



UNIDAD DE APRENDIZAJE DE ENSAMBLADORES

CRÉDITOS INSTITUCIONALES: 7

MANUAL DE PRÁCTICAS DE LABORATORIO: ENSAMBLADORES

PROGRAMA DE INGENIERÍA EN COMPUTACIÓN

CU UAEM VALLE DE CHALCO

AUTORES:

IVÁN FABIÁN LUNA

MARCO ALBERTO MENDOZA PÉREZ

PERIODO DE REALIZACIÓN:

MARZO - AGOSTO DE 2018





PRESENTACIÓN

El ingeniero en computación es el profesional que posee los conocimientos y habilidades en el desarrollo de sistemas computacionales, diseño y mantenimiento de hardware, comunicaciones y redes de computadoras así como en la administración de recursos computacionales.

Una de las principales actividades del ingeniero en computación es la programación, cuyas bases deben ser adquiridas en su formación. La programación, como una parte de la informática también evoluciona continuamente. Sin embargo, la programación estructurada en otros paradigmas de programación, es el conocimiento base y permite la adquisición de habilidades de programación que se requieren en la construcción de aplicaciones de mediana y alta complejidad.

La Unidad de Aprendizaje de Ensambladores, tiene como finalidad proporcionar los conceptos básicos de lo que es la programación de un ensamblador para una arquitectura de procesador en particular, de su funcionamiento y de las modificaciones que se pueden realizar.

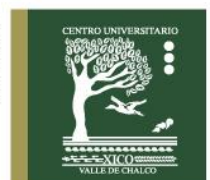
La enseñanza de este curso responde a las exigencias de la UAEM (de acuerdo a lo marcado en los dos últimos planes rectores) y de la facultad de ingeniería en particular, si se adopta un esquema metacognitivo, siempre y cuando se cumplan las condiciones:

1. Que los asistentes participen activamente en el curso.
2. Que el instructor sea un mediador, un guía y monitor del aprendizaje.
3. Que se usen materiales de enseñanza para el monitoreo del avance del curso.
4. Que se aprovechen los errores y sirvan como retroalimentación para el mejor aprendizaje.
5. Que se le dé una orientación a la enseñanza hacia la obtención del metaconocimiento.

Es un curso de formación inicial, dirigido a los alumnos de ingeniería en computación. El curso es de tipo teórico práctico, ya que se realizarán actividades en las cuales el alumno tendrá que demostrar que domina el uso de la programación secuencial para el planteamiento de soluciones a problemas de arquitecturas. Es un curso de nivel básico, sin embargo los asistentes deben tener conocimientos sobre *componentes de la computadora, conceptos de hardware y software, diseño digital y de sistemas de numeración.*

Dado que existen múltiples estrategias de enseñanza, sustentadas en diversas teorías, se consideran aquellas que promueven la actividad del docente sobre la acción del discente y las que permiten la creación de espacios que estimulan la actividad del alumno a través del contacto con situaciones reales o aproximadas a éstas.

Las presentes prácticas de laboratorio fueron desarrolladas en apego al programa de la Unidad de Aprendizaje de Ensambladores, con la finalidad de que los alumnos puedan poner en práctica los conocimientos teóricos que se van adquiriendo a lo largo del curso. Esta Unidad de Aprendizaje consta de la siguiente estructura:





1. Analizar los conceptos relacionados con sistemas numéricos, sistemas de cómputo, arquitectura de procesadores y programación de sistemas.
2. Conocer, analizar y comprender los elementos y etapas de un ensamblador y aplicarlo en el desarrollo de un ensamblador.
3. Conocer, analizar y comprender los elementos y fases de un cargador.

Este manual consta de 10 prácticas, cuya complejidad es acorde al tema que abarca.

