Hierarquia de Memória

Sistemas Operacionais

Pontifícia Universidade Católica de Campinas

Prof. Dr. Denis M. L. Martins

Objetivos de Aprendizado

Ao final desta aula, você será capaz de:

- Compreender abstrações de memória e arquivos em SO
- Descrever a estrutura física dos dispositivos de armazenamento secundário e o efeito da estrutura do dispositivo em seus usos.
- Explicar as características de desempenho dos dispositivos de armazenamento em massa.

Disclaimer

Parte do material apresentado a seguir foi adaptado de:

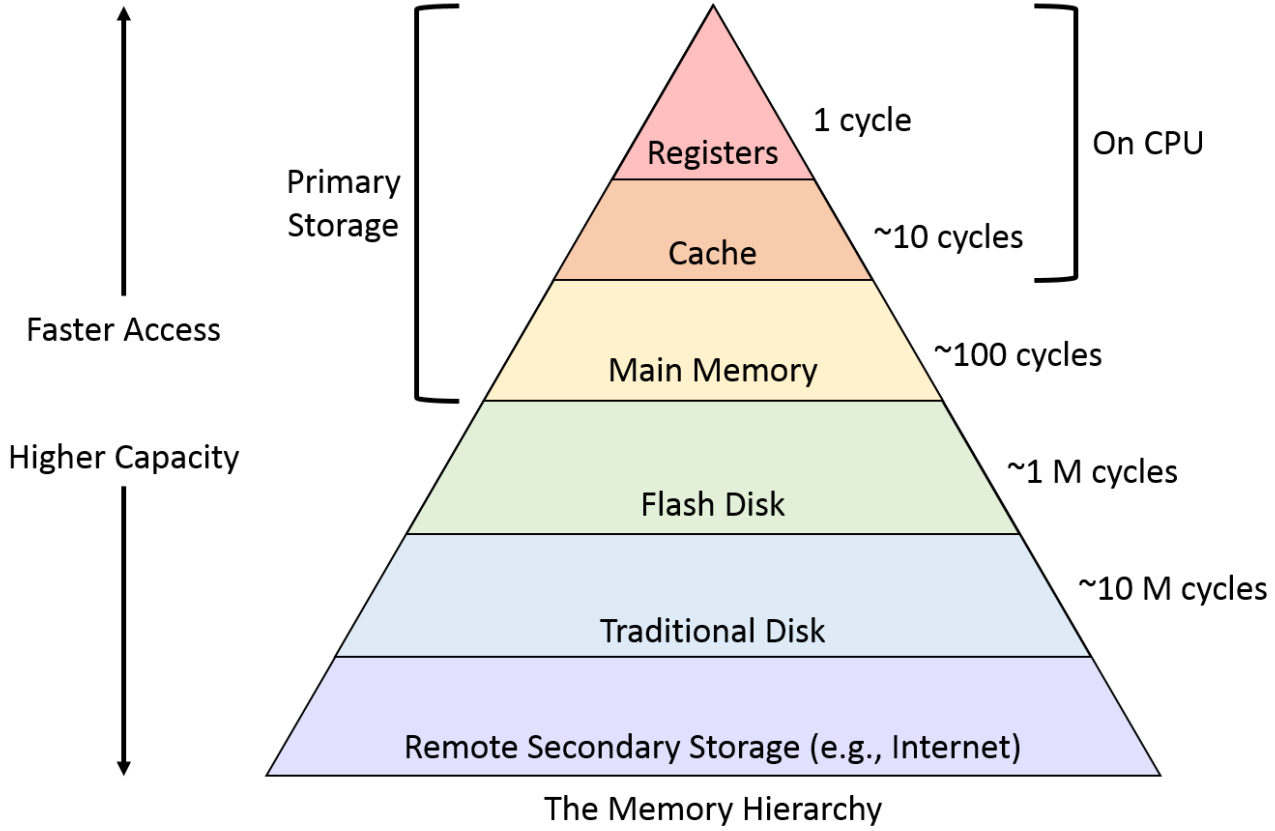
- IT Systems – Open Educational Resource, produzido por Jens~Lechtenböger; e
- Open Education Hub - Operating Systems

