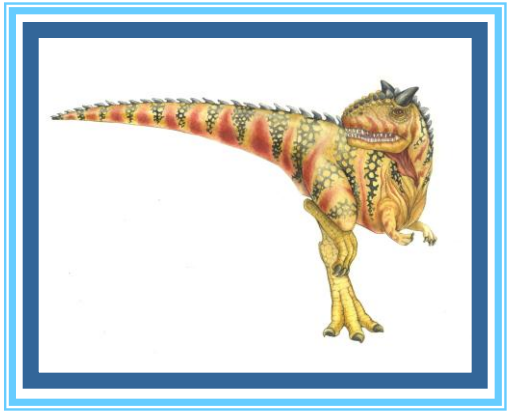
Capítulo 11: Implementação de Sistemas de Arquivos

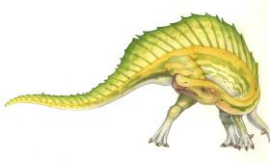


Sobre a apresentação (About the slides)

Os slides e figuras dessa apresentação foram criados por Silberschatz, Galvin e Gagne em 2009. Esse apresentação foi modificada por Cristiano Costa (cac@unisinos.br). Basicamente, os slides originais foram traduzidos para o Português do Brasil.

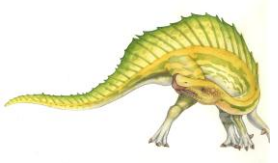
É possível acessar os slides originais em http://www.os-book.com

Essa versão pode ser obtida em http://www.inf.unisinos.br/~cac



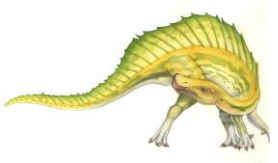
Capítulo 11: Implementação de Sistemas de Arquivos

- Estrutura do Sistemas de Arquivos
- Implementação de Sistemas de Arquivos
- Implementação de Diretório
- Métodos de Alocação
- Gerenciamento do Espaço Livre
- Eficiência e Desempenho
- Recuperação
- Sistemas de Arquivos Baseados em Registro de Operações (*Log*)
- NFS



Objetivos

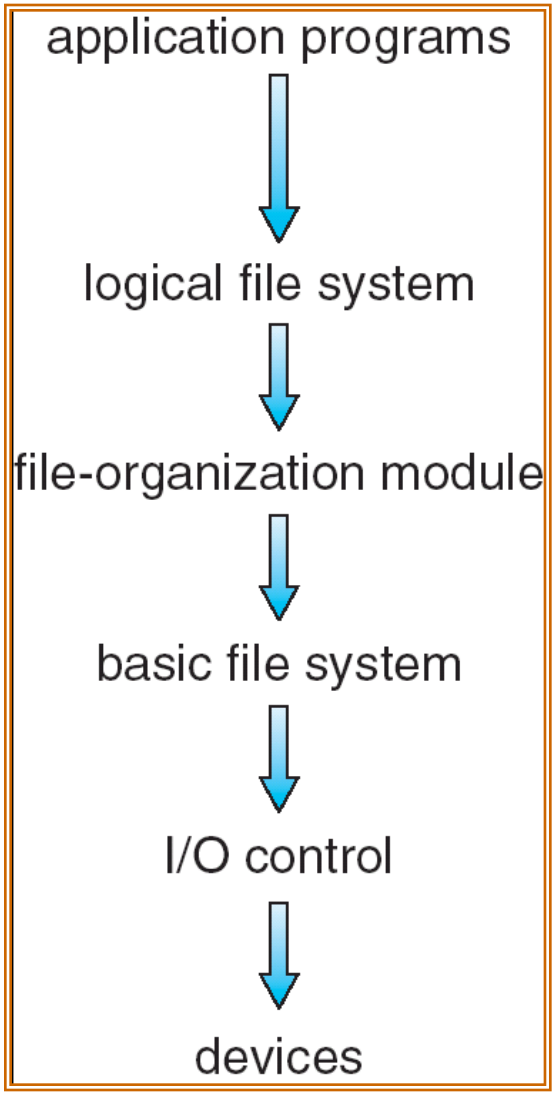
- Descrever os detalhes de implementação de sistemas de arquivos locais e estruturas de diretórios
- Descrever a implementação de sistemas de arquivos remotos
- Discutir alocação de blocos e algoritmos de blocos livres e relação custo-benefício

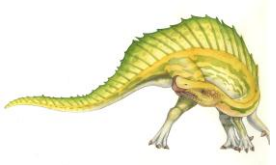


Estrutura de Sistemas de Arquivos

- Estrutura do Arquivo
  - Unidade de Armazenamento Lógica
  - Coleção de informações relacionadas
- Sistema de arquivos organizado em camadas.
- **Sistema de arquivos** reside em armazenamento secundário (discos).
  - Fornecem acesso eficiente e conveniente aos discos permitindo o fácil armazenamento de dados e sua localização
- **Bloco de Controle de Arquivo** *(File control block)* – estrutura de armazenamento contendo informações sobre um arquivo.
- **Driver de dispositivo** (*Device driver)* controla o dispositivo físico

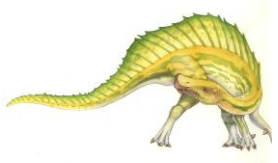
Sistemas de Arquivos em Camadas





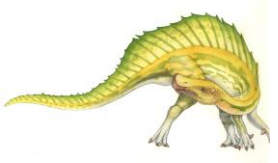
Implementação de Sistemas de Arquivos

- **Boot control block** contém informações necessárias pelo sistema para iniciar o SO naquele volume
- **Volume control block** contém detalhes do volume
- Estrutura de diretório organiza os arquivos
- **File Control Block (FCB)** por arquivo contém vários detalhes sobre os arquivos



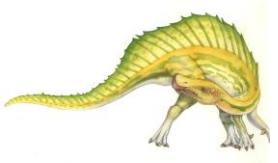
Um Bloco de Controle de Arquivo Típico

file permissions
file dates (create, access, write)
file owner, group, ACL
file size
file data blocks or pointers to file data blocks

