Sistemas Operacionais

Gerenciamento de Arquivos

Prof. Arliones Hoeller

arliones.hoeller@ifsc.edu.br

Junho de 2014

baseado no material do Prof. Fröhlich em
http://www.lisha.ufsc.br/~guto

Gerenciamento de Arquivos

- Motivação
  - Interface comum para manipulação transparente de dados no armazenamento secundário
- Sistema de Arquivos
  - Abstrair um dispositivo de armazenamento como
    - Um conjunto de arquivos (dados) e
    - Uma estrutura de diretórios (informação de controle)
  - Interação com dispositivos de armazenamento através de serviços exportados pelos *device drivers*
    - Dispositivos: array linear de blocos
  - Uma das estruturas mais visíveis de um SO
  - Exemplos:
    - FAT, UFS, EXT2, EXT3, EXT4, NTFS, ISO9660, etc

Arquivos

- Arquivo
  - Seqüência de bits, bytes, linhas ou registros não-voláteis e com um nome
- Arquivo “tipado”
  - Estrutura interna definida pelo SO
    - Arquivos executáveis, arquivos gráficos, arquivos de texto, etc
  - Número limitado de tipos conhecidos
- Arquivo não “tipado”
  - Fluxos de bytes cujo sentido é definido pelo usuário
  - Ilimitado e flexível

Atributos dos arquivos

- Nome
  - String de caracteres identificando o arquivo para os usuários
- Tipo (apenas para arquivos “tipados”)
  - Informação interna de tipo (depende do SO)
- Localização no dispositivo
- Tamanho
- Propriedade
- Controle de acesso
  - Quem pode acessar o arquivo e para quais operações
- Histórico de acesso
  - Datas, horários, usuários, contadores, etc

Operações em arquivos

- Criação
  - Aloca espaço no sistema de arquivos
  - Cria uma entrada no diretório de destino
- Remoção
  - Busca o diretório pelo arquivo nomeado
  - Libera espaço no sistema de arquivos
  - Remove entrada do diretório correspondente
- Escrita/Leitura
  - Busca o diretório pelo arquivo nomeado
  - Determina a localização no sistema de arquivos onde se deve operar
  - Escreve ou lê dado
  - Atualiza o apontador do arquivo no sistema de arquivos

Operações em arquivos

- Posicionamento
  - Busca o diretório pelo arquivo nomeado
  - Move o apontador do arquivo para outra posição
- Abertura/fechamento
  - Como toda operação em arquivos requer uma busca de diretório, é comum implementar operações para trazer informações significativas de arquivos do dispositivo secundário para uma “tabela de arquivos abertos”, diminuindo o custo de busca.
- Mapeamento em memória
  - Associar uma parte do espaço de endereçamento do processo a uma sessão de um arquivo, fazendo com que operações de leitura e escrita sejam realizadas na memória, mais rapidamente, antes de serem sincronizadas com o armazenamento secundário.

Métodos de acesso a arquivos

- Sequencial
  - Acesso ordenado - um registro após o outro (modelo de fita)
  - Ponteiro de arquivo incrementado após cada operação
  - Rebobinamento move o ponteiro de arquivo para o início
- Direto
  - Ponteiro de arquivo pode ser movido arbitrariamente (modelo de disco)
- Indexado
  - Baseado no método de acesso direto
  - Índices associados a uma chave de busca de registros

Semântica para Consistência de Arquivos

- Unix
  - Escritas a um arquivo aberto por um usuário são imediatamente visíveis a todos outros usuários que estão com o mesmo arquivo aberto
  - Mecanismo bloqueante para sincronizar acesso
- Sessão (Andrew)
  - Toda nova abertura retorna uma cópia de um arquivo
  - Não há concorrência no acesso ao arquivo (cópias privadas)
  - Atualização no fechamento (visível a novas sessões)
- Arquivo compartilhado imutável (Bullet)
  - Arquivos compartilhados são somente-leitura

