

Threads



Roteiro

- **Conceito**
- Modelos
- Exercícios

Conceito

- De forma geral, cada **fluxo de execução do sistema**, seja associado a um processo ou no interior do núcleo, é denominado **thread**;
- Threads executando dentro de um processo são chamados de **threads de usuário**.

Conceito

- Cada thread de usuário corresponde a uma tarefa a ser executada dentro de um processo;
- Por sua vez, os fluxos de execução reconhecidos e gerenciados pelo núcleo do sistema operacional são chamados de **threads de núcleo**;
- Os threads de núcleo representam tarefas que o núcleo deve realizar.

Conceito

- Os sistemas operacionais mais antigos **não ofereciam suporte a threads** para a construção de aplicações;
- Sem poder contar com o sistema operacional, os desenvolvedores de aplicações contornaram o problema construindo **bibliotecas** que permitiam criar e gerenciar threads dentro de cada processo, sem o envolvimento do núcleo do sistema.

Conceito

- Usando essas bibliotecas, uma aplicação pode lançar vários threads conforme sua necessidade, mas o núcleo do sistema irá sempre perceber (e gerenciar) **apenas um fluxo de execução dentro de cada processo.**

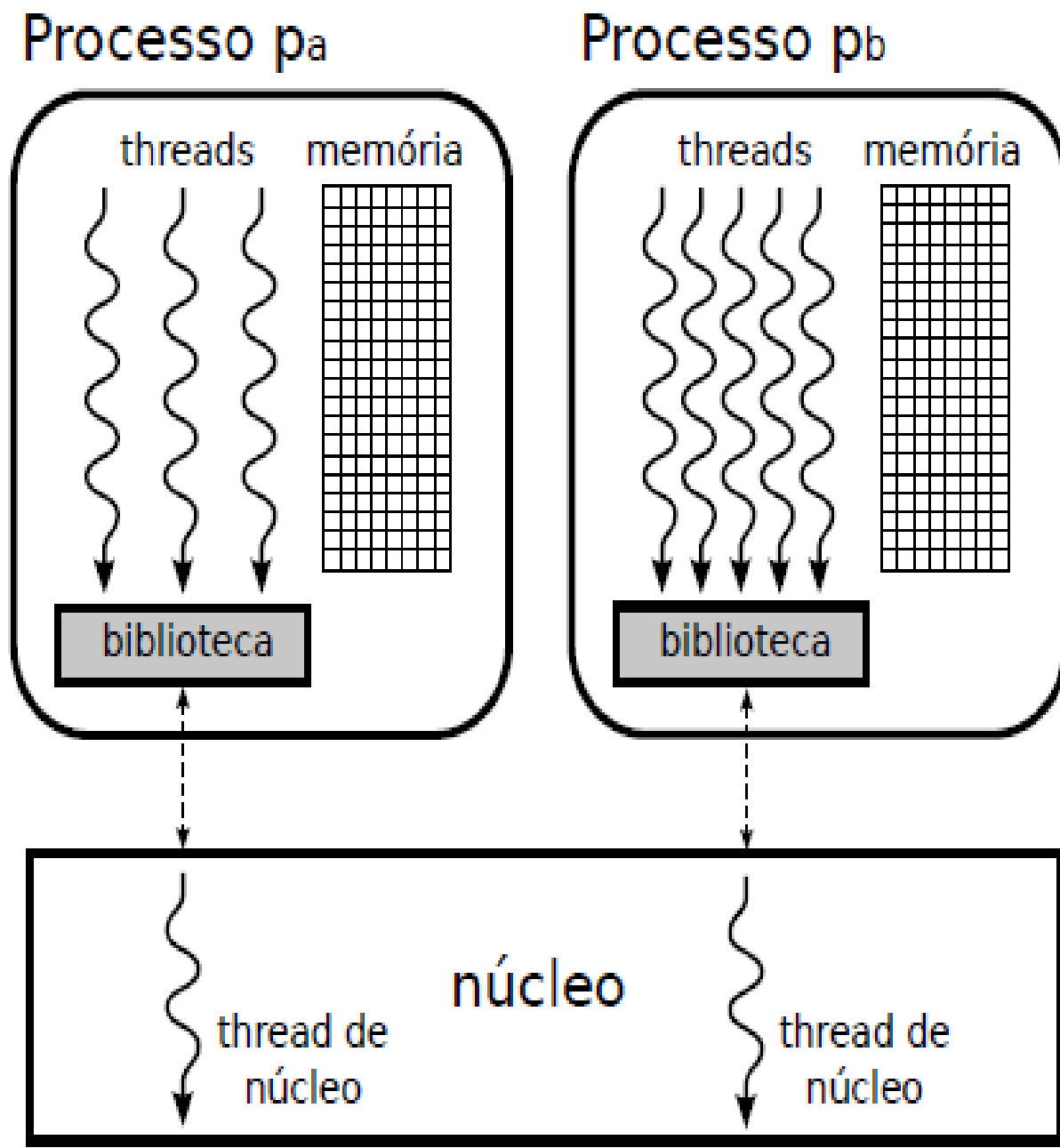
Roteiro

- Conceito
- **Modelos**
- Exercícios

Modelo de Threads N:1

- N threads no processo, mapeados em um único thread de núcleo;
- É muito utilizado, por ser leve e de fácil implementação;
- Como o núcleo somente considera uma thread, a carga de gerência imposta ao núcleo é pequena e não depende do número de threads dentro da aplicação.

Modelo de Threads N:1



Modelo de Threads N:1

- O modelo de threads N:1 apresenta **problemas** em algumas situações, sendo o mais grave deles relacionado às **operações de entrada/saída**. Como essas operações são intermediadas pelo núcleo, se um thread de usuário solicitar uma operação de E/S (recepção de um pacote de rede, por exemplo) o thread de núcleo correspondente será **suspenso** até a conclusão da operação, fazendo com que todos os threads de usuário associados ao processo **parem de executar** enquanto a operação não for concluída.

Modelo de Threads N:1

- Outro problema desse modelo diz respeito à **divisão de recursos entre as tarefas**;
- O núcleo do sistema divide o tempo do processador entre os fluxos de execução que ele conhece e gerencia: **as threads de núcleo**.
- Assim, uma aplicação com 100 threads de usuário irá receber o **mesmo tempo de processador** que outra aplicação com apenas um thread (considerando que ambas as aplicações têm a mesma prioridade).

