

Capítulo 11: Implementação do sistema de arquivos



Objetivos

- Descrever os detalhes da implementação de sistemas de arquivos e estruturas de diretório locais
- Descrever a implementação de sistemas de arquivo remotos
- Discutir algoritmos e opções de a alocação em bloco e bloco livre

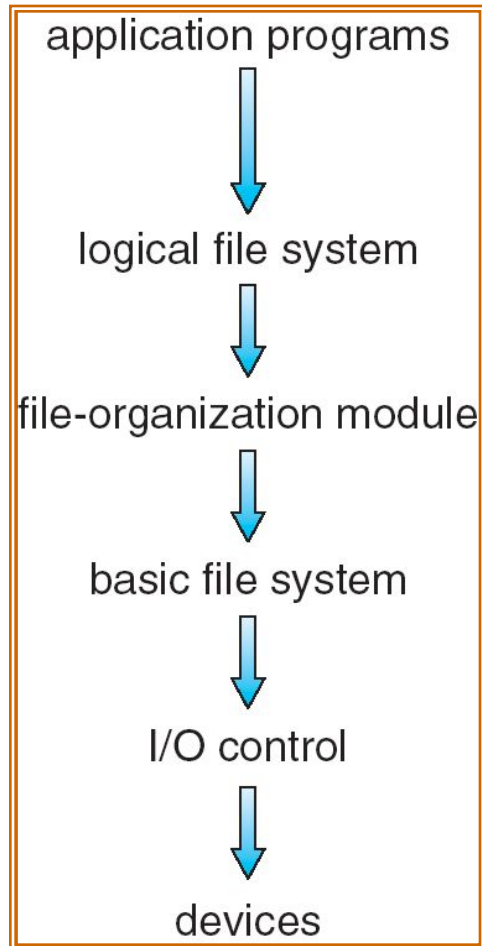


Estrutura do sistema de arquivos

- Estrutura de arquivo
 - Unidade de armazenamento lógico
 - Coleta de informações relacionadas
- Sistema de arquivos reside no armazenamento secundário (discos)
- Sistema de arquivos organizado em camadas
- **Bloco de controle de arquivo** – estrutura de armazenamento consistindo em informações sobre um arquivo



Sistema de arquivos em camadas



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

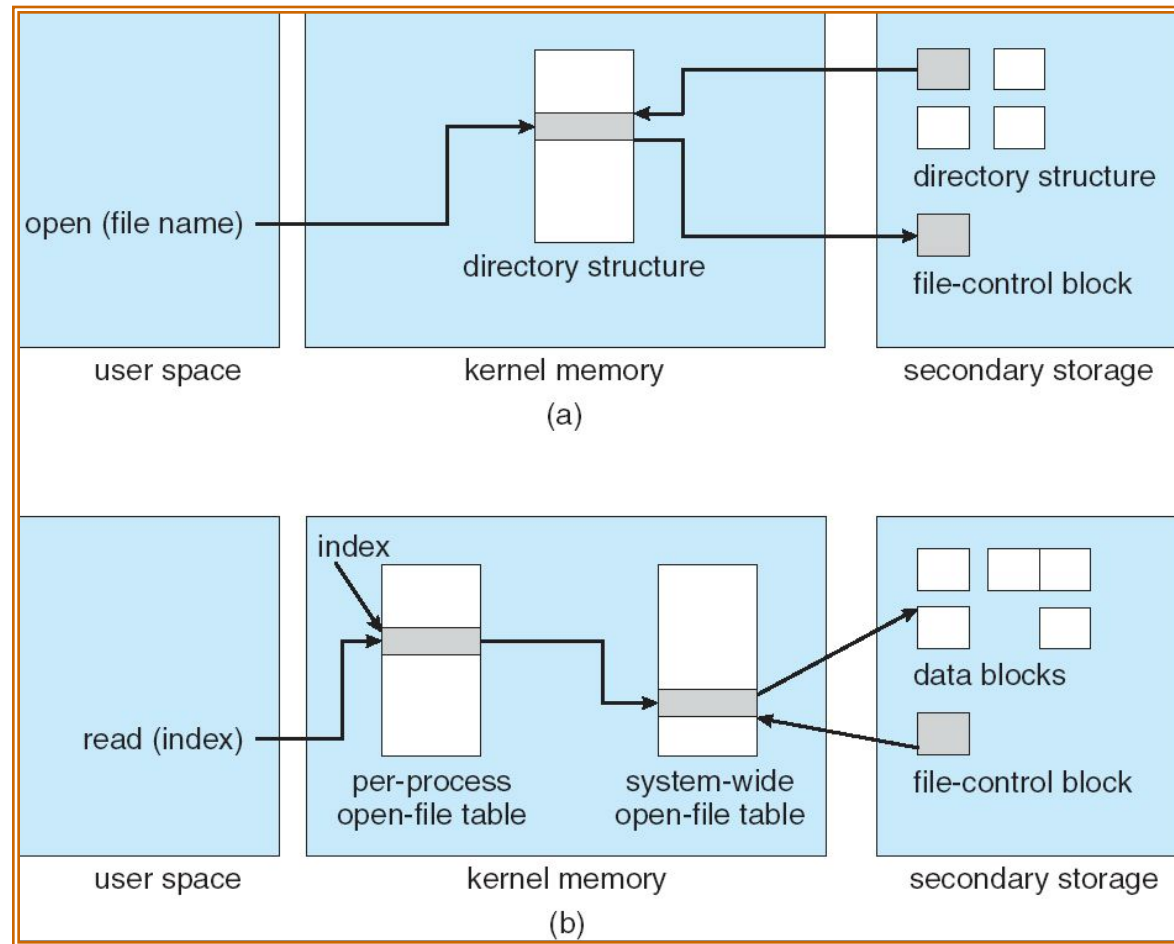


Estruturas de sistema de arquivos na memória

- A figura a seguir ilustra as estruturas necessárias do sistema de arquivos fornecidas pelos sistemas operacionais.
- Figura 12.3(a) refere-se à abertura de um arquivo.
- Figura 12.3(b) refere-se à leitura de um arquivo.



