

Instruções

1. Esta avaliação deve ser feita individualmente, em dupla ou trio.
2. Data de entrega: **10/04/2023 até 18:55**. Não serão aceitos trabalhos entregues em atraso.
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. Porém, a utilização e suporte a threads e IPC (comunicação entre processos) pela linguagem escolhida é de responsabilidade do(s) aluno(s), seja usado corretamente o conceito de threads e comunicação entre processos. 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. O uso de modelos de aprendizado de máquina do tipo generativo (principalmente associados a processamento de linguagem natural) para geração de código devem ser explicados e uso inadequado resultará em zero.
7. 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 do projeto.
 - 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.
8. 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.
9. 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

Uma Indústria de Alimentos de Santa Catarina chamada FoodSec S.A. possui a tarefa de escanear alimentos por meio de câmeras e verificar se os mesmos estão corretos. Os alimentos podem passar por uma das duas esteiras disponíveis. As duas esteiras são controladas por um por um único computador centralizado. Esse computador recebe dados de um sensor em cada uma das esteiras que captura a contagem de itens que são identificados como seguros. A contagem é exibida em um display perto das esteiras (todos os displays são controlados pela mesma função, é apenas uma replicação).

Além disso, os pesos dos itens que passam por cada uma das esteiras são armazenados em um único vetor de dados. A cada 500 unidades de produtos, das duas esteiras, é necessário atualizar o peso total de itens processados. Sendo assim, a empresa aceita uma pausa na quantidade de itens sendo contados e pesados para realizar a pesagem total, mas ela necessita saber quanto tempo é necessário para isso.

A empresa também fornece uma análise das 2 esteiras:

- Esteira 1: produtos de maior peso (5 Kg) – passa 1 item a cada 2 segundos pelo sensor.
- Esteira 2: produtos de peso médio (2 Kg) – passa 1 item a cada 1 segundo pelo sensor.
- A contagem de peso deve interromper todo o sistema e deve acontecer ininterruptamente.
- A exibição no display deve atualizar a cada 2 segundos para os operadores poderem acompanhar.

Com isso, você deve implementar três versões da solução e comparar qual é a que apresenta o melhor desempenho quanto ao tempo de computação (tempo de processamento). A primeira solução deve usar IPC do tipo memória compartilhada. A segunda solução deve considerar IPC via *pipe* (de preferência nomeado). Já a terceira solução deve ser implementado via multithread (ou multi-thread, como queira descrever).

Os códigos disponibilizados pelo professor podem ser usados para o desenvolvimento do trabalho (na verdade, é extremamente recomendado). A avaliação do tempo deve ser entre a execução da contagem e na comunicação, sendo recomendado desconsiderar a criação de threads, *pipe* ou memória compartilhada.

Obs: considere que todos os processos/threads funcionam na mesma máquina (não a necessidade de comunicação entre computadores).

Você pode considerar simular a contagem com simples incremento de variável.

Pontuação extra em prova:

A fim de estimular a ampliação dos seus conhecimentos sobre paralelismo e concorrência em Sistemas Operacionais, será concedido de 0,5 à 1 ponto extra na nota da prova para os trabalhos que apresentarem uma implementação do algoritmos do trabalho em OpenMP (além da biblioteca que você escolheu inicialmente). A pontuação (de 0,0 a 1,0) fica a critério do professor e será baseada na qualidade da entrega feita. Essa implementação não é obrigatória.