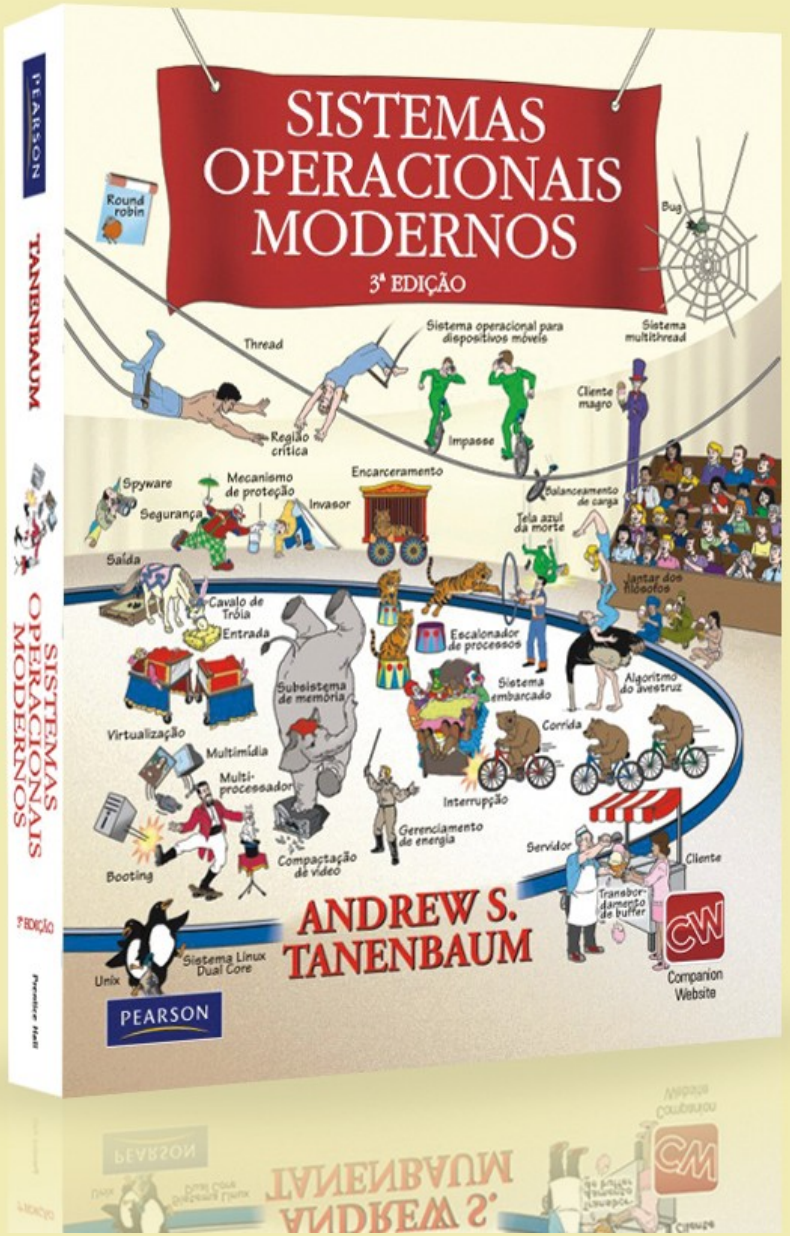
Sistemas operacionais modernos

Terceira edição
ANDREW S. TANENBAUM

Capítulo 8 Sistemas com múltiplos processadores



SISTEMAS OPERACIONAIS MODERNOS
3ª EDIÇÃO

Sistemas com múltiplos processadores

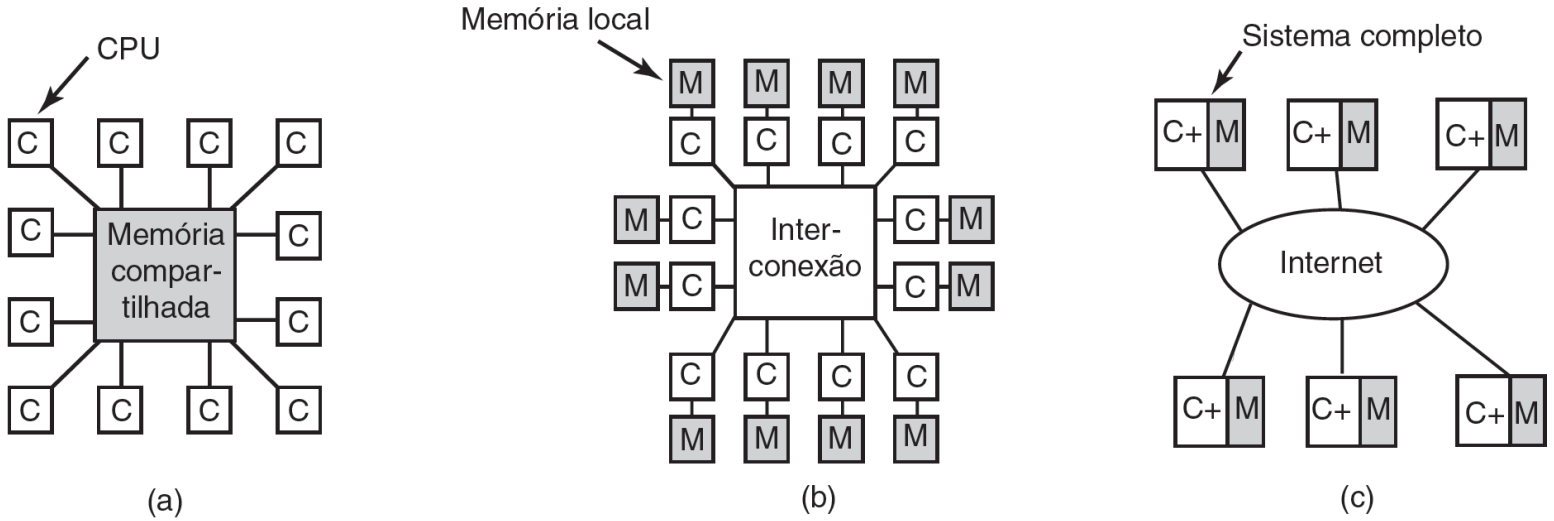


Figura 8.1 (a) Multiprocessador de memória compartilhada. (b) Multicomputador com troca de mensagens. (c) Sistema distribuído com rede de longa distância.

SISTEMAS OPERACIONAIS MODERNOS
3ª EDIÇÃO

Multiprocessadores *Uniform Memory Access* (UMA) com arquiteturas baseadas em barramentos

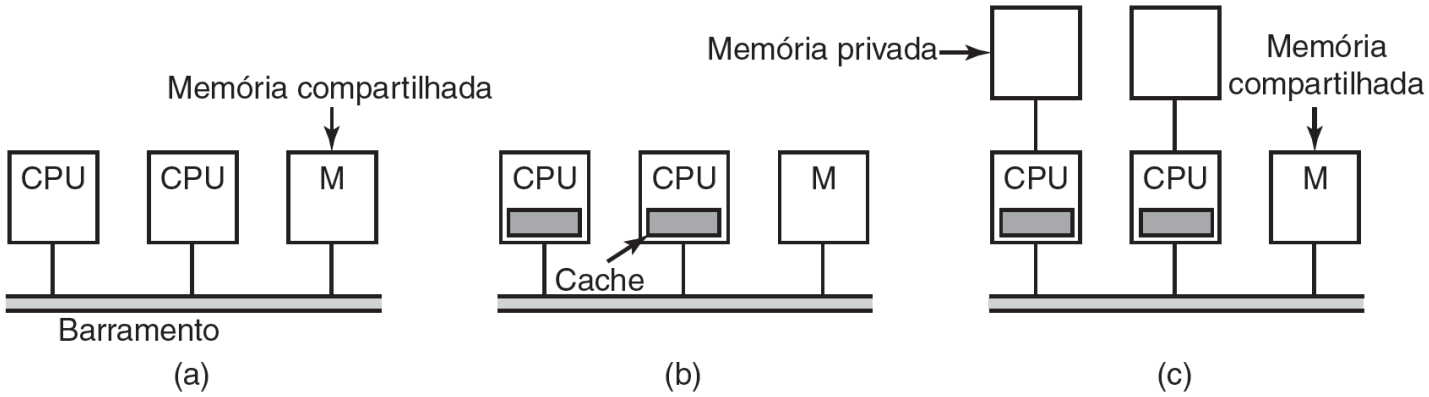
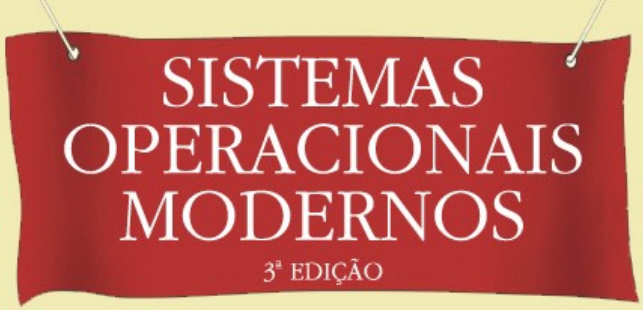


Figura 8.2 Três multiprocessadores baseados em barramentos. (a) Sem a utilização de cache. (b) Com a utilização de caches. (c) Com memórias privadas e utilização de caches.



Multiprocessadores *Nonuniform Memory Access* (NUMA): acesso a memória local é mais rápido do que o acesso remoto

Características de máquinas NUMA:

1. Existe um espaço de endereçamento único, visível a todas as CPUs.
2. O acesso à memória remota é feito via instruções **LOAD** e **STORE**.
3. O acesso à memória remota é mais lento do que o acesso à memória local.

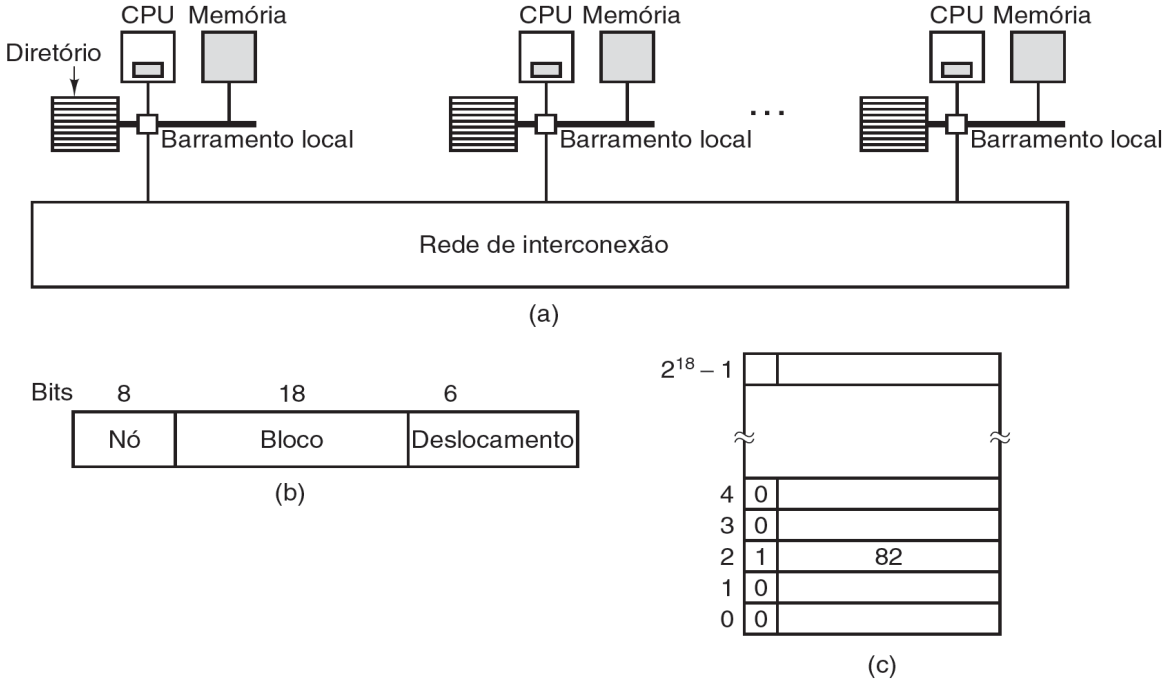


Figura 8.6 (a) Um multiprocessador de 256 nós com base em diretório. (b) Divisão de um endereço de memória de 32 bits em campos. (c) O diretório no nó 36.

- Cada nó tem 16 MB (os 256 nós totalizam 2^32 bytes) de cache (CC-NUMA, *cache-coherent NUMA*);
- Memória (*cache*) estaticamente alocada aos nós: 0-16MB ao nó 0, 16-32MB ao nó 1, etc. Cada cache tem 2^18 linhas de 64 bytes. 8 bits mais significativos identificam o nó, próximos 18 bits a linha na cache e os últimos 6 bits o deslocamento dentro da linha.
- Exemplo: nó 20 executa LOAD 0X24000108: nó 36, linha 4, deslocamento 8. Vide figura (c), essa entrada não está na memória: busca na RAM e atualiza tabela para indicar que a entrada está agora no nó 20;
- Agora o nó 20 acessa a entrada 2 do nó 36: entrada indica que ela está no nó 82; atualiza tabela indicando que agora está no nó 20, fazendo com que o nó 36 instrua o nó 82 a repassar essa entrada para o nó 20 e, logo após, invalide a entrada na sua tabela (do nó 82).