Estruturas de sistema de arquivos na memória

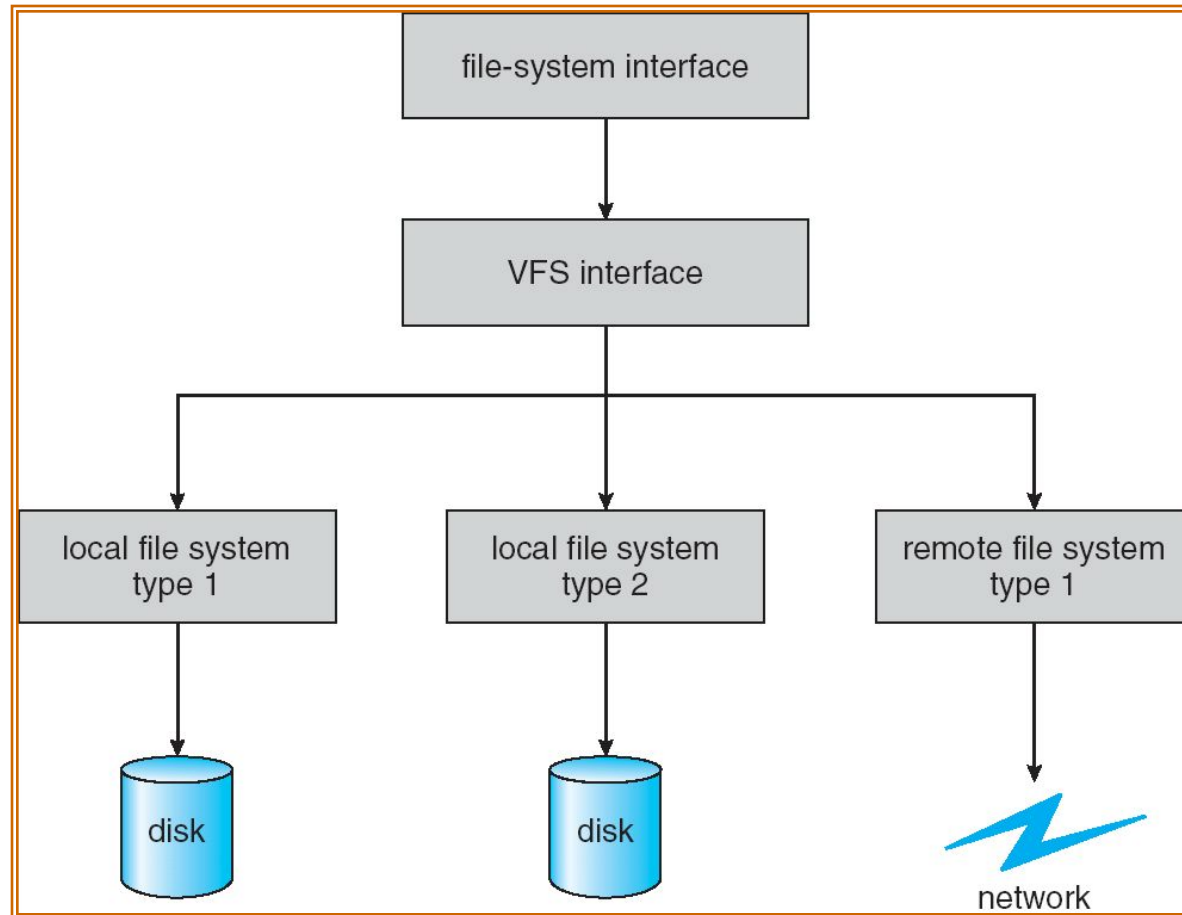


Sistemas de arquivo virtuais

- ❑ Virtual File Systems (VFS) oferecem um modo orientado a objeto para implementar sistemas de arquivos.
- ❑ VFS permite que a mesma interface de chamada do sistema (a API) seja usada para diferentes tipos de sistemas de arquivos.
- ❑ A API é para a interface VFS, e não qualquer tipo específico de sistema de arquivos.



Visão esquemática do Virtual File System



Implementação do diretório

- **Lista linear** dos nomes com ponteiro para os blocos de dados.
 - simples de programar
 - demorado para executar

- **Tabela de hash** – lista linear com estrutura de dados em hash.
 - diminui tempo de busca de diretório
 - **colisões** – situações onde dois nomes de arquivos referem-se ao mesmo local
 - tamanho fixo



Métodos de alocação

- Um método de alocação refere-se a como os blocos de disco são alocados para arquivos:
- **Alocação contígua**
- **Alocação vinculada**
- **Alocação indexada**

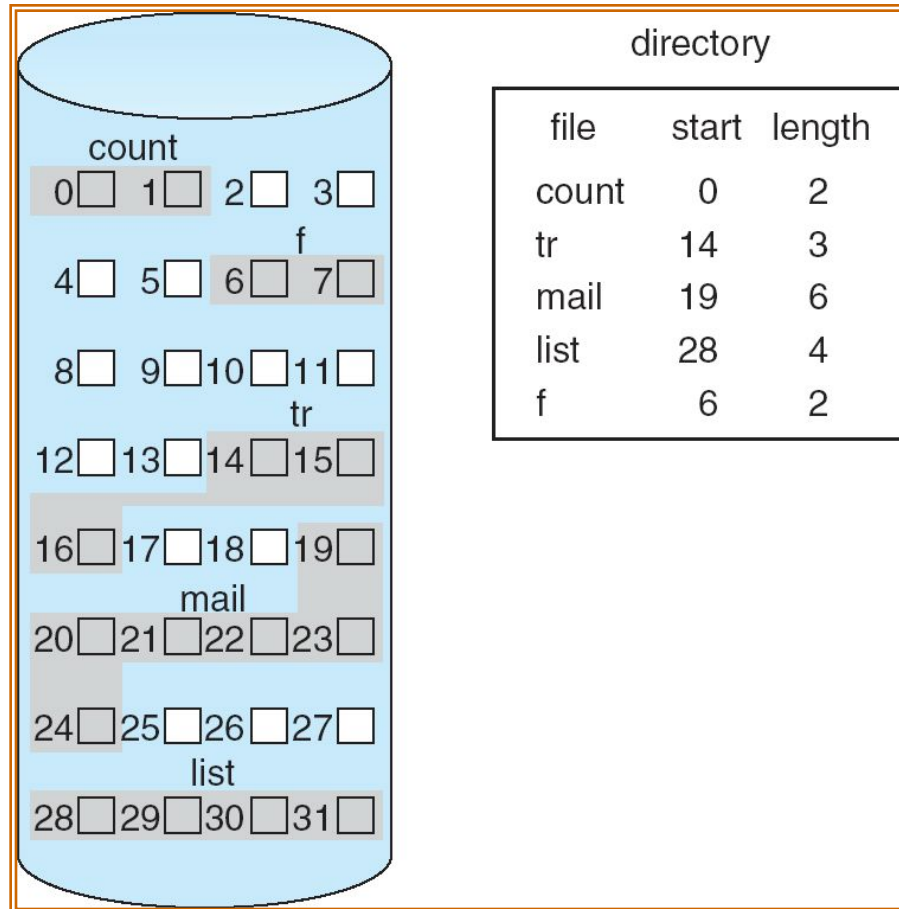


Alocação contígua

- ❑ Cada arquivo ocupa um conjunto de blocos contíguos no disco
- ❑ Simples – requer somente local inicial (# bloco) e tamanho (número de blocos)
- ❑ Acesso aleatório
- ❑ Desperdício de espaço (problema de alocação dinâmica de armazenamento)
- ❑ Arquivos não podem crescer



Alocação contígua de espaço em disco



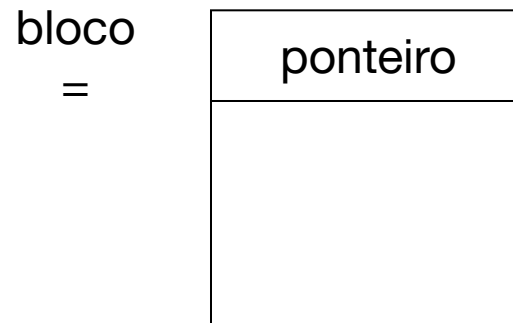
Sistemas baseados em extensão

- Muitos sistemas de arquivos mais novos (p.e., Veritas File System) usam um esquema de alocação contígua modificado
- Sistemas de arquivos baseados em extensão alocam blocos de disco em **extensões**
- Uma **extensão** é um bloco contíguo de discos
 - Extensões são alocadas para alocação de arquivo
 - Um arquivo consiste em uma ou mais extensões.



Alocação vinculada

- Cada arquivo é uma lista vinculada de blocos de disco: blocos podem estar espalhados por todo o disco