Cada Práctica consta de Título, Objetivos, Descripción, Actividad, Material y Equipo a Utilizar, Procedimiento, Duración, Actividad que debe desarrollar el discente, Evaluación y Retroalimentación del tema. El alumno debe replicar de forma práctica lo que se encuentra en el apartado de Procedimiento y resolver lo que se pide en el apartado de Actividad que debe desarrollar el discente. La evaluación del curso se hace de manera sistemática y continua y pueden utilizarse las herramientas sugeridas en cada una de las prácticas.

De cada Práctica, el alumno deberá elaborar su Reporte de Práctica correspondiente, el cual lo debe entregar de forma: digital (infografía o artículo) o impresa (infografía o artículo), con los siguientes puntos: Carátula, Título, Resumen (máx. 150 palabras) y Palabras clave (máx. 3 y ordenadas de forma ascendente), Introducción, Problema de Investigación, Objetivos (General y Específicos), Marco Referencial, Desarrollo (diagramas y evidencias originales con explicación), Resultados (evidencias originales con explicación), Conclusiones Individuales, Trabajos Futuros y Bibliografía/Referencias Electrónicas (ambas en formato APA).





ÍNDICE DE CONTENIDO

PRESENTACIÓN.....	2
PRÁCTICA 1 HOLA MUNDO.....	5
PRÁCTICA 2 DEBUGEADOR TD.....	7
PRÁCTICA 3 CICLO FOR.....	9
PRÁCTICA 4 CICLO WHILE.....	11
PRÁCTICA 5 PILA (STACK).....	13
PRÁCTICA 6 USO DE ARREGLOS.....	15
PRÁCTICA 7 ARREGLOS Y PILAS.....	17
PRÁCTICA 8 USO DE MÁS DE UN ARREGLO.....	19
PRÁCTICA 9 MACROS.....	21
PRÁCTICA 10 PUESTA EN PRÁCTICA LO APRENDIDO.....	23
REFERENCIAS.....	26





PRÁCTICA 1 HOLA MUNDO

Objetivos

Familiarización con el lenguaje de programación máquina y la estructura de un programa. Entender la forma de compilación y enlazamiento del programa en la compilación.

Descripción

El alumno programara el tradicional “Hola mundo” en lenguaje máquina para el procesador de Intel de arquitectura X86.

Actividad

Instalar el emulador DOSBOX para arquitecturas X86. Capturar el código en lenguaje máquina para el “Hola mundo”, realizar el ensamble, enlace y ejecución.

Material y Equipo a Utilizar

- PC de escritorio o Laptop
- Emulador DOSBOX
- Ensamblador MASM
- Enlazador LINK
- Editor de texto.
- Apuntes (Cuaderno)

Procedimiento

1. Instalar el emulador DOSBOX.
2. Crear una carpeta en raíz del disco local C y colocar los archivos con extensión .exe del MASM, LINK.
3. Capturar el código del “Hola mundo” y guardar con extensión .ASM en la carpeta donde están los archivos del paso 2.
4. Dar doble click en el icono de DOSBOX.
5. Colocar en la venta CLI que se abre como resultado del paso 4. Lo siguiente

- a) Z:\> mount unidad: unidad:\
- b) Z:\> mount unidad:\
- c) Unidad:\ **En este último punto ya estamos en el directorio raíz de la unidad.**

6. Dirigirnos a la carpeta donde están los archivos del punto 2 y 3.

- a) Unidad:\ cd **NOMBREDELA CARPETA**





7. Ensamblar el archivo donde se encuentra el programa “Hola mundo”
 - a) Unidad:\ MASM NOMBREDELARCHIVO:ASM

Enseguida dar tres “enter” seguidos.

8. Enlazar los archivos que resultaron del ensamblaje
 - a) Unidad:\ LINK NOMBREDELARCHIVO

Enseguida dar tres “enter” seguidos.

9. Ejecutar el programa resultado-
 - a) Unidad:\ NOMBREDELARCHIVO

Nota: Recuerde proporcionar apoyo al discente en caso de requerir.

Duración: 2 Horas

Actividad que debe desarrollar el discente: Programación, compilación y ejecución.

