**(a) Técnicas de Pipeline, Multitasking, Multithreading, Multiprocessing**

**1. Pipeline (Paralelismo)**

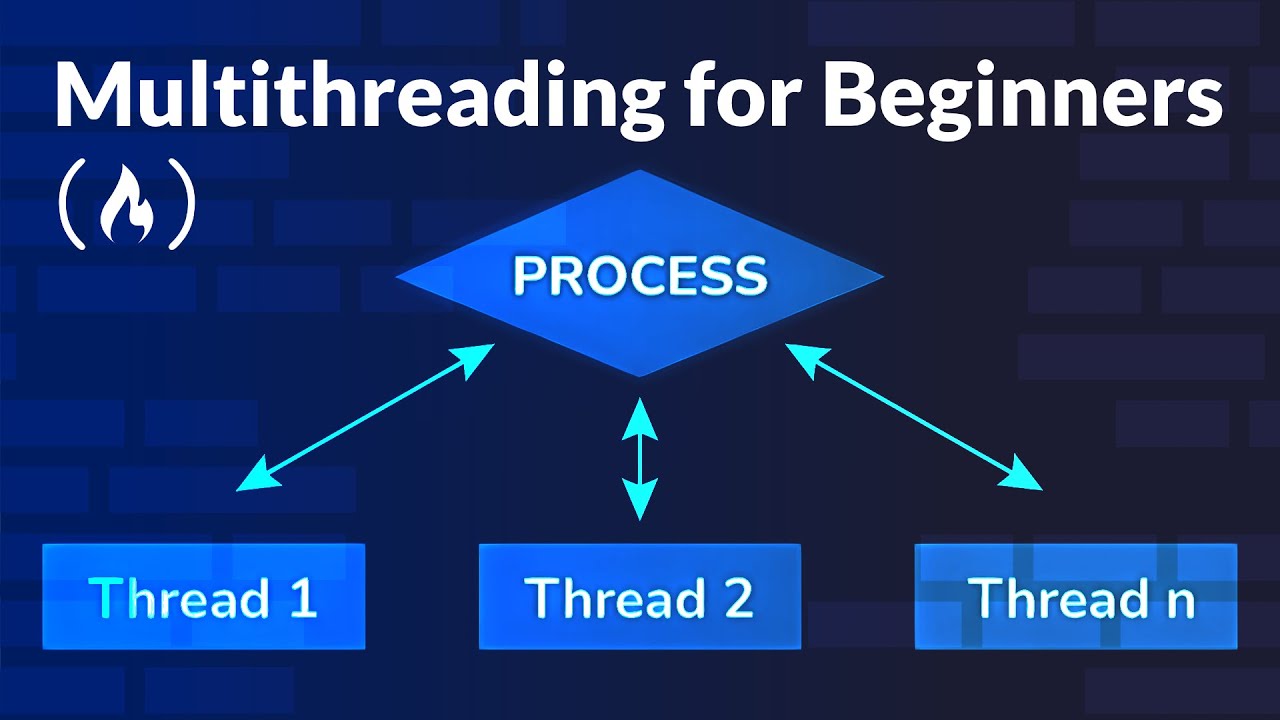
* Técnica usada em processadores para **executar várias partes de instruções ao mesmo tempo** - ocorre quando diferentes etapas de um processo são executadas simultaneamente em dispositivos separados
* **Analogia**: linha de montagem. Enquanto uma parte do processador busca a próxima instrução, outra decodifica, e outra executa.
* **Benefício**: aumenta a eficiência sem aumentar a frequência do clock.

**2. Multitasking (Multitarefa)**

* Capacidade de o sistema operacional **executar várias tarefas ao mesmo tempo**.
* Exemplo: ouvir música, navegar na internet e editar um texto simultaneamente.

**3. Multithreading (Múltiplas threads)**

* Um **processo dividido em múltiplas threads**, que compartilham o mesmo espaço de memória e recursos.
* Thread - unidade de execução dentro de um processo (como uma função que roda em paralelo com outra).
* Processo - instância de um programa em execução + seu contexto.
* Exemplo: em um navegador, cada aba pode ser um thread.



**4. Multiprocessing (Multiprocessamento)**

* Uso de **mais de um processador ou núcleo físico** para executar processos.
* Aumenta a velocidade e permite executar tarefas pesadas paralelamente.

