Actividad 9

Christian Geovany Muñoz Rodríguez

Ingeniería en computación

Código: 221350605

Seminario de Solución de Problemas de Traductores de lenguajes I – D09 (Lunes y Miércoles de 1 a 3)

Maestro: José Juan Meza Espinosa

Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías

09 de abril del 2023

Código:

```
org 100h


JMP inicio


variableX dw ?

variableY dw ?

clic dw 1 ; inicializa la variable clic con el valor 1


seno: db  0, 12, 23, 32, 38, 40, 38, 32, 23, 12, 5, -12, -23, -32, -38, -40, -38, -32, -23, -12, -9, 12, 23, 32, 38, 40, 38, 32, 23, 12, 1, -12, -23, -32, -38, -40, -38, -32, -23, -12, -2, 12, 23, 32, 38, 40, 38, 32, 23

coseno: db 40, 38, 32, 23, 12, 2, -12, -23, -32, -38, -40, -38, -32, -23, -12, -7, 12, 23, 32, 38, 40, 38, 32, 23, 12, 1, -12, -23, -32, -38, -40, -38, -32, -23, -12, -2, 12, 23, 32, 38, 40, 38, 32, 23, 12, 2, -12, -23, -32


inicio:


    MOV AL, 13h ; establece el modo grafico 320x200 con 256 colores

    MOV AH, 0h

    int 10h


    next:

        MOV AX, 0

        int 33h ; llama a la funcion INT 33h del mouse

    mouse:

        MOV AX, 3

        int 33h ; llama a la funcion INT 33h del mouse
```

```
        cmp BX, clic ; compara el valor del registro BX con la variable
clic
        jnz mouse ; salta a la etiqueta mouse si los valores no son
iguales

        shr CX, 1
        MOV variableX, CX ; guarda el valor de CX en la variable X
        MOV variableY, DX ; guarda el valor de DX en la variable Y

    circulo:
        xor CX, CX
        xor DX, DX
        xor AX, AX

        MOV CL, seno[BX] ; carga el valor de la tabla seno en el registro
CL
        MOV DL, coseno[BX] ; carga el valor de la tabla coseno en el
registro DL

        add CX, variableX ; suma el valor de la variable X al registro CX
        add DX, variableY ; suma el valor de la variable Y al registro DX

        cmp CL, 0 ; compara el valor del registro CL con 0
        jl regCX ; salta a la etiqueta regCX si CL es menor que 0

        sig: ;
            cmp variableX, 255 ; compara el valor de la variable X con
255
            jle clearCH ; salta a la etiqueta clearCH si X es menor o
igual a 255
            cmp variableY, 255 ; compara el valor de la variable Y con
255
```

```
            jle clearDH ; salta a la etiqueta clearDH si Y es menor o
igual a 255


        centro:
            xor AX, AX ; borra el contenido del registro AX

            MOV Al, 0Fh ; establece el color de fondo
            MOV AH, 0Ch ; establece el color del circulo
            int 10h ; llama a la funcion INT 10h para dibujar el circulo

            inc BX ; incrementa el valor del registro BX en 1
            cmp BX, 40 ; compara el valor del registro BX con 40
            jnz circulo ; salta a la etiqueta circulo si BX no es igual a
40

        fin:
            JMP next ; salta a la etiqueta next
            MOV AH, 0h ; llama a la funcion INT 16h para salir del
programa
            int 16h
            ret

        regCX:       ; Si CL es negativo, establece CH en 00h y salta a
la etiqueta "sig"
            MOV CH, 00h
            JMP sig

        clearCH:     ; Establece CH en 00h y comprueba si variableY es
menor o igual a 255,
                     ; si es cierto salta a "clearDH", en caso contrario
salta a "centro"
            MOV CH, 00h
```

```
    cmp variableY, 255
    jle clearDH
    JMP centro

clearDH:     ; Establece DH en 00h y salta a "centro"
    MOV DH, 00h
    JMP centro
```