Evaluación: Heteroevaluación / lista de cotejo.

Retroalimentación del tema: El docente preguntara sobre los pasos de la compilación y ejecución.





PRÁCTICA 2 DEBUGEADOR TD

Objetivos

El alumno aprenderá a utilizar el depurador TD esto con la finalidad de observar paso a paso el contenido de los registros y la ejecución de un programa escrito en lenguaje máquina.

Descripción

El alumno retomará la práctica 1 ejecutará el programa con la ayuda del depurador TD, identificará las direcciones de donde comienzan los y donde terminan los segmentos. Observara como cambian los valores de los diferentes como son contadores, punteros y registros.

Actividad

Colocar el depurador TD en la carpeta donde se encuentran los archivos del programa, el ensamblador, enlazador y depurador.

Material y Equipo a Utilizar

- PC de escritorio o Laptop
- Emulador DOSBOX
- Ensamblador MASM
- Enlazador LINK
- Editor de texto.
- Apuntes (Cuaderno)
- Depurador TD.

Procedimiento

1. El alumno ejecutará el programa de “Hola mundo” con el depurador TD, para esto debe colocar en la venta de comandos
 - a) Unidad:\ TD NOMBREDELARCHIVO
2. Colocar puntos de seguimiento o de parada para ello se desplazara con la ayuda de las flechas de arriba y abajo, llegando al punto presionara F2 la línea donde se colocó cambiara a color azul.
3. Ejecutar el programa presionando F9.





4. Anotar como cambian los valores de los registros, de los punteros, del contador de instrucciones, de los índice segmento de código, datos, pila.

Nota: Recuerde proporcionar apoyo al discente en caso de requerir.

Duración: 2 Horas

Actividad que debe desarrollar el discente: Programación, compilación, ejecución y debugación.

Evaluación: Heteroevaluación / lista de cotejo.

Retroalimentación del tema: El docente preguntara que secuencia sigue el programa.





PRÁCTICA 3

CICLO FOR

Objetivos

El alumno aprenderá a identificar secciones de código que realizan una tarea en específica, además de colocarlas en un ciclo “for” para aquellos casos en donde se requiere hacer la tarea un número determinado de veces.

Descripción

El alumno utilizará la instrucción “LOOP” que en lenguajes de alto nivel se le conoce como el ciclo iterativo “FOR”, para ello debe de entender y comprender las instrucciones que permiten su implementación.

Actividad

El alumno realizará las modificaciones que permitan desplegar un número determinado de veces el mensaje “Hola mundo”.

Material y Equipo a Utilizar

- PC de escritorio o Laptop
- Emulador DOSBOX
- Ensamblador MASM
- Enlazador LINK
- Editor de texto.
- Apuntes (Cuaderno)
- Depurador TD.

Procedimiento

1. Identificar de las practicas anteriores la sección de código que despliega el mensaje de “Hola mundo”
2. Convertir un numero de determinadas veces “n” a su equivalente en Hexadecimal, que se quiere que se imprima el mensaje.
3. Colocar la sección de código como se muestra a continuación

MOV CX, NUMEROVECESENHEXADECIMAL

NOMBREDELICLO:

SECCION DE CODIGO A REPETIR

LOOP NOMBREDELICLO





4. Compilar el programa y ejecutarlo.
5. Realizar lo anterior para valores de 30, 70 y 100.

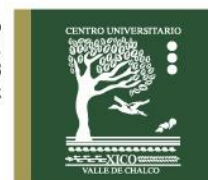
Nota: Recuerde proporcionar apoyo al discente en caso de requerir.

Duración: 2 Horas

Actividad que debe desarrollar el discente: Programación, compilación y ejecución.

Evaluación: Heteroevaluación / lista de cotejo.

Retroalimentación del tema: El docente preguntara sobre los pasos para implementar el ciclo iterativo.