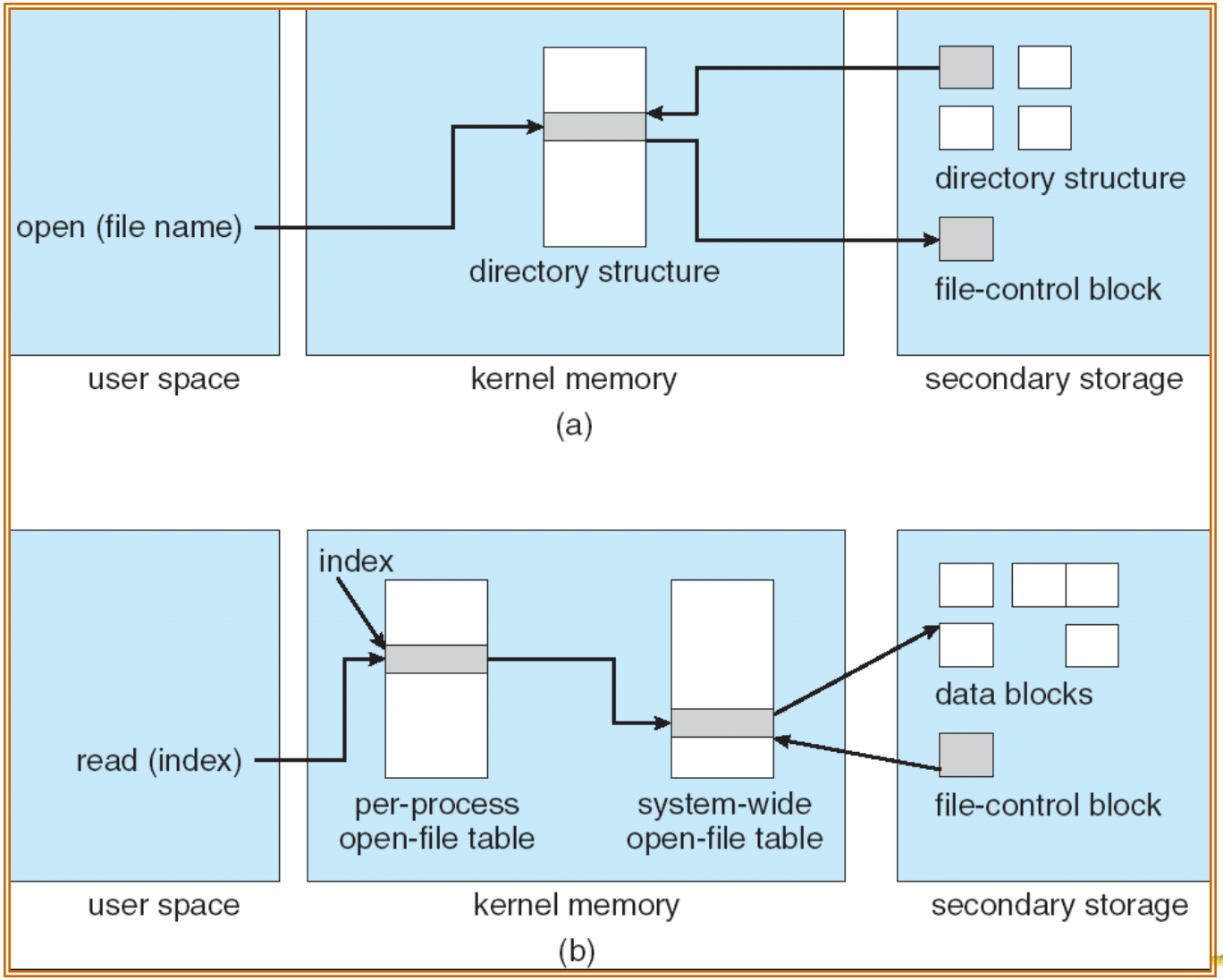


Estrutura de Sistemas de Arquivos na memória

- As figuras a seguir ilustram as estruturas de sistemas de arquivos necessárias e fornecidas pelo sistema operacional
- Figura 12-3(a) refere a abertura de um arquivo.
- Figura 12-3(b) refere a leitura de um arquivo.



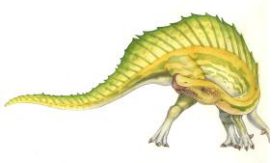
Estrutura de Sistemas de Arquivos na memória



Implementação de Diretório

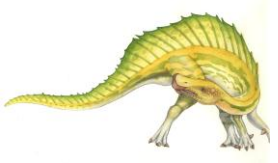
- **Lista Linear** de nomes de arquivos com ponteiros para os blocos de dados
  - Simples de programar
  - Execução lenta

- **Tabela Hash** – lista linear com estruturas de dados *hash*.
  - Diminui o tempo de procura no diretório
  - **colisões** – situações nas quais dois nome de arquivos obtém a mesma localização
  - Tamanho fixo



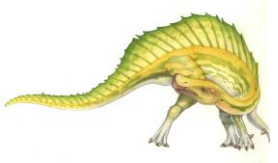
Métodos de Alocação

- Um método de alocação indica como os blocos de disco são alocados aos arquivos:
- **Alocação Contígua**
- **Alocação Encadeada**
- **Alocação Indexada**

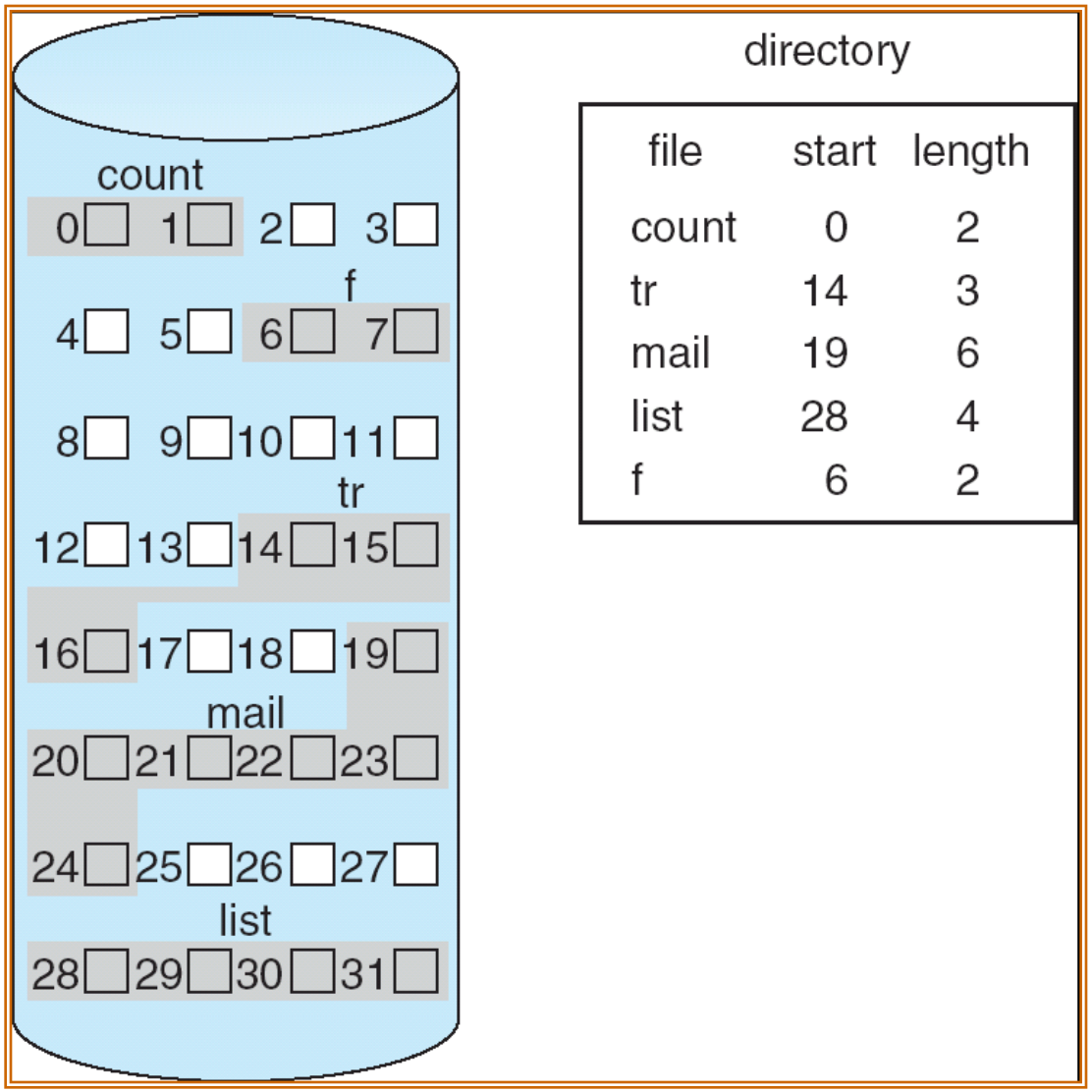


Alocação Contígua

- Cada arquivo ocupa um conjunto de blocos contíguos no disco
- Simples – somente é necessário armazenar a localização inicial (número do bloco) e o tamanho do arquivo (quantidade de blocos)
- Acesso Direto (randômico)
- Perda de Espaço (problema da alocação dinâmica)
- Arquivos não podem crescer

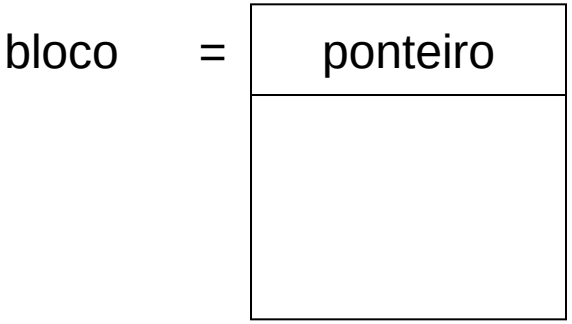


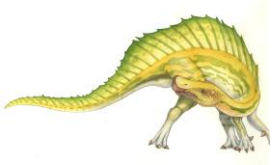
Exemplo de Alocação Contígua



Alocação Encadeada

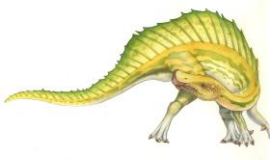
- Cada arquivo é uma lista encadeada de blocos em disco: blocos podem ser espalhados em qualquer lugar do disco.



