

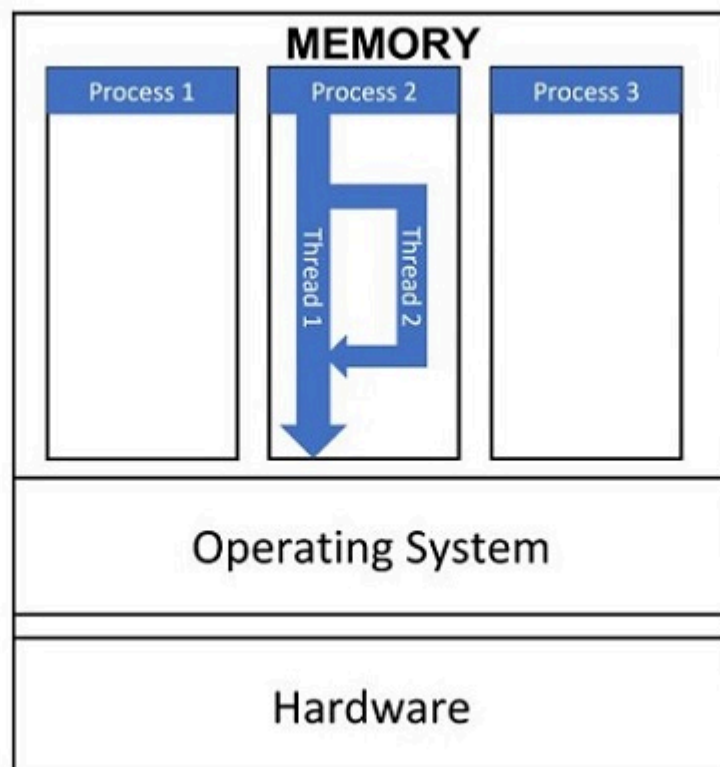
# Python - Multithreading

Por padrão, um programa de computador executa as instruções de maneira sequencial, do início ao fim. Multithreading refere-se ao mecanismo de dividir a tarefa principal em mais de uma subtarefa e executá-las de forma sobreposta. Isso torna a execução mais rápida em comparação com um único thread.

O sistema operacional é capaz de lidar com vários processos simultaneamente. Ele aloca um espaço de memória separado para cada processo, de modo que um processo não possa acessar ou gravar nada no espaço de outro. Por outro lado, um thread pode ser considerado um subprocesso leve em um único programa. Threads de um único programa compartilham o espaço de memória alocado a ele.

Vários threads dentro de um processo compartilham o mesmo espaço de dados com o thread principal e podem, portanto, compartilhar informações ou comunicar-se entre si com mais facilidade do que se fossem processos separados.

Por serem leves, não requerem muita sobrecarga de memória; eles são mais baratos que os processos.



Um processo sempre começa com um único thread (thread principal). Conforme e quando necessário, um novo thread pode ser iniciado e uma subtarefa é delegada a ele. Agora os dois threads estão funcionando de maneira sobreposta. Quando a tarefa atribuída ao thread secundário termina, ele se funde com o thread principal.

