



**Universidad Autónoma de Baja California  
Facultad de Ingeniería Arquitectura y Diseño**

**Organización de Computadoras**

**Docente:**

**Crespo Regland Jonatan**

**Alumno:**

**Garcia Ceseña Edgar Rafael**

**GRUPO 932**

**Ingeniería en Software y Tecnologías Emergentes**

**TALLER 8**



```

1 section .data
2 msg db 'Resultado: ', 0 ; Mensaje para imprimir
3 newline db 0xA ; Nueva línea (salto de línea)
4
5 section .bss
6 res resb 4 ; Espacio para el resultado
7
8 section .text
9 global _start
10 _start:
11 ; Instrucciones aritméticas
12 mov eax, 10 ; Coloca 10 en el registro EAX
13 mov ebx, 5 ; Coloca 5 en el registro EBX
14 add eax, ebx ; Suma EAX + EBX (10 + 5 = 15)
15
16 ; Instrucción lógica (AND)
17 and eax, 0xF ; Realiza AND bit a bit entre EAX y 15 (EAX será 15)
18
19 ; Instrucciones de manipulación de bits
20 shl eax, 1 ; Desplaza los bits de EAX una posición a la izquierda (15 << 1 = 30)
21
22 ; Guardar el resultado en la sección .bss
23 mov [res], eax ; Almacena el valor de EAX (30) en res
24
25 ; Imprimir "Resultado: "
26 mov eax, 4 ; Syscall para escribir
27 mov ebx, 1 ; Salida estándar
28 mov ecx, msg ; Mensaje a imprimir
29 mov edx, 11 ; Longitud del mensaje
30 int 0x80 ; Interrupción para ejecutar syscall de impresión

```

---

```

31
32 ; Imprimir el número en ASCII
33 mov eax, [res] ; Cargar el resultado (30) en EAX
34 add eax, '0' ; Convertir el número a carácter ASCII sumando 48
35 mov [res], eax ; Almacenar el carácter en res
36
37 mov eax, 4 ; Syscall para escribir

```

```
37 mov eax, 4 ; Syscall para escribir
38 mov ebx, 1 ; Salida estándar
39 mov ecx, res ; Dirección del carácter ASCII
40 mov edx, 1 ; Longitud de 1 carácter
41 int 0x80 ; Interrupción para ejecutar syscall de impresión
42
43 ; Imprimir nueva línea
44 mov eax, 4 ; Syscall para escribir
45 mov ebx, 1 ; Salida estándar
46 mov ecx, newline ; Dirección de la nueva línea
47 mov edx, 1 ; Longitud de 1 carácter
48 int 0x80 ; Interrupción para ejecutar syscall de impresión
49
50 ; Salir del programa
51 mov eax, 1 ; Syscall para salir
52 xor ebx, ebx ; Código de salida 0
53 int 0x80 ; Interrupción para finalizar
```

---