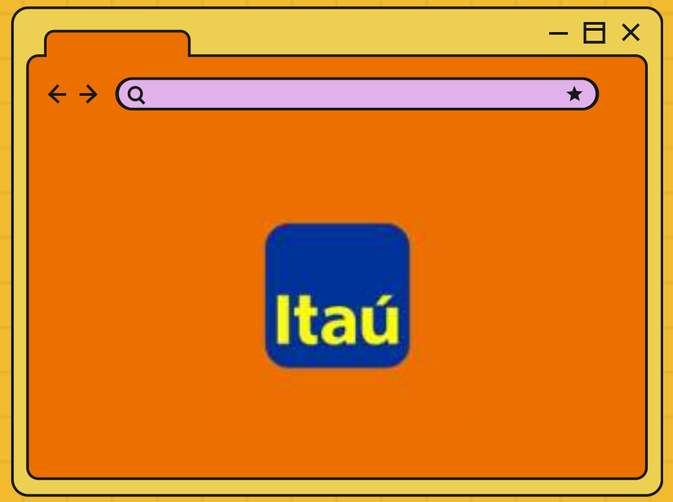
Programação paralela com Python

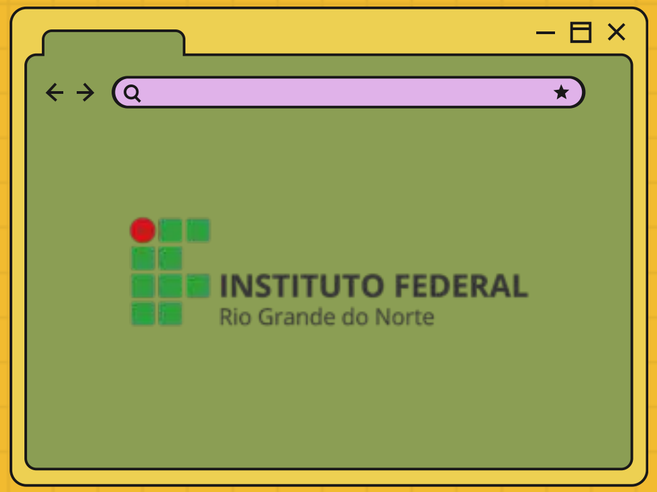
Lucas Matheus

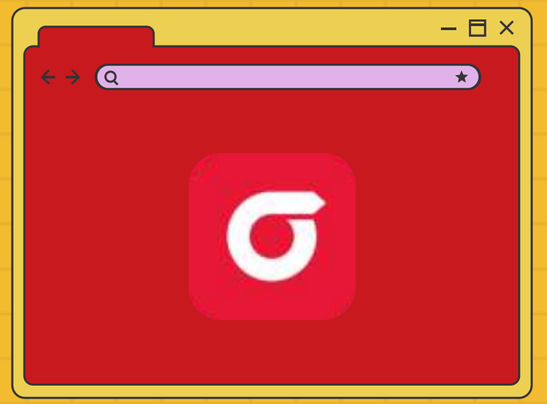
Quem sou eu?

- Técnico em Redes de Computadores - UFRN
- Técnico em Info/Internet - IFRN
- Estudante de Sistemas para Internet - IFRN
- Co-Fundador - LavaiLavem Turismo
- Pesquisador - Nocs Lab, Computação de Alto Desempenho e 5G
- Pesquisador - NiTec. Computação aplicada a ciência e matemática

