

Laboratório de Computação Concorrente

Comunicação entre threads via memória compartilhada

Introdução: O objetivo deste Laboratório é introduzir o uso de variáveis compartilhadas para 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 as questões colocadas.

Atividade 1

Objetivo: Mostrar o comportamento não-determinístico de um programa com três threads, ilustrando uma situação de condição de corrida.

Roteiro:

1. Abra o arquivo exemplo1.c e entenda o que o programa faz. Quais são os possíveis valores finais da variável y e por que? É possível y terminar com valor 3 (combinação binária dos números 1 e 2)? Por que?
2. Compile e execute o programa várias vezes e observe os resultados impressos na tela.
3. Altere o código do programa modificando a ordem em que as threads são criadas e repita as execuções
4. Esse programa apresenta o problema de condição de corrida? Se sim, ele é tolerável ou não nessa aplicação? (ou seja, é uma condição de corrida ruim?)

Atividade 2

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 exemplo2.c e entenda o que ele faz. Qual saída é esperada para o programa (valor final da variável s)?
2. Compile e execute o programa várias vezes e observe os resultados impressos na tela.
3. Os valores impressos foram sempre o esperado? Se não, explique por que isso aconteceu.
4. Esse programa apresenta o problema de condição de corrida? Se sim, ele é tolerável ou não nessa aplicação? (ou seja, é uma condição de corrida ruim?)

Atividade 3

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

Roteiro:

1. Abra o arquivo exemplo3.c e compreenda como locks são usados para implementar a exclusão mútua
2. Execute o programa várias vezes. Os valores impressos foram sempre o valor esperado?
3. Altere o número de threads e avalie os resultados.