Geek University

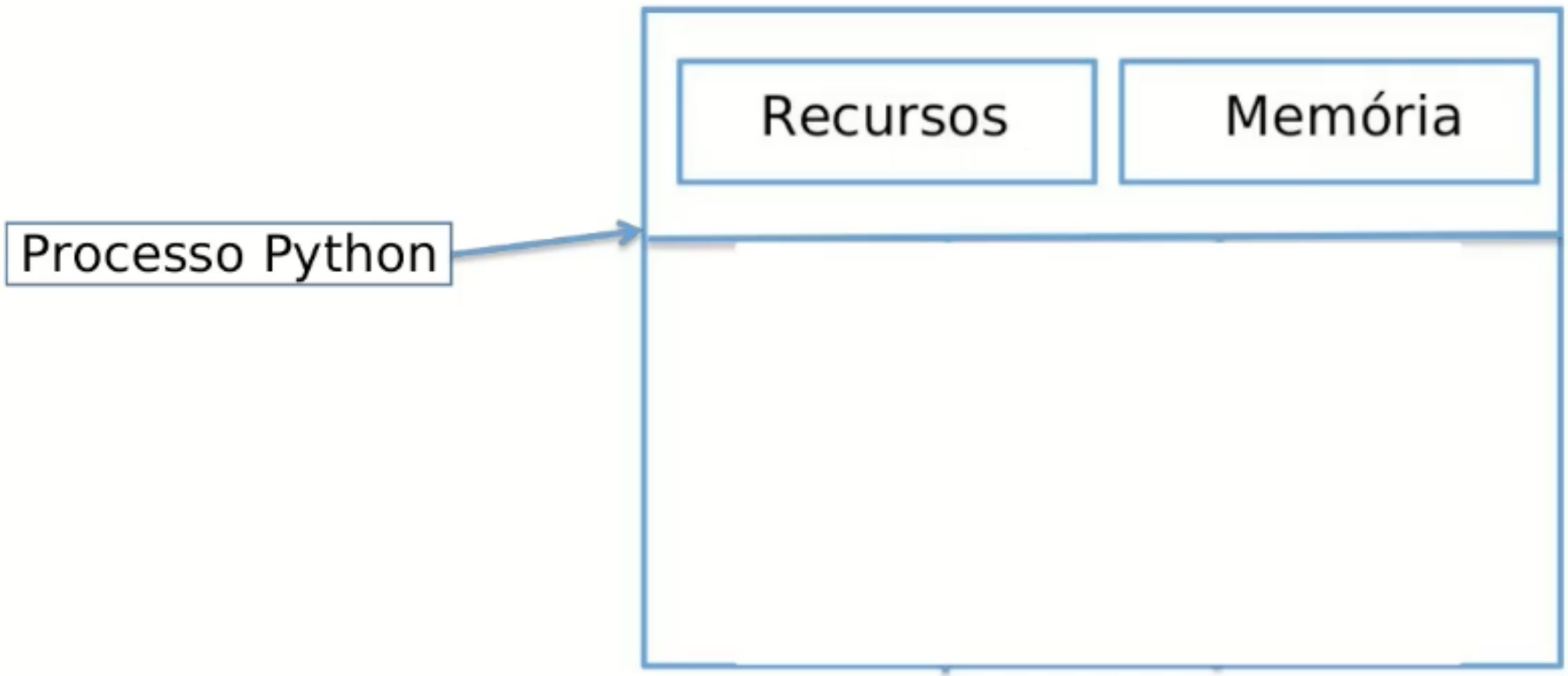
Evolua seu lado geek!

