Actividad 4

Christian Geovany Muñoz Rodríguez

Ingeniería en computación

Código: 221350605

Seminario de Solución de Problemas de Traductores de lenguajes – D09 (Lunes y Miércoles de 1 a 3)

Maestro: José Juan Meza Espinosa

Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías

22 de febrero del 2023

Código:

```
.model small

.stack 100h



.data

datos db
80,100,119,135,148,156,160,159,152,142,127,109,90,70,51,33,18,8,1,0,4,12,25,41,6
0,80,100,119,135,148,156,160,159,152,142,127,109,90,70



.code

start:

    mov ax, @data

    mov ds, ax

    mov si, offset datos

    mov al, [si]

    mov al, datos[3]

    mov al, datos+10

    mov si, offset datos

    mov bx, 6

    mov al, [si+bx]

    mov si, offset datos

    mov bx, 20

    mov al, [si+bx+3]

    mov si, offset datos
```

```
    mov cx, 4

    mov al, [si+cx-1]

    mov si, offset datos

    mov bx, 14

    mov cx, 6

    mov al, [si + bx + cx - 6]

    mov si, offset datos

    mov al, [si + 1 + 4*3]

    mov ah, 4ch

    int 21h

end
```

Desarrollo

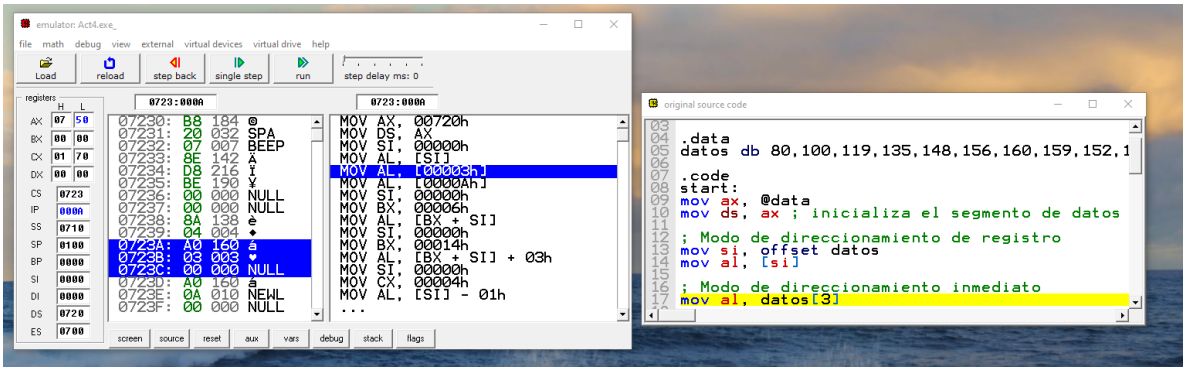
Este código muestra ejemplos de diferentes modos de direccionamiento en ensamblador x86. Primero, se inicializa el segmento de datos con la etiqueta "@data". Luego, se define una matriz de 27 bytes llamada "datos".

```
.model small
.stack 100h

.data
datos db 80,100,119,135,148,156,160,159,152,142,127,109,
```

A continuación, se muestran los siguientes modos de direccionamiento:

