

Sistemas Operacionais

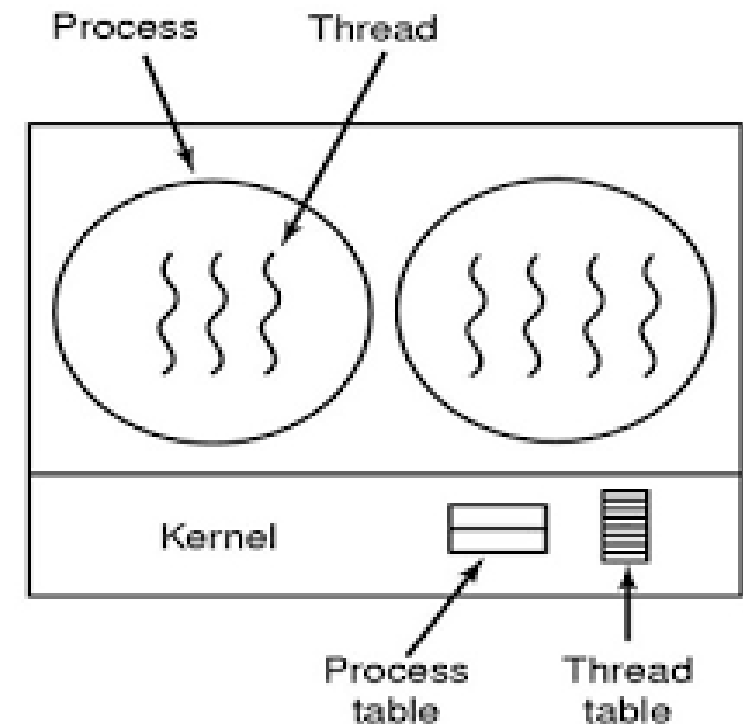
Prof. Dr. Helder Oliveira

Plano de Aula

- Introdução às Threads
- Aplicações com múltiplas Threads
- Concorrência e Compartilhamento
- Modelos de Sistemas

Thread

- Um thread é um tipo de processo leve, o qual, por sua vez, é um programa de execução.
- Linha de execução.
- Mesmo espaço de endereçamento.
- Executando quase em Paralelo.

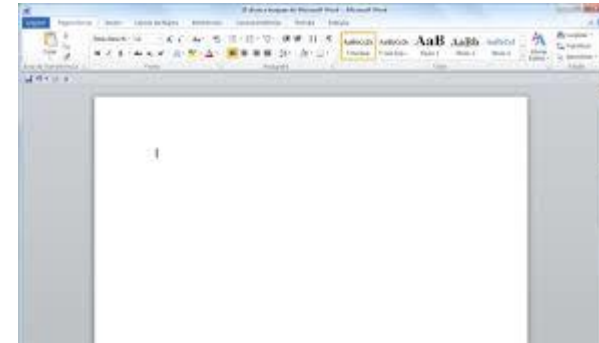


Threads

- Por que alguém iria querer ter um tipo de processo dentro de um processo?
- Há várias razões para a existência desses “miniprocessos”, chamados **threads**.
 - Decompor aplicação.
 - Mais rápidos.
 - Sobreposição de atividades.
 - Útil quando paralelismo real é possível.