Alocação vinculada

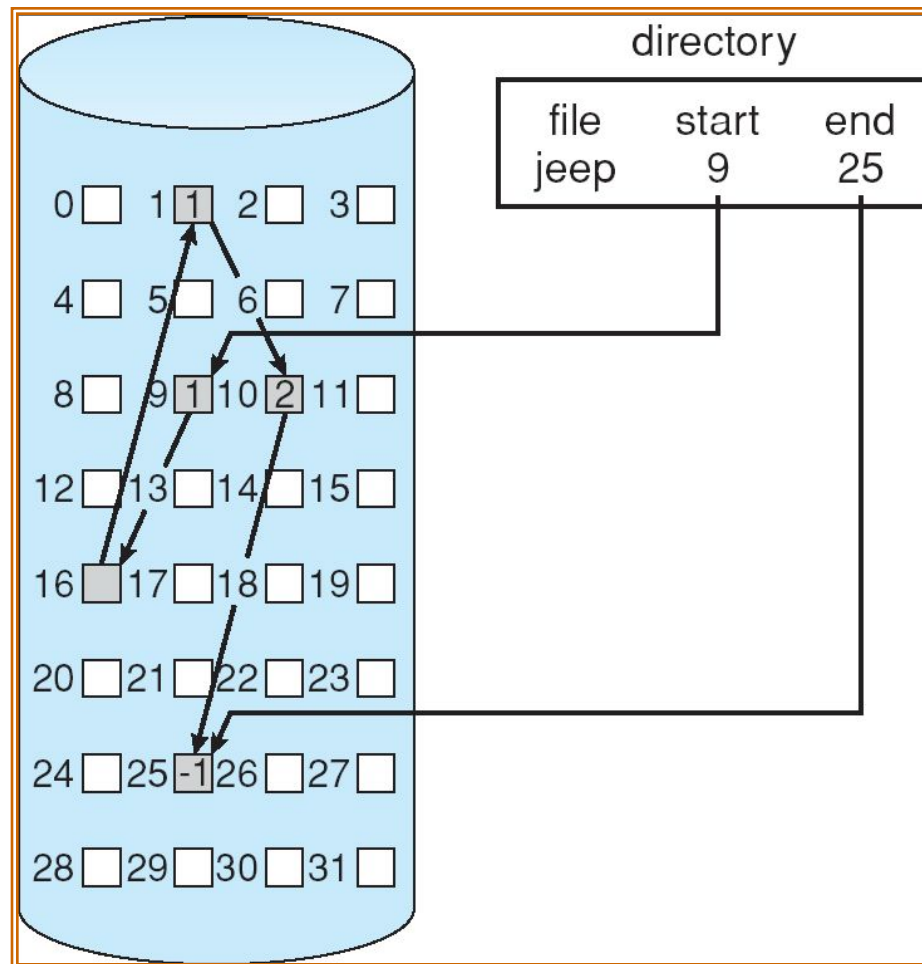
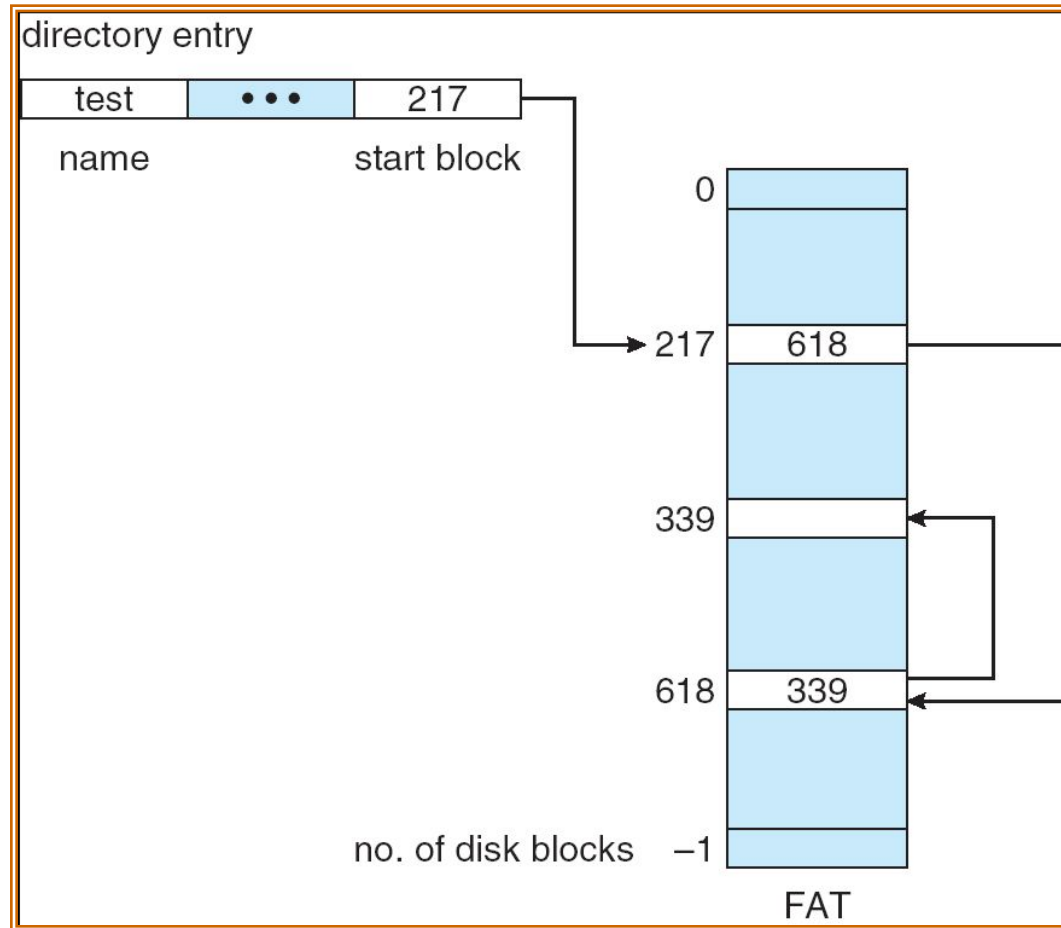


Tabela de alocação de arquivos



Alocação indexada

- Reúne todos os ponteiros no *bloco de índice*.
- Visão lógica.

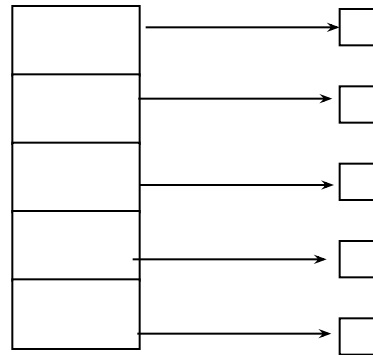
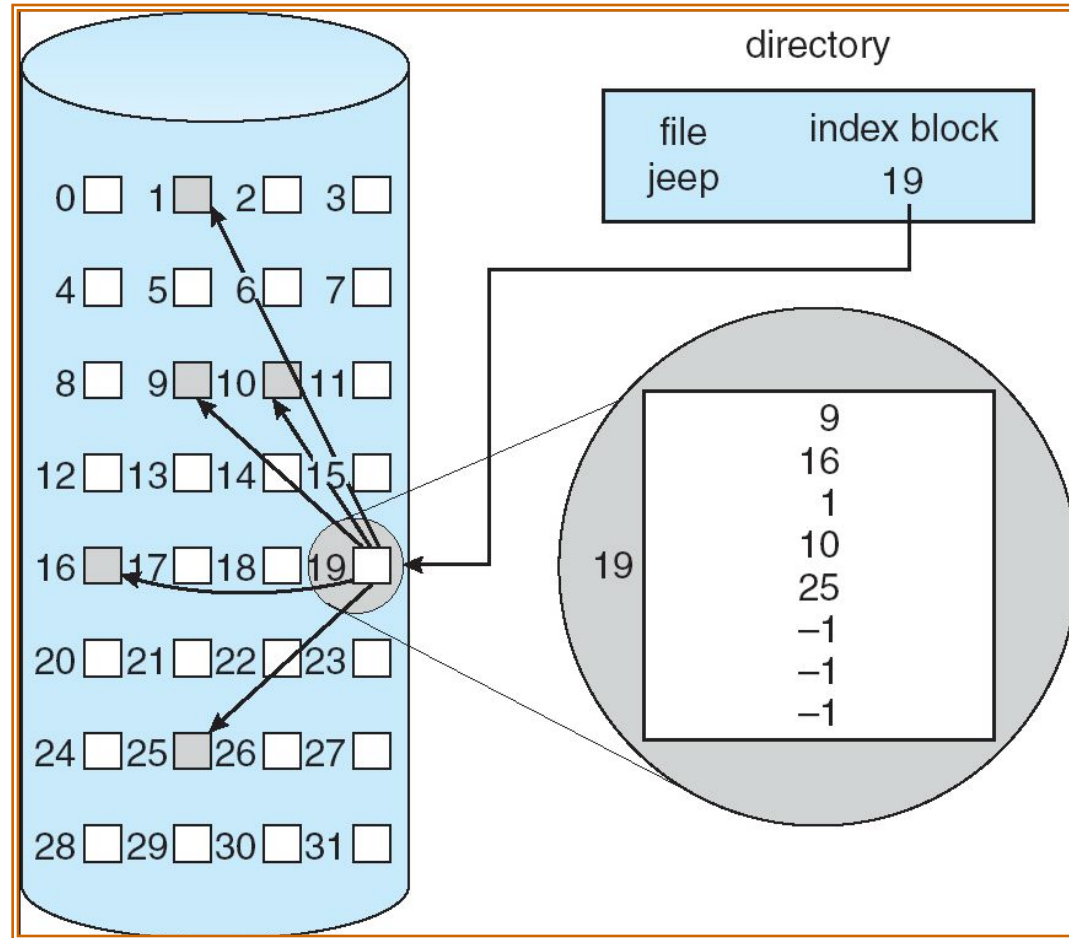