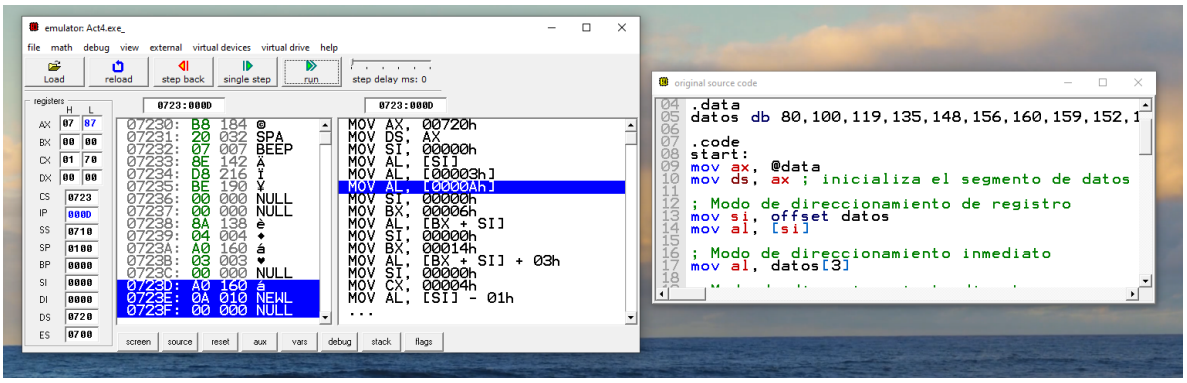
Modo de direccionamiento de registro: se carga la dirección base de la matriz en el registro SI y se carga el primer byte de la matriz en el registro AL.



```
.model small
.stack 100h

.data
datos db 80,100,119,135,148,156,160,159,152,142,127,109,
```

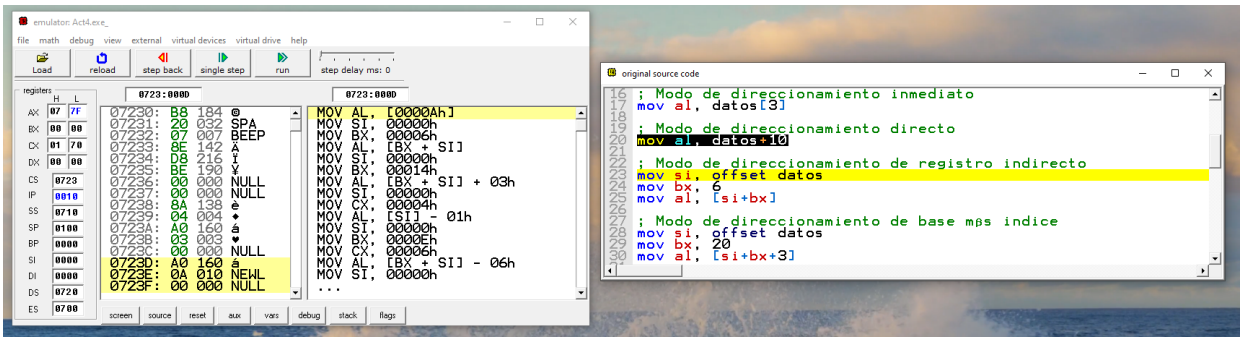
Modo de direccionamiento inmediato: se carga el byte en la posición de memoria 4 de la matriz directamente en el registro AL.




```
.model small
.stack 100h

.data
datos db 80,100,119,135,148,156,160,159,152,142,127,109,
```

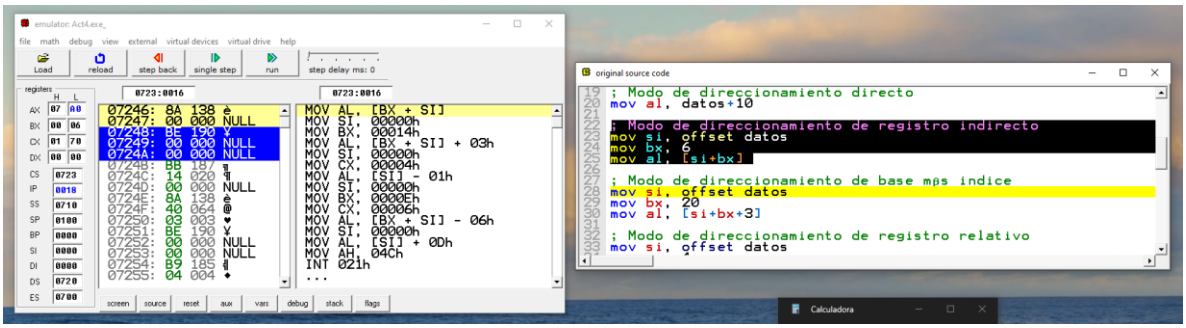
Modo de direccionamiento directo: se carga el byte en la posición de memoria 10 de la matriz sumando el valor de 10 a la dirección base de la matriz.




```
.model small
.stack 100h

.data
datos db 80,100,119,135,148,156,160,159,152,142,127,
```

