PUC-Rio – Departamento de Informática
Ciência da Computação
Introdução à Arquitetura de Computadores
Prof.: Anderson Oliveira da Silva

Trabalho 1

Parte I:
Crie um módulo escrito em linguagem C, chamado *matrix_lib.c*, que implemente duas funções com processamento vetorial para fazer operações aritméticas com matrizes, conforme descrito abaixo.

a. Função *int scalar_matrix_mult(float scalar_value, struct matrix *matrix)*

   Essa função recebe um valor escalar e uma matriz como argumentos de entrada e calcula o produto do valor escalar pela matriz. O resultado da operação deve ser retornado na matriz de entrada. Em caso de sucesso, a função deve retornar o valor 1. Em caso de erro, a função deve retornar 0.

b. Fução *int matrix_matrix_mult(struct matrix *matrixA, struct matrix * matrixB, struct matrix * matrixC)*

   Essa função recebe 3 matrizes como argumentos de entrada e calcula o valor do produto da matriz A pela matriz B. O resultado da operação deve ser retornado na matriz C. Em caso de sucesso, a função deve retornar o valor 1. Em caso de erro, a função deve retornar 0.

O tipo estruturado matrix é definido da seguinte forma:

```
struct matrix {
        unsigned long int height;
        unsigned long int width;
        float *rows;
};
```

Onde:

height = número de linhas da matriz (múltiplo de 8)
width = número de colunas da matriz (múltiplo de 8)
rows = sequência de linhas da matriz (height*width elementos)

Parte II:
Crie um programa em linguagem C, chamado *matrix_lib_test.c*, que implemente um código para testar a biblioteca *matrix_lib.c*. Esse programa deve receber um valor escalar float, a dimensão da primeira matriz, a dimensão da segunda matriz e o nome de quatro arquivos binários de floats na linha de comando de execução. O programa deve carregar as matrizes fornecidas em memória e usar a função *scalar_matrix_mult* com o valor escalar fornecido e com a primeira matriz. O resultado deve ser armazenado com o nome do terceiro arquivo de floats fornecido. Depois, deve usar a função *matrix_matrix_mult* com a matriz resultante da primeira operação e a segunda matriz. O resultado deve ser armazenado com o nome do quarto arquivo de floats fornecido.

Exemplo de linha de comando:
matrix_lib_test 5.0 8 16 16 8 floats_256_2.0f.dat floats_256_5.0f.dat result1.dat result2.dat

Onde,

5.0 é o valor escalar que multiplicará a primeira matriz;
8 é o número de linhas da primeira matriz;
16 é o número de colunas da primeira matriz;
16 é o número de linhas da segunda matriz;
8 é o número de colunas da segunda matriz;
floats_256_2.0f.dat é o nome do arquivo de floats que será usado para carregar a primeira matriz;
floats_256_5.0f.dat é o nome do arquivo de floats que será usado para carregar a segunda matriz;
result1.dat é o nome do arquivo de floats onde o primeiro resultado será armazenado;
result2.dat é o nome do arquivo de floats onde o segundo resultado será armazenado.

Os programas fontes matrix_lib.c, matrix_lib.h e matrix_lib_test.c devem ser enviados como anexo via e-mail com o título INF1029-Gn: Trab1, onde n identifica o número do grupo. Prazo de entrega: 13/9/2019 - 23:59h.