**(b) Diferença entre sistemas com e sem Pipeline**

**-> Sem Pipeline:**

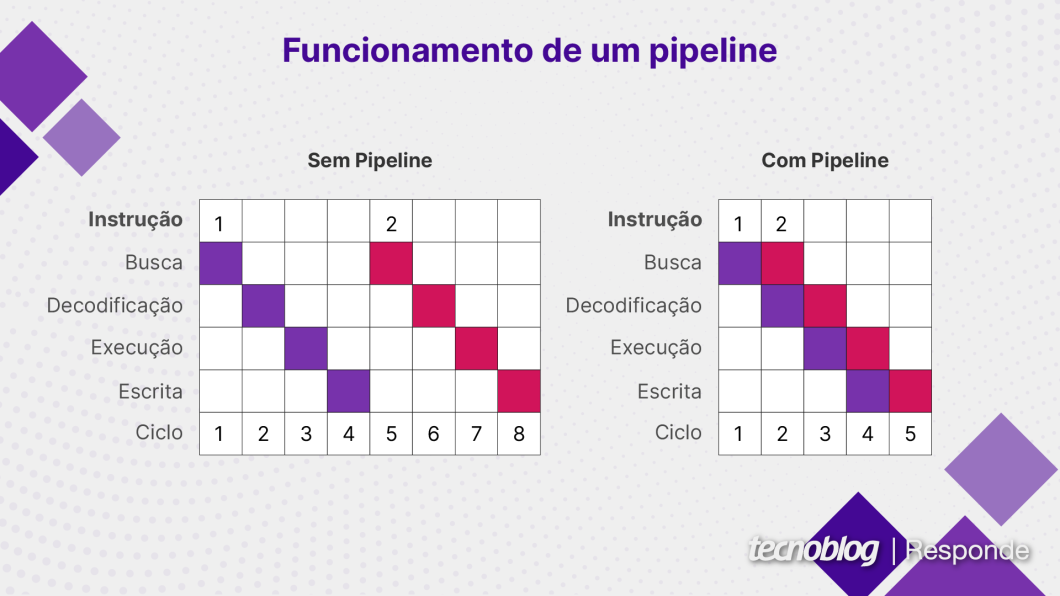
* As instruções são executadas uma de cada vez, do início ao fim.
* Processo sequencial, mais lento.

**-> Com Pipeline:**

* Divide a execução em etapas (buscar, decodificar, executar, etc.).
* Enquanto uma instrução está sendo executada, a próxima já pode estar sendo decodificada.
* Resultado: **maior desempenho**.

**---> Analogia:**

"Imagine uma fábrica onde cada trabalhador faz uma parte do trabalho em sequência, em vez de um único trabalhador fazendo tudo. O trabalho termina muito mais rápido."



**(c) Conceitos de Programa, Tarefa, Processo e Thread**

**1. Programa:**

* Conjunto de instruções armazenado no disco (não está em execução ainda).
* Exemplo: o instalador do Word.

**2. Tarefa:**

* Uma **atividade específica** que o sistema precisa realizar.
* Exemplo: salvar um arquivo ou imprimir um documento.

**3. Processo:**

* Um programa **em execução**, com seu próprio espaço de memória.
* Exemplo: Word aberto, editando um texto.

**4. Thread:**

* Parte menor de um processo.
* Vários threads podem rodar **dentro de um mesmo processo**.
* Exemplo: no Word, um thread pode cuidar da digitação e outra da verificação ortográfica.

**(d) Exemplos práticos**

Utilizando o navegador Google Chrome como exemplo:

* **Programa**: o Chrome instalado no computador.
* **Processo**: quando você abre o Chrome.
* **Tarefa**: abrir uma nova aba, baixar um arquivo.
* **Thread**: cada aba pode ser um thread, permitindo que uma não trave a outra.

**(e) Conceitos de Multitasking, Multithreading e Multiprocessing**

**1. Multitasking:**

* Executa **vários processos simultaneamente**.
* Exemplo: ouvir música e usar o Word ao mesmo tempo.

**2. Multithreading:**

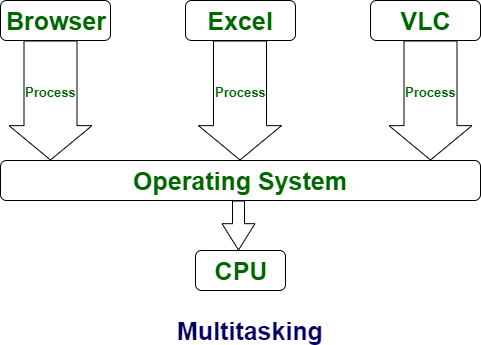
* Dentro de um processo, vários threads executam tarefas paralelas.
* Exemplo: em um jogo, um thread cuida do som, outra da imagem, outra dos controles.

**3. Multiprocessing:**

* Sistema usa **vários processadores físicos** (ou núcleos).
* Ideal para sistemas de alto desempenho e tarefas pesadas (como renderização de vídeo, cálculos científicos, etc.).

-> O multithreading e o multiprocessing utilizam a execução simultânea de múltiplas instâncias de threads e processos, respectivamente. Sendo assim, ambos são responsáveis por implementar multitasking em sistemas de computador.

-> Ademais, o multithreading pode ser considerado uma forma de multitasking dentro de um único processo, enquanto o multiprocessing amplia o conceito de multitasking para usar vários processadores físicos.



**(f) Diferença entre modelos Cooperativo e Preemptivo**

**1. Cooperativo:**

* Cada tarefa decide quando liberar o processador.
* Problema: se uma tarefa "segura" o controle, trava todo o sistema.

**2. Preemptivo:**

* O sistema operacional decide quando interromper e trocar de tarefa.
* Mais moderno e seguro. Evita que um processo monopolize a CPU.