PRÁCTICA 4 CICLO WHILE

Objetivos

El alumno aprenderá a identificar secciones de código que realizan una tarea en específica, además de colocarlas en un ciclo “while” para aquellos casos en donde se requiere hacer una tarea hasta que se cumpla una condición.

Descripción

El alumno utilizara la instrucción “CMP”, “JZ” y “JNZ” que en lenguajes de alto nivel se le conoce como el ciclo “while”, para ello debe de entender y comprender las instrucciones que permiten su implementación.

Actividad

El alumno realizará las modificaciones que permitan desplegar el mensaje “Hola mundo” y terminar hasta que se cumpla una condición, esta se leerá de teclado.

Material y Equipo a Utilizar

- PC de escritorio o Laptop
- Emulador DOSBOX
- Ensamblador MASM
- Enlazador LINK
- Editor de texto.
- Apuntes (Cuaderno)
- Debugeador TD.

Procedimiento

1. Elegir una condición (carácter) para ello debe de investigarse previamente su valor en Hexadecimal.
2. Observar en el TD el contenido de los registros al ejecutarse las líneas de código siguientes:

```
MOV AH, 01H  
INT 21H
```

La ventana de comandos se abrirá y esperará que se presione una tecla (carácter) observar en que registro se encuentra el valor en hexadecimal y recordar o anotar, importante no modificar este registro o si el caso respaldar en otro o en un variable.





3. Como primer paso para implementar el ciclo while se debe colocar lo siguiente:

NOMBRECICLO:

MOV AH, 01H

INT 21H

CMP VALORDECONDICION, REGISTRODONDEESTAELVALOR

JN **CONDICIONCUMPLIDA**

JNZ **CONDICIONNOCUMPLIDA**

CONDICION CUMPLIDA:

JMP SALIR

CONDICIONNOCUMPLIDA:

“CODIGO HOLA MUNDO”

JMP **NOMBRECICLO**

4. Compilar y ejecutar

Nota: Recuerde proporcionar apoyo al discente en caso de requerir.

Duración: 2 Horas

Actividad que debe desarrollar el discente: Programación, compilación y ejecución.

Evaluación: Heteroevaluación / lista de cotejo.

Retroalimentación del tema: El docente preguntara sobre el formato de los datos.





PRÁCTICA 5 PILA (STACK)

Objetivos

El alumno aprenderá a hacer uso de la Pila en ensamblador, de sus operaciones de inserción y extracción.

Descripción

La pila es un segmento de memoria reservada que se usa para almacenar valores de manera temporal, tiene como característica que los valores que se introducen al final son los primero es sacarse.

Actividad

El alumno realizará un programa donde empleará las operaciones PUSH (inserción) y POP (extracción) utilizara.

Material y Equipo a Utilizar

- PC de escritorio o Laptop
- Emulador DOSBOX
- Ensamblador MASM
- Enlazador LINK
- Editor de texto.
- Apuntes (Cuaderno)
- Depugeador TD.

Procedimiento

1. Realizar un programa que reciba como entrada una cadena y esta se introduzca en la pila. Para ello debe de ocupar el ciclo "while" de la practica 4.

En la sección donde dice no se cumple la condición se debe de colocar lo siguiente:

```
MOV AH, 01H  
INT 21H
```

```
PUSH AL
```

Además se debe colocar un contador el cual como su nombre lo indica tendrá el valor del total de los caracteres introducidos.





2. Con el valor del contador lo siguiente es implementar un ciclo for esto para sacar los elementos de la pila e imprimirlos.

MOV CX, CONTADOR

CICLOFOR:

POP DL

MOV AH, 02H

INT 21H

LOOP **CICLOFOR**

3. Identificar el puntero de la Pila y observar cómo se modifica su valor.
4. Compilar y ejecutar.

Nota: Recuerde proporcionar apoyo al discente en caso de requerir.

Duración: 2 Horas

Actividad que debe desarrollar el discente: Programación, compilación y ejecución.

Evaluación: Heteroevaluación / lista de cotejo.