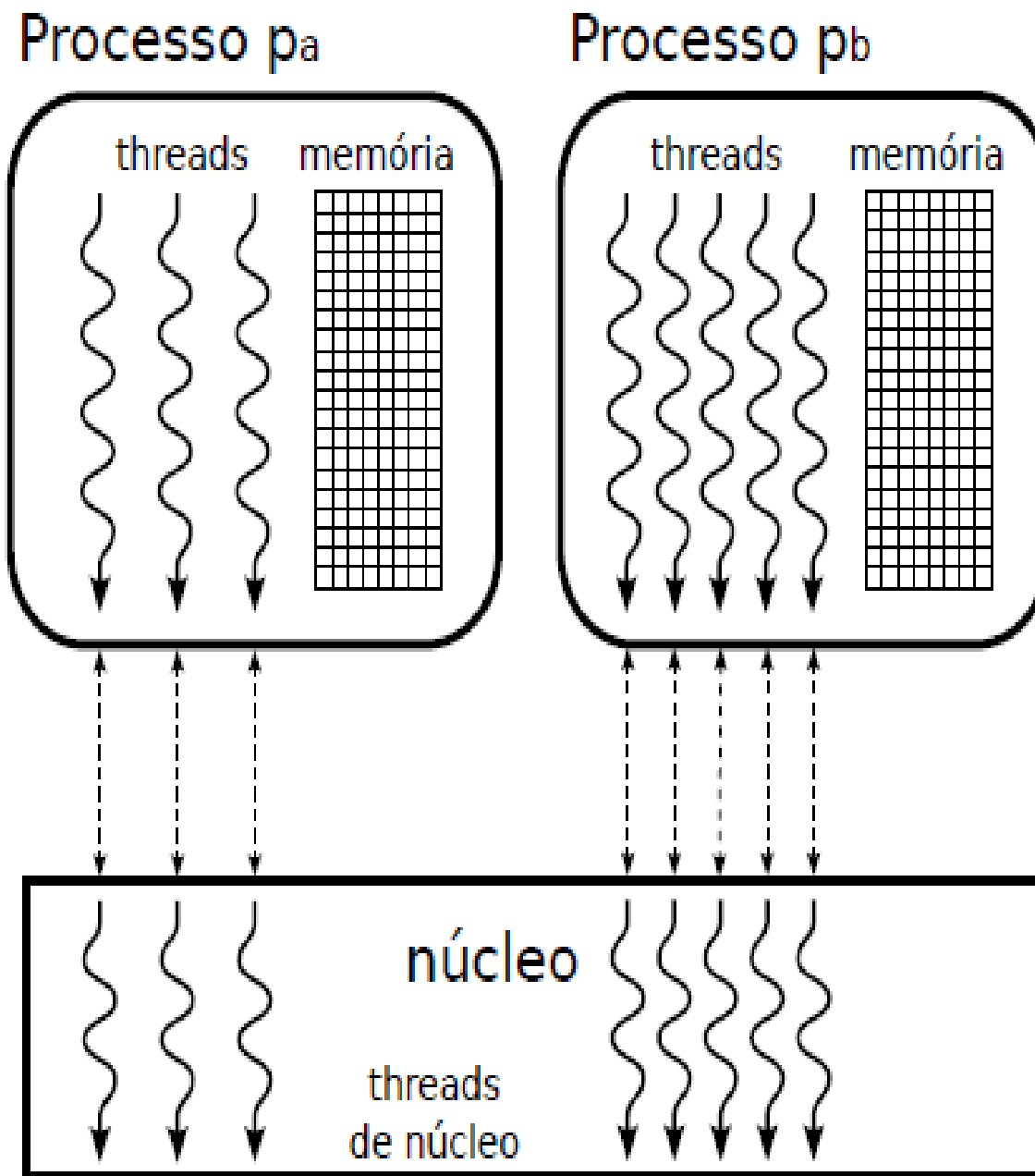
Modelo de Threads 1:1

- A necessidade de suportar aplicações com vários threads (multithreaded) levou os desenvolvedores de sistemas operacionais a incorporar a **gerência dos threads de usuário** ao núcleo do sistema;
- Para cada thread de usuário foi então definido um thread correspondente dentro do núcleo, suprimindo com isso a necessidade de **bibliotecas de threads**.

Modelo de Threads 1:1

- Caso um *thread* de usuário solicite uma operação bloqueante (leitura de disco ou recepção de pacote de rede, por exemplo), somente seu respectivo thread de núcleo será suspenso, **sem afetar os demais threads**.
- Caso o *hardware* tenha mais de um processador, mais *threads* da mesma aplicação podem **executar ao mesmo tempo**, o que não era possível no modelo anterior.