Imagens decorativas retiradas de Unsplash

Critérios para o SO

- Recuperar e armazenar o mais rápido possível
- Uso ótimo do espaço de memória
  - Quando os dados não estão sendo usados, a memória é liberada imediatamente
  - Minimizar o tempo em que a memória está reservada, mas não utilizada
  - Os dados devem ocupar o menor espaço necessário
- Segurança
  - Correção dos dados
  - Isolamento dos dados

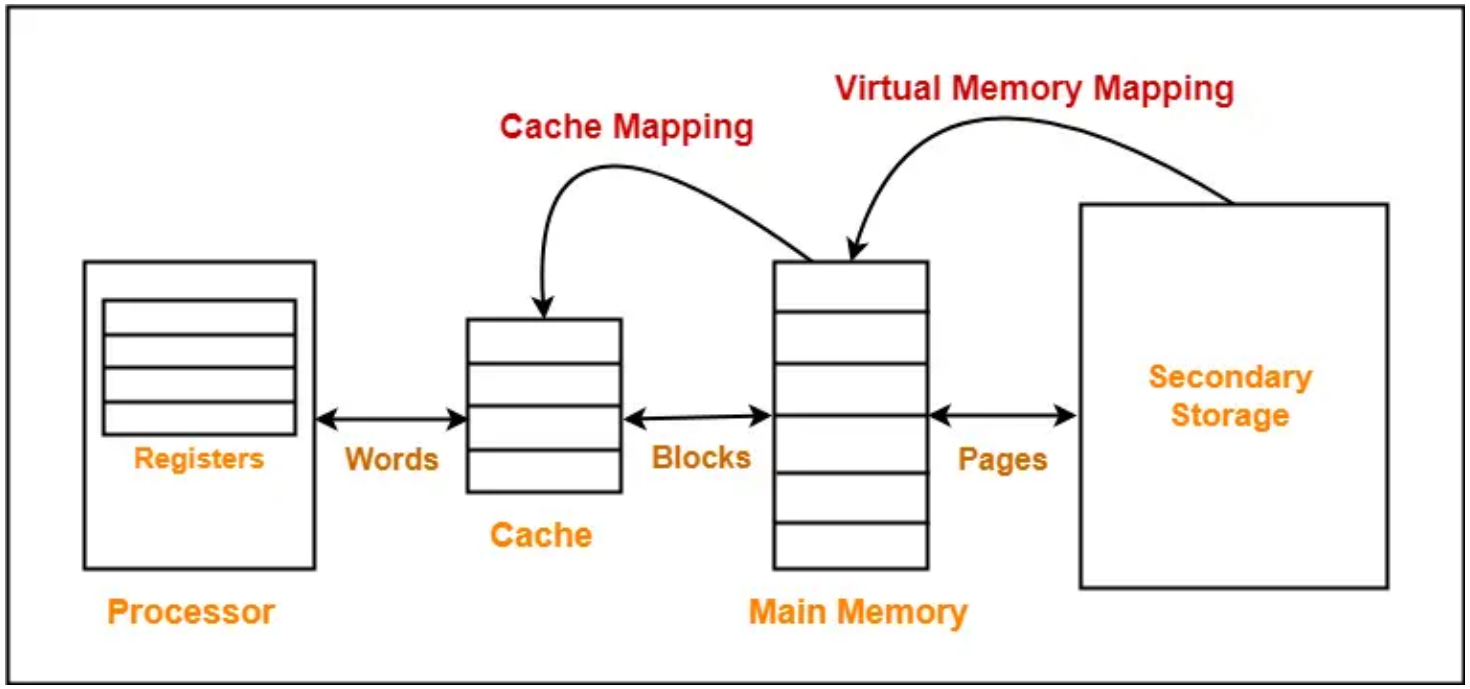
