Facultad de Ingeniería Ingeniería en Informática y Sistemas

Microprogramación

LABORATORIO NO. 02

“Ensamblador, DEBUG y Sistemas Numéricos”

# JULIO ANTHONY ENGELS RUIZ COTO - 1284719

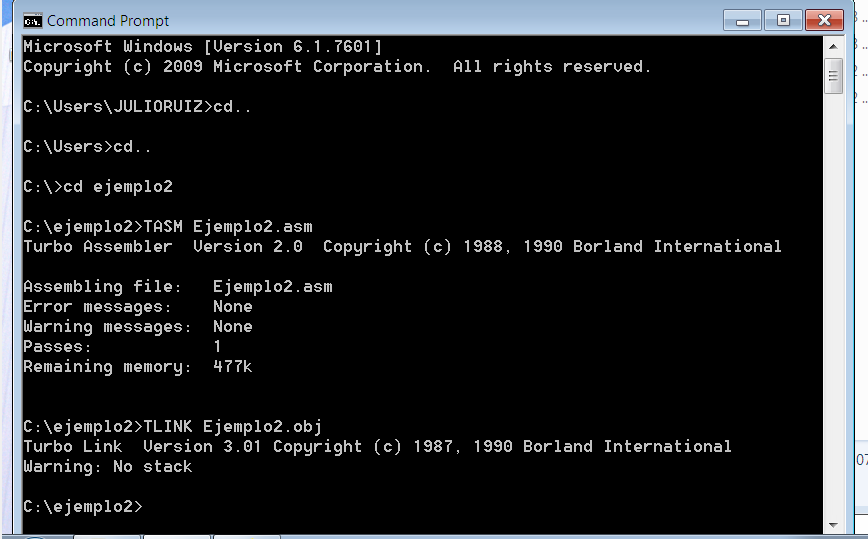
# Ejercicio 1: Utilización del Ensamblador y el Enlazador

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:

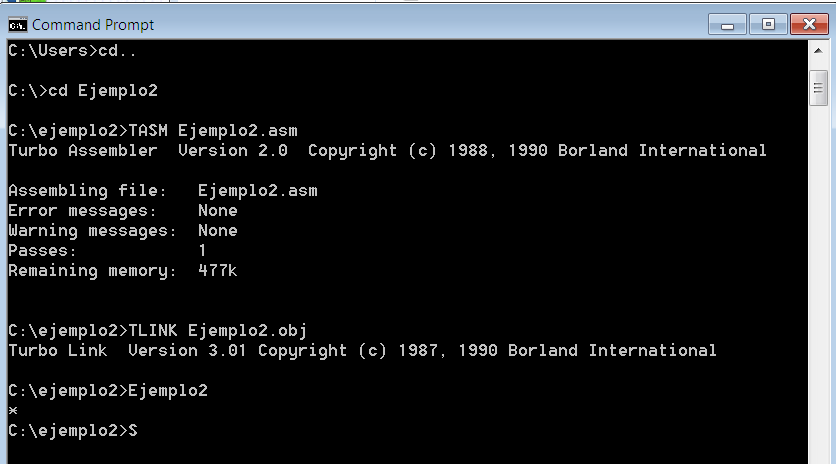
1. Cuando se genera el ejecutable del archivo “Ejemplo2.asm”, ¿cuál es la advertencia que se

muestra en pantalla? ¿Por qué muestra esa advertencia?



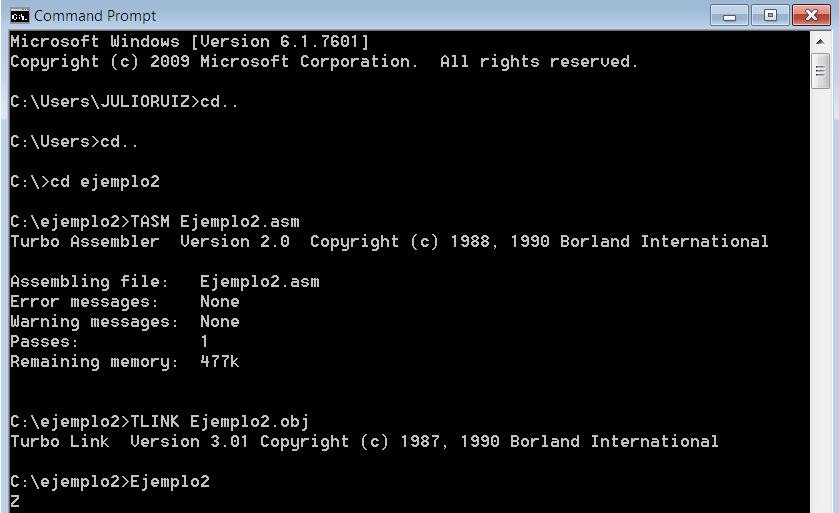
R// “Warning: No stack” se muestra porque el código original no está definido el segmento de pila.

