

Laboratório 1

Introdução a programação multithreading em C

Programação Concorrente (ICP-361) - 2023.1
Prof. Silvana Rossetto

¹Instituto de Computação/UFRJ

Introdução

O objetivo deste Laboratório é apresentar os conceitos mais básicos de programação concorrente, criando e avaliando programas concorrentes em C.

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

Considera-se que os programas serão executados em um terminal Linux. Quem estiver usando Windows, provavelmente precisará configurar o ambiente de programação C para incluir a biblioteca `pthread.h`.

Atividade 1

Objetivo: Mostrar como criar um programa concorrente em C, usando a biblioteca `pthread`.

Roteiro:

1. Abra o arquivo **helloC.c**, leia o código do programa e tente entender o que ele faz. **O que será impresso na tela quando esse programa for executado? Acompanhe a explicação da professora.**
2. Compile o programa *helloC.c* acrescentando a opção **-lpthread** na linha de comando (a opção `lpthread` acrescenta as funções da biblioteca *pthread*) (ex.: `gcc -o helloC helloC.c -lpthread`).
3. Execute o programa **várias vezes** (ex.: `./helloC`) e observe os resultados impressos na tela. **Os resultados estão de acordo com o esperado? Observe se há mudanças na ordem de execução das threads e tente entender porque isso pode acontecer.**
4. Altere o código acrescentando duas mensagens de log, antes da criação e depois da finalização das threads. Execute o programa várias vezes e observe os resultados.
5. Comente o trecho de código com a chamada *pthread_join*. Execute o programa várias vezes e observe os resultados. Aconteceu algo diferente do esperado? **Acompanhe a explicação da professora.**

Atividade 2

Objetivo: Implementar o seu primeiro programa concorrente! Escreva um programa com **duas threads**, para **multiplicar por 2** cada elemento de um vetor de 10000 elementos. (Para cada elemento a_i do vetor, calcular o novo valor e escrever o resultado na mesma posição do elemento.)

Roteiro:

1. Comece pensando em como dividir a tarefa de multiplicar todos os elementos do vetor entre duas threads e **como fazê-las acessar o mesmo vetor**. **Ambas as threads deverão executar a mesma função.**
2. Na função *main*, faça a inicialização do vetor; crie as duas threads; aguarde o término da execução das threads criadas e **verifique se os valores finais do vetor estão corretos**.
3. Execute o programa várias vezes e verifique se ele está funcionando corretamente.

Entrega do laboratório: Disponibilize o código implementado na **Atividade 2** 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 demais questões inseridas no formulário.