

Codigo 3: Tabla del 5

```
section .data
    factor dd 5
    newline db 10

section .bss
    buffer resb 12

section .text
    global _start

_start:
    mov ecx, 1

loop_tabla:
    mov eax, [factor]
    mul ecx

    ; convertir eax a ASCII
    lea edi, [buffer + 11]
    mov byte [edi], 0
    mov ebx, 10

conv:
    xor edx, edx
    div ebx
    add dl, '0'
    dec edi
    mov [edi], dl
    test eax, eax
    jnz conv

    ; imprimir número
    mov eax, 4
    mov ebx, 1
    mov ecx, edi
    mov edx, buffer + 12
    sub edx, ecx
    int 0x80

    ; salto de línea
    mov eax, 4
    mov ecx, newline
    mov edx, 1
    int 0x80

    inc ecx
```

```
cmp ecx, 11
jl loop_tabla
```

```
mov eax, 1
xor ebx, ebx
int 0x80
```

Este programa genera la tabla de multiplicar de un número específico (en este caso, el número 5 definido como factor). Para lograrlo, recorre los valores del 1 al 10 multiplicando cada uno por el número base y luego imprime el resultado en pantalla. La estructura del programa utiliza un bucle que controla el índice de multiplicación usando el registro `ecx`, comenzando en 1 y repitiendo hasta llegar a 10.

Antes de realizar la multiplicación, el valor de `ecx` se guarda en el registro `esi` para no perderlo, ya que las instrucciones `mul` y `div` modifican registros como `eax`, `edx` y a veces afectan valores que se desean conservar. Luego se multiplica el número en `factor` por el índice actual y se almacena el resultado en `eax`.

A continuación, el programa convierte el resultado de la multiplicación en una cadena de texto en formato ASCII. Para eso, apunta al final de un búfer reservado en la sección `.bss` y va llenándolo hacia atrás con los dígitos obtenidos al dividir repetidamente el número entre 10. Cada resto de la división representa un dígito decimal, que se convierte a su equivalente ASCII sumándole el valor de `'0'`.

Una vez convertida la cantidad, el programa calcula cuántos caracteres componen el número y lo imprime en la pantalla utilizando la llamada al sistema `int 0x80` con el servicio `sys_write`. Luego imprime un salto de línea para separar los resultados de forma legible.

El índice del bucle se incrementa y se restaura en `ecx` para continuar con la siguiente multiplicación. Este proceso se repite hasta completar la tabla desde 5×1 hasta 5×10 . Finalmente, el programa finaliza limpiamente usando `sys_exit`, indicando que se ejecutó sin errores. De este modo, el programa muestra la tabla de multiplicar de forma ordenada y eficiente, manipulando correctamente los registros para mantener el control del flujo.