Controle de Acesso a Arquivos

- Motivação
  - Sistemas de arquivos multiusuários necessitam que controle de acesso seja realizado pelo SO
- Tipos de acesso
  - Read, write, execute, append, delete
- Critérios de accesso
  - Sabendo o nome dos arquivos
  - Sabendo uma senha associada a um arquivo ou diretório
    - Impraticável para aplicações interativas
  - Ser incluído em uma lista de acesso a arquivo ou diretório
    - Associa usuários e permissões de acesso
    - Difícil de manter
    - Estruturas de dados de tamanho variável

Controle de Acesso a Arquivos

- Abordagem do Unix
  - Lista de acesso simplificada
    - Permissões para ler, escrever (e excluir), e executar
    - Permissões para proprietário, grupo do proprietário, e outros
  - Exemplo

    `drwxr-xr-x dir1` proprietário escreve, todos lêm e executam

    `-rwxrwxrwx fil1` todos podem fazer tudo

    `-r-x------ fil2` proprietário pode executar

    `-r--r--r-- fil3` todos podem ler

Diretório

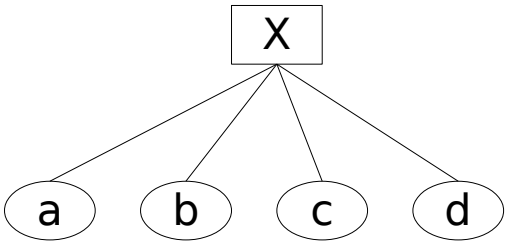
- Diretório
  - Coleção de informações sobre arquivos
  - Tablea de tradução (nome => info de controle)
- Device directory
  - Características físicas dos arquivos
    - Tamanho, localização no disco, proprietário, etc
- File directory
  - Tabela de conteúdo de um Volume
  - Associa nomes de arquivos a entradas do *device directory*

Operações em Diretórios

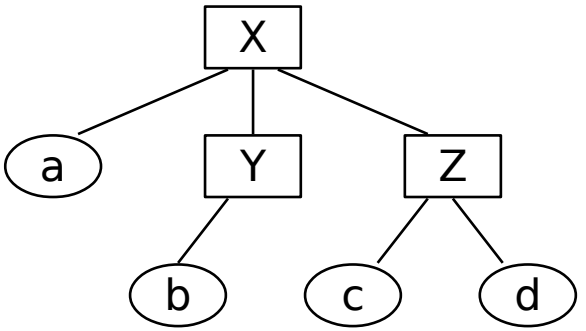
- Criar/deletar diretórios
- Adicionar/remover entradas de diretórios
  - Para criação/deleção de arquivos
- Manipular entradas de diretórios
  - Para renomear ou atualizar info de controle de arquivos
- Buscar por um arquivo ou padrão
- Listar
- Varrer (*Traversing*)
  - Para operações em todo o sistema como buscas ou backup

Organização de Diretórios

- Flat
  - Diretório único com todos arquivos

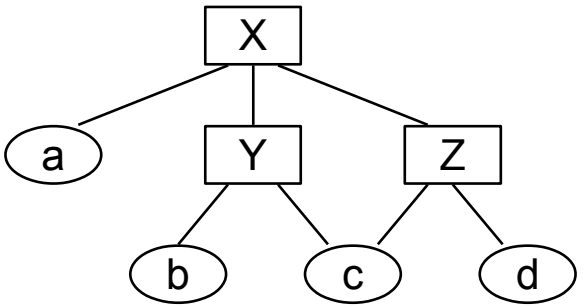


- Árvore
  - SO diferencia nós (diretórios) de folhas (arquivos)
  - Nó Raiz ('/')
  - Caminhos
    - Absolutos (começando da raiz)
    - Relativos (a partir do diretório atual)