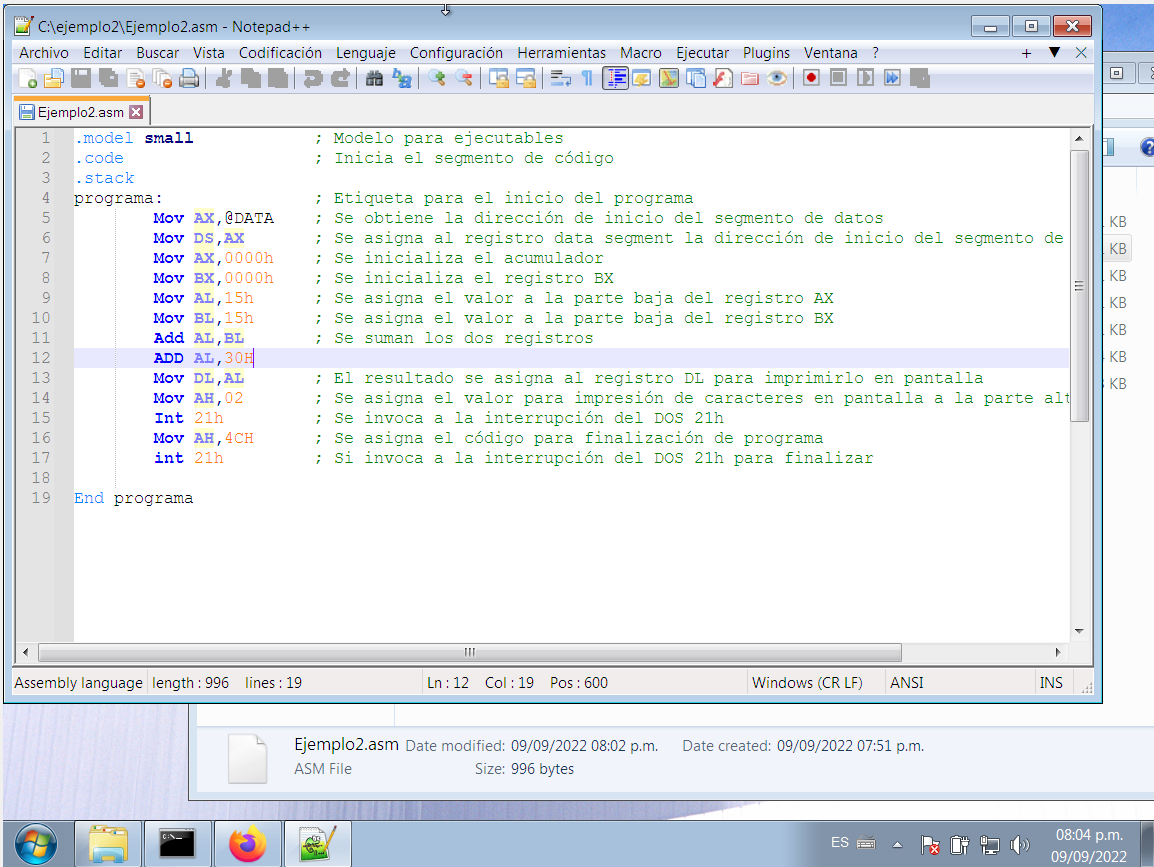
1. Modificado el programa para que no muestre la advertencia, ¿cuál es el resultado del programa, es decir, por qué se imprime ese carácter y no un 30?



R// Porque el resultado de la suma imprime su carácter correspondiente en la tabla ASCII, es decir, 15h + 15h = 2Ah, cuyo símbolo es \*.