Hierarquia de memória



Fonte da Imagem: CS31

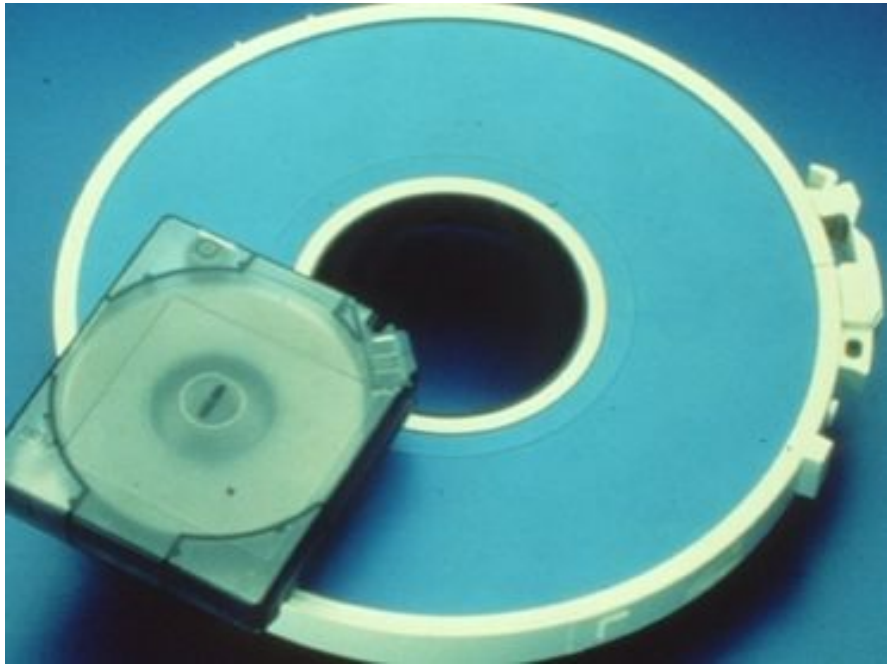
Ver também: Latency by Collin Scott

Hierarquia de memória



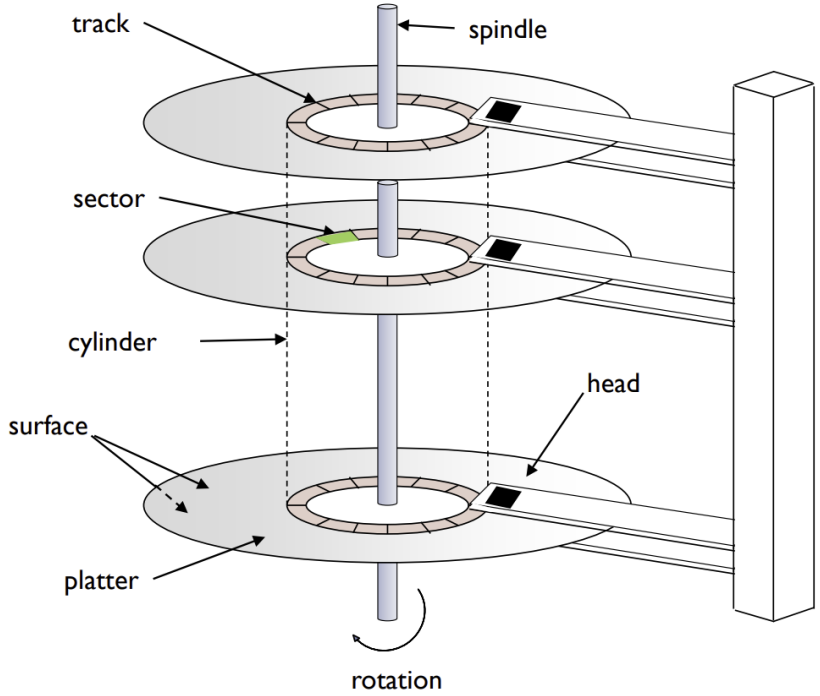
Fonte da Imagem: GateVidyalay

Fita Magnética



Fonte das Imagens: IBM

- Principalmente utilizada para backup, armazenamento de dados de uso infrequente.
- Capacidade de armazenamento típica varia entre 200 GB e 1,5 TB.
- Tempo de acesso lento.