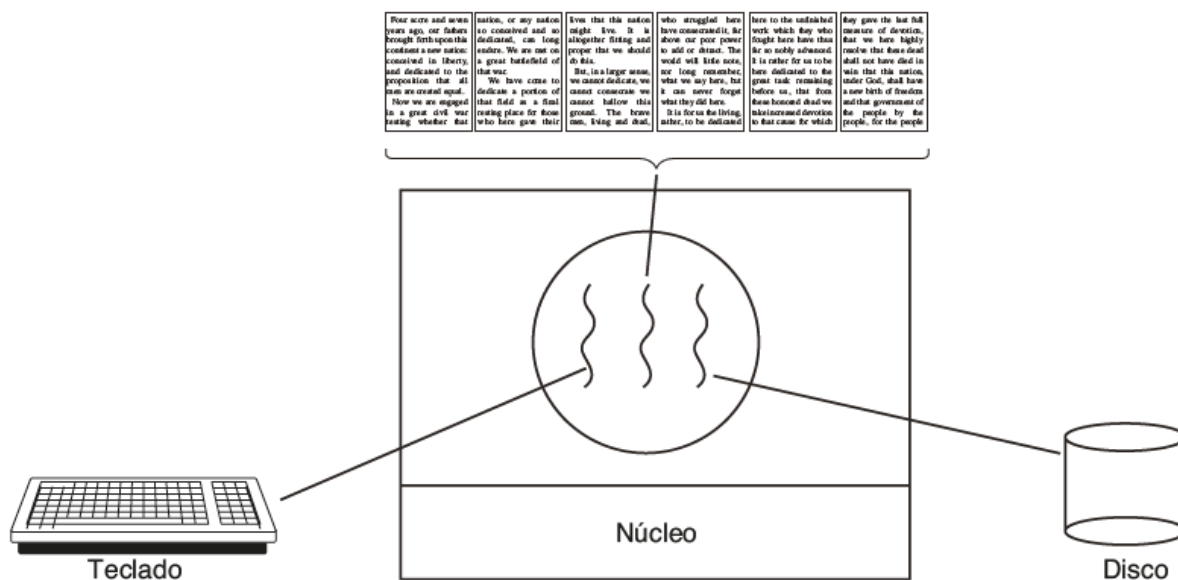
Threads – Exemplo concreto

- Imagine um processador de texto.
 - Páginas exatamente como será impressa.
- Suponha que um usuário esteja escrevendo um livro.
 - Dividir em capítulos ou único documento?
 - **Dividir em capítulos**
 - Se precisar editar algo globalmente, teria que editar cada capítulo individual
 - **Único documento**
 - O processador de texto terá que reformatar o livro inteiro se houver uma alteração no início.
 - Solução:
 - **Threads**



Threads

FIGURA 2.7 Um processador de texto com três threads.

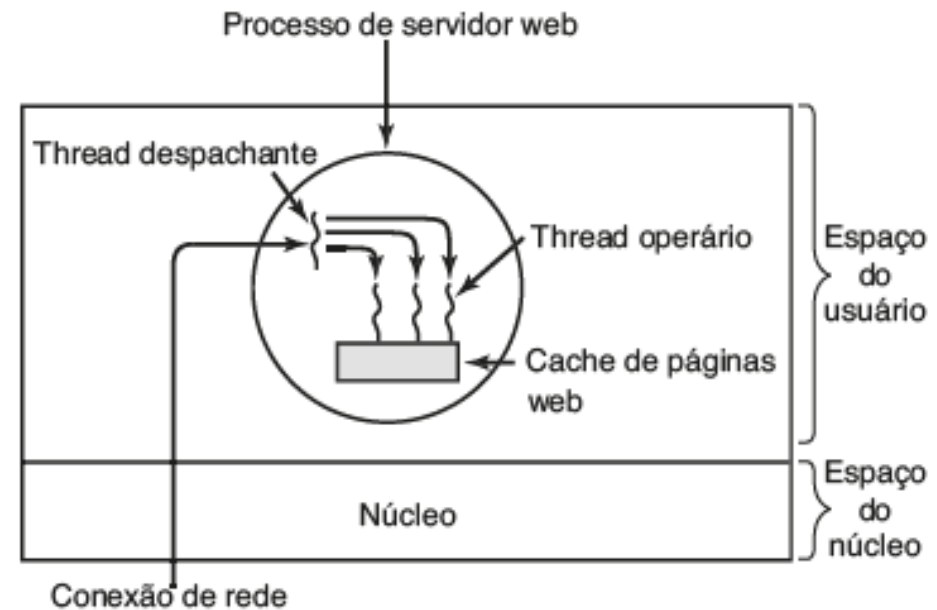


- 1ª thread: Interage com o usuário.
- 2ª thread: Reformata o documento quando solicitado.
- 3ª thread: Escreve os conteúdos da RAM para o disco periodicamente.

