

# UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA



## Organización de computadoras y lenguaje ensamblador

**Practica 7.** Estructura de control en lenguaje ensamblador para el procesador 8086

**Alumno:** Caudillo Sánchez Diego

**Matricula:** 1249199

**Grupo:** 551

**Docente:** Dr. Mauricio Alonso Sánchez Herrera

**Fecha de entrega:**

**Objetivo:** Familiarizarse con las estructuras de control para el lenguaje ensamblador para el 8086 en modo de 16 bits.

**Materiales:** TASM.exe, TLINK.exe, PCLIB06.lib, formato.asm, procs.inc

**Teoria:** Hacer una resena sobre el propósito de las siguientes directivas:

- DB
- DD
- DW
- END
- ENDS
- ENDP
- EQU
- PROC

**Desarrollo:**

PARTE 1.

Compilar(1), encadenar con biblioteca externa(2), y ejecutar (3) un primer programa en ensamblador. Ejecutar los siguientes comandos:

1. C:\OCLE tasm formato.asm
2. C:\OCLE tlink formato.obj,,PCLIB06.OBJ
3. C:\OCLE formato.exe

Actividad para validar el desarrollo de esta parte:

1. Tras ejecutar el programas *formato.exe*, se deberá borrar la pantalla y posteriormente mostrar el mensaje “Hola Mundo”.

PARTE 2.

Basado en el archivo formato.asm implementar en archivos individuales ejemplos básicos de las siguientes estructuras de control:

1. IF-THEN
2. IF-THEN-ELSE
3. CASE
4. FOR
5. WHILE-DO
6. DO-WHILE

Actividad para validar el desarrollo de esta parte:

1. Cada ejemplo deberá estar debidamente documentado con comentarios significativos.
2. La ejecución deberá informar con claridad la acción tomadas por la estructura de control mediante impresiones a pantalla.

## Teoría

**DB:** De las siglas *Define Byte*. La directiva asigna un dato de tamaño de un 1byte (8 bits) en memoria. El ensamblador asocia un *offset* a cada nombre de una variable definida en el programa.

**DD:** De las siglas *Define Doubleword*. La directiva asigna un dato de tamaño de 4 bytes (32 bits) en memoria. El ensamblador asocia un *offset* a cada nombre de una variable definida en el programa.

**DW:** De las siglas *Define Word*. La directiva asigna un dato de tamaño de 2 bytes (16 bits) en memoria. El ensamblador asocia un *offset* a cada nombre de una variable definida en el programa.

**END:** Cuando el ensamblador escanea el programa que debe ensamblarse, debe saber dónde termina el programa. No puede depender de una instrucción HALT para esto, porque algunos programas no contienen una instrucción de *halt* y otros no contienen un *halt* en absoluto. Un programa de aplicación utilizado, por ejemplo, en el monitoreo de procesos en el control puede ejecutarse continuamente y, por lo tanto, no contener una instrucción de detención. Por lo tanto, en un ensamblaje final, la directiva END debe ser la última instrucción.

**ENDP:** La declaración de ENDP marca el final de una definición de procedimiento utilizando la instrucción PROC. El nombre debe coincidir con el nombre utilizado en la sentencia PROC.

**EQU:** La directiva EQU asocia un nombre simbólico con una expresión entera o con algún texto arbitrario. Existen tres formatos:

- *nombre EQU expresión*
- *nombre EQU símbolo*
- *nombre EQU <texto>*

En el primer formato, *expresión* debe ser una expresión entera válida. En el segundo formato, *símbolo* es el nombre de un símbolo existente, que ya se ha definido con = o EQU. En el tercer formato, puede aparecer cualquier texto dentro de los signos < y >. Cuando el ensamblador se encuentra a *nombre* más adelante en el programa, sustituye el valor entero o el texto por ese símbolo. EQU puede ser útil cuando se define un valor que no se evalúa como entero. Por ejemplo, una constante numérica real puede definirse mediante EQU: PI EQU <3.1416>.

