**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?



No se definio un segmento especifico para 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 imprime la representación en el codigo ASCII

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

Se suma el número 48 decimal

**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?

En la posicion 0000

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

En la posicion 001F

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

No, ya que unicamente procesa las instrucciones

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

|  |  |
| --- | --- |
| Dirección de memoria | Instrucción |
| 0C58:0000 | Mov AX,@DATA |
| 0C58:0003 | Mov DS,AX |
| 0C58:0005 | Mov AX,0000h |
| 0C58:0008 | Mov BX,0000h |
| 0C58:000B | Mov AL,15h |
| 0C58:000D | Mov BL,15h |
| 0C58:000F | Add AL,BL |
| 0C58:0015 | Mov DL,AL |
| 0C58:0017 | Mov AH,02 |
| 0C58:0019 | Int 21h |
| 0C58:001B | Mov AH,4CH |
| 0C58:001D | int 21h |

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

 Comienza en 0C58:0000 el cual es apuntado por CS

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



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

Si, tiene la posicion inicial 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.