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





# Ejercicio 2: Sumas y Restas

1. Deberá declarar 3 variables numéricas con un valor fijo de un dígito 2. Realice las siguientes operaciones:
   1. A + B
   2. A - C
   3. A + B + C
   4. A + 2B -C

# Ejercicio 3: Utilización del Modo “DEBUG”

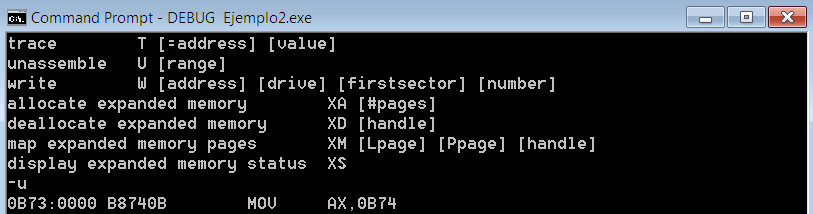
Comandos del Modo “DEBUG”:

* 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 “ DEBUG Ejemplo2.exe” y responda las siguientes

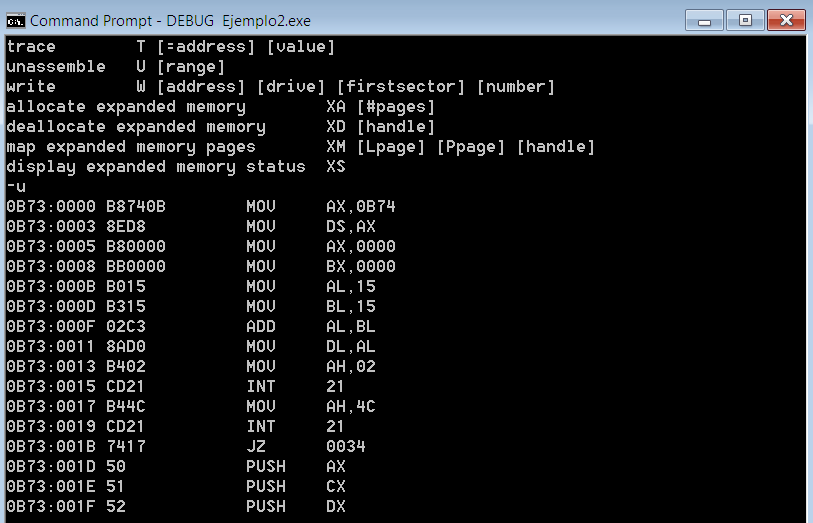
preguntas:

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



Primera Linea

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



Última Linea

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

R// No, porque cuando comienza un punto y coma quiere decir que lo que le sigue es un comentario, en todo caso solo se observa los movimientos entre registros y direcciones de memoria.

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

Facultad de Ingeniería Ingeniería en Informática y Sistemas

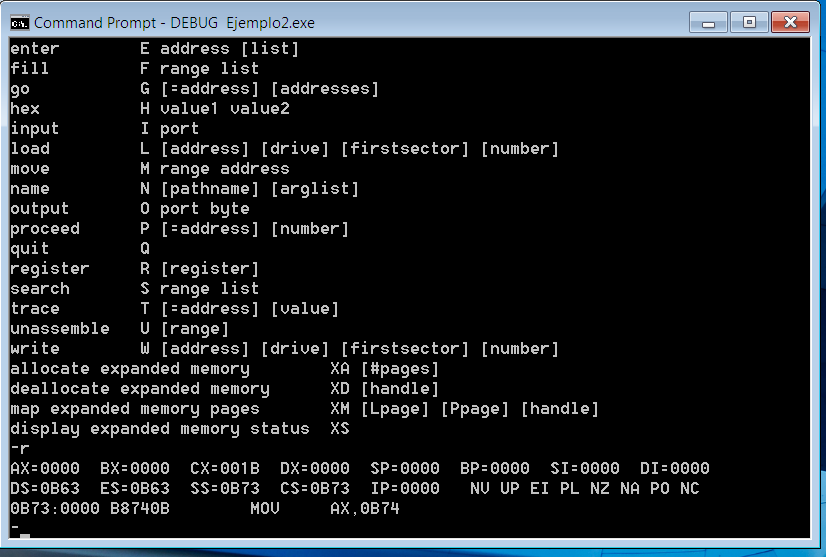
Microprogramación

|  |  |
| --- | --- |
| Dirección de memoria | Instrucción |
|  | Mov AX,@DATA |
|  | Mov DS,AX |
|  | Mov AX,0000h |
|  | Mov BX,0000h |
|  | Mov AL,15h |
|  | Mov BL,15h |
|  | Add AL,BL |
|  | Mov DL,AL |
|  | Mov AH,02 |
|  | Int 21h |
|  | Mov AH,4CH |
|  | int 21h |

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



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?

R// No, porque cambia cada vez que aplicamos el comando t en el modo debug, el siguiente paso en nuestro programa.

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.

