

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Arquitectura de Computadores y Ensambladores 1

Manual Técnico

Xhunik Nikol Miguel Mutzutz

201900462

# Manual Técnico

## Menú

El programa comienza con la ejecución de la etiqueta main.

La etiqueta menú es el punto de entrada del menú del programa. Se imprime el mensaje del menú y se espera a que el usuario ingrese una opción.

Si el usuario presiona la tecla 'ESC', el programa salta a la etiqueta opt5, que es la opción de salida.

Si el usuario ingresa cualquier otra tecla, se comparan con las opciones válidas ('1', '2', '3', '4', '5').

Si la tecla ingresada coincide con alguna de estas opciones, se salta a la etiqueta correspondiente.

Si la tecla ingresada no es válida, se vuelve a la etiqueta menú y se vuelve a imprimir el menú.

## Ingreso de datos

Si el usuario elige la opción 1 (etiqueta opt1), se piden al usuario que ingrese cada uno de los coeficientes de la ecuación. Esto se hace a través de la macro saveCoef, que lee un carácter del usuario y lo convierte a un número usando la macro convertAsciiToNum. Luego, se guarda el número en el registro especificado como argumento de la macro. Una vez que se han leído y guardado todos los coeficientes, se establece la bandera flagEcuacion en 1 para indicar que se ha ingresado una ecuación.

## Imprimir función

Si el usuario elige la opción 2 (etiqueta opt2), se imprimen cada uno de los coeficientes de la ecuación actual. Esto se hace a través de la macro printCoef, que convierte el coeficiente especificado como argumento a una cadena de caracteres y la imprime en pantalla.

## Calcular derivada

Si el usuario elige la opción 3 (etiqueta opt3), se calculan y se imprimen los coeficientes de la derivada de la ecuación actual. Esto se hace a través de la función calculateDiffCoefs, que calcula el coeficiente de cada término de la derivada y lo guarda en las variables correspondientes (coefADiff, coefBDiff, etc.). Luego, se imprimen estos coeficientes usando la macro printCoefDiffWord, que convierte el coeficiente especificado como argumento a una cadena de caracteres y la imprime en pantalla.

## Calcular Integral

Si el usuario elige la opción 4 (etiqueta opt4), se calculan y se imprimen los coeficientes de la integral de la ecuación actual a través de la función calculateIntegCoefs, que utiliza la función calculateIntegral para calcular el valor de cada término de la integral y lo guarda en las variables correspondientes (coefAInteg, coefBInteg, etc.). Luego, se imprimen estos coeficientes usando la macro printCoef, que convierte el coeficiente especificado como argumento a una cadena de caracteres y la imprime en pantalla.

## Graficar

Se presentan tres procedimientos, `calcValue`, `calcDiff`, y `calcInteg`, que se utilizan para calcular el valor de una función de segundo grado en un punto dado, su derivada y su integral, respectivamente. Estos valores se pueden utilizar para graficar un polinomio en ensamblador.

Para graficar un polinomio, primero se especifica el rango de valores de  $x$  para los cuales se desea graficar la función. Luego, se itera sobre este rango de valores de  $x$ , llamando a uno de estos procedimientos para calcular el valor de la función en cada punto. El resultado se utiliza para dibujar un pixel en la pantalla en la posición correspondiente a ese valor de  $x$  e  $y$ . Al hacer esto para cada punto en el rango especificado, se dibuja una gráfica del polinomio en la pantalla.

## Explicación de la arquitectura x86 usada en el programa

El 8086 es un microprocesador de 16 bits que fue producido por Intel a principios de la década de 1980. Fue el primer microprocesador de la familia x86, que ha sido la línea de microprocesadores más popular para computadoras personales y servidores desde entonces.

La arquitectura x86 se caracteriza por tener un conjunto de instrucciones de 32 bits y una estructura de registros de 16 bits, lo que significa que el microprocesador puede trabajar con valores de 16 bits y 32 bits. Esta arquitectura también incluye un conjunto de instrucciones de 8 bits y una estructura de registros de 8 bits, lo que significa que el microprocesador también puede trabajar con valores de 8 bits.

El 8086 tiene una arquitectura Von Neumann, lo que significa que tiene una memoria principal compartida para datos y código. Tiene un bus de 16 bits para comunicarse con la memoria principal y un bus de 8 bits para comunicarse con dispositivos de entrada/salida (E/S). El microprocesador también tiene un conjunto de 16 registros de 16 bits, que se pueden usar para almacenar y procesar datos.

La arquitectura x86 se ha mantenido muy similar a lo largo de los años, aunque ha habido algunos cambios y mejoras. Los microprocesadores x86 más recientes tienen un conjunto de instrucciones más grande y más registros, y también tienen capacidades de virtualización y protección de memoria.

## Los registros del 8086

Los registros del procesador 8086 son pequeñas áreas de memoria que se encuentran dentro del propio microprocesador y que se pueden usar para almacenar y procesar datos de manera más rápida que accediendo a la memoria principal.

El 8086 tiene 16 registros de 16 bits, que se dividen en dos grupos: los registros de propósito general y los registros de segmento. Los registros de propósito general son aquellos que se pueden usar para cualquier propósito y se dividen a su vez en dos grupos: los registros de 16 bits y los registros de 8 bits. Los registros de 16 bits son:

AX: se puede dividir en dos registros de 8 bits, AH y AL, que se pueden usar por separado o juntos.

