

MATA49

Programação de Software Básico

Vetores

Leandro Andrade
leandrojsa@ufba.br

Vetores

- 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...)
- Essas características permitem um acesso eficiente aos elementos do vetor

Vetores

- 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
 - O que caracteriza uma string para um máquina é uma sequência de caracteres, sendo que o último deste é o caractere nulo ('\0' ou o inteiro 0)

Vetores

- 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 convencionalizada como 0.

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

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
```

Vetores: Definição

- 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

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

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 = ??
```

Vetores: Escrita

- Exemplo:

```
.bss
vetor resd 30

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

Vetores

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

Ex: $\text{vet1} = [1, 2, 3, 4, 5] \rightarrow \text{vet2} = [5, 4, 3, 2, 1]$

```
vetor1 dw 1, 2, 3, 4, 5
```

```
vetor2 resw 5
```

Vetores Multidimensionais

- São armazenados na memória como vetores unidimensionais
- Duas dimensões:
 - Em C faríamos: `int a[3][2]`
 - $2 \times 3 = 6$ elementos

Index	0	1	2	3	4	5
Element	<code>a[0][0]</code>	<code>a[0][1]</code>	<code>a[1][0]</code>	<code>a[1][1]</code>	<code>a[2][0]</code>	<code>a[2][1]</code>

Vetores Multidimensionais

- Duas dimensões:
 - Modos de mapear:
 - Rowwise
 - Cada linha é contínua na memória
 - $a[i][j] = N \times i + j$
 - N = número de colunas
 - i = índice da linha ; j = índice da coluna
 - Columnwise
 - Cada coluna é contínua na memória
 - $a[i][j] = i + N \times j$
 - N = número de linhas
 - i = índice da linha ; j = índice da coluna

Vetores Multidimensionais

- Duas dimensões:

- Exercício:

Seja uma matriz 4x3 com seus valores já inicializados através do mapeamento rowwise (variável `matriz`). Os registradores `ebx` e `ecx` guardam respectivamente o índice da linha e da coluna do elemento selecionado da matriz.

Armazene o elemento selecionado na variável `resultado`.

`matriz resd 12`

`resultado resd 1`

Vetores Multidimensionais

- Mais de duas dimensões:
 - Mesmo raciocínio:
 - Para 3 dimensões:
 - Seja $a[L][M][N]$:
 - Então $a[i][j][k] = M \times N \times i + N \times j + k$