Cada CPU tem seu próprio sistema operacional: **otimização** possível seria cada CPU compartilhar cópia do código do SO

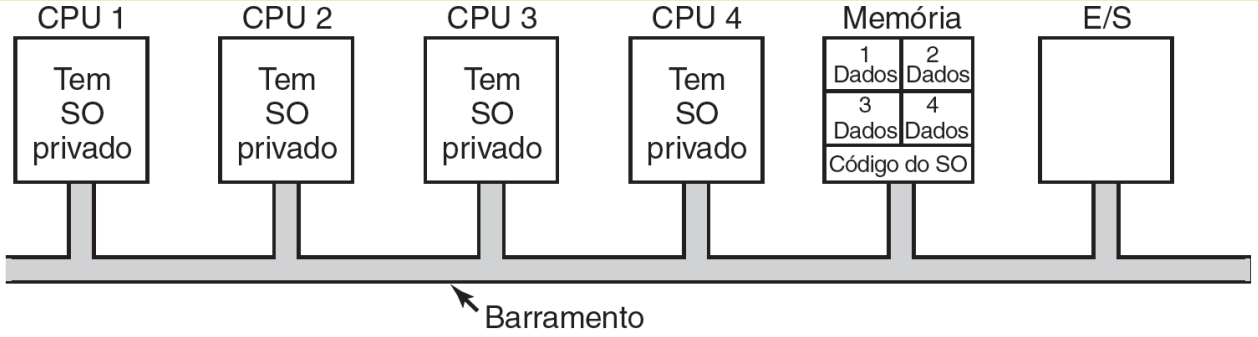


Figura 8.7 Compartilhamento da memória entre as quatro CPUs, mas compartilhando somente uma cópia do código do sistema operacional. As caixas identificadas como 'Dados' contêm os dados particulares do sistema operacional para cada CPU.

Multiprocessadores “mestre-escravo”

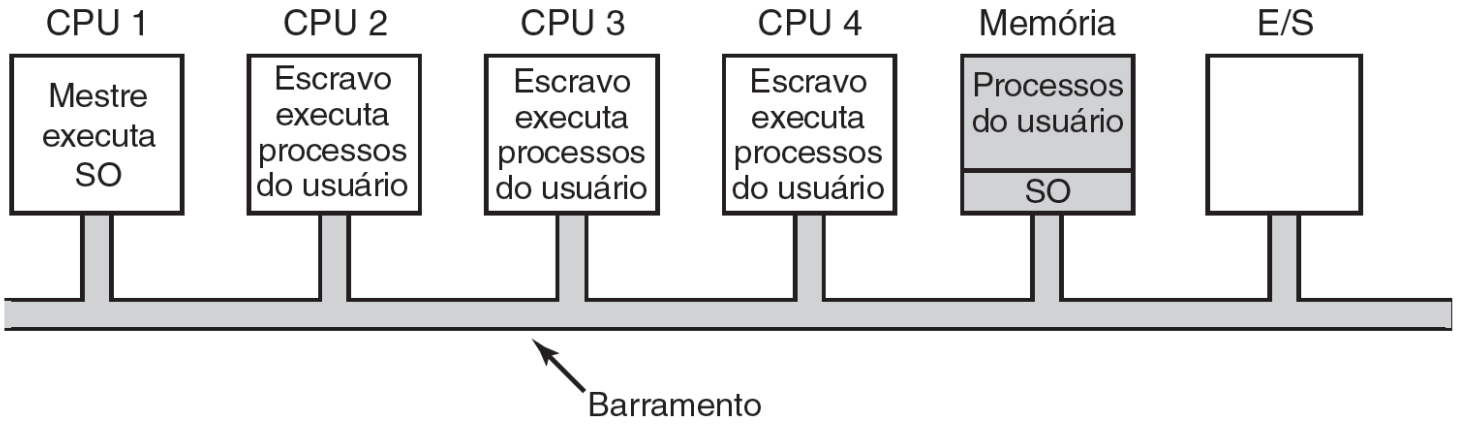


Figura 8.8 Um modelo de multiprocessadores mestre-escravo.

SISTEMAS OPERACIONAIS MODERNOS
3ª EDIÇÃO

Multiprocessadores simétricos (SMP): SO é dividido em regiões críticas, as quais podem ser acessadas uma por vez por cada CPU (mutex para implementar os *locks*).

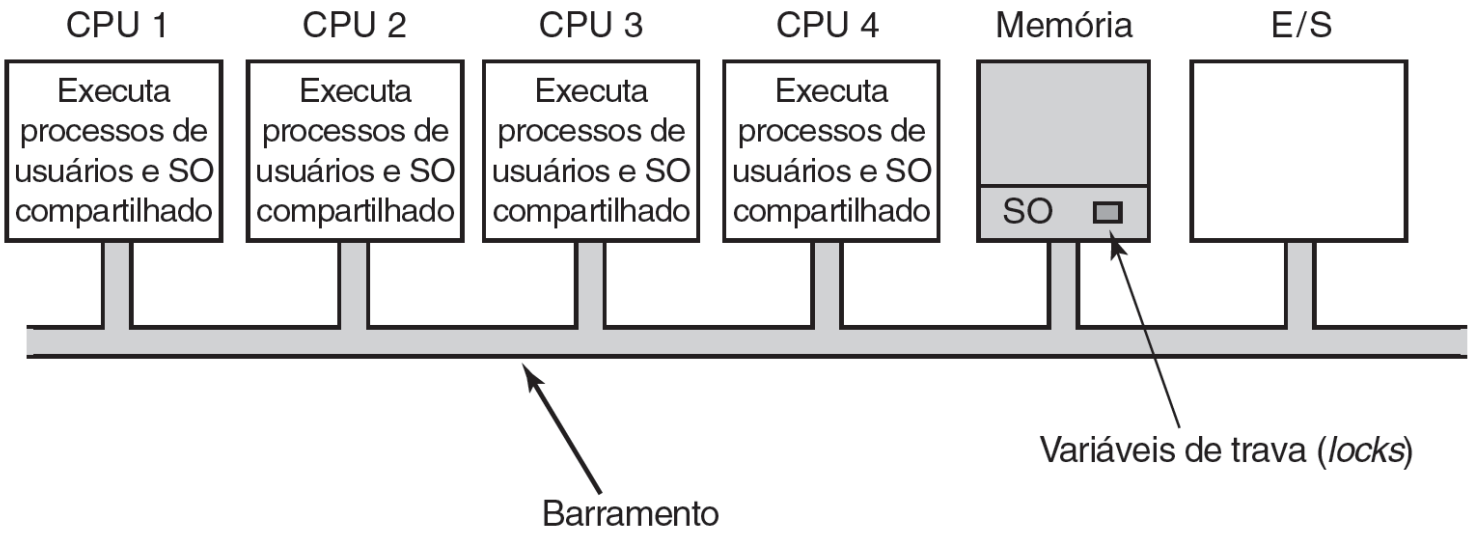


Figura 8.9 O modelo de multiprocessadores simétricos (SMP).

Sincronização em multiprocessadores: como as duas CPUs obtém um 0 da instrução TSL, as duas acessam a região crítica, não garantindo exclusão mútua! Com travamento do barramento, o TSL funciona corretamente mas tem um custo elevado (**ociosidade das outras CPUs**).

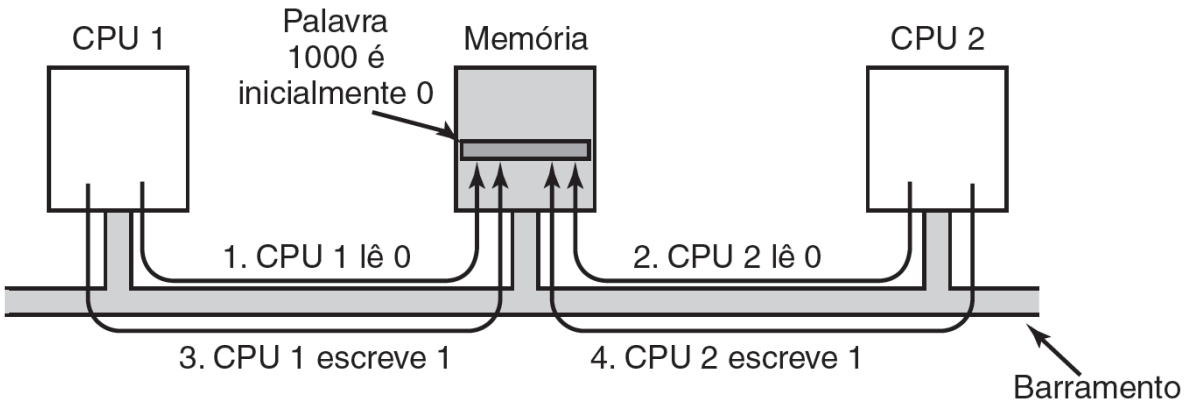


Figura 8.10 A instrução TSL pode falhar se o barramento não puder ser travado. As quatro etapas da figura mostram uma sequência de eventos na qual a falha é demonstrada.

Compartilhamento de tempo: (a) 16 CPUs e filas compartilhadas (por prioridade) de *threads*; (b) CPU 4 fica ociosa, obtém acesso exclusivo às filas de escalonamento e recebe a *thread* de mais alta prioridade (A); (c) CPU 12 fica ociosa e recebe a thread B.

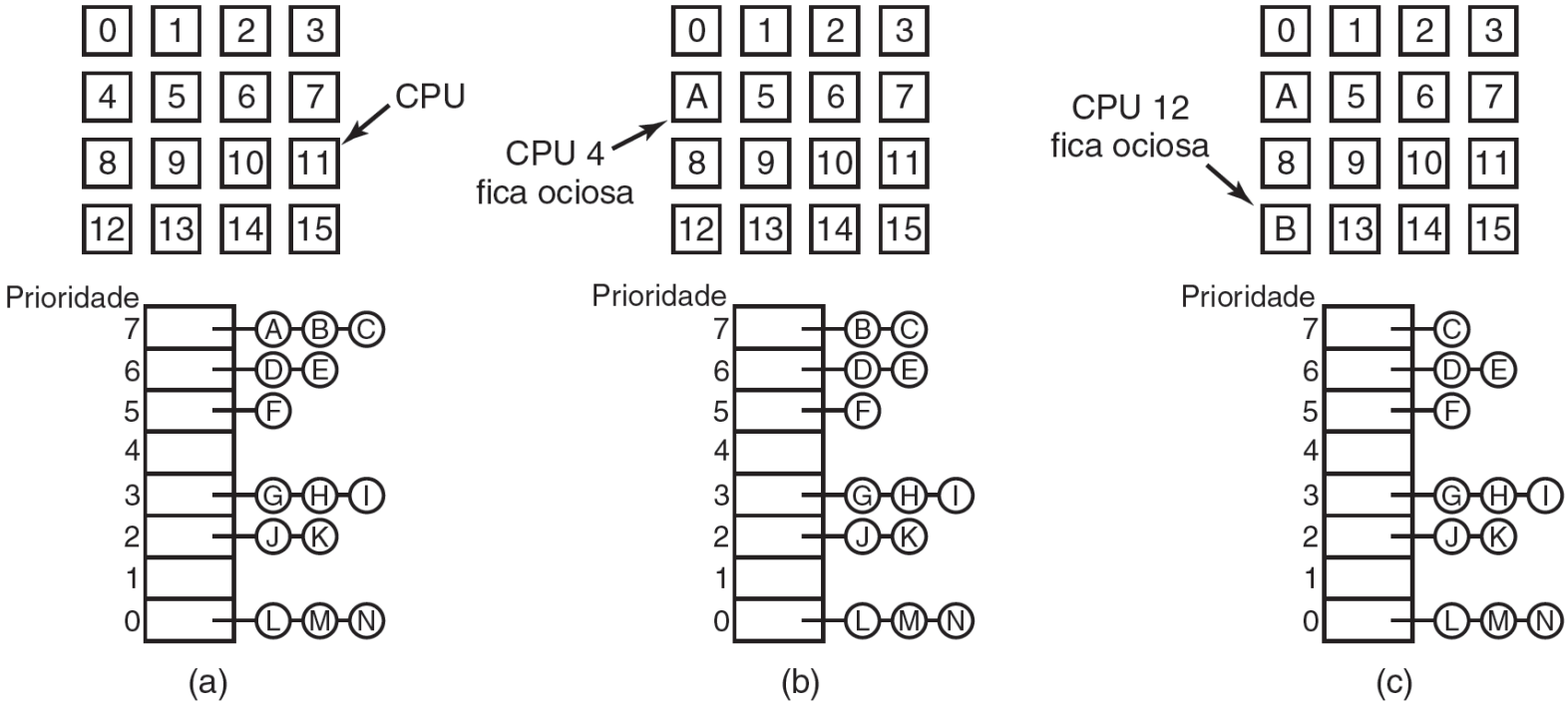


Figura 8.12 Uso de uma única estrutura de dados no escalonamento de um multiprocessador.