Contribuições







Agenda

Tópicos

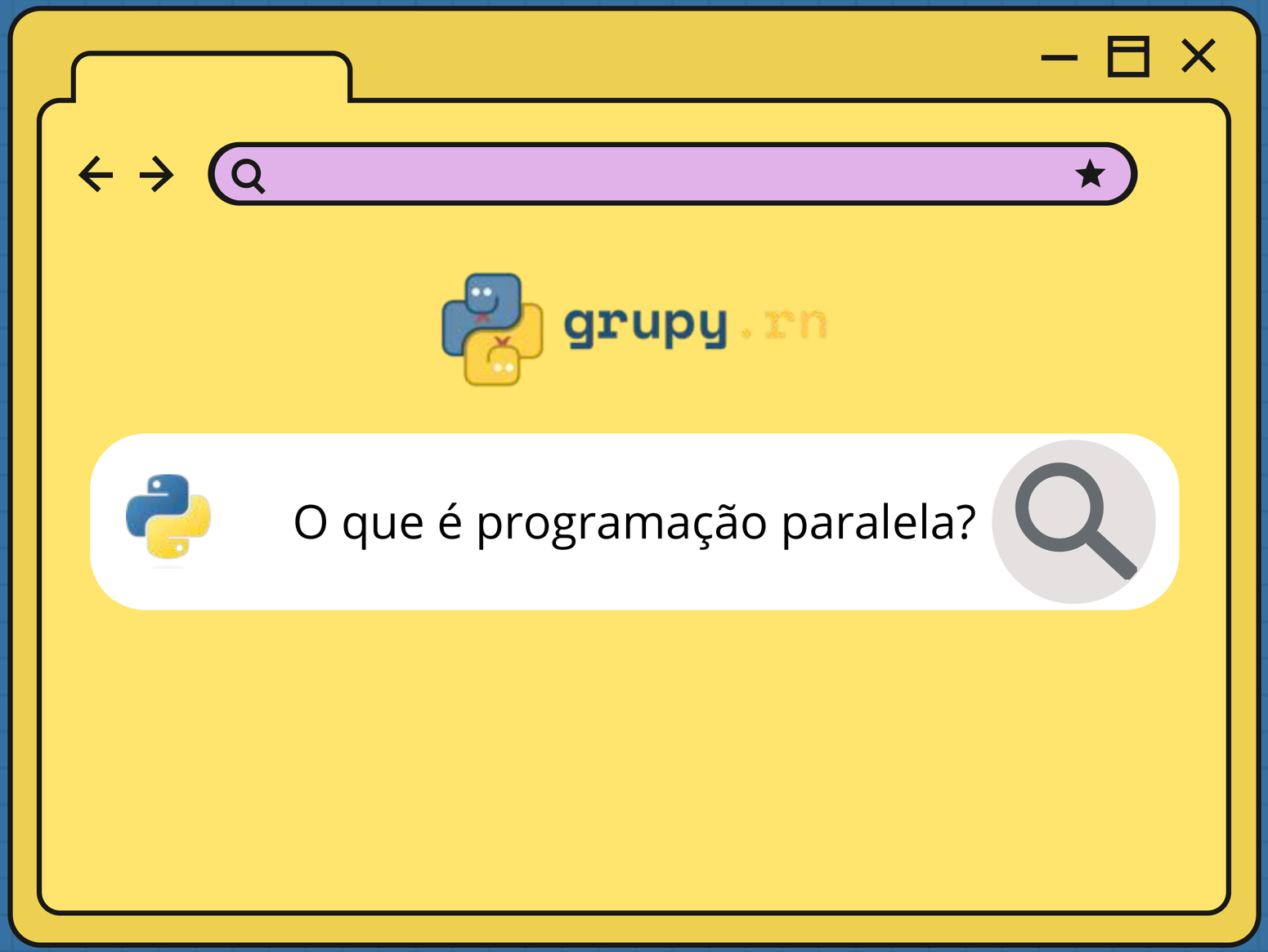
Programação Paralela

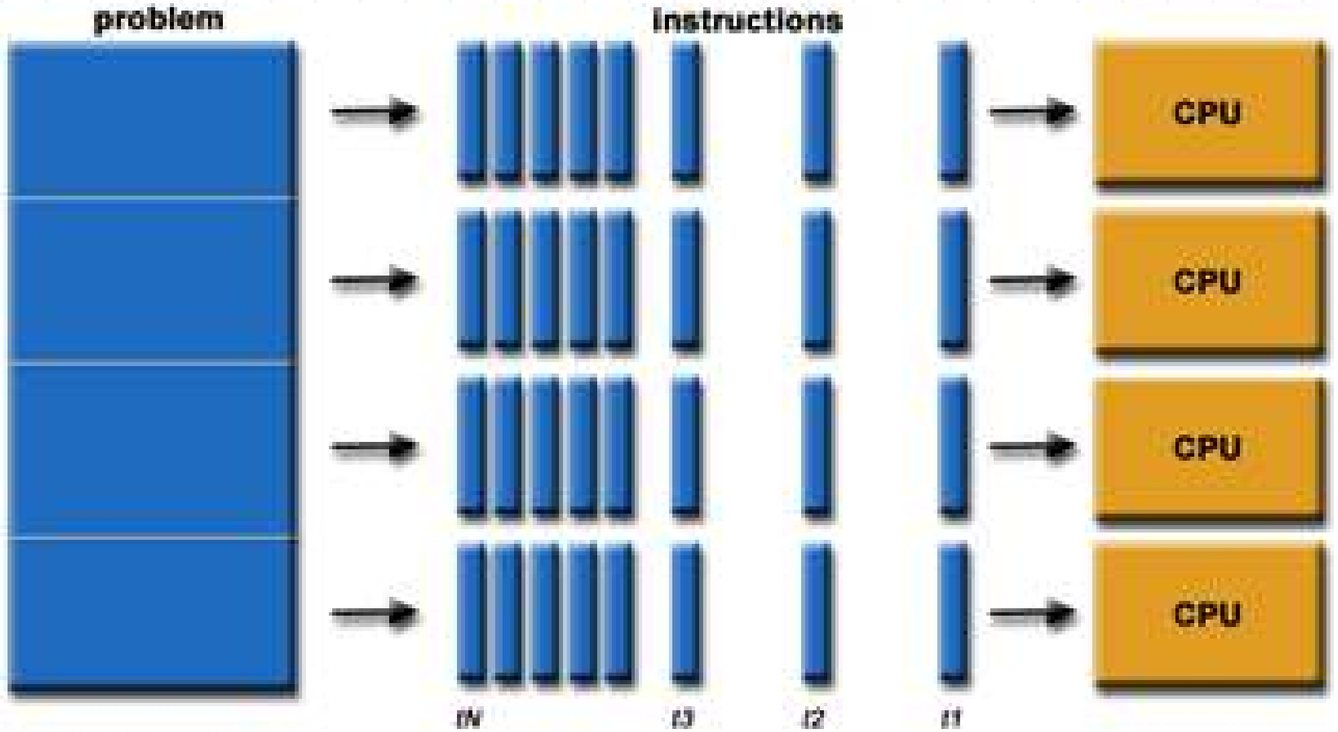
