

Instruções

1. Esta avaliação deve ser feita individualmente, em dupla ou trio.
2. Data de entrega: **14/04/2021 até 18:50**. Serão aceitos trabalhos entregues em atraso, com desconto de 2,0 ponto por dia de atraso, incluindo final de semana.
3. Esta avaliação tem por objetivo consolidar o aprendizado sobre conceitos de processos, threads e paralelismo.
4. A implementação deverá ser desenvolvida utilizando Pthreads e OpenMP. Poderá ser usado o equivalente no Windows. A linguagem Python é permitida desde que seja usado corretamente o conceito de threads e configurado e usado corretamente a biblioteca OpenMP.
5. O sistema deve ser entregue funcionando corretamente.
6. Deve ser apresentado um relatório eletrônico em formato PDF (em outro formato é descontado 1,5 ponto) que contenha:
 - a. Identificação do autor e do trabalho.
 - b. Enunciado do projeto.
 - c. Explicação e contexto da aplicação para compreensão do problema tratado pela solução.
 - d. Resultados obtidos com as simulações.
 - e. Códigos importantes da implementação.
 - f. Resultados obtidos com a implementação (tabelas, gráficos e etc).
 - g. Análise e discussão sobre os resultados finais.
7. Deve ser disponibilizado os códigos da implementação juntamente com o relatório (salvo o caso da disponibilidade em repositório aberto do aluno, que deve ser fornecido o link).

Descrição do projeto a ser desenvolvido

Projeto 1

Realize uma implementação em **C/C++ (ou Python)** de uma multiplicação entre matrizes utilizando o sistema single thread e multithread, no qual o último deve ser feito usando as bibliotecas **Pthread e OpenMP**. Realize uma análise comparativa no quesito tempo de processamento utilizando bibliotecas como **time.h (ou biblioteca equivalente)**. A operação de multiplicação deve ser entre, no mínimo, uma matriz 2X3 e outra 3x2, como no exemplo apresentado:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 6 & 5 & 4 \end{bmatrix} \text{ e } B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

$$A_{2 \times 3} * B_{3 \times 2} = \begin{bmatrix} (1*1) & (2*3) & (3*5) \\ (6*2) & (5*4) & (4*6) \end{bmatrix} = \begin{bmatrix} 1 & 6 & 15 \\ 12 & 20 & 24 \end{bmatrix}$$

Responda: Você conseguiu notar a diferença de processamento? O processamento (multiplicação) foi mais rápido com a implementação single thread ou multithread? Qual a biblioteca usada foi melhor para executar a operação? Explique os resultados obtidos.

Projeto 2

Realize uma implementação em **C/C++ (ou Python)** da soma de todas as posições com uma dimensão (um array 1D) utilizando o sistema single thread e multithread, no qual o último deve ser feito usando as bibliotecas **Pthread** e **OpenMP**. Varie o tamanho do vetor a ser somado em ambas as implementações e com ambas as bibliotecas. Varie os tamanhos do vetor, por exemplo de 10 até 150.000. Você também pode variar o tipo de dados, como inteiro e float. Exemplo de vetor: $A=[0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0]$

Responda: Você conseguiu notar a diferença de processamento? O processamento (soma) foi mais rápido com a implementação single thread ou multithread? Qual a biblioteca usada foi melhor para executar a operação? Explique os resultados obtidos.