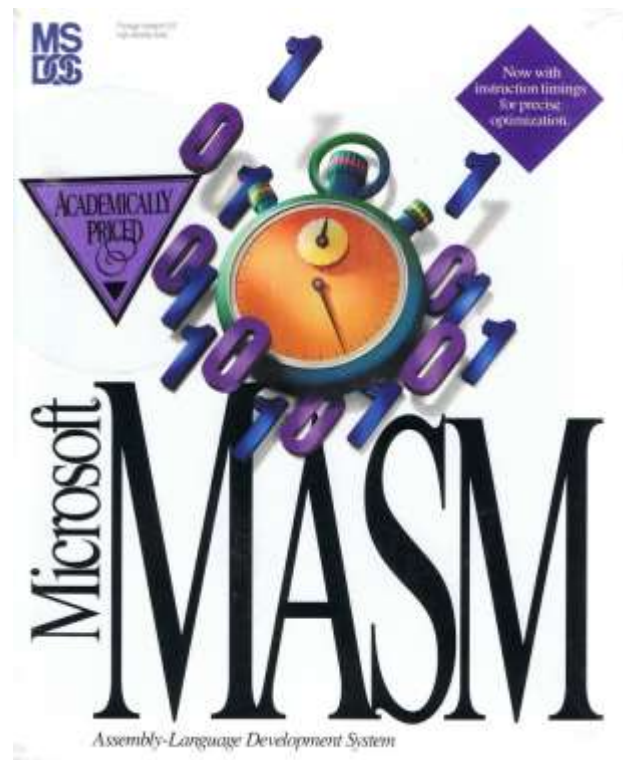


Proyecto 1

(Manual De Programador)



Contenido

Objetivos y alcances del sistema	3
Especificaciones Técnicas	3
Lógica del programa	4
Instalación de aplicación	6

Objetivos y alcances del sistema.

General:

Brindar al lector una guía que contenga la información del manejo de variables, métodos y macros para futuras actualizaciones y modificaciones realizadas por terceros.

Específicos:

- Mostrar al lector una descripción lo más completa y detallada posible del SO, IDE entre otros utilizados para el desarrollo de la aplicación.
- Proporcionar al lector una concepción y explicación técnica - formal de los procesos y relaciones entre métodos y atributos que conforman la parte operativa de la aplicación.

Especificaciones Técnicas

Sistema Operativo: Windows.

Lenguaje de Programación: MASM.

IDE: Visual Studio Code.

Lógica del programa.

Archivos utilizados:

- **main:** Es la clase principal donde se inicializa el programa.
- **macros:** Es la clase complementaria, que contienen todos los procedimientos del menú y sus funciones.

Librerías utilizadas:

No se utilizó ninguna clase de librería, todo fue con métodos nativos de MASM

Flujo del programa:

- ◆ El programa funciona de la siguiente manera:

Al iniciar el programa se ejecuta el archivo main.asm.