www.geekuniversity.com.br

Processos vs Threads

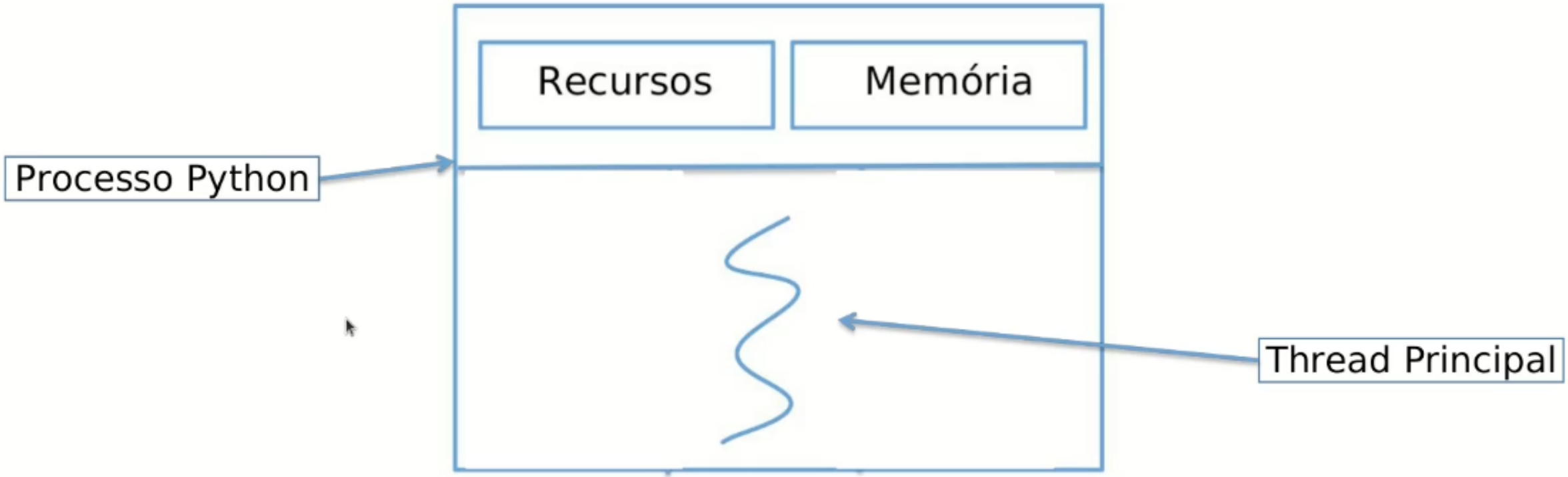
Processos vs Threads

Processo é a instância em execução de um programa.



