

Sistemas Operacionais



Threads

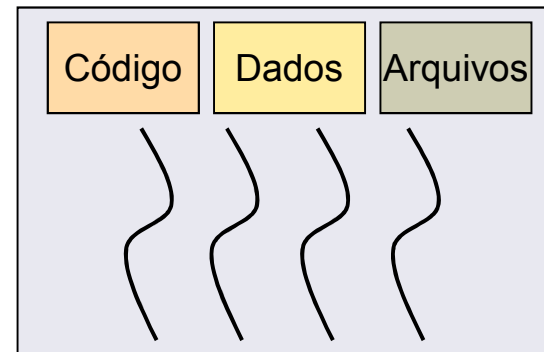
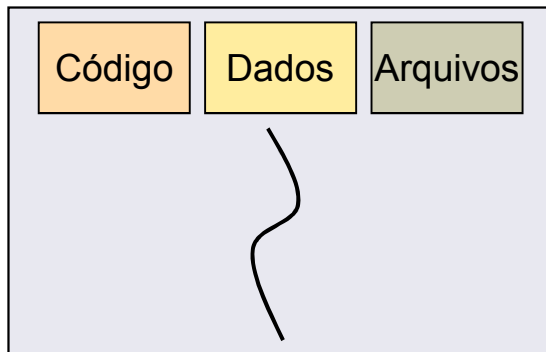
Threads

□ Processo

- Um único fluxo de execução
- Espaço de endereçamento, descritores de arquivos abertos, permissões de acesso, quotas...

□ *Threads*

- Múltiplos fluxos de execução
- *Multithreading* → mesmo espaço de endereçamento



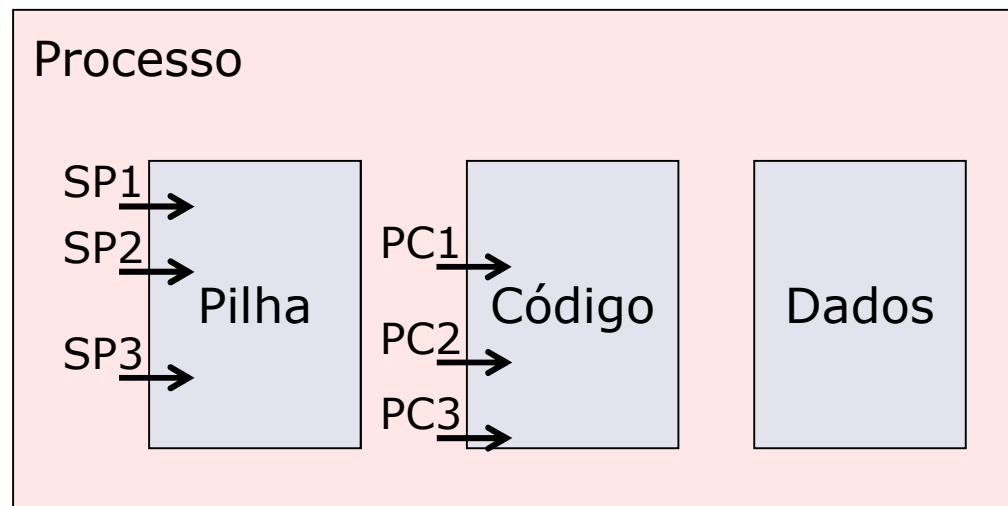
Multithreading: Associar vários fluxos de execução a um único

Threads

- Um fluxo de instrução é implementado através do Contador de Programa (PC) e do *Stack Pointer* (SP)
- Estruturas comuns compartilhadas entre os threads
 - Código
 - Dados
 - Descritor de processo

Vários Fluxos de Execução (*threads*)

- Vários fluxos de execução em um único processo



Threads

- ❑ Processo leve
- ❑ Unidade básica de utilização da CPU
- ❑ Existem no interior de um processo

Descritor de thread

- ❑ ID Thread
- ❑ Prioridade
- ❑ Contador de Programa
- ❑ Registradores
- ❑ Pilha

- ❑ Compartilha com *Threads* do mesmo processo

- Espaço de endereçamento (Seção de dados, Seção de código, arquivos abertos, recursos do SO...)
- Gerência é mais leve
- Chaveamento de contexto mais rápido

Threads

- Aplicações em PCs são *multithreading*
 - Thread único
 - Um cliente de cada vez
 - *Multithread*
 - Aceita requisições de vários clientes
 - Navegador Web
 - Servidor Web