Llamamos el otro archivo de macros con nuestros procedimientos, `.model small` para inicializar la aplicación, llamamos el `stack` y `.data` donde iran todas nuestras variables:

```
include macros.asm; copia lo del archivo de macros
;-----
.model small; Encargado de establecer el tamaño de memoria que usaremos, en este caso small.

.stack; Proporciona al procesador el tamaño de la pila.

.data; Segmento donde estableces los espacios de memoria y las variables a utilizar.
; variables auxiliares para las multiplicaciones y divisiones
```

```
; variables auxiliares para las multiplicaciones y divisiones
numerrr dw 2, '$'
numero dw 0010h, '$'
num2 dw 2, '$'
num3 dw ?, '$', '$', '$', '$', '$', '$'
expo6 db 0006h, '$'
expo5 db 0005h, '$'
expo4 db 0004h, '$'
expo3 db 0003h, '$'
expo2 db 0002h, '$'
; Cadenas que me sirven para el menu
menuTexto db 0ah, 0dh, '-----Menu Proyecto Unico-----: ', '$'
cadena db 0ah, 0dh, 'Escoje una opcion: ', '$'
cadenaDos db 0ah, 0dh, '1) Ingresa tu funcion ', '$'
cadenaTres db 0ah, 0dh, '2) Imprimir funcion almacenada ', '$'
cadenaCuatro db 0ah, 0dh, '3) Imprimir la derivada de la funcion almacenada ', '$'
cadenaCinco db 0ah, 0dh, '4) Imprimir la integral de la funcion almacenada ', '$'
cadenaSeis db 0ah, 0dh, '5) Graficar ', '$'
cadenaSiete db 0ah, 0dh, '6) Metodo de Newton ', '$'
cadenaOcho db 0ah, 0dh, '7) Metodo de Steffensen ', '$'
cadenaNueve db 0ah, 0dh, '8) Salir de la aplicacion ', '$'
cadenaExtra db 0ah, 0dh, 'Introduce tu numero deseado ', '$'

cadenaCoCinco db 0ah, 0dh, 'Introduce tu coeficiente para el exponente 5: ', '$'
cadenaCoCuatro db 0ah, 0dh, 'Introduce tu coeficiente para el exponente 4: ', '$'
cadenaCoTres db 0ah, 0dh, 'Introduce tu coeficiente para el exponente 3: ', '$'
cadenaCoDos db 0ah, 0dh, 'Introduce tu coeficiente para el exponente 2: ', '$'
cadenaCoUno db 0ah, 0dh, 'Introduce tu coeficiente para el exponente 1: ', '$'
cadenaCoCero db 0ah, 0dh, 'Introduce tu coeficiente para el termino constante: ', '$'

backMenu db 0ah, 0dh, 'Presiona cualquier tecla para volver al Menu ', '$'
despedida db 0ah, 0dh, 'Gracias por usar mi program!! ', '$'
aceptado db 0ah, 0dh, 'Tu ecuacion fue almacenada ', 10, 13, '$'
respuesta db 0ah, 0dh, 'Esta es tu ecuacion almacenada: ', 10, 13, '$'
respuesta2 db 0ah, 0dh, 'La derivada de tu ecuacion es: ', 10, 13, '$'
respuesta3 db 0ah, 0dh, 'La integral de tu ecuacion es: ', 10, 13, '$'
```

En la sección de code llamamos a nuestra macro de menú que nos mantiene en un loop el menú con sus opciones, si elige alguna que no está en las opciones predefinidas te volverá a dejar elegir:

```
.code;Segmento de todo el código o bien el conocido main.  
main PROC;Inicio de mi clase main.  
    mov dx, @DATA  
    mov ds, dx  
    menu;Mi método menu principal  
main ENDP;Fin de mi clase main.
```

```
menu MACRO;macro que mantiene el menú y lo repite en loop.  
menuPrincipal;;imprimo todas las opciones del menú  
    textos  
    leerchar;leo el carácter que introducen  
    cmp al, 31h;compruebo si el carácter es igual a 31h que es el código hexadecimal del número 1 en ASCII  
    je opcionUno;uso el jmp equals para ver si es igual y si sí, pues entro en la etiqueta  
    cmp al, 32h  
    je opcionDos  
    cmp al, 33h  
    je opcionTres  
    cmp al, 34h  
    je opcionCuatro  
    cmp al, 35h  
    je opcionCinco  
    cmp al, 36h  
    je opcionSeis  
    cmp al, 37h  
    je opcionSiete  
    cmp al, 38h;si eligen 8, salgo del programa  
    je salir  
    jmp menuPrincipal;vuelvo a poner el menú principal si ponen algún carácter que no es los que considere  
opcionUno;;etiquetas de las opciones, me manda a su macro especial  
    trabajoUno  
    print backMenu  
    leerchar;al terminar el trabajo de la macro debo dar alguna tecla para volver al menú  
    limpiar  
    jmp menuPrincipal
```

Este trabaja a base de comparaciones y te manda a su bandera específica para realizar su macro individual, ya sea ingresar los datos, imprimir la ecuación guardada, imprimir la derivada de la función o imprimir la integral.

```

trabajoUno MACRO;macro de ingresar ecuacion
    limpiar
    print cadenaCoCinco
    leerTexto coeCinco;almacena cada coeficiente en su variable

    limpiar
    print cadenaCoCuatro
    leerTexto coeCuatro

    limpiar
    print cadenaCoTres
    leerTexto coeTres

    limpiar
    print cadenaCoDos
    leerTexto coeDos

    limpiar
    print cadenaCoUno
    leerTexto coeUno

    limpiar
    print cadenaCoCero
    leerTexto coeCero

    limpiar
    print aceptado
ENDM