Thread

Concorrência

Paralelismo

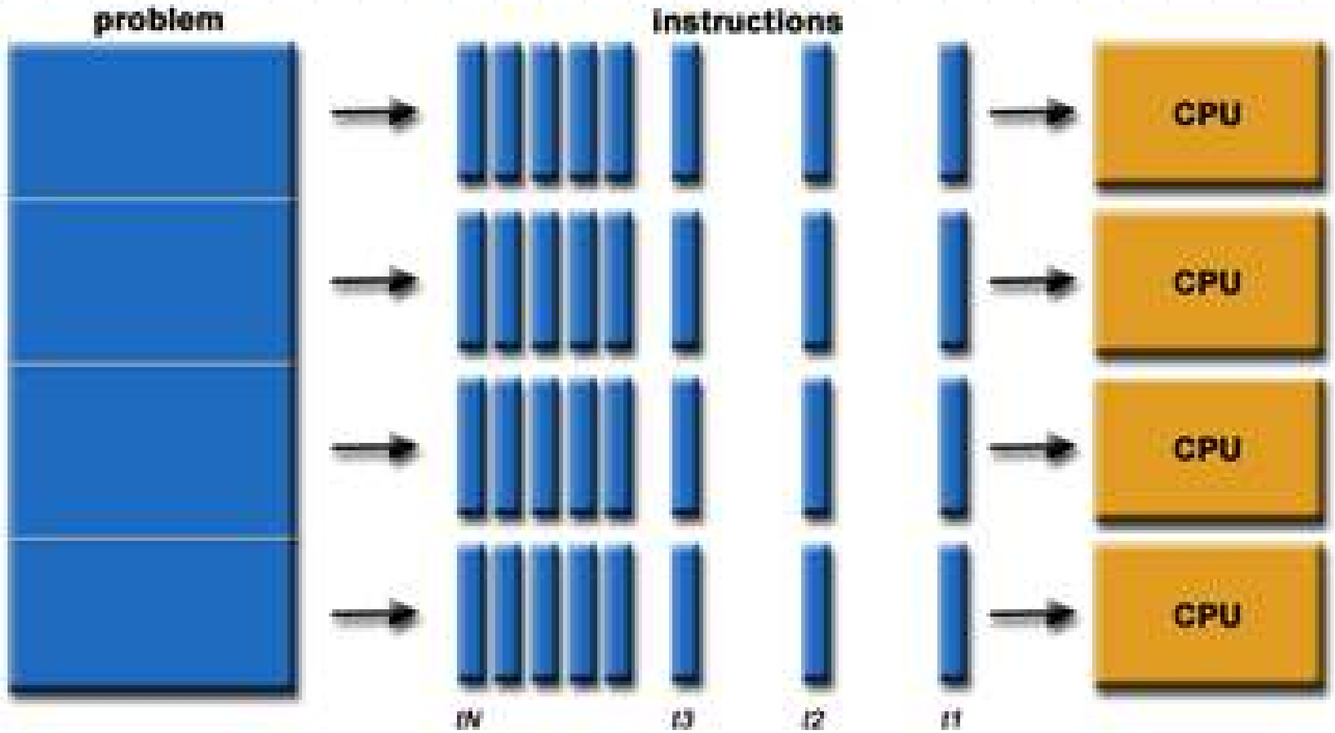
Bônus: Programação Assíncrona


grupy.rn
O que é programação paralela?