SISTEMAS OPERACIONAIS MODERNOS
3ª EDIÇÃO

Compartilhamento de espaço é o escalonamento de múltiplas *threads* de um mesmo processo ao mesmo tempo. Conjunto de *threads* são escalonados caso exista o número correspondente de CPUs disponíveis. Com o tempo, cria-se partições (cujo tamanho varia de acordo com a criação/término de processos). **Põe fim a necessidadede de chaveamento de contexto, mas pode gerar desperdício de CPU quando uma ou mais *threads* do conjunto bloqueiam.**

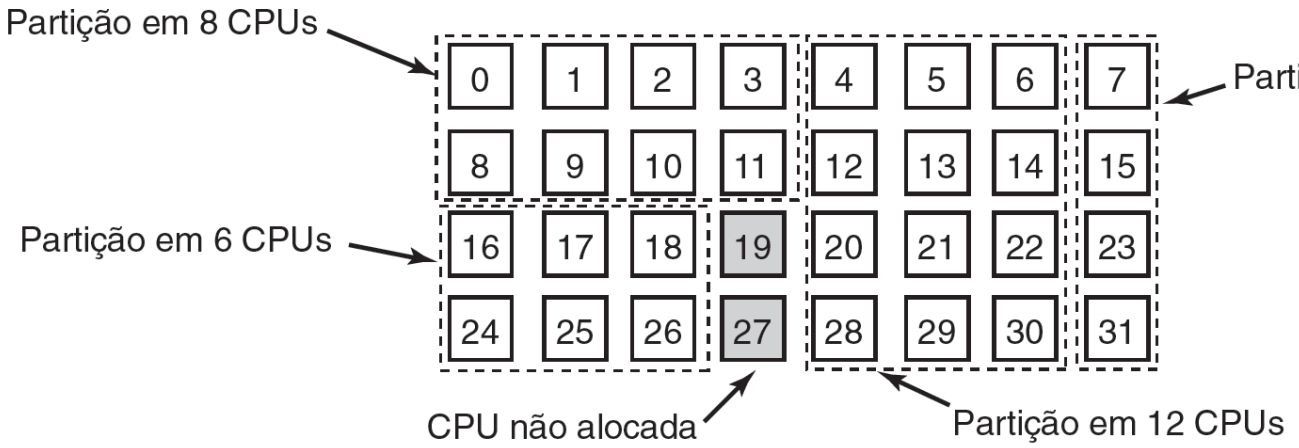


Figura 8.13 Conjunto de 32 CPUs agrupadas em quatro partições, com duas CPUs disponíveis.

Escalonamento em bando (*gang scheduling*)*:* no exemplo abaixo, A0 e A1 comunicam-se entre si mas estão fora de fase (i.e., não executam simultaneamente), gerando um atraso de 200 *ms* entre mensagens. **Para evitar o desperdício da solução anterior, esse escalonamento atua no espaço e no tempo!**

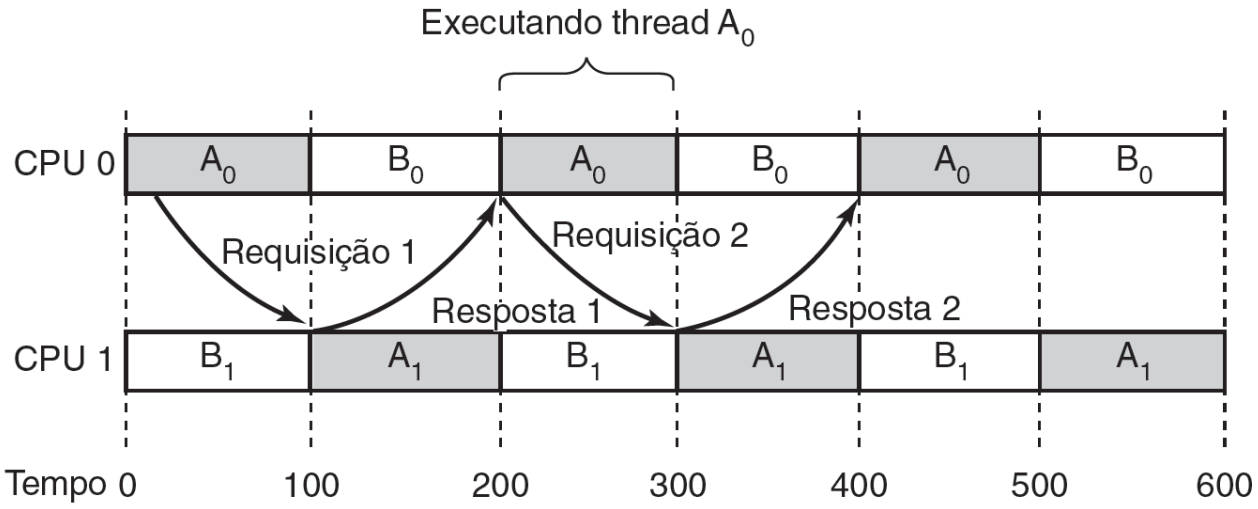
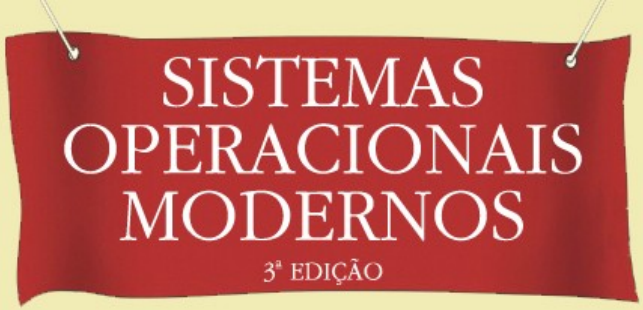


Figura 8.14 Comunicação entre dois threads pertencentes ao thread *A* que estão sendo executados fora de fase.



As três partes do escalonamento em bando:

1. Os grupos de *threads* relacionados são escalonados como uma unidade chamada bando.
2. Todos os membros de um bando executam simultaneamente, em diferentes CPUs com tempo compartilhado.
3. Todos os membros do bando iniciam e finalizam juntos suas fatias de tempo.

Intervalo de tempo	CPU 0	1	2	3	4	5
0	A_0	A_1	A_2	A_3	A_4	A_5
1	B_0	B_1	B_2	C_0	C_1	C_2
2	D_0	D_1	D_2	D_3	D_4	E_0
3	E_1	E_2	E_3	E_4	E_5	E_6
4	A_0	A_1	A_2	A_3	A_4	A_5
5	B_0	B_1	B_2	C_0	C_1	C_2
6	D_0	D_1	D_2	D_3	D_4	E_0
7	E_1	E_2	E_3	E_4	E_5	E_6

Figura 8.15 Escalonamento em bando.

SISTEMAS OPERACIONAIS MODERNOS
3ª EDIÇÃO

Multicomputadores (*clusters*):Tecnologia de interconexão

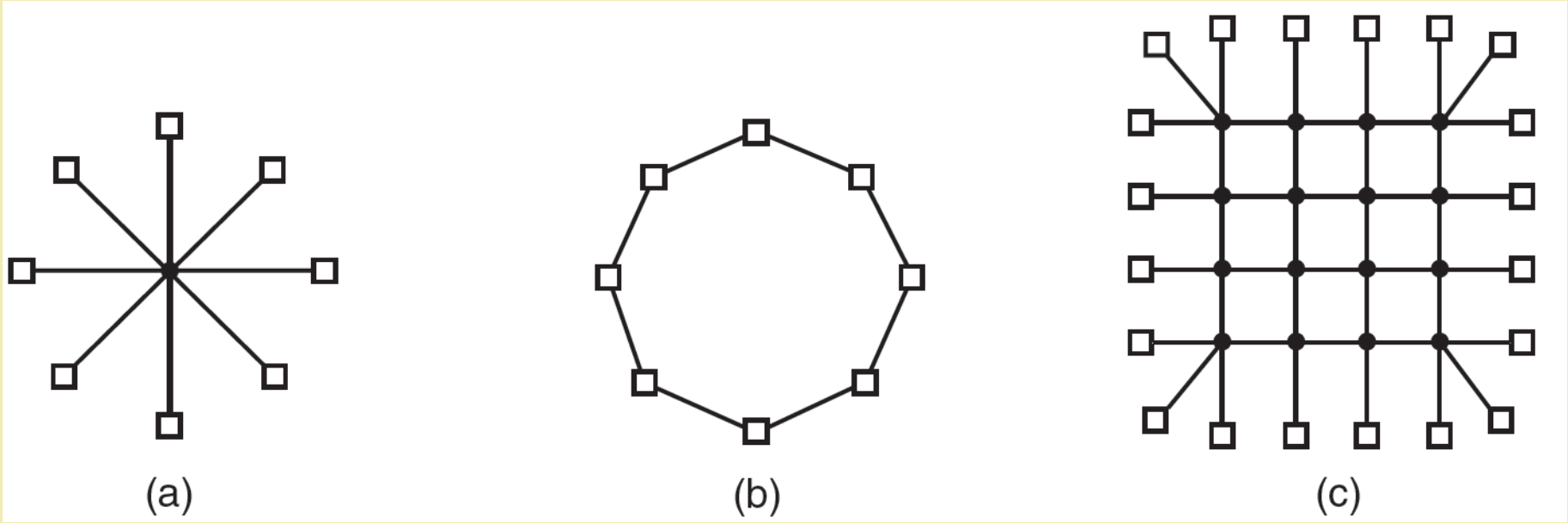


Figura 8.16 Várias topologias de interconexão. (a) Uma chave simples. (b) Um anel. (c) Uma grade. (d) Um toro duplo. (e) Um cubo. (f) Um hipercubo 4D.

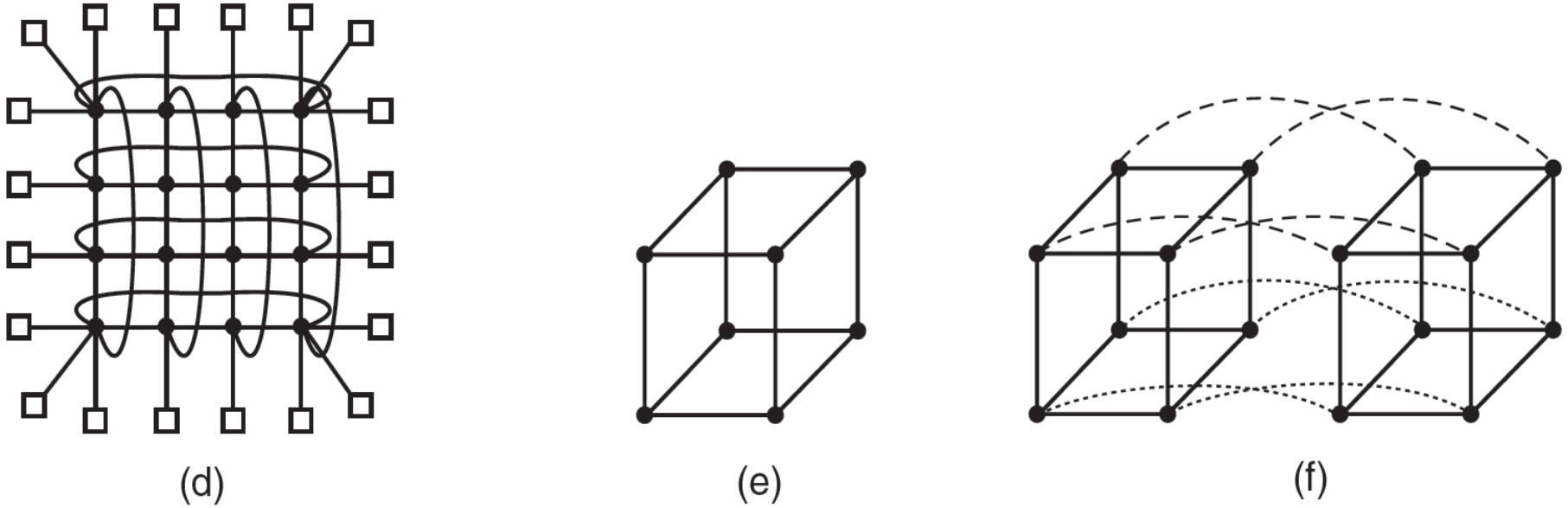


Figura 8.16 Várias topologias de interconexão. (a) Uma chave simples. (b) Um anel. (c) Uma grade. (d) Um toro duplo. (e) Um cubo. (f) Um hipercubo 4D.

