

Proyecto 2023-A

Christian Geovany Muñoz Rodríguez

Ingeniería en computación

Código: 221350605

**Seminario de Solución de Problemas de
Traductores de Lenguajes I – D04 (Martes y
Jueves de 11 a 1)**

Maestro: José Juan Meza Espinoza

Universidad de Guadalajara

**Centro Universitario de Ciencias Exactas e
Ingenierías**

17 de mayo del 2023



Código:

```
.MODEL SMALL
```

```
.STACK 100h
```

```
.DATA
```

```
    x1 DW 160
```

```
    y1 DW 100
```

```
    x2 DW 320
```

```
    y2 DW 200
```

```
    dxs DW ?
```

```
    dy DW ?
```

```
    E DW ?
```

```
    Dn DW ?
```

```
    Dp DW ?
```

```
    x DW ?
```

```
    y DW ?
```

```
.CODE
```

```
    ASSUME DS:DATA, CS:CODE
```

```
BEGIN:
```

```
    MOV AX, @DATA
```

```
    MOV DS, AX
```

CALL GRAPHMODE

CALL BRESENHAM

CALL LINEA

CALL GETKEY

MOV AX, 4C00h

INT 21h

GRAPHMODE PROC NEAR

MOV AX, 0

MOV AL, 13h

INT 10h

RET

GETKEY PROC NEAR

MOV AH, 10h

INT 16h

RET

BRESENHAM PROC NEAR

; Calcula las diferencias en las coordenadas x e y

MOV AX, x2

SUB AX, x1

MOV DX, AX

MOV dxs, DX

MOV AX, y2

SUB AX, y1

MOV DX, AX

MOV dy, DX

; Calcula E, Dn y Dp

MOV AX, 2

MUL dy

SUB AX, dxs

MOV E, AX

MOV AX, 2

MUL dy

MOV DX, AX

SUB DX, dxs

MOV Dp, DX

MOV AX, 2

MUL dy

MOV Dn, AX

; Inicializa las coordenadas x, y

MOV AX, x1

MOV x, AX

MOV AX, y1

MOV y, AX

RET

LINEA PROC NEAR

MOV CX, dxs

; Calcula el desplazamiento horizontal para centrar la línea

MOV AX, 320

SUB AX, dxs

SHR AX, 1

MOV DX, AX

ADD x, DX

; Calcula el desplazamiento vertical para centrar la línea

MOV AX, 200

SUB AX, dy

SHR AX, 1

MOV DX, AX

ADD y, DX

DRAW_LINE:

MOV AH, 0Ch

INT 10h

MOV AX, E

```
CMP AX, 0  
  
JL ELSE_CONDITION
```

```
ADD AX, Dp  
  
ADD x, 1  
  
MOV E, AX  
  
JMP CONTINUE_LOOP
```

```
ELSE_CONDITION:
```

```
ADD AX, Dn  
  
MOV E, AX
```

```
CONTINUE_LOOP:
```

```
LOOP DRAW_LINE
```

```
RET
```

```
END BEGIN
```

Desarrollo:

El programa es un dibujante de líneas en modo gráfico de 320x200 píxeles, que utiliza el algoritmo de Bresenham para trazar una línea entre dos puntos dados por el usuario. A continuación se explican las secciones más importantes del código:

Sección DATA

En esta sección, se definen las variables que serán utilizadas en el programa. Se utiliza la directiva DW para definir valores de palabra (word), y se asignan valores

a las coordenadas x e y de los dos puntos que se conectarán con una línea, así como a otras variables que se utilizarán posteriormente en el código.

```
.DATA
x1 DW 160
y1 DW 100
x2 DW 320
y2 DW 200
dxs DW ?
dy DW ?
E DW ?
Dn DW ?
Dp DW ?
x DW ?
y DW ?
```

Sección CODE

En esta sección se encuentra el código principal del programa, que comienza con la etiqueta BEGIN.

```
.CODE
    ASSUME DS:DATA, CS:CODE
BEGIN:
    MOV AX, @DATA
    MOV DS, AX

    CALL GRAPHMODE
    CALL BRESENHAM
    CALL LINEA
    CALL GETKEY

    MOV AX, 4C00h
    INT 21h
```

Subrutina GRAPHMODE

Esta subrutina cambia el modo de vídeo a 320x200 píxeles en 256 colores, utilizando la interrupción de BIOS INT 10h.

```
GRAPHMODE PROC NEAR
    MOV AX, 0
    MOV AL, 13h
    INT 10h
    RET
```