Modelo de Threads 1:1



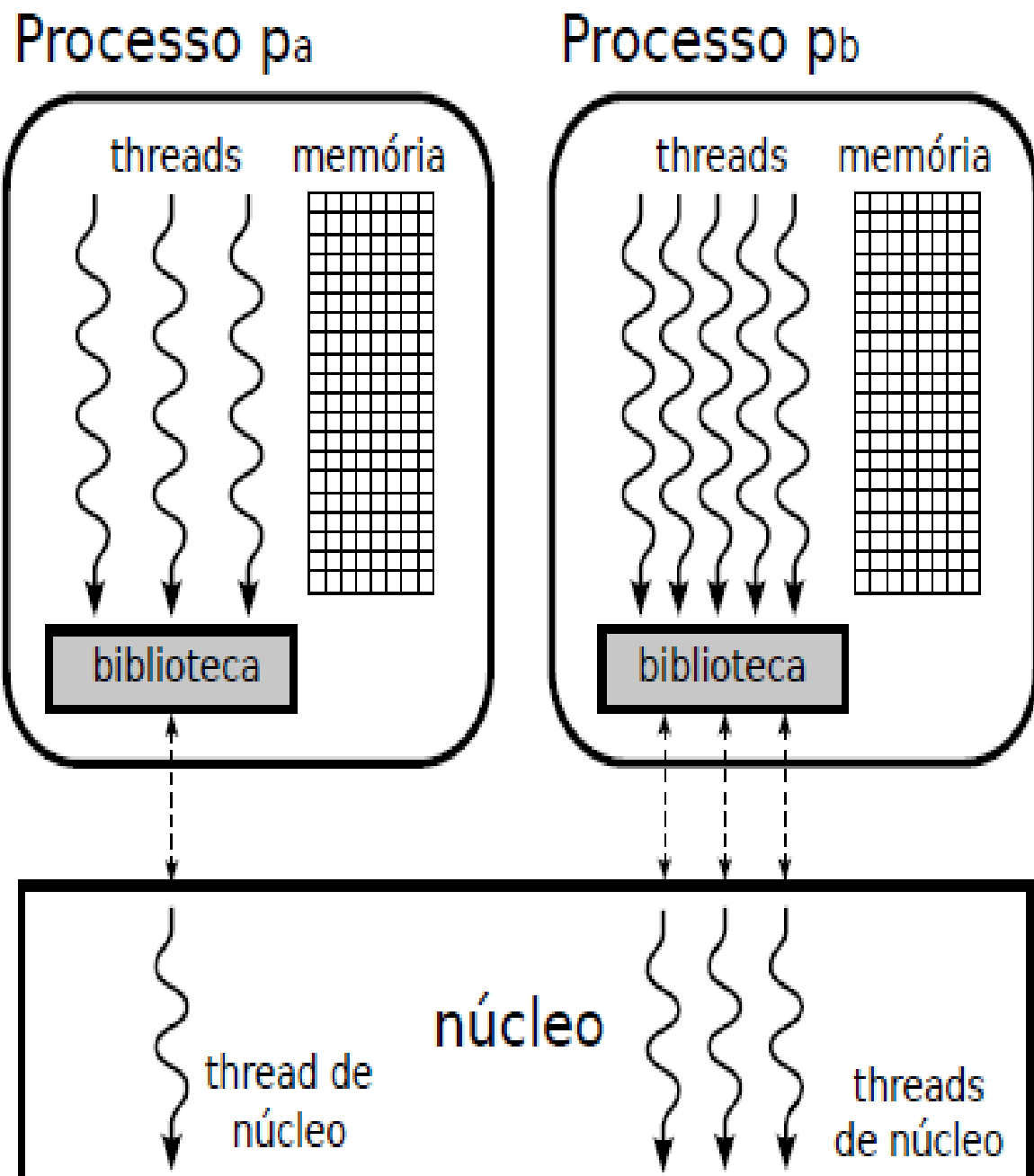
Modelo de Threads 1:1

- O modelo é adequado para a maioria das situações e atende bem às necessidades das aplicações interativas e servidores de rede.
- É **pouco escalável**: a criação de um grande número de threads impõe uma **carga** significativa ao núcleo do sistema, inviabilizando aplicações com muitas tarefas (como grandes servidores Web e simulações de grande porte).

Modelo de Threads N:M

- Alguns sistemas operacionais implementam um **modelo híbrido**, que agrega características dos modelos anteriores;
- Nesse novo modelo, uma biblioteca gerencia um conjunto de threads de usuário (dentro do processo), que é **mapeado** em um ou mais threads do núcleo.
- Essa abordagem híbrida é denominada Modelo de Threads N:M, onde N threads de usuário são mapeados em M threads de núcleo.

Modelo de Threads N:M



Modelo de Threads N:M

- Ele alia as vantagens de maior **interatividade** do modelo 1:1 à maior **escalabilidade** do modelo N:1;
- Como desvantagens desta abordagem podem ser citadas sua **complexidade de implementação** e **maior custo de gerência dos threads** de núcleo, quando comparado ao modelo 1:1.

Roteiro

- Conceito
- Modelos
- **Exercícios**

Exercícios

- 1) Um processo é representado no sistema operacional, por um bloco de controle do processo. Que tipo de informações normalmente está em um Processo?
- 2) Nos sistemas operacionais, o escalonamento de processos consiste:
 - a) Priorizar o processo a ser executado.
 - b) Alterar a ordem dos processos para utilização da CPU e demais recursos.
 - c) Selecionar um processo da fila de ready e alocar a CPU para o mesmo.
 - d) Transferir um processo da fila de wait para a fila de ready.
 - e) Executar processos mais demorados antes dos mais rápidos.

Exercícios

- 3) Alguns sistemas operacionais permitem que seus processos criem múltiplos threads de execução. Em operação normal, o que é previsto que os threads de um mesmo processo do sistema operacional compartilhem?
- a) Arquivos
 - b) Contador
 - c) Assíncrono de Instruções
 - d) Heap da Pilha
 - e) Variáveis locais de cada thread
 - f) Contador de instrução