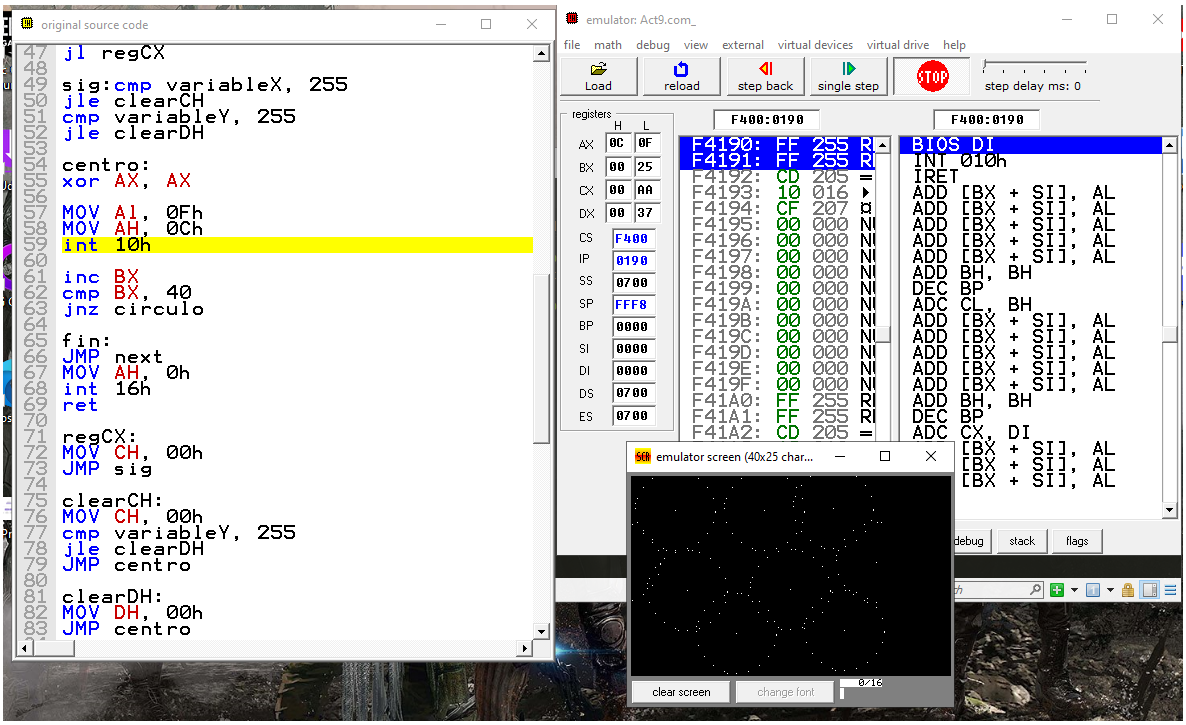
Desarrollo:

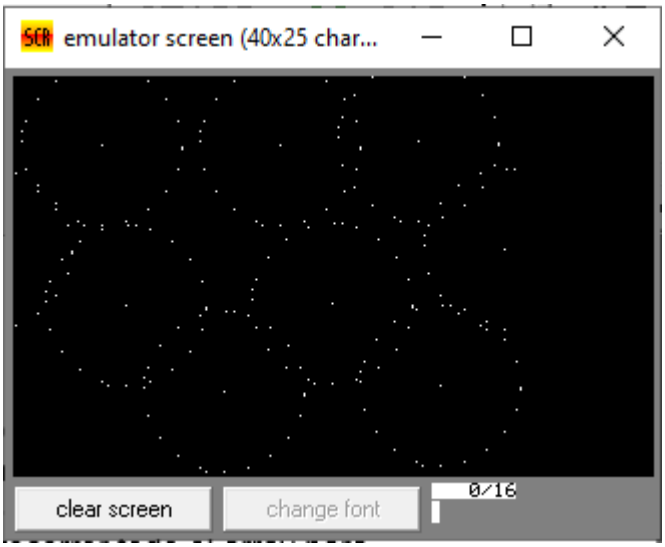
El programa anterior es un ejemplo de un programa en lenguaje ensamblador que dibuja un círculo en la pantalla en función de la posición del mouse.

En términos generales, el programa comienza con una sección de datos en la que se definen las variables que se utilizarán en el programa, incluyendo las coordenadas del mouse y una variable para registrar el clic del mouse. También hay dos tablas que se utilizan para calcular los valores de seno y coseno necesarios para dibujar el círculo.

