

Natalia Mariel Calderón Echeverría

202200007

LABORATORIO ARQUITECTURA DE COMPUTADORES Y ENSAMBLADORES I

Sección A

PRÁCTICA 3

Primer semestre 2024



Manual de Tecnico

Requerimientos básicos:

- Lenguaje programado: ASSEMBLY
- Compilador: DOSBOS
- Extensiones de VisualStudio:
 - MASM
 - MASM/TASM
 - VSCode DOSBox



```
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>MOUNT X C:\Users\natal\Documents\NATALIA\5sem\ARQUI\LAB\ACYE1_PRACTICA3_2022
00007
Drive X is mounted as local directory C:\Users\natal\Documents\NATALIA\5sem\ARQU
I\LAB\ACYE1_PRACTICA3_202200007\

Z:\>x:

X:\>
```

Funcionalidades importantes del programa

El programa tiene la funcionalidad de simular un tablero de ajedrez, en donde las piezas de un jugador están delimitadas por su inicial en mayúscula y las contrincantes por la inicial de la pieza en minúscula.

```
=====
Universidad De San Carlos De Guatemala
Facultad De Ingenieria
Escuela de ciencias y sistemas
Arquitectura de computadores y ensambladores 1
SECCION A
=====
PRIMER SEMESTRE 2024
Natalia Mariel Calderon Echeverria
----> 202200007 - PRACTICA 3
=====

=====CHESS - NATALIA MARIEL =====
1 -->(1) Nuevo Juego      1
1 -->(2) Puntajes         1
1 -->(3) Reportes         1
1 -->(4) Salir            1
--->>>> Seleccione del menu:
```

El juego muestra un simple Menú, en donde el jugador puede ir seleccionando distintas opciones de acuerdo a lo que desea realizar, es decir, puede realizar:

1. Inicio de juego
2. Puntajes
3. Reporte
4. Salir

Para la realización de este Menú se hizo uso de los macros: Print y Opción.

```
getOption MACRO regOpcion
    MOV AH, 01h
    INT 21h

    MOV regOpcion, AL
ENDM
```

```
Print MACRO registroPrint
    MOV AH, 09h
    LEA DX, registroPrint
    INT 21h
ENDM
```

Print MACRO:

Este macro se utiliza para imprimir una cadena de caracteres en la consola.

MOV AH, 09h: Esta instrucción carga el valor hexadecimal 09h, que es la función de servicio 09h de la interrupción INT 21h, en el registro AH. Esta función se utiliza para mostrar una cadena de caracteres en la pantalla.

LEA DX, registroPrint: La instrucción LEA (Load Effective Address) carga la dirección efectiva de la cadena almacenada en registroPrint en el registro DX. Esto prepara la dirección de memoria de la cadena para que pueda ser utilizada por la interrupción 21h para imprimir la cadena.

INT 21h: Esta es una interrupción de software que imprime la cadena cuya dirección está en DX en la pantalla.

getOpcion MACRO:

Este macro se utiliza para obtener un carácter de entrada desde el teclado y almacenarlo en una variable, es decir para las distintas opciones/decisiones que va tomando el usuario dentro del programa.

MOV AH, 01h: Esta instrucción carga el valor hexadecimal 01h, que es la función de servicio de la interrupción, la función se utiliza para leer un carácter desde la entrada estándar (teclado).

INT 21h: Esta interrupción de software invoca el servicio especificado en AH. En este caso, la función 01h lee un carácter desde la entrada y lo almacena en el registro AL.

MOV regOpcion, AL: Esta instrucción mueve el valor del registro AL, en donde está el valor leído, a la ubicación de memoria especificada por regOpcion.

Esto es lo que hace posible que el usuario pueda seleccionar opciones del Menú dentro del programa.

Posteriormente también se hace uso de una técnica que permite simplificar el manejo de arreglo de 2 dimensiones, convirtiéndolo en una dimensión. Esta simplificación permite el manejo y movimiento de las piezas. Al igual que el macro de impresión de tablero, que es el macro que indica los valores dentro del arreglo que se desea imprimir.

