

Vetores Multidimensionais

Aula 08

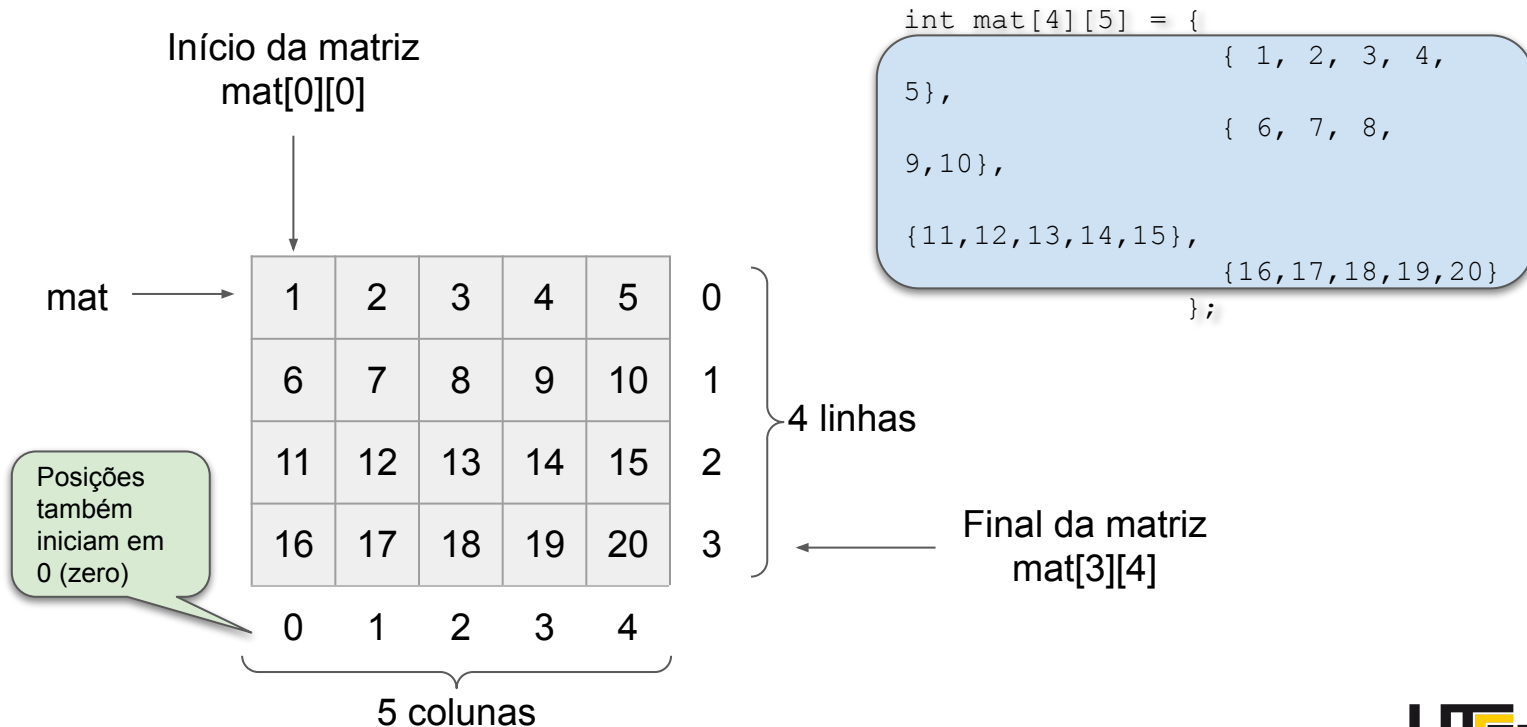
Marcos Silvano Almeida
Departamento de Computação
UTFPR Campo Mourão

Questões sobre vetores/strings/matrizes

- Não é possível **atribuir vetores**
 - Não existe cópia de vetores, apenas **inicialização**
- Não é possível **comparar** dois vetores pelos operadores =,<,<=,>,>=,!=
 - É necessário percorrer os dois vetores e comparar elemento a elemento
- Um vetor não armazena seu tamanho
 - É necessário sempre guardá-lo em uma variável
 - Strings possuem um marcador especial ao final: NULL (0 ou '\0')
- Todo vetor passado como parâmetro à uma função é sempre uma referência (ponteiro)
 - Qualquer alteração dentro da função reflete no vetor passado

Matrizes: vetores bidimensionais

- Podemos definir vetores de múltiplas dimensões
- Caso mais comum: vetor bidimensional -> matriz



Inicialização e acesso

- Na declaração da matriz
 - Primeira dimensão pode ser omitida (demais devem ser explícitas)

```
void main() {  
    printf("MATRIZES E ELEMENTOS\n");  
  
    int mat[ ][5] = { { 1, 2, 3, 4, 5}, // <- 4 x 5 elementos  
                     { 6, 7, 8, 9,10},  
                     {11,12,13,14,15},  
                     {16,17,18,19,20}  
                   };  
  
    int lin = 4;  
    int col = 5;  
    printf("primeiro elemento: %d\n", mat[0][0]);  
    printf("ultimo elemento: %d\n", mat[lin-1][col-1]);  
}
```

Inicialização

- Não é permitido declarar matrizes de tamanho indefinido com inicialização.

// PROBLEMA!

```
int rows = 4;
```

```
int cols = 5;
```

```
int mat[rows][cols] = {  
    { 1, 2, 3, 4, 5},  
    { 6, 7, 8, 9,10},  
    {11,12,13,14,15},  
    {16,17,18,19,20}  
};
```

// OK!

```
int rows = 4;
```

```
int cols = 5;
```

```
int mat[rows][cols];
```

Pode declarar matriz de tamanho indefinido, desde que não haja inicialização.

Impressão

```
void main() {  
    int mat[4][5] = {  
        { 1, 2, 3, 4, 5},  
        { 6, 7, 8, 9,10},  
        {11,12,13,14,15},  
        {16,17,18,19,20}  
    };  
  
    // percorre e imprime matriz  
    for (int i = 0; i < 4; i++) {  
        for (int j = 0; j < 5; j++) {  
            printf(" %2d", mat[i][j]);  
        }  
        printf("\n");  
    }  
}
```

Uma matriz é um **vetor de vetores**:
Cada elemento do primeiro vetor é
outro vetor.

Passando matriz para função

- Ordem dos parâmetros: dimensões (linhas e colunas) + matriz

```
void printMatrix(int rows, int cols, int m[rows][cols]) {  
    for (int i = 0; i < rows; i++) {  
        for (int j = 0; j < cols; j++) {  
            printf(" %2d", m[i][j]);  
        }  
        printf("\n");  
    }  
}  
  
void main() {  
    int mat[2][5] = {  
        { 1, 2, 3, 4, 5},  
        {11,12,13,14,15}  
    };  
  
    printMatrix(4, 5, mat);  
}
```