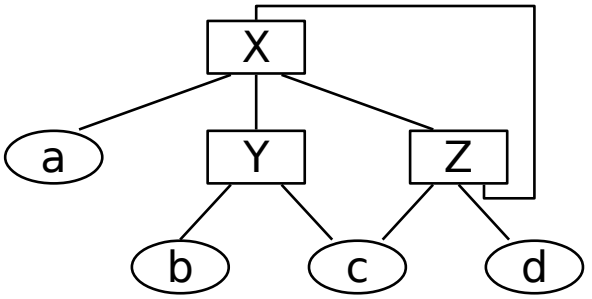
Organização de Diretórios

- Grafo Acíclico
  - Hard link
    - Contador de referências
    - Arquivo de link são indistinguíveis
    - Não aplicável a diretórios
  - Symbolic link
    - Baseado em um caminho (*pathname*)
    - Arquivo e link podem ser distinguidos
    - Podem quebrar
  - Problemas com aliasing de nomes
    - Deleção
    - Varredura



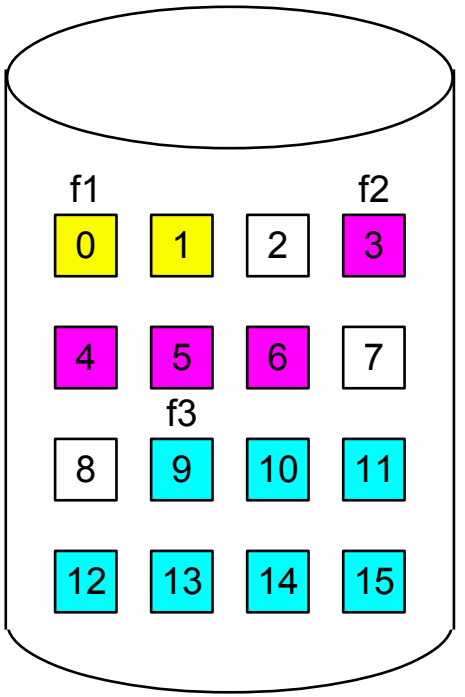
Organização de Diretórios

- Grafo comum
  - Ciclos são permitidos no diretório
    - *Hard links* para outros diretórios
  - Algoritmo de busca precisa detectar ciclos
    - Evitar laçõs infinitos
  - Coleta de lixo (auto-referência)



Métodos de Alocação de Blocos

- Alocação contígua
  - Diretório = (nome,início, comprimento)
  - Acesso sequencial ótimo
  - Acesso direto
  - Tamanho de arquivo definido na criação
    - Um conjunto suficiente de blocos contíguos precisa ser encontrado (first/best/worst-fit)
  - Fragmentação externa
    - Coleta de lixo

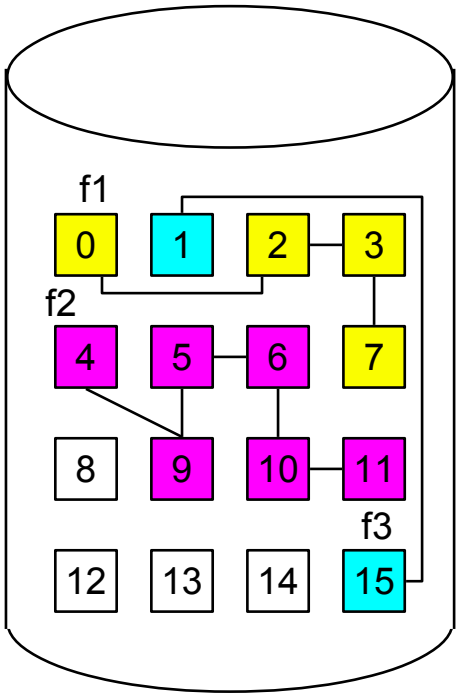


Directory

file	start	length
f1	0	2
f2	3	4
f3	9	7

Métodos de Alocação de Blocos

- Alocação encadeada
  - Diretório = (nome, início, fim)
  - Arquivos são listas encadeadas de blocos
    - Qualquer bloco pode ser ligado a qualquer arquivo
    - Sem fragmentação externa
  - Não há acesso direto
  - Pouco confiável

