**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?

‘Warning: no stack’

Se muestra la advertencia porque falta la instrucción ‘.STACK’ en el código del archivo .asm

1. 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?

Se imprime un \* porque el programa está sumando 15 + 15 pero en base hexadecimal, lo cual devuelve un valor de 2A el cual es el correspondiente al código ASCII del asterisco.

1. 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?

0B67:0000 B8690B

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

0B67:001F 20434F

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

No aparecen porque al pasar las instrucciones a lenguaje máquina se eliminan los comentarios y las líneas en blanco.

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

|  |  |
| --- | --- |
| Dirección de memoria | Instrucción |
| 0B67:0000 B8690B | Mov AX,@DATA |
| 0B67:0003 8ED8 | Mov DS,AX |
| 0B67:0005 B80000 | Mov AX,0000h |
| 0B67:0008 BB0000 | Mov BX,0000h |
| 0B67:000B B015 | Mov AL,15h |
| 0B67:000D B315 | Mov BL,15h |
| 0B67:000F 02C3 | Add AL,BL |
| 0B67:0015 8AD0 | Mov DL,AL |
| 0B67:0017 B402 | Mov AH,02 |
| 0B67:0019 CD21 | Int 21h |
| 0B67:001B B44C | Mov AH,4CH |
| 0B67:001D CD21 | int 21h |

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

0B67:0000 B8690B

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

Tienen los valores asignados previamente por algún programa ejecutado con anterioridad, o el valor de cero por dafault si no se han modificado.

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

El valor de IP al iniciar el programa es 0003 y coincide con la dirección de la segunda instrucción pero no con la primera.

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.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instrucción** | **AX** | **BX** | **CX** | **DX** | **SP** | **BP** | **SI** | **DI** | **DS** | **ES** | **SS** | **CS** | **IP** |
| **Mov AX,@DATA** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Mov DS,AX** | 0B69 | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B57 | 0B57 | 0B69 | 0B67 | 0003 |
| **Mov AX,0000h** | 0B69 | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B69 | 0B57 | 0B69 | 0B67 | 0005 |
| **Mov BX,0000h** | 0000 | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B69 | 0B57 | 0B69 | 0B67 | 0008 |
| **Mov AL,15h** | 0000 | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B69 | 0B57 | 0B69 | 0B67 | 000B |
| **Mov BL,15h** | 0015 | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B69 | 0B57 | 0B69 | 0B67 | 000D |
| **Add AL,BL** | 0015 | 0015 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B69 | 0B57 | 0B69 | 0B67 | 000F |
| **Mov CL,30h** | 002A | 0015 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B69 | 0B57 | 0B69 | 0B67 | 0011 |
| **Add AL,CL** | 002A | 0015 | 0030 | 0000 | 0400 | 0000 | 0000 | 0000 | 0B69 | 0B57 | 0B69 | 0B67 | 0013 |
| **Mov DL,AL** | 005A | 0015 | 0030 | 0000 | 0400 | 0000 | 0000 | 0000 | 0B69 | 0B57 | 0B69 | 0B67 | 0015 |
| **Mov AH,02h** | 005A | 0015 | 0030 | 005A | 0400 | 0000 | 0000 | 0000 | 0B69 | 0B57 | 0B69 | 0B67 | 0017 |
| **int 21h** | 025A | 0015 | 0030 | 005A | 0400 | 0000 | 0000 | 0000 | 0B69 | 0B57 | 0B69 | 0B67 | 0019 |