

Cual es la importancia del "%" en macros ensamblador x86

Definir macros con parámetros

Cuando defines una macro con parámetros, usas el % para referenciar estos parámetros dentro del cuerpo de la macro.

Le indica al preprocesador de NASM que se trata de un parámetro, no de un valor literal.

La forma %1, %2, etc., representa los argumentos posicionales.

Llamar a una macro

Cuando llamas a una macro, simplemente escribes su nombre y pasas los argumentos separados por comas. No usas al llamar, solo al referenciar parámetros dentro de la macro. Aunque no se usa en la llamada, la expansión se usa internamente para sustituir argumentos.

Macros con parámetros opcionales

NASM permite definir macros con parámetros opcionales especificando el número máximo de argumentos y comprobando si se pasaron todos.

%0 indica cuántos argumentos fueron pasados a la macro.

%+ permite concatenar símbolos o partes de código.

%ifdef y %if %0 permiten verificar la cantidad de argumentos.

Genera estructuras de datos en ensamblador, para simular nuevos tipos de datos como fecha dd/mm/yyyy, correo electrónico, dirección completa (compuesto de datos como calle, número de casa y colonia) y una cadena de texto tipo curp. Además, añade ejemplos de cómo podías acceder, manipular y objetivos de utilizar esos tipos de datos. Debes tratar de usar arreglos y/o matrices o en los ejemplos

Fecha

```
section .data
```

```
fecha1:
```

```
    dd 1      ; Día (dd)
```

```
    dd 5      ; Mes (mm)
```

```
    dd 2025   ; Año (yyyy)
```

Acceder

```
mov eax, [fecha1]    ; Día
```

```
mov ebx, [fecha1 + 4] ; Mes
```

```
mov ecx, [fecha1 + 8] ; Año
```

Correo electrónico

```
section .data
correo1 db "usuario@dominio.com", 0
```

Acceder

```
mov edx, len_correo
mov ecx, correo1
mov ebx, 1
mov eax, 4
int 0x80
```

```
len_correo equ $ - correo1
```

Direccion

```
section .data
direccion1:
    calle    db "Av. Siempre Viva", 0
    numero   db "742", 0
    colonia  db "Springfield", 0
```

Acceder

```
mov ecx, calle    ; Imprimir calle
mov ecx, numero   ; Imprimir número
mov ecx, colonia  ; Imprimir colonia
```

Curp

```
section .data
curp1 db "GARC850101HMCLRL09", 0
```

Acceder

```
mov ecx, curp1
; Puedes recorrer carácter por carácter si es necesario
```

Código 111

```
section .data
    num1 db 5          ; Define el primer número (5)
    num2 db 11         ; Define el segundo número (11)
    result db 0         ; Variable donde se almacenará el resultado
    message db "Resultado: ", 0 ; Mensaje a mostrar antes del número

section .bss
    buffer resb 4       ; Espacio reservado para imprimir el número como texto

section .text
    global _start       ; Punto de entrada para el programa

; Macro para imprimir una cadena
%macro PRINT_STRING 1
    mov eax, 4          ; syscall número 4 = sys_write
    mov ebx, 1          ; descriptor 1 = salida estándar (pantalla)
    mov ecx, %1         ; dirección de la cadena a imprimir
    mov edx, 13         ; longitud de la cadena (13 caracteres)
    int 0x80            ; llamada al sistema
%endmacro

; Macro para imprimir un número de un dígito
%macro PRINT_NUMBER 1
    mov eax, %1         ; cargar valor en eax
    add eax, '0'        ; convertir el número a carácter ASCII
    mov [buffer], eax   ; guardar el carácter en el buffer
    mov eax, 4          ; syscall sys_write
    mov ebx, 1          ; salida estándar
    mov ecx, buffer     ; dirección del buffer
    mov edx, 1          ; longitud = 1 carácter
    int 0x80            ; llamada al sistema
%endmacro

_start:
    mov al, [num1]      ; cargar el valor de num1 en AL
    add al, [num2]      ; sumar el valor de num2 a AL
    mov [result], al    ; guardar el resultado en la variable result

    PRINT_STRING message ; imprimir el mensaje "Resultado: "
    PRINT_NUMBER [result] ; imprimir el resultado de la suma

; Salir del programa
    mov eax, 1          ; syscall número 1 = sys_exit
    mov ebx, 0          ; código de salida 0 (éxito)
    int 0x80            ; llamada al sistema para salir
```

Código 112

```
section .data
    message db "La suma de los valores es: ", 0    ; Mensaje que se imprimirá antes de
    mostrar el resultado
    newline db 10, 0                                ; Nueva línea para imprimir después del resultado
    num1 db 5                                         ; Primer número (5)
    num2 db 7                                         ; Segundo número (7)
    num3 db 3                                         ; Tercer número (3)
    result db 0                                       ; Variable para almacenar el resultado de la suma

section .bss
    buffer resb 4                                    ; Buffer para convertir el número a caracteres

section .text
    global _start                                    ; Punto de entrada del programa

; Macro para imprimir una cadena
%macro PRINT_STRING 1
    mov eax, 4                                       ; syscall número 4 = sys_write
    mov ebx, 1                                       ; descriptor 1 = salida estándar (pantalla)
    mov ecx, %1                                     ; dirección de la cadena a imprimir
    mov edx, 22                                     ; longitud de la cadena (tamaño de "La suma de los
valores es: ")
    int 0x80                                         ; llamada al sistema
%endmacro

; Macro para imprimir un número de un solo dígito
%macro PRINT_NUMBER 1
    mov al, %1                                       ; cargar el valor del número en AL
    add al, '0'                                     ; convertir el número en su carácter ASCII
    mov [buffer], al                                ; guardar el carácter en el buffer
    mov eax, 4                                       ; syscall sys_write
    mov ebx, 1                                       ; salida estándar (pantalla)
    mov ecx, buffer                                  ; dirección del buffer
    mov edx, 1                                       ; longitud = 1 carácter
    int 0x80                                         ; llamada al sistema
%endmacro

; Macro para sumar tres números
%macro SUMAR_TRES_NUMEROS 3
    mov al, [%1]                                     ; cargar el primer número en AL
    add al, [%2]                                     ; sumar el segundo número a AL
    add al, [%3]                                     ; sumar el tercer número a AL
    mov [%4], al                                     ; guardar el resultado en 'result'
%endmacro

_start:
```

```
; Sumar los tres números (num1, num2, num3) y almacenar el resultado en 'result'  
SUMAR_TRES_NUMEROS num1, num2, num3, result
```

```
; Imprime el mensaje "La suma de los valores es: "  
PRINT_STRING message
```

```
; Imprime el resultado de la suma  
PRINT_NUMBER [result]
```

```
; Imprime una nueva línea  
PRINT_STRING newline
```

```
; Salir del programa  
mov eax, 1 ; syscall número 1 = sys_exit  
mov ebx, 0 ; código de salida 0 (éxito)  
int 0x80 ; llamada al sistema
```