

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Vetores e modos de endereçamento

- Vetores unidimensionais
 - Vetores unidimensional: é uma lista de elementos do mesmo tipo.
 - Por ordem entende-se que há um primeiro, um segundo, um terceiro elemento, e assim por diante até um último elemento;
 - Um vetor *A* de seis elementos é denotado por:

A[0], A[1], A[2], A[3], A[4], A[5]

- Exemplos de vetores:

MSG DB 'abcde' ;vetor composto por um *string* de 5 caracteres ASCII

W DW 1010h,1020h,1030h ;vetor de 3 valores de 16 bits

- Onde *W* se situa na memória a partir do *offset* de endereço 0200h como:

	Offset de endereço	Endereço simbólico	Conteúdo
W →	0200h	W	10h
	0201h		10h
	0202h	W+2	20h
	0203h		10h
	0204h	W+4	30h
	0205h		10h

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Vetores e modos de endereçamento

- Operador DUP: define vetores com valores repetidos (duplicatas)

GAMA	DB	100 DUP (0)	;cria um <i>vetor</i> de 100 <i>bytes</i> ;com valor inicial zero, a partir ;do <i>offset</i> definido por GAMA
BETA	DW	200 DUP (?)	;cria um <i>vetor</i> de 200 <i>palavras</i> ; (16 bits) não inicializados, a ; partir do <i>offset</i> BETA

LINHA	DB	5, 4, 3 DUP (2, 3 DUP (0), 1)	;DUP's encadeados
-------	----	-------------------------------	-------------------

- Equivalente a definição de LINHA dada abaixo:

LINHA	DB	5, 4, 2, 0, 0, 0, 1, 2, 0, 0, 0, 1, 2, 0, 0, 0, 1
-------	----	---

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Modos de endereçamento

- A forma em que um operando é especificado numa instrução é conhecido como Modo de Endereçamento.
- Modos de endereçamento:
 1. Modo por Registrador: o operando está em um registrador da CPU;
 2. Modo Imediato: o operando é uma constante fornecida imediatamente;
 3. Modo Direto: o operando é uma variável declarada no .DATA, ou seja uma posição de memória com endereço bem determinado;
 4. Modo Indireto por Registrador;
 5. Modo por Base;
 6. Modo Indexado;
 7. Modo por Base Indexado;

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Modos de endereçamento

1. **Modo Registrador:** o operando está em registrador(es) da CPU.
 - Exemplo: **ADD AX,BX**
2. **Modo Imediato:** o operando é uma constante fornecida imediatamente na própria instrução.
 - Exemplo: **MOV AX,5**
3. **Modo Direto:** o operando é uma variável declarada no **.DATA**, ou seja uma posição de memória com endereço bem determinado.

- Exemplo

```
DADO DB 5
```

```
.....
```

```
MOV AL, DADO
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Modos de endereçamento

4. **Modo Indireto por Registrador:** O *offset* do endereço do operando está armazenado num registrador.
- O registrador especificado atua como ponteiro para a posição de memória
 - Formato do operando: [registrador]
 - Registradores utilizados:
 - BX, SI, DI juntamente com o registrador de segmento DS o endereço é formado por DS:[registrador]
MOV BX, OFFSET DADO
MOV AL,[BX]
 - BP juntamente com o registrador de segmento SS o endereço é formado por SS:[BP]
 - Exemplo : Escreva um programa que acumule em AX a soma dos 10 elementos do vetor LISTA abaixo e depois imprima a soma:

LISTA DW 0,1,2,1,1,0,1,1,1,1

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