Retroalimentación del tema: El docente preguntara sobre el funcionamiento de la pila.





PRÁCTICA 6 USO DE ARREGLOS

Objetivos

El alumno aprenderá a hacer uso de arreglos, del manejo de sus punteros (registros índices), los cuales nos permite acceder y guardar los datos.

Descripción

El alumno definirá arreglos de un determinado tamaño para ello utilizará la palabra reservada DUP (VALORDEINICIALIZACION), se tiene como opciones inicializarlo con valores previos ó simplemente en vacío. Además de introducir y sacar los valores de un arreglo.

Actividad

El alumno realizará un programa el cual guarde una cadena en un arreglo, par esto debe de emplear los conocimientos adquiridos en las prácticas anteriores.

Material y Equipo a Utilizar

- PC de escritorio o Laptop
- Emulador DOSBOX
- Ensamblador MASM
- Enlazador LINK
- Editor de texto.
- Apuntes (Cuaderno)
- Debugeador TD.

Procedimiento

1. Realizar un programa el cual reciba como entrada una cadena y guardarla en un arreglo.

Entrada: *Esta es una cadena cualquiera.*

Las cadenas se inicializan en el segmento de datos como:

Nombre_de_la_cadena dw(tipo) tamaño dup(inicializar_en_un_valor_dado)

dw es porque el tipo de dato es una palabra, si quiero un arreglo que almacene byte el tipo será de ***db***.





Para trabajar con los arreglos es necesario primero obtener la dirección del mismo y esto se hace de la siguiente manera:

lea registro_indice_o_destino, nombre_de_la_cadena

Una vez hecho lo anterior para guardar en el arreglo lo que se hace es

mov [registro_indice_destino_o_fuente], valor

Para obtener un dato del arreglo se hace lo siguiente

mov valor, [registro_indice_destino_o_fuente]

Recuerde incrementar o decrementar según sea el caso a los registros índices destino o fuente.

2. Una vez guardada la cadena en el arreglo lo siguiente es pedir al usuario una letra y contabilizar cuantas veces aparece está en el arreglo.
3. Imprimir la cadena introducida al revés, esto es con la finalidad de aprender a usar constantes y guardar las direcciones o valores que sean utilizados en otras secciones del código.

La definición de las constantes es:

nombre_de_la_constante d_tipo valor_inicial

Y se hace en el segmento de datos.

4. Compilar y ejecutar.

Nota: Recuerde proporcionar apoyo al discente en caso de requerir.

Duración: 2 Horas

Actividad que debe desarrollar el discente: Programación, compilación y ejecución.

Evaluación: Heteroevaluación / lista de cotejo.

Retroalimentación del tema: El docente preguntara sobre cosas específicas de los arreglos como son direcciones, punteros.





PRÁCTICA 7 ARREGLOS Y PILAS

Objetivos

El alumno aprenderá a combinar los arreglos y pilas en un solo programa, y de reafirmar los conocimientos adquiridos en las prácticas 6 y 7.

Descripción

En los ambientes de programación combinar las diferentes estructuras se dificultada al no saber bien su correcto funcionamiento, para ello es necesario observar cómo cambian los valores de los registros que utilizan las diferentes estructuras y en aquellos casos donde se ocupe un registro varias veces, es necesario auxiliarse de variables para no perder los valores.

Actividad

Realizar un programa que introduzca una cadena en un arreglo y a su vez en la pila. Lo anterior se hará con el ciclo while, para introducir y sacar los caracteres. En la parte de sacar lo que se hará es comparar el carácter sacado del arreglo con el carácter sacado de la pila si son iguales que realice la comparación hasta que se vacié el arreglo y la pila. Si al final del ciclo se tiene que por lo menos un carácter no es igual que coloque una bandera en uno y en otro caso en 0. Con esta bandera se determina si una palabra es palíndromo o no.

