

Laboratório 7

Semáforos

Programação Concorrente (ICP-361)
Profa. Silvana Rossetto

¹Instituto de Computação/UFRJ

Introdução

O objetivo deste Laboratório é introduzir e praticar o uso de semáforos para implementar exclusão mútua e sincronização condicional em programas concorrentes.

Atividade 1

Objetivo: Introduzir o uso de semáforos na linguagem C e sistema operacional Linux.

Roteiro: Abra o arquivo **semaf-1.c** e siga os passos abaixo:

1. Leia o programa e compreenda como o mecanismo de semáforo é usado em um programa C para implementar exclusão mútua.
2. [Com quais valores o semáforo deve ser inicializado? Acompanhe a explicação da professora.](#)
3. Execute o programa **várias vezes**.
4. [Os valores impressos foram sempre o valor esperado?](#)

Atividade 2

Objetivo: Mostrar um exemplo de uso de semáforos para atender requisitos de ordem de execução das threads.

Roteiro: Abra o arquivo **semaf-2.c** e siga os passos abaixo:

1. Leia o programa para entender como ele funciona. [Acompanhe a explicação da professora.](#)
2. [Com quais valores os semáforos devem ser inicializados?](#)
3. Execute o programa **várias vezes** e observe os resultados impressos na tela.

Atividade 3

Objetivo: Projetar e implementar um programa concorrente em C para ler, manipular e imprimir na tela os caracteres de um arquivo texto. Use **semáforos** para tratar as demandas de sincronização. [Acompanhe a explicação da professora.](#)

Descrição: O programa deverá ser implementado com 3 threads:

1. a thread 1 carrega os caracteres do arquivo de entrada em blocos de tamanho N e os deposita no **buffer1** de tamanho N.
2. a thread 2 processa os caracteres do buffer1 acrescentando um caracter de **final de linha** a cada $2n + 1$ caracteres lidos ($n = 0, 1, 2..10$) e deposita os caracteres no **buffer2**. Depois de n atingir o valor 10, o caracter de **final de linha** deverá ser inserido a cada bloco de 10 caracteres lidos.
3. a thread 3 imprime os caracteres do **buffer2** na tela.

Importante:

1. Os caracteres devem ser impressos na mesma ordem que foram escritos no arquivo de entrada.
2. Gere arquivos de entrada para teste.
3. Execute o programa várias vezes, alterando os parâmetros de entrada e ateste sua corretude.
4. A **organização e modularidade do código** serão levadas em consideração na avaliação do exercício.

Entrega do laboratório Disponibilize o código implementado na **Atividade 3** 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 acrescentar comentários ou dúvidas.