Exemplos análogos

- Planilha eletrônica
- Processar grandes quantidades de dados
- Servidor para um website

FIGURA 2.8 Um servidor web multithread.



Maneiras de Construir um Servidor

FIGURA 2.10 Três maneiras de construir um servidor.

| Modelo | Características |
|----------------------------|---|
| Threads | Paralelismo, chamadas de sistema bloqueantes |
| Processo monothread | Não paralelismo, chamadas de sistema bloqueantes |
| Máquina de estados finitos | Paralelismo, chamadas não bloqueantes, interrupções |

Processo

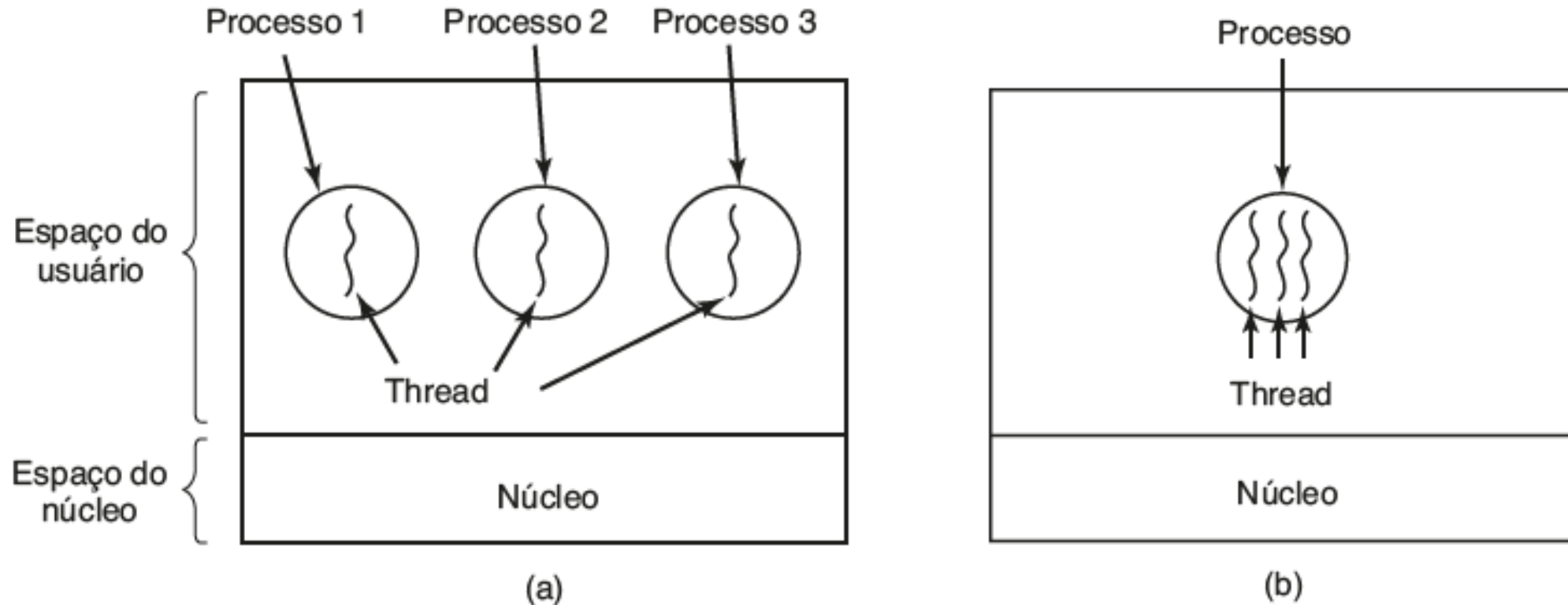
- Modo para agrupar recursos.
 - Espaço de endereçamento
 - Código, dados de programa e outros recursos.
- Thread
 - Contador de programa.
 - Registradores
 - Pilha
- Processos são usados para agrupar recursos; threads são as entidades escalonadas para execução na CPU.