Comutação de pacotes no modo *store-and-forward* (obs.: também há *switches* que implementam a tecnologia *cut-through*)

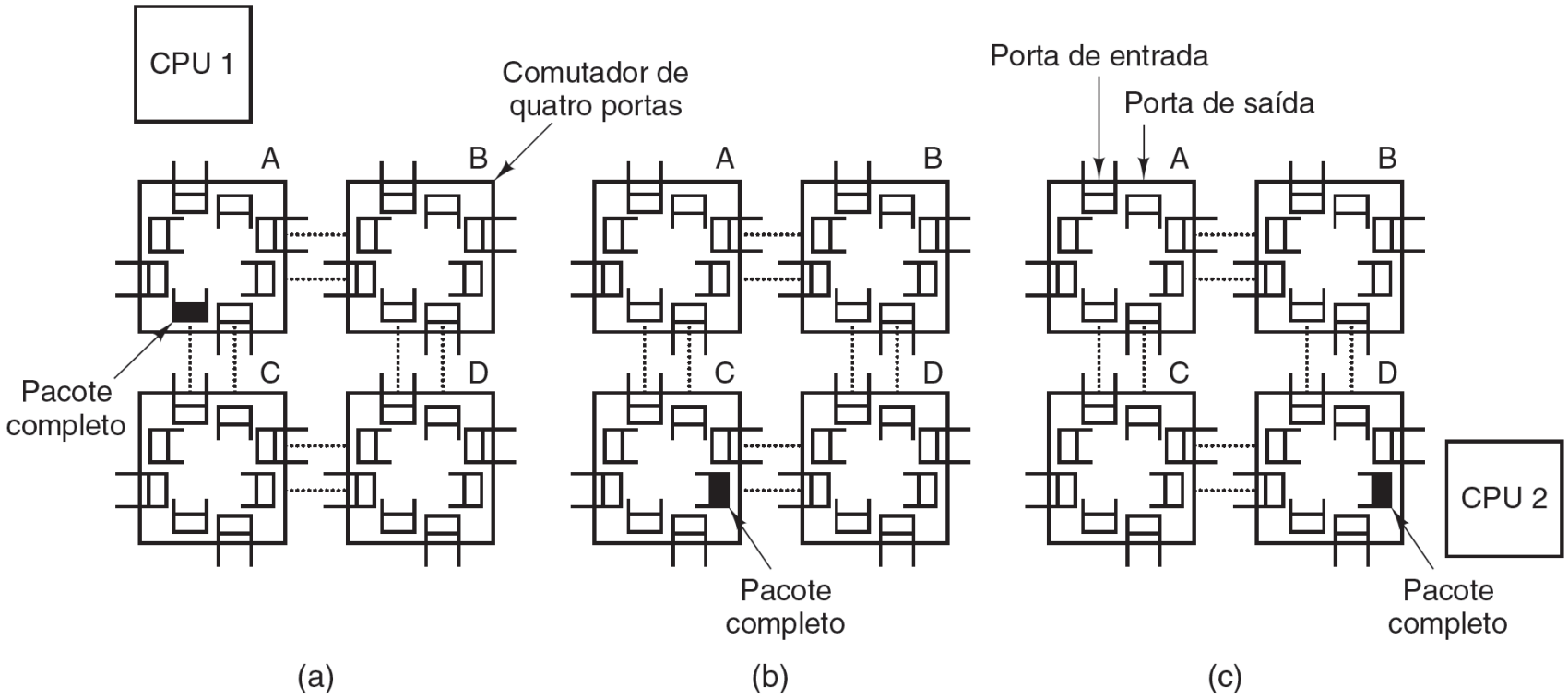


Figura 8.17 Comutação de pacotes armazenar e encaminhar.

SISTEMAS OPERACIONAIS MODERNOS
3ª EDIÇÃO

Interfaces de rede

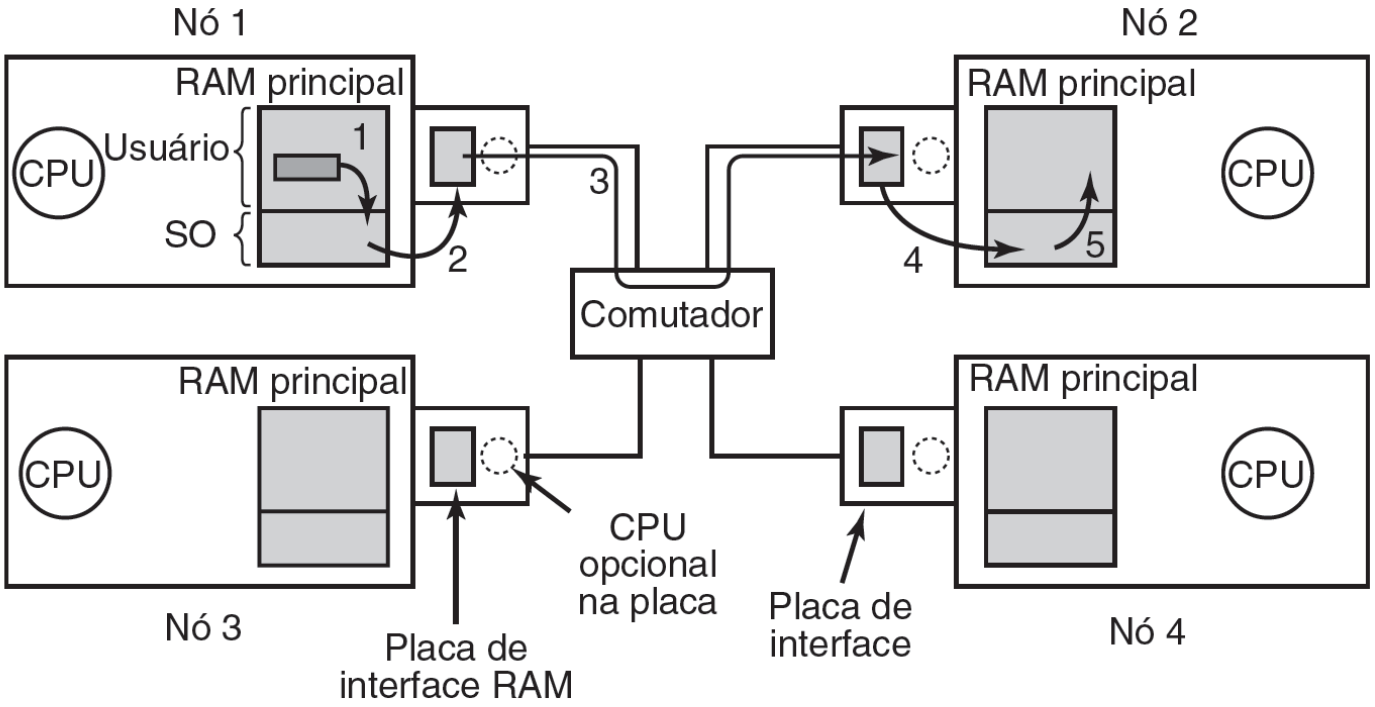


Figura 8.18 Posição das placas de interface de rede em um multicomputador.

SISTEMAS OPERACIONAIS MODERNOS
3ª EDIÇÃO

Chamadas bloqueantes *versus* não bloqueantes

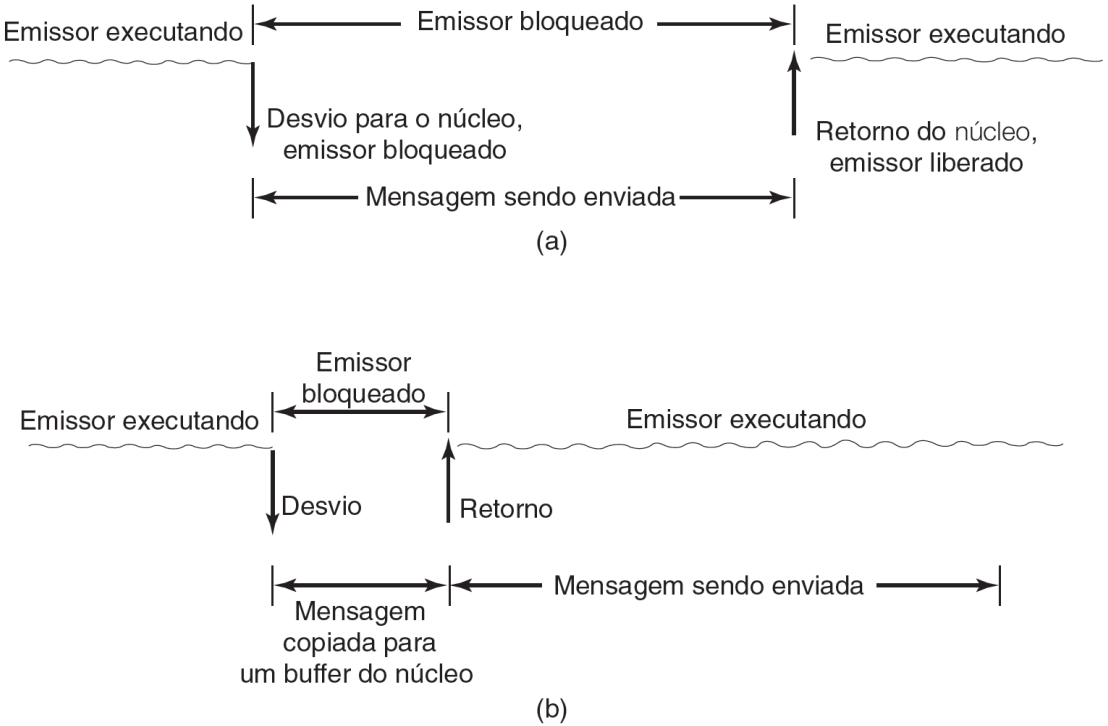
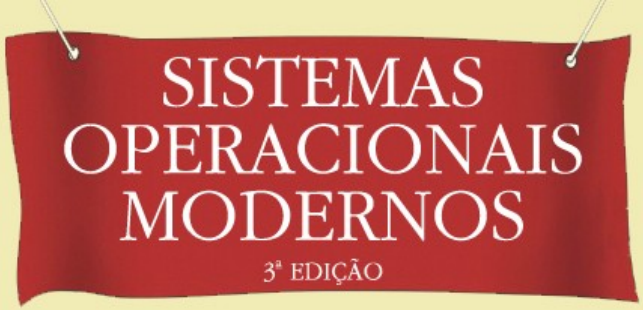


Figura 8.19 (a) Uma chamada *send* bloqueante. (b) Uma chamada *send* não bloqueante.



Escolhas do lado do emissor:

1. Envio bloqueante (a CPU fica ociosa durante a transmissão da mensagem, caso não haja mudança de contexto).
2. Envio não bloqueante com cópia (tempo da CPU desperdiçado para cópia extra). **É importante lembrar que o emissor não poderia modificar o *buffer* antes do envio da mensagem (já que ele não está bloqueado, isso poderia acontecer mesmo acidentalmente!).**
3. Envio não bloqueante com interrupção (torna a programação difícil): não precisa cópia, pois avisa (interrompe) o emissor após envio;
4. Cópia na escrita (uma cópia extra eventualmente é necessária): **somente caso o *buffer* seja modificado antes de terminar envio**.

Chamada de procedimento remoto (*remote procedure call, RPC*): programas (cliente e servidor) devem ser ligados a procedimentos (*stubs*) da biblioteca.

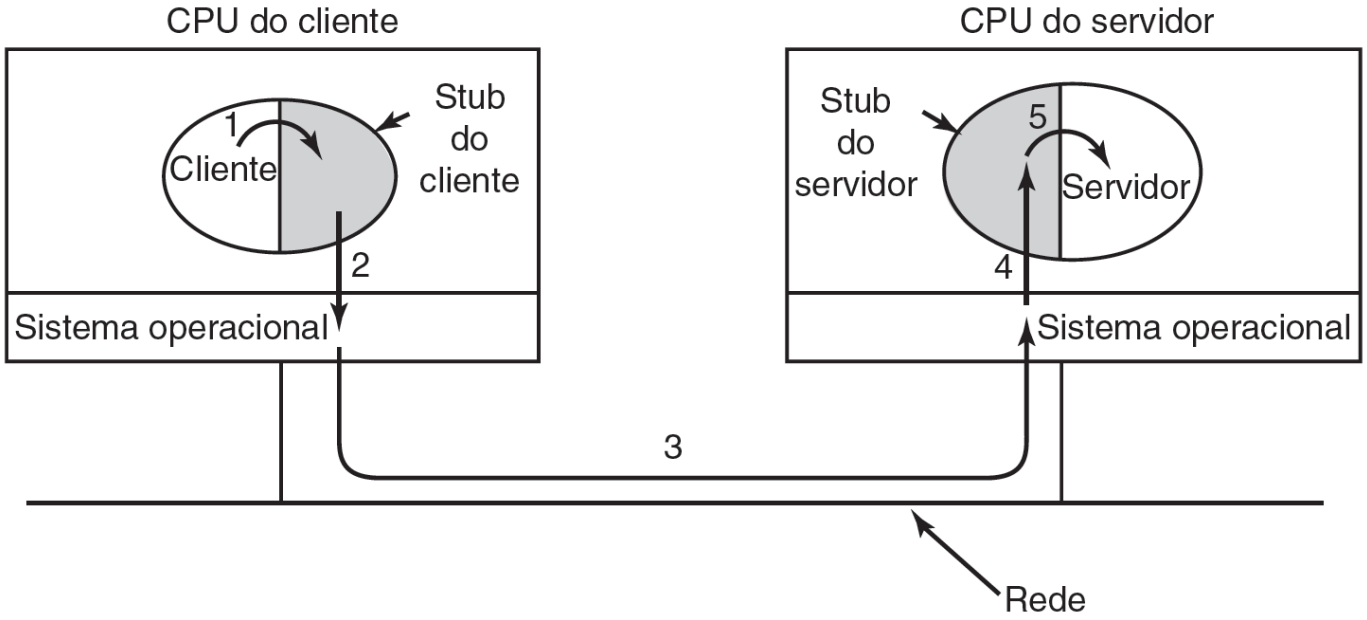


Figura 8.20 Passos na realização de uma chamada de procedimento remoto. Os stubs estão pintados de cinza.

Memória compartilhada distribuída (*distributed shared memory*): (a) compartilhamento real (sistema multiprocessado); (b) DSM (a nível de SO); (c) em algum outro nível de *software*.

