INSTITUTO FEDERAL
Paraná
Campus Pinhais

Bacharelado em ciência da computação
Arquitetura de computadores e sistemas operacionais
Atividade de implementação: concorrência (leitores e escritores) 8 de janeiro de 2025

Esta atividade foi adaptada de uma atividade de Carlos Maziero, UFPR.

1 Leitores e escritores

O objetivo deste projeto é construir um sistema com *threads* que acessam de forma concorrente uma fila de inteiros. Nesse sistema existem dois tipos de *threads*:

1. **Leitora**: que percorre a fila e imprime na tela o seu conteúdo e a média dos valores encontrados;
2. **Escritora**: que modifica o conteúdo da fila, removendo um elemento do início e acrescentando um elemento (aleatório) no final.

Observe que não há condição de disputa quando somente *threads* leitoras acessam a fila, pois as leituras não modificam a fila e podem ser simultâneas. No entanto, uma *thread* escritora deve ter acesso exclusivo à fila para fazer suas modificações, sem acessos concorrentes de outras *threads*, sejam leitoras ou escritoras, para evitar condições de disputa.

1.1 Soluções

Existem três tipos de soluções para esse problema de sincronização:

- **Priorização dos leitores**: sempre que um leitor quiser ler e não houver escritor escrevendo (pode haver escritor esperando), ele tem acesso à fila. Nesta solução, um escritor pode ter de esperar indefinidamente (inanição, ou *starvation*), pois novos leitores sempre chegam.
- **Priorização dos escritores**: quando um escritor desejar escrever, mais nenhum leitor pode fazer leituras enquanto o escritor não for atendido. Nesta solução, um leitor pode ter de esperar indefinidamente (inanição), pois novos escritores sempre chegam.
- **Prioridades iguais**: não há risco de inanição, pois leitores e escritores têm as mesmas chances de acesso à fila; pode haver uma queda de desempenho em relação às soluções anteriores.

2 Entrega

Você deve pesquisar as três soluções e implementá-las, usando como estrutura de dados uma fila de inteiros, *threads* POSIX (*pthreads*) e semáforos POSIX. Não serão aceitas soluções empregando espera ocupada (*busy-wait*). Em sua implementação, use três *threads* leitoras e duas escritoras.

O trabalho pode ser feito em grupos de até 3 pessoas. Apenas uma pessoa da equipe deve entregar a tarefa no Google Classroom. Você deve entregar **um** arquivo compactado (`.zip`), cujo nome é formado pelas iniciais dos integrantes do grupo, separados por um traço (exemplo: `gvcc-fs.zip`).

Ao ser descompactado, seu arquivo deve gerar uma pasta com as iniciais dos integrantes (`gvcc-fs`). Dentro desse diretório, **deve** existir um arquivo *README.md* que contenha o nome completo dos integrantes e outras informações relevantes sobre o trabalho (compilação, problemas, ...).

Além disso, você deve incluir, dentro do diretório, os arquivos C/C++ das três implementações, implementadas separadamente, conforme a figura 1. Note que, para cada implementação, tanto a *thread* leitora quanto a escritora devem estar no mesmo arquivo C/C++.

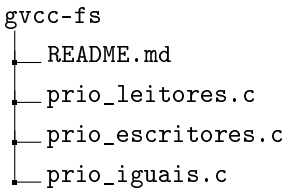

Figura 1: Exemplo de organização da entrega, para arquivos em C.