

Módulo 4 - Laboratório 7

Introdução à programação concorrente em Python

Programação Concorrente (ICP-361)
Prof. Silvana Rossetto

¹Instituto de Computação/CCMN/UFRJ - 2023-1

Introdução

O objetivo deste Laboratório é **introduzir a programação concorrente em Python**. Para cada atividade, siga o roteiro proposto e responda às questões colocadas.

Atividade 1

Objetivo: Mostrar como criar threads em Python estendendo a classe Thread.

Roteiro: Abra o arquivo **helloPython.py** e siga o roteiro abaixo.

1. Execute o programa várias vezes, e observe os resultados impressos.

Atividade 2

Objetivo: Mostrar um exemplo de uso de variável compartilhada.

Roteiro: Abra o arquivo **incremento.py** e siga o roteiro abaixo.

1. Leia o programa e compreenda como ele funciona. Qual é o resultado (correto) esperado na saída? **Dado que apenas uma thread executa de cada vez em Python, há condição de corrida neste programa?**
2. Execute o programa várias vezes, e observe os resultados impressos. Os resultados são corretos?
3. As linhas 17 e 18 mostram o uso do mecanismo de `lock` para implementar uma seção crítica. Descomente essas linhas, execute o programa novamente e observe os resultados.

Atividade 3

Objetivo: Mostrar um exemplo de programa com uma thread que lê a entrada padrão.

Roteiro: Abra o arquivo **io.py** e siga o roteiro abaixo.

1. Leia o programa e compreenda como ele funciona.
2. Execute o programa e observe seus resultados. **Qual o benefício da computação concorrente neste exemplo?**
3. Acompanhe a explanação da professora.

Atividade 4

Objetivo: Mostrar um exemplo de programa *IO-bound* que se beneficia da programação multithreading em Python.

Roteiro: Abra o arquivo **web.py** e siga o roteiro abaixo.

1. Leia o programa e compreenda como ele funciona.
2. Crie uma conta de acesso à API de informações do tempo neste link: <https://www.weatherapi.com/>
3. Acompanhe a explanação da professora.

Atividade 5

Objetivo: Implementar um **programa concorrente em Python** para calcular a série que gera o valor de π mostrada abaixo. Divida a tarefa entre as threads de forma balanceada.

$$\pi = 4 \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots + (-1)^n \frac{1}{2n+1} + \cdots \right)$$

Roteiro:

1. Na thread *main*, crie e dispare as threads, aguarde todas as threads terminarem e verifique se o cômputo final está correto (compare com o valor de π);
2. Teste seu programa variando o valor de n e o número de threads.
3. Inclua a tomada de tempo e compare os resultados executando o programa com 1, 2 e 4 threads.

Entrega: Disponibilize o código implementado na **Atividade 5** 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 colocadas.**