

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

PROGRAMAÇÃO DE VIDEO

- **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).



ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

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

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

- **O adaptador CGA tem 3 resoluções:**
 - Resolução baixa – 160 X 100 (não suportada pela INT 10H)
 - Resolução média – 320 X 200
 - Resolução alta – 640 X 200
- **O adaptador CGA tem uma memória de 16KB no segmento B800h.**
- **A memória endereçável vai de B800:0000 até B800:3FFF**
- **Cada pixel é representado por um ou mais pontos, dependendo do modo:**
 - Resolução média 2 bits por pixel
 - Resolução alta 1 bit por pixel

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Seleção de modo:

INT 10H Função 0

Seleciona o modo gráfico

Entrada: AH = 0

AL = número do modo

Saída: nenhuma

Exemplo

MOV AH,0

MOV AL,06h ; MODO CGA, 2 CORES 640 X 200

INT 10H

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

CGA Resolução média pode trabalhar com 16 cores, 4 cores por vez

I R G B

0 0 0 0

0 0 0 1

0 0 1 0

0 0 1 1

0 1 0 0

0 1 0 1

0 1 1 0

0 1 1 1

1 0 0 0

1 0 0 1

1 0 1 0

1 0 1 1

1 1 0 0

1 1 0 1

1 1 1 0

1 1 1 1

COR

preto

azul

verde

cyan

vermelho

magenta

marrom

branco

cinza

azul claro

verde claro

cyan claro

rosa

magenta claro

amarelo

branco intenso

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

CGA Resolução média

- **4 cores por vez (limitado pela memória de vídeo)**
 - **320 X 200 = 64.000 pixels, cada pixel codificado por 2 bits – 128.000 bits ou 16.000 bytes (16k)**
 - **Para permitir diferentes combinações de 4 cores: PALETAS**

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

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

- **Como Selecionar a Paleta e a Cor de Fundo**

INT 10H, função 0Bh

Sub função 0 : Seleciona cor de fundo

**entrada: AH = 0BH
 BH = 0
 BL = número da cor (0 -15)
saída : nenhuma**

Sub função 1 : Seleciona paleta

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

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

- **Exemplo – Selecionar paleta 1 e cor de fundo azul claro,**

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

- Exemplo – Selecionar paleta 1 e cor de fundo azul claro,

MOV AH,0Bh	; função 0Bh
MOV BH, 00H	; cor de fundo
MOV BL,9	; azul claro
INT 10h	
MOV BH,1	; seleciona paleta
MOV BL,1	; paleta 1
INT 10H	

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

- **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

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

- **Exemplo – Copiar um pixel da coluna 50, linha 199, para um pixel na coluna 20, linha 40.**

MOV AH,0Dh	; le um pixel
MOV CX, 50	; coluna 50
MOV DX, 199	;linha 199
INT 10h	; em AL o valor do pixel
MOV AH,0Ch	; escreve um pixel
MOV CX, 20	; coluna 20
MOV DX, 40	;linha 40
INT 10h	; em AL o valor do pixel

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

- **Exemplo – Copiar um pixel da coluna 50, linha 199, para um pixel na coluna 20, linha 40.**

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

CGA Resolução alta

- **2 cores por vez**
 - Cada pixel – 0 ou 1
 - Possível escolher cor de fundo com a INT 10h, função 0Bh
 - Quando cor de fundo é selecionado, o valor do pixel 0 é a cor de fundo e o valor do pixel 1 é branco
- **Exemplo – Escrever um programa que:**
 - Seta o modo de vídeo 6 (CGA alta resolução)
 - Desenha uma linha
 - Le uma tecla de entrada
 - Seleciona o modo 3 (modo texto)