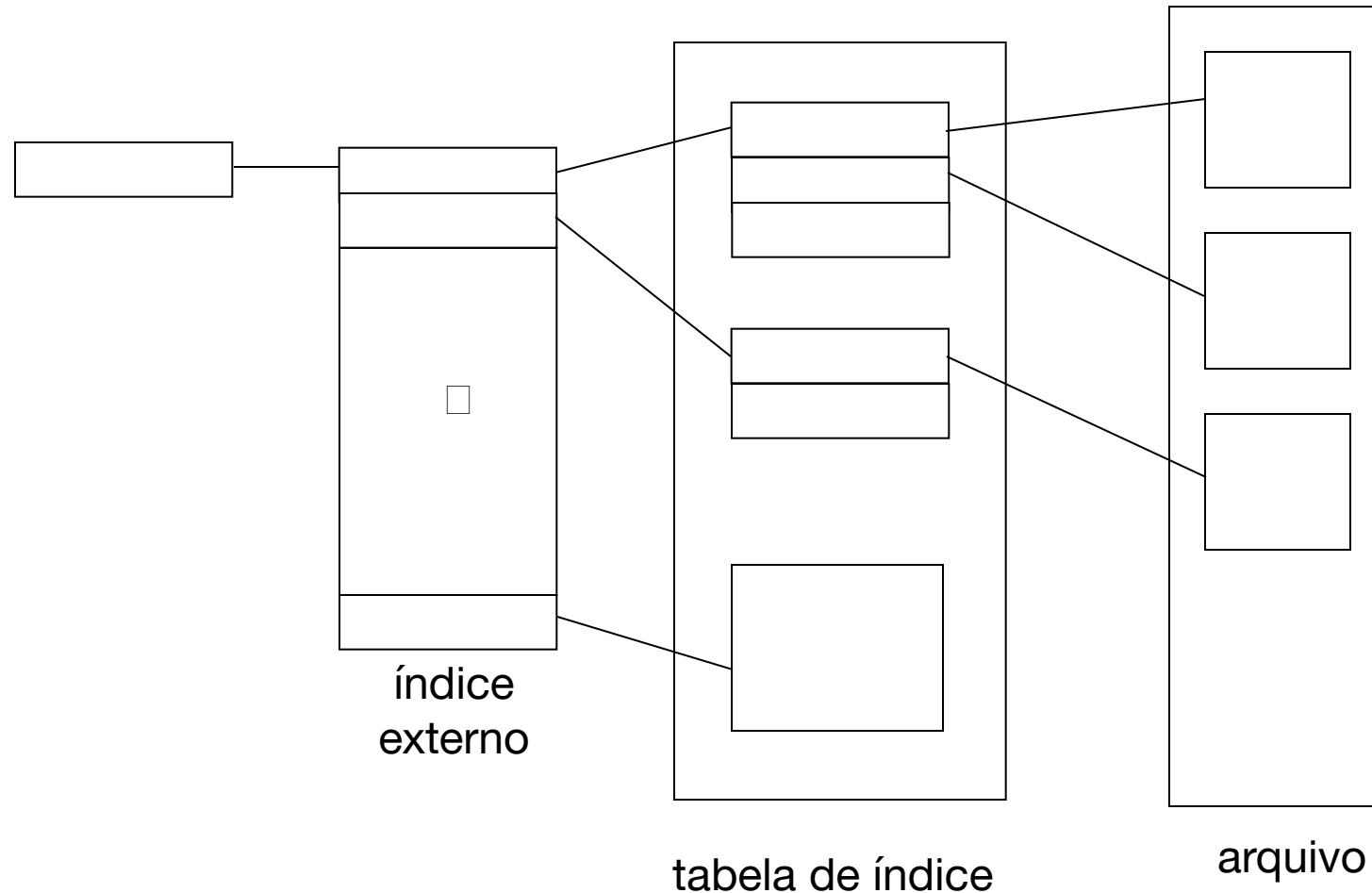
Tabela de índice



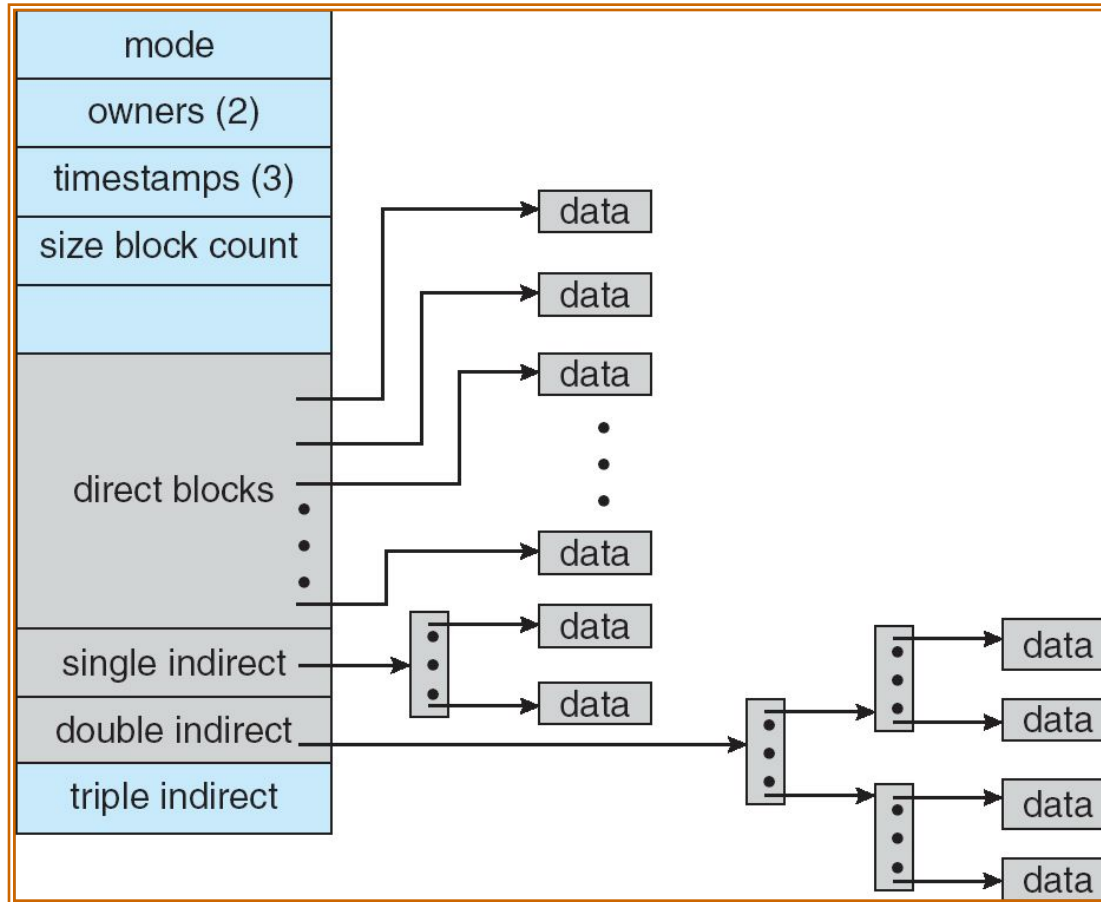
Exemplo de alocação indexada



Alocação indexada – mapeamento (cont.)

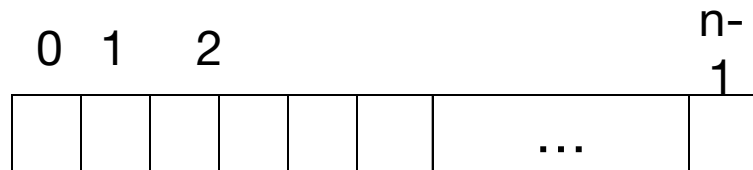


Esquema combinado: UNIX (4K bytes por bloco)



Gerenciamento de espaço livre

□ Vetor de bits (n blocos)



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

Cálculo do número de bloco

(número de bits por word) *
(número de words de valor 0) +
deslocamento do primeiro bit 1

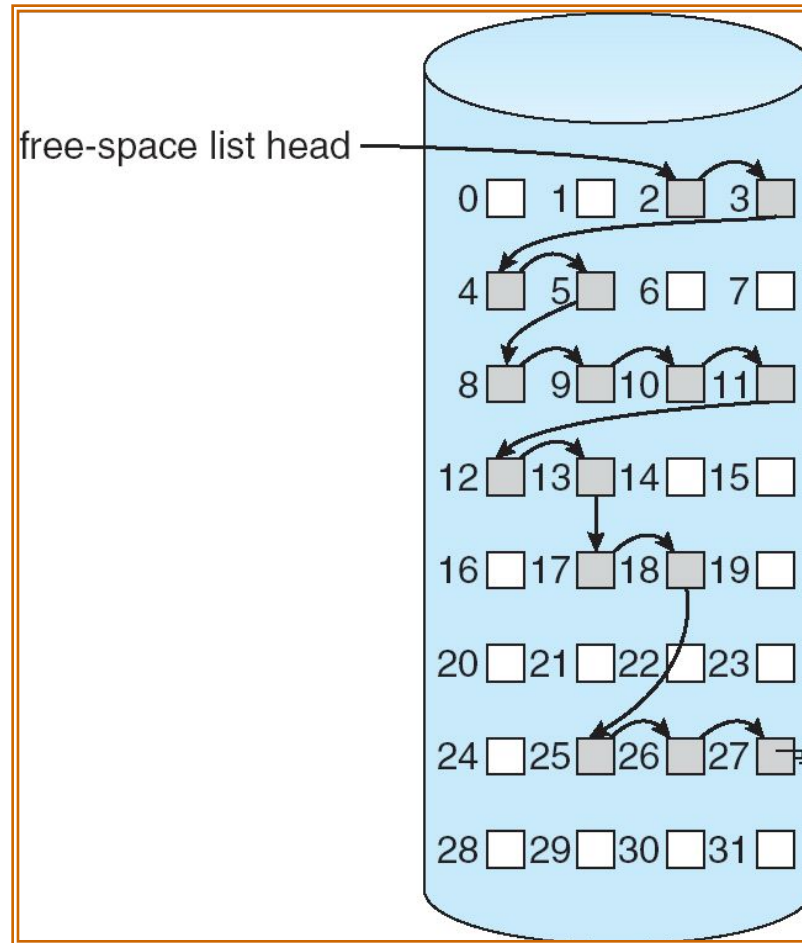


Gerenciamento de espaço livre (cont.)