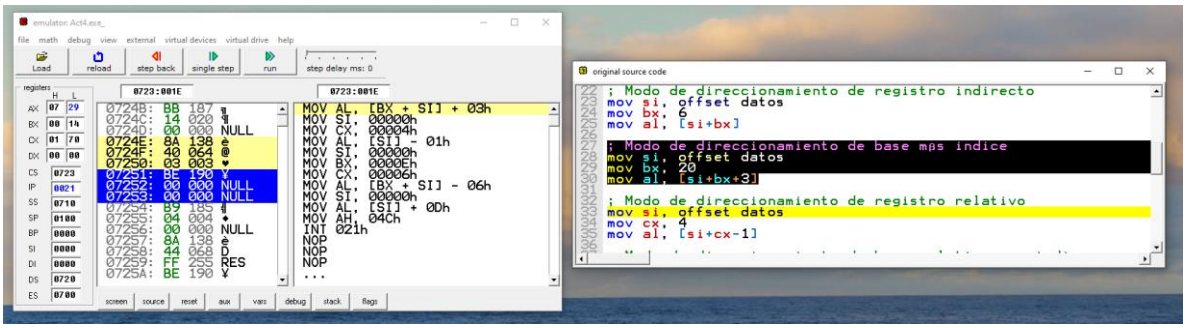
Modo de direccionamiento de registro indirecto: se carga la dirección base de la matriz en el registro SI y se carga el sexto byte de la matriz sumando el valor de 6 al registro SI.




```
.model small
.stack 100h

.data
datos db 80,100,119,135,148,156,160,
```

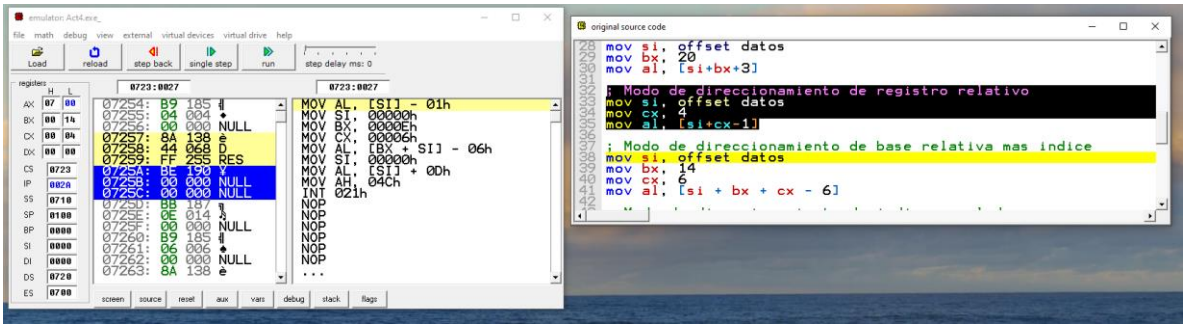
Modo de direccionamiento de base más índice: se carga la dirección base de la matriz en el registro SI y se carga el byte en la posición de memoria 20 + 3 (23) sumando el valor de 20 y el valor de 3 al registro SI.



