

PRACTICA 2

ARQUITECTURA DE COMPUTADORES Y
ENSAMBLADORES 1

MANUAL TÉCNICO

RODRIGO ALEJANDRO HERNÁNDEZ DE LEÓN

CARNET:201900042

I. Introducción

Objetivo

Otorgar el apoyo al programador con la comprensión del programa hecho en lenguaje ensamblador con compilador MASM y DOSBox para la ejecución de pruebas para la calculadora.

II. Especificación técnica

- Computadora portátil o de escritorio.
- Sistema Operativo Windows o alguna distribución de Linux
- DOSBox
- Compilador MASM
- Visual Studio Code

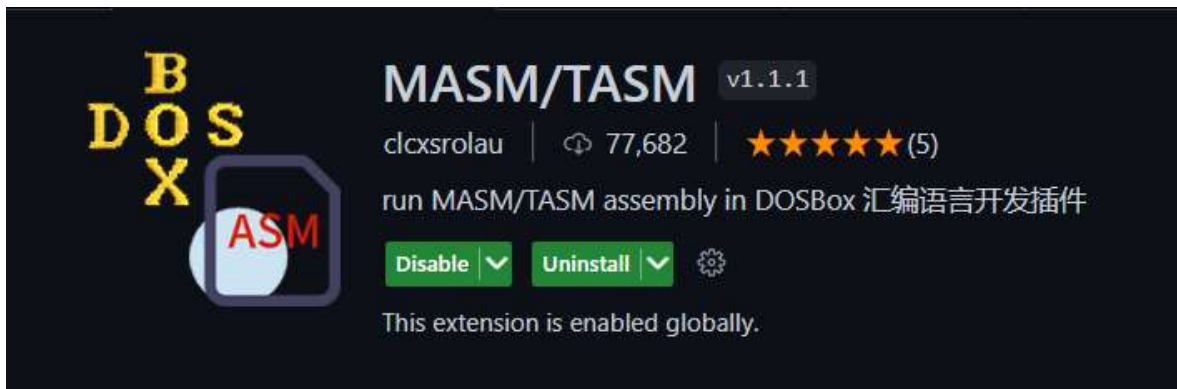
III. Lógica del Programa

En las siguientes paginas se estará explicando el código desarrollado de la calculadora:

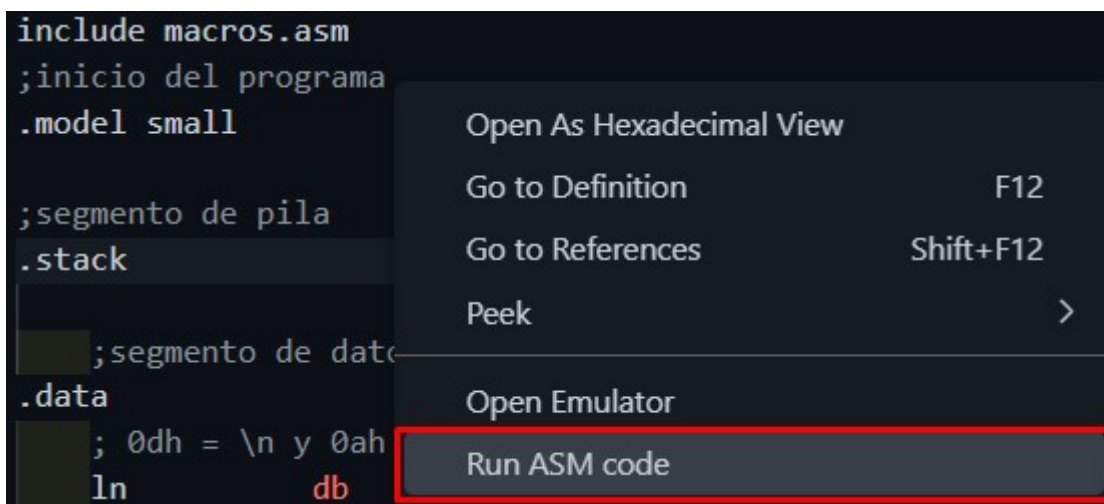
1. Ingreso a la aplicación por Visual Studio Code
2. Segmento Data
3. Segmento Code
4. Macros