- Precisa proteger:
 - Ponteiro para lista livre
 - Mapa de bits
 - Deve ser mantido em disco
 - Cópias na memória e disco podem diferir
 - Não pode permitir que $\text{bloco}[i]$ tenha situação onde $\text{bit}[i] = 1$ na memória e $\text{bit}[i] = 0$ em disco
 - Solução:
 - Defina $\text{bit}[i] = 1$ no disco
 - Aloque $\text{bloco}[i]$
 - Defina $\text{bit}[i] = 1$ na memória



Lista vinculada de espaço livre em disco



Eficiência e desempenho

- Eficiência dependente de:
 - alocação de disco e algoritmos de diretório
 - tipos de dados mantidos na entrada de diretório do arquivo

- Desempenho
 - cache de disco – seção separada da memória principal para blocos usados freqüentemente
 - free-behind e read-ahead – técnicas para otimizar acesso seqüencial
 - melhora desempenho do PC dedicando seção da memória como disco virtual, ou disco de RAM

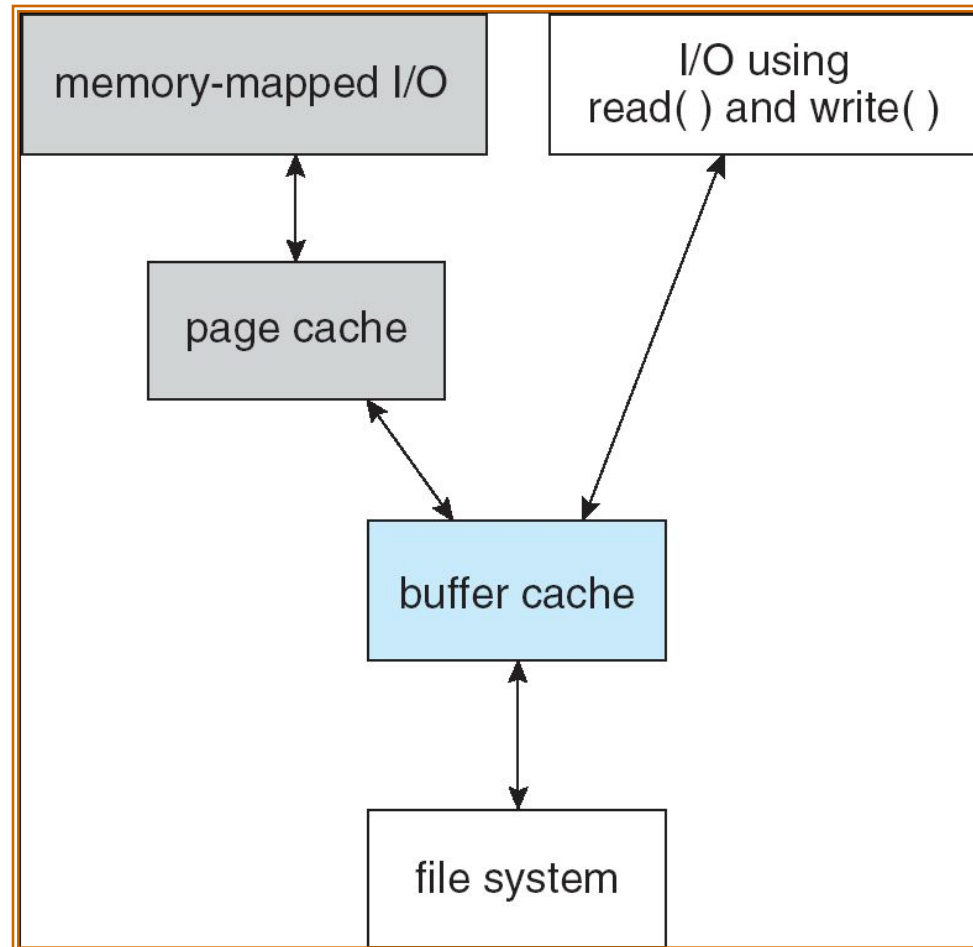


Cache de página

- Um **cache de página** guarda páginas ao invés de blocos de disco, usando técnicas de memória virtual
- E/S mapeada na memória usa um cache de página
- E/S de rotina pelo sistema de arquivos usa o cache de buffer
- Isso leva à figura a seguir