Threads

- Permitem que ocorram múltiplas execuções no mesmo ambiente, com um alto grau de independência uma da outra.
 - Ter múltiplos threads executando em paralelo em um processo
- =
- Ter múltiplos processos executando em paralelo em um computador.
 - Processos leves

Processo vs. Thread

FIGURA 2.11 (a) Três processos, cada um com um thread. (b) Um processo com três threads.



Threads em um processo

- Mesmo espaço de endereçamento.
- Compartilham as mesmas variáveis globais.
- Não há proteção entre threads:
 1. É impossível
 2. Não é necessário.
- Cooperar, não lutar.

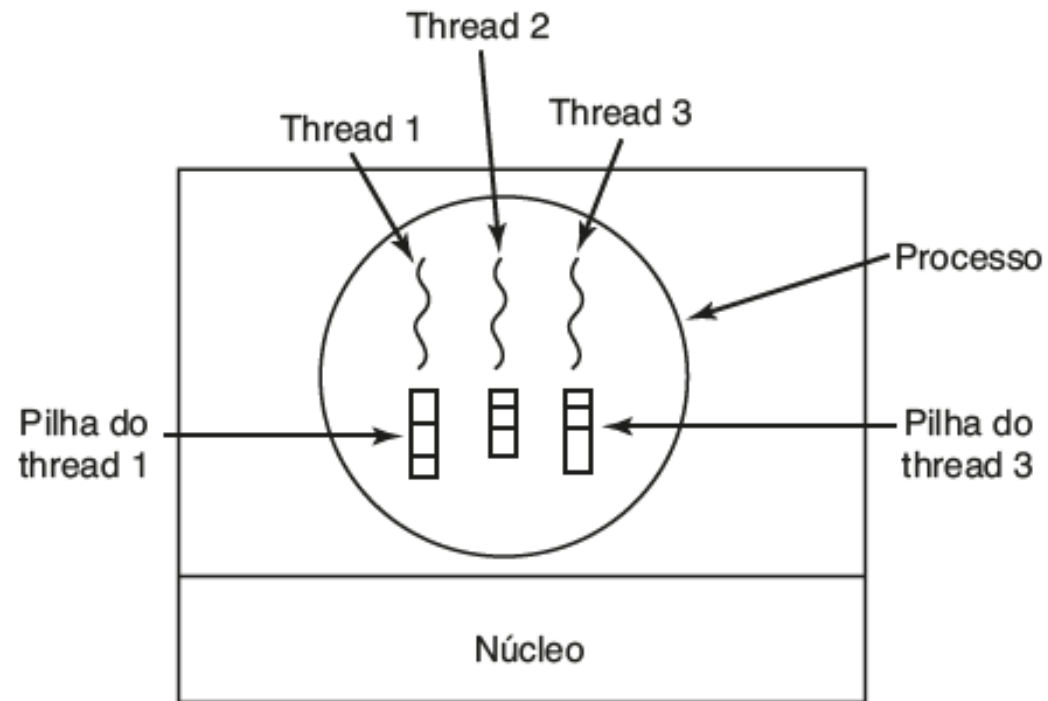
Propriedade de itens

FIGURA 2.12 A primeira coluna lista alguns itens compartilhados por todos os threads em um processo. A segunda lista alguns itens específicos a cada thread.

| Itens por processo | Itens por thread |
|-------------------------------|----------------------|
| Espaço de endereçamento | Contador de programa |
| Variáveis globais | Registradores |
| Arquivos abertos | Pilha |
| Processos filhos | Estado |
| Alarmes pendentes | |
| Sinais e tratadores de sinais | |
| Informação de contabilidade | |

Pilhas na Threads

FIGURA 2.13 Cada thread tem a sua própria pilha.