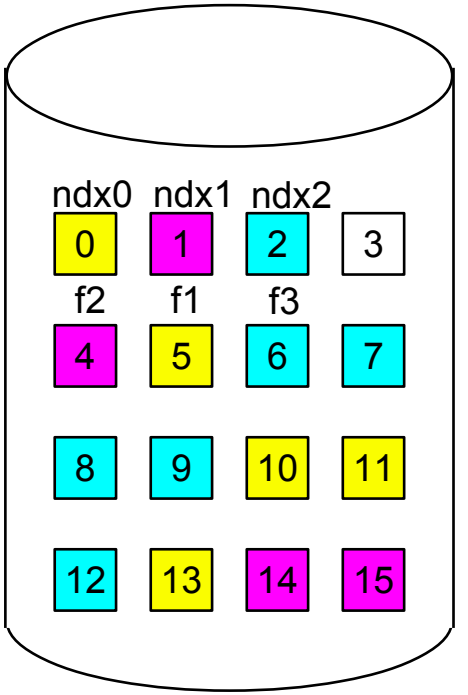


Directory

file	start	end
f1	0	7
f2	4	11
f3	15	1

Métodos de Alocação de Blocos

- Alocação indexada
  - Diretório = (nome, índice) + índice
  - Similar a paginação
  - Acesso direto sem fragmentação externa
  - Arquivos grandes
    - Índices encadeados
    - Índices multinível



Directory

file	index
f1	0
f2	1
f3	2

index0
5
10
13
11
-

index1
4
14
15
-
-

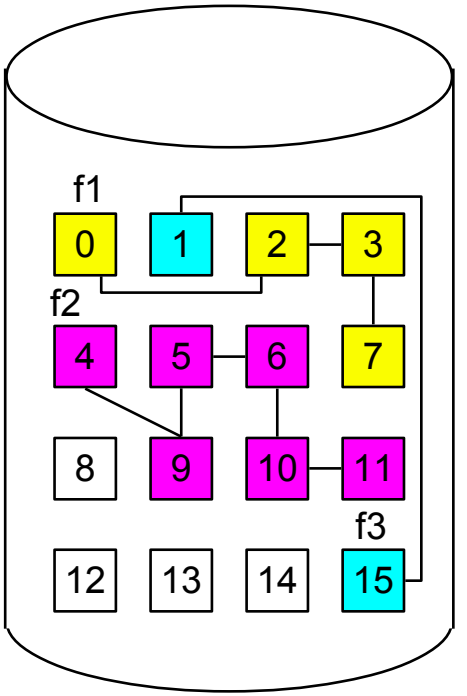
index2
6
7
8
9
12

Gerenciamento de Blocos Livres

- Bit map
  - Cada bloco é representado por um bit (livre/em uso)
  - Fácil localizar sequências de bits no mesmo estado
    - Suporta alocação contígua
    - Otimiza acesso sequencial
  - Precisa ser mantido em memória para ser eficiente
- Lista Encadeada
  - Blocos livres são ligados em uma lista
  - Alocação e liberação implicam em I/O (acesso ao disco)
- Agrupamento
  - Primeiro bloco livre agruma índices de blocos livres e contém um ponteiro para o (eventual) próximo bloco índices de blocos livres
  - Blocos livres contíguos podem ser representados por ponteiros para o primeiro bloco e um contador de blocos

Estudo de Caso: Tabela de Alocação de Arquivos do MS-DOS (FAT)

- MS-DOS FAT
  - Diretório = (nome, início, fim) + FAT
  - Tabela com uma entrada por bloco é mantida separadamente (FAT)
  - Valores especiais para blocos livres e finais de arquivos
  - Permite acesso direto