Computação Paralela

“É uma forma de computação em que vários cálculos são realizados ao mesmo tempo operando sobre o principio de que problemas grandes podem ser divididos em problemas menores”



Computação Paralela

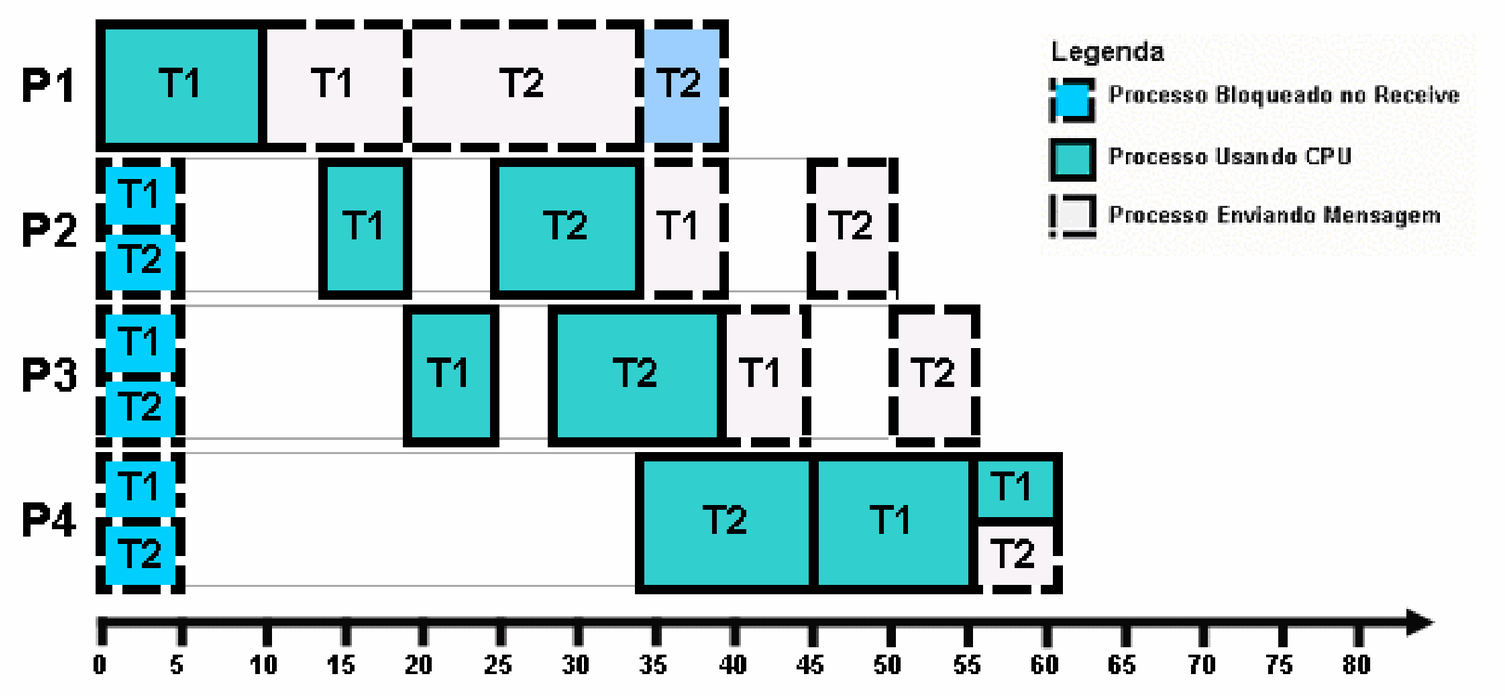
“Nos computadores, o processamento paralelo é o processamento de instruções do programa, dividindo-as entre vários processadores/núcleos com o objetivo de executar um programa em menos tempo”

Como é possível?

Nos primeiros computadores, apenas um programa executava de cada vez. tarefas sem requisitos temporais explícitos, que normalmente executam sem intervenção do usuário

Como é possível?