Chamadas de Threads

FIGURA 2.14 Algumas das chamadas de função do Pthreads.

| Chamada de thread | Descrição |
|----------------------|--|
| Pthread_create | Cria um novo thread |
| Pthread_exit | Conclui a chamada de thread |
| Pthread_join | Espera que um thread específico seja abandonado |
| Pthread_yield | Libera a CPU para que outro thread seja executado |
| Pthread_attr_init | Cria e inicializa uma estrutura de atributos do thread |
| Pthread_attr_destroy | Remove uma estrutura de atributos do thread |

Implementando threads

- Há dois lugares principais para implementar threads:
 - No espaço do usuário.
 - No núcleo.
 - Híbrida,

Implementando threads

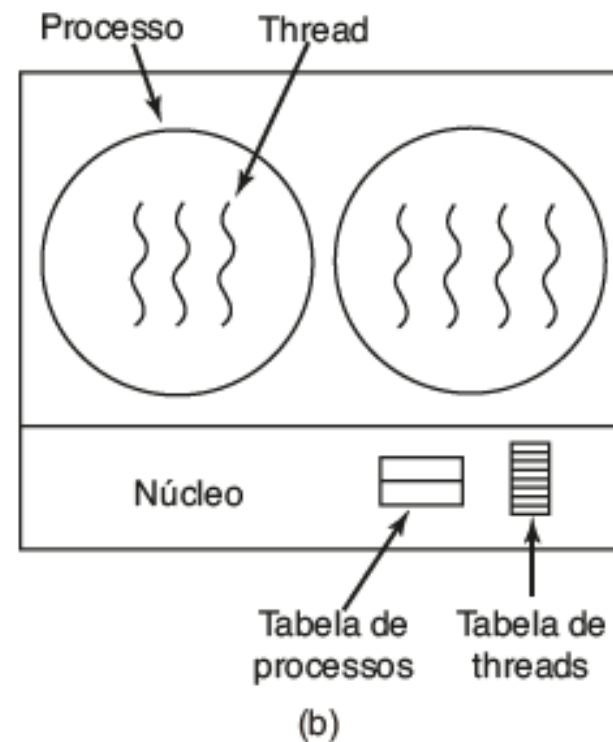
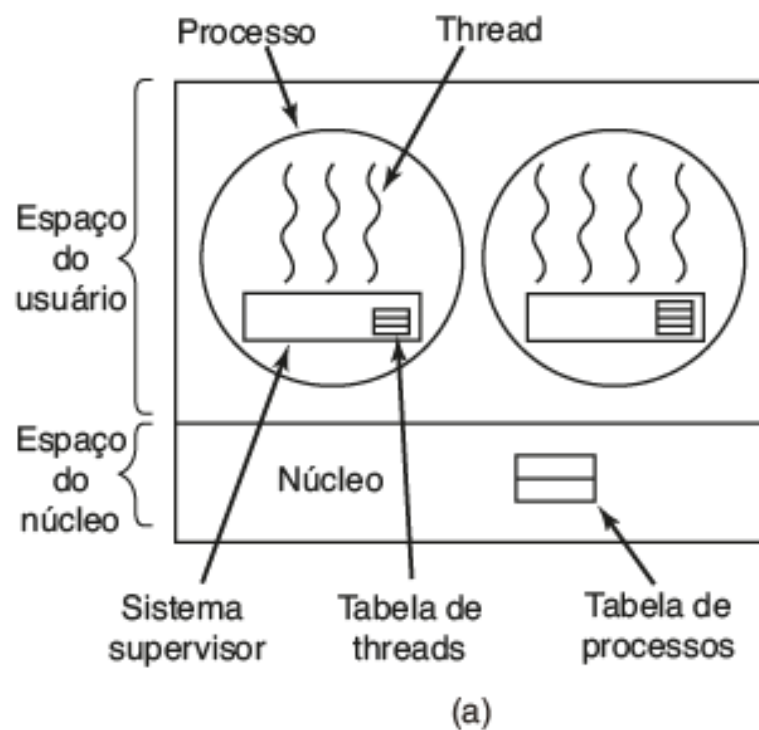
- **No espaço do usuário**
 - Coloca o pacote de threads inteiramente no espaço do usuário.
 - O núcleo não sabe nada a respeito deles.
 - As threads executam em cima de um sistema de tempo de execução, que é uma coleção de rotinas que gerencia as threads.
 - Cada processo precisa da sua própria tabela de threads privada para controlá-los naquele processo.