Directory

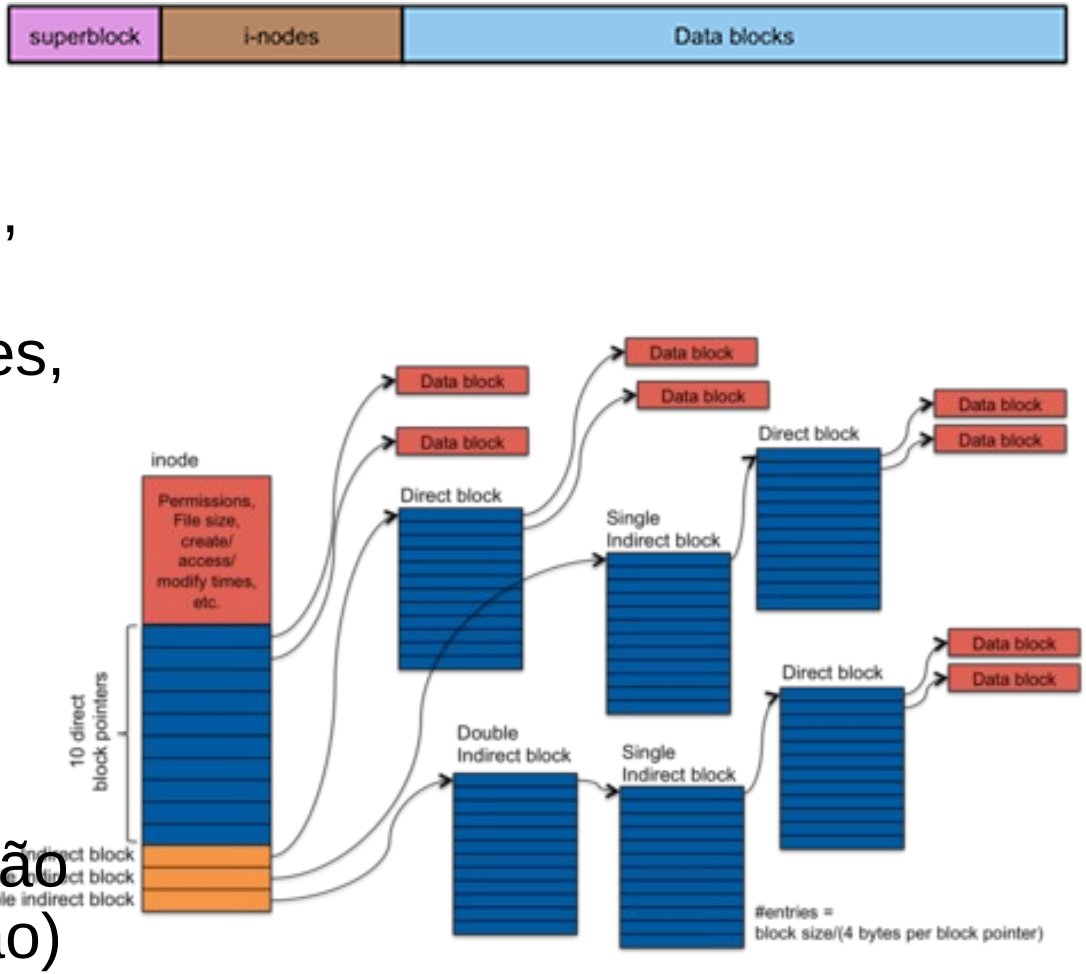
file	start	end
f1	0	7
f2	4	11
f3	15	1

FAT

2	eof	3	7
9	6	10	eof
free	5	11	eof
free	free	free	1

Estudo de Caso: Unix File System (UFS)

- Superblock
  - Estrutura com info de controle do sistema de arquivos
  - Guarda infos como nome, tipo, tamanho, estado, device usado, blocos livres, etc
- I-node
  - File Control Block (FCB)
  - Identificador único de um arquivo
  - Contém informações SOBRE o arquivo, mas não os dados (metainformação)