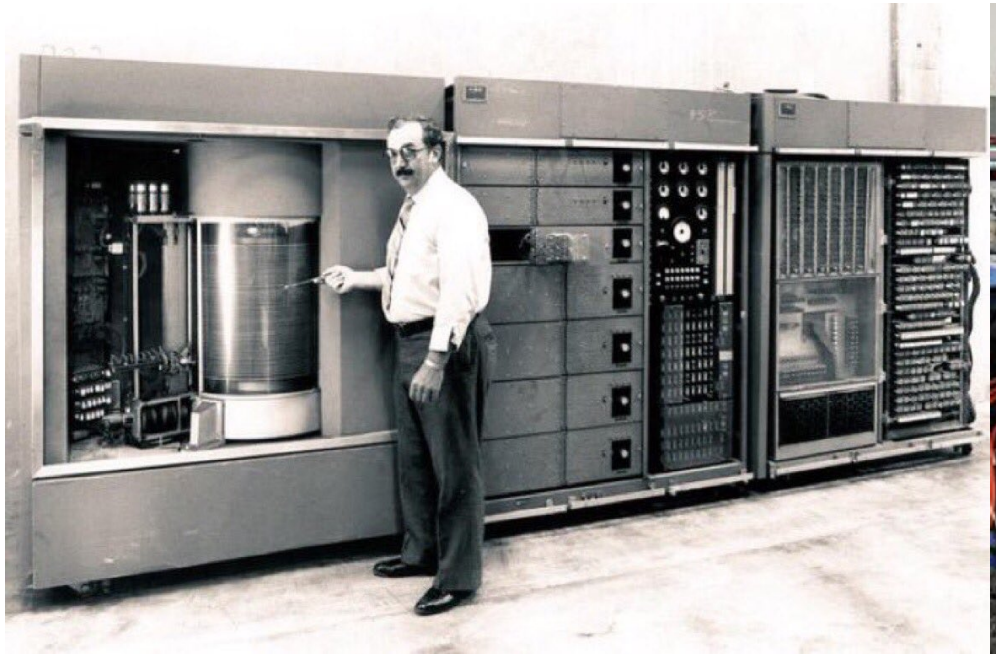


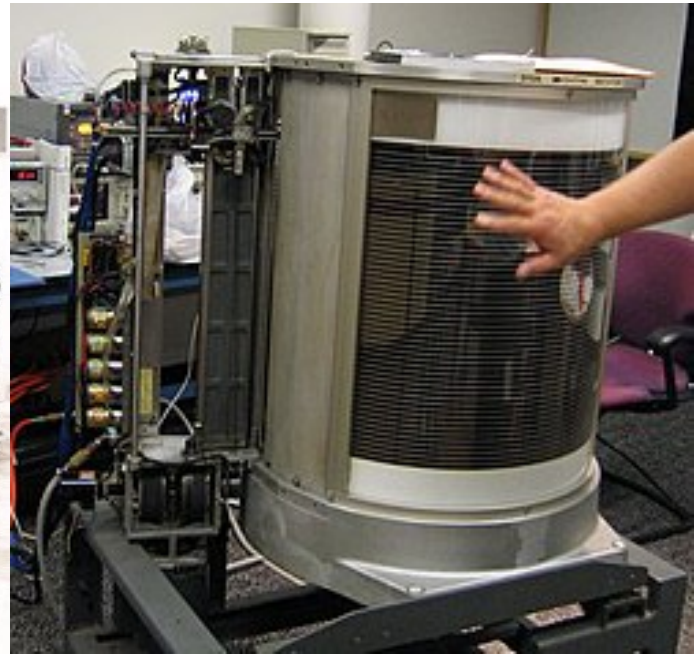
Hard Disk Drives (HDDs)

- Moving-head Disk Mechanism
- A capacidade varia de 30 GB a 3 TB por unidade.
- Tempo de Busca de 3 ms a 12 ms: 9 ms é comum para unidades de desktop.
- Latência dependente da velocidade do spindle
- Assista: How Hard Drives Work YouTube
- Fonte das imagens: Unsplash e Shy @ Verlog.io

O Primeiro Disco Rígido Comercial

- Em 1956, o computador IBM RAMDAC incluiu o sistema de armazenamento em disco IBM Modelo 350
- Capacidade de 5 milhões (7 bits) de caracteres, com 50 pratos de 50 x 24", e um tempo de acesso < 1s





Fonte das imagens: IBM e Wikipedia

Dispositivos de Memória Não Volátil

- Quando semelhantes às unidades de disco rígido: **discos de estado sólido (SSDs)**
- Outras formas incluem dispositivos USB (pen/flash drives), substitutos de DRAM (memória dinâmica de acesso aleatório) montados na placa-mãe e armazenamento principal em dispositivos como smartphones
- Podem apresentar maior confiabilidade do que HDDs
- Apresentam menor capacidade de armazenamento, mas mais veloz
- São mais caros por MB (megabyte). Podem ter vida útil mais curta
- Ausência de partes móveis elimina o tempo de busca e a latência rotacional

