

CALCULADORA ASSEMBLER – PROYECTO 2

MANUAL TÉCNICO

**PROGRAMA REALIZADO POR EL ALUMNO ALLEN
GIANKARLO ROMÁN VÁSQUEZ CARNET 202004745 PARA
EL CURSO ARQUITECTURA DE COMPUTADORAS Y
ENSAMBLADORES 1 DE LA ESCUELA DE CIENCIAS Y
SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA
UNIVERSIDAD SAN CARLOS DE GUATEMALA**

Programa Utilizado

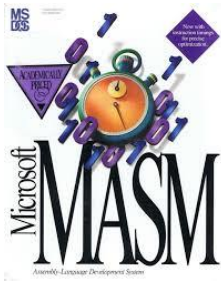


Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.



DOSBox es un emulador que recrea un entorno similar al sistema MS-DOS con el objetivo de poder ejecutar programas y videojuegos originalmente escritos para dicho sistema en computadoras más modernas o en diferentes arquitecturas.

Lenguaje Utilizado

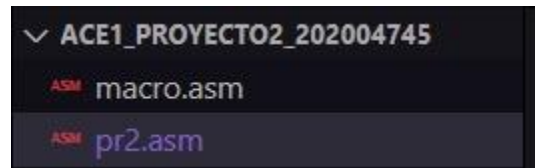


MASM es un ensamblador para la familia x86 de microprocesadores. Fue producido originalmente por Microsoft para el trabajo de desarrollo en su sistema operativo MS-DOS, y fue durante cierto tiempo el ensamblador más popular disponible para ese sistema operativo. El MASM soportó una amplia variedad de facilidades para macros y programación estructurada.

Código

Clases

El programa cuenta con dos archivos de tipo .asm que uno es el main en el que se encuentra principalmente la funcionalidad del menú, la declaración de variables a utilizar, importar las macros utilizadas en el programa.



Pr2.asm

Al iniciar el pr2.asm la data tiene los principales arreglo de char que son utilizados y las variables que son utilizadas para ayudarse con coeficientes de cada función.

```
.data
;encabezado programa
encabezado db 10,13,7,"Arquitectura de Computadoras y Ensambladores 1",10,"Proyecto 2",10,"Nombre: Allen Roman",10,"Carnet: 202004745",
;mensajes del menu
msg1 db 10,13,7,"--Menu Calculadora Grafica--","$"
msg2 db 10,13,7,"1. Ingresar Ecuacion (Funcion).","$"
msg3 db 10,13,7,"2. Imprimir la funcion almacenada.","$"
msg4 db 10,13,7,"3. Imprimir la derivda.","$"
msg5 db 10,13,7,"4. Imprimir la integral.","$"
msg6 db 10,13,7,"5. Graficar la funcion original, derivada o integral.","$"
msg7 db 10,13,7,"6. Encontrar los ceros de la funcion por medio del metodo de Newton.","$"
msg8 db 10,13,7,"7. Encontrar los ceros de la funcion por medio del metodo de Steffensen.","$"
msg9 db 10,13,7,"8. Salir","$"
salto db 10,13,"$"
error db 10,13,7,"Error: Opcion no valida, intente de nuevo","$"
errorEntero db 10,13,7,"Error: Ingrese un valor entero, intente de nuevo","$"
memoriaVacía db 10,13,7,"Error: No hay ninguna funcion almacenada, intente de nuevo","$"

;titulos de los submenus
menuIngresar db 10,13,7,"--Menu Ingresar Ecuacion--","$"
menuImprimirFuncion db 10,13,7,"--Menu Imprimir Funcion--","$"
menuImprimirDerivada db 10,13,7,"--Menu Imprimir Derivada--","$"
menuImprimirIntegral db 10,13,7,"--Menu Imprimir Integral--","$"
menuGraficar db 10,13,7,"--Menu Graficar--","$"
menuCerosNewton db 10,13,7,"--Menu Ceros Newton--","$"
menuCerosSteffensen db 10,13,7,"--Menu Ceros Steffensen--","$"
adios db 10,13,7,"Adios, tenga un buen dia.","$"

;ingresarFuncion
stringQuinta db 10,13,7,"Ingresa el coeficiente de grado 5 (x^5): ","$"
stringCuarta db 10,13,7,"Ingresa el coeficiente de grado 4 (x^4): ","$"
stringTercera db 10,13,7,"Ingresa el coeficiente de grado 3 (x^3): ","$"
stringSegunda db 10,13,7,"Ingresa el coeficiente de grado 2 (x^2): ","$"
stringPrimera db 10,13,7,"Ingresa el coeficiente de grado 1 (x^1): ","$"
stringConstante db 10,13,7,"Ingresa el coeficiente de grado 0 (Constante): ","$"
```