1. Ingreso a la aplicación por VS Code

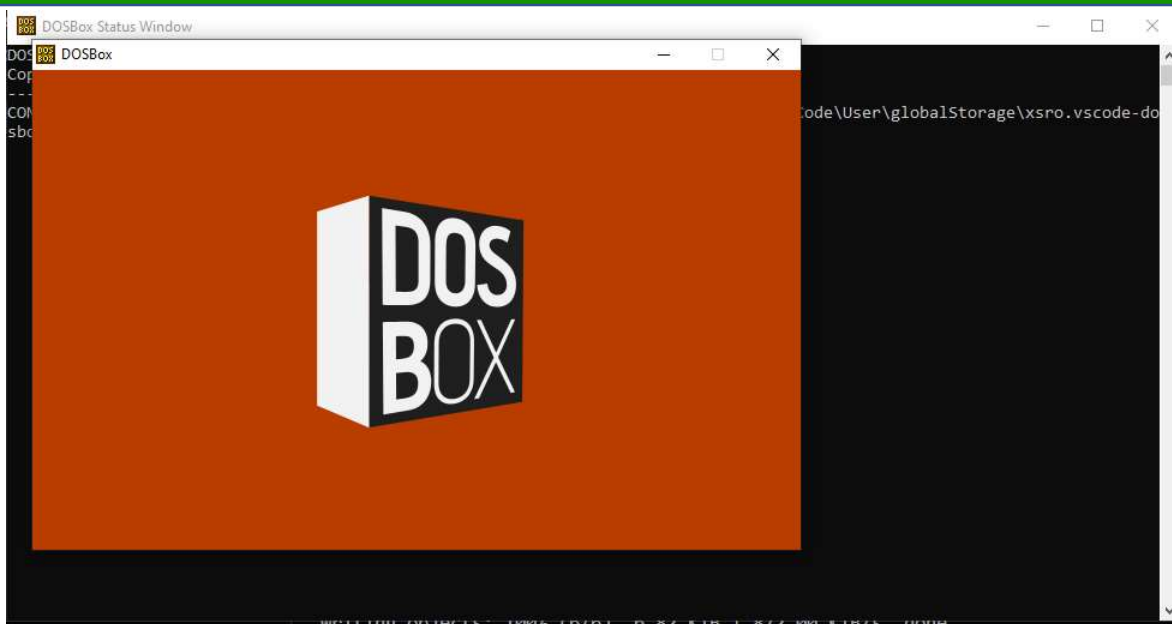
Tiene que tener instalada la siguiente extensión:



Luego hay que ubicarse en el archivo de *main.asm* y presionar click derecho, después seleccionar la opción de Run ASM:



Y posteriormente abrirá el DOSBox.



Y aparecerá el menú principal:



2. Segmento data

Se declararon las siguientes variables que se utilizaron en el programa:

```

Ln      db      0ah, '$'

;INTRO DE INICIO
inicio  db      '===== Practica 2 / Proyecto 2 =====','$'
inicio1 db      '= Universidad de San Carlos de Guatemala      ='','$'
inicio2 db      '= Facultad de Ingenieria                      ='','$'
inicio3 db      '= Escuela de Ciencias y Sistemas              ='','$'
inicio4 db      '= Arquitectura de Computadores y Ensambladores 1 ='','$'
inicio5 db      '= Seccion: N                                   ='','$'
inicio6 db      '= Vacaciones de Diciembre 2022                ='','$'
inicio7 db      '= Rodrigo Alejandro Hernandez de Leon          ='','$'
inicio8 db      '= 201900042                                     ='','$'
inicio9 db      '= Presione cualquier tecla para continuar      ='','$'
inicio10 db     '=====','$'

;MENU DE OPCIONES
inicio9 db      'Ingrese el numero de la opcion que desea, o presione ESC para salir','$'
inicio10 db     '= (1) Ingresar ecuacion','$'
inicio11 db     '= (2) Imprimir la funcion almacenada','$'
inicio12 db     '= (3) Imprimir la derivada de la funcion almacenada','$'
inicio13 db     '= (4) Imprimir la integral de la funcion almacenada','$'
inicio14 db     '= (5) Graficar la funcion original, derivada o integral','$'
inicio15 db     '= (6) Metodo de Newton','$'
inicio16 db     '= (7) Metodo de Steffensen','$'
inicio17 db     '= (8) Salir de la aplicacion','$'
op5      db      'OPCION 5','$'
op6      db      'OPCION 6','$'
op7      db      'OPCION 7','$'

;MENSAJE DE ERRORES
operror  db      'Opcion no valida, presione ENTER para continuar','$'
msger1   db      'Numero no valido, Ingrese un numero entre 0 y 5','$'
msger2   db      'No se ha ingresado ninguna funcion','$'
num      db      55
naux     db      2 dup('0'), '$'
baux     db      10

;PETICION DE LOS COEFICIENTES
ins1     db      168,'Cual es el maximo exponente de la funcion?','$'
exp5     db      'Ingrese el coeficiente del valor x^5','$'
exp4     db      'Ingrese el coeficiente del valor x^4','$'
exp3     db      'Ingrese el coeficiente del valor x^3','$'
exp2     db      'Ingrese el coeficiente del valor x^2','$'
exp1     db      'Ingrese el coeficiente del valor x^1','$'
exp0     db      'Ingrese el coeficiente del valor x^0','$'
prueba   db      'Si entra','$'
unidades db      0
decenas  db      0

;VARIABLES DE LOS COEFICIENTES DE LA FUNCION
coef5    db      0
coef4    db      0
coef3    db      0
coef2    db      0
coef1    db      0
coef0    db      0

;VARIABLES DE LOS COEFICIENTES DE LA DERIVADA
der5     db      0
der4     db      0
der3     db      0
der2     db      0
der1     db      0

;VARIABLES DE LOS COEFICIENTES DE LA INTEGRAL
int5     db      0
int4     db      0
int3     db      0
int2     db      0
int1     db      0
int0     db      0
ac       db      0

;MENSAJES PARA MOSTRAR EL RESULTADO DE LA FUNCION
msg       db      'f(x) = ', '$'
msgDerivada db      'La derivada de la funcion ingresada es: ', '$'
msgIntegral db      'La integral de la funcion ingresada es: ', '$'
x6        db      'x^6 ', '$'
x5        db      'x^5 ', '$'
x4        db      'x^4 ', '$'
x3        db      'x^3 ', '$'
x2        db      'x^2 ', '$'
x1        db      'x ', '$'
verif     db      0
constant  db      'C ', '$'
mas       db      '+ ', '$'
menos     db      '- ', '$'

textaux   label byte
maxtextaux db      4
longtextaux db      ?
numtextaux db      4 dup('0'), '$'

```

3. Segmento code

INTRO:

Este apartado se encuentra la información inicial.

A screenshot of a code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The text 'INTRO:' is at the top left of the code area. Below it, there is a list of assembly instructions. The first column contains instructions like 'cls', 'print', 'pause', and 'jmp'. The second column contains labels and values like 'inicio', 'ln', 'inicio1', 'inicio2', 'inicio3', 'inicio4', 'inicio5', 'inicio6', 'inicio7', 'inicio8', 'inicio09', 'inicio010', and 'MENU'.

```
INTRO:
      cls
      print      inicio
      print      ln
      print      inicio1
      print      ln
      print      inicio2
      print      ln
      print      inicio3
      print      ln
      print      inicio4
      print      ln
      print      inicio5
      print      ln
      print      inicio6
      print      ln
      print      inicio7
      print      ln
      print      inicio8
      print      ln
      print      inicio09
      print      ln
      print      inicio010
      print      ln
      pausa
      jmp      MENU
```

MENU:

Se encuentra para imprimir el menú principal.

MENU:

```
cls
print inicio9
print ln
print inicio10
print ln
print inicio11
print ln
print inicio12
print ln
print inicio13
print ln
print inicio14
print ln
print inicio15
print ln
print inicio16
print ln
print inicio17
print ln
pausa
cmp al, 49
je OPCION1
cmp al, 50
je OPCION2
cmp al, 51
je OPCION3
cmp al, 52
je OPCION4
cmp al, 53
je OPCION5
cmp al, 54
je OPCION6
cmp al, 55
je OPCION7
cmp al, 27
je SALIR
cmp al, 56
je SALIR
print ln
jmp OPCIONERROR
```

En este apartado se encuentra la lógica del ingreso de coeficientes de la función.

En este apartado se encuentra la lógica del ingreso de coeficientes de la función.