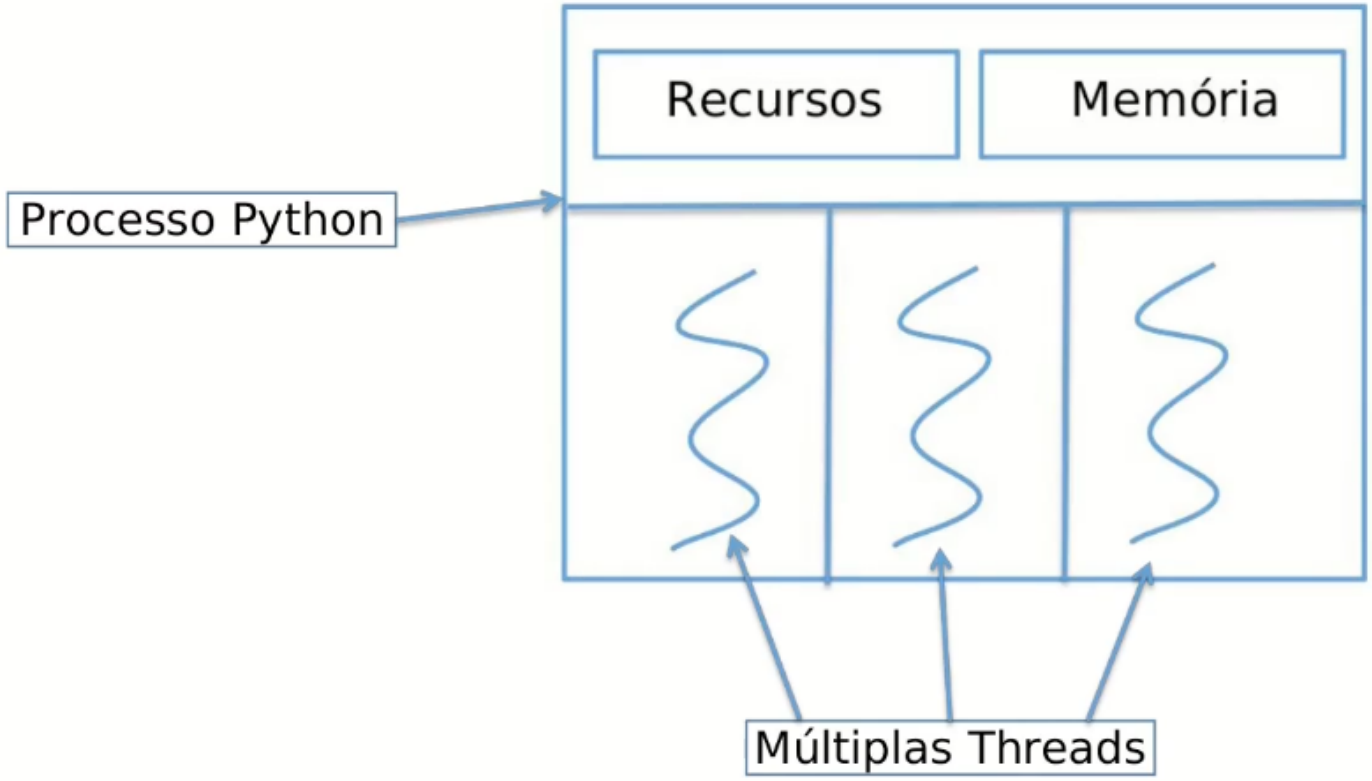
Processos vs Threads

Um processo cria uma thread, chamada de thread principal, para executar o programa.



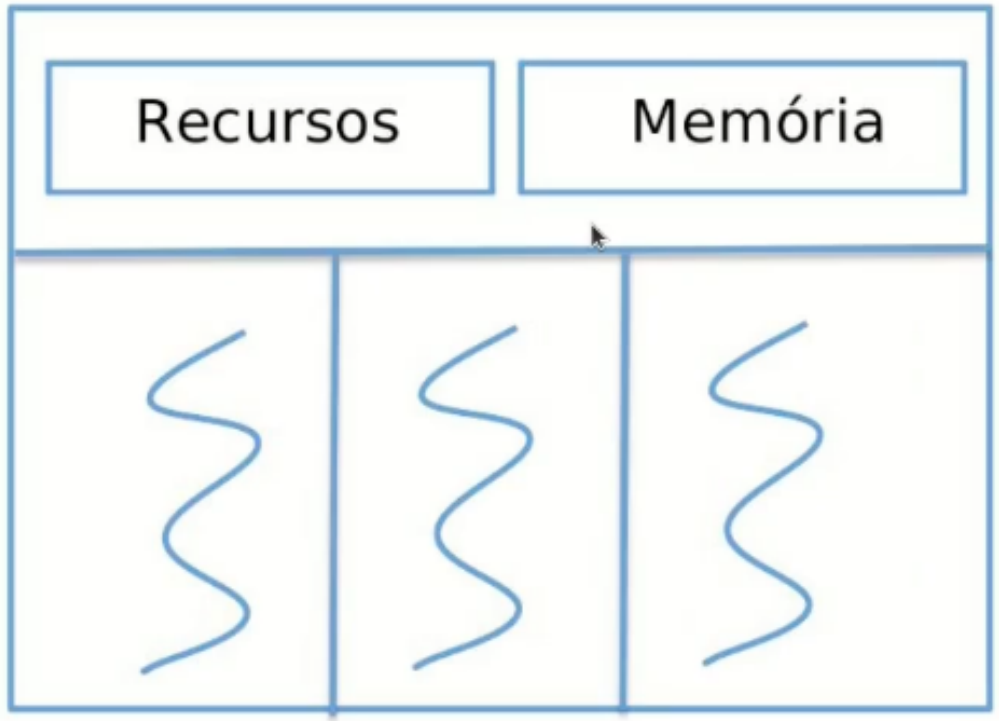
Processos vs Threads

Um processo pode ter múltiplas threads. Qualquer thread além da principal será “filha” da thread principal.