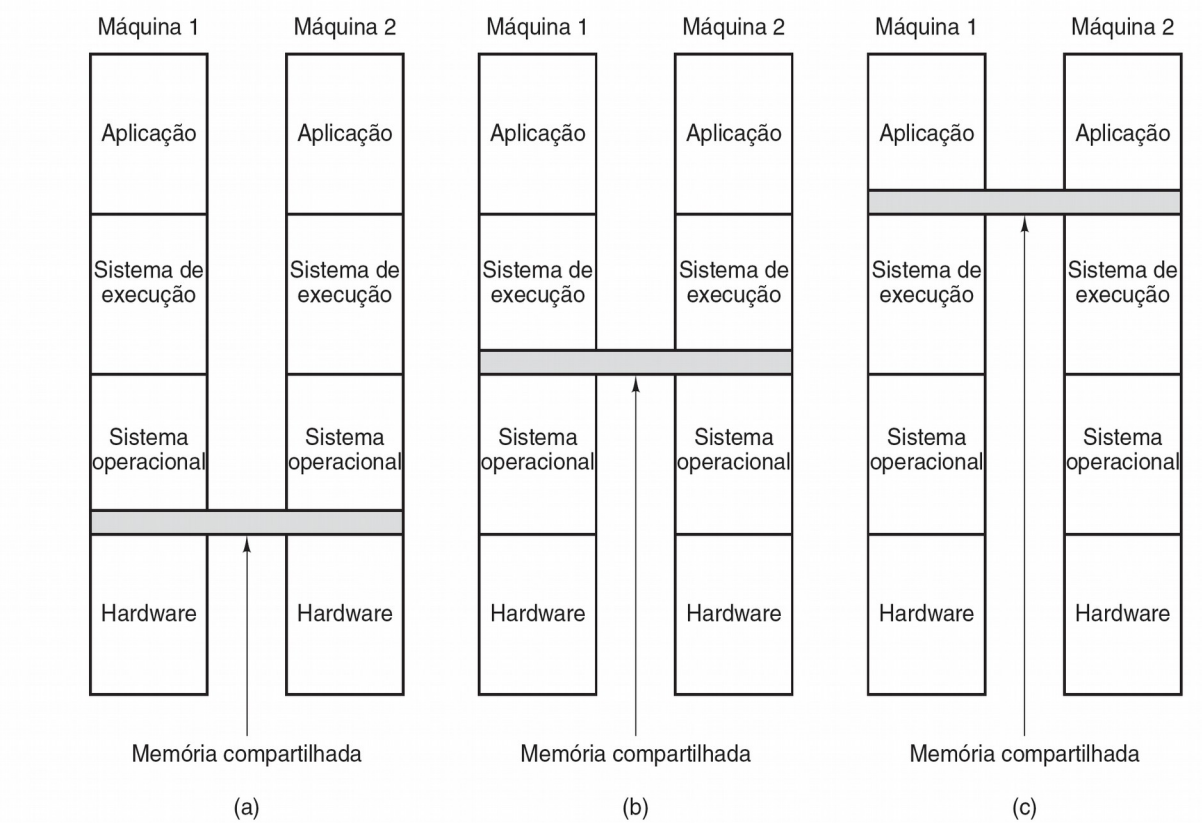


Figura 8.21 Diversas camadas nas quais a memória compartilhada pode ser implementada. (a) No hardware. (b) No sistema operacional. (c) No nível do usuário.

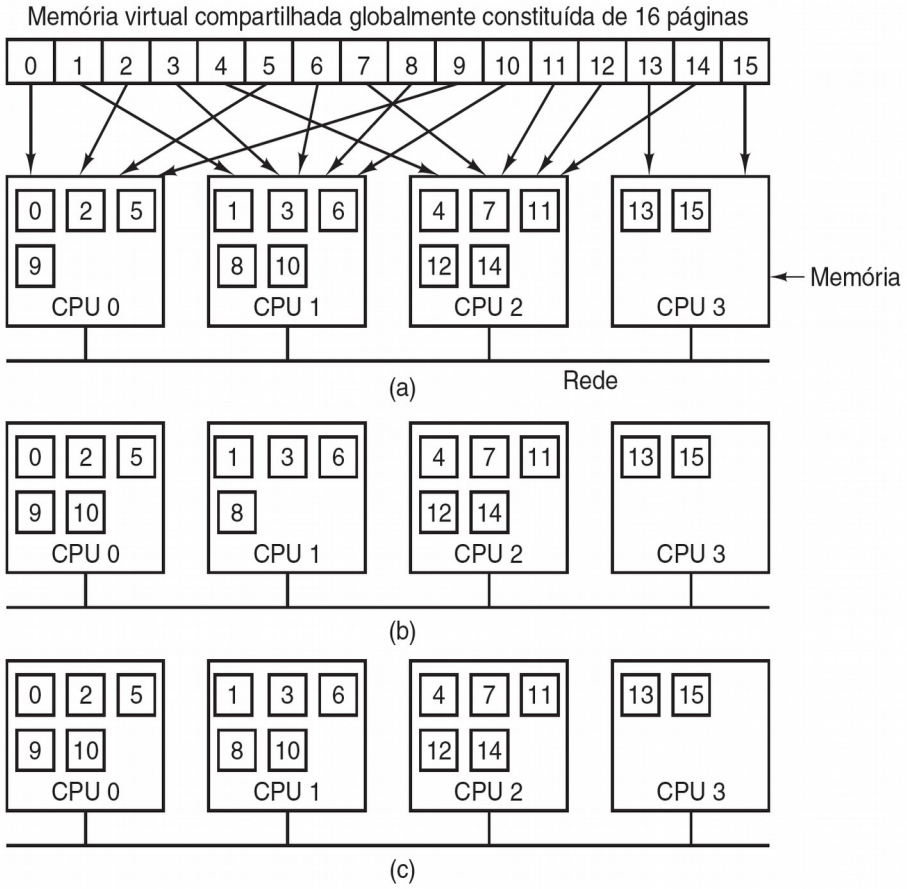


Figura 8.22 (a) Páginas do espaço de endereçamento distribuídas entre quatro máquinas. (b) Situação após a CPU 0 referenciar a página 10 e esta página ser movida para lá. (c) Situação se a página 10 é do tipo somente leitura e a replicação é usada.

Falso compartilhamento: na CPU1 utiliza-se a varíavel A com frequência e na CPU2 utiliza-se a variável B com frequência. Como ambas estão na mesma página, tem-se a falsa impressão que os processos distintos estão compartilhando a mesma página. Nesse caso, a página (e não somente as duas variáveis A e B) fica transitando entre as duas máquinas.

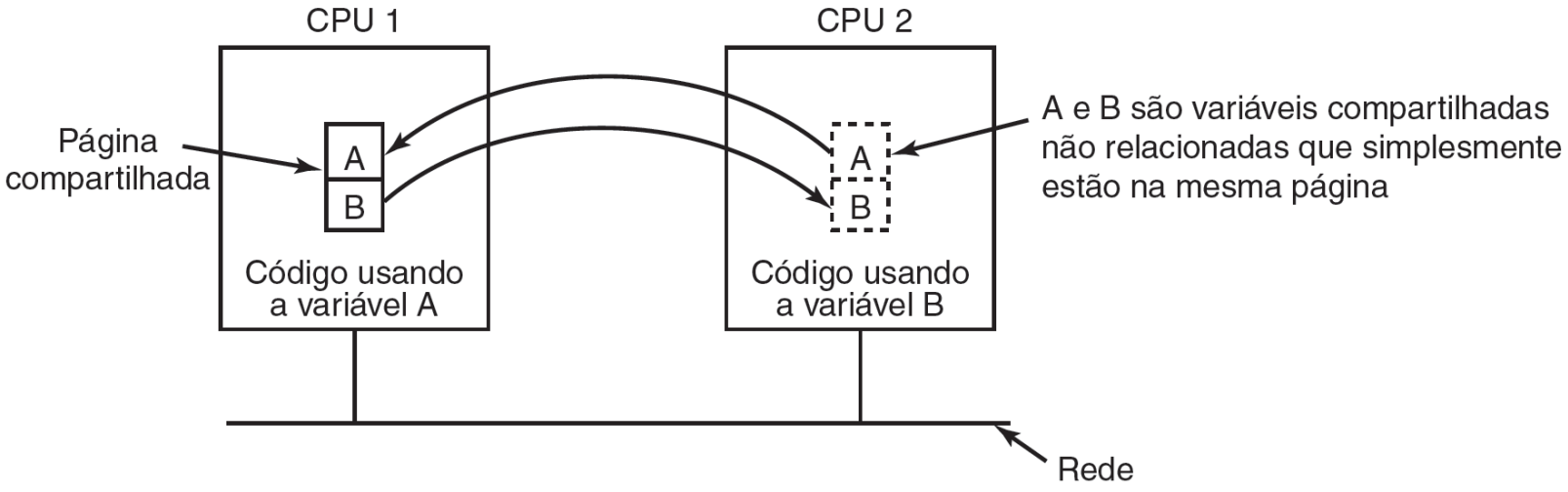


Figura 8.23 Falso compartilhamento de uma página contendo duas variáveis não relacionadas.

SISTEMAS OPERACIONAIS MODERNOS
3ª EDIÇÃO

Escalonamento em multicomputador: algoritmo determinístico teórico de grafos (vértices=processo; aresta=fluxo de mensagem entre os processos). Arestas dentro de um sub-grafo representam comunicação intra-máquina e os demais são via rede. Solução deve alocar os processos às máquinas e minimizar tráfego em rede satisfazendo todas as restriçoes (limite de recursos: mémoria, CPU, etc). (a) fluxo = 30 (e.g., 30 Mbps); (b) fluxo = 28 Mbps.

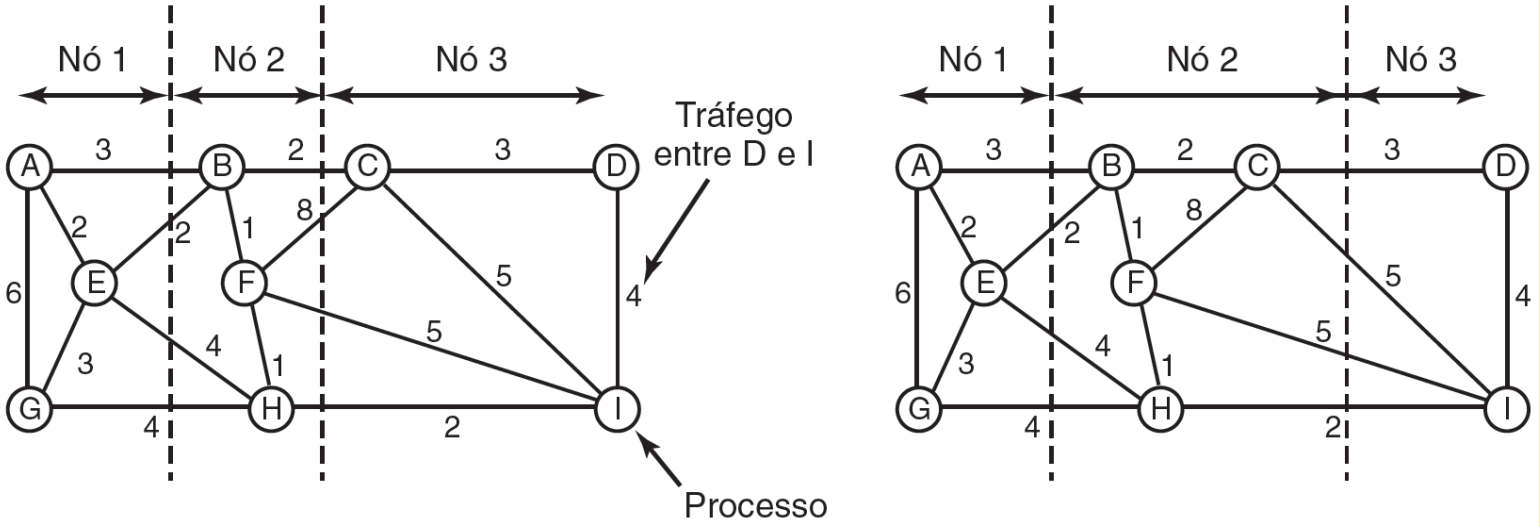


Figura 8.24 Duas maneiras de alocar nove processos em três nós.

Escalonamento: algoritmo heurístico distribuído iniciado pelo emissor (a) e pelo receptor (b).