Armazenamento nos Videogames

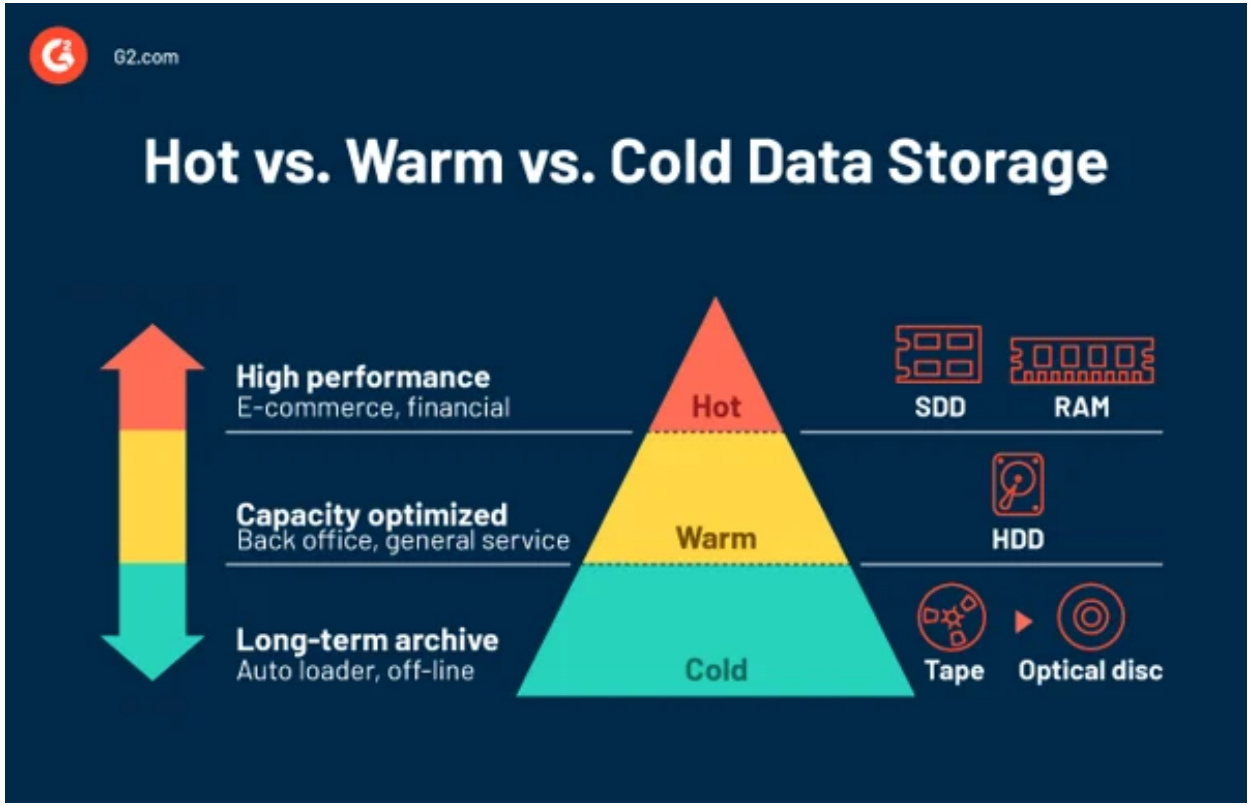
- **Final Fantasy VI (1994)**: O jogo apresenta elementos notáveis, incluindo gráficos de *16 bits* de ótima qualidade (que inspira muitos jogos ainda hoje), personagens cativantes e memoráveis que acompanham o jogador por cerca de 40 horas.
- **Tamanho**: *2,14 MB*.
- Fonte: The Gamer
- Fonte da Imagem: Flicker

Armazenamento nos Videogames

- **Final Fantasy VII (1997)**: O tamanho do jogo é tão grande que ele se estende por três discos diferentes.
- **Tamanho**: Cada disco possui cerca de *450 MB*, totalizando *1317 MB*.
- Fonte: The Gamer.
- Fonte da imagem: Arcade Art Work

