

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Ciencias Químicas e Ingeniería



Práctica 2

Materia:
Organización y Arquitectura de Computadoras

Docente:
Garcia Rocha Jose Isabel

Elaborado por:
Liang Liu Kevin

Tijuana, B.C. a 18 de febrero de 2026

Objetivo

Distinguir las características de la organización y arquitectura del microprocesador de una computadora de propósito general, analizando sus recursos de hardware y software, para conocer capacidades y limitaciones de forma organizada y responsable.

Desarrollo

En el código se deben agregar las siguientes partes:

1. En él .data agregar una constante de texto:

```
C/C++  
msj: db 'Ingrese un dígito (0-9)',0x0A  
len: equ $-msj
```

Esto define una cadena de texto con salto de línea (0x0A) y una constante len que guarda la longitud del mensaje.

2. En él .bss agregar una variable:

```
C/C++  
num1 resb 1
```

Aquí se reserva un byte para almacenar el valor ingresado.

3. En él .text se debe colocar el siguiente código:

```
C/C++  
mov eax, 4  
mov ebx, 1  
mov ecx, msj  
mov edx, len  
int 80h
```

¿Qué piensa usted que hará este código?

```
Practica2 > ASM codigo.asm
1  %include "./pc_io.inc"
2  global _start
3
4 > section .data ...
11
12 > section .bss ...
18
19 section .text
20 _start:
21
22     ;INSTRUCCION 3
23
24     ;sys_write(stdout, message, length)
25     mov eax, 4
26     mov ebx, 1
27     mov ecx, msj
28     mov edx, len
29     int 80h
```

Imprime la cadena almacenada en 'msj' a la terminal.

¿Qué hizo realmente este código?

```
● @liang8451 → /workspaces/OAC_2026_01/Practica2 (main) $ ./codigo
Ingrrese un dígito (0-9)
```

Imprime la cadena “Ingrrese un dígito (0-9)” a la terminal.

```
C/C++
mov eax, 3
mov ebx, 0
mov ecx, num1
mov edx, 1
int 80h
```

¿Qué piensa usted que hará este código?

```
20      start:  
21  
22          ;INSTRUCCION 3  
23  
24          ;sys_write(stdout, message, length)  
25          mov eax, 4  
26          mov ebx, 1  
27          mov ecx, msj  
28          mov edx, len  
29          int 80h  
30  
31          mov eax, 3  
32          mov ebx, 0  
33          mov ecx, num1  
34          mov edx, 1  
35          int 80h
```

Va a pedir una entrada de un byte al usuario y lo almacena en la variable num1

¿Qué hizo realmente este código?

- @liang8451 → /workspaces/OAC_2026_01/Practica2 (main) \$./codigo
Ingrrese un dígito (0-9)
9

Espera a que el usuario ingrese una entrada con el teclado y de ENTER

C/C++

```
mov eax, 4  
mov ebx, 1  
mov ecx, num1  
mov edx, 1  
int 80h
```

¿Qué piensa usted que hará este código?

```
21      ;INSTRUCCION 3
22
23      ;sys_write(stdout, message, length)
24      mov eax, 4
25      mov ebx, 1
26      mov ecx, msj
27      mov edx, len
28      int 80h
29
30
31      mov eax, 3
32      mov ebx, 0
33      mov ecx, num1
34      mov edx, 1
35      int 80h
36
37      sys_write(stdout, message, length)
38      mov eax, 4
39      mov ebx, 1
40      mov ecx, num1
41      mov edx, 1
42      int 80h
```

Va a mostrar en la terminal el primer carácter que el usuario ingreso.

¿Qué hizo realmente este código?

- [@liang8451 → /workspaces/OAC_2026_01/Practica2 \(main\)](#) \$./codigo
Ingrrese un dígito (0-9)
9
- [9@liang8451 → /workspaces/OAC_2026_01/Practica2 \(main\)](#) \$

Muestra en la terminal el primer carácter que el usuario ingreso.

4. Del paso anterior se debe explicar textualmente que se piensa que hace cada sección de código, con sus propias palabras.

5. Ahora el código anterior lo vamos a modificar:

a. El msj ahora debe quedar de la siguiente manera:

C/C++

```
msj: db 'Ingrrese un dígito (0-9)', 0x0
```

b. El código del paso 3 se debe comentar y agregar el siguiente:

C/C++

```
mov edx, msj
call puts
```

¿Qué piensa usted que hará este código?

```
44 ;INSTRUCCION 5  
45  
46     mov edx, msj  
47     call puts  
48
```

Va a mostrar en la terminal la cadena dentro de 'msj'

¿Qué hizo realmente este código?

- @liang8451 → /workspaces/OAC_2026_01/Practica2 (main) \$./codigo
- Ingrese un dígito (0-9)@liang8451 → /workspaces/OAC_2026_01/Practica2 (main) \$ █

Muestra en la terminal la cadena almacenada en 'msj'

C/C++

```
call getch
```

¿Qué piensa usted que hará este código?

```
44 ;INSTRUCCION 5  
45  
46     mov edx, msj  
47     call puts  
48  
49     call getch  
50
```

Va a pedir al usuario que ingrese un carácter

¿Qué hizo realmente este código?