Material y Equipo a Utilizar

- PC de escritorio o Laptop
- Emulador DOSBOX
- Ensamblador MASM
- Enlazador LINK
- Editor de texto.
- Apuntes (Cuaderno)
- Debugeador TD.

Procedimiento

Parte 1

1. Introducir en el arreglo y la pila.
2. Contabilizar los caracteres.





Parte 2

3. Inicializar la bandera en cero.
4. Sacar del arreglo y la pila.
5. Comparar carácter con carácter, si son diferentes colocar bandera en 1 en caso contrario no modificar el valor.

Parte 3

6. Compara el valor de bandera si es cero mandar la leyenda de si es palíndromo la palabra, en caso contrario no es palíndromo.
7. Compilar y ejecutar.

Nota: Recuerde proporcionar apoyo al discente en caso de requerir.

Duración: 4 Horas

Actividad que debe desarrollar el discente: Programación, compilación y ejecución.

Evaluación: Heteroevaluación / lista de cotejo.

Retroalimentación del tema: El docente preguntara sobre la forma de la implementación y de las modificaciones que se pueden realizar.





PRÁCTICA 8 USO DE MÁS DE UN ARREGLO

Objetivos

El alumno usara arreglos para determinar si un número es primo o no, para ello tiene que saber usar los registros SI y DI vistos previamente.

Descripción

El alumno debe de manipular los dos únicos registros que permiten almacenar y acceder a los valores que están en el arreglo. Lo anterior es para el caso ideal pero se deben de hacer ciertas manipulaciones para más de dos arreglos. Los registros índice y destino lo que hacen es direccionarlos a la primera posición del arreglo, en otras palabras nos da la dirección de memoria donde comienza el arreglo.

Actividad

Realizar un programa que lea una serie de dígitos y que los guarde en un arreglo, como en los casos anteriores se debe llevar un contador esto con la finalidad de sacar los valores del arreglo y convertirlos en un número en hexadecimal, esto es porque la cadena entra como se mencionó anteriormente en dígito por dígito y en un formato ASCII. Para ello debe de tener además del arreglo de los dígitos otro arreglo que contenga las potencias de 10 es decir de 1, 10, 100, 1000.

Material y Equipo a Utilizar

- PC de escritorio o Laptop
- Emulador DOSBOX
- Ensamblador MASM
- Enlazador LINK
- Editor de texto.
- Apuntes (Cuaderno)
- Depurador TD.

Procedimiento

1. Desarrollar un programa el cual determine si un número es primo. Para ello debe de hacer lo siguiente:
 - a) Guardar en un arreglo los números introducidos por teclado hasta que se presione enter, para ello les debe de restar 30H recuerde que por teclado se recibe el ASCII de cada uno de los dígitos que conforman el número.
 - b) Multiplicar el número del arreglo por la potencia correspondiente (arreglo potencia), proceso iterativo Figura 1. Respalidar el número obtenido.



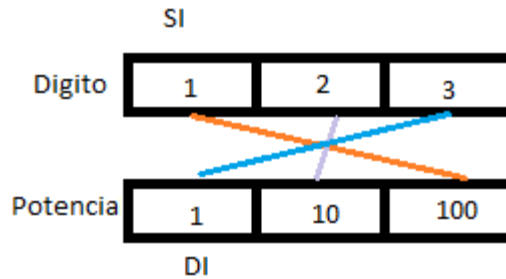


Figura 1. Obtención de un número en decimal.

En la figura anterior el número que introdujo por teclado fue 123. Recuerde se lee carácter por carácter y en su correspondiente ASCII, en otras, es 31H, 32H, 33H.

- c) Respalidar el número en Hexadecimal dos veces, en otras palabras HEXADECIMAL1, HEXADECIMAL2. Dividir el Hexadecimal 1 entre el Hexadecimal 2 si la división es cero incrementar un contador, paso siguiente decrementar en uno el Hexadecimal 2 y realizar la misma división. Si el número del contador es igual a 2 es un numero primo, si es distinto de 2 no es un numero primo.