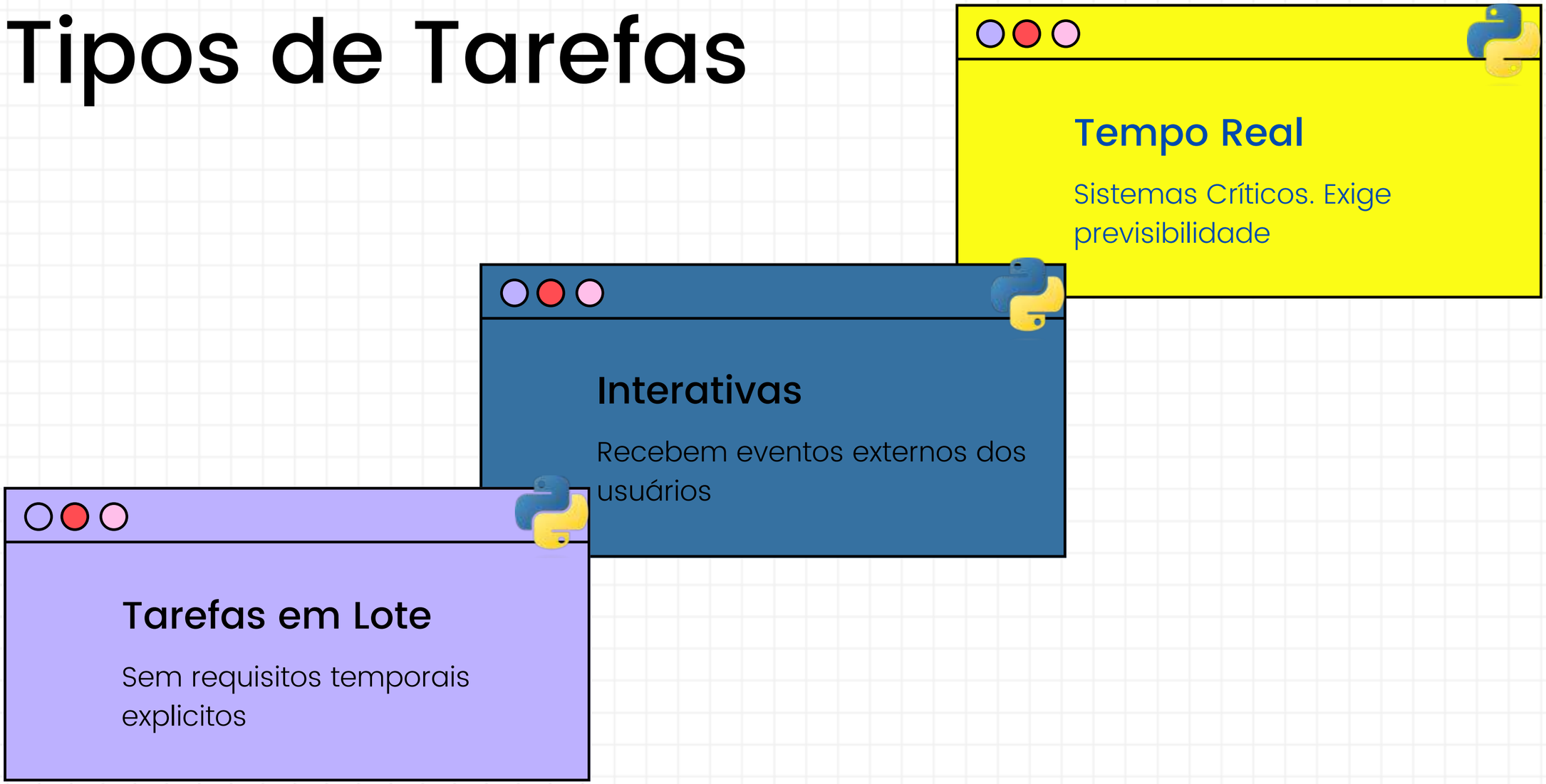
A tecnologia evoluiu permitindo que vários programas executassem “ao mesmo tempo”. Mas isso somente se tornou possível devido ao avanço dos Sistemas Operacionais que passaram a escalonar as tarefas, decidindo quem e quando receberia um tempo para executar algo na CPU.



Escalonador de Tarefas

decide a ordem de execução das tarefas prontas. O algoritmo utilizado no escalonador define o comportamento do sistema operacional, permitindo obter sistemas que tratem de forma mais eficiente e rápida as tarefas a executar, que podem ter características diversas: aplicações interativas, processamento de grandes volumes de dados, programas de cálculo numérico, etc.

Tipos de Tarefas



Por que programação paralela?

Tradicionalmente, a programação paralela foi motivada pela resolução/simulação de problemas fundamentais da ciência/engenharia de grande relevância científica e económica, denominados como Grand Challenge Problems (GCPs).

Por que programação paralela?

Reduzir o tempo necessário para solucionar um problema, Resolver problemas mais complexos e de maior dimensão, Ultrapassar limitações de memória quando a memória disponível num único computador é insuficiente para a resolução do problema

Processos com Mr.Robot

eps1.3_da3m0ns.mp4"

Programa

Um programa é um conjunto de instruções.

