

**UNIVERSIDAD AUTÓNOMA DE CHIAPAS**

**FACULTAD DE NEGOCIOS, CAMPUS IV**

*Licenciatura en ingenieria en desarrollo y tecnologias de software*

**TRADUCTORES DE BAJO NIVEL.**

Asignatura.

**BENAVIDES GARCIA VANESSA**

DOCENTE

**5-D**

Grado y Grupo

**Documentación Proyecto Ensamblador.**

Trabajo

**Luis Alberto Hurtado Gomez**

**Kevin Alejandro Francisco Gonzales**

**Juan Luis González Villalobos**

Alumno

Tapachula Chiapas, 29 de octubre de 2023

Lugar y Fecha

Índice

[***1.- Segmentos y Organización de la Memoria:*** 3](#_Toc150722158)

[***2. Variables y Registros de Memoria:*** 3](#_Toc150722159)

[***3. Directivas:*** 3](#_Toc150722160)

[***4. Macros:*** 3](#_Toc150722161)

[***5. Procedimientos:*** 3](#_Toc150722162)

[***6. Interrupciones:*** 4](#_Toc150722163)

[***7. Operaciones de entrada/salida:*** 4](#_Toc150722164)

[***8. Saltos Condicionales e Incondicionales:*** 4](#_Toc150722165)

[***9. Códigos ASCII:*** 4](#_Toc150722166)

[***10. Bucle:*** 4](#_Toc150722167)

[***11. Manipulación de cuerdas:*** 4](#_Toc150722168)

[***12. Terminación:*** 4](#_Toc150722169)

[***13. Tipos de datos:*** 4](#_Toc150722170)

[***14. Comentarios:*** 5](#_Toc150722171)

[***15. Asamblea condicional:*** 5](#_Toc150722172)

[***16. Representación Numérica:*** 5](#_Toc150722173)

[***17. Concatenación de cadenas:*** 5](#_Toc150722174)

[***18. Etiquetado:*** 5](#_Toc150722175)

# ***1.- Segmentos y Organización de la Memoria:***

Segmento de datos (Data Segment): En el ensamblaje x86, los datos de un programa se organizan en segmentos. El registro del segmento de datos (DS) apunta al comienzo del segmento de datos. El SEGMENTO DE DATOS en el código se usa para definir variables, cadenas y buffers que usará el programa.

Segmento de Código (Code Segment): El registro de Segmento de Código (CS) apunta al comienzo del segmento de código. Contiene las instrucciones del programa. El SEGMENTO CODIGO en el código es donde reside la lógica principal del programa.

# ***2. Variables y Registros de Memoria:***

Variables (Valor1, Valor2, Cadena, etc.): Se utilizan para almacenar y manipular datos durante la ejecución del programa. En el ensamblado x86, los datos a menudo se almacenan en la memoria mediante variables.

Registros (AH, AL, DX, SI, CX, CL): los registros son ubicaciones de almacenamiento pequeñas y rápidas dentro de la CPU. En el ensamblaje x86, se utilizan para diversos fines, como almacenar datos, direcciones o indicadores. AH y AL se refieren a los bytes alto y bajo del registro AX, respectivamente.

# ***3. Directivas:***

ASUMIR: Esta directiva se utiliza para indicarle al ensamblador que se debe suponer que ciertos registros apuntan a ciertos segmentos. Ayuda en la inicialización del registro de segmento.

FINALIZA: Marca el final de un segmento.

# ***4. Macros:***

MACRO: Una macro es una secuencia de instrucciones a la que se le ha dado un nombre. Permite al programador definir fragmentos de código reutilizables.

# ***5. Procedimientos:***

Procedimientos (OPERACIONES\_MENU, NUMERO\_REPETIR, Num1, Num2, etc.): Son secciones modulares de código que realizan una tarea específica. Promueven la reutilización y mantenibilidad del código.

# ***6. Interrupciones:***

INT 21H: En DOS, esta es la interrupción de software 21H. Es un servicio de DOS versátil que se puede utilizar para diversas tareas, como leer desde el teclado, escribir en la pantalla y más. Cada función se identifica mediante un valor en el registro AH.

# ***7. Operaciones de entrada/salida:***

MOV AH, 09H: Este es un servicio de DOS que escribe una cadena en la salida estándar (generalmente la consola).

MOV AH, 01H: Este es un servicio de DOS para leer un carácter de la entrada estándar sin esperar Enter.

MOV AH, 02H: Este es un servicio de DOS para imprimir un carácter en una salida estándar.

# ***8. Saltos Condicionales e Incondicionales:***

CMP y saltos condicionales (JA, JB, JE): se utilizan para comparaciones y los saltos se toman en función del resultado de la comparación.

JMP: salta incondicionalmente a una etiqueta o dirección especificada.

# ***9. Códigos ASCII:***

Códigos ASCII (13, 30H): Se utilizan para comparación y conversión de caracteres. Por ejemplo, 13 representa la tecla Intro y 30H se utiliza para convertir el carácter ASCII '0' a su equivalente numérico.

# ***10. Bucle:***

LOOP y DEC: Se utilizan para implementar bucles.

# ***11. Manipulación de cuerdas:***

MOVS: instrucción de movimiento de cadena, utilizada para mover un bloque de memoria.

# ***12. Terminación:***

INT 21H, 4CH: Función DOS para terminación de programa.

# ***13. Tipos de datos:***

DB: Define un byte, que son 8 bits.

# ***14. Comentarios:***

Comentarios (;): Se utiliza para anotar y explicar el código.

# ***15. Asamblea condicional:***

IF...ELSE...ENDIF: Las directivas de ensamblaje condicional permiten incluir o excluir ciertas partes del código según condiciones específicas.

# ***16. Representación Numérica:***

Decimal (10, 13), Hexadecimal (30H): Se utilizan diferentes bases numéricas para representar valores.

# ***17. Concatenación de cadenas:***

DB 100 dup('$'): reserva espacio para una cadena.

# ***18. Etiquetado:***

Etiquetas (INICIO, OPERACIONES\_MENU, MayorA, SALIR, etc.): Se utilizan para marcar ubicaciones en el código, facilitando el flujo de control.