E/S sem um cache de buffer unificado

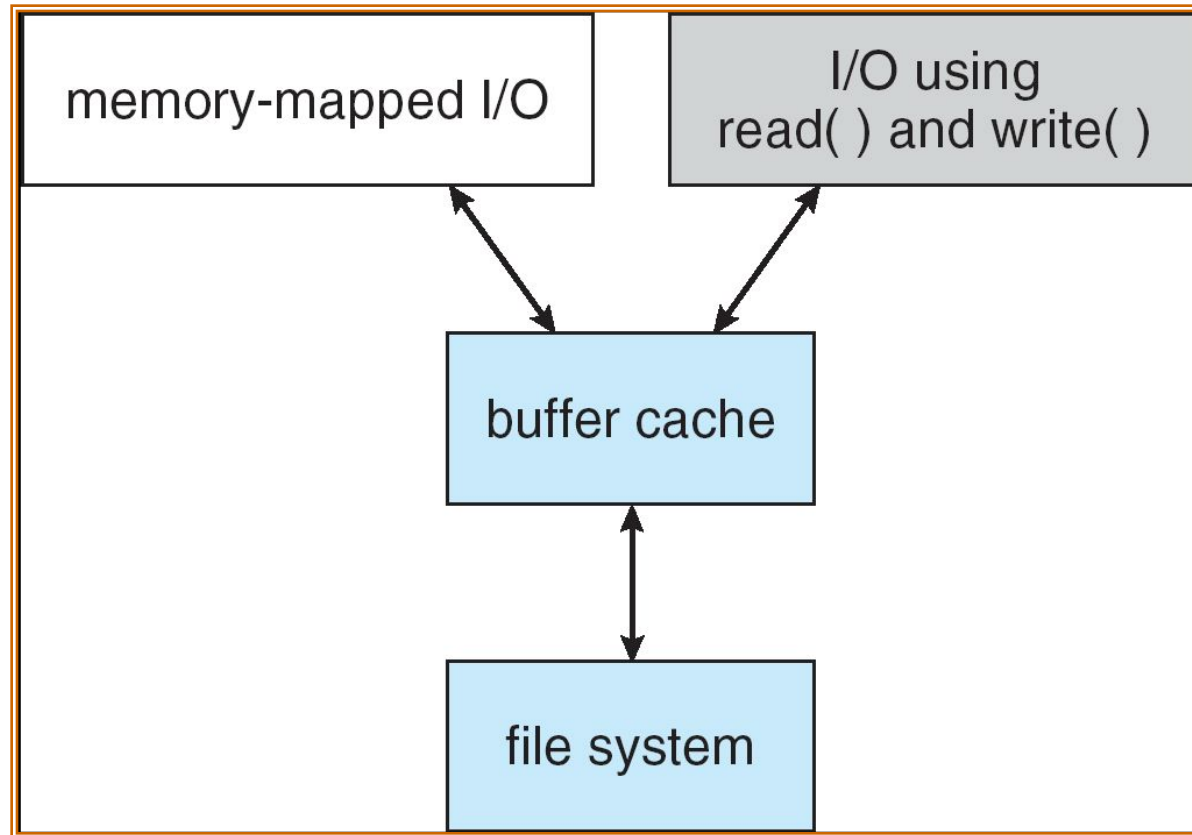


Cache de buffer unificado

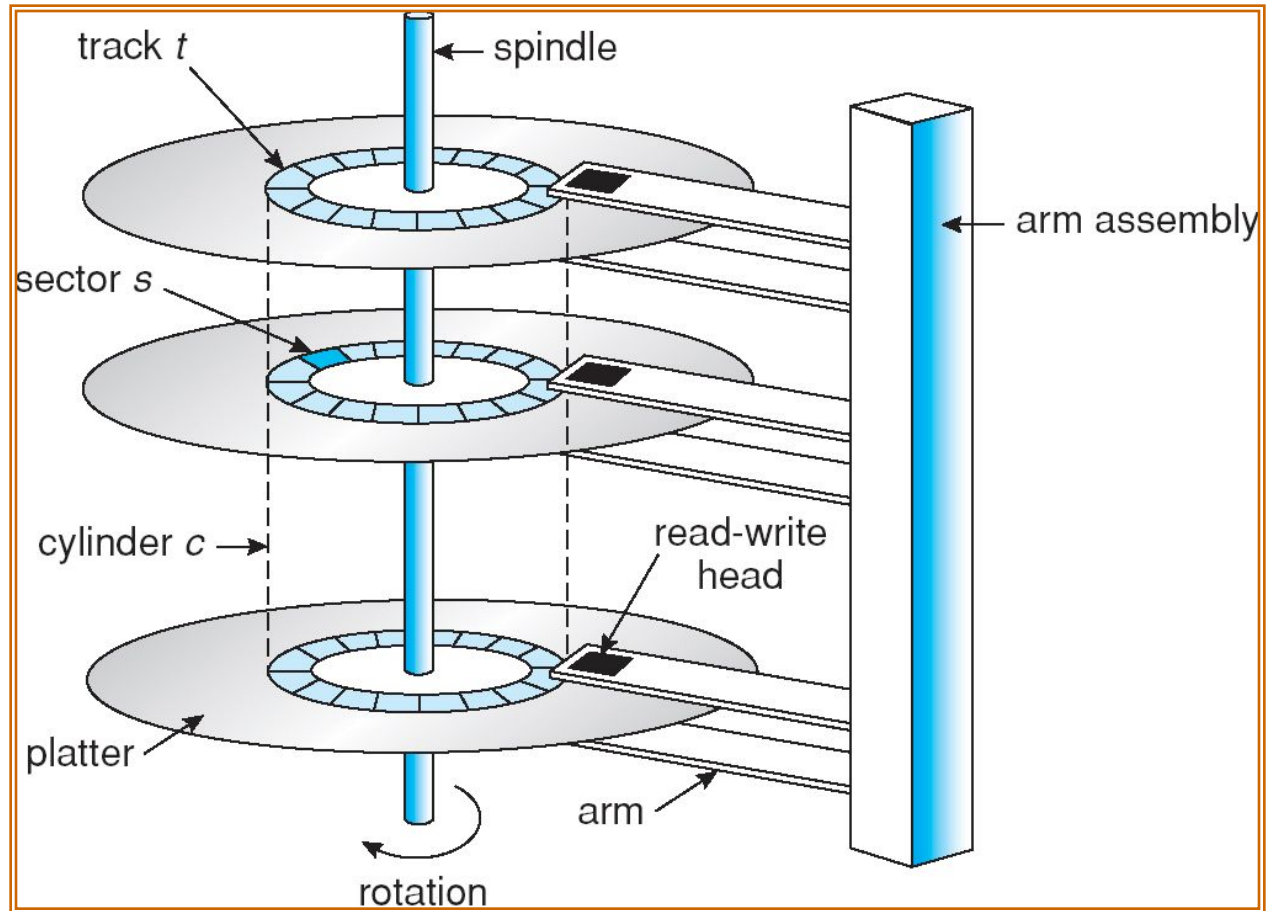
- Um cache de buffer unificado usa o mesmo cache de página para guardar páginas mapeadas na memória e E/S normal do sistema de arquivos



E/S usando um cache de buffer unificado



Mecanismo de disco com cabeça móvel



Visão geral da estrutura de armazenamento em massa (cont.)

□ Fita magnética

- Foi antigo meio de armazenamento secundário
- Relativamente permanente e mantém grandes quantidades de dados
- Tempo de acesso lento
- Acesso aleatório ~1000 vezes mais lento que o disco
- Usada principalmente para backup, armazenamento de dados usados com pouca frequência, meio de transferência entre sistemas
- mantida em um spool e avança e retrocede sob uma cabeça de leitura/escrita
- Quando dados estão sob a cabeça, possui taxas de transferência comparáveis ao disco
- 20-200GB de armazenamento típico
- Tecnologias comuns: 4mm, 8mm, 19mm, LTO-2 e SDLT



Estrutura de disco

- Unidades de disco são endereçadas como grandes arrays unidimensionais de *blocos lógicos*, onde o bloco lógico é a menor unidade de transferência.
- O array unidimensional de blocos lógicos é mapeado nos setores do disco seqüencialmente.
 - Setor 0 é o primeiro setor da primeira trilha no cilindro mais externo.
 - Mapeamento prossegue na ordem por essa trilha, depois o restante das trilhas nesse cilindro, e depois pelo restante dos cilindros de fora para dentro.



Escalonamento de disco

- O sistema operacional é responsável por usar o hardware de forma eficiente – para as unidades de disco, isso significa ter um tempo de acesso rápido e largura de banda de disco.
- Tempo de acesso tem dois componentes principais
 - *Tempo de busca* é o tempo para o disco mover as cabeças até o cilindro contendo o setor desejado.
 - *Latência de rotação* é o tempo adicional aguardando o disco girar o setor desejado até a cabeça do disco.
- Minimiza tempo de busca
- Tempo de busca \approx distância de busca
- Largura de banda de disco é o número total de bytes transferidos, dividido pelo tempo total entre a primeira solicitação de serviço e o término da última transferência.



Escalonamento de disco (cont.)

- Existem vários algoritmos para escalonar o atendimento das solicitações de E/S de disco.
- Ilustramos com uma fila de solicitação (0-199).

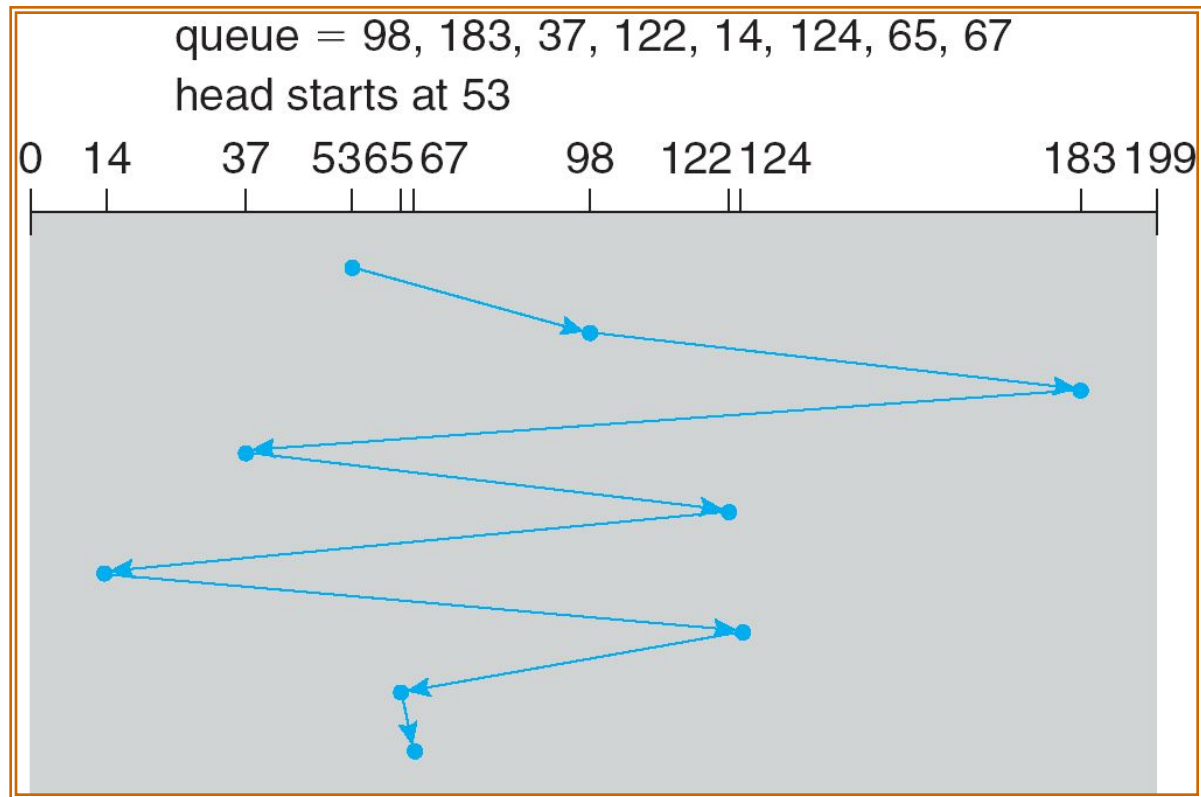
98, 183, 37, 122, 14, 124, 65, 67

Ponteiro da cabeça 53



FCFS

Ilustração mostra movimento total da cabeça de 640 cilindros.

