

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE INGENIERÍA ARQUITECTURA Y DISEÑO

INGENIERÍA EN SOFTWARE Y TECNOLOGÍAS EMERGENTES



ORGANIZACIÓN DE COMPUTADORAS

## Taller 8

ADRIAN BALDERAS ROSAS

Jonatan Crespo Ragland

## a. ¿Qué imprime el programa y por qué? (Con documentación en el código)

Este programa realiza algunas operaciones aritméticas, lógicas y de manipulación de bits. Aquí está el código con comentarios detallados:

```

1 * section .data
2     msg db "Resultado: ", 0 ; Mensaje que se imprimirá
3     newline db 0xA ; Carácter de nueva línea (ASCII 0x0A)
4
5 * section .bss
6     res resb 4 ; Reserva 4 bytes para almacenar el resultado
7
8 section .text
9 global _start
10
11 _start:
12     ; Instrucciones aritméticas
13     mov eax, 10 ; Carga el valor 10 en el registro EAX
14     mov ebx, 5 ; Carga el valor 5 en el registro EBX
15     add eax, ebx ; Suma EAX y EBX, resultado: EAX = 15
16
17     ; Instrucción lógica (AND)
18     and eax, 0xF ; Realiza AND lógico entre EAX y 0xF (15 en decimal)
19     ; Como EAX ya es 15, el resultado de AND es 15
20
21     ; Instrucciones de manipulación de bits
22     shl eax, 1 ; Realiza un desplazamiento lógico a la izquierda (SHL) en EAX por 1 bit
23     ; EAX = 15 << 1 (desplazar a la izquierda): resultado EAX = 30
24
25     ; Guardar el resultado en la sección .bss
26     mov [res], eax ; Almacena el valor de EAX (30) en la variable 'res'
27
28     ; Imprimir el mensaje "Resultado: "
29     mov eax, 4 ; Syscall número 4 para escribir
30     mov ebx, 1 ; Salida estándar (pantalla)
31     mov ecx, msg ; Dirección del mensaje a imprimir
32     mov edx, 11 ; Longitud del mensaje (11 caracteres)
33     int 0x80 ; Interrupción para imprimir el mensaje
34
35     ; Imprimir el número (resultado almacenado en 'res')
36     mov eax, [res] ; Carga el valor almacenado en 'res' (30) en EAX
37     add eax, '0' ; Convierte el número en su equivalente ASCII sumando el valor '0'
38     ; 30 + '0' = carácter con valor 30 en ASCII (no imprimible)
39
40     mov [res], eax ; Almacena el carácter ASCII en 'res'
41     mov eax, 4 ; Syscall número 4 para escribir
42     mov ebx, 1 ; Salida estándar
43     mov ecx, res ; Dirección del resultado
44     mov edx, 1 ; Longitud de 1 carácter
45     int 0x80 ; Interrupción para imprimir el carácter
46
47     ; Imprimir una nueva línea
48     mov eax, 4 ; Syscall número 4 para escribir
49     mov ebx, 1 ; Salida estándar
50     mov ecx, newline ; Dirección del carácter de nueva línea
51     mov edx, 1 ; Longitud de 1 carácter
52     int 0x80 ; Interrupción para imprimir nueva línea
53
54     ; Terminar el programa
55     mov eax, 1 ; Syscall para salir
56     xor ebx, ebx ; Código de salida 0
57     int 0x80 ; Interrupción para terminar el programa

```

