**LABORATORIO NO. 02**

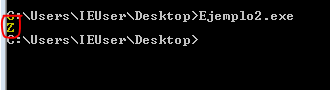
“Ensamblador, DEBUG y Sistemas Numéricos”

**Ejercicio 1: Utilización del Ensamblador y el Enlazador**

Utilizando los archivos “Ejemplo1.asm” y “Ejemplo2.asm” genere el código objeto y el programa ejecutable utilizando el Ensamblador “TASM” y el Enlazador “TLINK”.

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? *Falta declarar el segmento de pila en el código*
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? *imprime el símbolo de \**
3. Modifique el código del archivo “Ejemplo2.asm” y utilizando la tabla de códigos ASCII, imprima en pantalla una letra “Z”.



**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?



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

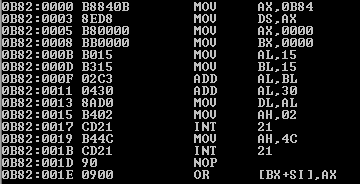


*Termina en la dirección 0B82:001B CD21*

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

*No aparecen los comentarios, debido a que el modo debug ignora los comentarios, ya que lo que interesa es ver el movimiento de los registros*

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



|  |  |
| --- | --- |
| Dirección de memoria | Instrucción |
| 0B82:0000 B8840B | Mov AX,@DATA |
| 0B82:0003 8ED8 | Mov DS,AX |
| 0B82:0005 B80000 | Mov AX,0000h |
| 0B82:0008 BB0000 | Mov BX,0000h |
| 0B82:000B B015 | Mov AL,15h |
| 0B82:000D B315 | Mov BL,15h |
| 0B82:000F 02C3 | Add AL,BL |
| 0B82:0011 0430 | Add AL, 30h |
| 0B82:0013 8AD0 | Mov DL,AL |
| 0B82:0015 B402 | Mov AH,02 |
| 0B82:0017 CD21 | Int 21h |
| 0B82:0019 B44C | Mov AH,4CH |
| 0B82:001E 0900 | int 21h |

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

CS=0B82

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

*AX: 0B84*

*BX: 0000*

*CX: 001D*

*DX: 0000*

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

*IP si coincide con el inicio de programa*

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