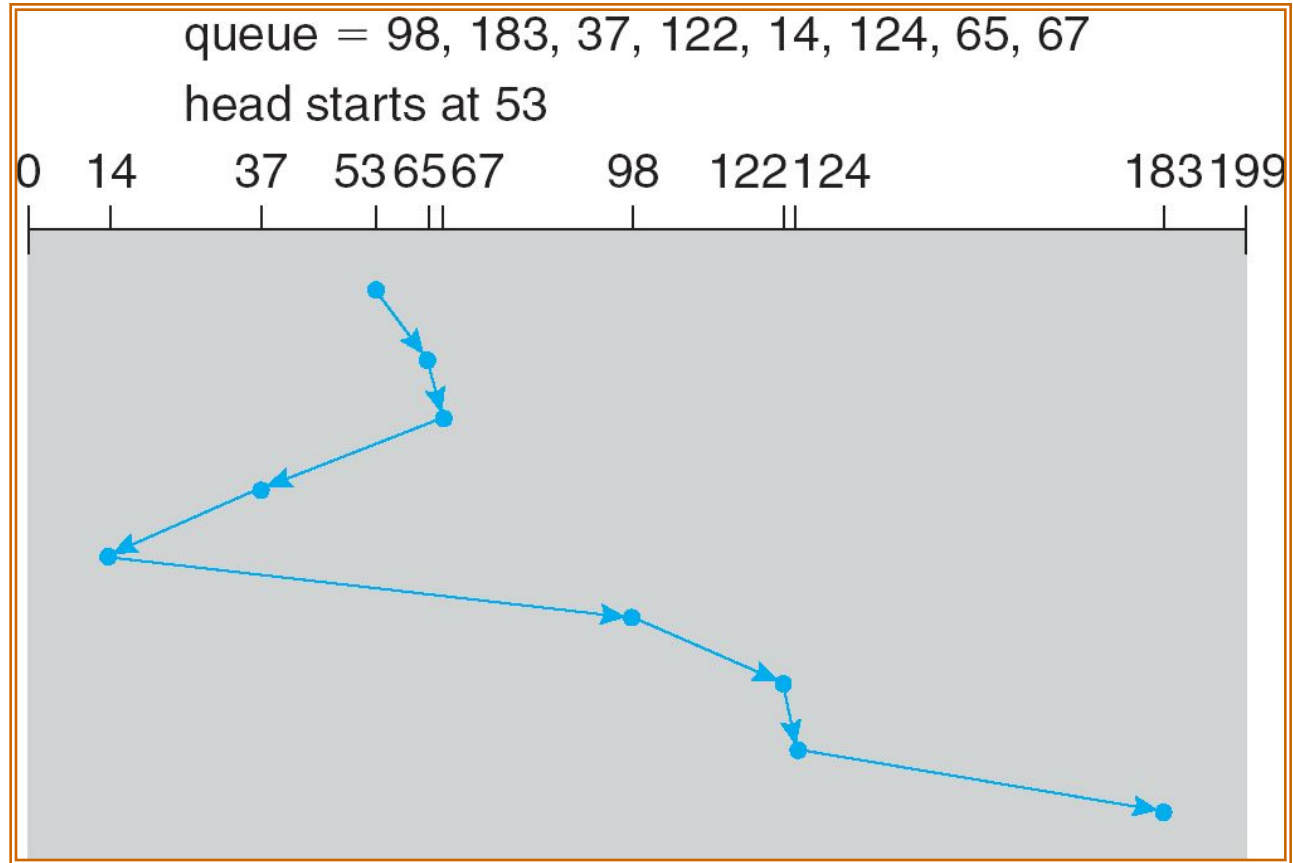


SSTF

- ❑ Seleciona a solicitação com o tempo de busca mínimo a partir da posição atual da cabeça.
- ❑ Escalonamento SSTF é uma forma de escalonamento SJF; pode causar starvation de algumas solicitações.
- ❑ Ilustração mostra movimento total da cabeça de 236 cilindros.



SSTF (cont.)

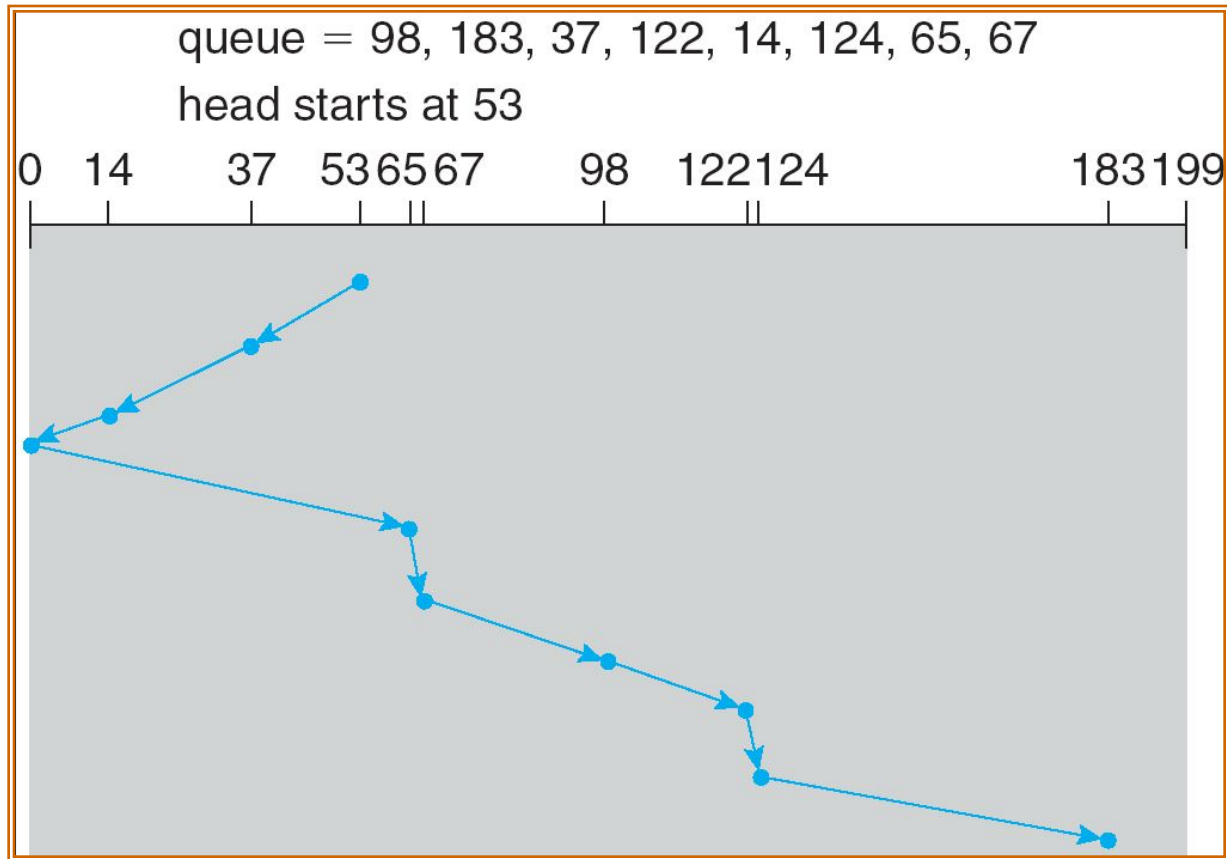


SCAN

- ❑ O braço do disco começa em uma extremidade do disco, e se move para a outra extremidade, atendendo solicitações até que chegue à outra extremidade, onde o movimento da cabeça é revertido e o atendimento continua.
- ❑ Às vezes chamado de *algoritmo do elevador*.
- ❑ Ilustração mostra movimento total da cabeça de 208 cilindros.