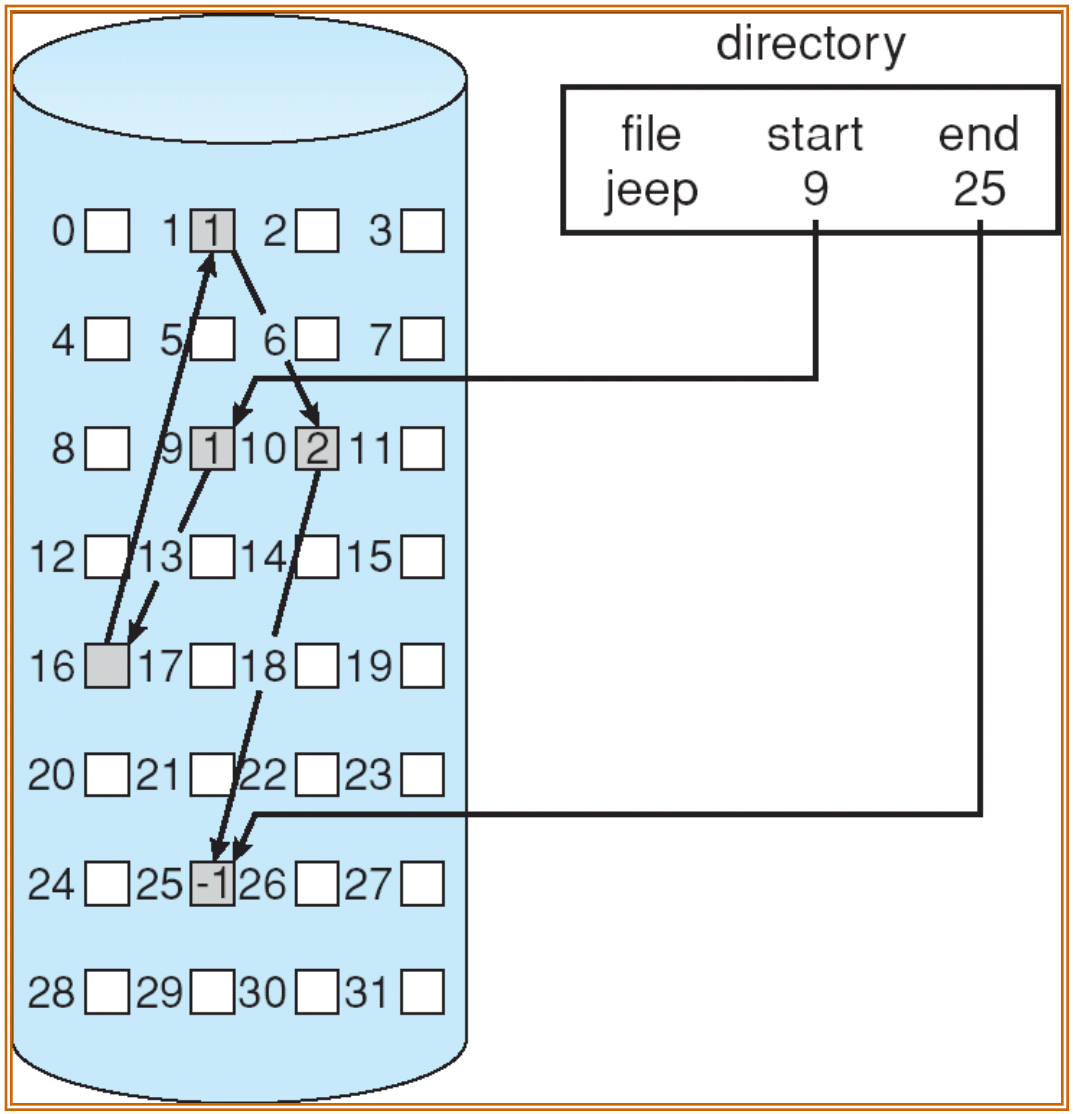


Alocação Encadeada (Cont.)

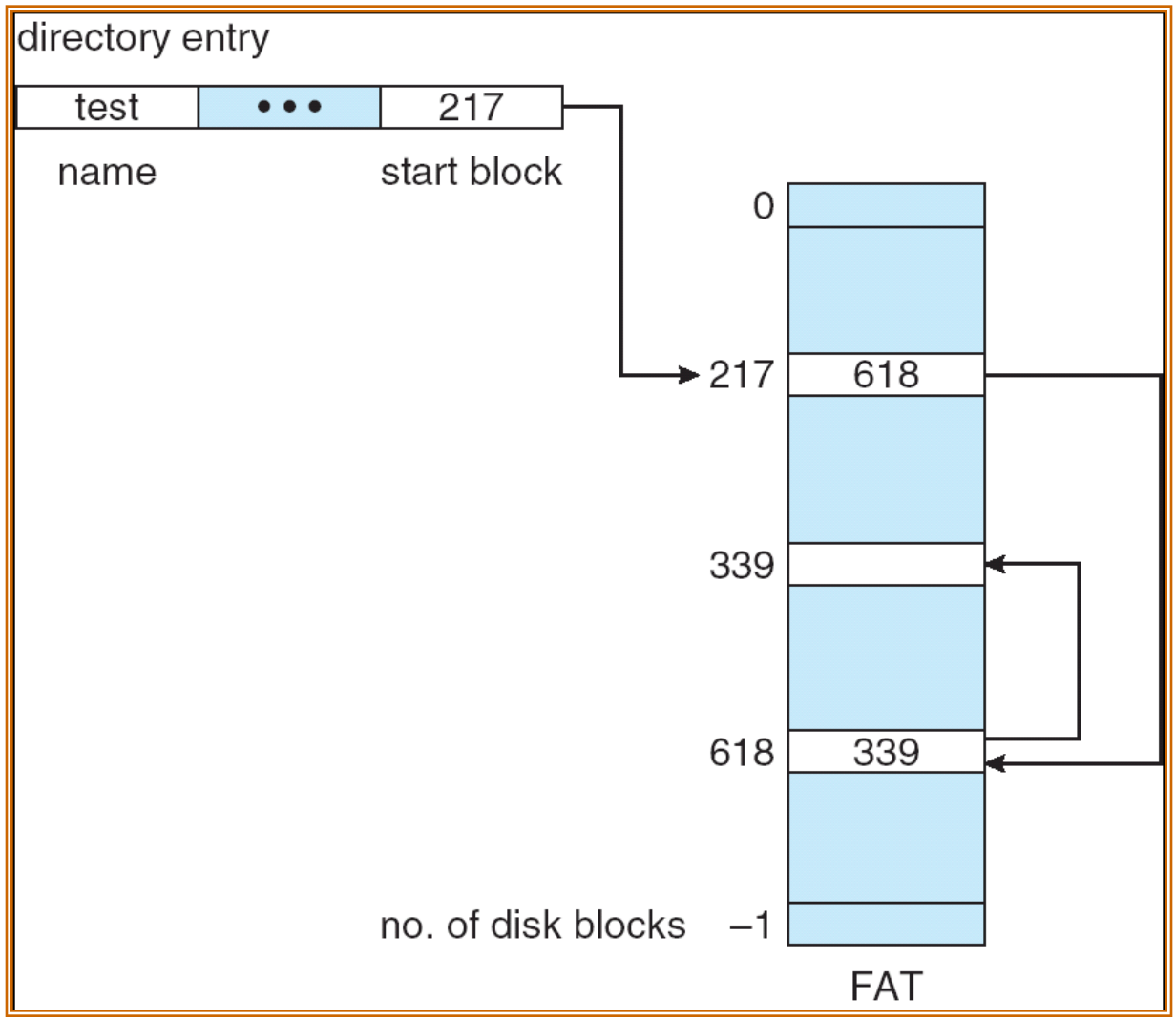
- Simples – necessita somente do endereço inicial
- Sistema de Gerenciamento de Espaço Livre – sem perda de espaço
- Sem acesso direto (randômico)
- Desvantagens:
  - percorrer o encadeamento até encontrar posição
  - armazenamento requerido para ponteiros, desperdício de 0,78% (512 bytes – 4 bytes)
  - paliativo: uso de clusters de blocos = fragmentação interna
  - possíveis consequências da perda ou dano de um ponteiro?

