

# Programação Paralela e Distribuída

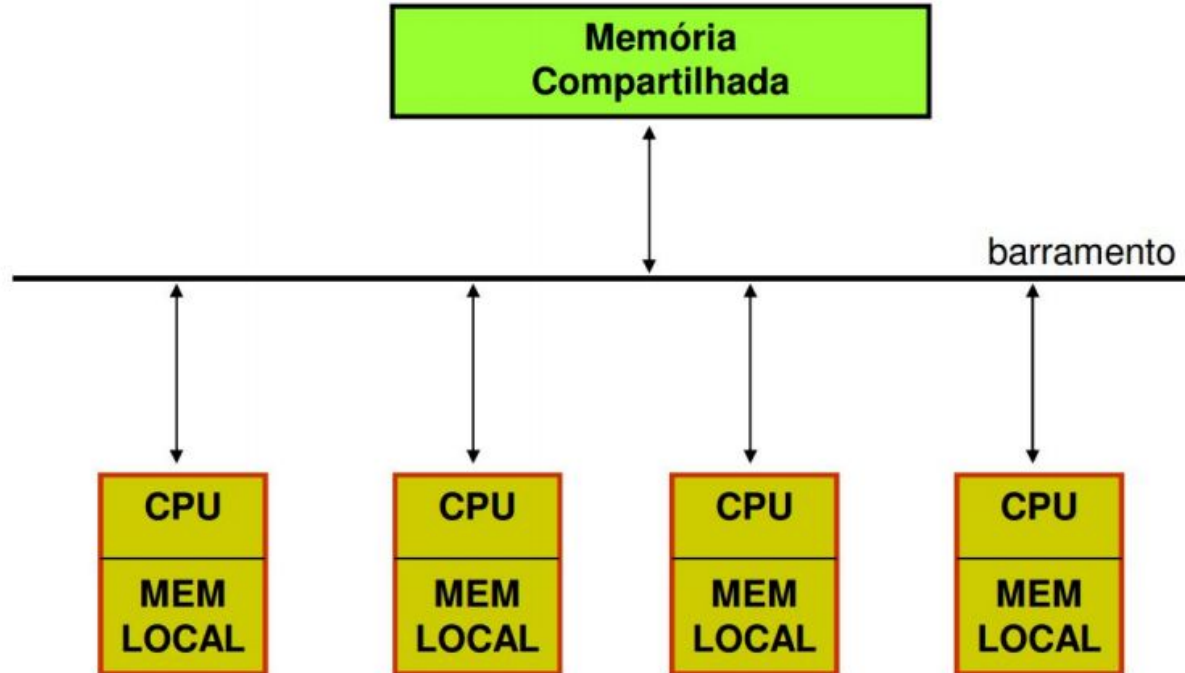
Prof. Hugo Alberto Perlin

# Programação Paralela

- Consiste em executar simultaneamente várias simultaneamente várias partes de uma mesma aplicação
- Tornou-se possível a partir do desenvolvimento de sistemas operacionais multi-tarefa, multi-thread e paralelos
- Aplicações são executadas paralelamente:
  - Em um mesmo processador
  - Em uma máquina multiprocessada
  - Em um grupo de máquinas interligadas que se comporta como uma só máquina