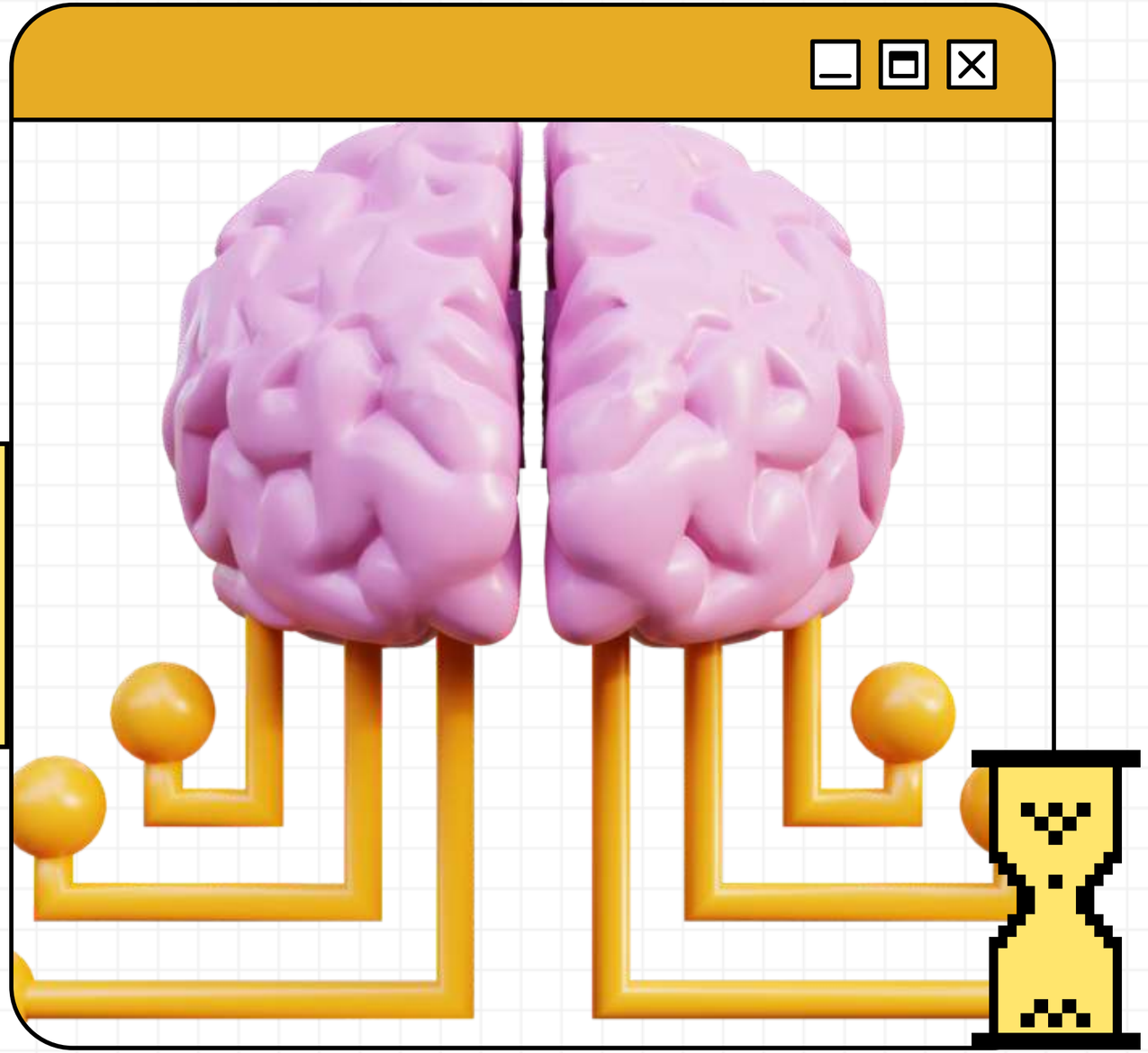
Processo

é uma instância de um programa em execução

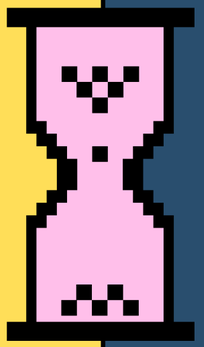
Thread

Uma thread é a menor unidade de execuço

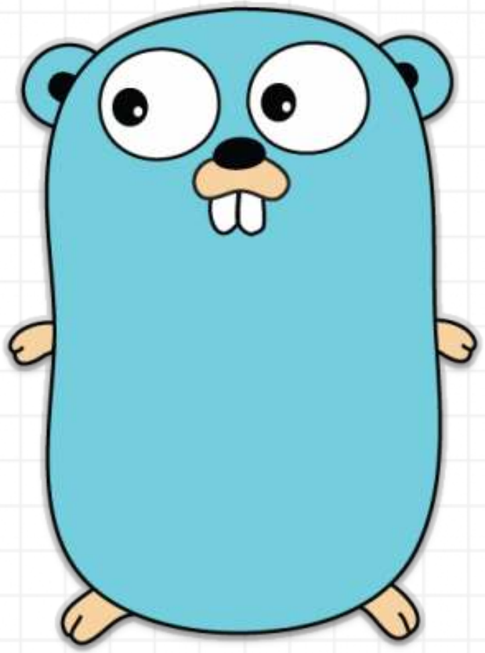
Analogia da mente



Paralelismo x Concorrência



Rob Pike



Concorrência

Concorrência é sobre **lidar** com várias coisas ao mesmo tempo

“”

Paralelismo
é sobre **fazer** várias coisas ao mesmo tempo

Concorrência ou Paralelismo Potencial

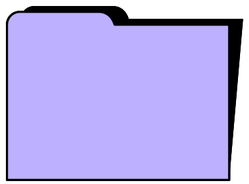
- Concorrência ou paralelismo potencial diz-se quando um programa possui **tarefas** (partes contíguas do programa) que podem ser executadas em qualquer ordem sem alterar o resultado final.

