Universidad Autónoma de Baja California
Facultad de Ingeniería Arquitectura y Diseño

Ingeniería en Software y Tecnologías Emergentes

Materia: Organización de Computadoras

Alumno: Jesus Eduardo Rodríguez Ramírez

Profesor: Jonatan Crespo Ragland

Grupo 932

Trabajo: CÓDIGOS Taller 7

Ensenada, B.C; a 29 de octubre del 2024

Desarrollar lo siguiente en su cuaderno o computadora:

1. De acuerdo al código de prueba 1, responde y desarrolla lo siguiente:

```
section .data
    num1 db 5          ; Declarar un byte llamado num1 con el valor 5
    num2 db 11         ; Declarar un byte llamado num2 con el valor 11
    result db 0        ; Declarar un byte llamado result inicializado a 0
    msg db 'Resultado: ', 0 ; Mensaje para mostrar en la salida

section .bss
    buffer resb 4      ; Reservar 4 bytes sin inicializar para el buffer

section .text
    global _start

_start:
    mov al, [num1]     ; Mover el valor de num1 al registro AL
    add al, [num2]     ; Sumar el valor de num2 al registro AL
    mov [result], al   ; Almacenar el resultado de la suma en result

    ; Convertir el resultado numérico a su valor ASCII
    movzx eax, byte [result] ; Mover el byte en result a EAX, extendiéndolo a 32 bits
    add eax, 48        ; Convertir el valor numérico a su equivalente ASCII ('0' = 48)
    mov [buffer], al   ; Almacenar el carácter ASCII en buffer

    ; Imprimir el mensaje "Resultado: "
    mov eax, 4
    mov ebx, 1
    mov ecx, msg
    mov edx, 11
    int 0x80           ; Llamada al sistema para escribir en salida estándar

    ; Imprimir el valor en buffer (resultado en ASCII)
    mov eax, 4
    mov ebx, 1
    mov ecx, buffer
    mov edx, 1
    int 0x80           ; Llamada al sistema para escribir en salida estándar

    ; Salir del programa
    mov eax, 1
    xor ebx, ebx
    int 0x80
```

e. Modifica el programa para que imprima lo siguiente: A, \, $, & y 1. Documenta tu procedimiento.

```
section .data
    ; Valores para obtener los caracteres A, \, $, & y 1
    numA db 65      ; Valor ASCII para 'A'
    numBack db 92   ; Valor ASCII para '\'
    numDollar db 36 ; Valor ASCII para '$'
    numAmp db 38    ; Valor ASCII para '&'
    numOne db 49    ; Valor ASCII para '1'
    msg db 'Resultado: ', 0

section .bss
    buffer resb 1   ; Reservar 1 byte para almacenar cada carácter temporalmente

section .text
    global _start

_start:
    ; Imprimir el mensaje "Resultado: "
    mov eax, 4
    mov ebx, 1
    mov ecx, msg
    mov edx, 11
    int 0x80          ; Llamada al sistema para escribir en salida estándar

    ; Imprimir cada símbolo secuencialmente

    ; Cargar y mostrar 'A'
    mov al, [numA]   ; Mover el valor de 'A' al registro AL
    mov [buffer], al ; Almacenar el valor en buffer
    call print_buffer

    ; Cargar y mostrar '\'
    mov al, [numBack] ; Mover el valor de '\' al registro AL
    mov [buffer], al  ; Almacenar el valor en buffer
    call print_buffer

    ; Cargar y mostrar '$'
    mov al, [numDollar] ; Mover el valor de '$' al registro AL
    mov [buffer], al    ; Almacenar el valor en buffer
    call print_buffer

    ; Cargar y mostrar '&'
    mov al, [numAmp]    ; Mover el valor de '&' al registro AL
    mov [buffer], al    ; Almacenar el valor en buffer
    call print_buffer

    ; Cargar y mostrar '1'
    mov al, [numOne]    ; Mover el valor de '1' al registro AL
    mov [buffer], al    ; Almacenar el valor en buffer
    call print_buffer

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

; Subrutina para imprimir el contenido de buffer
print_buffer:
    mov eax, 4       ; Syscall número 4 (sys_write)
    mov ebx, 1       ; File descriptor 1 (stdout)
    mov ecx, buffer  ; Dirección del buffer a imprimir
    mov edx, 1       ; Número de bytes a escribir
    int 0x80         ; Llamada al sistema
    ret
```

g. Utilizando de nuevo el código de prueba original, modifica el código para que ahora utilice el modo de direccionamiento inmediato e indirecto (en programas separados) para que imprima el carácter '@'. Documenta tus resultados.

```
section .data
    msg db 'Resultado: ', 0

section .bss
    buffer resb 1  ; Reservar 1 byte para almacenar el carácter

section .text
    global _start

_start:
    ; Imprimir el mensaje "Resultado: "
    mov eax, 4
    mov ebx, 1
    mov ecx, msg
    mov edx, 11
    int 0x80            ; Llamada al sistema para escribir en salida estándar

    ; Asignar el valor ASCII de '@' directamente usando direccionamiento inmediato
    mov byte [buffer], 64  ; Mover el valor 64 (ASCII '@') directamente a buffer

    ; Imprimir el contenido de buffer (el carácter '@')
    mov eax, 4
    mov ebx, 1
    mov ecx, buffer
    mov edx, 1
    int 0x80             ; Llamada al sistema para escribir en salida estándar

    ; Salir del programa
    mov eax, 1
    xor ebx, ebx
    int 0x80
```