Funciones en pr2.asm

Las funciones que tiene principalmente el pr2.asm que tiene el menú y su funcionalidad que dirige a cada una de las otras funciones que ejecutan la funcionalidad seleccionada en el menú.

```
.code

; procedimiento principal main
main PROC

    ; carga en memoria las variables del segmento de datos
    MOV ax, @data
    MOV ds, ax
    xor ax, ax

    limpiarPantalla ;limpia la pantalla
    imprimir encabezado ;imprime el encabezado del programa

    menuPrincipal: ;etiqueta menu principal...
    ingresarEcuacion:...
    imprimirFuncion:...
    imprimirDerivada:...
    imprimirIntegral:...
    graficar:...
    cerosNewton:...
    cerosSteffensen:...
    salir:...
    errorEntradaFuncion:...

main ENDP
```

Análisis léxico

El menú principal compara y ejecuta según la respuesta de la bandera.

```
menuPrincipal: ;etiqueta menu principal
    ;imprime el menu del programa
    imprimir msg1
    imprimir msg2
    imprimir msg3
    imprimir msg4
    imprimir msg5
    imprimir msg6
    imprimir msg7
    imprimir msg8
    imprimir msg9
    imprimir salto
    obtenerOpcion ;obtiene la opcion del menu
    cmp al,'1' ;compara la opcion con 1
    je ingresarEcuacion ;si es 1, ingresa a la funcion ingresarEcuacion
    cmp al,'2' ;compara la opcion con 2
    je imprimirFuncion ;si es 2, ingresa a la funcion imprimirFuncion
    cmp al,'3' ;compara la opcion con 3
    je imprimirDerivada ;si es 3, ingresa a la funcion imprimirDerivada
    cmp al,'4' ;compara la opcion con 4
    je imprimirIntegral ;si es 4, ingresa a la funcion imprimirIntegral
    cmp al,'5' ;compara la opcion con 5
    je graficar ;si es 5, ingresa a la funcion graficar
    cmp al,'6' ;compara la opcion con 6
    je cerosNewton ;si es 6, ingresa a la funcion cerosNewton
    cmp al,'7' ;compara la opcion con 7
    je cerosSteffensen ;si es 7, ingresa a la funcion cerosSteffensen
    cmp al,'8' ;compara la opcion con 8
    je salir ;si es 8, ingresa a la funcion salir
    ;limpiarPantalla ;limpia la pantalla
    imprimir error ;si no es ninguna de las anteriores, imprime error
    imprimir salto ;imprime salto de linea
    jmp menuPrincipal ;si no es ninguna de las anteriores, vuelve al menu principal
```

Análisis Sintáctico

En el server en la carpeta src/analizador se encuentra el archivo de json que contiene la gramática libre de contexto que a su vez genera el AST con todos los terminales, no terminales, producciones y símbolo inicial.

```
instrucciones : instrucciones instruccion { $$ = $1; $$.$push($2); }
              | instruccion              { $$ = new Array(); $$.$push($1); }
              ;

instruccion : declaracion { $$ = $1; }
            | startwith   { $$ = $1; }
            | writeline   { $$ = $1; }
            | asignacion  { $$ = $1; }
            | sent_if     { $$ = $1; }
            | sent_while  { $$ = $1; }
            | sent_Dowhile { $$ = $1; }
            | BREAK PYC   { $$ = new detener.default(); }
            | sent_switch { $$ = $1; }
            | sent_for    { $$ = $1; }
            | ID DECRE PYC { $$ = new asignacion.default($1, new aritmetica.default(new identificador.default($1, @1.first_line, @1.last_column), new aritmetica.default($1, @1.first_line, @1.last_column)); }
            | ID INCRE PYC { $$ = new asignacion.default($1, new aritmetica.default(new identificador.default($1, @1.first_line, @1.last_column), new aritmetica.default($1, @1.first_line, @1.last_column)); }
            | CONTINUE PYC { $$ = new continuar.default(); }
            | funciones    { $$ = $1; }
            | llamada PYC  { $$ = $1; }
            | RETURN PYC   { $$ = new retorno.default(null); }
            | RETURN e PYC { $$ = new retorno.default($2); }
            | error        { console.log("Error Sintactico: " + yytext
                                      + " linea: " + this._$.first_line
                                      + " columna: " + this._$.first_column);

                          new errores.default("Sintactico", "No se esperaba el caracter "+ yytext ,
                                      this._$.first_line ,this._$.first_column);

            }
            ;
```

Ingresar Ecuación

Obtiene los coeficientes de cada parte de la función.

```
ingresarEcuacion:
    ;ingresa a la funcion ingresarEcuacion
    limpiarPantalla ;limpia la pantalla
    imprimir menuIngresar
    imprimir salto
    ingresoFuncionOriginal ;obtiene los coeficientes de la ecuacion
    jmp menuPrincipal ;vuelve al menu principal
```