Benefícios do Uso de Threads

- Capacidade de Resposta
 - Navegador Web
- Compartilhamento de Recursos
 - Mesmo espaço de endereço do processo
 - Memória, recursos do processo que pertence
- Economia
 - Troca de contexto de threads (alocação de memória e recursos) x processos
- Aplicações multiprocessador
 - Paralelismo

Threads de Usuário

- ❑ Implementadas através de biblioteca ligada ao programa do usuário (Threads de usuário ou do nível do usuário)
- ❑ Visíveis ao programador e desconhecidos do Kernel
- ❑ Gerenciamento (criação/escalonamento) através de bibliotecas de threads
 - Pthreads Posix
 - C-threads Mach
 - threads Solaris
- ❑ Gerenciamento mais rápido (Kernel não envolvido)

Threads de Kernel

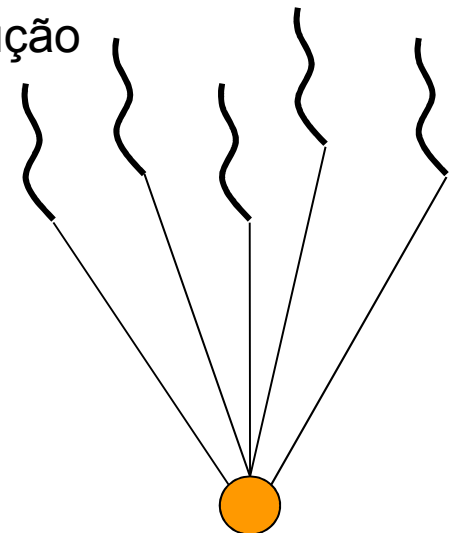
- ❑ O SO suporta diretamente o conceito de Thread
 - A gerência do SO é voltada ao conceito de thread
- ❑ Gerenciamento através do Kernel
- ❑ São escalonados para execução nos vários processadores
- ❑ Mais lentos
- ❑ Suportadas diretamente pelo SO

Os SO fornecem suporte a *threads* de usuário e de kernel

Modelos de *Multithreading*

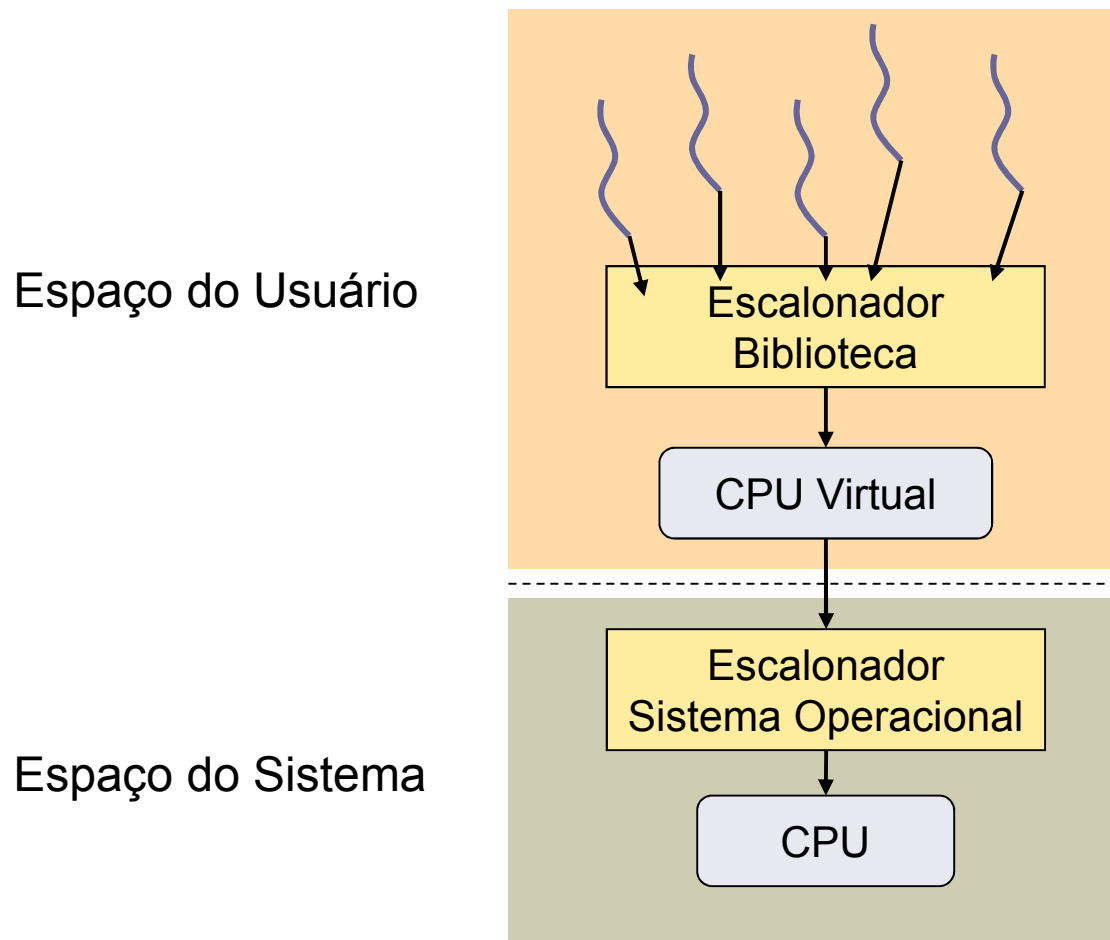
□ N:1 Muitos-para-um (*many-to-one*)

