

## Laboratório 4

### Comunicação entre threads via memória compartilhada

Programação Concorrente (ICP-361) 2023-1  
Prof. Silvana Rossetto

<sup>1</sup>Instituto de Computação/UFRJ

#### Introdução

O objetivo deste Laboratório é introduzir o uso de variáveis compartilhadas (globais) para permitir a comunicação entre as threads de uma aplicação e mostrar quais problemas essa comunicação pode gerar.

Para cada atividade, siga o roteiro proposto e responda às questões colocadas.

#### Atividade 1

**Objetivo:** Mostrar um exemplo simples de programa com uma variável compartilhada entre threads, e os possíveis efeitos indesejáveis do acesso compartilhado.

##### Roteiro:

1. Abra o arquivo **exemplo1.c** e entenda o que ele faz. **Qual saída é esperada para o programa (valor final da variável soma)?**
2. Compile e execute o programa **várias vezes** e observe os resultados impressos na tela.
3. **Os valores impressos foram sempre o valor esperado? Se não, explique por que isso aconteceu?**
4. Esse programa apresenta o problema de **condição de corrida**?
5. É possível que a variável **soma** termine com valor acima de 200000? Por que?

**Relatório da atividade:** Em um arquivo de texto, inclua as respostas para as questões colocadas acima.

#### Atividade 2

**Objetivo:** Introduzir o uso de *locks* provido pela biblioteca Pthreads.

##### Roteiro:

1. Abra o arquivo **exemplo1a.c** e compreenda como locks são usados para implementar a exclusão mútua (*acompanhe a explicação da professora*).
2. Execute o programa **várias vezes**. Os valores impressos foram sempre o valor esperado? Por que?
3. Altere o número de threads e avalie os resultados.

**Relatório da atividade:** No mesmo arquivo de texto, acrescente as respostas para as questões colocadas acima.

### Atividade 3

**Objetivo:** Mostrar um exemplo de aplicação com threads e memória compartilhada em Java.

**Roteiro:** Abra o arquivo **TIncrementoBase.java** e siga o roteiro abaixo.

1. Leia o programa para entender o que ele faz (**acompanhe a explicação da professora**). **Qual é a seção crítica do código?** Qual saída é esperada para o programa (valor final de s)?
2. Compile o programa, execute-o **várias vezes** e observe os resultados impressos na tela. **Os valores impressos foram sempre o valor esperado? Por que?**

**Relatório da atividade:** No mesmo arquivo de texto, acrescente as respostas para as questões colocadas acima.

### Atividade 4

**Objetivo:** Mostrar como implementar **exclusão mútua** em Java.

**Roteiro:** Ainda no arquivo **TIncrementoBase.java**:

1. Comente as linhas 17-23; e descomente as linhas 27-33.
2. Acompanhe a explicação da professora sobre o uso de `synchronized` em Java.
3. Compile o programa, execute-o **várias vezes** e observe os resultados impressos na tela. **Os valores impressos foram sempre o valor esperado? Por que?**

**Relatório da atividade:** No mesmo arquivo de texto, acrescente as respostas para as questões colocadas acima.

**Entrega:** [Envie o arquivo texto com suas respostas pelo Google Sala de Aula, como resposta desta atividade \(Lab4\).](#)