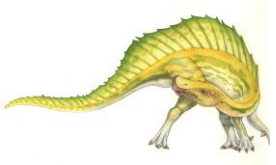


Exemplo de Alocação Encadeada



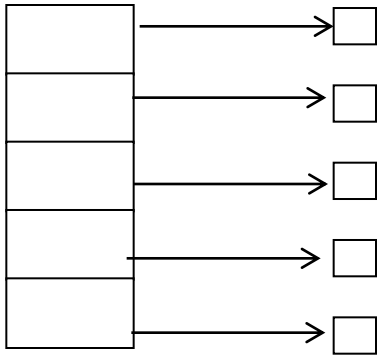
File-Allocation Table - FAT



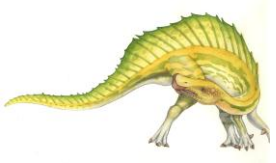


Alocação Indexada (em Tabela)

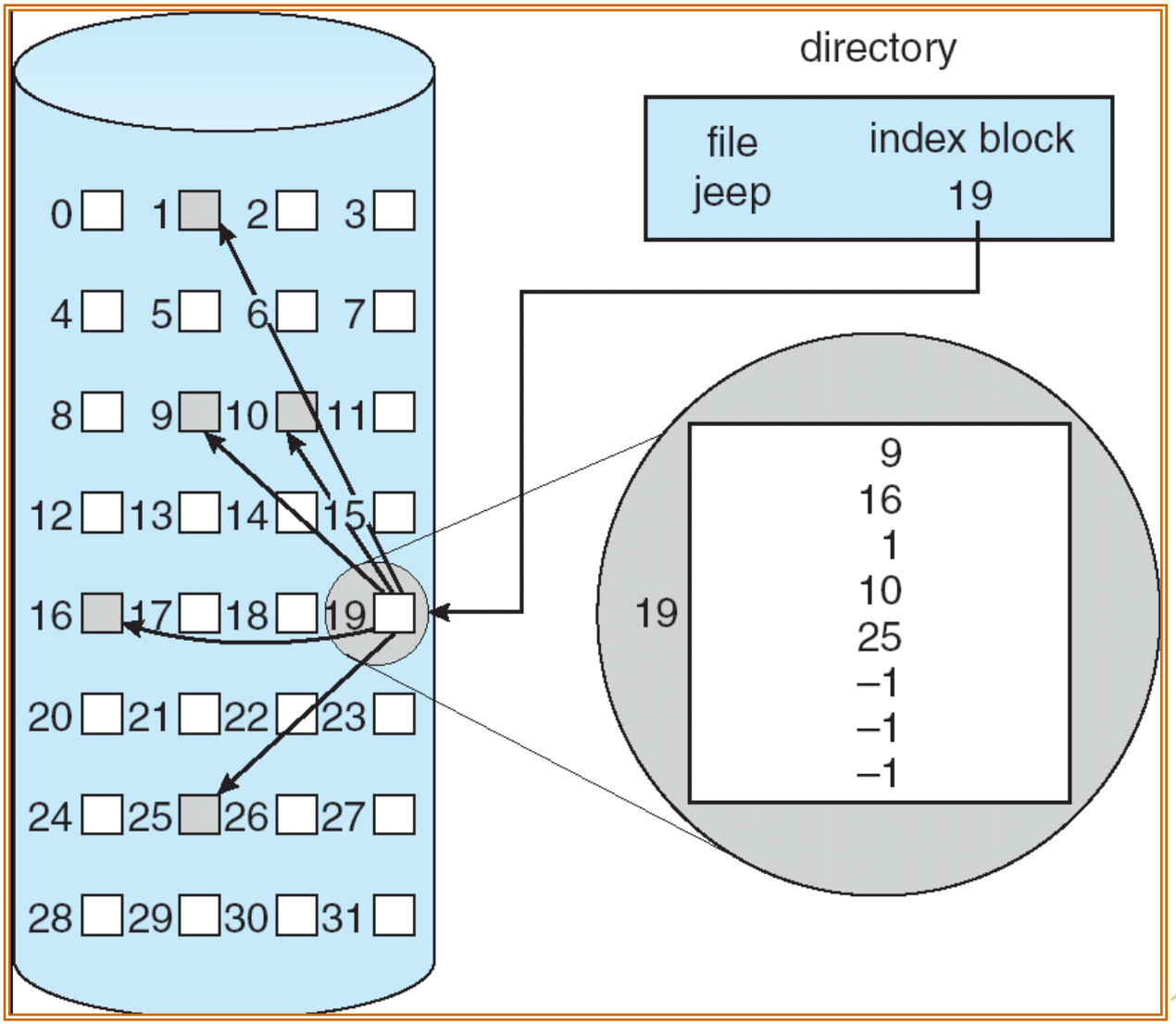
- Juntar todos os ponteiros em uma tabela de índices (*index block).*
- Também chamados de *Inodes* ou Nós-Índices
- Visão Lógica.



index table

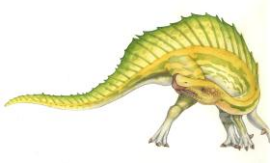


Examplo de Alocação Indexada



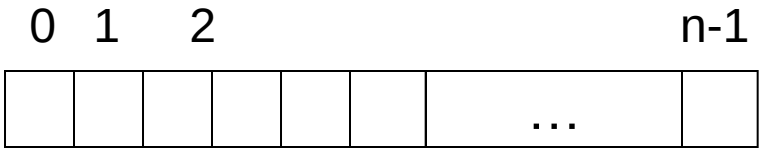
Alocação Indexada (Cont.)

- Esquema encadeado
- Índice multinível
- Esquema combinado



Gerenciamento do Espaço Livre

- **Vetor ou mapa de Bits** (*n* blocos)



$$bit[i] = \begin{cases} 0 \Rightarrow \text{bloco}[i] \text{ livre} \\ 1 \Rightarrow \text{bloco}[i] \text{ ocupado} \end{cases}$$

Cálculo do número do Bloco

(número de bits por palavra) *
(número de palavras com valor 0) +
deslocamento do primeiro bit 1

Gerenciamento do Espaço Livre (Cont.)