● Imprimir Board

Inicializa los registros BX y SI a cero mediante la operación XOR para iniciar el recorrido del tablero. Despliega las etiquetas de las columnas (A-H). Recorre cada fila del tablero (desde la fila 1 hasta la fila 8) y cada columna (desde la columna A hasta la columna H).

Luego, determina qué pieza de ajedrez debe ser mostrada en la posición actual del tablero utilizando la matriz de datos. Imprime la pieza de ajedrez correspondiente siguiendo las pautas de impresión definidas en las secciones SetPValueRow1, SetPValueRow2, SetPValueRow7 y SetPValueRow8.

```

19  ImprimirBoard MACRO
20      LOCAL fila, columna
21      XOR BX, BX
22      XOR SI, SI
23
24      Print columnas
25      MOV CL, 0
26
27      fila:
28          Print salto
29          MOV AH, 02h
30          MOV DL, filas[BX]
31          INT 21h
32
33          MOV DL, 32
34          INT 21h
35
36          columna:
37              MOV DL, matriz[SI]
38              CMP filas[BX], 49 ;
39              JE SetValueRowAux
40              CMP filas[BX], 50 ; fila 2
41              JE SetValueRow2Aux
42              CMP filas[BX], 55 ; 7
43              JE SetValueRow7Aux
44              CMP filas[BX], 56 ; 8
45              JE SetValueRow8Aux
46              INT 21h
47              JMP CheckEndOfRowAux
48
49      SetValueRow2Aux:
50      JMP SetValueRow2
51      SetValueRow7Aux:
52      JMP SetValueRow7
53      SetValueRow8Aux:
54      JMP SetValueRow8
55
56      CheckEndOfRowAux:
57      JMP CheckEndOfRow
58      SetValueRow1:
59      MOV DL, 't' ; Column A
60      CMP CL, 0
61      JE PrintColumn
62      MOV DL, 'c' ; Column B
63      CMP CL, 1
64      JE PrintColumn
65      MOV DL, 'a' ; Column C
66      CMP CL, 2
67      JE PrintColumn
68      MOV DL, 'r' ; Column D
69      CMP CL, 3
70      JE PrintColumn
71      MOV DL, 's' ; Column E
72      CMP CL, 4
73      JE PrintColumn
74      MOV DL, 'a' ; Column F
75      CMP CL, 5
76      JE PrintColumn
77      MOV DL, 'c' ; Column G
78      CMP CL, 6
79      JE PrintColumn
80      MOV DL, 't' ; Column H
81      CMP CL, 7
82      JE PrintColumn
83      JMP CheckNextRow
84
85      SetValueRow2:
86      MOV DL, 'p' ; valor p
87      INT 21h
88
89      SetValueRow7:
90      MOV DL, 'P' ; valor P
91      INT 21h
92      JMP CheckEndOfRow
93
94      SetValueRow8:
95      MOV DL, 'T' ; Column A
96      CMP CL, 0
97      JE PrintColumn
98      MOV DL, 'C' ; Column B
99      CMP CL, 1
100     JE PrintColumn
101     MOV DL, 'R' ; Column D
102     CMP CL, 3
103     JE PrintColumn
104     MOV DL, '*' ; Column E
105     CMP CL, 4
106     JE PrintColumn
107     MOV DL, 'A' ; Column F
108     CMP CL, 5
109     JE PrintColumn
110     MOV DL, 'C' ; Column G
111     CMP CL, 6
112     JE PrintColumn
113     MOV DL, 'T' ; Column H
114     CMP CL, 7
115     JE PrintColumn
116     JMP CheckNextRow
117
118     PrintColumn:
119         INT 21h
120         JMP CheckEndOfRow
121
122     columnaAux:
123         JMP columna
124     filaAux:
125         JMP fila
126
127     CheckNextRow:
128         MOV DL, matriz[SI]
129         INT 21h
130         CheckEndOfRow:
131         MOV DL, 124
132         INT 21h
133         INC CL
134         INC SI
135         CMP CL, 8
136         JB columnaAux
137
138         MOV CL, 0
139         INC BX
140         CMP BX, 8
141         JB filaAux
142     ENDM

```

Utiliza el carácter de separación "|" para delimitar las columnas. Incrementa el contador de columna (CL) y el índice de la matriz (SI). Si todas las columnas de la fila actual han sido impresas, avanza a la siguiente fila (**CheckNextRow**). Estos pasos se repiten hasta que todas las columnas y filas hayan sido impresas.

