



Programação de Software Básico – Linguagem Assembly - Vetores

MATA49

Prof. Babacar Mane

2021.1 – Aula 3



Programação de Software Básico

Vetores

Conteúdo

- Definição
- Acesso
- Escrita



Programação de Software Básico

Vetores

❑ Definição

- É uma lista em um bloco de memória contínuo
- Cada elemento tem o mesmo tipo, ou seja o mesmo número de bytes
- Podem armazenar qualquer tipo de dado (char, inteiro, float...)



Programação de Software Básico

Vetores

Definição

As “strings” também são vista como vetores

- Vetores de caracteres
- Cada posição do vetor armazena um caractere
- As “strings” são comumente definidas como vetores de bytes



Programação de Software Básico

Vetores

❑ Definição

- Os vetores podem ser definidos com seus valores inicializados ou não-inicializados
- Deve-se declarar os valores de cada elemento (inicializados) ou o número de posições (não inicializados)
 - – Informando qual o tipo (byte, word, double word, quadword...)
- Podem ser declarados nos segmentos .data e .bss

Programação de Software Básico

Vetores

❑ Definição

```
1 segment .data
2 ; define array of 10 double words initialized to 1,2,...,10
3 a1          dd    1, 2, 3, 4, 5, 6, 7, 8, 9, 10
4 ; define array of 10 words initialized to 0
5 a2          dw    0, 0, 0, 0, 0, 0, 0, 0, 0, 0
6 ; same as before using TIMES
7 a3          times 10 dw 0
8 ; define array of bytes with 200 0's and then a 100 1's
9 a4          times 200 db 0
10             times 100 db 1
11
12 segment .bss
13 ; define an array of 10 uninitialized double words
14 a5          resd   10
15 ; define an array of 100 uninitialized words
16 a6          resw   100
```

Programação de Software Básico

Vetores

Exemplos:

Offset	Value
0000:	10
0001:	20
0002:	30
0003:	40

vetor1 db 10,20,30,40

Tipo: Byte

Offset	Value
0000:	1
0002:	2
0004:	3
0006:	4
0008:	5

vetor2 dw 1,2, 3, 4, 5

Tipo: Word

Offset	Value
0000:	1
0004:	2
0008:	3
000C:	4
0010:	5

vetor3 dd 1,2, 3, 4, 5

Tipo: Double word



Programação de Software Básico

Vetores

❑ Acesso

- O acesso aos elementos do vetor pode ser computado a partir de três dados:
 - – O endereço da primeira posição do vetor
 - – O número de bytes de cada elemento
 - – O índice do elemento
- Assim com em C, o índice da primeira posição do vetor é convencionalmente 0.

Programação de Software Básico

Vetores

❑ Acesso:

- Acesso não é feito como na maioria das linguagens de alto nível (array1[2], array2[4])
- Acesso:
 - $\text{elemento_array} = [\text{<variavel>} + \text{<numero_bytes>}]$
<variavel>: guarda o endereço inicial do vetor
<numero_bytes>: um inteiro que indica o deslocamento
- Não restringe acesso além do final do vetor

Programação de Software Básico

Vetores

❑ Acesso:

32-bit Words	64-bit Words	Bytes	Ender.
Addr = 0000	Addr = 0000		0000
			0001
			0002
			0003
Addr = 0004	Addr = 0008		0004
			0005
			0006
			0007
Addr = 0008	Addr = 0008		0008
			0009
			0010
			0011
Addr = 0012			0012
			0013
			0014
			0015

Programação de Software Básico

Vetores

❑ **Acesso:** - Considere os vetores abaixo

```
array1    db    5, 4, 3, 2, 1    ;array de bytes
```

```
array2    dw    5, 4, 3, 2, 1    ;array de words
```

```
mov       al, [array1]           ; al = array1[0]
```

```
mov       al, [array1 + 1]       ; al = array1[1]
```

```
mov       [array1 + 3], al       ; array1[3] = al
```

```
mov       ax, [array2]           ; ax = array2[0]
```

```
mov       ax, [array2 + 2]       ; ax = array2[1] (NOT array2[2] !)
```

```
mov       [array2 + 6], ax       ; array2[3] = ax
```

```
mov       ax, [array2 + 1]       ; ax = ??
```

