

Introduzindo variáveis compartilhadas e analisando corrida de dados

Atividade 1

- 3 Os valores foram diferentes pois, devido a intercalação das threads, está ocorrendo uma corrida de dados. Em um programa concorrente, mesmo com a mesma entrada, a ordem das threads e o momento em que elas executam cada linha depende do computador e será diferente sempre que rodarmos o programa, como nesse exemplo temos uma variável global sendo usada por mais de uma thread, pode ocorrer que em algum momento as duas threads acessem essa variável ao mesmo tempo, assim, duas threads vão ler e retornar o mesmo valor, causando problemas no resultado final;
- 4 Não é possível que a soma fique acima de 2000 (ou no caso nthreads $\times 100000$) pois esse é o número de somas que há no programa independente da ordem que as threads executam. Ou seja, só podemos ter número menores que 200000, não é possível incrementar na soma, apenas diminuir, no caso em que há corrida de dados.

Atividade 2

- 2 Os valores foram sempre os mesmos pois utilizamos um lock, ou seja, identificamos o momento em que a variável global deve ser acessada de maneira isolada e “trancamos” essa parte do código, assim, sempre que uma thread está executando nessa parte do código outra thread não pode executar a mesma parte.

Atividade 3

- 2 O programa deve imprimir ALGUNS múltiplos de 10 dentro da faixa assumida pela variável soma, já que a thread extra vai executar enquanto as outras executam ela pode ou não analisar um número que a variável soma assume;
- 3 Os valores não são os esperados pois alguns deles não são múltiplos de 10;
- 4 Ainda sim, espera-se que ALGUNS múltiplos de 10 sejam impressos na saída, pela mesma razão anterior, nem todos os número serão testados na thread extra. Obs.: O código foi corrigido acrescentando o mesmo lock “mutex” na linha anterior à linha em que a variável soma é analisada (com relação a ser múltiplo de 10) e na linha após o print (quando a váriaval soma já foi printada no caso de ser múltiplo de 10);
- 5 Sim, o problema foi resolvido, o programa faz a soma correta sem que uma thread interfira na outra e só imprime números múltiplos de 10.