```
TITLE vetor versao 0
.MODEL SMALL
.DATA

    NSOMA DB 13,10,'A SOMA = ', '$'
    LISTA DW 0,1,2,1,1,0,1,1,1,1
.CODE
MAIN PROC
    MOV AX,@DATA
    MOV DS,AX

    XOR AX,AX      ;inicializa AX com zero
    LEA BX,LISTA    ;SI recebe o offset de end. de LISTA
    MOV CX, 10      ;contador inicializado no. de elementos
SOMA:
    ADD AX,[BX]     ;acumula AX com o elemento de LISTA apontado por SI
    ADD BX,2        ;movimenta o ponteiro para o pr?ximo
    LOOP SOMA       ;faz o laco até CX = 0
    LEA DX, NSOMA
    MOV AH,09
    INT 21H
    MOV DX,AX
    OR DL,30H       ; numero em caractere
    MOV AH,2        ; imprime a soma
    INT 21H
    MOV AH,4CH      ; termina o programa
    INT 21H
MAIN ENDP
END MAIN
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

```
TITLE vetor versão 1
; podemos usar DI no lugar de SI
.MODEL SMALL
.DATA
    NSOMA DB 13,10,'A SOMA = ', '$'
    LISTA DW 0,1,2,1,1,0,1,1,1,1
.CODE
MAIN PROC
    MOV AX,@DATA
    MOV DS, AX

    XOR AX,AX        ;inicializa AX com zero
    LEA SI,LISTA      ;SI recebe o offset de end. de LISTA
    MOV CX, 10        ;contador inicializado no. de elementos
SOMA:
    ADD AX,[SI]       ;acumula AX com o elemento de LISTA apontado por SI
    ADD SI,2          ;movimenta o ponteiro para o pr?ximo
    LOOP SOMA         ;faz o laço até CX = 0

    LEA DX, NSOMA
    MOV AH,09
    INT 21H
    MOV DX,AX
    OR DL,30H
    MOV AH,2
    INT 21H
    MOV AH,4CH
    INT 21H
MAIN ENDP
END MAIN
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Modos de endereçamento

5. Modo por Base (*Based Mode*)

- O *offset* do endereço do operando é obtido adicionando um deslocamento ao conteúdo de um registrador base
- O deslocamento pode ser:
 - O *offset* de endereço de uma variável;
 - Uma constante (positiva ou negativa);
 - O *offset* de endereço de uma variável mais ou menos uma constante;
- Formatos possíveis do operando:
 - **[registrador + deslocamento]** ou [deslocamento + registrador]
 - [registrador] + deslocamento ou deslocamento + [registrador]
 - **deslocamento [registrador]**
- Registradores utilizados:
 - BX (*base register*) juntamente com o registrador de segmento DS
 - BP (*base pointer*) juntamente com o registrador de segmento SS
- Exemplo: Supondo que BX contendo o valor 4d:

```
LISTA    DW    10,20,30,40,50,60,70,80,90,100
MOV AX, LISTA[BX]
MOV AX, [LISTA + BX]      ;resulta AX = 30
```


ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Modos de endereçamento

5.. Modo por Base (*Based Mode*)

- Exemplo : Escreva um trecho de programa que acumule em AX a soma dos 10 elementos do *vetor* LISTA abaixo, usando endereçamento por Base:

LISTA DW 0,1,2,1,1,0,1,1,1,1

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

TITLE vetor versao 3

.MODEL SMALL

.DATA

NSOMA DB 13,10,'A SOMA = ', '\$'

LISTA DW 0,1,2,1,1,0,1,1,1,1

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

XOR AX,AX ;inicializa AX com zero

XOR BX,BX ;limpa o registrador base

MOV CX, 10 ;contador inicializado no. de elementos

SOMA:

ADD AX,LISTA+[BX] ;acumula AX com o elemento de LISTA apontado LISTA + BX

ADD BX,2 ;movimenta o ponteiro para o pr?ximo

LOOP SOMA ;faz o laCo atE CX = 0

LEA DX, NSOMA

LEA DX, NSOMA ; imprime mensagem NSOMA

MOV AH,09

INT 21H

MOV DX,AX

OR DL,30H ; número em caractere

MOV AH,2 ; imprime a soma

INT 21H

MOV AH,4CH ; termina o programa

INT 21H

MAIN ENDP

END MAIN

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Modos de endereçamento

6. Modo Indexado (*Indexed Mode*)

- O *offset* do endereço do operando é obtido adicionando um deslocamento ao conteúdo de um registrador indexador
- As opções de deslocamento são as mesmas do Modo por Base
- Formatos possíveis do operando:

[registrador + deslocamento] ou [deslocamento + registrador]
[registrador] + deslocamento ou deslocamento + [registrador]
deslocamento [registrador]

- Registradores utilizados:
 - SI e DI juntamente com o registrador de segmento DS

Exemplo: Supondo que SI contenha o *offset* de endereço de LISTA:

LISTA	DW	10,20,30,40,50,60,70,80,90,100
LEA SI,LISTA		;SI recebe o <i>offset</i> de LISTA
MOV AX, [SI + 12]		;resulta que AX = 70

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Modos de endereçamento

6. Modo Indexado (*Indexed Mode*)

- Exemplo : Escreva um trecho de programa que acumule em AX a soma dos 10 elementos do vetor LISTA abaixo, usando endereçamento indexado:

LISTA DW 0,1,2,1,1,0,1,1,1,1

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

TITLE Vetor versão 2

.MODEL SMALL

.DATA

NSOMA DB 13,10,'A SOMA = ', '\$'

LISTA DW 0,1,2,1,1,0,1,1,1,1

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

XOR AX,AX ;inicializa AX com zero

XOR SI,SI

MOV CX, 10 ;contador inicializado no. de elementos

SOMA:

ADD AX,LISTA[SI] ;acumula AX com o elemento de LISTA apontado por SI

ADD SI,2 ;movimenta o ponteiro para o próximo

LOOP SOMA ;faz o laço até CX = 0

LEA DX, NSOMA ; imprime mensagem NSOMA

MOV AH,09

INT 21H

MOV DX,AX

OR DL,30H ; número em caractere

MOV AH,2 ; imprime a soma

INT 21H

MOV AH,4CH ; termina o programa

INT 21H

MAIN ENDP

END MAIN

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Modos de endereçamento

7. Modo por Base Indexado (*Based Indexed Mode*)

- O *offset* do endereço do operando é obtido somando:
 - o conteúdo de um registrador de base (BX ou BP);
 - o conteúdo de um registrador índice (SI ou DI);
 - opcionalmente, o *offset* de endereço de uma variável;
 - opcionalmente, uma constante (positiva ou negativa).
- Formatos possíveis do operando:
 - variável [reg_de_base] [reg_índice]
 - variável [reg_de_base + reg_índice + constante]
 - constante [reg_de_base + reg_índice + variável]
 - [reg_de_base + reg_índice + variável + constante]
- Registradores utilizados:
 - SI, DI e BX juntamente com o registrador de segmento DS
 - SI, DI e BP juntamente com o registrador de segmento SS
- Exemplo1: Supondo a variável LISTA abaixo, e que SI = 14 e BX = 2:

LISTA	DW	10,20,30,40,50,60,70,80,90,100
MOV AX, LISTA[BX][SI]		;resulta AX = [LISTA+2+14] = 90

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Modos de endereçamento

7. Modo por Base Indexado (*Based Indexed Mode*)

- Exemplo2: Supondo a variável MATRIZ abaixo:

MATRIZ	DW	10,20,30,40
	DW	50,60,70,80
	DW	90,100,110,120
	DW	130,140,150,160

