

THREADS

Uma **thread** é uma unidade de execução dentro de um processo. Um **processo** é, basicamente, um programa em execução. Dentro de um processo, você pode ter várias threads, que são "linhas" independentes de execução que compartilham recursos do mesmo processo.

Diferença entre Processo e Thread

Processo: É uma instância de um programa em execução. Cada processo tem seu próprio espaço de memória e recursos. Quando você executa um programa, ele se torna um processo. Exemplo: um navegador de internet aberto é um processo.

Thread: É uma unidade de execução dentro de um processo. Quando um processo é criado, ele geralmente começa com uma thread principal. Se esse processo for projetado para dividir o trabalho, ele pode criar outras threads para executar tarefas simultaneamente. Exemplo: no navegador, a thread principal pode ser responsável por carregar a página, enquanto outras threads podem ser responsáveis por fazer download de imagens, tocar vídeos ou processar o histórico.

Metáfora: O Processo seria uma empresa inteira, com seus recursos e a capacidade de dividir seu trabalho entre diferentes departamentos (as threads). E uma Thread seria o departamento (ou uma função dentro da empresa), responsável por uma tarefa específica dentro do processo.

Problemas em Concorrência

Deadlock (Interbloqueio)

Um deadlock ocorre quando dois ou mais processos (ou threads) ficam **bloqueados** esperando um pelo outro, de forma que nenhum deles pode continuar a execução. Ou seja, há um ciclo de dependências em que cada processo espera um recurso que é mantido por outro processo do ciclo, e nenhum deles pode seguir.

Race Condition (Condição de Corrida)

Ocorre quando dois ou mais processos ou threads tentam acessar e modificar dados compartilhados ao mesmo tempo, e o resultado da operação depende da ordem em que as threads executam as ações. Isso pode levar a comportamentos inesperados ou incorretos. Ex.: suponha que dois processos A e B estão tentando atualizar um valor compartilhado. O processo A lê o valor, faz uma alteração e o escreve de volta, mas antes que o processo A termine, o processo B também lê o valor, faz sua própria alteração e escreve o valor de volta. O problema é que ambos os processos podem sobrescrever as alterações do outro, causando um erro.

NÚCLEO

Um **núcleo** (ou **core**, em inglês) é a parte de um processador (CPU) responsável por executar instruções de programas.

Processador e Núcleos

Um processador (ou CPU) é o cérebro do computador, responsável por executar as instruções dos programas. O processador pode ter um ou mais núcleos.

- **Processador de 1 núcleo:** um processador com apenas 1 núcleo pode executar uma tarefa por vez. Ou seja, ele realiza uma tarefa e, quando termina, começa a próxima.
- **Processador de múltiplos núcleos:** Quando o processador tem mais núcleos, ele pode executar várias tarefas simultaneamente. Cada núcleo pode ser responsável por processar uma tarefa separada. Isso melhora o desempenho, especialmente em sistemas que precisam de multitarefa.

Hyper-Threading (HT)

Com **Hyper-Threading**, um processador pode **simular** ter mais núcleos lógicos. Por exemplo, um processador com 4 núcleos físicos pode, com Hyper-Threading, se comportar como se tivesse 8 núcleos lógicos, já que cada núcleo físico pode processar dois threads (ou tarefas) ao mesmo tempo.