Programação de Software Básico

Vetores

❑ Escrita: - Exemplo

```
section .bss
    vetor resd 30
section .text
    mov ebx, 0
    mov ecx, 30
lp:
    call read_int
    mov [vetor+ebx], eax
    add ebx, 4
loop lp
```

Programação de Software Básico

Vetores

- section .bss

buffer:	resb	64	;reserve 64 bytes
wordvar:	resw	1	;reserve a word
Realarray:	resq	10	;array of ten reals

Programação de Software Básico

Vetores

- **Exercício:**

- **Escreva um programa em assembly que inverte os dados armazenados no vetor1 em um segundo vetor (vetor2 não inicializado)**

Ex: vet1 = [1, 2, 3, 4, 5] → vet2= [5,4,3,2,1]

vetor1 dw 1, 2, 3, 4, 5

vetor2 resw 5



Programação de Software Básico – (operações aritméticas) CBW, CWD, ou CDQ



MATA49

Prof. Babacar Mane

2021.1 – Aula extra



Programação de Software Básico (operações aritméticas)

- São instruções para estender o sinal de números inteiros
- – **CBW (convert byte to word)**
 - ● Estende o sinal do AL para o AH
 - ● Exemplo:
 - MOV AL, 9Bh ; -101
 - CBW ;AH = FF9Bh = -101
- – **CWD (convert word to double word)**
 - ● Estende o sinal de AX para DX
 - ● Exemplo: MOV AX, FF9Bh
 - CWD ;DX:AX = FFFFFFFF9Bh



Programação de Software Básico (operações aritméticas)

- São instruções para estender o sinal de números inteiros
- – CDQ (convert doubleword to quadword)
 - ● Estende o sinal do EAX para o EDX
 - ● Exemplo:
 - MOV EAX, FFFFFFFF9Bh
 - CDQ ;EDX:EAX = FFFFFFFFFFFFFFFF9Bh

Programação de Software Básico (operações aritméticas)

- **IDIV**

- Para execução do IDIV é necessário estender o sinal do dividendo para conservar o sinal no resto

- Usaremos CBW, CWD, ou CDQ

- **Exemplo:**

MOV AL, -48

CBW ;Sinal estendido em AH

MOV BL, 5

IDIV BL ; AL = - 9 , AH = -3 o resto



Programação de Software Básico (operações aritméticas)

- **IDIV**

- **Exemplo:**

MOV AX, -5000

CWD ;estende o sinal para DX

MOV BX, 256

IDIV BX ;AX= -19 (DX=-136 resto)



Programação de Software Básico (operações aritméticas)

Exercícios:

a) `var4 = (var1 * 5) / (var2 - 3);`

b) `var4 = (var1 * -5) / (-var2 % var3);`



Programação de Software Básico (operações aritméticas)

Exercícios: $\text{var4} = (\text{var1} * 5) / (\text{var2} - 3)$

MOV EAX, [var1]

MOV EBX, 5

IMUL EBX

MOV EBX, [var2]

SUB EBX, 3

CDQ

IDIV EBX

MOV [var4], EAX

Programação de Software Básico (operações aritméticas)

Exercícios: $\text{var4} = (\text{var1} * -5) / (-\text{var2} \% \text{var3})$

MOV EAX, [var2]

NEG EAX

CDQ

IDIV [var3]

MOV EBX, EDX

MOV EAX, -5

IMUL [var1]

CDQ

IDIV EBX

MOV [var4], EAX



Programação de Software Básico – (Registradores)

MATA49

Prof. Babacar Mane

2021.1 – Aula extra



Programação de Software Básico (Registradores)

- **3 categorias de registradores:**
 - Registradores de uso gerais
 - Registradores de controle
 - Registradores de segmento

