**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?

Porque no está creado el Stack

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?

Porque trabaja con hexadecimales entonces imprime el carácter al cual le corresponde el valor de 30 en hexadecimal y no el número.

1. Modifique el código del archivo “Ejemplo2.asm” y utilizando la tabla de códigos ASCII, imprima en pantalla una letra “Z”.

**Se le sumo 30h a lo previamente sumado para que se imprima 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?

BB600B

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

19A90008

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

No, porque al momento de usar el debug este no encontró errores

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

|  |  |
| --- | --- |
| Dirección de memoria | Instrucción |
| 0B42:0000 B8600B | Mov AX,@DATA |
| 0B42:0003 8ED8 | Mov DS,AX |
| 0B42:0005 B80000 | Mov AX,0000h |
| 0B42:0008 BB0000 | Mov BX,0000h |
| 0B42:000B B015 | Mov AL,15h |
| 0B42:000D B315 | Mov BL,15h |
| 0B42:000F 02C3 | Add AL,BL |
| 0B420011 B130 | Mov DL,AL |
| 0B42:0017 B402 | Mov AH,02 |
| 0B42:0019 CD21 | Int 21h |
| 0B42:001B B44C | Mov AH,4CH |
| 0B42:001D CD21 | int 21h |

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

Es el segmento 0B42

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

AX: 0000

BX: 0000

CX: 001F

DX: 0000

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

Si, coincide ya que se encuentra en 0000

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.

El contenido de los registros se describe en la tabla de abajo

(VER siguiente página)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instrucción | AX | BX | CX | DX | SP | BP | SI | DI | DS | ES | SS | CS | IP |  |
| Mov AX,@DATA | 0000 | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B44 | 0B32 | 0B44 | 0B42 | 0000 |  |
| Mov DS,AX | 0B44 | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B44 | 0B32 | 0B44 | 0B42 | 0003 |  |
| Mov AX,0000h | 0B44 | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B44 | 0B32 | 0B44 | 0B42 | 0005 |  |
| Mov BX,0000h | 0000 | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B44 | 0B32 | 0B44 | 0B42 | 0008 |  |
| Mov AL,15h | 0000 | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B44 | 0B32 | 0B44 | 0B42 | 000B |  |
| Mov BL,15h | 0015 | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B44 | 0B32 | 0B44 | 0B42 | 000D |  |
| Add AL,BL | 0015 | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B44 | 0B32 | 0B44 | 0B42 | 000F |  |
| Mov DL,AL | 002A | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B44 | 0B32 | 0B44 | 0B42 | 0011 |  |
| Mov AH,02 | 002A | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B44 | 0B32 | 0B44 | 0B42 | 0013 |  |
| Int 21h | 005A | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B44 | 0B32 | 0B44 | 0B42 | 0015 |  |
| Mov AH,4CH | 005A | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B44 | 0B32 | 0B44 | 0B42 | 0017 |  |
| int 21h | 025A | 0000 | 001F | 0000 | 0400 | 0000 | 0000 | 0000 | 0B44 | 0B32 | 0B44 | 0B42 | 107C |  |