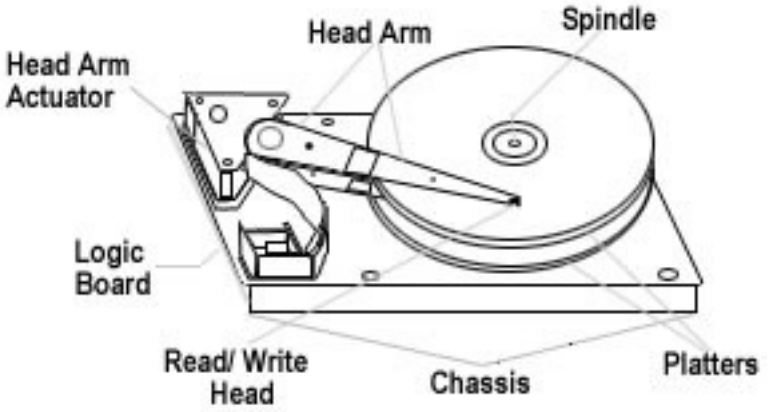


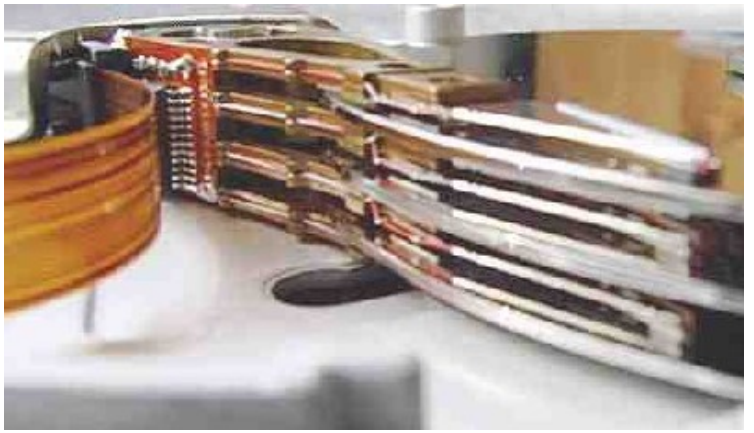
Gerenciamento de Memória Secundária

- Motivação
  - Memória principal é pequena, cara e volátil
  - Memória secundária é grande, barata e persistente
    - Tipicamente, discos
- Base para componentes importantes do SO
  - Swapping
  - Memória Virtual
  - Sistema de Arquivos

Dispositovos de Disco (Disk Drives)

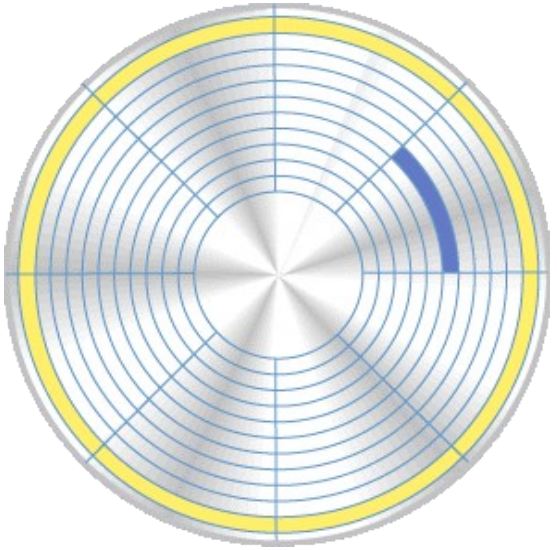
- Estrutura física
  - Media
    - rígido ou flexível, fixo ou removível
  - Driver (mecânico)
    - Rotação de discos e movimento de cabeçotes
  - Controlador (eletrônico)
    - Operação e interface com sistema
- Tecnologias
  - Magnético
  - Ótico
  - Optomagnético





Discos

- Estrutura física
  - Trilhas co-cêntricas divididas em setores
  - Espaços entre setores
  - Setores tipicamente formatados para terem 512 bytes cada
- Estrutura lógica
  - Array unidimensional de blocos lineares
  - Block = 1 ou *n* setores
- Partição
  - Conjunto de blocos contíguos em disco considerados pelo SO como um disco lógico autônomo



- Volume
  - Disco, discos, ou partições que contém UM sistema de arquivo
  - Um volume pode utilizar mais de uma partição ou disco