[illegible]

OPCION2:

En este apartado se encuentra la lógica de imprimir la función ingresada.

```
OPCION2:
    verificarFuncion coef5, coef4, coef3, coef2, coef1, coef0, verif
    cmp             verif, 0
    je              OPCIONERROR2
    jne             IMPFUNC
```

OPCION3:

En este apartado se encuentra la lógica de imprimir la derivada de la función.

```
OPCION3:
    verificarFuncion coef5, coef4, coef3, coef2, coef1, coef0, verif
    cmp             verif, 0
    je              OPCIONERROR2
    jne             IMPDER
```

OPCION4:

En este apartado se encuentra la lógica de imprimir la integral de la función.

```
OPCION3:
    verificarFuncion coef5, coef4, coef3, coef2, coef1, coef0, verif
    cmp             verif, 0
    je              OPCIONERROR2
    jne             IMPDER
```

IMPFUNC:

En este apartado se manda a llamar la impresión de la función.

```
IMPFUNC:
    cls
    printFunc      coef5, coef4, coef3, coef2, coef1, coef0
    pausa
    jmp            MENU
```

IMPDER:

En este apartado se manda a llamar la impresión de la derivada.

```
IMPDER:
        cls
        printDerivada    der5, der4, der3, der2, der1
        pausa
        jmp              MENU
```

IMPINT:

En este apartado se manda a llamar la impresión de la integral.

```
IMPINT:
        cls
        printIntegral    int5, int4, int3, int2, int1, int0
        pausa
        jmp              MENU
```

4. Macros

Print

En esta macro se encarga de imprimir cadenas.

```
print macro txt
    ; carga en memoria las variables del segmento de datos
    mov ax, @data
    mov ds, ax

    ; impresion por pantalla
    mov ah, 09h
    lea dx, txt
    int 21h
endm
```

Pausa


Esta macro se encarga de dar una pausa para poder leer funciones y el menú principal.



```
pausa macro
    mov ah, 01h
    int 21h
endm
```

Printnum

En esta macro se encarga de imprimir los números.



```
printnum macro num
    mov al, num
    AAM
    mov bx, ax
    mov ah, 02h
    mov dl, bh
    add dl, 30h
    int 21h

    mov ah, 02h
    mov dl, bl
    add dl, 30h
    int 21h
endm
```

getNumero

Esta macro se encarga de obtener los números ingresados

```

getNumero macro var
    LOCAL          n1,n2,n3,n2n,negativo,negativo1,salir
    limpiarNumero numtextaux
    mov            ah, 0ah
    lea            dx, textaux
    int            21h
    cmp            longtextaux,1
    je             n1
    cmp            longtextaux,2
    je             n2
    ; VERIFICA QUE EL NUMERO INGRESADO SEA DE 1 DIGITO (POSITIVO)
n1:
    mov            al,numtextaux
    sub            al, 30h                      ;48
    mov            var, al
    jmp            salir
    ; VERIFICA QUE EL NUMERO INGRESADO SEA DE 2 DIGITOS (POSITIVO) O 1 DIGITO (NEGATIVO)
n2:
    mov            unidades,0
    mov            decenas,0

    mov            al, numtextaux[0]
    sub            al, 30h
    mov            decenas, al

    mov            al, numtextaux[1]
    sub            al, 30h
    mov            unidades, al

    mov            al, decenas
    mov            bl, 10
    mul            bl
    add            al, unidades
    mov            var, al

    salir:
endm

```

limpiarNumero

Esta macro se encarga de limpiar la entrada de numeros.

```
limpiarNumero macro text
    LOCAL repetir
    xor     bx, bx
    mov     cx, lengthof text
repetir:
    mov     text[bx], '$'
    inc     bx
    loop    repetir
endm
```

Multiplicar

Esta macro se encarga de multiplicar 2 números positivos.

```
multiplicar macro coefi, expo, total
    mov     al, coefi
    mov     bl, expo
    mul     bl
    mov     total, al
    mov     ax, 0000h
    mov     al, bl
endm
```