**PROC:** Es una declaración utilizada para indicar el comienzo de un procedimiento o subrutina.

**PARTE 1.** Compilar(1), encadenar con biblioteca externa(2), y ejecutar (3) un primer programa en ensamblador.

```
X:\FORMATO>tasm formato
Turbo Assembler  Version 2.01  Copyright (c) 1988, 1990 Borland International

Assembling file:   formato.ASM
Error messages:   None
Warning messages: None
Passes:           1
Remaining memory: 491k

X:\FORMATO>tlink formato,,,pclib06
Turbo Link  Version 3.01 Copyright (c) 1987, 1990 Borland International

X:\FORMATO>formato_
```

**Figura 1.** Ensamblaje y enlazador del programa.

A screenshot of a DOSBox window. The title bar is orange and contains the text "DOSBox 0.74-2, Cpu speed: 3000 cycles, Frameskip 0, Program: FORMATO". The main window area is black with white text. The first line of text is "Hola Mundo\_".

DOSBox 0.74-2, Cpu speed: 3000 cycles, Frameskip 0, Program: FORMATO

Hola Mundo\_

**Figura 2.** Ejecución del programa.

## PARTE 2.

### IF-THEN

```
DOSBOX 0.74-2, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

X:\P7>IF2.EXE
Ingrese un dato: a
No es numero
X:\P7>IF2.EXE
Ingrese un dato: 5
Es numero
X:\P7>
```

### IF-THEN-ELSE

```
Ingrese un dato: 5
Ingrese un dato: 5
Son iguales
X:\P7>_
```

```
Ingrese un dato: 9
Ingrese un dato: 0
No son iguales
X:\P7>
```

### CASE

```
Opcion 1
Opcion 2
Opcion 3
Opcion 4
Opcion 5
Elija una opcion: 4
Opcion 4
X:\P7>
```

### FOR

```
1
2
3
4
5
6
7
8
9
```

```
X:\P7>_
```

## WHILE-DO

```
Dentro del ciclo
Dentro del ciclo
Dentro del ciclo
Dentro del ciclo
Dentro del ciclo
Dentro del ciclo
```

```
Fin del ciclo
```

```
X:\P7>
```

## DO-WHILE

DOSBOX 0.74-2, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
X:\P7>DOWHILE.EXE
```

```
Presione "s" para salir: 1
Presione "s" para salir: 2
Presione "s" para salir: 3
Presione "s" para salir: 4
Presione "s" para salir: 5
Presione "s" para salir: 1
Presione "s" para salir: 2
Presione "s" para salir: 3
Presione "s" para salir: 1
Presione "s" para salir: 2
Presione "s" para salir: 2
Presione "s" para salir: 3
Presione "s" para salir: 1
Presione "s" para salir: 2
Presione "s" para salir: 3
Presione "s" para salir: 1
Presione "s" para salir: 2
Presione "s" para salir: 2
Presione "s" para salir: 3
Presione "s" para salir: 1
Presione "s" para salir: 2
Presione "s" para salir: 2
Presione "s" para salir: 3
Presione "s" para salir: 1
Presione "s" para salir: 2
Presione "s" para salir: s
```

```
Fin del ciclo
```

```
X:\P7>
```

## **Conclusión**

Con la realización de esta practica se pudo analizar los ciclos de control que hemos manejado en otros lenguajes de programación de alto nivel, pero ahora aplicados en ensamblador. La práctica, aunque no fue difícil de realizar, tuvo partes en las cuales se tenia que analizar bien el problema ya que no se puede pensar en aplicarlo en un lenguaje y después traducirlo a ensamblador, ya que en ocasiones difiere.

## **Bibliografía**

<https://nptel.ac.in/courses/108107029/module6/lecture36.pdf>

<http://www.cs.virginia.edu/~evans/cs216/guides/x86.html>

[http://www.keil.com/support/man/docs/a166/a166\\_st\\_endp.htm](http://www.keil.com/support/man/docs/a166/a166_st_endp.htm)

[http://spot.pcc.edu/~wlara/asmx86/asmx86\\_manual\\_8.pdf](http://spot.pcc.edu/~wlara/asmx86/asmx86_manual_8.pdf)