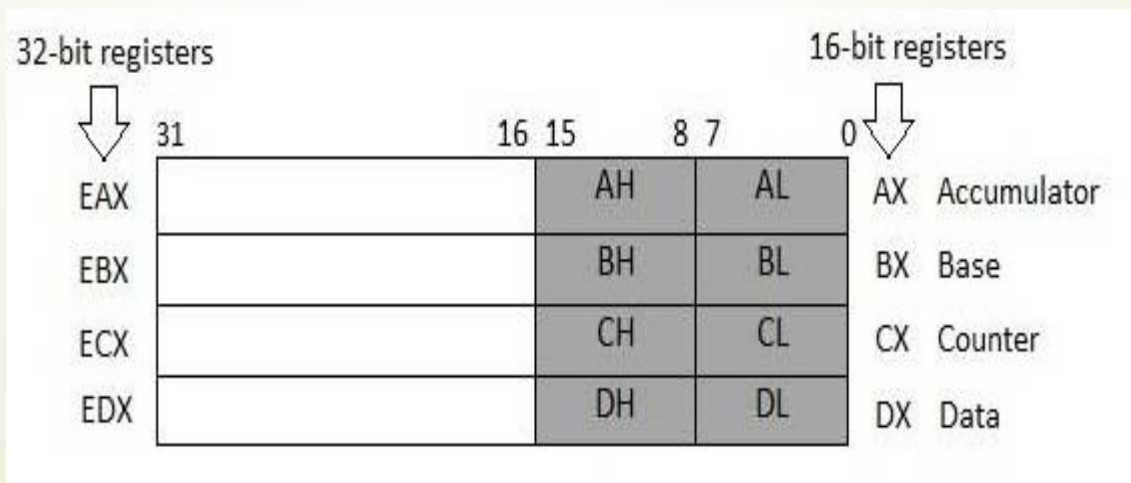


Programação de Software Básico (Registradores)

- **3 categorias de registradores:**
 - **Registradores de uso gerais**
 - **Registradores de dados (operações aritméticas e lógicas)**
 - **Registradores de base**
 - **Registradores de Índice**

Programação de Software Básico (Registradores)

- 3 categorias de registradores:
 - Registradores de uso gerais
 - Registradores de dados (operações aritméticas e lógicas)



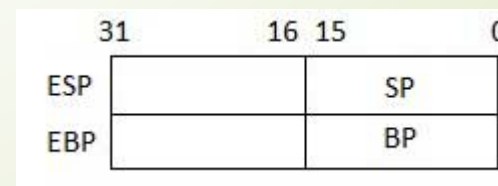


Programação de Software Básico (Registradores)

- 3 categorias de registradores:
 - Registradores de uso gerais
 - Registradores de dados (operações aritméticas e lógicas)
 - AX: Acumulador (Input/output e operações aritméticas como multiplicação e divisão – operando fica no EAX, AX ou AL)
 - BX: registrador de base (endereçamento indexado)
 - CX: Contador (ECX, CX) registradores armazenam a contagem de loop em operações iterativas.
 - DX: registrador de dados – I/O operações, operações de multiplicação e divisão, junto com AX

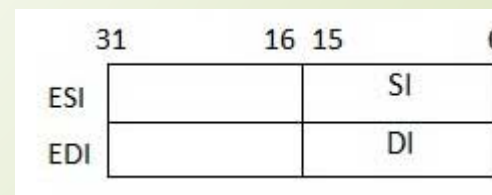
Programação de Software Básico (Registradores)

- 3 categorias de registradores:
 - Registradores de uso gerais
 - Registradores de base (EIP, ESP, EBP 32 bits)
 - Ponteiro de instrução (IP) ou Contador de programa (PC)
CS:IP - fornece o endereço completo da instrução atual no segmento de código
 - SP : Ponteiro de pilha
 - BP: Ponteiro Base



Programação de Software Básico (Registradores)

- 3 categorias de registradores:
 - Registradores de uso gerais
 - Registradores de Índice - ESI e EDI (endereçamento indexado e às vezes usado em adição e subtração)
 - Source Index (SI) – Índice Fonte (operações sobre “strings”)
 - Destination Index (DI) – Índice destino (operações sobre “strings”)





Programação de Software Básico (Registradores)

- **3 categorias de registradores:**
 - **Registradores de controle: Eflags**
 - **Registradores de Segmento:** são áreas específicas definidas em um programa para conter dados, código e pilha
 - CS (Segmento de Códigos): armazena o endereço inicial do segmento de código
 - DS (Segmento de Dados): armazena o endereço inicial do segmento de dados
 - SS (Segmento de pilha): armazena o endereço inicial da pilha
 - ES (Segmento Extra)



Programação de Software Básico – (Modos de endereçamento)



MATA49

Prof. Babacar Mane

2021.1 – Aula extra