

TALLER #11 <https://onecompiler.com/assembly/445rnr3r5>

```
section .data
msg db "Resultado: ", 0
len equ $ - msg
err db "Error: division por cero", 0
lenErr equ $ - err
newline db 10, 0
lenNL equ $ - newline
section .bss
resultado resb 1 ; solo un dígito como en el original
section .text
global _start
_start:

; =====
; Números hardcoded (ASCII -> entero)
; =====

mov al, '8' ; primer número (ASCII)
sub al, '0' ; convertir a entero (8)
mov bl, '2' ; segundo número (ASCII)
sub bl, '0' ; convertir a entero (2)
; =====
; Verificar división por cero
; =====
cmp bl, 0 ; ¿BL == 0?
jz div_zero ; si es cero, saltar a manejo de error
; =====
; División AL / BL
; =====
xor ah, ah ; limpiar AH para div (AL/BL)
div bl ; cociente en AL, residuo en AH
; =====
; Convertir resultado a ASCII (un dígito)
; =====
add al, '0'
mov [resultado], al
```

```

; =====
; Imprimir "Resultado: "
; =====
mov eax, 4
mov ebx, 1
mov ecx, msg
mov edx, len
int 0x80
; =====
; Imprimir el resultado
; =====
mov eax, 4
mov ebx, 1
mov ecx, resultado
mov edx, 1
int 0x80
jmp fin_linea ; continuar al salto de línea y salida
div_zero:
; =====
; Imprimir mensaje de error
; =====
mov eax, 4

mov ebx, 1
mov ecx, err
mov edx, lenErr
int 0x80
fin_linea:

; =====
; Imprimir salto de línea
; =====
mov eax, 4
mov ebx, 1
mov ecx, newline
mov edx, lenNL
int 0x80
; =====
; Salir
; =====
mov eax, 1
xor ebx, ebx
int 0x80

```