

Laboratório 1

Este laboratório é composto de três exercícios que devem ser realizados nos respectivos arquivos fornecidos. O código deve estar indentado, organizado e comentado. A entrega do laboratório deverá ser feita até o dia 13/11 às 23:59, através do GitHub Classroom.

Exercício 1: Hello World em Thread (3 pontos)

Objetivo: Familiarizar-se com a criação e gerenciamento de threads em C++.

Desenvolvimento do Programa: Crie um arquivo `CMakeLists.txt` e configure-o para suportar a biblioteca de threads. Crie um arquivo `ex1.cpp` e desenvolva um programa que cria 100 threads. Cada thread deve executar uma função que realiza as seguintes ações: - Aguarda por um período de tempo aleatório entre 1 e 20 segundos. - Imprime a mensagem: `Hello world from thread <id da thread>`, onde `<id da thread>` deve ser substituído pelo identificador único da thread em execução.

Dicas: Garanta que o programa principal aguarde a conclusão de todas as threads antes de finalizar.

Entrega: Para este exercício, apenas submeta o código no seu repositório do GitHub Classroom.

Exercício 2: Hello World em Thread Recursivo (2 pontos)

Objetivo: Familiarizar-se com a criação e gerenciamento de threads em C++ utilizando recursão.

Desenvolvimento do Programa: Crie um arquivo `ex2.cpp` e implemente o mesmo programa do Exercício 1, porém criando as threads de maneira recursiva.

Entrega: Para este exercício, apenas submeta o código no seu repositório do GitHub Classroom.

Exercício 3: Multiplicação de Matriz Paralela (5 pontos)

Objetivo: Explorar a paralelização em C++ para acelerar computações e entender os desafios associados.

Contexto: A multiplicação de matrizes é uma operação fundamental em álgebra linear e desempenha um papel crucial em diversas áreas da ciência e engenharia, desde a resolução de sistemas de equações lineares até a transformação de dados em gráficos computacionais e processamento de sinais. Dada sua importância, é essencial compreender o processo e as propriedades dessa operação. Em termos

simples, a multiplicação de matrizes envolve a combinação de linhas de uma matriz com colunas de outra, resultando em uma nova matriz. No entanto, é importante notar que, para que duas matrizes sejam multiplicáveis, o número de colunas da primeira matriz deve ser igual ao número de linhas da segunda matriz. A eficiência e precisão com que essa operação é realizada têm implicações diretas em muitos sistemas e aplicações modernas, tornando-a um tópico de estudo e otimização contínuos.

Desenvolvimento: Configure o CMake para compilar o programa fornecido. Analise o programa sequencial no arquivo `ex3.cpp`, que realiza a multiplicação de matrizes. As suas tarefas para esse laboratório são: - Modificar a multiplicação de matrizes para que seja calculada de forma paralela usando threads. - Medir e exibir os tempos de execução (em milissegundos) das versões serial e paralela. - Gerar conjuntos de dados de entrada com tamanhos crescentes para testar o desempenho do programa. - Calcular e exibir o speedup, que é a relação entre o tempo da versão serial e o tempo da versão paralela. - Realizar uma análise e refletir sobre as mudanças no desempenho à medida que o tamanho da matriz aumenta. Experimente também variar o número de threads e observar como isso afeta o desempenho. - Implementar uma função que compara os resultados da multiplicação de matrizes das versões serial e paralela. Identifique e discuta possíveis discrepâncias nos resultados. Por que elas ocorrem? Como você pode resolvê-las?

Tarefa Opcional: Implemente a multiplicação de matriz em blocos com paralelismo e analise o efeito no desempenho.

Entrega: Para este exercício, além de submeter o código no seu repositório do GitHub Classroom, também submeta na tarefa do Google Classroom um relatório com as análises pedidas.