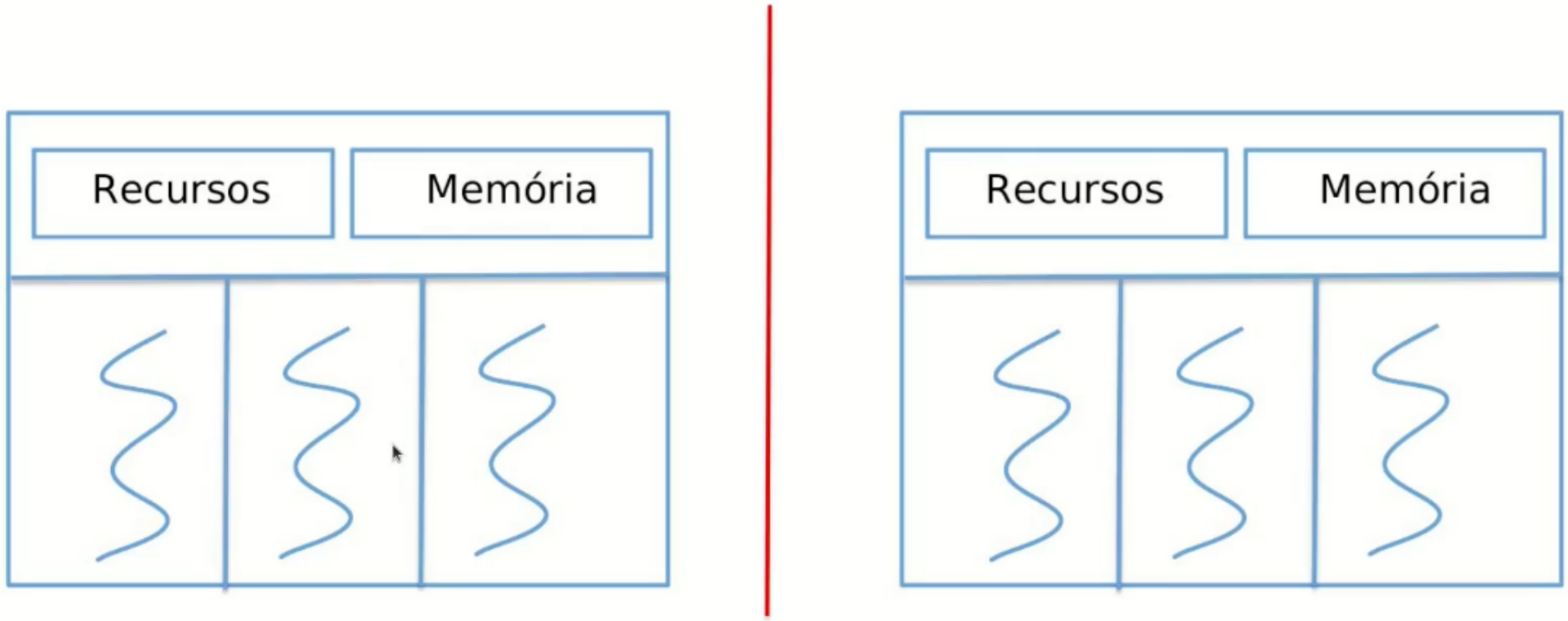
Processos vs Threads

As threads de um processo compartilham recursos e memória entre si.



Processos vs Threads

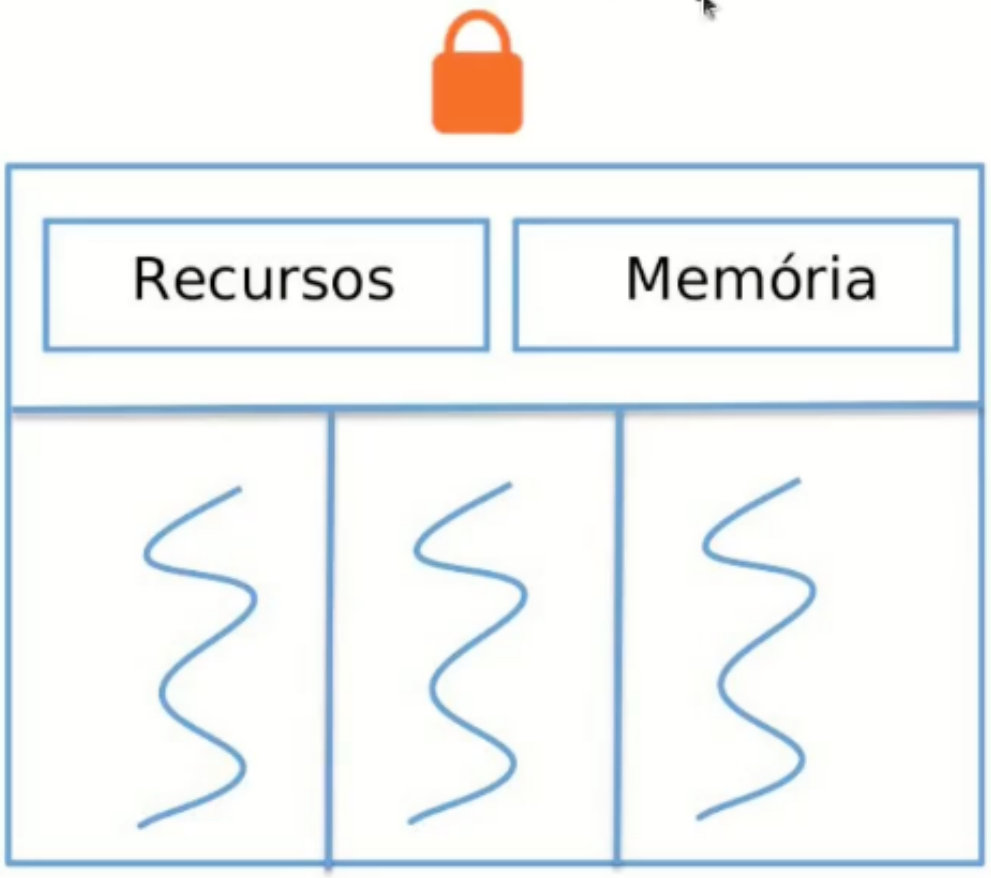
Mas as threads/processos não* compartilham recursos e memória de outros processos/threads.

