Balderrama Campos Pedro Carlos Grupo 932

1. Identificar a qué tipo de direccionamiento pertenece el código que se presenta a continuación.
2. Realizar un código similar al código de prueba, sin embargo utiliza cada uno de los métodos de direccionamiento vistos en clase (registro, inmediato, directo, registro indirecto, base mas índice, relativo a registro, relativo a base mas indice)
3. Cada código debe de tener sus bloques .date _start and .text.
4. Cada línea de código debe tener un comentario adicional que explique su funcionamiento.
5. Al final de cada código, explicar el funcionamiento del modo de direccionamiento utilizado.
6. Cada uno de los códigos debe ejecutarse sin errores.
7. Realizar una conclusión de al menos 4 párrafos que explique lo aprendido en el taller (modos de direccionamiento, ventas y desventajas de cada uno, por que usar uno en lugar de otro, etc.).
8. Documentar su desarrollo, resultados y conclusión en este documento.
9. Incluir referencias (si aplica).

```
 1   section .data
 2   numero1 dd 5
 3   numero2 dd 7
 4   resultado dd 0
 5
 6   section .text
 7   global _start
 8   _start:
 9
10   mov eax, [numero1]
11   add eax, [numero2]
12   mov [resultado], eax
13
14   mov eax, 1
15   xor ebx, ebx
16   int 0x80
```

Balderrama Campos Pedro Carlos Grupo 932

1.- Directo

```
section .data
    numero1 dd 5
    numero2 dd 7
    resultado dd 0

section .text
global _start
_start:

    ; Registro (registro-directo)
    mov eax, numero1  ; Mueve el valor de numero1 al registro eax
    add eax, numero2  ; Suma el valor de numero2 al registro eax
    mov resultado, eax  ; Mueve el resultado de eax a la variable resultado

    ; Inmediato (inmediato)
    mov eax, 5  ; Mueve el valor inmediato 5 al registro eax
    add eax, 7  ; Suma el valor inmediato 7 al registro eax
    mov resultado, eax  ; Mueve el resultado de eax a la variable resultado

    ; Directo (directo)
    mov eax, [numero1]  ; Mueve el valor almacenado en la dirección de memoria numero1 a eax
    add eax, [numero2]  ; Suma el valor almacenado en la dirección de memoria numero2 a eax
    mov [resultado], eax  ; Mueve el resultado de eax a la dirección de memoria resultado

    ; Registro indirecto (registro-indirecto)
    mov ebx, numero1  ; Mueve la dirección de memoria de numero1 al registro ebx
    mov eax, [ebx]  ; Mueve el valor almacenado en la dirección de memoria apuntada por ebx a eax
    add ebx, 4  ; Incrementa ebx para apuntar a la siguiente posición de memoria
    add eax, [ebx]  ; Suma el valor almacenado en la dirección de memoria apuntada por ebx a eax
    mov [resultado], eax  ; Mueve el resultado de eax a la dirección de memoria resultado

    ; Base más índice (base-mas-indice)
    mov ebx, numero1  ; Mueve la dirección de memoria de numero1 al registro ebx
    add ebx, numero2  ; Suma el valor de numero2 al registro ebx
    mov eax, [ebx]  ; Mueve el valor almacenado en la dirección de memoria apuntada por ebx a eax
    mov [resultado], eax  ; Mueve el resultado de eax a la dirección de memoria resultado

    ; Relativo a registro (registro-relativo)
    mov ebx, resultado  ; Mueve la dirección de memoria de resultado al registro ebx
    mov [ebx], 5  ; Mueve el valor inmediato 5 a la dirección de memoria apuntada por ebx

    ; Relativo a base más índice (base-mas-indice-relativo)
    mov eax, resultado  ; Mueve la dirección de memoria de resultado al registro eax
    add eax, 4  ; Incrementa eax para apuntar a la siguiente posición de memoria
    mov [eax], 7  ; Mueve el valor inmediato 7 a la dirección de memoria apuntada por eax

    ; Salida
    mov eax, 1  ; Carga el sistema de llamadas 1 (sys_write) en eax
    xor ebx, ebx  ; Limpia ebx
    int 0x80  ; Llama al sistema para terminar el programa
```

Explicación del modo de direccionamiento utilizado:

Registro (registro-directo): Se accede directamente a los valores almacenados en los registros de la CPU.
Inmediato (inmediato): Los valores están incrustados directamente en las instrucciones.
Directo (directo): Se accede directamente a los valores almacenados en una dirección de memoria específica.

Balderrama Campos Pedro Carlos Grupo 932

Registro indirecto (registro-indirecto): Se accede a los valores a través de la dirección almacenada en un registro.
Base más índice (base-mas-indice): Se calcula la dirección sumando una base y un índice.
Relativo a registro (registro-relativo): Se accede a los valores utilizando una dirección relativa a la dirección almacenada en un registro.
Relativo a base más índice (base-mas-indice-relativo): Se accede a los valores utilizando una dirección relativa a una base más un índice.
Conclusión: Durante este taller, aprendimos sobre los diferentes modos de direccionamiento en el ensamblador x86. Cada modo tiene sus propias ventajas y desventajas. El direccionamiento directo es simple y directo, pero puede ser menos flexible en comparación con otros modos. El direccionamiento relativo es útil cuando se trabaja con estructuras de datos dinámicas, ya que permite acceder a los datos relativos a una posición de memoria base. El direccionamiento por registro es rápido y eficiente, pero puede ser limitado en la cantidad de datos que se pueden manipular al mismo tiempo. En general, la elección del modo de direccionamiento depende de las necesidades específicas del programa y de la eficiencia requerida. Es importante entender los diferentes modos de direccionamiento para poder escribir código eficiente y optimizado en ensamblador.