
 GOBIERNO DEL ESTADO DE MÉXICO	<h1 style="text-align: center;">PRÁCTICA 1</h1> <p style="text-align: center;">Ing. y Esp. Rodolfo Guadalupe Alcántara Rosales</p>	 TECNOLOGICO DE ESTUDIOS SUPERIORES JILOTEPEC
---	--	--

NOMBRE DE LA PRÁCTICA	ESCRITURA DE UNA CADENA DE CARACTERES			No.	UNIDAD 1
ASIGNATURA:	LENGUAJE INTERFAZ	CARRER A:	ISIC	PLAN:	ISIC-2010-204

Nombre: Jesús Navarrete Martínez  
Grupo: 3501

Objetivo: Desplegar una cadena de caracteres en lenguaje ensamblador, utilizando los registros de datos

Utilizando los registros acumuladores y de datos, elabora un programa en ensamblador que permita desplegar una cadena de caracteres, utilizando las instrucciones respectivas para ubicar el inicio de texto. Indica lo que realiza cada renglón.

Escribe las instrucciones y captura de pantalla que demuestre que el programa si corrió:

### Código Fuente Escrito

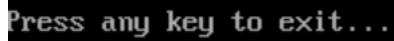
```

1 CR EQU 13      ; Definición de CR (Carriage Return) como 13
2 LF EQU 10      ; Definición de LF (Line Feed) como 10
3
4 ; Datos
5 SEGMENT
6 LINEA1 DB CR, LF, 'Jesus Navarrete Martinez', CR, LF, '$'
7 LINEA2 DB 'Tecnologico de Estudios Superiores de Jiotepec', CR, LF, '$'
8 LINEA3 DB 'ING EN SISTEMAS COMPUTACIONALES', CR, LF, '$'
9 LINEA4 DB 'Jesus :-)', CR, LF, '$'
10 Datos ENDS
11
12 ; Pila
13 SEGMENT STACK
14 DB 64 DUP(' ') ; Reserva de 64 bytes para la pila
15 Pila ENDS
16
17 ; Código
18 SEGMENT
19 ASSUME CS:CODIGO, DS:Datos, SS:PILA ; Asume los segmentos correspondientes
20
21 ; Procedimiento principal
22 INICIO:
23     MOV AX, Datos ; Carga la dirección de segmento de Datos en AX
24     MOV DS, AX    ; Mueve el valor de AX a DS para trabajar con el segmento de datos
25
26     ; Escribe la primera línea
27     LEA DX, LINEA1 ; Carga la dirección de LINEA1 en DX
28     CALL ESCRIBE   ; Llama al procedimiento ESCRIBE para mostrar la línea
29
30     ; Escribe la segunda línea
31     LEA DX, LINEA2 ; Carga la dirección de LINEA2 en DX
32     CALL ESCRIBE   ; Llama al procedimiento ESCRIBE para mostrar la línea
33
34     ; Escribe la tercera línea
35     LEA DX, LINEA3 ; Carga la dirección de LINEA3 en DX
36     CALL ESCRIBE   ; Llama al procedimiento ESCRIBE para mostrar la línea
37
38     ; Escribe la cuarta línea
39     LEA DX, LINEA4 ; Carga la dirección de LINEA4 en DX
40     CALL ESCRIBE   ; Llama al procedimiento ESCRIBE para mostrar la línea
41
42     ; Termina el programa
43     MOV AX, 4C00H ; Carga la función de terminación de programa en AX
44     INT 21H       ; Interrupción para terminar el programa
45
46 ; Procedimiento para escribir cadenas
47 ESCRIBE PROC
48     MOV AH, 9      ; Función 9 de la interrupción 21H: Mostrar cadena de caracteres
49     INT 21H       ; Interrupción 21H para mostrar en pantalla
50     RET           ; Retorno al procedimiento llamante
51 ESCRIBE ENDP      ; Fin del procedimiento ESCRIBE
52
53 CODIGO ENDS      ; Marca el final del programa y establece el punto de inicio
54 END INICIO

```



Ing. y Esp. Rodolfo Guadalupe Alcántara Rosales



### CUESTIONARIO:

1. **¿Qué dificultades encuentras al programar en ensamblador?**

- Complejidad y bajo nivel de abstracción: El ensamblador trabaja directamente con el hardware y requiere un conocimiento profundo de la arquitectura del procesador, lo que puede hacer que sea difícil de aprender y comprender.
- Propenso a errores: Debido a la precisión requerida, un pequeño error puede tener consecuencias serias y difíciles de depurar.
- Mantenimiento complicado: Los programas en ensamblador son difíciles de leer y mantener, ya que suelen ser muy largos y detallados, y carecen de la claridad que tienen los lenguajes de alto nivel.
- Portabilidad limitada: El código ensamblador está estrechamente ligado a una arquitectura de procesador específica, lo que significa que no es fácilmente portable entre diferentes sistemas.

2. **¿Consideras importante el uso de ensamblador en la programación? ¿Por qué?**



Sí, el uso del ensamblador sigue siendo importante en algunas áreas, aunque no es tan común como en el pasado. Es crucial en situaciones donde se necesita:

- **Optimización extrema:** Para mejorar el rendimiento en aplicaciones críticas como sistemas embebidos, controladores de dispositivos o videojuegos.
- **Control directo del hardware:** En áreas donde se requiere un control fino del hardware, como en la programación de microcontroladores o sistemas operativos.
- **Seguridad y sistemas críticos:** En sistemas donde la seguridad y la eficiencia son primordiales, ensamblador puede garantizar que se minimicen los errores y se maximice el rendimiento.

3. **¿Consideras que un lenguaje de alto nivel no requiere el uso de ensamblador? Explica.**

En la mayoría de los casos, los lenguajes de alto nivel no requieren el uso directo de ensamblador porque los compiladores se encargan de traducir el código de alto nivel a instrucciones de máquina de manera eficiente. Los lenguajes de alto nivel abstraen los detalles del hardware, lo que permite a los programadores centrarse en la lógica del programa en lugar de en la arquitectura del procesador.

Sin embargo, en ciertas áreas como la programación de sistemas embebidos o la optimización de código para rendimiento, el uso ocasional de ensamblador o intrínsecos específicos puede ser útil o incluso necesario. En la mayoría de los proyectos, combinar ensamblador con lenguajes de alto nivel es raro, y suele reservarse solo para situaciones específicas donde es esencial el control de bajo nivel.

 GOBIERNO DEL ESTADO DE MÉXICO	<h1>PRÁCTICA 1</h1> Ing. y Esp. Rodolfo Guadalupe Alcántara Rosales	 TECNOLOGICO DE ESTUDIOS SUPERIORES JILOTEPEC
---	--	--

## CONCLUSIONES:

En esta práctica de programación en ensamblador, hemos demostrado cómo utilizar los registros acumuladores y de datos para manipular y desplegar una cadena de caracteres. A través del uso de instrucciones específicas, como la ubicación del inicio del texto en memoria, hemos controlado el flujo del programa y gestionado el almacenamiento temporal de los datos en los registros.

Este ejercicio refuerza la importancia de comprender el funcionamiento a bajo nivel de un procesador, ya que el ensamblador requiere un manejo preciso de cada instrucción y registro. Además, hemos evidenciado cómo la manipulación directa de la memoria y los registros proporciona un control completo sobre el hardware, aunque requiere una atención rigurosa para evitar errores.