Subrutina GETKEY

Esta subrutina espera a que el usuario presione una tecla para salir del programa, utilizando la interrupción de BIOS INT 16h.

```
GETKEY PROC NEAR
    MOV AH, 10h
    INT 16h
    RET
```

Subrutina BRESENHAM

Esta subrutina implementa el algoritmo de Bresenham, que es un método eficiente para dibujar líneas en una pantalla de píxeles. Primero se calculan las diferencias en las coordenadas x e y, y se almacenan en las variables dxs y dy, respectivamente. Luego se calculan las variables E, Dn y Dp, que son utilizadas en el algoritmo de Bresenham para determinar qué píxeles dibujar. Por último, se inicializan las coordenadas x e y con los valores de los puntos de inicio de la línea.

```
BRESENHAM PROC NEAR
    ; Calcula las diferencias en las coordenadas x e y
    MOV AX, x2
    SUB AX, x1
    MOV DX, AX
    MOV dxs, DX
    MOV AX, y2
    SUB AX, y1
    MOV DX, AX
    MOV dy, DX

    ; Calcula E, Dn y Dp
    MOV AX, 2
    MUL dy
    SUB AX, dxs
    MOV E, AX
    MOV AX, 2
    MUL dy
    MOV DX, AX
    SUB DX, dxs
    MOV Dp, DX
    MOV AX, 2
    MUL dy
    MOV Dn, AX

    ; Inicializa las coordenadas x, y
    MOV AX, x1
    MOV x, AX
    MOV AX, y1
    MOV y, AX

    RET
```


Subrutina LINEA

Esta subrutina dibuja la línea utilizando el algoritmo de Bresenham implementado en la subrutina BRESENHAM. Primero se calculan los desplazamientos horizontal y vertical para centrar la línea en la pantalla. Luego se utiliza un bucle para dibujar cada píxel de la línea utilizando la interrupción de BIOS INT 10h y las variables E, Dn y Dp calculadas anteriormente.

```
LINEA PROC NEAR
    MOV CX, dxs
    ; Calcula el desplazamiento horizontal para centrar la línea
    MOV AX, 320
    SUB AX, dxs
    SHR AX, 1
    MOV DX, AX
    ADD x, DX
    ; Calcula el desplazamiento vertical para centrar la línea
    MOV AX, 200
    SUB AX, dy
    SHR AX, 1
    MOV DX, AX
    ADD y, DX
DRAW_LINE:
    MOV AH, 0Ch
    INT 10h
    MOV AX, E
    CMP AX, 0
    JL ELSE_CONDITION
    ADD AX, Dp
    ADD x, 1
    MOV E, AX
    JMP CONTINUE_LOOP
ELSE_CONDITION:
    ADD AX, Dn
    MOV E, AX
CONTINUE_LOOP:
    LOOP DRAW_LINE
RET
```

Etiqueta END BEGIN

Esta es la última línea del código y marca el final del programa.

Otras secciones

Además de las secciones mencionadas anteriormente, también se utilizan las directivas ASSUME para asociar segmentos de memoria a los registros de segmento, y STACK para definir el tamaño de la pila. También se utilizan instrucciones de movimiento de datos (MOV), suma (ADD), resta (SUB), comparación (CMP), salto condicional (JL) y bucles (LOOP), entre otras.

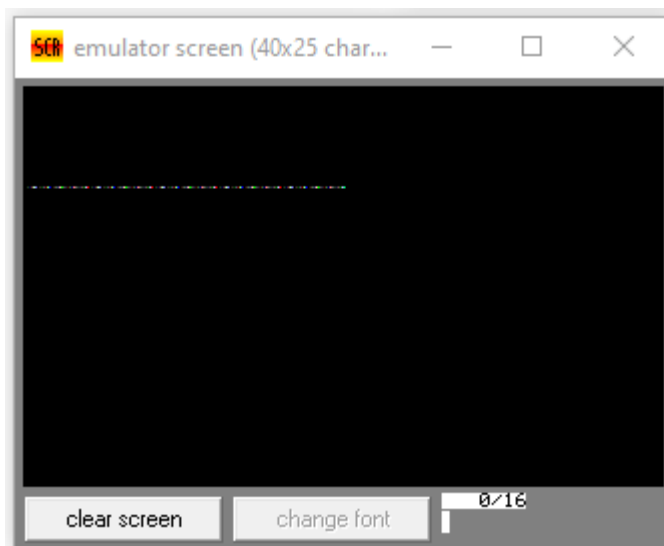
The screenshot displays the x86-64 emulator interface with three main windows:

- Assembly Code Window:** Shows the assembly code being executed. The current instruction is `ADD w[00012h], 01h` at address `0722:0005`. The code includes instructions for moving data, comparing, jumping, and drawing lines.
- Registers Window:** Shows the state of the CPU registers. The `AX` register contains `12 29`, `DX` contains `00 3E`, and `SI` contains `0000`.
- Memory Window:** Shows the memory address `0722:0005` with the value `00 00 00 00`.

