

## Módulo 2 - Laboratório 6

### Problemas clássicos de concorrência usando locks e variáveis de condição (barreira)

Computação Concorrente (ICP-117)  
Prof. Silvana Rossetto

<sup>1</sup>DCC/IM/UFRJ

#### Introdução

O objetivo deste laboratório é continuar praticando a programação concorrente usando locks e variáveis de condição. Implementaremos uma aplicação que requer sincronização coletiva com **barreira**.

#### Atividade 1

**Objetivo:** Projetar e implementar uma aplicação que requer **sincronização coletiva** entre as threads.

**Descrição:** A aplicação alvo definirá um vetor de inteiros com tamanho  $N$  igual ao número de threads que serão criadas. Esse vetor será inicializado com números aleatórios entre 0 e 9 (na função `main`). Em seguida as  $N$  threads serão criadas.

Todas as threads executarão a mesma função. Nessa função, as threads criarão uma variável local para acumular o somatório de todos os elementos do vetor de inteiros nas  $N$  iterações. Em seguida executarão as  $N$  iterações. Em cada iteração, primeiro elas somarão todos os elementos do vetor de inteiros e acrescentarão o resultado na sua variável local acumuladora. Em seguida deverão aguardar todas as threads terminarem suas somas. Depois de concluído esse passo, cada thread deverá gerar um novo valor aleatório e escrever esse valor na posição do vetor de inteiros correspondente ao seu identificador (ID). Em seguida, novamente deverão aguardar as demais threads terminarem esse passo. Depois disso, passam para a próxima iteração.

Ao término das iterações, cada thread deverá retornar o somatório total acumulado na sua variável local após as  $N$  iterações. A fluxo da função `main` deverá aguardar o término, receber e imprimir na tela o valor recebido de cada thread; e verificar se todos os valores recebidos são iguais.

#### Roteiro:

1. Certifique-se de ter compreendido a descrição completa da aplicação, em seguida projete a sua solução. No projeto da solução, receba como parâmetro o número de threads que serão criadas.
2. Implemente a solução projetada, acrescentando no seu código a impressão de **informações que permitam acompanhar a execução da aplicação para verificar se as condições lógicas do problema são satisfeitas**.
3. Execute a aplicação **várias vezes** e avalie os resultados obtidos.

4. Altere o número de threads (e consequentemente o tamanho do vetor compartilhado) e reexecute a aplicação.

Disponibilize o código implementado em um ambiente de acesso remoto ([GitHub](#) ou [GitLab](#)). Use o formulário de entrega desse laboratório para enviar o link do repositório do código implementado e responder às questões propostas.