

Práctica 6

Modos de Direccionamiento

Luis Eduardo Galindo Amaya (1274895)

Asignatura	Organización de Computadoras (331)
Docente	Arturo Arreola Alvarez
Fecha	23-09-2022

Modos de Direcccionamiento

Luis Eduardo Galindo Amaya (1274895)

23-09-2022

Objetivo

Identificar los modos de direccionamiento adecuados para manejo de memoria en aplicaciones de sistemas basados en microprocesador mediante la distinción de su funcionamiento, de forma lógica y responsable.

Desarrollo

Actividad 1

El programa solicita al usuario el ingreso de su nombre y despliega un mensaje de saludo.

Código

```
1  ;;
2  ;;   AUTHOR: Luis Eduardo Galindo Amaya
3  ;;   DATE: 27-09-2022
4  ;; ASSEMBLE:
5  ;;   LINK:
6  ;;   RUN:
7  ;;
8
9  %include "./lib/pc_io.inc"
10
11 section .data
12     ingrese_una_cadena db 'Nombre? ', 0x0
13     saludo db 'Hola ', 0x0
14
15 section .bss
```

```
16     NOMBRE resb 255                ;para el nombre del usuario
17
18 section .text
19 global __start
20
21 __start:
22     mov edx, ingrese_una_cadena ;ingrese_una_cadena
23     call puts
24
25     mov ebx, NOMBRE                ;mover al registro de
26                                   ;direccion
27
28 capturar:
29     call getche                    ;capturar el caracter en 'al'
30     mov byte [ebx], al             ;mover el contenido a AL
31     inc ebx                        ;incrementar la posiscion de EBX
32     cmp al,10
33     jne capturar                   ;saltar cuando es diferente a 0
34     mov byte [ebx], 0
35
36     mov edx, saludo                ;mostrar "hola "
37     call puts
38
39     mov edx, NOMBRE                ;NOMBRE
40     call puts
41
42
43     ; TERMINAR PROGRAMA
44     mov eax, 1
45     mov ebx, 0
46     int 80h
```

Actividad 2

Cree un programa llamado ApellidoNombreP6.asm que contenga las instrucciones necesarias para hacer lo que se indica a continuación:

- Reservar dos espacios en memoria no inicializados, uno de 32 bytes etiquetado como A y el otro de 1 byte etiquetado como N.
- Solicitar una cadena que se almacene en A.
- Copiar el caracter en la posición 0 de A en la variable N. Use un modo de direccionamiento base.
- Reemplazar el caracter en la posición 3 de A por un asterisco '*', usando un modo de direccionamiento base con desplazamiento.

- e) Reemplazar el caracter en la posición 6 de A por un arroba '@' usando un direccionamiento base con índice escalado.
- f) Copiar el caracter en la posición 1 de A y almacenarlo en los bits 15-8 del acumulador.
- g) Reemplazar el caracter en la posición 9 de A por el caracter en los bits 15-8 del acumulador, usando un direccionamiento base con índice escalado y desplazamiento.
- h) Solicite un caracter al usuario y guárdelo en la posición 5 de A.

Código

```

1  ;;
2  ;;AUTHOR:    Luis Eduardo Galindo Amaya
3  ;;DATE:      23-09-2022
4  ;;ASSEMBLE:  nasm -f elf t1.asm
5  ;;LINK:      ld -m elf_i386 t1.o ./lib/libpc_io.a -o entrada
6  ;;RUN:       ./entrada
7  ;;
8
9  %include "lib/pc_io.inc"
10
11 section .data
12     ingrese_una_cadena db 'ingrese una cadena', 0xA, 0x0
13     ingrese_un_caracter db 'ingrese un caracter', 0xA, 0x0
14     salto_de_linea     db 0xA, 0x0
15     asterisco           db '*'
16     arroba              db '@'
17
18 section .bss
19     ;A) RESERVAR DOS ESPACIOS EN MEMORIA NO INICIALIZADOS, UNO
20     ;    DE 32 BYTES ETIQUETADO COMO A Y EL OTRO DE 1 BYTE
21     ;    ETIQUETADO COMO N.
22
23     A resb 32                ;variable de 32 bytes
24     N resb 1                ;variable de 1 byte
25
26 section .text
27 global _start
28
29 _start:
30     mov edx, ingrese_una_cadena ;ingrese_una_cadena
31     call puts
32
33     ;B) SOLICITAR UNA CADENA QUE SE ALMACENE EN A.
34
35     mov ebx, A                ;mover 'A' al registro de
36                                ;direccion