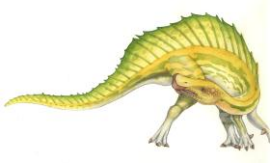
- Vetor de bits requer espaço extra. Exemplo:

  tamanho do bloco = 2^{12} bytes

  tamanho do disco = 2^{30} bytes (1 gigabyte)

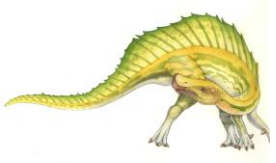
  $n = 2^{30}/2^{12} = 2^{18}$ bits (ou 32K bytes)

- Fácil para manter arquivos contíguos
- Lista encadeada (Lista de blocos livres)
  - Difícil manter alocação contígua
  - Sem desperdício de espaço
- Agrupamento
- Contagem

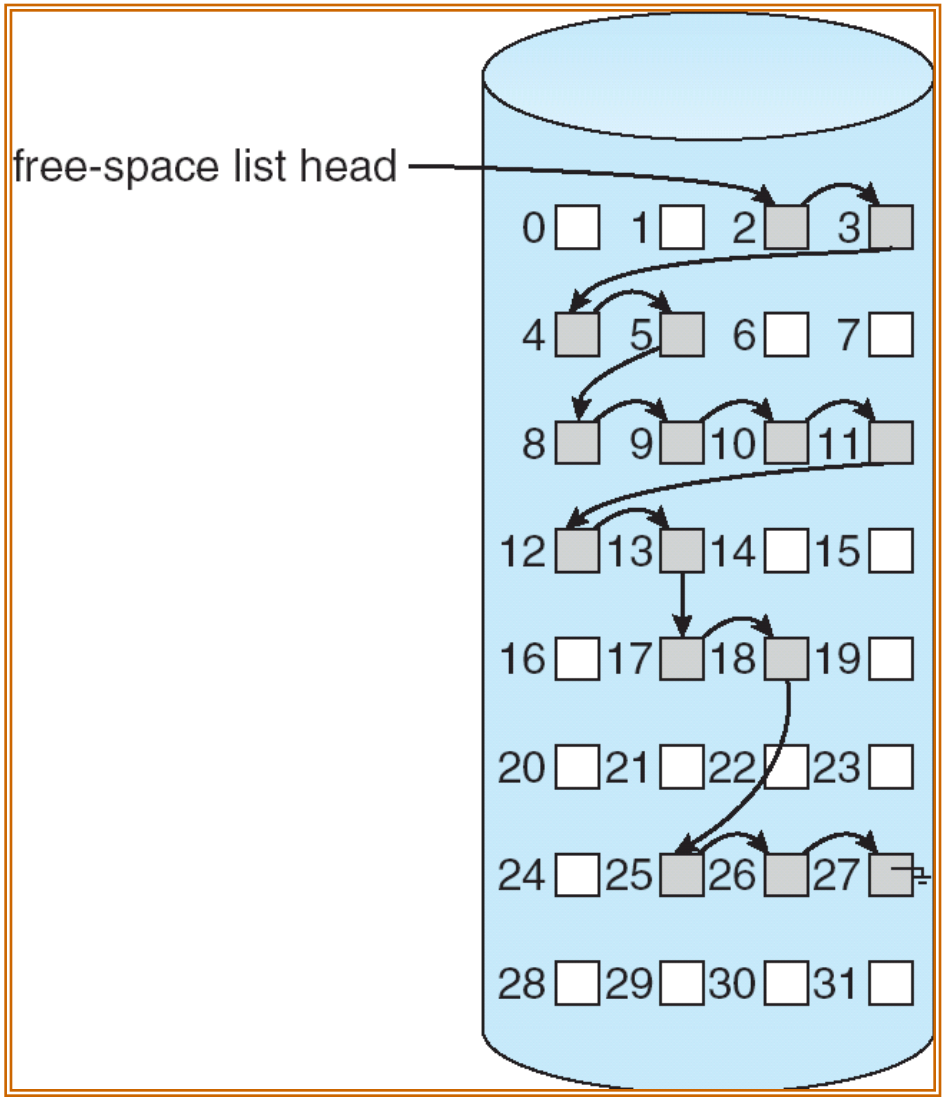


Gerenciamento do Espaço Livre (Cont.)

- **Outras técnicas:**
- Lista encadeada (Lista de blocos livres)
  - Difícil manter alocação contígua
  - Sem desperdício de espaço
- Agrupamento
  - Agrupa n blocos livres no primeiro bloco livre, assim sucessivamente
- Contagem
  - Registra no primeiro bloco livre o nº de blocos livres contíguos
- Mapa de espaços
  - Mapa de bits. Ex: ZFS da sun

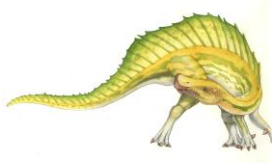


Lista Encadeada de Espaço Livre no Disco



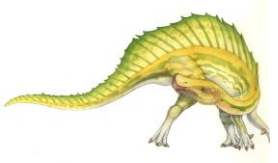
Eficiência e Desempenho

- **Disco é o principal gargalo para o desempenho do sistema**
- Eficiência depende de:
  - Alocação de disco e algoritmos de diretório
  - Tipos de dados mantidos na entrada do arquivo no diretório
- Desempenho
  - *cache* de disco – seção separada de memória principal para blocos freqüentemente usados
  - Liberação antecipada (*free-behind*) e leitura antecipada (*read-ahead*) – técnicas para otimizar o acesso sequencial
  - Aumentar o desempenho do PC dedicando partes da memória como discos virtuais (*RAM disks*).

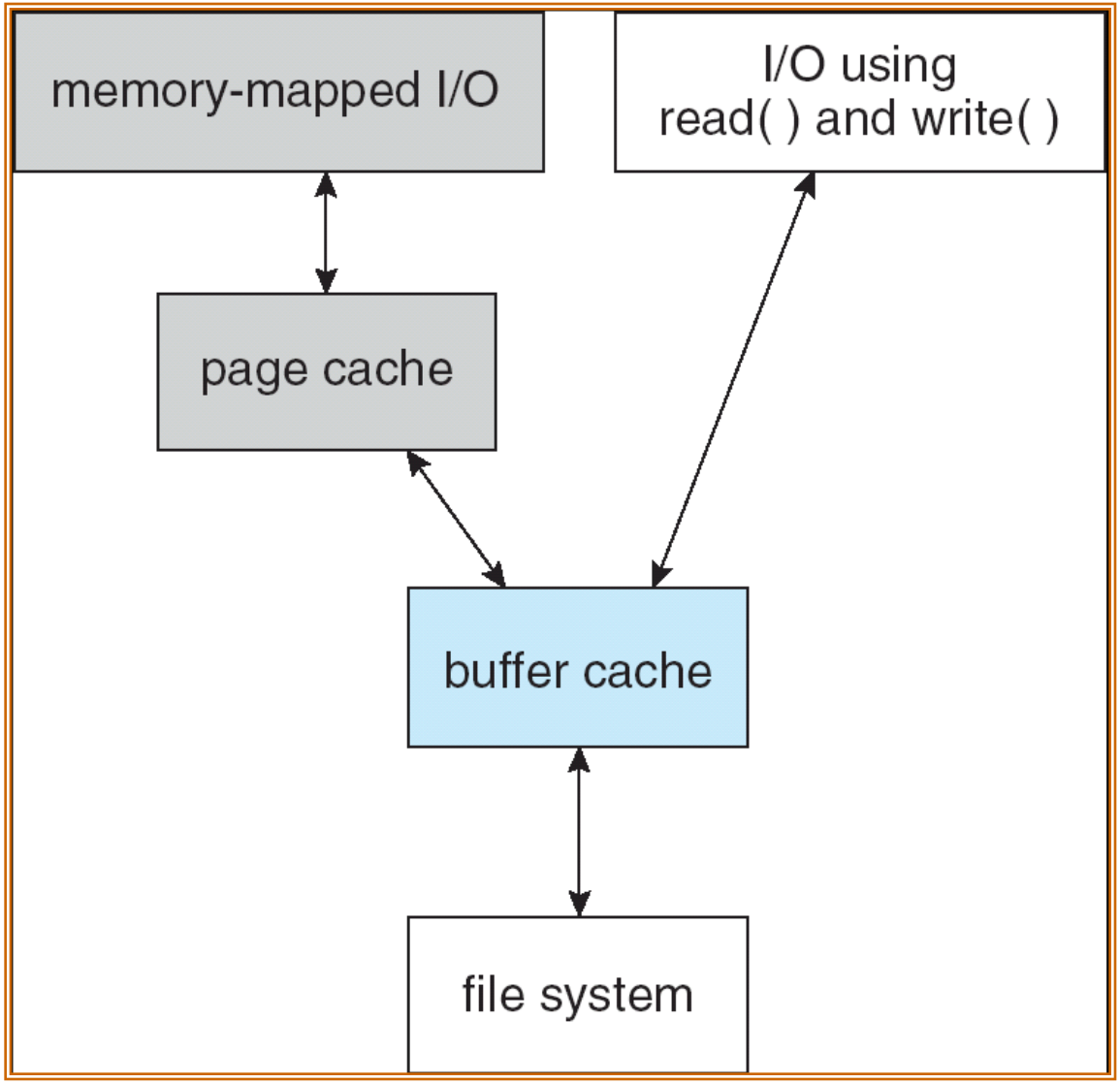


Cache de Páginas

- Um **cache de páginas** armazena páginas ao invés de blocos de disco usando técnicas de memória virtual
- E/S mapeado em memória usa cache de página
- Rotina de E/S através do sistema de arquivos usa o cache de buffer (do Disco)
- Isso leva a figura a seguir