● RowMajorMatrix

```

RowMajorMatrix MACRO
    MOV AL, row
    MOV BL, col

    SUB AL, 49
    SUB BL, 65

    MOV BH, 8

    MUL BH
    ADD AL, BL

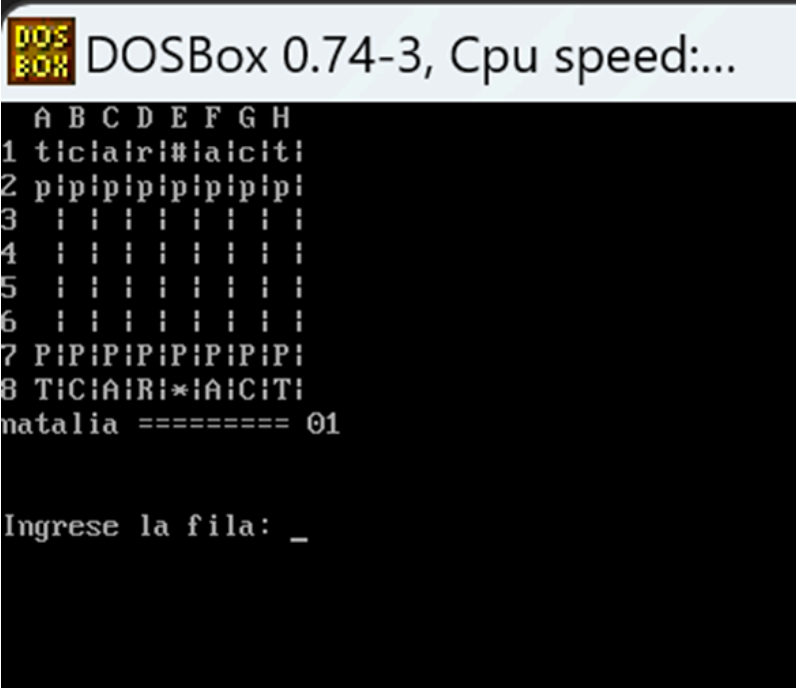
    MOV SI, AX
    MOV matriz[SI], 80
ENDM

```

El macro RowMajorMatrix toma los índices de fila (row) y columna (col) como entrada y los convierte en un índice único para acceder a la matriz.

Primero, ajusta los valores de fila y columna para que comiencen desde cero restándoles 1 y 65 respectivamente. Luego, multiplica el valor de la fila por el número total de columnas (8) y le suma el valor de la columna. Este resultado

se utiliza como un índice en la matriz, y se asigna un valor de 80 a la posición correspondiente.



DOS
BOX

DOSBox 0.74-3, Cpu speed:...

```
A B C D E F G H
1 t i c i a i r i # i a i c i t i
2 p i p i p i p i p i p i p i
3 | | | | | | | |
4 | | | | | | | |
5 | | | | | | | |
6 | | | | | | | |
7 P i P i P i P i P i P i P i
8 T i C i A i R i * i A i C i T i
natalia ===== 01

Ingrese la fila: _
```

Conclusión

Durante la elaboración de esta práctica, se aplicaron algunas de las interrupciones aprendidas en el curso de laboratorio, así como también se utilizó de manera básica el lenguaje de programación ensamblador.

La creación de un programa extenso en un solo archivo presentó ciertas dificultades relacionadas con la gestión de los saltos y su longitud. Fue fundamental recurrir a la revisión de la teoría y clases anteriores para seleccionar las etiquetas adecuadas conforme a las necesidades planteadas, lo que contribuyó a mejorar la eficiencia y legibilidad del código.