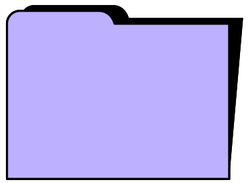
Paralelismo

- Paralelismo diz-se quando as tarefas de um programa são executadas em simultâneo em mais do que um processador.

Threading

Threading





Limitação do GIL

Compartilham o mesmo espaço e estão sujeitas ao GIL

Não recomendado para paralelismo

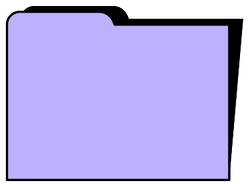
Devido as limitações do GIL o Python não possui

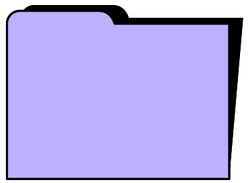
Tarefas em único processo

Ideal para tarefas de iNPUT/OUTPUT

Concurrent.features

Concurrent.features





Permite usar Múltiplos Cores

"Engando" o GIL com novos interpretadores

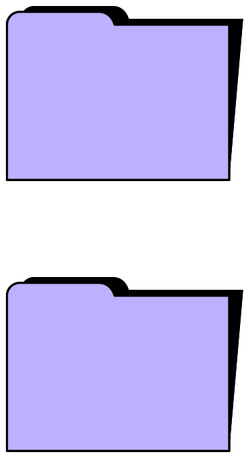
Maneira eficaz de recuperar os dados

cria novos interpretadores como subprocessos do interpretador principal

Execução de funções em Threads

Multiprocessing

Multiprocessing



Processo diferenciado

Cada processo seu próprio espaço na memória e interpretador Python]

Ideal para paralelismo

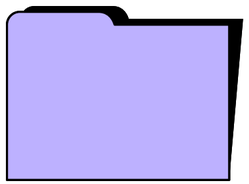
útil quando se tem um grande poder de processamento

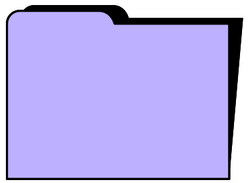
Não é afetado pelo GIL

Pois não compartilham o mesmo memória

Async.IO

Async.IO





orientada a eventos usando corrotinas

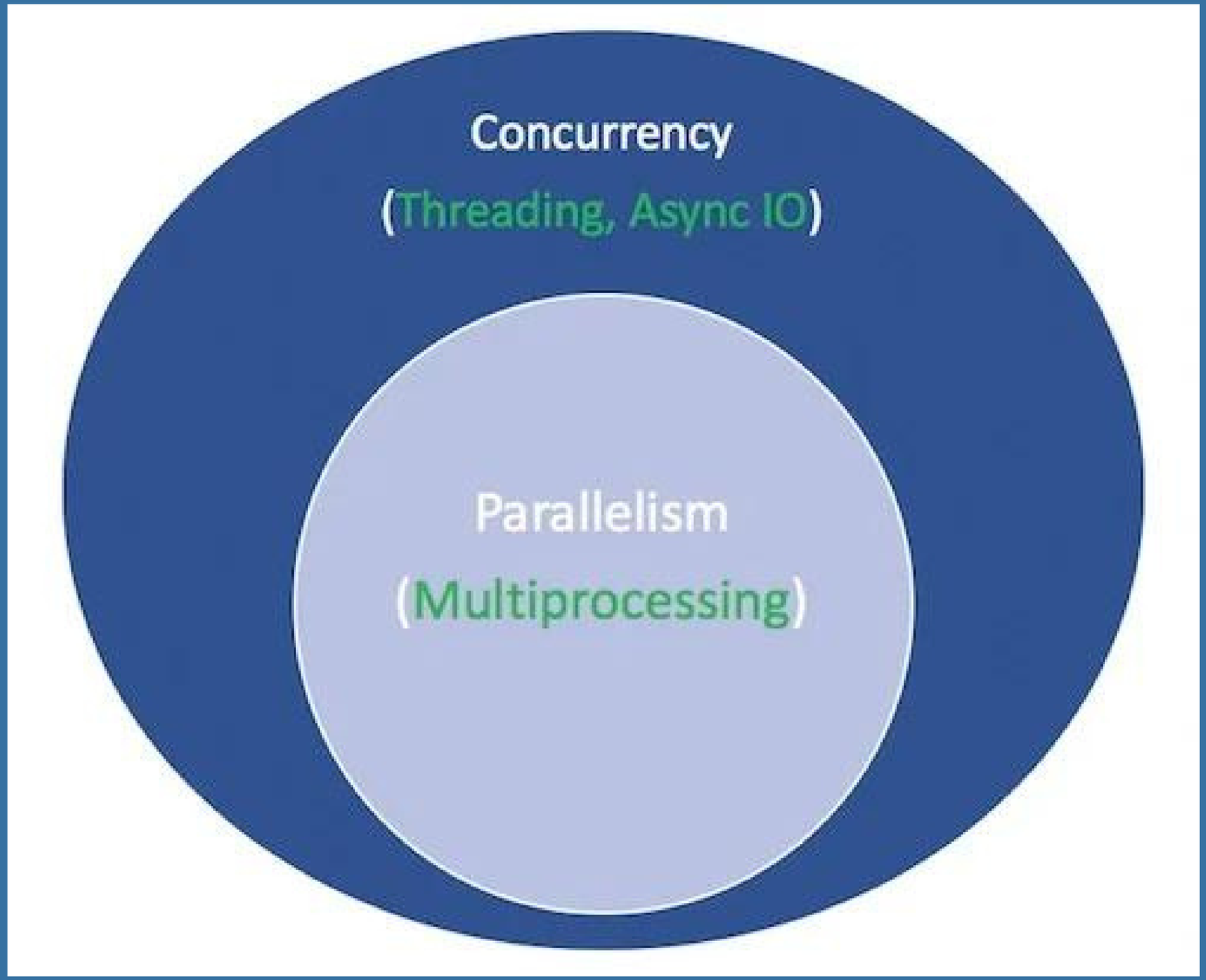
Tarefas de E/S intensiva, como operações de rede, onde a espera bloqueante deve ser minimizada