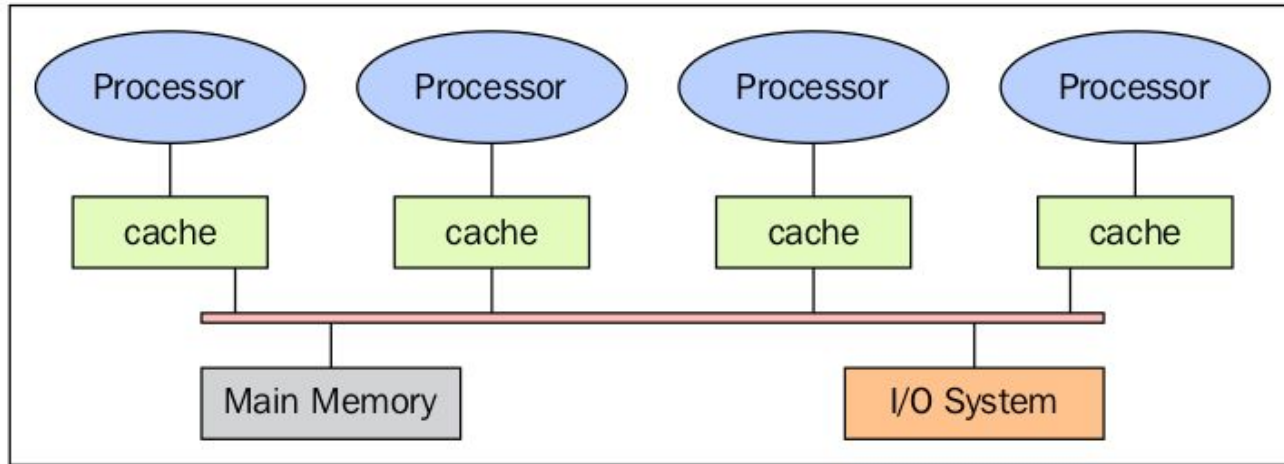
# Sistema Paralelo

*Múltiplos processadores conectados por uma memória compartilhada*



# Programação Paralela – Arquitetura dos Sistemas

- Memória Compartilhada



# Sistema Distribuído

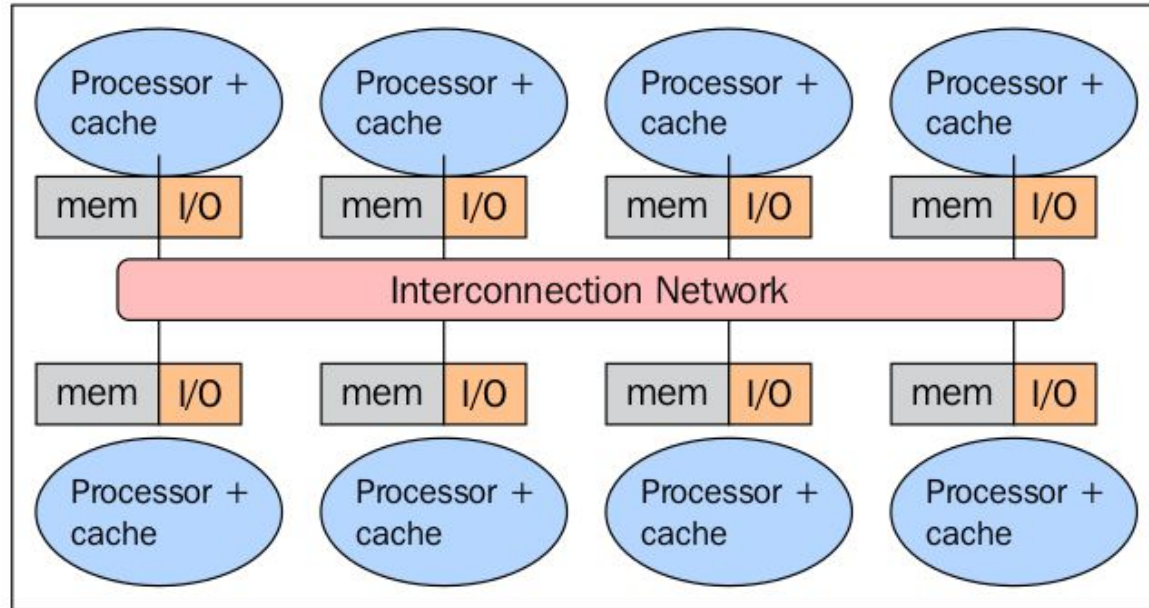
*Múltiplos processadores conectados por uma rede de comunicação*



Segundo (Garg, 2004), são sistemas compostos por múltiplos processadores conectados por uma rede de comunicação, sendo a rede de comunicação uma LAN (Ethernet) ou WAN (Internet). Neste tipo de sistema os processadores se comunicam por troca de mensagens (não por memória compartilhada)

