

Tarefa 2 – Threads

Objetivo: Avaliar como se comporta o tempo de execução de um algoritmo com e sem concorrência entre os algoritmos (sequencial x paralelo), usando threads

Inicialmente faça um programa sequencial para calcular a soma de dois vetores (vetA e vetB) de 1.000 posições (testar também com vetores de 10.000 posições), armazenando o resultado em um terceiro vetor (vetC), tomando o tempo de execução do algoritmo de soma.

Em seguida paralelize o algoritmo de soma utilizando 8 threads. O vetor vetA deve ser inicializado com o valor=1 em todos os elementos e vetB com valor=2. Cada processo vai executar a soma em 125 (ou 1.250) posições dos vetores, de forma a não usarem a mesma posição de memória. Tome o tempo decorrido na execução do algoritmo de soma.

Compare o tempo de execução dos algoritmos sequencial e paralelo. Para tomar o tempo de execução dos algoritmos é recomendável que seja utilizada a função `gettimeofday()`.

Compare ainda os tempos obtidos no trabalho 1 (com processos e memória compartilhada).

Elabore um relatório contendo o código fonte do programa, apresentando os resultados obtidos e a discussão dos resultados. Indique ainda o que está e o que não está funcionando no seu programa.

A tarefa pode ser feita em dupla. Você deve fazer upload do relatório (.pdf) e do código fonte (.c) no site do EAD. Indique claramente no relatório e no código fonte os nomes e matrículas dos alunos que compõem o grupo.