Implementando threads

- **No núcleo.**
 - Não é necessário um sistema de tempo de execução em cada um.
 - Não há uma tabela de thread em cada processo.
 - Todas as chamadas que poderiam bloquear um thread são implementadas como chamadas de sistema.

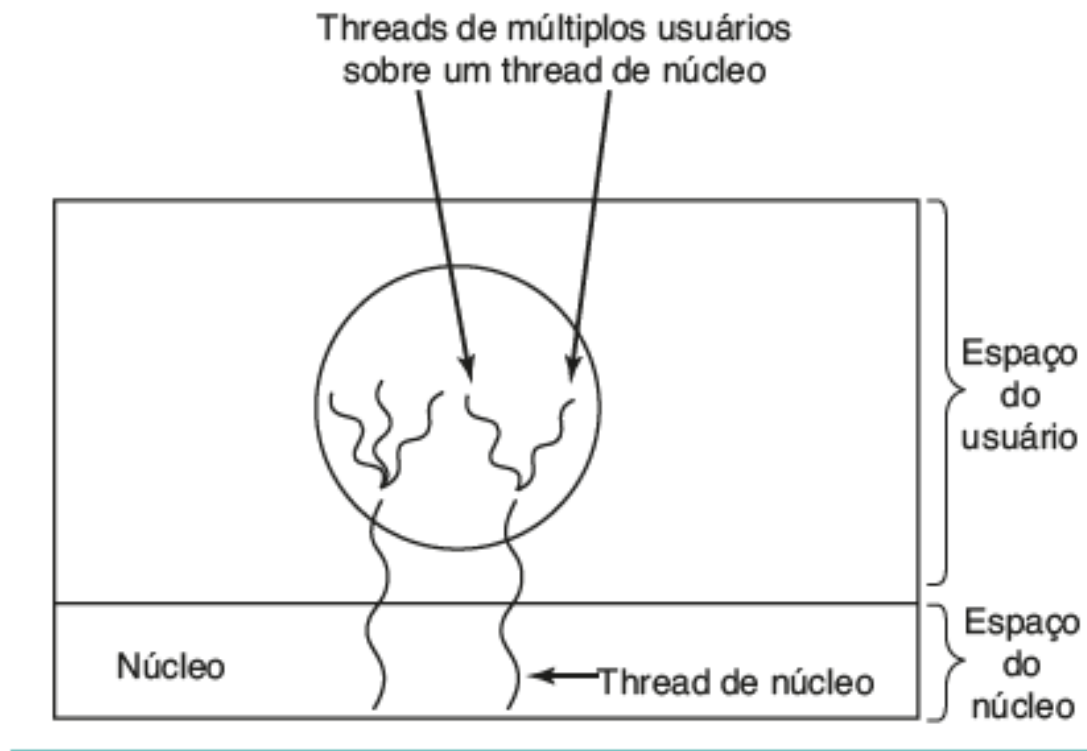
Implementação de Threads

FIGURA 2.16 (a) Um pacote de threads no espaço do usuário. (b) Um pacote de threads gerenciado pelo núcleo.



Implementação de Threads

FIGURA 2.17 Multiplexando threads de usuário em threads de núcleo.



Leitura

- SISTEMAS OPERACIONAIS MODERNO 4^a edição
 - 2.2 Threads

Dúvidas?