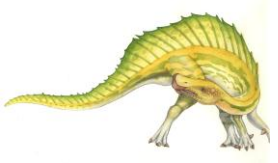


E/S sem um Cache de Buffer Unificado

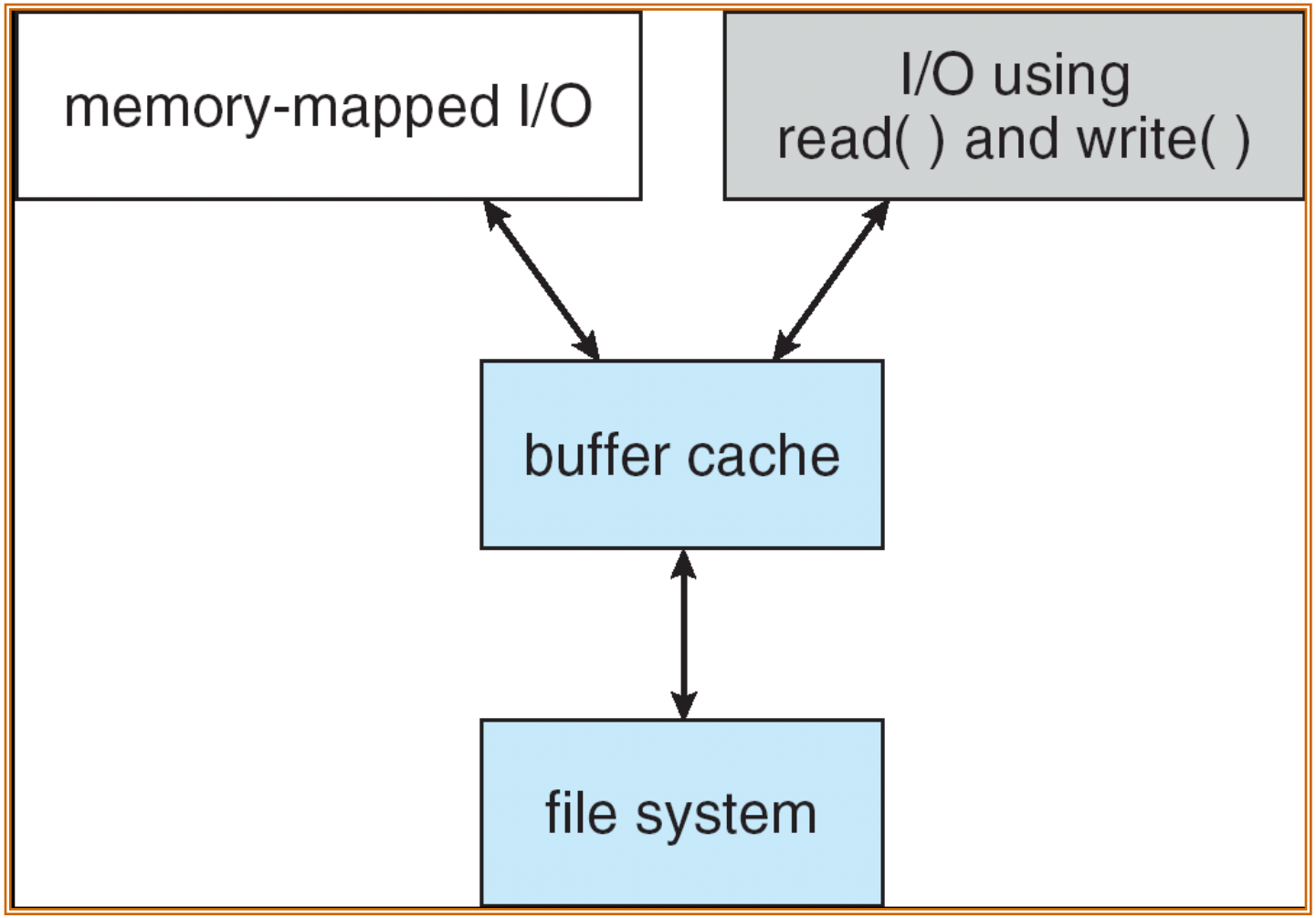


Cache de Buffer Unificado

- Um cache de buffer unificado usa o mesmo cache de páginas para armazenar páginas mapeadas na memória e E/S ao sistema de arquivos padrão

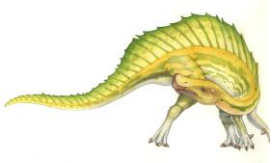


E/S com um Cache de Buffer Unificado



Recuperação

- Arquivos e diretórios são mantidos em disco e memória, atenção para possíveis falhas no sistema não resultem em perdas ou inconsistências
- Teste de Consistência – comparar dados nas estruturas de diretórios com blocos de dados no disco, e tentar consertar as inconsistências.
- Usar programas de sistemas para criar cópias de segurança (*back up)* do disco para outros dispositivos de armazenamento (disquete, disco magnético).
- Recuperar arquivos ou discos perdidos restaurando dados do *backup*.

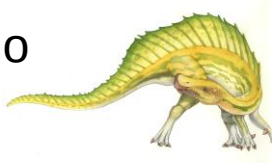


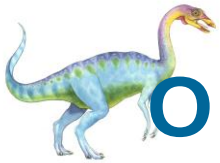
Sistemas de Arquivos Baseados em Registro de Operações

- Sistemas de arquivos com **Estrutura de Registro de Operações - *Log*** (ou *journaling*) registram cada atualização no sistema de arquivos como uma **transação**

- Todas as transações são escritas em um **registro de operações**
  - Uma transação é considerada encerrada com sucesso (*commit*) uma vez que é gravada no registro de operações
  - Entretanto, o sistema de arquivos pode não ter sido atualizado ainda

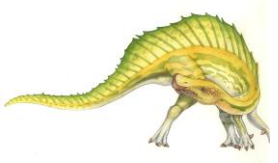
- As transações do registro de operações são gravadas de forma assíncrona no sistema de arquivos
  - Quando o sistema de arquivos é modificado, a transação é removida do registro de operações
- Se o sistema de arquivos trava, todas as transações restantes no log precisam ainda ser realizadas





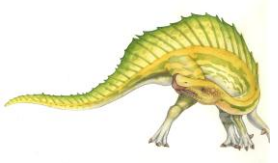
O *Network File System* (NFS) da Sun

- Uma implementação e especificação de um software para acessar arquivos remotos através das redes locais (ou metropolitanas)
- A implementação é parte dos sistemas operacionais Solaris e SunOS executando em estações de trabalho Sun usando um protocolo de datagramas não confiável (UDP/IP) e Ethernet



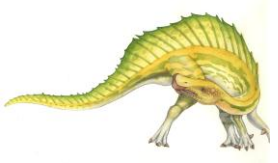
NFS (Cont.)