División en ensamblador

```
MOV DX, 0000H ; Limpiar DX esto porque el residuo de la división se almacena en este.  
MOV BX, DIVISOR  
MOV AX, DIVIDENDO  
DIV BX ;Divide AX entre BX
```

Multiplicación en ensamblador

```
MOV BX, Multiplicador  
MOV AX, Multiplicando  
DIV BX ;Multiplica AX por BX
```

2. Compilar y ejecutar.

Nota: Recuerde proporcionar apoyo al discente en caso de requerir.

Duración: 4 Horas

Actividad que debe desarrollar el discente: Programación, compilación y ejecución.

Evaluación: Heteroevaluación / lista de cotejo.

Retroalimentación del tema: El docente preguntara sobre la metodología que se emplearía para usar n arreglos y del uso de sus registros índices.





PRÁCTICA 9 MACROS

Objetivos

El alumno usara los macros, los cuales son líneas de código que realizan una determinada tarea y son útiles cuando segmentos de código son repetitivos.

Descripción

En la programación en lenguajes de alto nivel se utilizan procedimientos, funciones, clases para implementar tareas que se realizan un número determinado de veces y que si se programan, como se resultado se obtienen un gran número de líneas haciendo que sea más complicado encontrar errores o modificaciones. En ensamblador eso se hace con los macros.

Actividad

Implementar un macro que despliegue una cadena, lo que se debe es modificar las practicas 1, 2 y 3 en lugar de colocar las instrucciones que despliegan el mensaje es mandar a llamar el macro que hace lo anterior.

Material y Equipo a Utilizar

- PC de escritorio o Laptop
- Emulador DOSBOX
- Ensamblador MASM
- Enlazador LINK
- Editor de texto.
- Apuntes (Cuaderno)
- Debugeador TD.

Procedimiento

1. Implementar el uso de macros en las prácticas 1, 2, 3.

Recuerde que un macro se define como

Nombredelmacro **MACRO**parametro1, parametro2, ...,parametron

Línea código 1

Línea código 2

Línea código 3

Línea código 4

•





.
.

Línea código n

ENDM

Las letras en verde son palabras reservadas que identifican al macro.

Para incluir los macros en el programa principal se debe de incluir el archivo donde están.

include nombredearchivo.lib

.model small

.data

.stack

.code

Etiquetadeliniciodelpograma

Lo macros se llaman por el nombre del macro

Nombredelmacro (solo el nombre del macro si no recibe parámetros)

Nombredelmacro parametro1, parametro2,...,parametron (en caso del que el macro reciba parámetros)

END **Etiquetadeliniciodelpograma**

2. Compilar y ejecutar.
3. Observar que pasa con los registros de propósito general AX, BX, CX y DX antes de mandar a llamarlos y después de mandar a llamarlos.

Nota: Recuerde proporcionar apoyo al discente en caso de requerir.

Duración: 2 Horas

Actividad que debe desarrollar el discente: Programación, compilación y ejecución.

Evaluación: Heteroevaluación / lista de cotejo.

Retroalimentación del tema: El docente preguntara cual es la diferencia entre macros con etiquetas y si etiquetas.





PRÁCTICA 10

PUESTA EN PRÁCTICA LO APRENDIDO

Objetivos

El alumno utilizara los conocimientos previos de macros, acceso a arreglos. Y de las operaciones que se pueden realizar entre registros.

Descripción

El alumno pondrá en práctica lo aprendido durante las clases de teoría y de práctica, utilizará todo lo aprendido para mostrar los resultados que se obtienen al introducir datos para que se realice una tarea en particular la complejidad radica en que en lenguaje de bajo nivel todo se procesa a nivel de byte (ASCII).

