Código 5: Suma de Matrices 2x2

```
section .data
   matriz1 dd 1, 2, 3, 4      ; Matriz 2x2
   matriz2 dd 5, 6, 7, 8
   espacio db ' '
   newline db 10

section .bss
   resultado resd 4
   buffer resb 12

section .text
   global _start

_start:
   xor esi, esi            ; Índice = 0

sumar:
   mov eax, [matriz1 + esi*4]
   add eax, [matriz2 + esi*4]
   mov [resultado + esi*4], eax
   inc esi
   cmp esi, 4
   jne sumar

   xor esi, esi            ; Índice = 0

imprimir:
   mov eax, [resultado + esi*4]
   call imprimir_num

   ; Imprimir espacio o salto de línea
   inc esi
   test esi, 1             ; ¿Es impar? (cada 2 números damos salto)
   jnz imprime_espacio
   call salto_linea

imprime_espacio:
   cmp esi, 4
   jl imprimir

   ; Salir
   mov eax, 1
   xor ebx, ebx
   int 0x80

; ------------------------------------------------
```

```
; Subrutina: imprimir número (en EAX)
; ------------------------------------------------
imprimir_num:
   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

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

; ------------------------------------------------
; Subrutina: salto de línea
; ------------------------------------------------
salto_linea:
   mov eax, 4
   mov ebx, 1
   mov ecx, newline
   mov edx, 1
   int 0x80
   ret
```

El programa realiza la suma de dos matrices de 2x2, cuyos valores están definidos al inicio de la sección .data. Las matrices matriz1 y matriz2 contienen cuatro enteros cada una (representando una matriz de dos filas por dos columnas). El resultado se guarda en una tercera matriz, llamada resultado, que se reserva en la sección .bss.

Para sumar las matrices, el programa utiliza un bucle controlado por el registro esi. En cada iteración del bucle sumar, carga un elemento de matriz1, le suma el valor correspondiente de matriz2, y guarda el resultado en la posición correspondiente de la matriz resultado.

Después de realizar la suma, se utiliza otro bucle (imprimir) para mostrar los valores de la matriz resultante. Para eso, cada número se convierte a ASCII con una subrutina llamada imprimir_num, que divide el número por 10 repetidamente para obtener sus dígitos (de manera similar a cómo se convierte cualquier número entero a texto). El resultado se guarda en un búfer llamado buffer, y luego se imprime en pantalla usando la llamada al sistema int 0x80.

El programa organiza la impresión de forma que cada dos números se imprime un salto de línea. Esto se hace verificando si el índice actual es impar usando test esi, 1. Si lo es, se imprime un espacio; si no, se llama a la subrutina salto_linea, que imprime un carácter de nueva línea.

Finalmente, después de imprimir toda la matriz resultado, el programa se termina de forma limpia usando sys_exit con el valor de salida 0 (mov eax, 1; xor ebx, ebx; int 0x80).

Este programa demuestra cómo hacer operaciones con arreglos (vectores o matrices) y cómo mostrar resultados en pantalla usando llamadas al sistema en un entorno Linux de 32 bits. Además, enseña cómo usar subrutinas para mejorar la organización del código y cómo convertir números a texto sin necesidad de bibliotecas externas.