

25. Matrices

DIM0321

2015.1

Sumário

1 Introdução

2 Uso

1 Introdução

2 Uso

Motivação

Muitos problemas computacionais precisam representar uma malha multidimensional de dados:

- Jogos (tabuleiro, simulação)
- Desenho auxiliado por computador (CAD)
- Plotagem de gráficos
- Sistemas de equações (matrizes, álgebra linear)
- Simulação de reservatórios, da atmosfera, dos oceanos
- Simulação de fluidos

Arranjos multidimensionais (ou **matrizes**) = armazenar conjuntos de dados **homogêneos** cujos elementos necessitam ser endereçados por mais de um índice.

Matrizes

Definição

- matriz 2D : arranjo de arranjos
- matriz 3D : arranjo de arranjos de arranjos

Alinhamento de dados

- Em memória, a matriz é alocada como um arranjos de linha.
- A matriz $m = \{\{1, 2\}, \{3, 4\}\}$ fica como:

1 2 3 4

Matriz 2D

Sintaxe

```
1 tipo nome[X] [Y];  
2 tipo nome[X] [Y] =  
3     {{x00, x01, x02, ....},  
4      {x10, x11, ....},  
5      ...  
6     };
```

Ação

- Essa declaração aloca uma zona de memória de $X * Y * \text{sizeof}(\text{tipo})$
- $v[i][j]$ é o elemento de $i * Y + j$: a alocação é feita linha por linha
 - ▶ em Fortran é coluna por coluna

Matriz 3D

Sintaxe

```
1 tipo nome[X] [Y] [Z];  
2 tipo nome[X] [Y] =  
3   {{{x11, x12, x13, ...},  
4    {}}  
5   { }. {}  
6   }  
7 ;
```

Observação

- Notação `[]` extensível a qualquer número fixo de dimensões.

1 Introdução

2 Uso

Exemplo de uso

```
1 #include <stdio.h>
2 #define MAX_LIN 2
3 #define MAX_COL 3
4
5 int main(void)
6 {
7     int matriz[MAX_LIN][MAX_COL] = {{5, 8, 15}, {42, 11, 33}};
8     int i, j;
9     for (i = 0; i < MAX_LIN; ++i) {
10         for (j = 0; j < MAX_COL; ++j) {
11             printf("%2i", matriz[i][j]);
12             if (j < MAX_COL - 1)
13                 printf(" ");
14             else
15                 printf("\n");
16         }
17     }
18 }
```

Uso como parâmetros de função

- Acessar o elemento $v[2][3]$ de uma matriz de tamanho $4 \times 5 =$ encontrar o elemento no bloco ($2 * 5 + 3 = 13$).
- Precisamos então do tamanho **5**

Usos (in)corretos

- É impossível usar arranjos genéricos,

Errado

```
1 void imprime_matriz(int matriz[] [], int n, int m);  
2 void imprime_matriz(int n, int m, int **matriz);
```

Correto

```
1 void imprime_matriz(int n, int m, int matriz[n][m]);  
2 void imprime_matriz(int n, int m, int matriz[][m]);
```

Observações

- As variáveis de tamanho devem **preceder** a do arranjo
- O tamanho deve ser especificado pela variável do arranjo

Uso

```
1 void soma_matrizes(int N, int M, float matriz1[N][M], float matriz2[][M])
2 {
3     int i, j;
4     for(i = 0; i < N; ++i) {
5         for(j = 0; j < M; ++j) {
6             printf("%.2f", matriz1[i][j] + matriz2[i][j]);
7             if(j < M - 1)
8                 printf(" ");
9             else
10                printf("\n");
11        }
12    }
13 }
```