- Gerenciamento de threads é feito no espaço do usuário (aplicação)
 - São implementadas no espaço do usuário (uso de biblioteca ligada ao programa)
 - API para uso de *threads*
- Chaveamento de Contexto mais rápido
- Menor custo para criação/destruição
- O SO divide o tempo entre os processos
 - São manipuladas através de um único fluxo de instrução pertencente a um processo
- A troca de contexto entre threads é feita no modo usuário pelo escalonador embutido na biblioteca
- Se um thread está efetuando E/S, acontece o bloqueio de todas as threads do sistema
- Não é possível utilizar multiprocessadores
- Um thread é escalonado de cada vez



Modelos de *Multithreading*

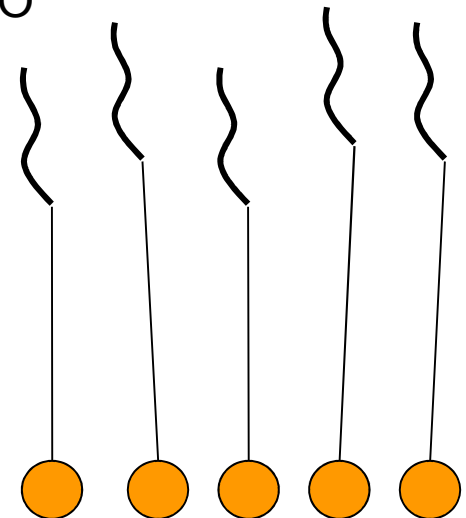
□ N:1 Muitos-para-um (*many-to-one*)



Modelos de *Multithreading*

□ 1:1 Um-para-um (*one-to-one*)

- Maior concorrência
- Threads a nível do sistema
 - O sistema operacional mantém informações sobre processos e threads
- Maior custo de gerenciamento
 - Exigem chamadas ao SO
 - Troca de contexto exige a intervenção do SO
- É possível a execução paralela (ambiente multiprocessado)
- Permite sobrepor operações de processamento com I/O
- Desempenho da aplicação
 - Depende número de threads suportados pelo sistema
- Win NT, OS/2

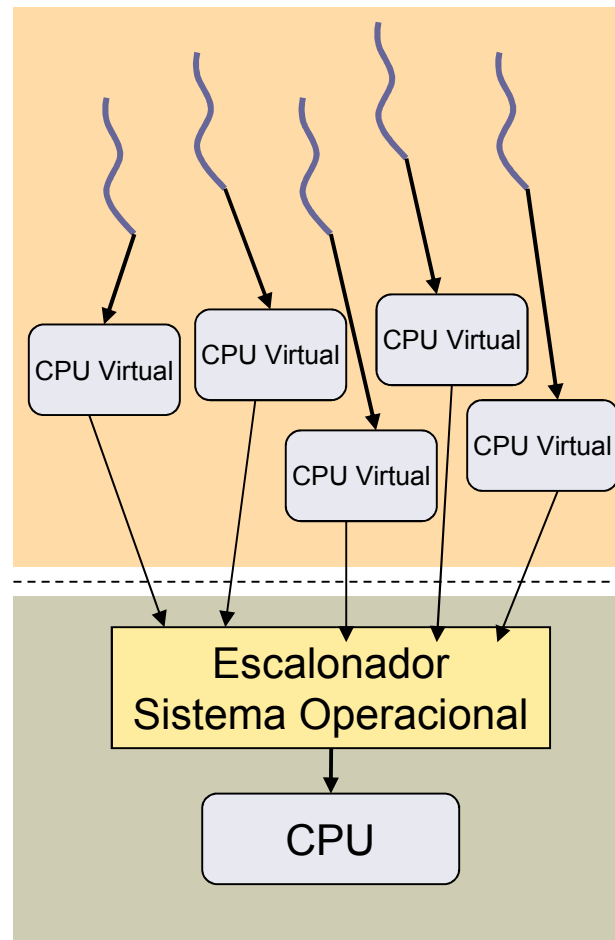


Modelos de *Multithreading*

□ 1:1 Um-para-um (*one-to-one*)

