

Natalia Mariel Calderón Echeverría

202200007

LABORATORIO ARQUITECTURA DE COMPUTADORES Y ENSAMBLADORES 1

Sección A

PROYECTO 2

Primer semestre 2024



Manual de Tecnico

Requerimientos básicos:

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



Funcionalidades importantes del programa

Este programa en ensamblador está diseñado para manipular datos almacenados en un archivo CSV. Comienza solicitando al usuario el nombre del archivo CSV a abrir, luego lee los datos del archivo, los ordena y realiza diversas operaciones estadísticas sobre ellos, cómo calcular el promedio, la mediana, el máximo, el mínimo y la moda, así como contar el total de datos presentes. También puede generar una tabla de frecuencias. Para interactuar con el entorno de DOS, el programa utiliza varias interrupciones del sistema, como la interrupción 21h para operaciones de entrada/salida y la interrupción 10h para limpiar la consola.

```
CMP opcion, 97 ; Abrir el archivo(a)
JE AbrirAux
;-----
CMP opcion, 112 ; promedio(p)
JE PromedioAux
;-----
CMP opcion, 109 ; mediana (m)
JE MedianaAux
;-----
CMP opcion, 99 ; contador(c)
JE ContadorAux
;-----
CMP opcion, 111 ; moda(o)
JE ModaAux
;-----
CMP opcion, 120 ; max(x)
JE MaxAux
;-----
CMP opcion, 110 ; min(n)
JE MinAux
;-----
CMP opcion, 108 ; limpiar(l)
JE LimpiarAux
;-----
CMP opcion, 102 ; tabla frec(f)
JE TablaAux
;-----
CMP opcion, 114 ; reporte(r)
JE ReporteAux
;-----
CMP opcion, 105 ; info(i)
```

```
=====
PROYECTO 2 ::::::::::: 1S 2024
=====
N
D
Abrir Archivo (a)
promedio(p)
mediana(m)
contador(c)
moda(o)
max(x)
min(n)
TablaFrec (f)
Limpiar(l)
Reporte(r)
info(i)
salir(s)
=====
```

Además, se han definido varias macros para simplificar tareas comunes, tales como imprimir una cadena, abrir un archivo, cerrar un archivo, y leer un archivo CSV , entre otras.

LÓGICA DEL PROGRAMA

MENÚ PRINCIPAL, CARGAR ARCHIVO CSV

```
=====
aNombre del archivo con los datos a analizar CSV: Data.csv
--->> Archivo csv leído correctamente
=====
```

El flujo del programa comienza con la función `AbrirL`, donde solicita al usuario el nombre del archivo CSV a abrir. Una vez que se proporciona el nombre, el programa procede a abrir el archivo, obtener su tamaño, leer los datos del archivo CSV y luego cerrarlo. Después de obtener los datos, el programa pasa a la etapa de ordenamiento utilizando el algoritmo de ordenamiento de burbuja

```
;-----abrir-----
AbrirL:
Println pedirNomberCsv
PedirCadena filename

AbrirCSV
tamFile handlerFile
leerCSV handlerFile, numCSV
cerrarCSV handlerFile
;ir llenando
GenerarRep nombreArchivo, handlerReporte
; * los ordena en burbuja
burbuja

JMP Menu
```

La macro **BURBUJA** implementa el algoritmo de ordenamiento de burbuja, que compara elementos adyacentes y los intercambia si están en el orden incorrecto.

Este proceso se repite hasta que todos los elementos estén en su posición correcta. Se decidió ordenar los datos al momento de obtenerlos esto con el objetivo de garantizar que cualquier operación estadística realizada

posteriormente, como calcular el promedio, la mediana o encontrar valores extremos, se realice sobre datos ordenados, lo que facilita y agiliza la manipulación y análisis de los mismos.

```
burbuja MACRO
    LOCAL forA, forB, Rmovimiento, terminarforB
    XOR AX, AX
    XOR CX, CX
    ✨ XOR DX, DX
|
|
|   MOV CX, numDatos
|   DEC CX
|   MOV DL, 0
|   forA:
|       PUSH CX
|
|       MOV CX, numDatos
|       DEC CX
|       SUB CX, DX
|       MOV BX, 0
|       forB:
|           MOV AL, bufferDatos[BX]
|           MOV AH, bufferDatos[BX + 1]
|           CMP AL, AH
|           JA Rmovimiento
|           INC BX
|           LOOP forB
|           JMP terminarforB
|
|       Rmovimiento:
|           XCHG AL, AH
|           MOV bufferDatos[BX], AL
|           MOV bufferDatos[BX + 1], AH
|           INC BX
|
|       LOOP forB
|
|   terminarforB:
|       POP CX
|       INC DL
|       LOOP forA
|
|   ENDM
```

