

## Módulo 1 - Laboratório 3

### Implementação e avaliação de aplicações concorrentes (parte 2)

Computação Concorrente (MAB-117) 2020.1 REMOTO

Prof. Silvana Rossetto

<sup>1</sup>DCC/IM/UFRJ

#### Introdução

O objetivo deste Laboratório é projetar e implementar uma versão concorrente para o problema de **calcular a soma de uma série de valores reais que aproxima o valor de  $\pi$** ; e avaliar o desempenho da aplicação em termos de tempo de execução. Usaremos a linguagem C e a biblioteca *Pthreads*.

Acompanhe a explanação da professora nas vídeo-aulas deste laboratório. Se tiver dúvidas, entre em contato por email.

#### Atividade 1

##### Roteiro:

1. Implemente uma função sequencial para calcular o valor de  $\pi$  usando a série abaixo:  
$$\pi = 4 * [1 - 1/3 + 1/5 - 1/7 + 1/9 - ...]$$

**Tente minimizar o erro numérico.** O número de elementos (N) da série deve ser informado pelo usuário na chamada do programa. *Sugestão: use o tipo **double** para definir a variável que receberá o resultado da soma da série e imprima todos os valores com 15 casas decimais.*
2. Para verificar o quanto o resultado calculado se aproxima do valor de  $\pi$ , compare-o com a constante *M\_PI* (de *math.h*).
3. Aumente o valor de N e verifique se o resultado da série se aproxima mais do valor de  $\pi$ . *Sugestão: defina a variável N do tipo **long long int** e use a função **atoll()** para converter o valor recebido do usuário (string) para long long int.*

#### Atividade 2

##### Roteiro:

1. Estenda o código da Atividade 1, e **implemente uma solução concorrente para calcular o valor de  $\pi$** , usando a mesma série da Atividade 1. Divida a tarefa igualmente entre as threads, **tentando minimizar o erro numérico**. O número de elementos (N) e de threads (T) deve ser informado pelo usuário na chamada do programa. Use a função *pthread\_exit()* para retornar o valor calculado pela thread para a função *main()*
2. Compare o valor calculado pela função sequencial com a solução concorrente para os **mesmos valores de N**. **Os resultados coincidem? Por que? Qual solução se aproxima mais rapidamente do valor de  $\pi$ ?**
3. Faça a tomada de tempo de execução do seu código, sequencial e concorrente (apenas da parte central de processamento, incluindo a sobrecarga de criação e finalização das threads). Exiba os resultados na saída padrão.

4. Calcule o ganho de desempenho (aceleração) obtido com a versão concorrente:  $(T_{\text{sequencial}}/T_{\text{concorrente}})$ , para os seguintes valores de N:  $10^3$ ,  $10^5$ ,  $10^7$ ,  $10^9$ .

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 questões propostas.