Dividir

Macro para dividir 2 números

```
dividir macro coefi, expo, total
    mov     al, coefi
    mov     bl, expo
    div     bl
    mov     total, al
    mov     ax, 0000h
    mov     al, bl
endm
```

printFunc

Macro para imprimir la función ingresada.

```
printFunc macro c5, c4, c3, c2, c1, c0
    LOCAL C0F5,C0F4,C0F3,C0F2,C0F1,C0F0,PR5,PR4,PR3,PR2,PR1,PR0,salir
    print msg
    cmp c5, 0
    je C0F4
    jne C0F5
C0F5:
    cmp c5, 0
    je C0F4
    jne PR5
PR5:
    printn c5
    print x5
    print mas
C0F4:
    cmp c4, 0
    je C0F3
    jne PR4
PR4:
    printn c4
    print x4
    print mas
C0F3:
    cmp c3, 0
    je C0F2
    jne PR3
PR3:
    printn c3
    print x3
    print mas
C0F2:
    cmp c2, 0
    je C0F1
    jne PR2
PR2:
    printn c2
    print x2
    print mas
C0F1:
    cmp c1, 0
    je C0F0
    jne PR1
PR1:
    printn c1
    print x1
    print mas
C0F0:
    printn c0
salir:
    print ln
    print inicio09
    print ln
endm
```

printDerivada

Macro para imprimir la derivada.

```
printDerivada macro c4, c3, c2, c1, c0
    LOCAL C0F4,C0F3,C0F2,C0F1,C0F0,PR4,PR3,PR2,PR1,PR0,salir
    print msgDerivada
    print ln
    cmp c4, 0
    je C0F3
    jne C0F4
C0F4:
    cmp c4, 0
    je C0F3
    jne PR4
PR4:
    printn c4
    print x4
    print mas
C0F3:
    cmp c3, 0
    je C0F2
    jne PR3
PR3:
    printn c3
    print x3
    print mas
C0F2:
    cmp c2, 0
    je C0F1
    jne PR2
PR2:
    printn c2
    print x2
    print mas
C0F1:
    cmp c1, 0
    je C0F0
    jne PR1
PR1:
    printn c1
    print x1
    print mas
C0F0:
    printn c0
salir:
    print ln
    print inicio09
    print ln
endm
```

printIntegral

Este macro se encarga de imprimir la integral.

```

printIntegral macro c5, c4, c3, c2, c1, c0
    LOCAL
    COF5,COF4,COF3,COF2,COF1,COF0,PR5,PR4,PR3,PR2,PR1,PR0,salir
    print msgIntegral
    print ln
    cmp c5 , 0
    je COF4
    jne COF5
    COF5:
        cmp c5 , 0
        je COF4
        jne PR5
    PR5:
        printn c5
        print x6
        print mas
    COF4:
        cmp c4, 0
        je COF3
        jne PR4
    PR4:
        printn c4
        print x5
        print mas
    COF3:
        cmp c3, 0
        je COF2
        jne PR3
    PR3:
        printn c3
        print x4
        print mas
    COF2:
        cmp c2, 0
        je COF1
        jne PR2
    PR2:
        printn c2
        print x3
        print mas
    COF1:
        cmp c1, 0
        je COF0
        jne PR1
    PR1:
        printn c1
        print x2
        print mas
    COF0:
        printn c0
        print x1
        print mas
    salir:
        print constant
        print ln
        print inicio09
        print ln
endm

```


VerificarFuncion

Este macro verifica si se ingreso datos a la función.

```
verificarFuncion macro c5,c4,c3,c2,c1,c0, verificador
    LOCAL
    C0F5,C0F4,C0F3,C0F2,C0F1,C0F0,PR5,PR4,PR3,PR2,PR1,PR0,salir
    mov     verificador, 0
    cmp     c5 , 0
    je      C0F4
    jne     C0F5
C0F5:      cmp     c5 , 0
    je      C0F4
    jne     PR5
PR5:       mov     verificador, 1
    jmp     salir
C0F4:      cmp     c4, 0
    je      C0F3
    jne     PR4
PR4:       mov     verificador, 1
    jmp     salir
C0F3:      cmp     c3, 0
    je      C0F2
    jne     PR3
PR3:       mov     verificador, 1
    jmp     salir
C0F2:      cmp     c2, 0
    je      C0F1
    jne     PR2
PR2:       mov     verificador, 1
    jmp     salir
C0F1:      cmp     c1, 0
    je      C0F0
    jne     PR1
PR1:       mov     verificador, 1
    jmp     salir
C0F0:      cmp     c0, 0
    je      salir
    jne     PR0
PR0:       mov     verificador, 1
salir:
endm
```