The assembly code window shows the following instructions:

```

0722:0005 MOV AX, [0000Ch]
0722:0006 CMP AX, 00000h
0722:0007 JLE 0AFh
0722:0008 ADD w[00010h], 01h
0722:0009 MOV [0000Ch], AX
0722:000A JMP 0B6h
0722:000B ADD AX, [0000Eh]
0722:000C MOV [0000Ch], AX
0722:000D LOOP 0195h
0722:000E RET
0722:000F NOP
0722:0010 NOP
0722:0011 NOP
0722:0012 NOP
0722:0013 ...
0722:0014 ...
0722:0015 ...
0722:0016 ...
0722:0017 ...
0722:0018 ...
0722:0019 ...
0722:001A ...
0722:001B ...
0722:001C ...
0722:001D ...
0722:001E ...
0722:001F ...
0722:0020 ...
0722:0021 ...
0722:0022 ...
0722:0023 ...
0722:0024 ...
0722:0025 ...
0722:0026 ...
0722:0027 ...
0722:0028 ...
0722:0029 ...
0722:002A ...
0722:002B ...
0722:002C ...
0722:002D ...
0722:002E ...
0722:002F ...
0722:0030 ...
0722:0031 ...
0722:0032 ...
0722:0033 ...
0722:0034 ...
0722:0035 ...
0722:0036 ...
0722:0037 ...
0722:0038 ...
0722:0039 ...
0722:003A ...
0722:003B ...
0722:003C ...
0722:003D ...
0722:003E ...
0722:003F ...
0722:0040 ...
0722:0041 ...
0722:0042 ...
0722:0043 ...
0722:0044 ...
0722:0045 ...
0722:0046 ...
0722:0047 ...
0722:0048 ...
0722:0049 ...
0722:004A ...
0722:004B ...
0722:004C ...
0722:004D ...
0722:004E ...
0722:004F ...
0722:0050 ...
0722:0051 ...
0722:0052 ...
0722:0053 ...
0722:0054 ...
0722:0055 ...
0722:0056 ...
0722:0057 ...
0722:0058 ...
0722:0059 ...
0722:005A ...
0722:005B ...
0722:005C ...
0722:005D ...
0722:005E ...
0722:005F ...
0722:0060 ...
0722:0061 ...
0722:0062 ...
0722:0063 ...
0722:0064 ...
0722:0065 ...
0722:0066 ...
0722:0067 ...
0722:0068 ...
0722:0069 ...
0722:006A ...
0722:006B ...
0722:006C ...
0722:006D ...
0722:006E ...
0722:006F ...
0722:0070 ...
0722:0071 ...
0722:0072 ...
0722:0073 ...
0722:0074 ...
0722:0075 ...
0722:0076 ...
0722:0077 ...
0722:0078 ...
0722:0079 ...
0722:007A ...
0722:007B ...
0722:007C ...
0722:007D ...
0722:007E ...
0722:007F ...
0722:0080 ...
0722:0081 ...
0722:0082 ...
0722:0083 ...
0722:0084 ...
0722:0085 ...
0722:0086 ...
0722:0087 ...
0722:0088 ...
0722:0089 ...
0722:008A ...
0722:008B ...
0722:008C ...
0722:008D ...
0722:008E ...
0722:008F ...
0722:0090 ...
0722:0091 ...
0722:0092 ...
0722:0093 ...
0722:0094 ...
0722:0095 ...
0722:0096 ...
0722:0097 ...
0722:0098 ...
0722:0099 ...
0722:009A ...
0722:009B ...
0722:009C ...
0722:009D ...
0722:009E ...
0722:009F ...
0722:00A0 ...
0722:00A1 ...
0722:00A2 ...
0722:00A3 ...
0722:00A4 ...
0722:00A5 ...
0722:00A6 ...
0722:00A7 ...
0722:00A8 ...
0722:00A9 ...
0722:00AA ...
0722:00AB ...
0722:00AC ...
0722:00AD ...
0722:00AE ...
0722:00AF ...
0722:00B0 ...
0722:00B1 ...
0722:00B2 ...
0722:00B3 ...
0722:00B4 ...
0722:00B5 ...
0722:00B6 ...
0722:00B7 ...
0722:00B8 ...
0722:00B9 ...
0722:00BA ...
0722:00BB ...
0722:00BC ...
0722:00BD ...
0722:00BE ...
0722:00BF ...
0722:00C0 ...
0722:00C1 ...
0722:00C2 ...
0722:00C3 ...
0722:00C4 ...
0722:00C5 ...
0722:00C6 ...
0722:00C7 ...