Espaço do Usuário

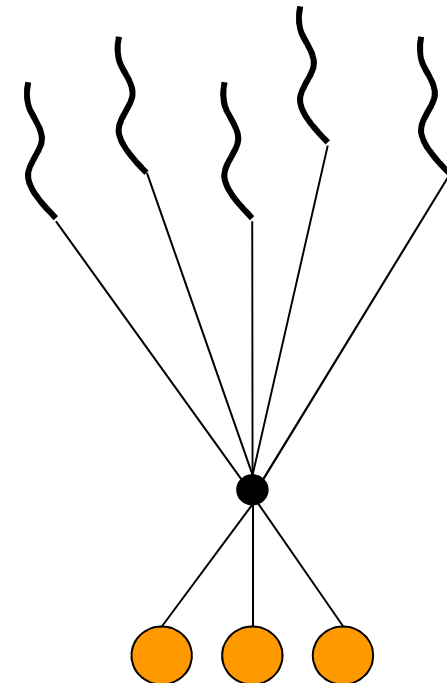
Espaço do Sistema



Modelos de Multithreading

□ M:N Muitos-para-muitos (*many-to-many*)

- Combina dos modelos N:1 e 1:1
- M *threads* do usuário
- N *threads* do SO
- Multiplexa threads de usuários em um número menor ou igual de threads de kernel
- Escalonamento de 2 níveis
 - Nível de usuário
 - Nível de sistema
- É possível criar tantos threads quantos forem necessários
 - Há um número de threads específico por aplicação ou máquina
- Bloqueio → escalonar outra para execução
- Há paralelismo

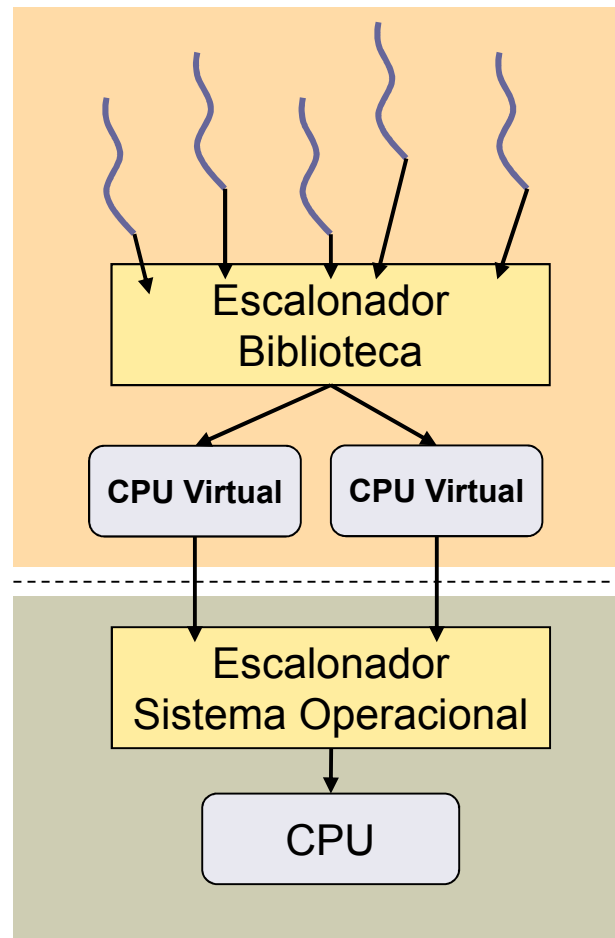


Modelos de *Multithreading*

□ M:N Muitos-para-muitos (*many-to-many*)

Espaço do Usuário

Espaço do Sistema



Vantagens de *Multithreading*

□ Threads x Processos

- Tempo de criação/destruição
- Troca de contexto
- Threads compartilham o mesmo espaço de endereçamento do processo que as criou, assim é possível o compartilhamento de memória sem a interação com o núcleo do sistema.

Referências Bibliográficas

- ❑ SILBERSCHATZ, Abraham; GALVIN, Peter; GAGNE, Greg. Sistemas Operacionais: com Java. Rio de Janeiro: Elsevier, 2004.
- ❑ OLIVEIRA, R. S.; CARISSIMI, A. S.; TOSCANI, S. T. Sistemas Operacionais. Porto Alegre: Sagra-Luzzatto, 2000.