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

```
TITLE Desenha linha CGA
; desenha linha horizontal em HD
; na linha 100 coluna 301 até coluna 600
.MODEL SMALL
.STACK 0100H
.CODE
MAIN PROC
    MOV AX,6           ; modo 6, HD
    INT 10H
; desenha linha
    MOV AH, 0CH        ; escreve pixel
    MOV AL,1           ; branco
    MOV CX,301          ; coluna inicio
    MOV DX,100          ; linha
L1:    INT 10H
        INC CX
        CMP CX,600
        JLE L1
```

```
; le teclado
    MOV AH,0
    INT 16H
; seleciona modo texto
    MOV AX,3
    INT 10H
; retorno ao DOS
    MOV AH,4CH
    INT 21H
MAIN END
END MAIN
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

- **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
B800:3EF0 - LINHA 199



ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

- **EXEMPLO:** No modo de vídeo 4, determine o endereço do byte e a posição dos bits do pixel na linha 5 coluna 10
 - Linha 5 é a terceira linha ímpar, portanto começa em $2000h + 2 \times 50h = 20A0h$
 - No modo 4, cada pixel tem 2 bits, então cada byte armazena 4 pixels
 - A coluna 10 é a décima primeira coluna na linha, então o pixel deve ser o terceiro pixel no terceiro byte, ou seja o bits 3 e 2 do terceiro byte. O endereço do byte é $20A0h + 2 = 20A2h$

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

- **EXEMPLO:** Supor que o vídeo esteja no modo 4, escrever um pixel com valor 10b, na linha 5 coluna 10.

MOV AX,0B800H	; segmento memoria video
MOV ES,AX	
MOV DI, 20A2h	; offset do byte
MOV AL,ES:[DI]	; move byte para AL
AND AL,11110011B	; limpa o bit de posicao
OR AL,1000B	; escreve 10b na posicao 3 e 2
STOSB	

Obs.

AL	01010101
AND AL, F3H	01010001
OR AL, 00001000B	01011001

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

- Exercício: Desenhar um X na tela, escrevendo direto na memória de vídeo

TITLE Desenha um X na tela, escrevendo direto na memória de vídeo

.MODEL SMALL

MOSTRA_PIXEL MACRO

MOV AL,ES:[DI]

; move byte para AL

AND AL,11110011B

; limpa o bit de posicao

OR AL,1000B

; escreve 10b na posicao 3 e 2

STOSB

; carrega na memoria

ENDM

.STACK 100H

.CODE

MAIN PROC

MOV AH,0

MOV AL,4h

; seleciona modo 4

INT 10h

MOV AX,0B800H

; segmento memoria video

MOV ES,AX

; ES aponta para a mem. video

MOV CX,0

; inicializa CX

MOV DI,0000H

; inicializa DI

MOV SI,2000H

; inicializa SI

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

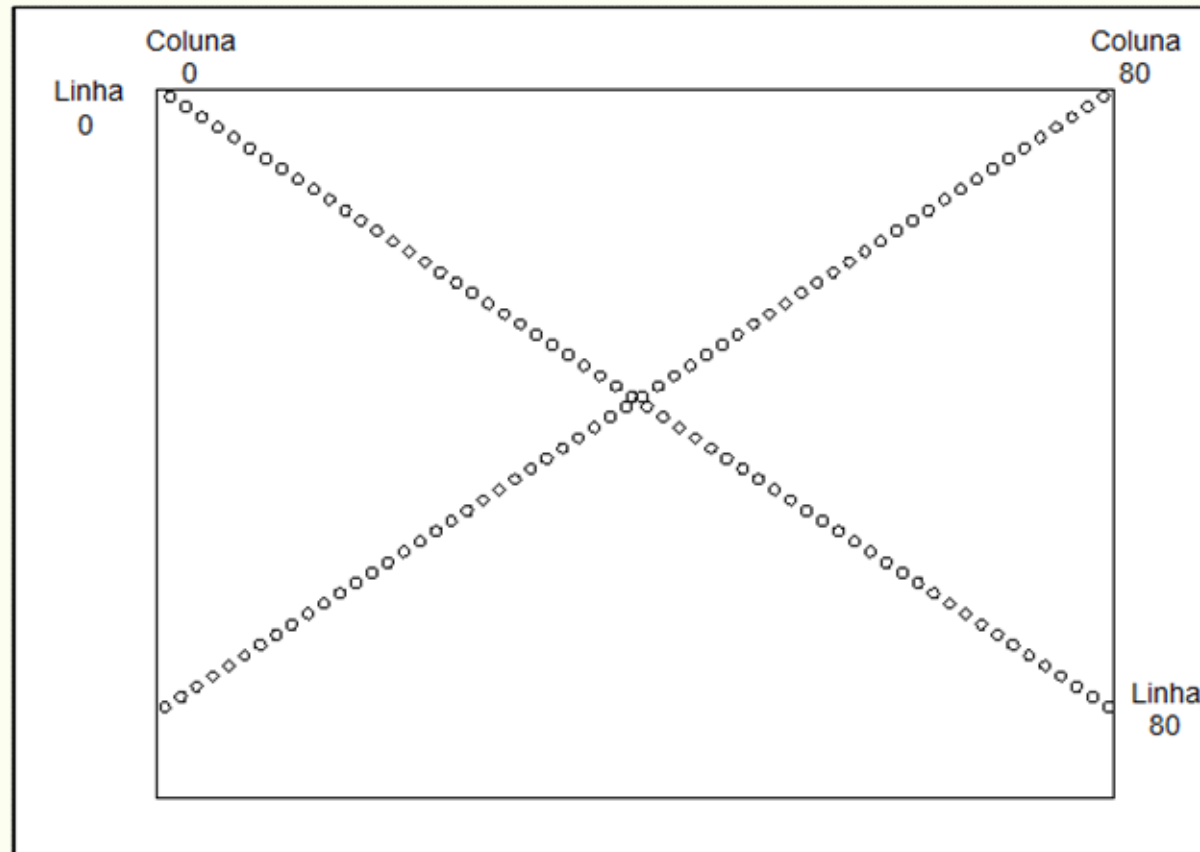
Continuação

```
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
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Resultado



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

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

- **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

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

- **EXEMPLO:** Escrever a letra A em vermelho no canto direito superior da tela. Use modo 4 e cor de fundo cinza

```
MOV AH,0           ; SELECIONA MODO
MOV AL, 04H        ; MODO 4
INT 10H
MOV AH,0BH         ; SELECIONA
MOV BH, 00H        ; COR DE FUNDO
MOV BL,7           ; cinza
INT 10H
MOV AH,02          ; POSICIONA CURSOS
MOV BH,0           ; PAGINA 0
MOV DH,0           ; LINHA 0
MOV DL, 39         ; COLUNA 39
INT 10H
MOV AH,9           ;ESCREVE CHARACTER
MOV AL,'A'         ; CHARACTER A
MOV BL,2           ; COR VERMELHA
MOV CX,1           ; UMA VEZ
INT 10H
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

EXEMPLO – DESENHAR LINHA USANDO INTERRUPÇÃO

TITLE 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

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

- **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:** Escreva um programa que mostre as 256 cores default como 256 pixels na linha 100

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

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, sub função 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