- Estações de trabalho interconectadas são vistas como um conjunto de máquinas independentes com sistemas de arquivos independentes. NFS permite compartilhamento entre esses sistemas de arquivos de forma transparente
  - Um diretório remoto é montado sobre um sistema de diretórios local
    - O diretório montado aparece como uma sub-árvore integrada ao sistema de arquivos local, substituindo as ramificações descendentes do subdiretório local
  - Especificação do diretório remoto para a operação de montagem não é transparente; O nome da máquina do diretório remoto deve ser fornecido
    - Arquivos no diretório remoto podem então ser acessados de forma transparente
  - Sujeito aos direitos de acesso, potencialmente qualquer sistema de arquivos (ou diretório nesse sistema de arquivos) pode ser montado remotamente em qualquer diretório local

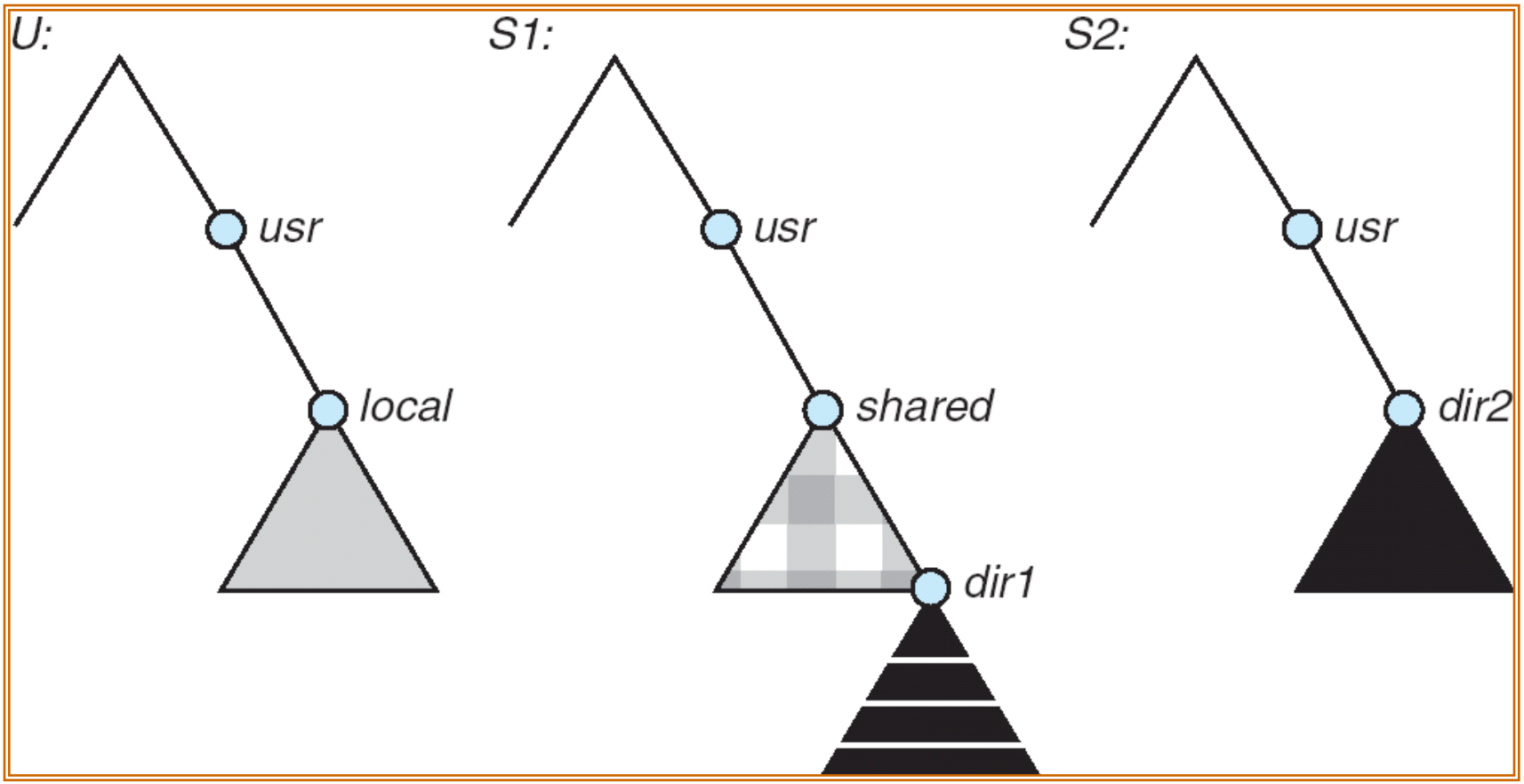


NFS (Cont.)

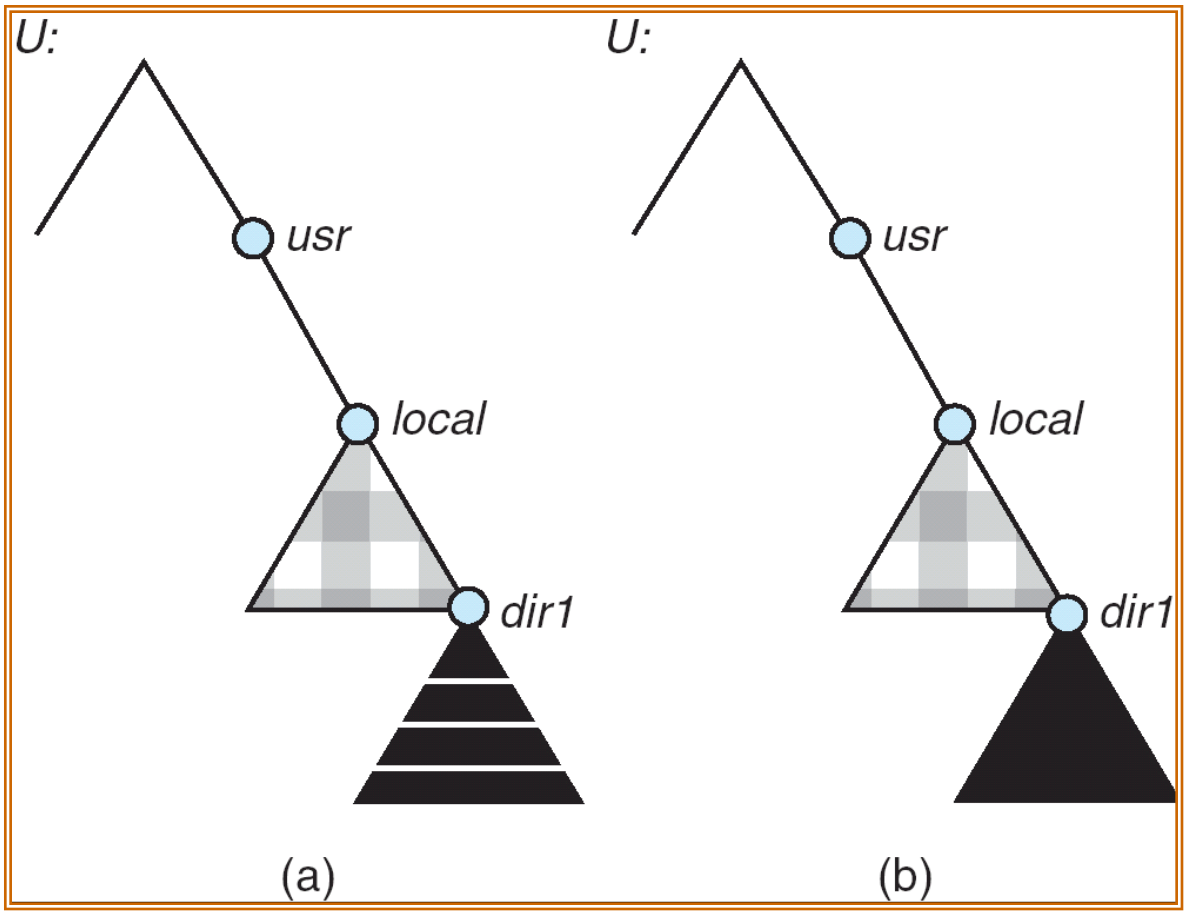
- NFS é projetado para operar em ambientes heterogêneos de diferentes máquinas, sistemas operacionais e arquiteturas de rede; a especificação do NFS independe dessas mídias
- A independência é obtida através do uso de primitivas de RPC construídas sobre o protocolo *External Data Representation* (XDR) usado entre duas interfaces independentes de implementação
- A especificação do NFS separa serviços fornecidos pelo mecanismo de montagem e serviços de acesso aos arquivos remotos



