**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:

* Cuando se genera el ejecutable del archivo “Ejemplo2.asm”, ¿cuál es la advertencia que se muestra en pantalla? ¿Por qué muestra esa advertencia? **La advertencia es que no hay stack. No está escrita la directiva .stack en el archivo .asm**
* 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? **Porque trabajamos con hexadecimal, al realizar la suma 15h + 15h nos da 2A que le corresponde el caracter "\*"**
* Modifique el código del archivo “Ejemplo2.asm” y utilizando la tabla de códigos ASCII, imprima en pantalla una letra “Z” **al sumarle 30h da como resultado 5A, a lo que corresponde el caracter "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:

* ¿En qué dirección de memoria inicia el código del programa?
* **0B11:0101 8976FA**
* ¿En qué dirección de memoria termina el código del programa?
* **0B11:011F 0B550A**
* Aparecen los comentarios en pantalla ¿Sí? ¿No? ¿Por qué?
* **No aparecen. Son ignorados al momento de crear el codigo ejecutable.**
* Para cada una de las instrucciones del programa, escriba la dirección de memoria que tiene asignada:

|  |  |
| --- | --- |
| Dirección de memoria | Instrucción |
| **0B11:0100 7880** | Mov AX,@DATA |
| **0B11:0102 3E** | Mov DS,AX |
| **0B11:0103 0C98** | Mov AX,0000h |
| **0B11:0105 017503** | Mov BX,0000h |
| **0B11:0108 E8CEE0** | Mov AL,15h |
| **0B11:010B 3C2E** | Mov BL,15h |
| **0B11:010D 7509** | Add AL,BL |
| **0B11:010F FE065199** | Mov DL,AL |
| **0B11:0113 C6065099FF** | Mov AH,02 |
| **0B11:0118 3C3F** | Int 21h |
| **0B11:011A 7503** | Mov AH,4CH |
| **0B11:011C 3400** | int 21h |

* ¿Cuál es la dirección del segmento de código? **0B11**
* Antes de iniciar la ejecución por pasos del programa, ¿cuáles son los valores de los registros de propósito general?



* El valor del IP, ¿coincide con la dirección de inicio del programa? **IP = 0100, Sí coincide.**
* Utilice el comando para el rastreo instrucción por instrucción y, por cada línea del código, escriba el contenido de los registros internos del CPU.