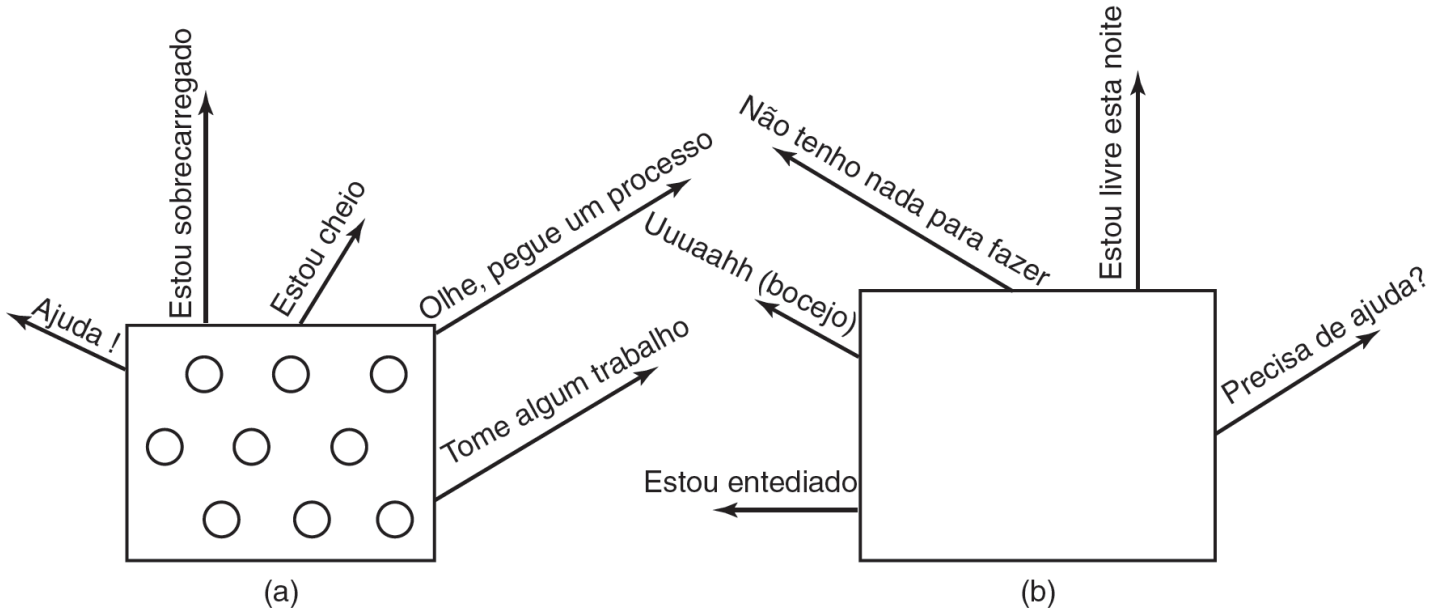


Figura 8.25 (a) Um nó superatarefado procurando por um nó menos carregado para o qual possa repassar processos. (b) Um nó vazio procurando trabalho para fazer.

SISTEMAS OPERACIONAIS MODERNOS
3ª EDIÇÃO

Virtualização: **Hipervisor tipo 1** (ou **monitor de máquina virtual**), é o único programa funcionando no modo *kernel*, gerenciando múltiplas cópias virtuais do *hardware* real (máquinas virtuais). Em CPUs com suporte a virtualização (*e.g.*, Intel e AMD atuais), uma *trap* (armadilha) é acionada toda a vez que uma instrução sensível (previligiada) for executada pelo SO hóspede.

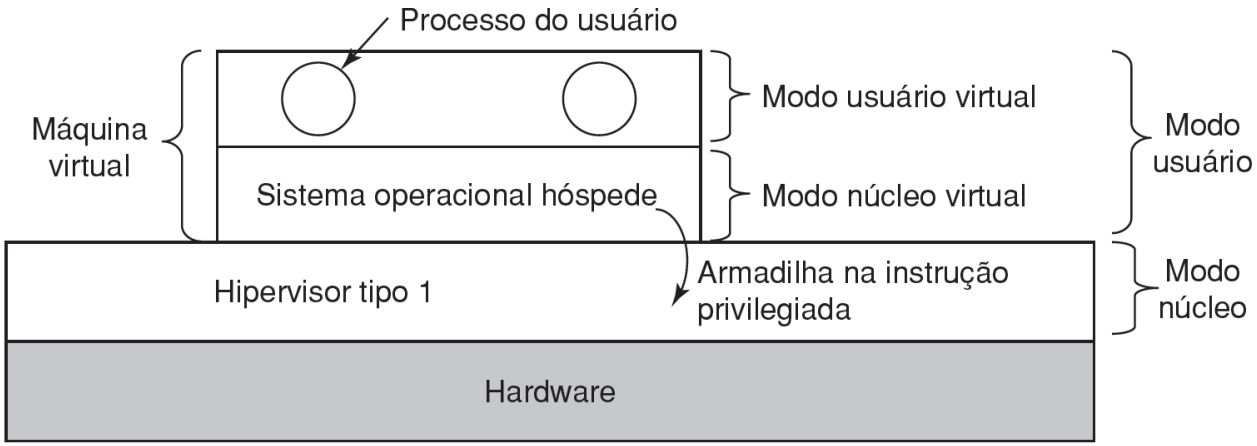


Figura 8.26 Quando o sistema operacional em uma máquina virtual executa uma instrução do modo núcleo, ela é capturada pelo hipervisor se a tecnologia de virtualização estiver presente.

Paravirtualização: utiliza um SO hóspede (*guest*) modificado, com instruções sensíveis removidas. O hipervisor é, de fato, um *microkernel* modificado que interfaceia de forma mais direta e otimizada com o SO hóspede (modificado).

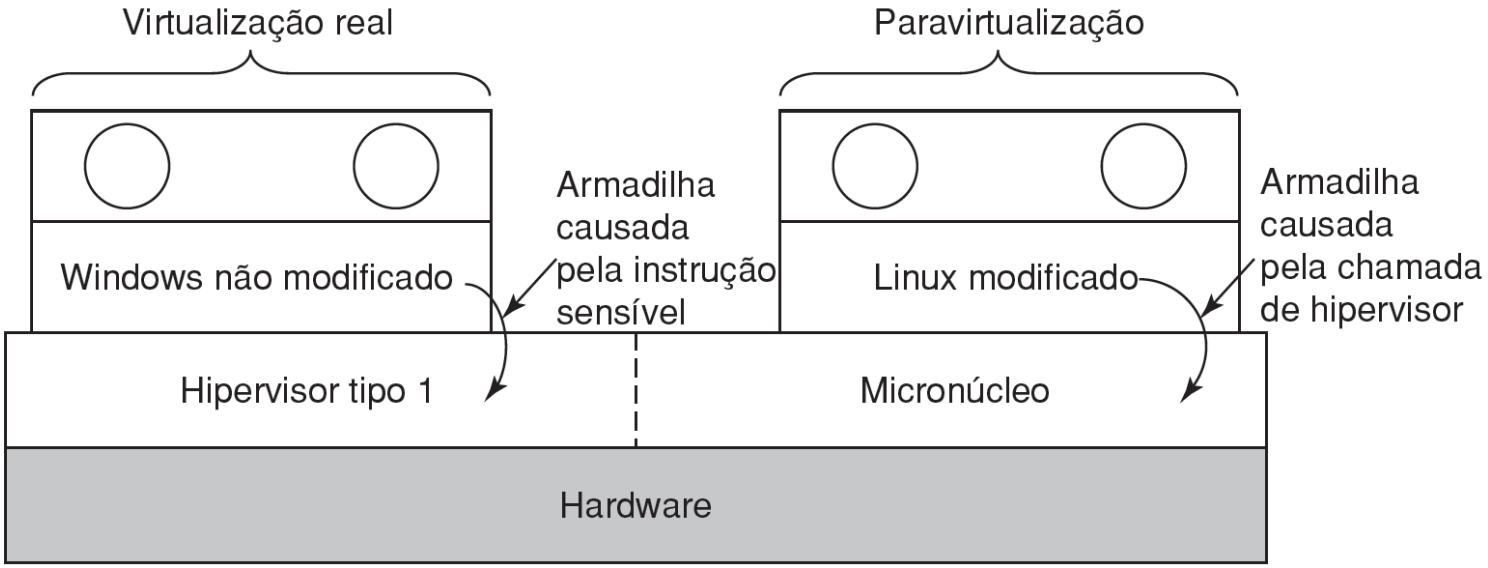


Figura 8.27 Um hipervisor controlando tanto uma virtualização real quanto uma paravirtualização.

VMI (*virtual machine interface*): interface de máquina virtual, forma uma camada de baixo nível que faz a interface padrão (independente de *hardware* ou hipervisor) do SO hóspede **modificado** com o *hardware* ou hipervisor.

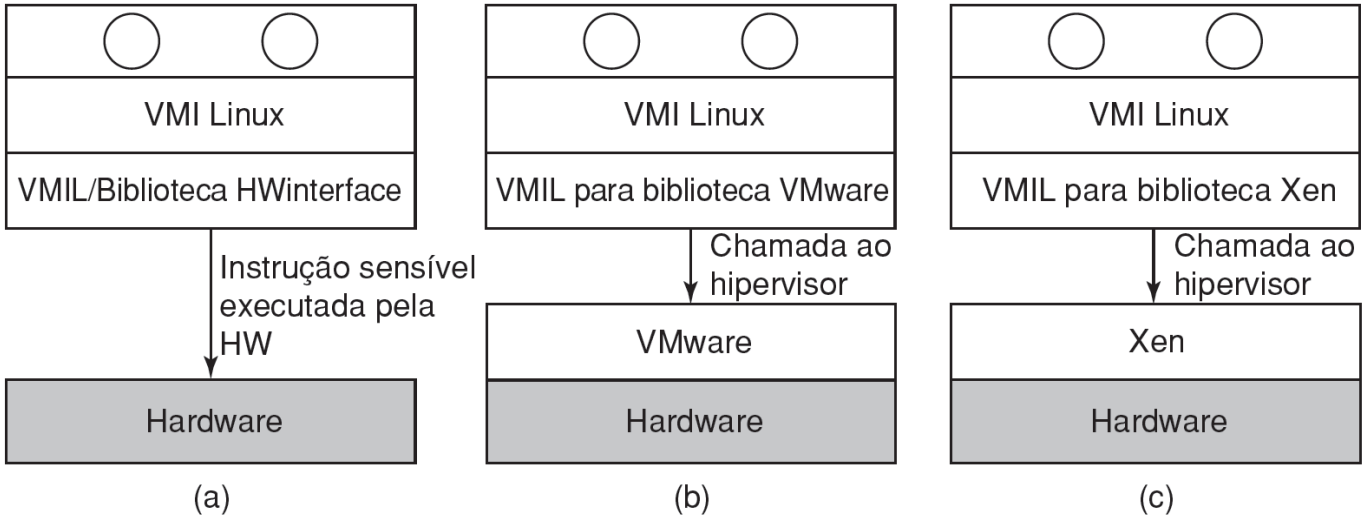
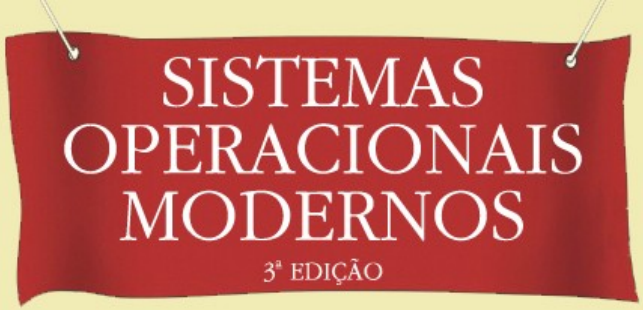


Figura 8.28 Uma interface da máquina virtual Linux funcionando (a) em uma máquina convencional; (b) com VMWare; (c) com Xen.



Sistemas distribuídos (distributed systems)

Item	Multiprocessador	Multicomputador	Sistema distribuído
Configuração do nó	CPU	CPU, RAM, interface de rede	Computador completo
Periféricos do nó	Tudo compartilhado	Exc. compartilhada, talvez disco	Conjunto completo por nó
Localização	Mesmo rack	Mesma sala	Possivelmente espalhado pelo mundo
Comunicação entre nós	RAM compartilhada	Interconexão dedicada	Rede tradicional
Sistemas operacionais	Um, compartilhado	Múltiplos, mesmo	Possivelmente todos diferentes
Sistemas de arquivos	Um, compartilhado	Um, compartilhado	Cada nó tem seu próprio
Administração	Uma organização	Uma organização	Várias organizações

Tabela 8.1 Comparação de três tipos de sistemas com múltiplas CPUs.

