

## Exercício de Programação CUDA: Multiplicação de Matrizes

**Complexidade: baixa**  
**Categoria: Matrizes**

**Linguagem: C/CUDA**  
**Modelo de Programação: CUDA**

Autores: Guilherme Martins e Paulo Sérgio Lopes de Souza

*Este material pode ser utilizado e modificado desde que os direitos autorais sejam explicitamente mencionados e referenciados. Utilizar considerando a licença GPLv2 (GNU General Public License version 2) ou posterior.*

*Última revisão em Novembro de 2020 por Paulo Sérgio Lopes de Souza*

Considere as seguintes matrizes 3\*3:

A=

1	2	3
4	5	6
7	8	9

B=

9	8	7
6	5	4
3	2	1

O produto de A \* B é:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \times \begin{pmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 \times 9 + 2 \times 6 + 3 \times 3 & 1 \times 8 + 2 \times 5 + 3 \times 2 & 1 \times 7 + 2 \times 4 + 3 \times 1 \\ 4 \times 9 + 5 \times 6 + 6 \times 3 & 4 \times 8 + 5 \times 5 + 6 \times 2 & 4 \times 7 + 5 \times 4 + 6 \times 1 \\ 7 \times 9 + 8 \times 6 + 9 \times 3 & 7 \times 8 + 8 \times 5 + 9 \times 2 & 7 \times 7 + 8 \times 4 + 9 \times 1 \end{pmatrix} = \begin{pmatrix} 30 & 24 & 18 \\ 84 & 69 & 54 \\ 138 & 114 & 90 \end{pmatrix}$$

A matriz  $C = A * B$ , é portanto:

30	24	18
84	69	54
138	114	90

Faça um programa em CUDA C, considerando os conceitos já estudados no curso, para calcular o produto de duas matrizes quadradas do tipo **double**. Utilize as memórias global, compartilhada e local de maneira eficiente na sua solução. Use sincronizações onde necessário.

Considere como entrada um arquivo de texto contendo, na primeira linha, a dimensão *dim* das matrizes. A partir da segunda linha e até a linha *dim+1*, estão os elementos da matriz A, do tipo **double**, onde as linhas são separadas por uma quebra de linha simples e as colunas por um único espaço. A partir da linha  $2*dim+1$  até o fim do arquivo, estão os elementos da matriz B, também do tipo **double** e de mesma dimensão. As matrizes devem ser lidas por meio do redirecionamento de fluxo de entrada (**stdin**), ou seja, não use ponteiros para abrir/fechar/acessar os arquivos.

Conteúdo do arquivo de entrada, por exemplo, **entrada.txt**:

```
3
1.0 2.0 3.0
4.0 5.0 6.0
7.0 8.0 9.0
9.0 8.0 7.0
6.0 5.0 4.0
3.0 2.0 1.0
```

Para executar no **bash**, por exemplo, utilize este padrão:

```
./mm-cuda < entrada.txt <enter>
```

Obs: na linha de comando acima, considera-se que o programa foi inserido em **mm-cuda.cu** e o executável chama-se **mm-cuda** e está no diretório atual.

A saída deve ser impressa, utilizando o *output* (**stdout**) padrão, apenas com os elementos correspondentes a matriz C, resultante após o cálculo do produto das matrizes A e B. Há um espaço a mais no final da linha. Cada métrica é separada por uma quebra de linha simples. Há também uma quebra de linha extra no fim da impressão.

Saída:

**30.0 24.0 18.0**

**84.0 69.0 54.0**

**138.0 114.0 90.0**

### 3. Versão sequencial do problema: um ponto de partida para a solução do exercício

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
int main(int argc,char **argv){
```

```
    //Declara as matrizes
```

```
    double *matrizA,*matrizB,*matrizC;
```

```
    //Declara as variáveis de tamanho e índice
```

```
    int tam,i,j,k;
```

```
    //Lê a dimensão da matriz
```

```
    fscanf(stdin,"%d\n",&tam);
```

```
    //Aloca as matrizes
```

```
    matrizA=(double*)malloc(tam*tam*sizeof(double));
```

```
    matrizB=(double*)malloc(tam*tam*sizeof(double));
```

```
    matrizC=(double*)malloc(tam*tam*sizeof(double));
```

```
    //Lê as matrizes A e B
```

```
    for(i=0;i<tam;i++)
```

```
        for(j=0;j<tam;j++)
```

```
            fscanf(stdin, "%lf ", &matrizA[i * tam + j]);
```

```
    for(i=0;i<tam;i++)
```

```
        for(j=0;j<tam;j++)
```

```
            fscanf(stdin, "%lf ", &matrizB[i*tam+j]);
```

```
    //Calcula C=A*B
```

```
    for(i=0;i<tam;i++)
```

```
for(j=0;j<tam;j++)  
    for(k=0;k<tam;k++)  
        matrizC[i*tam+j]+=matrizA[i*tam+k]*matrizB[k*tam+j];
```

```
//Imprime o resultado
```

```
for(i=0;i<tam;i++){  
    for(j=0;j<tam;j++)  
        printf("%.1lf ",matrizC[i*tam+j]);  
    printf("\n");  
}
```

```
//Desaloca as matrizes
```

```
free(matrizA);  
free(matrizB);  
free(matrizC);
```

```
return 0;
```

```
}
```