SCAN (cont.)

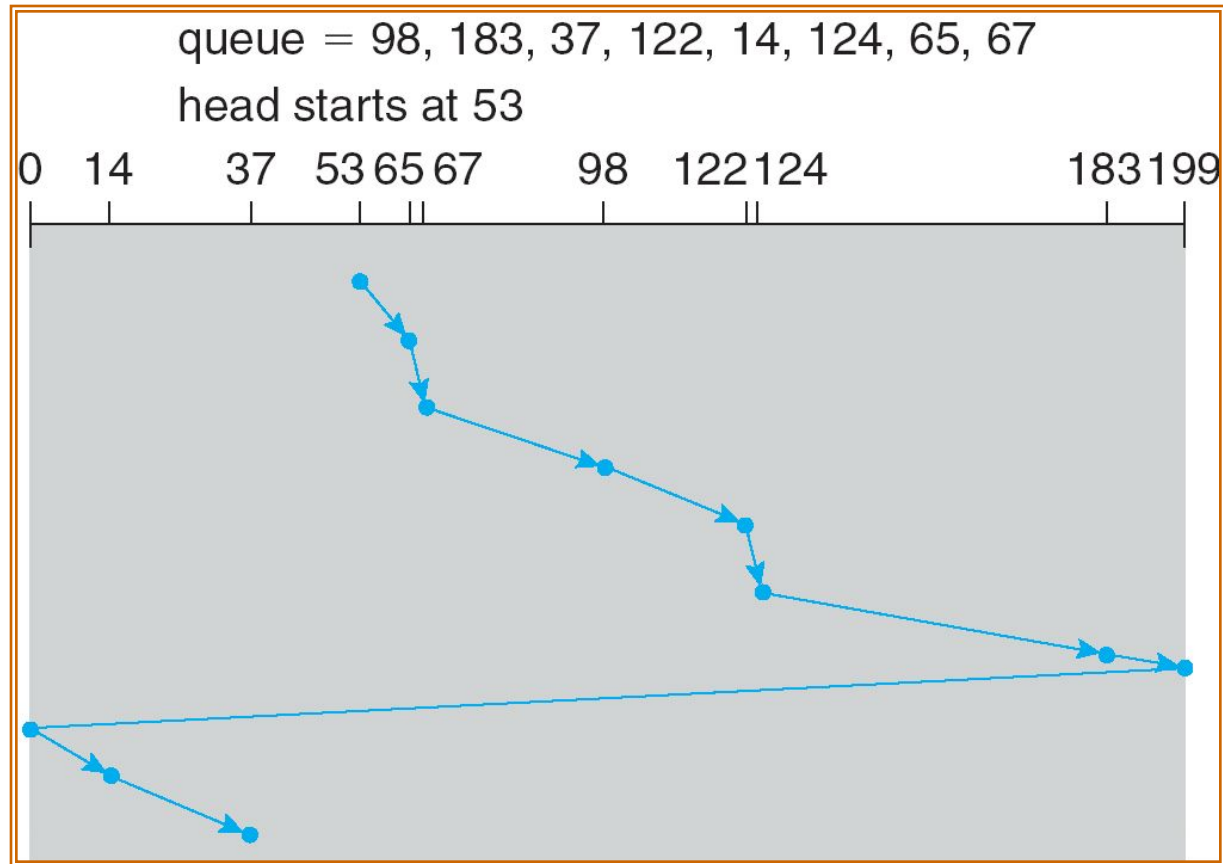


C-SCAN

- ❑ Fornece um tempo de espera mais uniforme que SCAN.
- ❑ A cabeça se move de uma extremidade do disco para a outra, atendendo solicitações enquanto prossegue. Quando atinge o outro extremo, imediatamente retorna ao início do disco, sem atender quaisquer solicitações no retorno.
- ❑ Trata os cilindros como uma lista circular que contorna o último cilindro e volta ao primeiro.



C-SCAN (cont.)

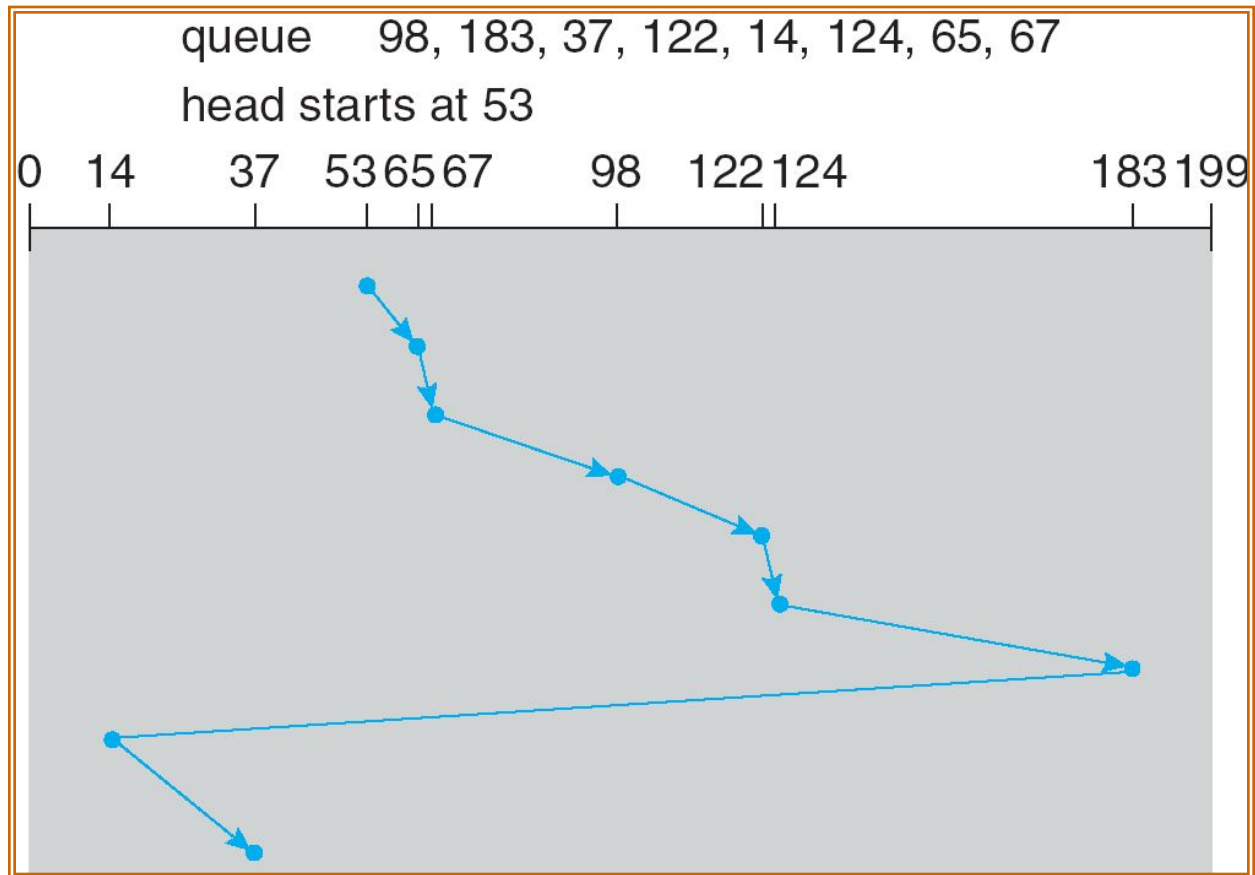


C-LOOK

- Versão de C-SCAN
- O braço só vai até a distância da última solicitação em cada direção, depois reverte a direção imediatamente, sem primeiro ir até o final do disco.



C-LOOK (cont.)



Selecionando um algoritmo de escalonamento de disco

- ❑ SSTF é comum e tem um apelo natural
- ❑ SCAN e C-SCAN funcionam melhor para sistemas que têm cargas pesadas sobre o disco
- ❑ O desempenho depende do número e tipo de solicitações.
- ❑ Requisições para serviço de disco podem ser influenciadas pelo método de alocação de arquivo.
- ❑ O algoritmo de escalonamento de disco deve ser escrito como um módulo separado do sistema operacional, permitindo que seja substituído por um algoritmo diferente, se necessário.
- ❑ SSTF ou LOOK é uma escolha razoável para o algoritmo padrão.



Final do Capítulo 11