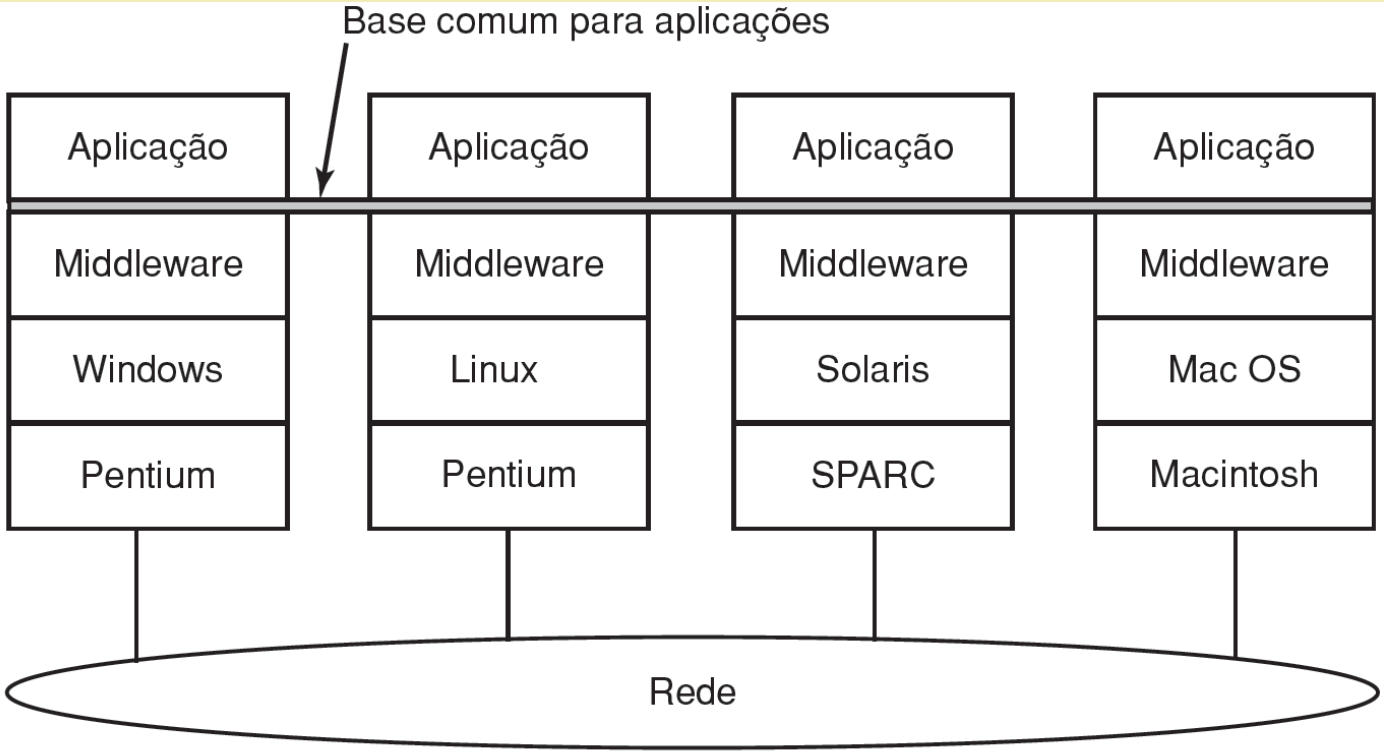
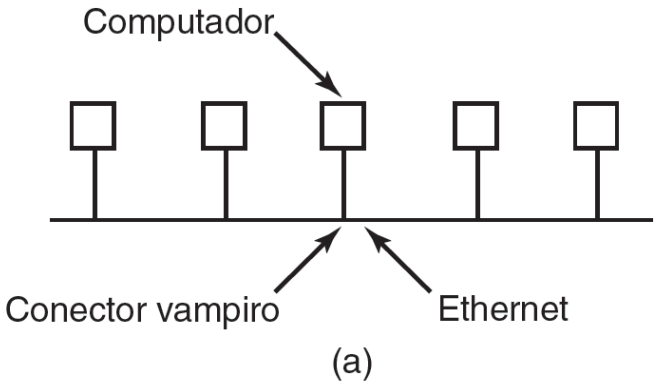


Figura 8.29 Posicionamento do middleware em um sistema distribuído.

Rede local IEEE 802.3 (*Ethernet*)



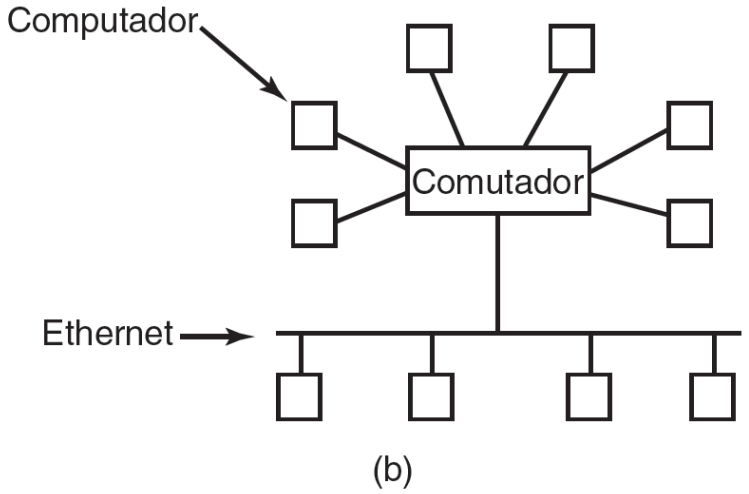


Figura 8.30 (a) Ethernet clássica. (b) Ethernet utilizando comutadores.

SISTEMAS OPERACIONAIS MODERNOS
3ª EDIÇÃO

A Internet

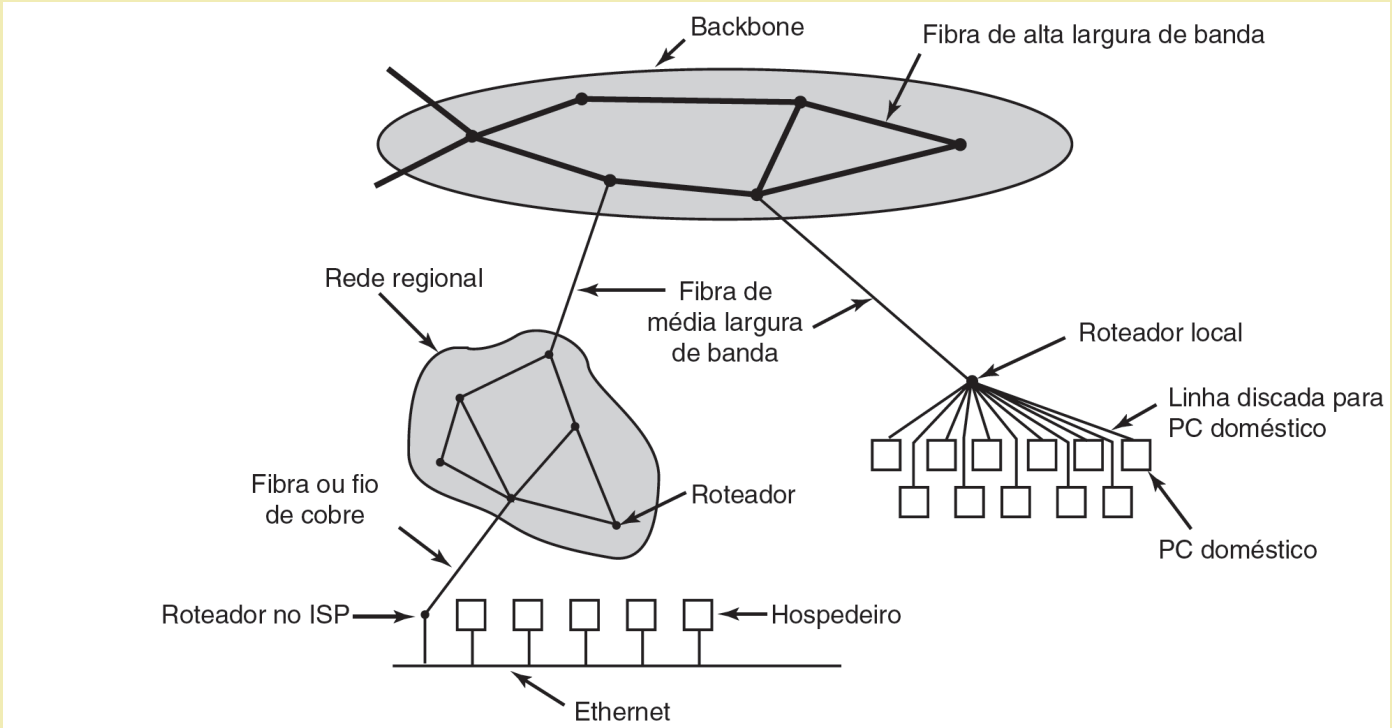
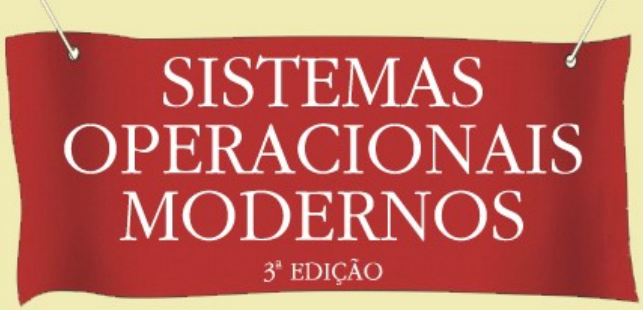


Figura 8.31 Uma parte da Internet.



Protocolos de rede

	Serviço	Exemplo
Orientado a conexão	Fluxo de mensagens confiável	Sequência de páginas de um livro
	Fluxo de bytes confiável	Login remoto
	Conexão não confiável	Voz digitalizada
Sem conexão	Datagrama não confiável	Pacotes de teste de rede
	Datagrama com confirmação	Correio registrado
	Solicitação-réplica	Consulta a um banco de dados

Tabela 8.2 Seis tipos diferentes de serviços de rede.

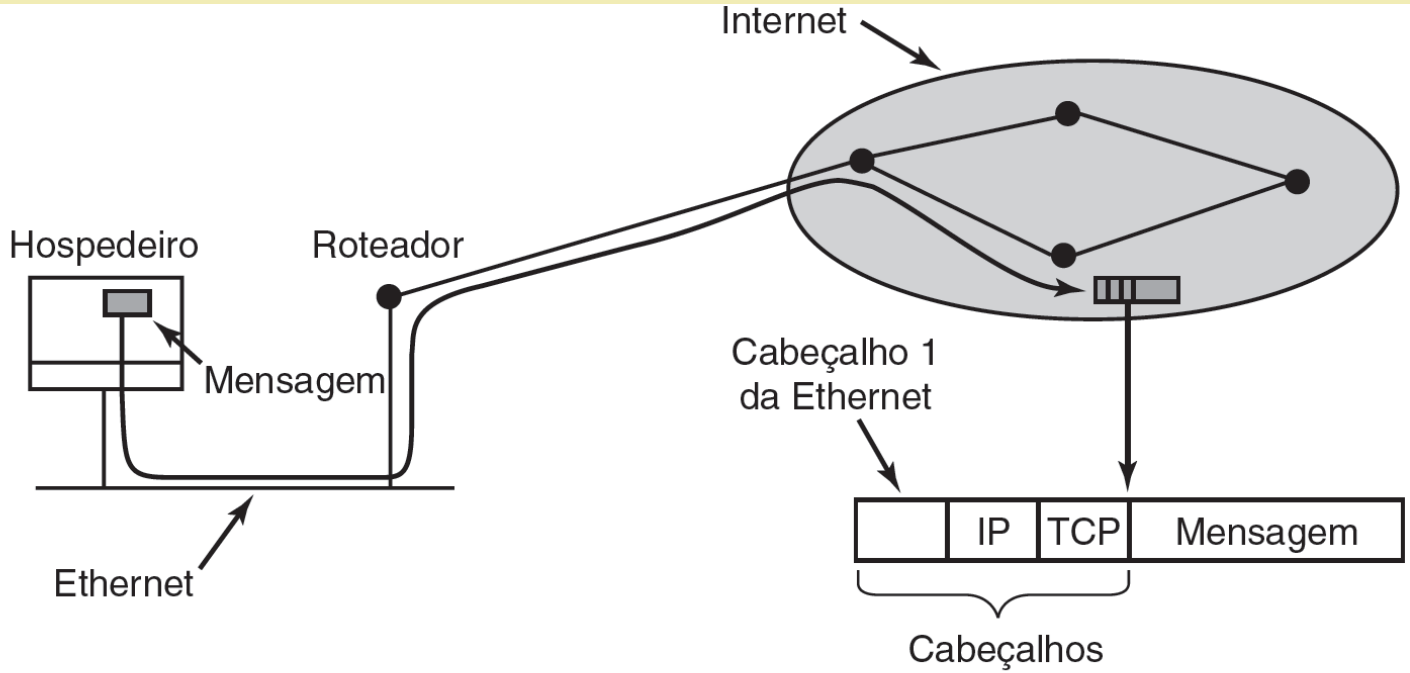


Figura 8.32 Acúmulo de cabeçalhos de pacotes.

