

LABORATÓRIO DE LINGUAGEM DE MONTAGEM

PROGRAMAÇÃO DE VIDEO MODO TEXTO

Capítulo 16 do livro texto, págs. 331 a 356

- ❖ **PIXEL:** elemento de figura. É uma posição na tela.
- ❖ **RESOLUÇÃO:** número de linhas e colunas. Por exemplo, o modo 320 X 200 tem 64000 pixels.



- ❖ **APA (All points addressable):** modo gráfico, onde um pixel é mapeado em um simples ponto (dot).

❖ Seleção de modo:

INT 10H Função 0
Seleciona o modo gráfico
Entrada: AH = 0
 AL = número do modo
Saída: nenhuma

Vídeo Adapter Graphics Display Modes

número	CGA graphics
4	320 X 200 4 cores
5	320 X 200 4 cores (burst off)
6	640 X 200 2 cores
	EGA graphics
D	320 X 200 16 cores
E	640 X 200 16 cores
F	640 X 350 monocromático
10	640 X 350 16 cores
	VGA graphics
11	640 X 480 2 cores
12	640 X 480 16 cores
13	320 X 200 256 cores

❖ CGA MODO GRÁFICO

- . 160 x 100 - resolução baixa (não suportada pela INT 10H da BIOS)
- . 320 x 200 - resolução média
- . 640 x 200 - resolução alta.

O CGA pode trabalhar com 16 cores:

I R G B	COR
0 0 0 0	preto
0 0 0 1	azul
0 0 1 0	verde
0 0 1 1	cyan
0 1 0 0	vermelho
0 1 0 1	magenta
0 1 1 0	marrom
0 1 1 1	branco
1 0 0 0	cinza
1 0 0 1	azul claro
1 0 1 0	verde claro
1 0 1 1	cyan claro
1 1 0 0	rosa
1 1 0 1	magenta claro
1 1 1 0	amarelo
1 1 1 1	branco intenso

❖ PALHETAS

Palheta	valor do pixel	cor
0	0	fundo
	1	verde
	2	vermelho
	3	marrom
1	0	fundo
	1	cyan
	2	magenta
	3	branco

❖ Como Selecionar a Paleta e a Cor de Fundo

INT 10H, função 0Bh
Seleciona paleta ou cor de fundo

Subfunção 0 : Seleciona cor de fundo
entrada: AH = 0BH
BH = 0
BL = número da cor (0 -15)
saída : nenhuma

Subfunção 1 : Seleciona paleta
entrada: AH = 0BH
BH = 1
BL = número da paleta (0 - 1)
saída: nenhuma

❖ Ler e escrever um pixel.

INT 10H função 0CH

Escreve um pixel

entrada: AH = 0CH

AL = valor do pixel

BH = página

CX = coluna

DX = linha

saída: nenhuma

INT 10H função 0DH

lê um pixel

entrada: AH = 0CH

BH = página

CX = coluna

DX = linha

saída: AL = valor do pixel

❖ Escrevendo diretamente na memória

Para atualização rápida da tela, como em um vídeo game, é necessário escrever diretamente na memória de vídeo.

A memória é dividida em duas regiões, onde os pixels das linhas pares são armazenados nos primeiros 8K (B800:0000 até B800:1FFF), e os das linhas ímpares nos 8K restantes (B800:2000 até B800:3FFF). Cada linha é representada por 50H bytes.

B800:0000 - LINHA 0

B800:2000 - LINHA 1

B800:0050 - LINHA 2

B800:2050 - LINHA 3

B800:1EFO - LINHA 198

B800:3EFO - LINHA 199



❖ **ESCREVENDO UM TEXTO NO MODO GRÁFICO**

No modo gráfico, caracteres não são gerados pelo circuito gerador de caracteres, e sim de fontes de caracteres armazenados na memória. No modo gráfico, o cursor também não aparece, mas podemos posicioná-lo pela INT 10H, função 2.

Resolução Gráfica	coluna (texto)	linha (texto)
320 X 200	40	25
640 X 200	80	25
640 X 350	80	25
640 X 480	80	25

❖ **VGA MODO GRÁFICO**

O adaptador VGA tem maior resolução que o EGA e pode mostrar 640 X 480 no modo 12h. Ele pode ainda gerar 64 níveis de verde, vermelho e azul, produzindo uma combinação de 256K diferentes cores. Um máximo de 256 cores pode ser mostrado simultaneamente. O VGA pode emular o EGA e o CGA.

No VGA modo 13h, que suporta 256 cores, cada valor de pixel tem 1 byte, e seleciona um registrador de cores. Estes registradores são carregados inicialmente com as cores default, mas é possível alterá-los usando:

INT 10H, função 10H, subfunção 10H
seta registrador de cores

entrada: AH = 10H

AL = 10H

BX = REGISTRADOR DE CORES

CH = VALOR DE VERDE

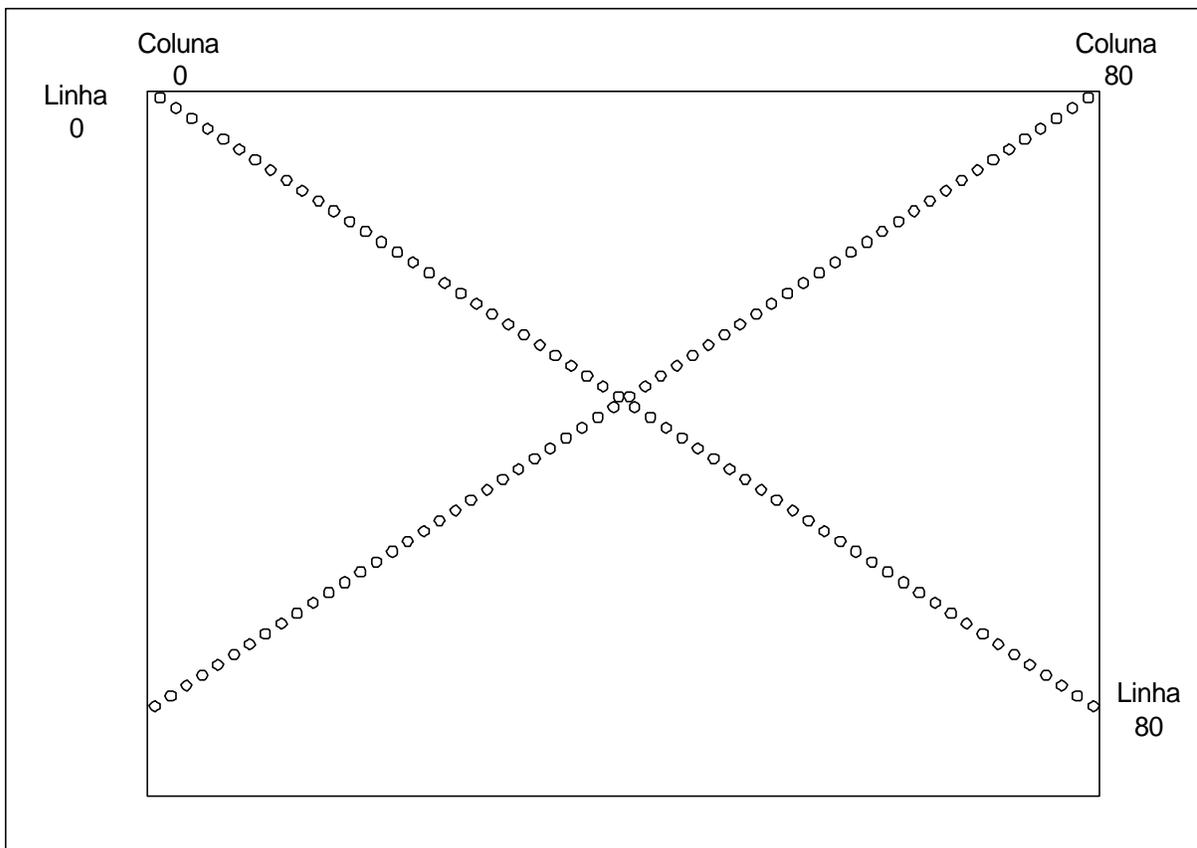
CL = VALOR DE AZUL

DH = VALOR DE VERMELHO

saída: nenhuma

❖ EXEMPLO 1:

O programa mostrado a seguir fazendo uso de uma macro, escreve diretamente na memória de vídeo. É programado um padrão de pixels em forma de um X como descrito na figura a seguir:



- **Exercício 1.** Modifique este programa para desenhar duas linhas paralelas nos lados esquerdo e direito da tela.

```

TITLE PGM12_1: Graphics display_0
.MODEL SMALL
    MOSTRA_PIXEL MACRO
        MOV AL,ES:[DI]           ; move byte para AL
        AND AL,1110011B         ; limpa o bit de posicao
        OR AL,1000B             ; escreve posicao 3,2
        STOSB                    ; carrega na memoria
    ENDM
.STACK 100H
.CODE
MAIN PROC
    MOV AH,0
        MOV AL,4h
        INT 10h
    MOV AX,0B800H                ; segmento memoria video
    MOV ES,AX                    ; aponta em ES
    MOV CX,0                      ; inicializa CX
    MOV DI,0000H                 ; inicializa DI
    MOV SI,2000H                 ; inicializa SI
mostra:  MOSTRA_PIXEL           ; linha "zero" = B800:0000
        MOV BX,DI                ; linha "um" = B800:2000
        MOV DI,SI                ; linha "dois" = B800:0050
        MOSTRA_PIXEL            ; linha "tres" = B800:2050
        MOV SI,DI
        MOV DI,BX
        ADD DI,4EH                ; pula para proxima linha
        ADD SI,50H
        INC CX
        CMP CX,80                ; escreve 80 pixels
        JL mostra
        MOV AH,0                  ; espera o teclado
        INT 16h
        MOV AH,4CH
        INT 21H
    MAIN ENDP
END MAIN

```

❖ EXEMPLO 2:

```
TITLE PGM16_1: DESENHA LINHA
.MODEL SMALL
.STACK 100H
.CODE
MAIN PROC
    MOV AH,0
    MOV AL,13           ; seleciona modo 13
    INT 10h
    MOV AH,0CH         ; escreve pixel
    MOV AL,1
    MOV CX,0           ; coluna inicial
    MOV DX,100        ; linha
Loop: INT 10H          ; desenha linha
    INC CX
    CMP CX,319        ; escreve 320 pixels
    JLE Loop
    MOV AH,0           ; espera o teclado
    INT 16h
    MOV AH,4CH        ;DOS EXIT
    INT 21H
MAIN ENDP
END MAIN
```

EXERCÍCIOS:

- **Exercício 1.** Modifique o programa do exemplo 1 para desenhar duas linhas paralelas nos lados esquerdo e direito da tela. (utilize escrita diretamente na memória)
- **Exercício 2:** Modifique o programa do exemplo 2 para desenhar duas linhas, dividindo a tela em quatro partes iguais. (utilize interrupções)
- **Exercício 3:** Mostrar uma letra "A" em vermelho, no canto superior direito da tela. Use o modo 4 e cor de fundo azul.
- **Exercício 4:** Assuma que você esteja usando uma palheta de 16 cores, escreva um pixel verde na página 2, na coluna 0 linha 0.
- **Exercício 5:** Escreva um programa que mostre as 256 cores default como 256 pixels na linha 100