

Práctica 11

Macros y Lenguaje Ensamblador de Alto Nivel

Luis Eduardo Galindo Amaya (1274895)

Asignatura	Organización de Computadoras (331)
Docente	Arturo Arreola Alvarez
Fecha	18-11-2022

Macros y Lenguaje Ensamblador de Alto Nivel

Luis Eduardo Galindo Amaya (1274895)

18-11-2022

Objetivo

Identificar el funcionamiento y beneficio de utilizar macros en un lenguaje de bajo nivel. Identificar la estructura del lenguaje ensamblador de alto nivel para el desarrollo de sistemas basados en microprocesador.

Desarrollo

Cree un archivo llamado P11_Menu.asm en donde deberá escribir un programa en lenguaje ensamblador que haga lo siguiente:

1. El programa deberá mostrar un menú principal con al menos 2 opciones y una opción para terminar el programa.
2. Cree la macro print_menu que le ayudara a la impresión de los menús.
3. Cuando el usuario seleccione una de las opciones se debe desplegar un submenú con al menos 2 opciones y una opción para regresar al menú principal.

Ejemplo:

Menú

|Operaciones Aritméticas

|Suma

|Resta

|Regresar

|Operaciones Binarias

|AND

|OR

|Regresar

|Salir

Conclusiones y comentarios

Cuando trabajamos con macros debemos ser cuidadosos, pienso que no debemos abusar de ellos, en mi opinion un macro adecuado no debe tener mas de 10 lineas de codigo y no ser abusado exesivamente.

Código

```
1  ;; AUTHOR: Luis Eduardo Galindo Amaya
2  ;; DATE: 21-10-2022
3  ;; ASSEMBLE:
4  ;; LINK:
5  ;;
6
7  %include "./lib/pc_io.inc"
8
9  ;; Imprime %1 ". " al inicio de un string
10 %macro INDEX 1
11     mov     al, ' '
12     call    putchar
13     mov     al, %1
14     call    putchar
15     mov     al, '.'
16     call    putchar
17     mov     al, ' '
18     call    putchar
19 %endmacro
20
21 ;; imprime el menu
22 %macro PRINT_MENU 4
23     mov     edx, %1
24     call    puts
25     INDEX '1'
26     mov     edx, %2
27     call    puts
28     INDEX '2'
29     mov     edx, %3
30     call    puts
31     INDEX '0'
32     mov     edx, %4
33     call    puts
34 %endmacro
35
36 section .data
37     MENU db "MENU",10,0
```

```
38     SALIR db "SALIR", 10, 0
39     REGRESAR db "REGRESAR", 10, 0
40
41     OPERACIONES_ARIMETICAS db "OPERACIONES ARIMETICAS",10,0
42     SUMA db "SUMA", 10, 0
43     RESTA db "RESTA", 10, 0
44
45     OPERACIONES_BINARIAS db "OPERACIONES BINARIAS",10,0
46     SAND db "AND", 10, 0
47     SOR db "OR", 10, 0
48
49     section .bss
50
51     section .text
52     global _start
53
54     _start:
55         PRINT_MENU MENU, OPERACIONES_ARIMETICAS, OPERACIONES_BINARIAS, SALIR
56
57         call getch                ; captura en eax
58
59         cmp eax, '0'
60         je terminar_programa
61         cmp eax, '1'
62         je menu_operaciones_arimeticas
63         cmp eax, '2'
64         je menu_operaciones_binarias
65
66         jmp _start
67
68
69     menu_operaciones_arimeticas:
70         PRINT_MENU OPERACIONES_ARIMETICAS, SUMA, RESTA, REGRESAR
71
72         call getch
73
74         cmp eax, '1'                ; case 1
75         je .ar_case_1
76         cmp eax, '2'                ; case 2
77         je .ar_case_2
78         cmp eax, '0'                ; case 0
79         je _start
80
81         jmp menu_operaciones_arimeticas ; default
82
83     .ar_case_1:
84         mov edx, SUMA
85         call puts
```

```
86     jmp menu_operaciones_arimeticas
87
88 .ar_case_2:
89     mov edx, RESTA
90     call puts
91     jmp menu_operaciones_arimeticas
92
93
94 menu_operaciones_binarias:
95     PRINT_MENU OPERACIONES_BINARIAS, SAND, SOR, REGRESAR
96
97     call getch
98
99     cmp eax, '1'                ; case 1
100    je .bi_case_1
101    cmp eax, '2'                ; case 2
102    je .bi_case_2
103    cmp eax, '0'                ; case 0
104    je _start
105
106    jmp menu_operaciones_binarias ; default
107
108 .bi_case_1:
109     mov edx, SAND
110     call puts
111     jmp menu_operaciones_binarias
112
113 .bi_case_2:
114     mov edx, SOR
115     call puts
116     jmp menu_operaciones_binarias
117
118
119
120 terminar_programa:
121     ; TERMINAR PROGRAMA
122     mov eax, 1
123     mov ebx, 0
124     int 80h
```