```
XOR BX,BX
MOV SI,4
MOV AX, MATRIZ[BX][SI]      ;resulta AX = 30
```

```
MOV BX,8
MOV SI,4
MOV AX, MATRIZ[BX][SI]      ;resulta AX = 70
```

```
MOV BX,16
MOV SI,0
MOV AX, MATRIZ[BX][SI]      ;resulta AX = 90
```

```
MOV BX,24
MOV SI,6
MOV AX, MATRIZ[BX][SI]      ;resulta AX = 160
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Modos de endereçamento

7. Modo por Base Indexado (*Based Indexed Mode*)

Exemplo: Suponha que MATRIZ seja uma matriz 3X4 com elementos de tipo DW. Escreva um trecho de programa que zere os elementos da 3ª. linha de MATRIZ. O programa deve imprimir a matriz, como uma matriz.

- DATA

MATRIZ	DW 1,2,3,4
	DW 5,6,7,8
	DW 9,10,11,12

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

```
TITLE Matriz
.MODEL SMALL
.DATA
```

```
MATRIZ DW 1,2,3,4
        DW 5,6,7,8
        DW 9,10,11,12
```

```
NMATRIZ DB 13,10,'A MATRIZ = ', 13,10, '$'
PULA_LINHA DB, 13, 10, '$'
```

```
.CODE
MAIN PROC
MOV AX,@DATA
MOV DS,AX
```

```
XOR AX,AX      ;inicializa AX com zero
XOR SI,SI      ;inicializa o indexador de coluna
MOV BX,16      ;BX indica o 1o. elemento da linha 3
MOV CX,4       ;CX contem o número de elementos de
                linha
```

```
L1:  MOV MATRIZ [BX][SI], 0    ;carrega na linha 3
      ADD SI,2                ;incrementa 2 em SI -> tipo DW = 2
                                bytes
      LOOP L1                 ;faz o laço até que CX seja zero
```

```
LEA DX, NMATRIZ ; IMPRIME MENSAGEM EM NMATRIZ
MOV AH,09
INT 21H
```

```
MOV BX,0      ;inicializa o indexador de coluna
MOV DI,3      ;BX indica o 1o. elemento da linha 2
              ; número de LINHAS
```

L2:

```
XOR SI,SI
MOV CX,4      ;CX contem o número de elementos de
                linha
```

```
L3:  MOV DX, MATRIZ [BX][SI] ;carrega zero no operando
      calculado
```

```
OR DL,30H     ; nUmero em caractere
MOV AH,2      ; imprime a soma
```

```
INT 21H
```

```
ADD SI,2      ;incrementa 2 em SI -> tipo DW = 2 bytes
```

```
LOOP L3
```

```
LEA DX, PULA_LINHA
```

```
MOV AH,09H
```

```
INT 21H
```

```
ADD BX,8
```

```
DEC DI
```

```
JNZ L2
```

```
MOV AH,4CH    ; termina o programa
```

```
INT 21H
```

```
MAIN ENDP
```

```
END MAIN
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Modos de endereçamento

- **Observação:** Acessando a pilha por meio de endereçamento por base:
 - SP sempre aponta para o topo da pilha
 - Problema: como obter cópias de dados existentes na pilha sem modificá-la
 - Solução: endereçamento por base usando BP (*base pointer*)
- **Exemplo:** Escreva um trecho de programa que carregue AX, BX e CX com as três palavras mais superiores da pilha, sem modificá-la.