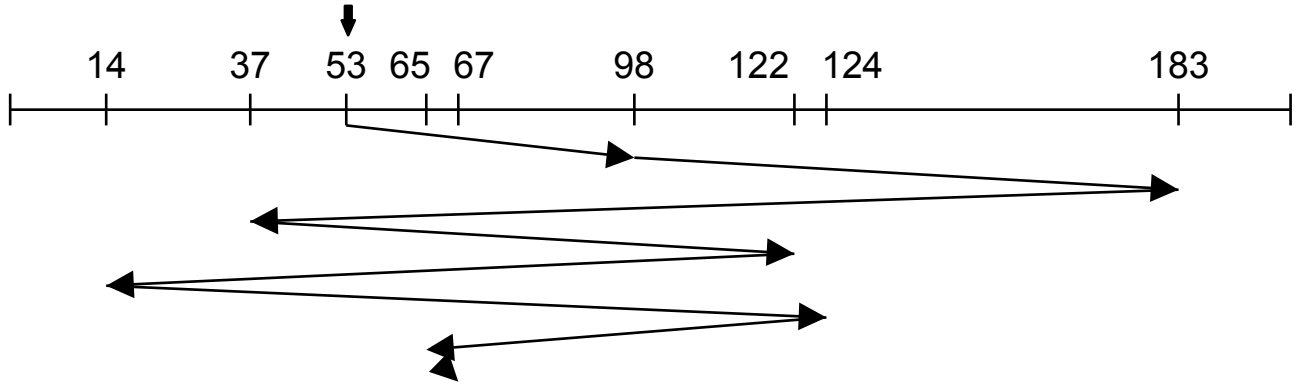
Escalonamento de Disco

- Parâmetros de tempo de acesso a disco
  - Seek: tempo para mover cabeçote até uma trilha
  - Latência: atraso até que um setor passe sob o cabeçote
  - Transferência: tempo para transferir dados do controlador de disco para a memória principal
- Requisições de acesso a disco
  - Endereço no disco + endereço na memória + tamanho
  - Fila de requisições
    - Ordenar requisições, reunindo aqueles à mesma trilha
    - Ordenar requisições para entre trilhas para reduzir o Seek
- Outros fatores de desempenho
  - Organização de arquivos (contíguos/dispersos)
  - Localização da informação de controle
  - Cache

Algoritmos de Escalonamento de Disco

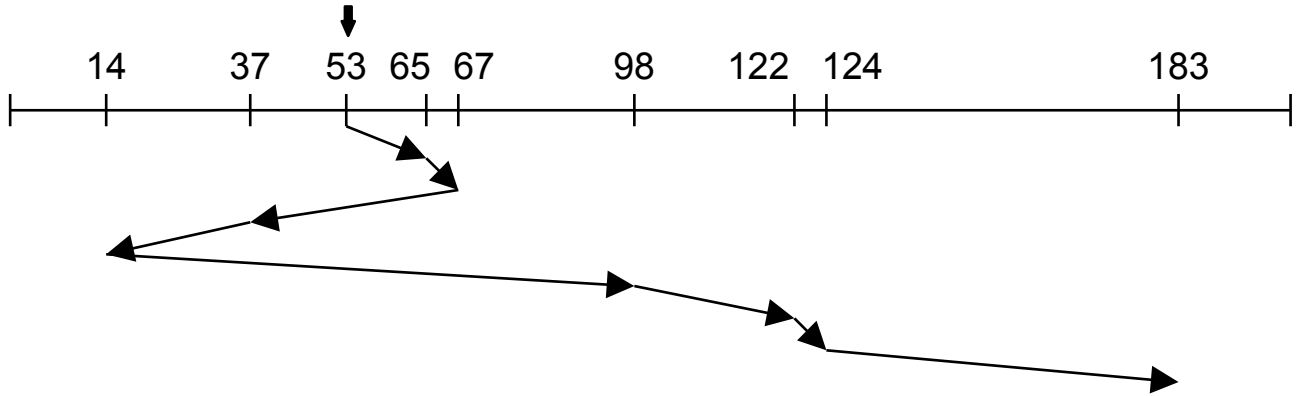
- First-Come First-Served (FCFS)

Queue: 98, 183, 37, 122, 14, 124, 65, 67 *Seek*: 640 tracks



- Shortest Seek Time First (SSTF)