• MACRO LINKNUM

```

11
12 LinkNum MACRO valor, cadena
13     LOCAL CICLO, DIVBASE, SALIRCC, ADDZERO, ADDZERO2
14
15     CICLO:
16         MOV DX, 0
17         MOV CX, valor
18         CMP CX, rgbb
19         JB DIVBASE
20
21         MOV BX, rgbb
22         MOV AX, valor
23         DIV BX
24         MOV cadena[SI], AL
25         ADD cadena[SI], 48
26         INC SI
27
28         MUL BX
29         SUB valor, AX
30
31         CMP rgbb, 1
32         JE SALIRCC
33
34     DIVBASE:
35         CMP valor, 0
36         JE ADDZERO
37
38         MOV AX, rgbb
39         MOV BX, 10
40         DIV BX
41         MOV rgbb, AX
42         JMP CICLO
43
44     ADDZERO:
45         MOV cadena[SI], 48
46         INC SI
47
48 SALIRCC:
49 ENDM

```

La macro LINKNUM convierte un número en una cadena de caracteres. Inicializa variables locales y comienza un bucle para dividir el valor por la base 10.

Cada dígito obtenido se convierte en un carácter ASCII y se agrega a la cadena de salida. Si el valor tiene menos dígitos que la base, se añaden ceros adicionales para mantener la longitud de la cadena.

Una vez que se han procesado todos los dígitos o se ha alcanzado la longitud deseada, la macro termina y la cadena de caracteres se almacena en el destino proporcionado.

• PROMEDIO

La macro Promedio calcula el promedio de un conjunto de datos almacenados en un buffer, los datos cargados y ordenados al momento de abrir y leer un archivo.

Comienza iniciando variables y limpiando registros. Luego, utiliza un bucle para sumar todos los datos en el buffer. Después de calcular la suma, divide el resultado por el número de datos para obtener el valor entero y decimal del promedio.

Utiliza la macro *LinkNum* para convertir el valor entero y decimal en una cadena de caracteres que se almacenará en *resImprimir*. Si el valor decimal tiene menos de dos dígitos, se añaden ceros a la cadena para mantener el formato. Finalmente, se imprime el resultado del promedio.

```

280 Promedio MACRO
281     LOCAL Sumatoria, CicloDecimal, Cont
282     XOR AX, AX
283     XOR BX, BX
284     XOR CX, CX
285     XOR DX, DX
286
287     MOV CX, numDatos
288     Sumatoria:
289         MOV DL, bufferDatos[BX]
290         ADD AX, DX
291         INC BX
292         MOV DX, 0
293         LOOP Sumatoria
294
295     MOV DX, 0
296     MOV BX, numDatos
297     DIV BX
298     MOV entero, AX
299     MOV decimal, DX
300     MOV SI, 0
301
302     LinkNum entero, resImprimir
303
304     MOV resImprimir[SI], 46
305     INC SI
306
307     CMP decimal, 0
308     JNE CicloDecimal
309
310     MOV resImprimir[SI], 48
311     INC SI
312     MOV resImprimir[SI], 48
313     JMP ContinuarProm
314
315     CicloDecimal:
316         MOV AX, decimal
317         MOV BX, 10
318         MOV DX, 0
319         MUL BX
320
321         MOV BX, numDatos
322         MOV DX, 0

```

Conclusión

El programa está diseñado para presentar análisis y resultados estadísticos básicos en función con la información proporcionada. Este programa supuso la finalización del curso de arquitectura de compiladores y ensambladores 1, siendo un proyecto que involucró todo lo aprendido a lo largo de las prácticas relacionadas con el lenguaje ensamblador, así como lo es el modo video y el uso de distintas interrupciones.