Imprimir Ecuación

Hace una validación para ver si en memoria hay una función e imprime la función sino manda un mensaje de error, también el mismo mecanismo para la derivada y la integral.

```
imprimirFuncion:
    ;ingresa a la funcion imprimirFuncion
    limpiarPantalla ;limpia la pantalla
    imprimir menuImprimirFuncion
    imprimir salto
    cmp coef5 [1], '$' ;compara el primer digito del coeficiente 5 con 0
    je FF4 ;si es 0, ingresa a la funcion mmVacía
    jmp mmllena; si no es 0, ingresa a la funcion mmllena
FF4:
    cmp coef4 [1], '$' ;compara el primer digito del coeficiente 4 con 0
    je FF3 ;si es 0, ingresa a la funcion mmVacía
    jmp mmllena; si no es 0, ingresa a la funcion mmllena
FF3:
    cmp coef3 [1], '$' ;compara el primer digito del coeficiente 3 con 0
    je FF2 ;si es 0, ingresa a la funcion mmVacía
    jmp mmllena; si no es 0, ingresa a la funcion mmllena
FF2:
    cmp coef2 [1], '$' ;compara el primer digito del coeficiente 2 con 0
    je FF1 ;si es 0, ingresa a la funcion mmVacía
    jmp mmllena; si no es 0, ingresa a la funcion mmllena
FF1:
    cmp coef1 [1], '$' ;compara el primer digito del coeficiente 1 con 0
    je mmVacía ;si es 0, ingresa a la funcion mmVacía
    jmp mmllena; si no es 0, ingresa a la funcion mmllena

mmllena:
    funcionMemoria ;imprime la funcion en memoria
    jmp menuPrincipal ;vuelve al menu principal
mmVacía:
    imprimir memoriaVacía ;si es 0, imprime memoria vacía
    imprimir salto
    jmp menuPrincipal ;vuelve al menu principal
```

Macro.asm

Contiene las macros utilizadas para poder ejecutar el programa.

```

> imprimir macro cadena;macro para imprimir una cadena de caracteres...
> obtenerOpcion macro;macro para obtener una opcion del teclado...
> limpiarPantalla macro;macro para limpiar la pantalla...
> leer macro array;macro para leer una cadena de caracteres...
> ingresoFuncionOriginal macro...
> funcionMemoria macro...
> funcionDerivada macro...
> funcionIntegral macro...

> validarEntrada macro coeficiente...
> validarLengthCoeficiente macro coeficiente...
> validarCoefR macro coeficiente;macro para validar la entrada de un coeficiente...
> obtenerLengthCoeficiente macro coeficiente...

> comprobarNumero macro char ;macro para comprobar si un caracter es un numero...
> limpiarVariables macro array,lengthB,caracter...
> convertirCoeficiente macro coeficiente...

> getEntero macro coeficiente...
> getSigno macro signo...
```

ingresoFuncionOriginal

Utiliza la macro para imprimir la solicitud, leer el coeficiente, valida la entrada y lo pasa a entero.

```
ingresoFuncionOriginal macro
    imprimir stringQuinta; Imprime el string del coeficiente a la quinta
    leer coef5; Lee el coeficiente a la quinta
    validarEntrada coef5; Comprueba que el coeficiente sea un numero valido
    convertirCoeficiente coef5; Convierte el coeficiente a un numero entero

    imprimir stringCuarta; Imprime el string del coeficiente a la cuarta
    leer coef4; Lee el coeficiente a la cuarta
    validarEntrada coef4; Comprueba que el coeficiente sea un numero valido
    convertirCoeficiente coef4; Convierte el coeficiente a un numero entero

    imprimir stringTercera; Imprime el string del coeficiente a la tercera
    leer coef3; Lee el coeficiente a la tercera
    validarEntrada coef3; Comprueba que el coeficiente sea un numero valido
    convertirCoeficiente coef3; Convierte el coeficiente a un numero entero

    imprimir stringSegunda; Imprime el string del coeficiente a la segunda
    leer coef2; Lee el coeficiente a la segunda
    validarEntrada coef2; Comprueba que el coeficiente sea un numero valido
    convertirCoeficiente coef2; Convierte el coeficiente a un numero entero

    imprimir stringPrimera; Imprime el string del coeficiente a la primera
    leer coef1; Lee el coeficiente a la primera
    validarEntrada coef1; Comprueba que el coeficiente sea un numero valido
    convertirCoeficiente coef1; Convierte el coeficiente a un numero entero

    imprimir stringConstante; Imprime el string del coeficiente constante
    leer coef0; Lee el coeficiente constante
    validarEntrada coef0; Comprueba que el coeficiente sea un numero valido
    convertirCoeficiente coef0; Convierte el coeficiente a un numero entero
endm
```


ingresoFuncionOriginal

La funcionMemoria valida cada coeficiente para imprimirlo y así con la derivada e integral.

```
funcionMemoria macro
    LOCAL F5,F4,F3,F2,F1,F0,F
    imprimir funcionOriginalString
    cmp coef5 [1],0
    je F4
    getEntero coef5
    imprimir x5
    jmp F4
F4:
    cmp coef4 [1],0
    je F3
    getEntero coef4
    imprimir x4
F3:
    cmp coef3 [1],0
    je F2
    getEntero coef3
    imprimir x3
F2:
    cmp coef2 [1],0
    je F1
    getEntero coef2
    imprimir x2
F1:
    cmp coef1 [1],0
    je F0
    getEntero coef1
    imprimir x
F0:
    cmp coef0 [1],0
    je F
    getEntero coef0
    imprimir salto
F:
endm
```

Repositorio: https://github.com/Allenrovas/ACE1_Proyecto2_202004745