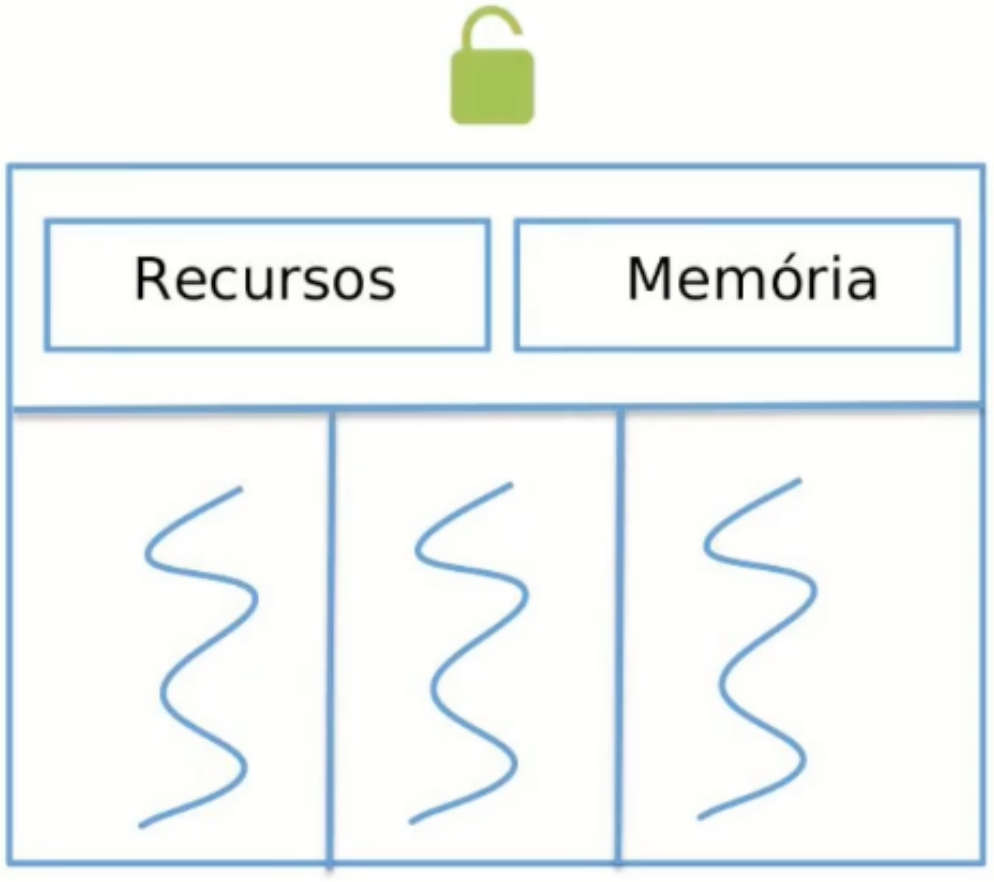


* Existem formas de fazer solicitações ao sistema operacional para acessar memória e recursos de outros processos. Mas este não é nosso caso ou necessidade.

Processos vs Threads

Quando ativamos um lock (acquire), estamos avisando que nenhuma thread pode acessar determinado recurso até que o lock seja liberado (release).

