**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?

Muestra advertencia que el archivo no tiene el segmento de stack. Es decir no esta definido el segmento en el archivo.

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 carácter, por que la suma se está haciendo en sistema de numeración hexadecimal, entonces el 15h no es lo mismo que 15 en sistema decimal. Por lo que al sumar 15h y 15h no es 30 sino el resultado es 2A, que al momento de convertir eso a su representación ascii, es el \* y no un 30.

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?

OCF7:0003 8ED8

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

0CF7:0022 0D0A49

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

No por que los comentarios no son parte del archivo, pues unicamente funcionan

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

|  |  |
| --- | --- |
| Dirección de memoria | Instrucción |
| OCF7: 0000 B8F90C | Mov AX,@DATA |
| OCF7: 0000 8ED8 | Mov DS,AX |
| OCF7: 0005 B80000 | Mov AX,0000h |
| OCF7: 0008 BB0000 | Mov BX,0000h |
| OCF7: 000B B015 | Mov AL,15h |
| OCF7: 000D B315 | Mov BL,15h |
| OCF7: 000F 02C3 | Add AL,BL |
| OCF7: 0013 8AD0 | Mov DL,AL |
| OCF7: 0015 B402 | Mov AH,02 |
| OCF7: 0017 CD21 | Int 21h |
| OCF7: 0019 B44C | Mov AH,4CH |
| OCF7: 001B CD21 | int 21h |

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

0CF7

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

AX = 0CF9 BX = 0000 CX= 001D DX = 0000

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

Si coincide pues la direcciòn de inicio es 0003 y el IP = 0003

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.





