BX: se puede dividir en dos registros de 8 bits, BH y BL, que se pueden usar por separado o juntos.

CX: se puede dividir en dos registros de 8 bits, CH y CL, que se pueden usar por separado o juntos.

DX: se puede dividir en dos registros de 8 bits, DH y DL, que se pueden usar por separado o juntos.

Los registros de 8 bits son:

AH, BH, CH, DH: son parte de los registros AX, BX, CX y DX, respectivamente, y se pueden usar por separado.

AL, BL, CL, DL: son parte de los registros AX, BX, CX y DX, respectivamente, y se pueden usar por separado.

Los registros de segmento son aquellos que se usan para direccionar la memoria. Son:

CS: registro de segmento de código, que indica el segmento de memoria donde se encuentra el código que se está ejecutando.

DS: registro de segmento de datos, que indica el segmento de memoria donde se encuentran los datos que se están procesando.

SS: registro de segmento de pila, que indica el segmento de memoria donde se encuentra la pila.

ES: registro de segmento de área extra, que se puede usar para almacenar datos adicionales.

Además, el 8086 tiene otros registros especiales, como el registro de instrucción pointer (IP) que indica la siguiente instrucción a ejecutar, el registro de flag (flag) que indica el estado del procesador y el registro de stack pointer (SP) que indica la última posición de la pila.

## Aritmética en los 8086

Las operaciones aritméticas con signo en el procesador 8086 se realizan mediante instrucciones de dos bytes. Algunas de las operaciones aritméticas con signo más comunes son:

**ADD:** Suma dos números con signo y almacena el resultado en un registro o en la memoria.

**ADC:** Realiza una suma con acarreo entre dos números con signo y almacena el resultado en un registro o en la memoria.

**SUB:** Resta dos números con signo y almacena el resultado en un registro o en la memoria.

**SBB:** Realiza una resta con préstamo entre dos números con signo y almacena el resultado en un registro o en la memoria.

**MUL:** Multiplica dos números con signo y almacena el resultado en el registro AX.

**IMUL:** Realiza una multiplicación con signo entre dos números con signo y almacena el resultado en el registro AX.

**DIV:** Divide dos números con signo y almacena el cociente en el registro AL y el resto en el registro AH.

**IDIV:** Realiza una división con signo entre dos números con signo y almacena el cociente en el registro AL y el resto en el registro AH.

Es importante tener en cuenta que algunas de estas operaciones pueden requerir la utilización de los registros de soporte, como el registro DX, para almacenar parte del resultado. Además, es necesario tener cuidado al utilizar estas operaciones con números con signo, ya que pueden generar resultados inesperados si se produce un desbordamiento durante la operación.

## API de DOS

La API (Application Programming Interface) de DOS es un conjunto de rutinas y funciones que se pueden utilizar para interactuar con el sistema operativo DOS. Estas funciones se invocan mediante la ejecución de una interrupción de software con el número 21h.

En el código proporcionado se han utilizado varias de estas funciones de la API de DOS. A continuación, se describen algunas de ellas:

**INT 21h, AH=00h:** Esta función se utiliza para leer un carácter del teclado y almacenarlo en el registro AL.

**INT 21h, AH=02h:** Esta función se utiliza para imprimir un carácter almacenado en el registro DL en la pantalla.

INT 21h, AH=09h: Esta función se utiliza para imprimir una cadena de caracteres almacenada en la memoria. El offset de la cadena se almacena en el registro DX y el segmento de la cadena se almacena en el registro DS.

INT 21h, AH=4Ch: Esta función se utiliza para finalizar la ejecución del programa y volver al sistema operativo.

INT 16h, AH=00h: Es una función de la API de DOS que se utiliza para leer caracteres desde el teclado. Esta función se invoca mediante una interrupción software con el código 16h y se puede utilizar para leer caracteres individuales o para leer cadenas de caracteres. Si se establece el registro AH con el valor 00h, la función INT 16h leerá un carácter desde el teclado y lo almacenará en el registro AL.

Es importante tener en cuenta que estas funciones de la API de DOS pueden tener diferentes parámetros y opciones según el valor del registro AH y otros registros de soporte. Por ejemplo, la función INT 21h, AH=09h puede tener diferentes comportamientos según el valor de los registros DS y DX. Además, es necesario tener cuidado al utilizar estas funciones, ya que pueden generar comportamientos inesperados si se proporcionan parámetros incorrectos o si el programa está tratando de acceder a una dirección de memoria no válida.

## Modo video

El modo de video es una configuración que determina cómo se muestra la información en la pantalla. El modo de video se establece utilizando el interruptor 10h de la BIOS (Basic Input/Output System) de la computadora.

El código ah = 00h, al = 12h establece el modo de video 12h, también conocido como modo de video de alta resolución. Este modo de video proporciona una resolución de 640x480 pixeles con 16 colores. Es utilizado principalmente en aplicaciones gráficas y juegos.

Para establecer el modo de video, se envía una interrupción 10h con el código de operación 00h y el código del modo de video en el registro al. En este caso, se está estableciendo el modo de video 12h.

Es importante tener en cuenta que el modo de video de alta resolución no es compatible con todas las tarjetas gráficas y monitores. Es posible que sea necesario utilizar un modo de video diferente dependiendo del hardware de la computadora.