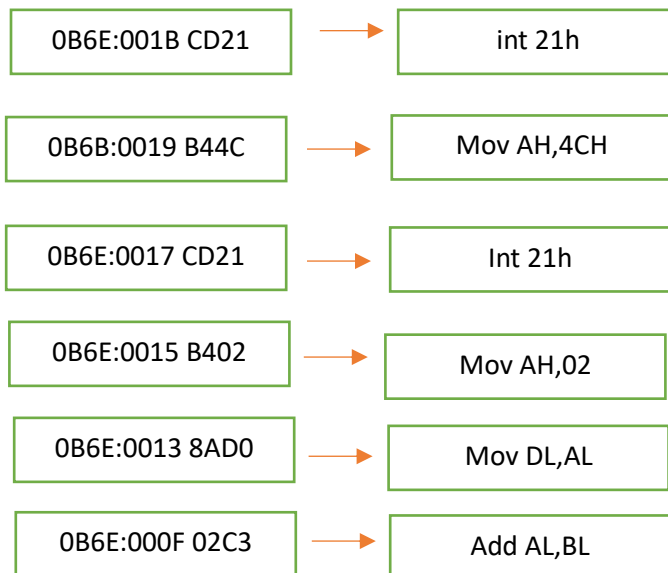
2. ¿En qué dirección de memoria termina el código del programa?

0B6E:001B CD21 INT 21

3. Aparecen los comentarios en pantalla ¿Sí? ¿No? ¿Por qué?

No aparecen los comentarios en la pantalla ya que lo que en realidad es importante es mostrar las instrucciones realizadas en el programa los MOV.

4. Para cada una de las instrucciones del programa, escriba la dirección de memoria que tiene asignada.



5. ¿Cuál es la dirección del segmento de código?

0B6E

6. Antes de iniciar la ejecución por pasos del programa, ¿cuáles son los valores de los registros de propósito general?

AX = 0B70 BX = 0000

7. El valor del IP, ¿coincide con la dirección de inicio del programa?

Si puede coincide con mi sistemas y la IP es 0003

8. Utilice el comando para el rastreo instrucción por instrucción y, por cada línea del código, escriba el contenido de los registros internos del CPU.

SCRENS

```
AX=0015 BX=0000 CX=001D DX=0000 SP=0400 BP=0000 SI=0000 DI=0000
DS=0B70 ES=0B5E SS=0B70 CS=0B6E IP=000D NV UP EI PL NZ NA PO NC
0B6E:000D B315 MOV BL,15
-T

AX=0015 BX=0015 CX=001D DX=0000 SP=0400 BP=0000 SI=0000 DI=0000
DS=0B70 ES=0B5E SS=0B70 CS=0B6E IP=000F NV UP EI PL NZ NA PO NC
0B6E:000F 02C3 ADD AL,BL
-T

AX=002A BX=0015 CX=001D DX=0000 SP=0400 BP=0000 SI=0000 DI=0000
DS=0B70 ES=0B5E SS=0B70 CS=0B6E IP=0011 NV UP EI PL NZ NA PO NC
0B6E:0011 0430 ADD AL,30

AX=025A BX=0015 CX=001D DX=005A SP=0400 BP=0000 SI=0000 DI=0000
DS=0B70 ES=0B5E SS=0B70 CS=0B6E IP=0017 NV UP EI PL NZ NA PE NC
0B6E:0017 CD21 INT 21
-T

AX=025A BX=0015 CX=001D DX=005A SP=03FA BP=0000 SI=0000 DI=0000
DS=0B70 ES=0B5E SS=0B70 CS=00A7 IP=107C NV UP DI PL NZ NA PE NC

AX=005A BX=0015 CX=001D DX=0000 SP=0400 BP=0000 SI=0000 DI=0000
DS=0B70 ES=0B5E SS=0B70 CS=0B6E IP=0013 NV UP EI PL NZ NA PE NC
0B6E:0013 8AD0 MOV DL,AL
-T

AX=005A BX=0015 CX=001D DX=005A SP=0400 BP=0000 SI=0000 DI=0000
DS=0B70 ES=0B5E SS=0B70 CS=0B6E IP=0015 NV UP EI PL NZ NA PE NC
0B6E:0015 B402 MOV AH,02
-T

AX=025A BX=0015 CX=001D DX=005A SP=0400 BP=0000 SI=0000 DI=0000
DS=0B70 ES=0B5E SS=0B70 CS=0B6E IP=0017 NV UP EI PL NZ NA PE NC
0B6E:0017 CD21 INT 21
```

### Ejercicio 3: Debug en RISC V

Utilizando el hola mundo proporcionado, en RISC V, deberá adjuntar una captura de pantalla sobre el modo debug y las direcciones de memoria del programa. Además, responda, ¿Dónde se detiene el programa la primera vez? ¿Por qué piensa que es así?

## CODIGO CON EL DEBUG

```

1 .global programa          #Directiva punto de arranque del programa
2 .data                    #Inicio segmento de datos
3 hola:
4     .string "Hola Mundo!!!"
5 .text                    #Inicio segmento de codigo
6 programa:
7
8     la a0, hola
9     li a7, 4             #Parametro para imprimir cadena de caracteres
10    ecall
11
12 finalizar:
13     li a7, 10            #Parametro para finalizar programa
14     ecall                #int 21h
15

```

**Text Segment**

Btpt	Address	Code	Basic	Source
0x00400000	0x00400000	0x00f01051	mulsq x10,0x00f010	5: la w0,hola
0x00400004	0x00400004	0x005051addi	x10,x10,0	
0x00400008	0x00400008	0x00400093	addi x17,w0,4	9: li a7, 4 #Parametro para imprimir cadena d...
0x0040000c	0x0040000c	0x00000007	jecall	10: ecall
0x00400010	0x00400010	0x00400093	addi x17,w0,10	13: li a7, 10 #Parametro para finalizar programa
0x00400014	0x00400014	0x00000007	jecall	14: ecall #int 2ih

**Data Segment**

Address	Value (-D)	Value (-C)	Value (-B)	Value (-C)	Value (-10)	Value (-14)	Value (-18)	Value (-1C)
0x10010000	0x140cf480	0x0ef76420	0x1211f464	0x00000001	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100A0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100C0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100E0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101A0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

☐ Hexadecimal Addresses 
 ☒ Hexadecimal Values 
 ☐ ASCII

**Messages** Run ID

```
Assembler: assembling C:\Users\jorda\Downloads\lab5.asm
Assembler: operation completed successfully.
```

Clear