

UNIVERSIDAD AUTÓNOMA METROPOLITANA
"AZCAPOTZALCO"
PRACTICAS DEL LABORATORIO DE
SISTEMAS DIGITALES CON MICROCONTROLADORES
M. en C. Ma. ANTONIETA GALVÁN

Práctica 1

"Introducción al MASM"

Objetivos

- Familiarizarse y aprender a usar el entorno de MASM.
- Identificar los pasos involucrados en la creación de un programa en lenguaje ensamblador.
- Analizar un programa en lenguaje ensamblador.

Trabajo a Desarrollar

Se analizará el código fuente de un programa en lenguaje ensamblador 8086 para, ensamblarlo, ligarlo y ejecutarlo. A fin de pasar del código fuente al código objeto y de este al programa ejecutable.

Instrucciones

1. Ejecutar el entorno de MASM (c:\Irvine\masm.bat) y capturar el siguiente código

```
title Test Alphabetic Input      (Pract1.ASM)
```

```
    ; This program reads and displays characters  
    ; until a non-alphabetic character is entered.
```

```
pila segment stack
```

```
    DB 100h DUP ('stack')
```

```
    TOS EQU THIS WORD
```

```
pila ends
```

```
codigo segment 'code'
```

```
    assume ss:pila, cs:codigo
```

```
main proc far
```

```
    mov ax, seg pila
```

```
    mov ss,ax
```

```
L1: mov ah,1      ; input a character
```

```
    int 21h      ; AL = character
```

```
    call Isalpha ; test value in AL
```

```
    jnz exit     ; exit if not alphabetic
```

```
    jmp L1       ; continue loop
```

```
exit:
```

```
    mov ax,4C00h ; return to DOS
```

UNIVERSIDAD AUTÓNOMA METROPOLITANA
"AZCAPOTZALCO"
PRACTICAS DEL LABORATORIO DE
SISTEMAS DIGITALES CON MICROCONTROLADORES
M. en C. Ma. ANTONIETA GALVÁN

```
int 21h
main endp

; Isalpha sets ZF = 1 if the character
; in AL is alphabetic.

Isalpha proc far
    push ax        ; save AX
    and al,11011111b ; clear bit 5
    cmp al,'A'     ; check 'A'..'Z' range
    jb B1
    cmp al,'Z'
    ja B1
    test ax,0      ; ZF = 1
B1:
    pop ax         ; restore AX
    ret
Isalpha endp
codigo ends
end main
```

2. Guarda el archivo con extensión asm, por ejemplo *prac1a.asm*. Para ensamblar archivo fuente selecciona la opción **assemble** del menú **run**. En caso de existir errores, éstos deberán ser corregidos. Observa que por cada error se muestra el número de renglón entre paréntesis para facilitar la ubicación la instrucción errónea, así mismo se proporciona una breve descripción del error encontrado.
3. Para ligar el código objeto utiliza la función **link** del menú **run**.
4. Ahora, para depuración del programa utiliza la función **debug** del menú **run**.
5. Ejecuta el programa. Para ello debes seleccionar del menú **run** la función **run**.
6. Juega un poco con el programa y descubre lo que hace.

UNIVERSIDAD AUTÓNOMA METROPOLITANA
"AZCAPOTZALCO"
PRACTICAS DEL LABORATORIO DE
SISTEMAS DIGITALES CON MICROCONTROLADORES
M. en C. Ma. ANTONIETA GALVÁN

CUESTIONARIO PRÁCTICA 1

1. ¿Cuál es la estructura (esqueleto) de un programa en ensamblador del 8086?
2. ¿Para qué se utiliza *int 21h*? Proporciona un ejemplo.
3. ¿Cómo se realiza el proceso de ensamblado y ligado?
4. ¿Qué tipo de instrucciones recuerdas haber utilizado en el programa de la práctica realizada?
5. ¿Para qué se utiliza el **debug**?
6. ¿Qué información se puede obtener del depurador?
7. ¿Qué función realizaba el programa de la práctica 1a?
8. ¿Qué funciones se le agregó al programa original?
9. ¿Porqué se agregó el segmento de datos?
10. Cuando se agregó el segmento de datos, qué más fue necesario agregar en el programa.
11. ¿Qué es el programa fuente?
12. ¿Qué es el programa objeto?
13. ¿Cómo se termina un programa para regresar el control al sistema operativo?