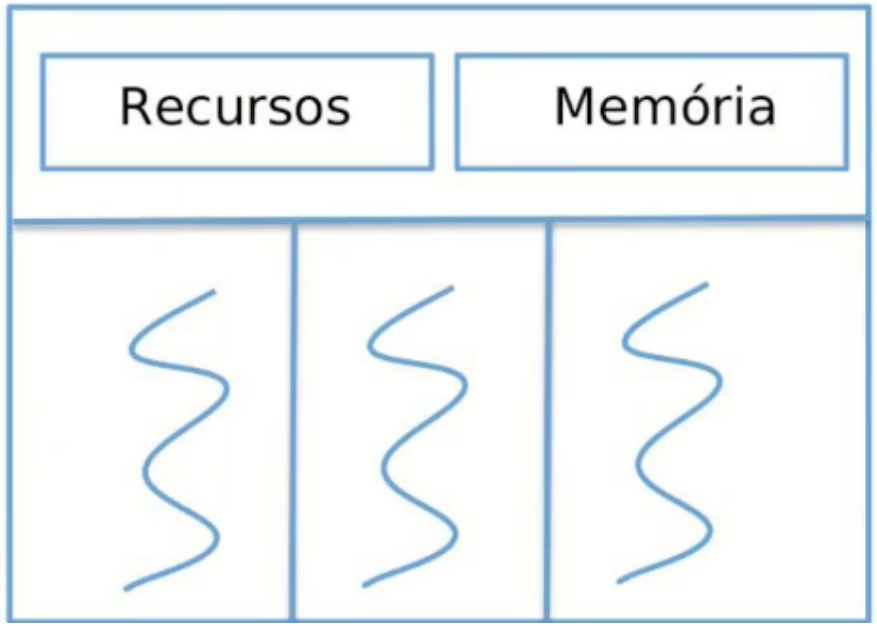




Processos vs Threads

O que deve ficar claro nesta seção é que o processo “nasce” primeiro e somente depois é criada uma thread principal. Todo processo terá pelo menos uma thread para executar o programa.

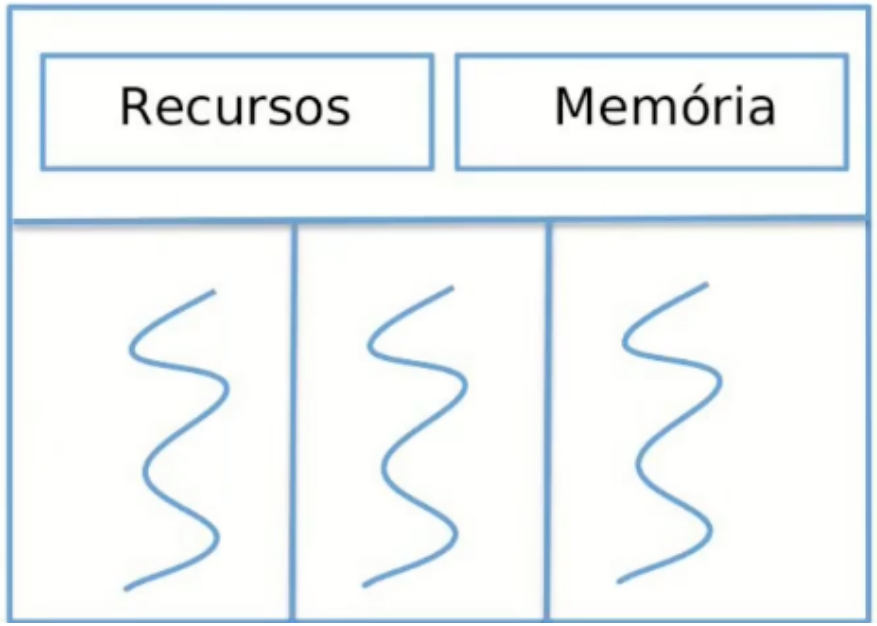
Separando a execução de um programa em múltiprocessos ao invés de multithread evitamos que o GIL bloqueie a execução em paralelo.



Processos vs Threads

O que deve ficar claro nesta seção é que o processo “nasce” primeiro e somente depois é criada uma thread principal. Todo processo terá pelo menos uma thread para executar o programa.

Separando a execução de um programa em múltiprocessos ao invés de multithread evitamos que o GIL bloqueie a execução em paralelo.



Na próxima aula iremos conhecer a API de multiprocessamento do Python…

Geek University

Evolua seu lado geek!

www.geekuniversity.com.br