```

```
37
38 capturar:
39     call getche                ;capturar el caracter en 'al'
40     mov byte [ebx], al        ;mover el contenido a AL
41     inc ebx                    ;incrementar la posiscion de EBX
42     cmp al,10
43     jne capturar              ;saltar cuando es diferente a 0
44     mov byte [ebx], 0
45
46     mov edx, A                 ;mostrar el string
47     call puts
48
49     ;C) COPIAR EL CARACTER EN LA POSICIÓN 0 DE A EN LA VARIABLE
50     ;   N. USE UN MODO DE DIRECCIONAMIENTO BASE.
51
52     mov bl, [A]
53     mov byte[N], bl
54
55     mov edx, N                 ;mostrar el valor de 'N'
56     call puts
57
58     mov edx, salto_de_linea
59     call puts
60
61     ;D) REEMPLAZAR EL CARACTER EN LA POSICIÓN 3 DE A POR UN
62     ;   ASTERISCO, USANDO UN MODO DE DIRECCIONAMIENTO BASE
63     ;   CON DESPLAZAMIENTO.
64
65     mov eax, A
66     mov bl, [asterisco]
67     mov byte[eax+3], bl
68
69     mov edx, A                 ;mostrar A
70     call puts
71
72     ;E) REEMPLAZAR EL CARACTER EN LA POSICIÓN 6 DE A POR UN
73     ;   ARROBA @ USANDO UN DIRECCIONAMIENTO BASE CON ÍNDICE
74     ;   ESCALADO.
75
76     mov esi, 3
77     mov bl, [arroba]
78     mov byte[A+esi*2], bl
79
80     mov edx, A                 ;muestra A
81     call puts
82
83     ;F) COPIAR EL CARACTER EN LA POSICIÓN 1 DE A Y ALMACENARLO
84     ;   EN LOS BITS 15–8 DEL ACUMULADOR.
```

```
85
86     mov bl , byte [A+1]
87     mov ah , bl
88
89     ;G) REEMPLAZAR EL CARACTER EN LA POSICIÓN 9 DE A POR EL
90     ;    CARACTER EN LOS BITS 15–8 DEL ACUMULADOR, USANDO UN
91     ;    DIRECCIONAMIENTO BASE CON ÍNDICE ESCALADO Y
92     ;    DESPLAZAMIENTO .
93
94     mov esi , 2
95     mov byte [A+1+esi*4] , ah
96
97     mov edx , A                ; muestra A
98     call puts
99
100    ;H) SOLICITE UN CARACTER AL USUARIO Y GUÁRDELO EN LA
101    ;    POSICIÓN 5 DE A.
102
103    mov edx , ingrese_un_caracter
104    call puts
105
106    call getche                ; captura caracter
107    push eax
108
109    mov [N] , eax              ; eliminar el terminador
110    mov bl , [N]
111
112    mov byte [A+5] , bl
113
114    mov edx , A                ; muestra A
115    call puts
116
117    ; TERMINAR PROGRAMA
118
119    mov eax , 1
120    mov ebx , 0
121    int 80h
```

Conclusiones y comentarios

Es interesante ver que hay mas maneras de redireccionar memoria, a pesar de que se ve un poco extraño redireccionar con índice escalado esto nos permite acceder mas rápidamente a las posiciones de memoria y es importante aprender a usarlo.

Dificultades en el desarrollo

En el inciso 'h' del ejercicio 2 al insertar el valor no mostraba el resultado final insertaba un salto de línea, tuve que pasar el valor capturado al valor 'N' para eliminar el salto de línea.