- @liang8451 → /workspaces/OAC_2026_01/Practica2 (main) \$./codigo
- Ingrese un dígito (0-9)@liang8451 → /workspaces/OAC_2026_01/Practica2 (main) \$ █

Espera a que el usuario ingrese una entrada con el teclado

C/C++

```
call putchar
```

¿Qué piensa usted que hará este código?

```
44 ;INSTRUCCION 5  
45  
46     mov edx, msj  
47     call puts  
48  
49     call getch  
50  
51     call putchar
```

Va a mostrar en la terminal el carácter que el usuario ingreso

¿Qué hizo realmente este código?

- @liang8451 → /workspaces/OAC_2026_01/Practica2 (main) \$./codigo
- Ingrese un dígito (0-9)1@liang8451 → /workspaces/OAC_2026_01/Practica2 (main) \$

Muestra en la terminal el carácter que el usuario ingreso

6. En la parte inferior del código, vamos a agregar la siguiente función:

```
C/C++  
salto:  
    mov al, 13  
    call putchar  
    mov al, 10  
    call putchar  
    ret
```

7. Ahora el código se modifica con salto y debe quedar de la siguiente manera:

```
C/C++  
mov edx, msj  
call puts  
call salto  
call getch  
call salto  
call putchar  
call salto
```

¿Qué piensa usted que hará todo este código?

```
53 ;INSTRUCCION 7  
54  
55 mov edx, msj  
56 call puts  
57 call salto  
58 call getch  
59 call salto  
60 call putchar  
61 call salto
```

Muestra en la terminal la cadena dentro de ‘msj’, luego hace un salto de linea, luego pide al usuario que ingrese un caracter, luego hace un salto de linea, luego imprime el caracter ingresado a la terminal y al ultimo hace un salto de linea.

¿Qué hizo realmente todo este código?

- @liang8451 → /workspaces/OAC_2026_01/Practica2 (main) \$./codigo
Ingrese un dígito (0-9)

- @liang8451 → /workspaces/OAC_2026_01/Practica2 (main) \$ █

Muestra en la terminal “Ingrese un dígito (0-9)”, luego espera a que el usuario ingrese un caracter y termina.

8. Vamos y modificamos la función de salto:

```
C/C++  
salto:  
pushad  
mov al, 13  
call putchar  
mov al, 10  
call putchar  
popad  
ret
```

¿Qué hizo realmente todo este código?

- @liang8451 → /workspaces/OAC_2026_01/Practica2 (main) \$./codigo
Ingrrese un dígito (0-9)

9

- @liang8451 → /workspaces/OAC_2026_01/Practica2 (main) \$ █

Muestra en la terminal “Ingrrese un digito (0-9)”, luego espera a que el usuario ingrese un carácter y muestra el carácter ingresado en la terminal.

9. Ahora agregamos la funcionalidad de capturar una letra, por ejemplo la a y sumarle un 1 e imprimirla en terminal.

Agregamos el siguiente código después del último call salto:

C/C++

```
mov ebx, num1
mov byte[ebx], al
add byte[ebx], 1
mov al, byte[ebx]
call putchar
```

¿Qué piensa usted que hará todo este código?

63	;INSTRUCCION 9
64	
65	mov ebx, num1
66	mov byte[ebx], al
67	add byte[ebx], 1
68	mov al, byte[ebx]
69	call putchar

Suma 1 al carácter guardado en ‘al’ y lo imprime en la terminal

¿Qué hizo realmente todo este código?

- @liang8451 → /workspaces/OAC_2026_01/Practica2 (main) \$./codigo
Ingrrese un dígito (0-9)

9

- :@liang8451 → /workspaces/OAC_2026_01/Practica2 (main) \$ █

Suma 1 al carácter guardado en ‘al’ y lo imprime en la terminal

10. Ahora para la prueba capture 3 y súmele 1 como el caso anterior.

11. Ahora capture otro número, modifique lo necesario para ello, y pruebe con una suma de $3 + 4 = 7$:

C/C++

```
mov ebx, num1
mov al, [ebx]
mov ebx, num2
add [ebx], al
mov al, byte[ebx]
call putchar
```

¿Qué piensa usted que hará todo este código?

```
70      ;INSTRUCCION 11
71
72      mov edx, msj
73      call puts
74      call salto
75      call getch
76      mov [num1], al
77      call putchar
78      call salto
79
80      call puts
81      call salto
82      call getch
83      mov [num2], al
84      call putchar
85      call salto
86
87      mov ebx, num1
88      mov al, [ebx]
89      mov ebx, num2
90      add [ebx], al
91      mov al, byte[ebx]
92      call putchar
93      |
```

Pedira al usuario ingresar dos digitos, los sumara y va a imprimir el resultado en la terminal

¿Qué hizo realmente todo este código?

```
● @liang8451 → /workspaces/OAC_2026_01/Practica2 (main) $ ./codigo  
Ingrrese un dígito (0-9)  
3  
Ingrrese un dígito (0-9)  
4  
g
```

Pide al usuario ingresar dos dígitos, suma sus valores en ascii y lo muestra en la terminal

Dificultades

No hubo mayores dificultades, se copió y pegó pedazos de código en cada paso de instrucción y se redactó el funcionamiento hipotético y real de cada pedazo de código.

Conclusión

A través de la realización de esta práctica, se analizó el funcionamiento de las interrupciones del sistema (int 80h), comprendiendo cómo el sistema operativo gestiona las operaciones básicas de entrada/salida.

La modificación de la función salto ilustró la importancia de preservar el estado de los registros utilizando pushad y popad para asegurar el correcto funcionamiento del programa.