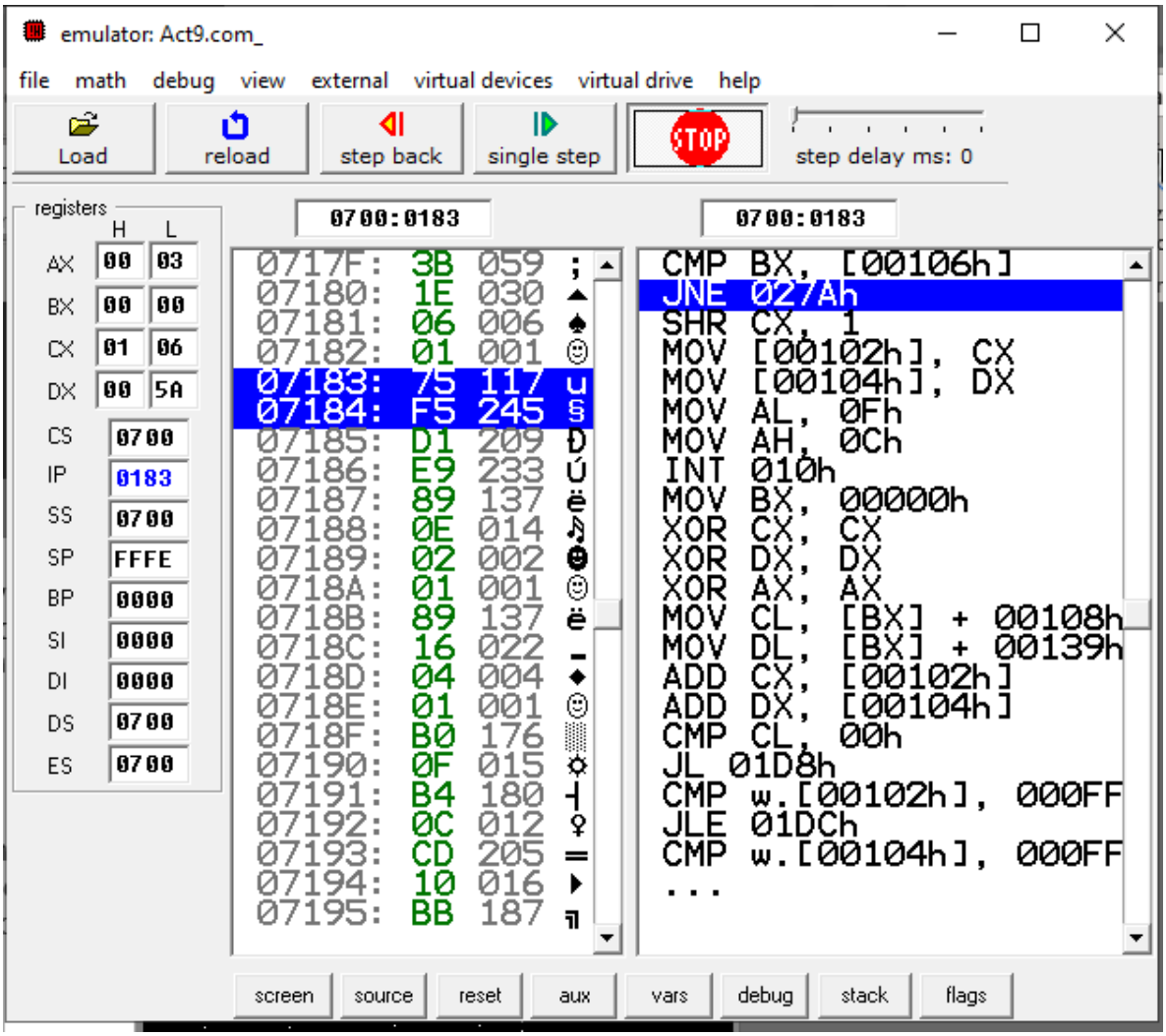
```
org 100h
    JMP inicio

    variableX dw ?
    variableY dw ?
    clic dw 1

    seno: db   0, 12, 23, 32, 38, 40, 38, 32, 23, 12, 5, -12, -23, -
    coseno: db 40, 38, 32, 23, 12, 2, -12, -23, -32, -38, -40, -38,
```

La sección de inicio del programa inicializa algunos registros y la pantalla de modo gráfico, y luego entra en un bucle que espera hasta que se detecta un clic del mouse. Cuando se hace clic en el mouse, se guarda la posición del mouse y se dibuja un círculo en esa posición utilizando valores de seno y coseno precalculados.





El programa utiliza una serie de etiquetas y saltos para controlar el flujo del programa, y contiene una mezcla de instrucciones de movimiento de datos, operaciones aritméticas y llamadas a la función de interrupción del BIOS de la pantalla para dibujar en la pantalla.



Conclusiones:

En conclusión, el programa hace uso de interrupciones para dibujar una circunferencia en pantalla. En particular, se utiliza la interrupción 10h para cambiar el modo gráfico y para dibujar píxeles en la pantalla, mientras que la interrupción 33h se utiliza para obtener las coordenadas del cursor del mouse. Estas interrupciones son esenciales para realizar la tarea de dibujar la circunferencia, lo que demuestra la importancia de las interrupciones en la programación de sistemas.

Bibliografía:

Brey, B. B. (2006). *Microprocesadores Intel : 8086/8088, 80186/80188, 80286, 80386 y 80486, Pentium, procesador Pentium Pro, Pentium II, Pentium III y Pentium 4: arquitectura, programación e interfaces*. Pearson Educación.