SISTEMAS OPERACIONAIS MODERNOS
3ª EDIÇÃO

Middleware com base em documentos (*World Wide Web*, WWW)

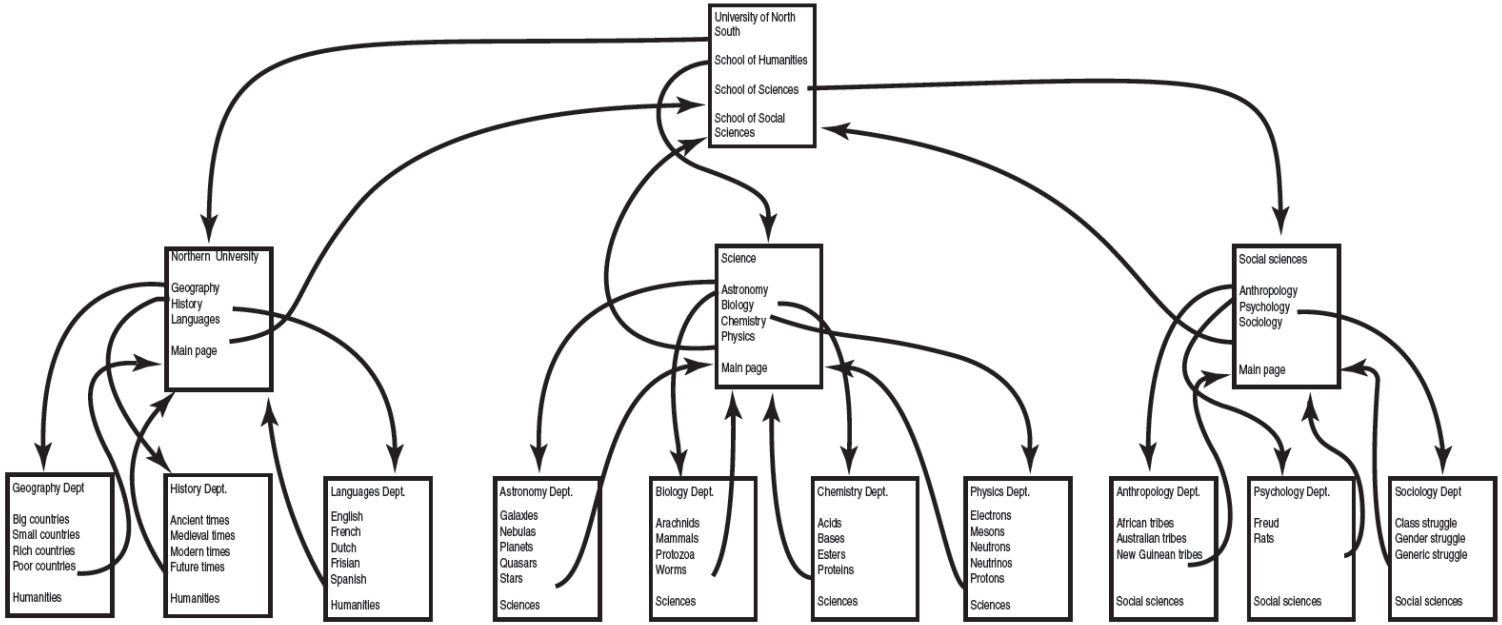
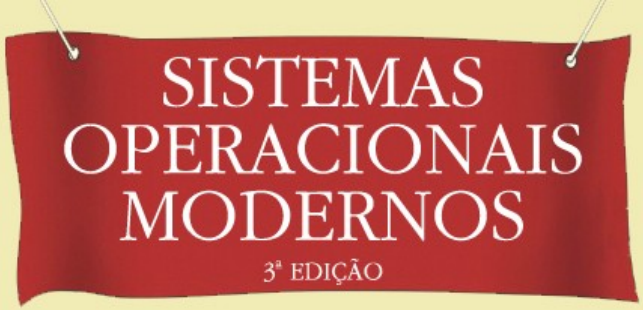


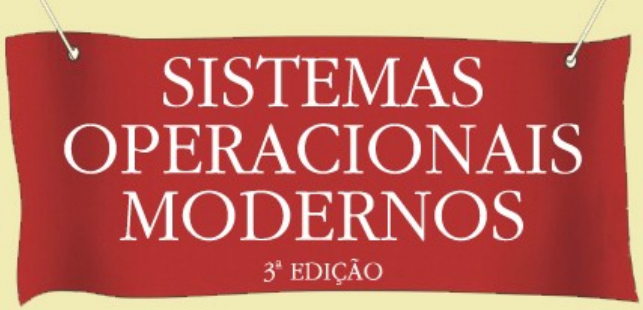
Figura 8.33 A Web é um grande grafo dirigido de documentos.



Quando o navegador chega à página *http://www.minix3.org/doc/faq.html.*

1. O navegador pergunta ao DNS pelo endereço IP de www.minix3.org.
2. DNS responde com 130.37.20.20.
3. O navegador abre uma conexão TCP com a porta 80 do endereço 130.37.20.20.
4. Ele, então, envia uma requisição perguntando pelo arquivo *doc/faq.html*.

. . .



. . .

5. O servidor www.acm.org envia o arquivo *doc/faq.html*.
6. A conexãoTCP é liberada.
7. O navegador mostra todo o texto em *doc/faq.html*.
8. O navegador busca e mostra na tela as imagens em *doc/faq.html*.

Modelo de transferência em *middleware* baseado no sistema de arquivos: (a) arquivo é baixado e, caso seja alterado localmente, enviado novamente ao servidor; (b) todo o processamento (mesmo alteração) é feita (via requisições) no próprio servidor.

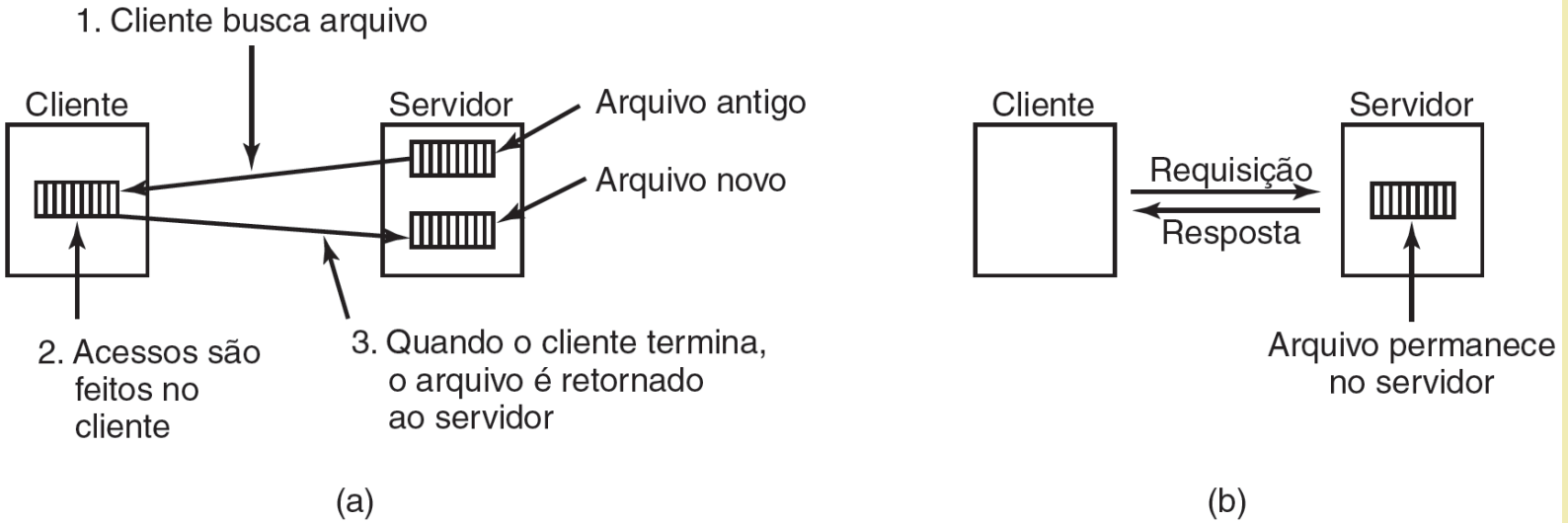


Figura 8.34 (a) O modelo *upload/download*. (b) O modelo de acesso remoto.

A hierarquia de diretório: dependendo da configuração, clientes podem ter visões diferentes.

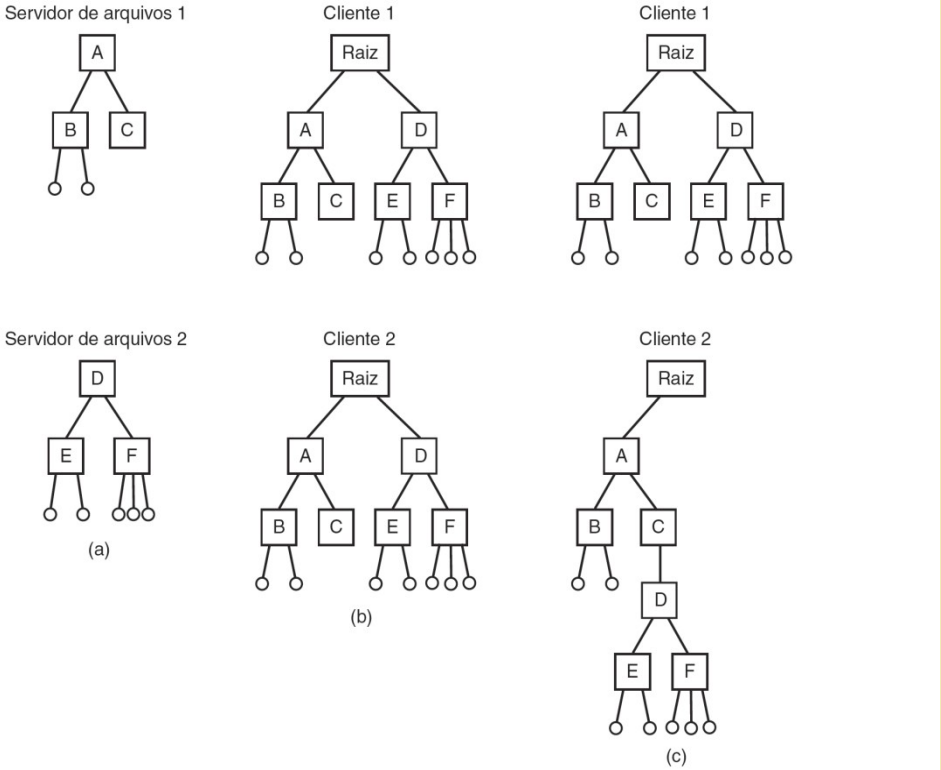
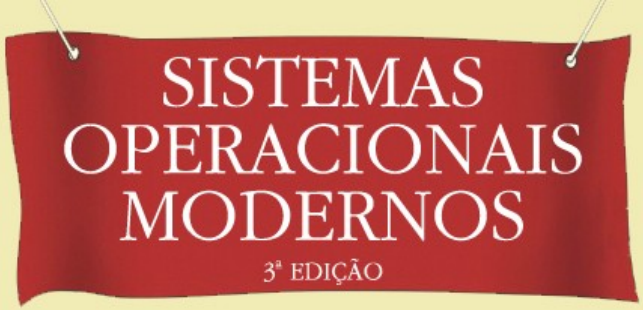


Figura 8.35 (a) Dois servidores de arquivos. Os quadrados são diretórios e os círculos são arquivos. (b) Um sistema no qual todos os clientes têm a mesma visão do sistema de arquivos. (c) Um sistema no qual clientes diferentes podem ter visões diferentes do sistema de arquivos.



Transparência de nomeação

Três abordagens comuns para nomeação de arquivos e diretórios:

1. Nomeação de máquina + caminho, tal como */machine/path* ou *machine:path*.
2. Montagem de sistemas de arquivos remotos sobre a hierarquia local de arquivos.
3. Um único espaço de nomes que parece o mesmo em todas as máquinas.

SISTEMAS OPERACIONAIS MODERNOS
3ª EDIÇÃO

Semântica do compartilhamento de arquivos

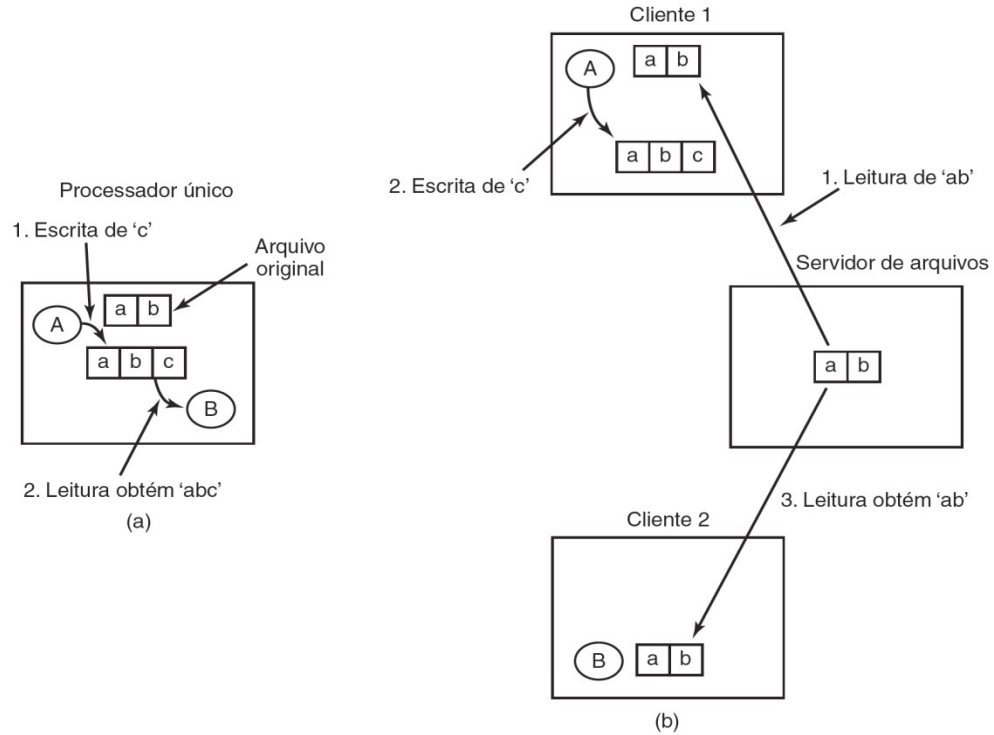


Figura 8.36 (a) Consistência sequencial. (b) Em um sistema distribuído com cache, a leitura de um arquivo pode retornar um valor obsoleto.