```
MOV BP,SP      ;BP aponta para o topo da pilha
MOV AX,[BP]     ;coloca o conteúdo do topo em AX
MOV BX,[BP+2]   ;coloca a 2a. palavra (abaixo do topo) em BX
MOV CX,[BP+4]   ;coloca a 3a. palavra em CX
```
- Neste caso, no Modo de endereçamento por Base:
 - BP pode especificar o offset de um endereço na pilha;
 - Pode ser somado um deslocamento (positivo ou negativo);
 - O operando final especificado pode navegar para dentro da pilha;

Segment Override

- Se for necessário acessar dados em outro segmento diferente de DS, por exemplo ES:

```
MOV AX, ES:[SI]
```
- Pode-se utilizar *segment override* nos modos de endereçamento por registrador indireto, por base e indexado.

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Modos de endereçamento

- Algumas diretivas

PTR

- A instrução **MOV [BX],1** é ilegal, pois o Montador não pode determinar se [BX] aponta para uma informação na memória do tipo byte ou do tipo word.
- Soluções:

MOV BYTE PTR [BX],1 ;define o destino como byte

MOV WORD PTR [BX],1 ;define o destino como word

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

Modos de endereçamento

- Algumas diretivas

LABEL

- É uma pseudo-instrução para alterar tipo de variáveis (*override*)

- Exemplo:

TEMPO	LABEL	WORD
HORAS	DB	10
MINUTOS	DB	20

- Esta declaração feita no segmento de dados (.DATA), permite que:

- TEMPO e HORAS recebam o mesmo endereço pelo Montador;
 - TEMPO (16 bits) engloba HORAS e MINUTOS (8 + 8 bits);
 - são legais as seguintes instruções:

```
MOV AH,HORAS
MOV AL,MINUTOS
```

e

```
MOV AX,TEMPO ;produz o mesmo efeito das acima
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

vetores bidimensionais

- Um vetor bidimensional, ou matriz é um vetor de vetores.
- Arranjo idealizado em linhas e colunas:
 - Matriz A de 3 linhas por 4 colunas;
 - Cada elemento é identificado como $A[i,j]$;
 - i varia de 1 a 3 e j varia de 1 a 4.
- A memória se organiza apenas em uma dimensão (arranjo linear)
- Como organizar os dados?
 - ordem por linha: os dados da linha 1 são organizados em ordem crescente, seguidos dos dados da linha 2, sucessivamente, até a linha N:
 - ex: $A[1,1]$ a $A[1,4]$, $A[2,1]$ a $A[2,4]$, $A[3,1]$ a $A[3,4]$ → método mais usual
 - ordem por coluna: idem, organizando-se por colunas:
 - ex: $A[1,1]$ a $A[3,1]$, $A[1,2]$ a $A[3,2]$, $A[1,3]$ a $A[3,3]$, $A[1,4]$ a $A[3,4]$

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

vetores bidimensionais

- Exemplo: Supondo a matriz A (3 x 4) abaixo, organizar os dados no segmento de dados em ordem por linha e ordem por coluna.

	1	2	3	4
1	10	20	30	40
2	50	60	70	80
3	90	100	110	120

organização por linha:

```
...  
.DATA  
A                DW    10,20,30,40  
                  DW    50,60,70,80  
                  DW    90,100,110,120
```

organização por coluna:

```
...  
.DATA  
A                DW    10,50,90  
                  DW    20,60,100  
                  DW    30,70,110  
                  DW    40,80,120
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

A instrução XLAT

- Instrução sem operando que converte um valor binário de tipo *byte* em outro valor binário, que é procurado numa tabela de conversão. O valor a ser convertido (1 byte) é assumido estar em AL
- O *offset* do endereço onde se inicia a tabela de conversão é assumido estar em BX.
- A instrução XLAT:
 - Soma o conteúdo de AL ao *offset* dado por BX e localiza a posição resultante dentro da tabela;
 - Substitui o conteúdo AL pelo valor localizado na tabela.

Exemplo: Conversão do conteúdo binário de 8 bits de AL, suposto ser um valor hexadecimal, para o caracter ASCII correspondente.

.DATA

Tabela	DB	30h,31h,32h,33h,34h,35h,36h,37h,38h,39h
	DB	41h,42h,43h,44h,45h,46h

.CODE

...

MOV AL,0Ch	;exemplo, converter 0Ch para caracter ASCII 'C'
LEA BX,TABELA	;BX recebe o <i>offset</i> de TABELA
XLAT	;é feita a conversão e AL recebe 43h = 'C'