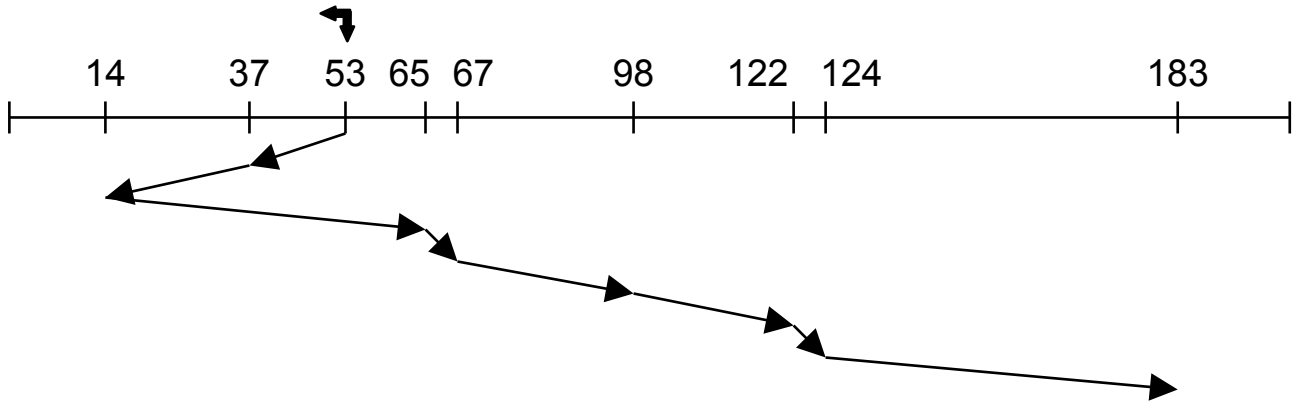
Queue: 98, 183, 37, 122, 14, 124, 65, 67 *Seek*: 236 tracks



Algoritmos de Escalonamento de Disco

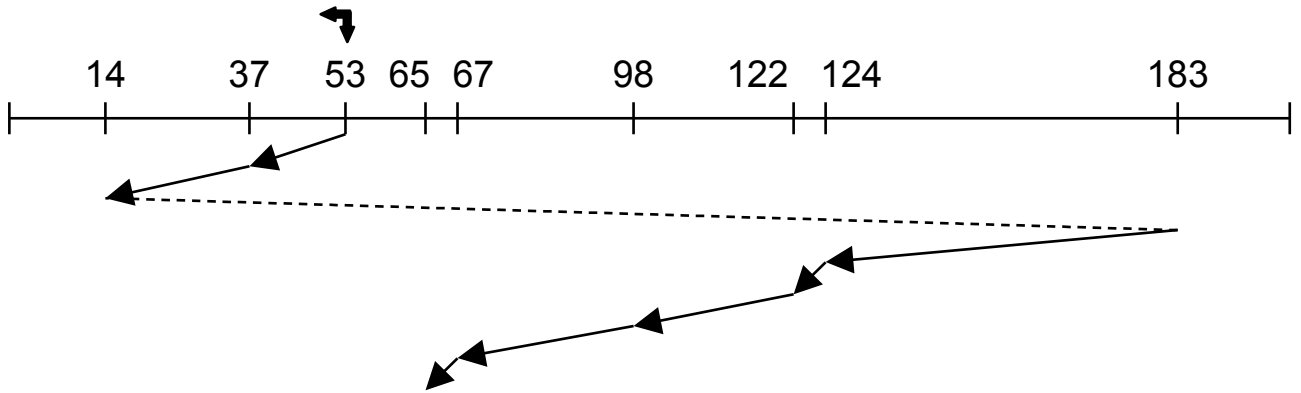
- Scan (Elevator)

Queue: 98, 183, 37, 122, 14, 124, 65, 67 *Seek*: 208 tracks



- Circular Scan (C-SCAN)

Queue: 98, 183, 37, 122, 14, 124, 65, 67 *Seek*: 326 tracks



Redundant Array of Independent Disks (RAID)

- RAID 0 (stripping)
  - Cada bloco é dividido em sub-blocos
  - Cada sub-bloco é armazenado em um disco diferente

  Alto desempenho
- RAID 1 (shadowing/mirroring)
  - Cada bloco é armazenado duas vezes
  - Alta confiabilidade
- RAID 5 (stripping + rotating parity)
  - Alto desempenho com boa confiabilidade