Universidade do Vale do Itajaí
Curso de Bacharelado em Ciência da Computação
Disciplina: Sistemas Operacionais
Professor: Felipe Viel
Avaliação 01 – Threads e Paralelismo

Instruções

1. Esta avaliação deve ser feita individualmente, em dupla ou trio.

2. Data de entrega: 19/09/2023 até 18:59. Trabalhos entregues em atraso terão o desconto de 2,0 pontos por dia de atraso, incluindo feriado e finais 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 a linguagem de programação de sua preferência (C, C++, Python, Java, C#, Javascript/Node.js, Rust, etc). Porém, a utilização e suporte a threads pela linguagem escolhida é de responsabilidade do(s) aluno(s), seja usado corretamente o conceito de threads. As bibliotecas usadas devem ser equivalentes a Pthreads. Bibliotecas que também implementem e que permitam usar conceitos de paralelismo também podem ser usadas, mas o aluno também é responsável pelo seu uso e apresentação, como por exemplo Taskflow e TinyThread++.

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:
   - Identificação do autor e do trabalho.
   - Enunciado dos projetos.
   - Explicação e contexto da aplicação para compreensão do problema tratado pela solução.
   - Resultados obtidos com as simulações.
   - Códigos importantes da implementação.
   - Resultados obtidos com a implementação (tabelas, gráficos e etc).
   - 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). O repositório deve estar aberto do momento da entrega em diante, sendo que o professor não responsabiliza caso o projeto não esteja disponível para consulta no momento da correção, sendo do(s) aluno(s) essa responsabilidade de manter disponível.

8. O trabalho deverá ser apresentado em data definida pelo professor. É de responsabilidade do(s) aluno(s) explicar os conceitos, comandos, bibliotecas usadas. É de responsabilidade do(s) aluno(s) fazer a solução funcionar e ela deverá ser baixada do local de entrega no momento da apresentação. Trabalhos não apresentados terão como nota máxima 5,0.

Descrição do projeto a ser desenvolvido

Projeto 1

Realize uma implementação em **sua linguagem de preferência** de uma multiplicação entre matrizes utilizando o sistema single thread e multithread (pelo menos duas threads), no qual o último deve ser feito usando as bibliotecas **thread suportada na linguagem escolhida**. Realize uma análise comparativa no quesito tempo de processamento utilizando bibliotecas como **time.h (como o exemplo fornecido no material ou biblioteca equivalente na linguagem escolhida)**. A operação de multiplicação deve usar duas abordagens, a multiplicação matricial e a posicional, e deve ser entre, no

mínimo, matrizes quadráticas de 3X3, como no exemplo apresentado e os números deve estar em float (ponto flutuante):

Matrizes a serem multiplicadas (exemplo):

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad e \quad B = \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

Multiplicação matricial:

$$AB = \begin{bmatrix} 1 \times 9 & 2 \times 6 & 3 \times 3 \\ 4 \times 8 & 5 \times 5 & 6 \times 2 \\ 7 \times 7 & 8 \times 4 & 9 \times 1 \end{bmatrix}$$

Multiplicação posicional:

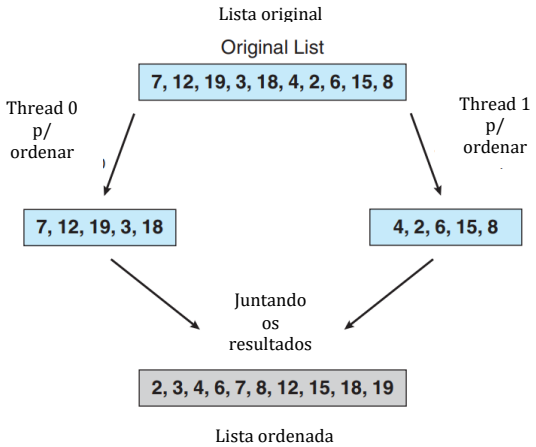
$$A@B = \begin{bmatrix} 1 \times 9 & 2 \times 8 & 3 \times 7 \\ 4 \times 6 & 5 \times 5 & 6 \times 4 \\ 7 \times 3 & 8 \times 2 & 9 \times 1 \end{bmatrix}$$

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

Você é livre para implementar estratégias diferentes para conseguir processar bem como usar recursos de aceleração em hardware das bibliotecas.

Projeto 2

Realize uma implementação em **sua linguagem de preferência** de algoritmo de ordenação de vetores *Bubble sort*. O vetor deverá ser de pelo menos 200 posições e deverá ser comparado um sistema singlethread com um sistema multithread (com pelo menos 2 threads). Além disso, a ordenação deverá ser na ordem crescente (do maior para o menor) e o vetor de valores deve ser iniciado do maior para o menor (descrente) não importando o valor das posições, desde que respeita essa regra. Isso gerará o pior caso de uso do *bubble sort*. Exemplo do bubble sort e outros algoritmos: exemplos.



Responda: Você conseguiu notar a diferença de processamento? O processamento foi mais rápido com a implementação single thread ou multithread? Explique os resultados obtidos.

Você é livre para implementar estratégias diferentes para conseguir processar bem como usar recursos de aceleração em hardware das bibliotecas.

Dicas para os projetos:

Observe que é exigido pelo menos duas threads (excluindo a que cria as threads), mas você pode utilizar mais. Além disso, você também tem que mensurar o custo de criação da thread, já que a comparação deverá ser justa com um código original. Outra observação é que você provavelmente não irá capturar o mesmo tempo entre diferentes testes, logo, repita pelo menos 5 vezes o processamento e você tem duas opções: (1) exibir a média ou (2) exibir o maior tempo de processamento. Além disso, você pode utilizar algoritmos prontos da literatura, onde o desafio do trabalho é a paralelização da operação.

Para a implementação do código, você pode fazer um fork do repositório da disciplina no github e usar o codespaces do github para a implementação, onde o mesmo irá executar o Visual Code em uma distro Linux Ubuntu com 2 núcleos e 8 GB de memória principal. Mas onde será executado seus códigos fica a critério do(s) aluno(s)

Pontuação extra em prova:

A fim de estimular a ampliação dos seus conhecimentos sobre paralelismo, concorrência e IPC em Sistemas Operacionais, será concedido de 0,5 à 1,5 ponto extra na nota da prova para os trabalhos que apresentarem uma implementação de um dos dois projetos usando a abordagem de comunicação de processos, podendo ser memória compartilhada ou troca de mensagens por *pipe*. Como é uma pontuação extra, não será descontado caso não realize essa implementação. Além disso, o professor, no momento da defesa não irá solicitar a apresentação, ficando a cargo dos alunos indicarem se implementaram e mostrar o funcionamento.