Actividad

Realizar un programa en donde se continúa con lo que se realizó en las prácticas anteriores, se tiene que optimizar lo que se tiene para obtener resultado óptimos, para así pasar a una programación más profesional. Como ya se mencionó los problemas que se tiene es de que todo es procesado carácter y carácter, no se tienen los beneficios de lenguajes de alto nivel donde eso ya está resuelto. En este caso lo que se pide es desarrollar un procedimiento, macro que realice la conversión de un número Hexadecimal a su correspondiente en decimal.

Material y Equipo a Utilizar

- PC de escritorio o Laptop
- Emulador DOSBOX
- Ensamblador MASM
- Enlazador LINK
- Editor de texto.
- Apuntes (Cuaderno)
- Debugeador TD.

Procedimiento

Parte 1.

1. Desarrollar un programa el cual dado un número de entrada lo procese, este proceso es convertirlo a Hexadecimal, recuerde que la forma en que recibe los números es su formato ASCII.

Ejemplo: A través de teclado se introduce el número 123. Lo que lee ensamblador son los códigos ASCII de cada número, el número en ese código se muestra en la Tabla 1.





Tabla 1. Formato de datos en ensamblador.

31H	32H	33H
-----	-----	-----

Para que se pueda trabajar con el 123 se necesita:

- a) Restarle 30H a cada número.
- b) En la práctica anterior esto se realizó y se nombró Hexa.

Una vez que pasa por Hexa lo siguiente es regresarlo al número original que en este ejemplo es 123. El número después de haber pasado por Hexa se tiene como resultado:

$$(123)_{10} = (7B)_{16}$$

El número que se obtiene es que se va a procesar para regresarlo a su forma original.

Parte 2.

- 1. Retomando el programa de la practica anterior, ahora se deben de desplegar son los números por los cuales es divisible.

Ejemplo 2. Que numero dividen al 60.

Tabla 2. Números entre los que es divisible el número 60.

Numero	Divisor	Residuo
60	1	0
60	2	0
60	3	0
60	4	0
60	5	0
60	6	0
60	7	i=0
60	8	j=0
60	9	i=0
60	10	0
60	11	j=0
60	12	0
60	13	j=0
60	14	j=0
60	15	0
.	.	.
.	.	.
.	.	.
60	60	0





Lo que debe de desplegar el programa son los números de la columna Divisor de la Tabla 2 y en los cuales el residuo es igual a cero, el número 60 no es que debe de procesar el programa es solo un ejemplo. El programa debe de aceptar cualquier número.

Para poder mostrar los números de la columna divisor se tiene que hacer un proceso inverso a de la conversión en Hexadecimal, se tiene que desarrollar un procedimiento el cual consiste en tomar los cocientes sumarle 30H y terminar hasta que la potencia sea igual a 1.

NumeroHexadecimal/Potencia

Tabla 3. Conversión de un número en Hexadecimal a Decimal.

Número (Residuo)	Cociente
328/100	3
28/10	2
8/1	8

Para visualizar el resultado en la línea de comandos es necesario sumarle 30H a cada uno de los elementos que en este caso son los de columna de cociente de la Tabla 3.

2. Compilar y ejecutar.

Nota: Recuerde proporcionar apoyo al discente en caso de requerir.

Duración: 4 Horas

Actividad que debe desarrollar el discente: Programación, compilación y ejecución.

Evaluación: Heteroevaluación / lista de cotejo.

Retroalimentación del tema: El docente preguntará sobre técnicas que se pueden implementar para procesar cadenas grandes de caracteres, difiriendo entre los tipos de datos que se manejan.





BIBLIOGRAFÍA

- Barry B. Brey. Programming the 80286, 80386, 80486, and Pentium-based Personal Computer. Editorial Prentice Hall. ISBN 0023142634.
- D.L. 80386: Guía de Programación. Editorial RA-MA. ISBN 8486381703.
- Intel. Intel Architecture Software Developer's Manual: Instruction set reference. Editorial Intel. ISBN 1555122736.
- J. Terry Godfrey. Lenguaje Ensamblador para Microcomputadoras IBM. Editorial Prentice Hall. ISBN 9688802042.