Três Sistemas de Arquivos Independentes



Montagem no NFS

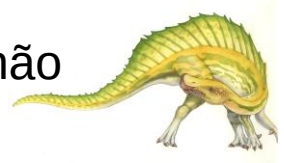


Mounts

Cascading mounts

Protocolo de Montagem do NFS

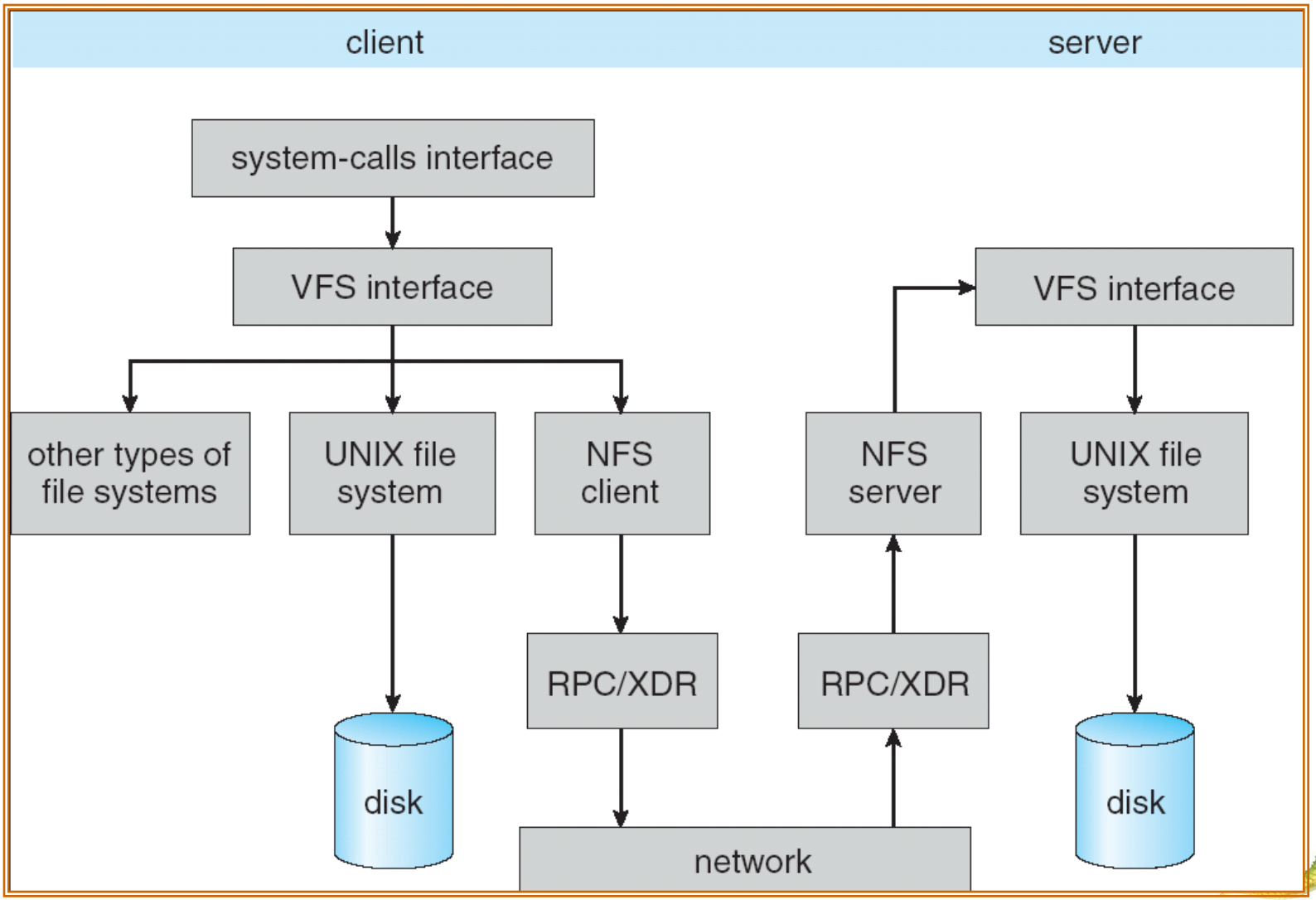
- Estabelece conexão lógica inicial entre cliente e servidor
- Operação de montagem inclui nome do diretório remoto a ser montado e o nome da máquina que armazena-o
  - Requisição de montagem é mapeada para o RPC correspondente e enviada ao servidor de montagem que executa na máquina servidora
  - *Export list (*Lista de Exportação) – especifica sistemas de arquivos locais que o servidor exporta para montagem junto com os nomes das máquinas que têm permissão para montá-los
- Seguindo uma requisição de montagem que está de acordo com a lista de exportação, o servidor retorna um manipulador de arquivo – uma chave para acesso futuro
- Manipulador de Arquivo (*handle*) – um identificador do sistema de arquivos e um número de inode para identificar diretório montado dentro do sistema de arquivos exportado
- A operação de montagem altera somente a visão do usuário e não afeta o lado do servidor



Protocolo do NFS

- Fornece um conjunto de chamadas de procedimentos remotos para operações em arquivos remotos. Os procedimentos suportam as seguintes operações:
  - procurar por um arquivo em um diretório
  - ler um conjunto de entradas de diretório
  - manipular links e diretórios
  - acessar atributos de arquivos
  - ler e escrever arquivos
- Servidores NFS são ***stateless*** (não armazenam estados); cada requisição tem que fornecer o conjunto completo de argumentos
  - NFS a partir da versão 4 é bem diferente, sendo ***statefull*** – armazena estados no servidor. Aqui são tratadas as versões anteriores
- Dados modificados devem ser gravados no disco do servidor antes dos resultados serem retornados ao cliente (perde as vantagens do uso de cache)
- O protocolo NFS não provê mecanismos de controle de concorrência

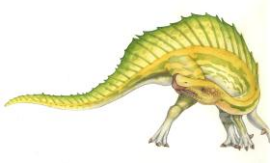
Visão Esquemática da Arquitetura NFS



Tradução de Caminhos (*Paths*) no NFS

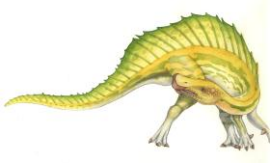
- Realizada quebrando o caminho em nomes de componentes e realizando uma chamada separada de *lookup* ao NFS para cada par de componentes (nome e *vnode* de diretório)

- Para tornar o *lookup* mais rápido, um cache de nome de diretório no lado do cliente armazena o *vnode* de nomes de diretórios remotos



Operações Remotas no NFS

- Existe quase uma correspondência um-para-um entre chamadas de sistemas UNIX regulares e o RPC do protocolo NFS (exceto para abertura e fechamento de arquivos – nas versões *stateless*)
- NFS adere ao paradigma de serviço remoto, mas emprega bufferização e técnicas de cache por razões de desempenho
- Cache de Blocos de Arquivos – quando um arquivo é aberto, o kernel verifica com o servidor remoto se deve buscar e trazer ou revalidar os atributos de cache
  - Blocos de Arquivos no cache são usados somente se os atributos correspondentes no cache estão atualizados
- Cache de atributos de arquivos – o cache de atributos é atualizado sempre que novos atributos chegam do servidor
- Clientes não liberam blocos de escrita até que o servidor confirme que os dados foram escritos no disco



Fim do Capítulo 11