0722:00C8 ...
0722:00C9 ...
0722:00CA ...
0722:00CB ...
0722:00CC ...
0722:00CD ...
0722:00CE ...
0722:00CF ...
0722:00D0 ...
0722:00D1 ...
0722:00D2 ...
0722:00D3 ...
0722:00D4 ...
0722:00D5 ...
0722:00D6 ...
0722:00D7 ...
0722:00D8 ...
0722:00D9 ...
0722:00DA ...
0722:00DB ...
0722:00DC ...
0722:00DD ...
0722:00DE ...
0722:00DF ...
0722:00E0 ...
0722:00E1 ...
0722:00E2 ...
0722:00E3 ...
0722:00E4 ...
0722:00E5 ...
0722:00E6 ...
0722:00E7 ...
0722:00E8 ...
0722:00E9 ...
0722:00EA ...
0722:00EB ...
0722:00EC ...
0722:00ED ...
0722:00EE ...
0722:00EF ...
0722:00F0 ...
0722:00F1 ...
0722:00F2 ...
0722:00F3 ...
0722:00F4 ...
0722:00F5 ...
0722:00F6 ...
0722:00F7 ...
0722:00F8 ...
0722:00F9 ...
0722:00FA ...
0722:00FB ...
0722:00FC ...
0722:00FD ...
0722:00FE ...
0722:00FF ...
0722:0100 ...
0722:0101 ...
0722:0102 ...
0722:0103 ...
0722:0104 ...
0722:0105 ...
0722:0106 ...
0722:0107 ...
0722:0108 ...
0722:0109 ...
0722:010A ...
0722:010B ...
0722:010C ...
0722:010D ...
0722:010E ...
0722:010F ...
0722:0110 ...
0722:0111 ...
0722:0112 ...
0722:0113 ...
0722:0114 ...
0722:0115 ...
0722:0116 ...
0722:0117 ...
0722:0118 ...
0722:0119 ...
0722:011A ...
0722:011B ...
0722:011C ...
0722:011D ...
0722:011E ...
0722:011F ...
0722:0120 ...
0722:0121 ...
0722:0122 ...
0722:0123 ...
0722:0124 ...
0722:0125 ...
0722:0126 ...
0722:0127 ...
0722:0128 ...
0722:0129 ...
0722:012A ...
0722:012B ...
0722:012C ...
0722:012D ...
0722:012E ...
0722:012F ...
0722:0130 ...
0722:0131 ...
0722:0132 ...
0722:0133 ...
0722:0134 ...
0722:0135 ...
0722:0136 ...
0722:0137 ...
0722:0138 ...
0722:0139 ...
0722:013A ...
0722:013B ...
0722:013C ...
0722:013D ...
0722:013E ...
0722:013F ...
0722:0140 ...
0722:0141 ...
0722:0142 ...
0722:0143 ...
0722:0144 ...
0722:0145 ...
0722:0146 ...
0722:0147 ...
0722:0148 ...
0722:0149 ...
0722:014A ...
0722:014B ...
0722:014C ...
0722:014D ...
0722:014E ...
0722:014F ...
0722:0150 ...
0722:0151 ...
0722:0152 ...
0722:0153 ...
0722:0154 ...
0722:0155 ...
0722:0156 ...
0722:0157 ...
0722:0158 ...
0722:0159 ...
0722:015A ...
0722:015B ...
0722:015C ...
```



Este análisis de código nos muestra un ejemplo sencillo pero eficiente de cómo se puede utilizar el algoritmo de Bresenham para dibujar líneas en una pantalla de píxeles. A lo largo del código se pueden observar diferentes técnicas de

programación utilizadas para optimizar el rendimiento y mejorar la experiencia del usuario.

Es importante destacar la relevancia de entender y saber utilizar algoritmos eficientes como el de Bresenham, ya que son fundamentales en el desarrollo de programas y aplicaciones que requieran de gráficos o animaciones en tiempo real. Además, la utilización de técnicas de optimización puede marcar una gran diferencia en la calidad y velocidad del dibujado, logrando una mejor experiencia para el usuario.

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.