

## FORMAÇÃO COMPLEMENTAR

Mentor: Felipe Moreira de Assunção

Conteúdo para formação complementar do Módulo de Threads

## INFORMAÇÕES DO MÓDULO

### Descrição

Conheça os conceitos básicos sobre Threads

### Objetivos do Ensino

Ao final desse módulo você será capaz de aprender:

- Como executar tarefas de forma paralela com threads
- Vantagens e desvantagens no uso de threads
- Objetos threads
- Exemplos

### Demonstração das Ferramentas

Utilização do Anaconda com Jupyter Notebook e Python ou editor de código de sua escolha

## THREADS

### 1. O que são threads?

Segundo a Wikipedia, Thread é a tarefa que um determinado programa realiza. É uma forma de um processo dividir a si mesmo em duas ou mais tarefas que podem ser executadas concorrentemente. O suporte a thread é fornecido pelo próprio sistema operacional, no caso da linha de execução a nível do núcleo (Kernel-Level Thread – KLT) ou implementada através de uma biblioteca de uma determinada linguagem, no caso de uma User-Level Thread (ULT). Um thread permite, por exemplo, que o utilizador de um programa utilize uma funcionalidade do ambiente enquanto outras linhas de execução realizam outros cálculos e operações.

Os sistemas que suportam um único thread (em real execução) são chamados de monothread enquanto que os sistemas que suportam múltiplas threads são chamados de multithread.

### 2. Onde podemos utilizar threads?

Um exemplo comum do uso de threads são jogos, que podem ser modelados com rotinas de execução diferentes como para gerar o desenho de uma imagem e uma outra rotina para o áudio. Podemos dizer que nesse caso, existe um thread para tratar a rotina de desenho e outro thread para tratar o áudio. No ponto de vista do usuário, a imagem é desenhada ao mesmo tempo em que o áudio é emitido pelos alto-falantes. Porém, para sistemas com uma única CPU, cada linha de execução é processada por vez.

### 3. O que podemos dizer sobre processos vs. threads?

- Processos incluem: código, dados e estado da informação. São instâncias independentes de um programa em execução
- Threads incluem: pacote independente de execução, subconjunto de um processo, o sistema operacional ou a linguagem capaz de gerenciar a ordem de execução dos threads

## 4. O que é o GIL?

Segundo o site, Real Python (<https://realpython.com/python-gil/>), O Python Global Interpreter Lock ou GIL, em palavras simples, é um mutex (ou um bloqueio) que permite que apenas um thread mantenha o controle do interpretador Python.

Isso significa que apenas um thread pode estar em um estado de execução a qualquer momento. O impacto do GIL não é visível para desenvolvedores que executam programas single-thread, mas pode ser um gargalo de desempenho em código vinculado à CPU e multi-threaded.

Uma vez que o GIL permite que apenas um thread seja executado por vez, mesmo em uma arquitetura multithread com mais de um núcleo de CPU, o GIL ganhou a reputação de um recurso “infame” do Python.

## BIBLIOGRAFIA

- Python Books: <https://wiki.python.org/moin/PythonBooks>
- Introdução à Programação com Python: Algoritmos e Lógica de Programação Para Iniciantes
- Python Fluente: Programação Clara, Concisa e Eficaz