# Programação Paralela – Arquitetura dos Sistemas

- Memória Distribuída

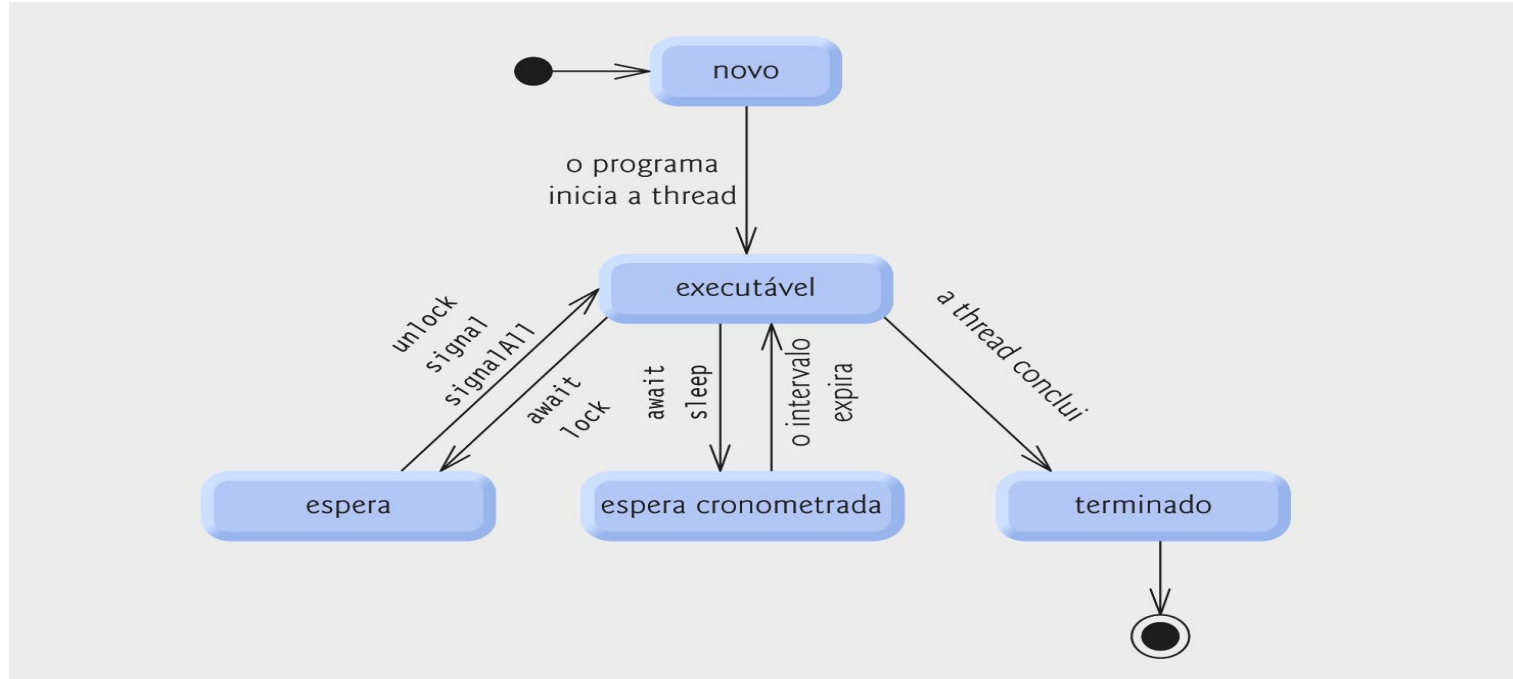


## Modelo Multi-thread – Threads

- Todo processo possui ao menos um fluxo de execução (thread)
- Nos SOs modernos é possível criar múltiplas threads
- Cada thread é parte de um processo
  - Compartilham recursos: memória e arquivos abertos
- Cada thread tem seu estado e segue um ciclo de vida
- Por que utilizar múltiplas threads?

# Thread

- Ciclo de vida de uma thread





## Modelo Multi-thread – Threads

- Threads são consideradas processos leves
- A criação e destruição é mais rápida, gerando menos overhead
- A troca de contexto entre threads é mais rápida que a troca de contexto entre processos
- Permitem um melhor uso de sistemas multiprocessados

## Condição de corrida (race condition)

- Se o desenvolvedor não gerenciar corretamente o acesso a recursos compartilhados, resultados indesejados podem ocorrer
- Lembre como ocorre o processo de leitura/escrita na memória
- Exemplo

# Modelo Multi-Processos

- O paralelismo pode ser obtido tanto por meio de threads quanto por meio de processos
- No caso do Python, o uso de processos-filhos, permite executar código em múltiplos núcleos de processamento ao mesmo tempo
- Apesar de ser uma forma que possua um custo computacional maior
- É uma forma de contornar uma limitação de projeto do Python chamada GIL (Global Interpreter Lock)