

Módulo 2 - Trabalho de Implementação (2020/2 REMOTO)

Computação Concorrente (MAB-117)

Prof. Silvana Rossetto

¹DCC/IM/UFRJ — 27 de abril de 2021

Descrição

Neste trabalho, vamos implementar o padrão **leitores e escritores**, com **prioridade para escrita**, e usá-lo para implementar uma **aplicação que simula um ambiente de monitoramento de temperatura**.

A prioridade para escrita significa que sempre que um escritor deseja escrever um dado novo, ele deve aguardar apenas os leitores que já estavam lendo terminarem suas leituras; novos leitores ficam impedidos de continuar até que o escritor seja atendido.

Uma aplicação que simula um ambiente de monitoramento de temperatura possui um ou mais **sensores** que medem a temperatura no ambiente monitorado com uma dada frequência (ex., a cada 1 minuto). Essas medidas são pré-processadas (por exemplo, para retirar ruídos ou para filtrá-las a partir de uma faixa de valores de interesse) e então são armazenadas ou encaminhadas para serem processadas por **atuadores**. Atuadores são dispositivos que controlam o funcionamento de algum equipamento, por exemplo, ar-condicionado ou alarme de incêndio.

A solução poderá ser desenvolvida na linguagem C, usando a biblioteca Pthreads, ou na linguagem Java.

Requisitos do trabalho

Os requisitos do trabalho estão divididos em: (a) requisitos da implementação do padrão leitores/escritores com prioridade para escrita; e (b) requisitos da aplicação que simula o monitoramento de temperatura.

Requisitos do padrão leitores/escritores

A implementação do padrão leitores/escritores com prioridade para escrita deverá ser feita em um módulo separado, com a garantia de ser *thread-safe*. Isso quer dizer que ele poderá ser instanciado e acessado por mais de um fluxo de execução (thread) ao mesmo tempo, sem levar a aplicação a problemas de condição de corrida e *deadlock*. A implementação deverá permitir qualquer número de leitores e de escritores.

Se a linguagem escolhida for C, esse módulo deverá contemplar a implementação de um TAD (tipo abstrato de dado), com um **arquivo .h** onde são definidas as estruturas de dados e as assinaturas das funções; e um **arquivo .c** onde são implementadas essas funções (sem incluir a função `main`).

Se a linguagem escolhida for Java, esse módulo deverá ser implementado como uma classe separada, com seus atributos e métodos.

Requisitos da aplicação de monitoramento

A aplicação deverá implementar a tarefa central dos **sensores** e dos **atuadores**. Ambos serão simulados por meio de **threads** independentes. A quantidade de sensores deverá ser passada como argumento do programa (na linha de comando).

Os **sensores** medirão a temperatura em intervalos de aproximadamente 1 segundo (para simular esses intervalos é possível usar a função `sleep()`). O valor medido poderá ser simulado gerando-se um número randômico no intervalo $[25, 40]$. Sempre que o valor lido for maior que 30, o sensor deverá escrevê-lo no **espaço de memória compartilhado com os demais sensores e atuadores**. O acesso a esse espaço de memória deverá seguir o padrão **leitores/escritores com prioridade para escrita**. O valor deverá ser registrado juntamente com o **identificador do sensor**, e o **identificador da leitura** (inicia com 0 e é incrementado a cada leitura daquele sensor).

A escrita nesse espaço de memória deverá seguir a ordem **FIFO**, sobreescrevendo-se o valor mais antigo. O tamanho desse espaço de memória deverá ser fixo com valor 60.

Simularemos um **atuador** de alarme de incêndio para cada sensor definido. Cada atuador deverá ler os valores registrados pelo sensor de mesmo *id* a cada 2 segundos. Se as 5 últimas leituras indicarem temperatura acima de 35 graus, o atuador irá emitir um alerta vermelho. Se entre as 15 últimas, 5 leituras indicarem temperatura acima de 35 graus, o atuador irá emitir um alerta amarelo. Caso contrário, o atuador irá emitir um alerta de condição normal. Além disso, ele deverá calcular a temperatura média de todas as leituras do sensor ainda disponíveis e exibí-la na tela.

A aplicação deve executar por tempo indeterminado.

Etapas do trabalho

A execução do trabalho deverá ser organizada nas seguintes etapas:

1. Compreender bem o problema proposto para este trabalho, seus artefatos e etapas de realização;
2. Projetar e implementar o TAD (em C) ou a classe (em Java) com o padrão leitores/escritores, com prioridade para escrita (**descrever os principais aspectos do projeto e implementação no relatório**).
3. Testar o TAD ou a classe para avaliar a sua corretude. Incluir log de execução que permita avaliar que todos os requisitos do padrão estão sendo atendidos, incluindo a prioridade para escrita (**descrever os testes realizados e seus resultados no relatório**).
4. Projetar e implementar a aplicação que simula o monitoramento de ambiente, seguindo os requisitos apresentados acima (**descrever os principais aspectos do projeto e implementação no relatório**).
5. Testar a aplicação desenvolvida para avaliar a sua corretude. Incluir log de execução que permita avaliar que todos os requisitos da aplicação estão sendo atendidos, incluindo o acesso aos dados seguindo o padrão leitores/escritores com prioridade para escrita (**descrever os testes realizados e seus resultados no relatório**).
6. Concluir a redação do relatório.

Artefatos que deverão ser entregues

- **Relatório:** com todas as informações solicitadas no item anterior.
- **Código fonte:** link para o repositório do código desenvolvido.

Critérios de avaliação

Os seguintes itens serão avaliados no trabalho com o respectivos pesos:

- Projeto, implementação e teste do padrão leitores/escritores com prioridade para escrita (**no relatório**): **3 pontos**
- Projeto, implementação e teste da aplicação que simula monitoramento de temperatura, e usa o o padrão leitores/escritores com prioridade para escrita (**no relatório**): **2 pontos**
- Interface de uso, organização e documentação do código fonte (**no código fonte**): **2 pontos**
- Execução correta de todos os códigos implementados (**na correção**): **3 pontos**

O trabalho pode ser feito individualmente ou em dupla. Os integrantes da equipe poderão ser chamados pela professora para explicar o trabalho.