MARIO JOSE ROLDAN HERNANDEZ

1117517

**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?

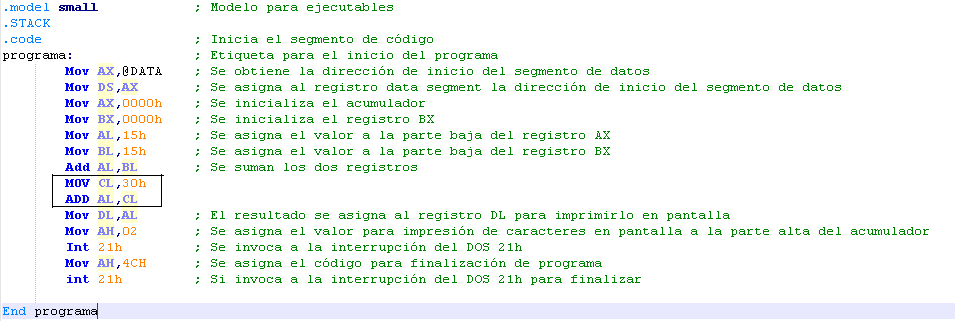
NO STACK

**Muestra un “Warning: No Stack” ya que en el archivo “Ejemplo2.asm” no se definió el segmento STACK. Se debe de agregar dicho segmento utilizando “.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 para imprimir un símbolo se debe de tener claro que uno NO imprime el numero en el que se está trabajando, sino que ese numero es la representación (ya sea en decimal o en hexadecimal) de un símbolo en código ASCII. Se diferencia entre hexadecimal utilizando la “h” al final del número.**

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



**MOV CL, 30H**

**ADD AL, CL**

**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?

**0CFC:0000 B8FE0C**

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

**0CFC:001F D22B**

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

**No, porque el ensamblador los ignora ya que se utilizo un comando especifico para que los ignore que es el “;” además, no se muestran ya que solo se muestran las INSTRUCCIONES que creamos en el archivo.asm.**

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

|  |  |
| --- | --- |
| Dirección de memoria | Instrucción |
| **0CFC:0000 B8FE0C** | Mov AX,@DATA |
| **0CFC:0003 8ED8** | Mov DS,AX |
| **0CFC:0005 B80000** | Mov AX,0000h |
| **0CFC:0008 BB0000** | Mov BX,0000h |
| **0CFC:000B B015** | Mov AL,15h |
| **0CFC:000D B315** | Mov BL,15h |
| **0CFC:000F 02C3** | Add AL,BL |
| **0CFC:0015 8AD0** | Mov DL,AL |
| **0CFC:0017 B402** | Mov AH,02 |
| **0CFC:0019 CD21** | Int 21h |
| **0CFC:001B B44C** | Mov AH,4CH |
| **0CFC:001D CD21** | int 21h |

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

**0CFC**

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, ya que el valor del IP es de 0000 y nuestra dirección de memoria del inicio del programa nos muestra el segmento del código 0CFC seguido con la primera dirección que seria 0000.**

**0CFC:0000 B8FE0C.**

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.

**AX=0000 BX=0000 CX=001F DX=0000 SP=0400 BP=0000 SI=0000 DI=0000**

**DS=0CEC ES=0CEC SS=0CFE CS=0CFC IP=0000**

**0CFC:0000 B8FE0C MOV AX,0CFE**

**AX=0CFE BX=0000 CX=001F DX=0000 SP=0400 BP=0000 SI=0000 DI=0000**

**DS=0CEC ES=0CEC SS=0CFE CS=0CFC IP=0003**

**0CFC:0003 8ED8 MOV DS,AX**

**AX=0CFE BX=0000 CX=001F DX=0000 SP=0400 BP=0000 SI=0000 DI=0000**

**DS=0CFE ES=0CEC SS=0CFE CS=0CFC IP=0005**

**0CFC:0005 B80000 MOV AX,0000**

**AX=0000 BX=0000 CX=001F DX=0000 SP=0400 BP=0000 SI=0000 DI=0000**

**DS=0CFE ES=0CEC SS=0CFE CS=0CFC IP=0008**

**0CFC:0008 BB0000 MOV BX,0000**

**AX=0000 BX=0000 CX=001F DX=0000 SP=0400 BP=0000 SI=0000 DI=0000**

**DS=0CFE ES=0CEC SS=0CFE CS=0CFC IP=000B**

**0CFC:000B B015 MOV AL,15**

**AX=0015 BX=0000 CX=001F DX=0000 SP=0400 BP=0000 SI=0000 DI=0000**

**DS=0CFE ES=0CEC SS=0CFE CS=0CFC IP=000D**

**0CFC:000D B315 MOV BL,15**

**AX=0015 BX=0015 CX=001F DX=0000 SP=0400 BP=0000 SI=0000 DI=0000**

**DS=0CFE ES=0CEC SS=0CFE CS=0CFC IP=000F**

**0CFC:000F 02C3 ADD AL,BL**

**AX=002A BX=0015 CX=001F DX=0000 SP=0400 BP=0000 SI=0000 DI=0000**

**DS=0CFE ES=0CEC SS=0CFE CS=0CFC IP=0011**

**0CFC:0011 B130 MOV CL,30**

**AX=002A BX=0015 CX=0030 DX=0000 SP=0400 BP=0000 SI=0000 DI=0000**

**DS=0CFE ES=0CEC SS=0CFE CS=0CFC IP=0013**

**0CFC:0013 02C1 ADD AL,CL**

**AX=005A BX=0015 CX=001F DX=0000 SP=0400 BP=0000 SI=0000 DI=0000**

**DS=0CFE ES=0CEC SS=0CFE CS=0CFC IP=0015**

**0CFC:0015 8AD0 MOV DL,AL**

**AX=005A BX=0015 CX=001F DX=005A SP=0400 BP=0000 SI=0000 DI=0000**

**DS=0CFE ES=0CEC SS=0CFE CS=0CFC IP=0017**

**0CFC:0017 B402 MOV AH,02**

**AX=025A BX=0015 CX=001F DX=005A SP=0400 BP=0000 SI=0000 DI=0000**

**DS=0CFE ES=0CEC SS=0CFE CS=0CFC IP=0019**

**0CFC:0019 CD21 INT 21**