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

Instruções

1. Esta avaliação deve ser feita individualmente e em até trio.

2. Data de entrega: 16/04/2024 até as 19:00. Trabalhos em atraso implicarão em 1,5 ponto de desconto por dia e, após 7 dias, não mais será aceita a entrega.

3. Esta avaliação tem por objetivo consolidar o aprendizado sobre conceitos de IPC, threads, concorrência 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

5. O sistema deve ser entregue funcionando corretamente. Sistemas não compilando e executando não serão aceitos. É de responsabilidade do(s) aluno(s) apresentar a execução funcionando corretamente.

6. Deve ser disponibilizado os códigos da implementação em repositório do(s) aluno(s) (github, bitbucket, etc...), deve ser fornecido o link e o repositório dever ser público.

7. O relatório deve seguir o formato de artigo científico ou seguindo as normas da ABNT e as orientações para produção de trabalhos acadêmicos da Univali contendo:

    - 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.

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. Cópias entre alunos implicará em nota zero para todos os envolvidos.

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, além dos descontos aplicados no restante da avaliação da implementação.

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, o peso total dos itens contabilizados deve ser atualizado no display a cada 500 unidades de produtos, das duas esteiras. A contagem de peso e de itens totais podem ser feitas de forma acumulativa, não necessitando armazena em um vetor.

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 exibição no display deve atualizar a cada 2 segundos para os operadores poderem acompanhar.

A empresa está avaliando a viabilidade de implementação nos Sistemas Operacionais Windows e Linux (qualquer distro) e também em duas formas de IPC. Com isso, você deve implementar duas versões da solução nos dois sistemas operacionais 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 *pipe* (de preferência nomeado), já a segunda solução deve ser implementada via multithread (ou multi-thread, como queira descrever). Logo, deve ser implementado as duas soluções para Windows e as duas para Linux. Você poderá usar o Codespace ou Cocalc, por exemplo, para implementar a versão em Linux. Para Windows, você pode usar máquinas virtuais também.

Os códigos disponibilizados pelo professor (de teste e exercícios) 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.