**-> Analogia:**

"Cooperativo é como dividir um microfone em uma roda de conversa — depende da boa vontade das pessoas passarem o microfone. Preemptivo é como o mediador que controla o tempo de fala."

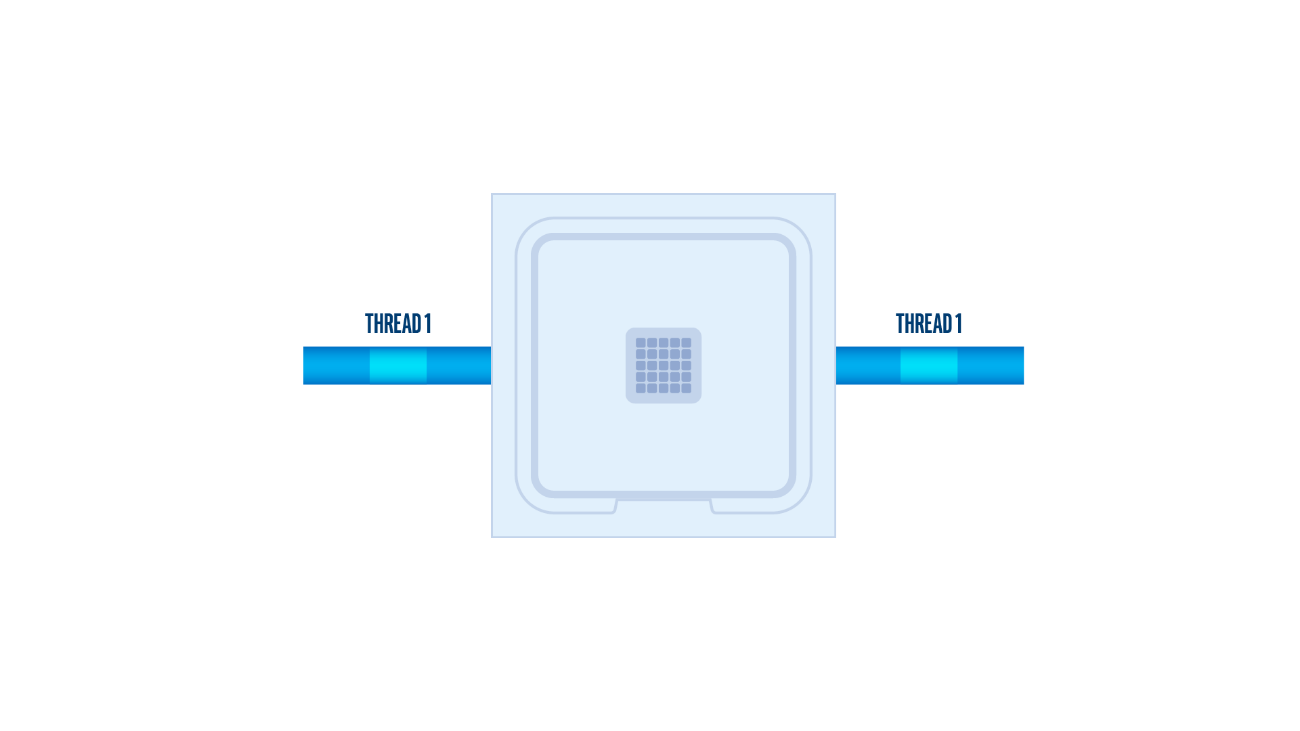
**(g) Hyper-Threading**

* Tecnologia da Intel que **permite um único núcleo físico se comportar como dois lógicos**.
* Melhora o aproveitamento da CPU, rodando mais threads simultaneamente.
* **Exemplo**: se o núcleo estiver ocioso esperando uma tarefa do disco, ele usa o tempo livre para executar outra thread.

**-> Analogia:**

"Como se um funcionário do caixa, ao esperar o troco sair, já começasse a atender o próximo cliente."

-> Intel® Hyper-Threading Technology is a hardware innovation that allows more than one thread to run on each core.



Referências:

<https://www-sciencedirect-com.translate.goog/topics/computer-science/pipeline-parallelism?_x_tr_sl=en&_x_tr_tl=pt&_x_tr_hl=pt&_x_tr_pto=wa>

<https://tecnoblog.net/responde/o-que-e-pipeline-processador/>

<https://focusnfe.com.br/blog/o-que-e-multithreading-e-como-a-tecnica-beneficia-seu-software/#:~:text=Sendo%20assim%2C%20ambos%20são%20responsáveis,para%20usar%20vários%20processadores%20físicos>.

<https://www-baeldung-com.translate.goog/cs/multiprocessing-multithreading?_x_tr_sl=en&_x_tr_tl=pt&_x_tr_hl=pt&_x_tr_pto=tc>

<https://www.ni.com/docs/en-US/bundle/labview/page/multitasking-multithreading-and-multiprocessing.html?srsltid=AfmBOopLCmHDKLZyML3YMcaotAdxiQfemC7F5UUtUwdz4le5YAbIisyP>

https://www.intel.com/content/www/us/en/gaming/resources/hyper-threading.html