Usa um único thread

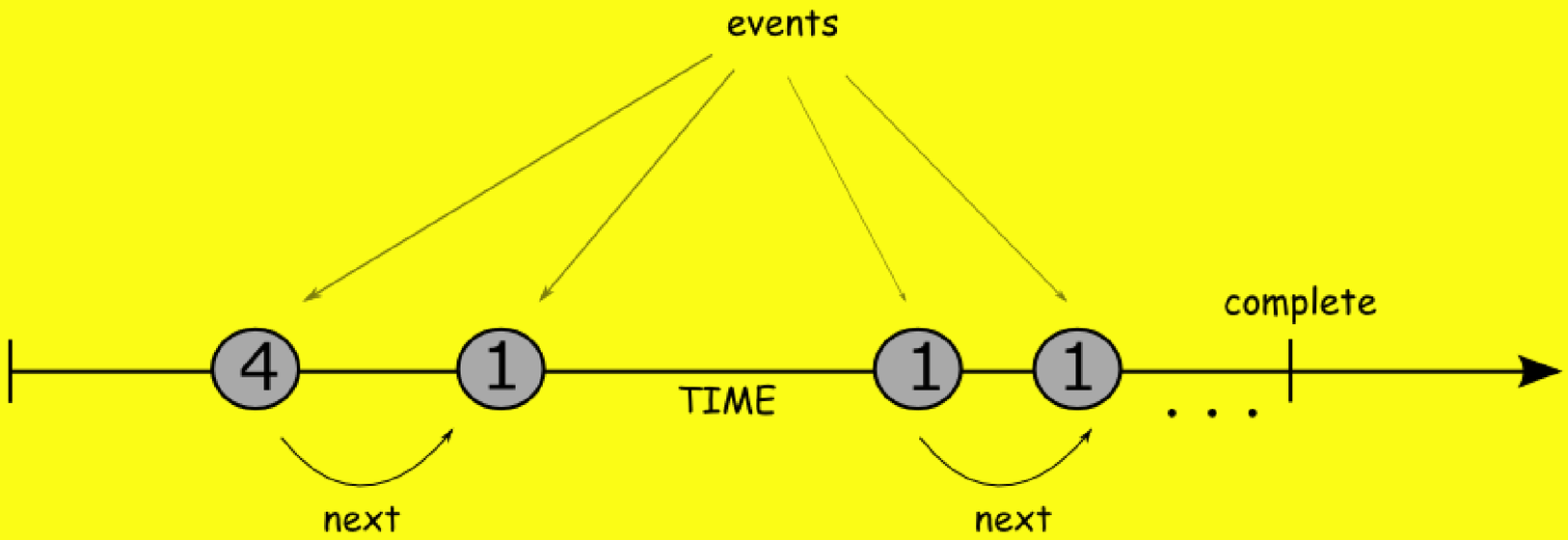
mas permite que tarefas sejam interrompidas durante operações de E/S, aproveitando o mecanismo de E/S não bloqueante

Programação Reativa

Pois não compartilham o mesmo memória


Concurrency
(Threading, Async IO)
Parallelism
(Multiprocessing)

Introduzindo programação reativa




SCAN ME