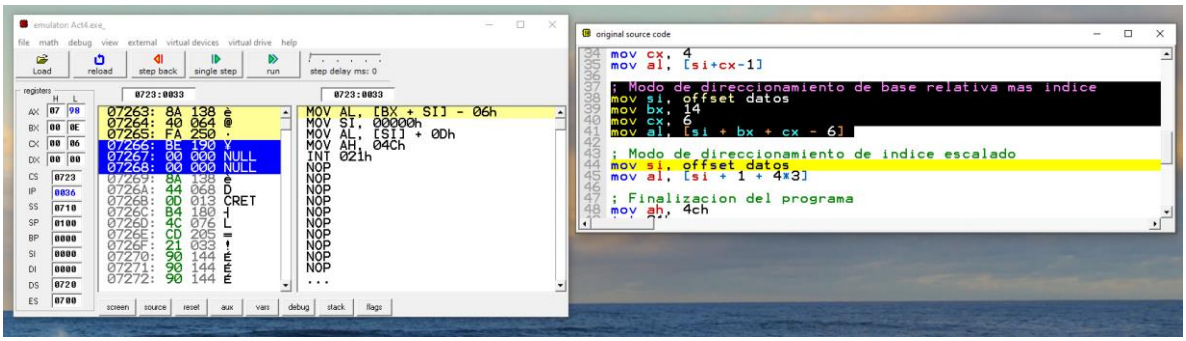

```
.model small
.stack 100h

.data
datos db 80,100,119,135,148,156,160,159,152,142,127,109,90,70,51,33,18,8,1,0,4,12,25,41,60,
```

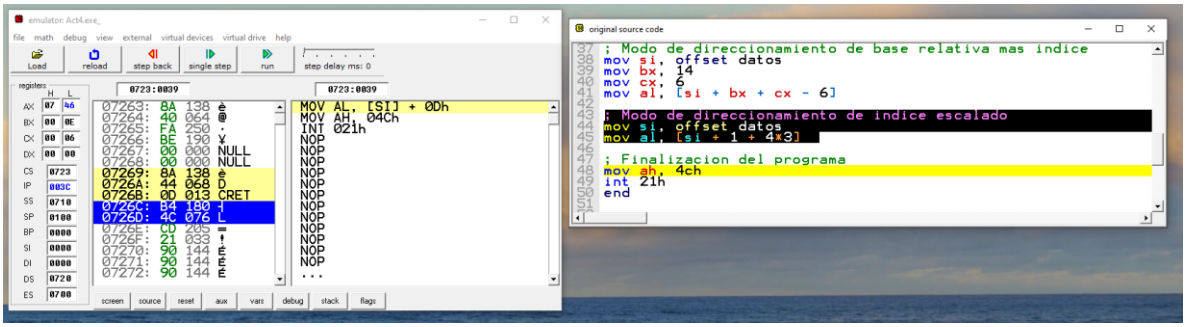
Modo de direccionamiento de registro relativo: se carga la dirección base de la matriz en el registro SI y se carga el cuarto byte de la matriz sumando el valor de 4 al registro CX y restando 1 para obtener la posición correcta de memoria.



Modo de direccionamiento de base relativa más índice: se carga la dirección base de la matriz en el registro SI y se carga el byte en la posición de memoria 14 + 6 (20) sumando el valor de 14 y el valor de 6 al registro BX y al registro CX.



Modo de direccionamiento de índice escalado: se carga el byte en la posición de memoria 1 + 4 * 3 (13) sumando el valor de 1 y el valor de 4 multiplicado por 3 al registro SI.



Conclusiones

En la programación de computadoras, el direccionamiento es una parte crucial del proceso de diseño y creación de programas. En este programa se demostró una variedad de modos de direccionamiento, cada uno con su propio propósito y aplicación.

Esta práctica me enseñó que el direccionamiento efectivo es esencial para escribir programas eficientes y funcionales. Además, aprendí que diferentes tipos de datos y variables pueden requerir diferentes modos de direccionamiento para acceder a ellos.

Por lo tanto, es importante para cualquier programador tener una comprensión sólida de los diferentes modos de direccionamiento, y cómo y cuándo utilizarlos de manera efectiva. Esto garantiza un código más legible y mantenible, y un rendimiento óptimo del programa. En resumen, esta práctica fue una excelente

oportunidad para mejorar mis habilidades de programación y comprensión del direccionamiento en ensamblador x86.

Bibliografía:

Brey, B. B. (2006). *Microprocesadores Intel : 8086/8088, 80186/80188, 80286, 80386 y 80486, Pentium, procesador Pentium Pro, Pentium II, Pentium III y Pentium 4: arquitectura, programación e interfaces*. Pearson Educación.