```

```

trabajoDos MACRO;Esta macro imprime la ecuacion almacenada, solo si son una variable, si son cero no los imprime
    limpiar
    comparaCero coeCinco, coeCuatro, coeTres, coeDos, coeUno, coeCero
ENDM
;
trabajoTres MACRO;Esta macro imprime la derivada de la ecuacion almacenada, solo si son una variable, si son cero no los imprime
    limpiar
    comparaUno coeCinco, coeCuatro, coeTres, coeDos, coeUno, coeCero
ENDM
;
trabajoCuatro MACRO;Esta macro imprime la integral de la ecuacion almacenada, solo si son una variable, si son cero no los imprime
    limpiar
    comparaDos coeCinco, coeCuatro, coeTres, coeDos, coeUno, coeCero
ENDM
;

```

La parte de la graficación se maneja de la siguiente manera:

```
onVideo MACRO
    xor ax,ax;limpiamos los registros
    xor bx,bx
    xor cx,cx
    xor dx,dx
    mov ah,0;para activar el modo video
    mov al,13h;que sea pantalla negra
    int 10h; activamos el modo video
    mov ax,0A000h;para acceder a los registros extra
    mov es,ax
ENDM
```

Activo el modo video para poder pintar los puntos sin problema.

```
ejes MACRO
    LOCAL linea, linea2
    xor dx,dx
    xor cx,cx
    xor ax,ax
    xor bx,bx
    mov ah,0ch
    mov al,0fh
    mov cx, 160
    mov dx,0
    int 10h
    linea:
        inc dx
        int 10h
        cmp dx,200
        jne linea
    mov cx, 0
    mov dx,100
    int 10h
    linea2:
        inc cx
        int 10h
        cmp cx,320
        jne linea2
ENDM
```

Construyó los ejes de las graficas con ciclos, primero declaramos el ah con el tamaño que usaremos de modo video que es 320*200, luego le seteamos el color que es negro 0fh de manera hexadecimal.

Empezamos a construir las líneas de arriba abajo colocándonos a la mitad con el 160.

Usamos el ciclo línea que pintara puntos hasta llegar a 200. Luego nos movemos a la izquierda al centro para pintar de izquierda a derecha hasta llegar a 320. Y así pintaremos nuestro eje.

Luego entramos al método que pintara los puntos de la función, que en resumen multiplica los coeficientes por valores ya quemados de 0 a 4, luego de multiplicar cada uno los va sumando y acumulando

```
unoPositivo;;multiplico el numero por
multiplicacion2 numeroUno, expo0
mov si, word ptr[numerrrou]
mov ax, word ptr[si]
add ax, bx
mov word ptr[si], ax
jmp salirUno
```


luego pinta el punto en el valor quemado coordenada x y coordenada y la suma de las multiplicaciones.

```
mov si, word ptr[numedos]
mov ax, word ptr[si]
mov cx, ax;muevo mi coordenada x

xor ax, ax

mov si, word ptr[numerrrou]
mov ax, word ptr[si]
mov dx, ax;muevo mi coordenada y

xor ax, ax

mov ah, 0ch;pongo el modeo video
mov al, 04h;pongo el color rojo para el punto
int 10h;pinto el punto
```

Incremento los contadores que uso y si es diferente de mis 4 puntos continuo:

```
inc numedos
inc expo0
cmp expo0,0004h
jne banderin
```

Cierro el modo video:

Moviendo los registros del segmento extra de nuevo

```
offVideo MACRO
    mov ax, 0003h
    int 10h
ENDM
```

Con esta macro mantengo en ciclo el modo video:

```
trabajoCinco MACRO
    LOCAL ciclo, SALIR
    onVideo;prendo modo video
    ejes;pinto mis ejes
    impFuncion coeCinco, coeCuatro, coeTres, coeDos, coeUno, coeCero;pinto mis puntos
    ciclo;;lo mantengo en ciclo hasta que precione esc para salir
        MOV AH, 10H
        INT 16h
        CMP AL,27
        JE SALIR
        jmp ciclo
    SALIR:
    offVideo;apago el modo video
ENDM
:
```

Repositorio del proyecto.

Todo el código se encuentra comentado.

https://github.com/stvnrrr/PROYECTO_UNICO_ARQUI1_DIC-2022.git