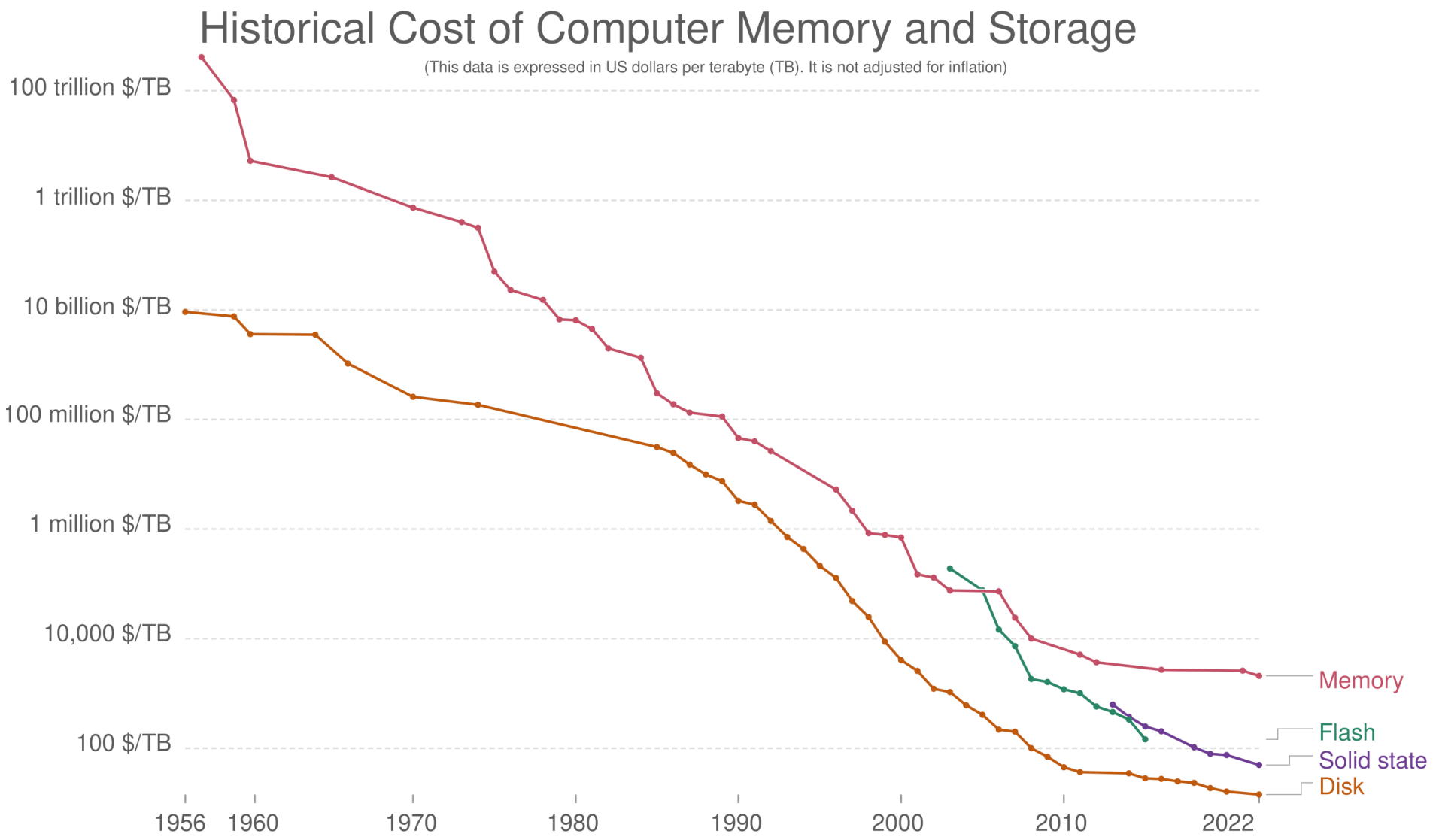
Data Tiering

- **Quente**: para dados estruturados e acessados com muita frequência.
  - Utilizados por funcionários ou clientes
- **Morno**: para dados estruturados e acessados com frequência moderada.
  - Utilizados para relatórios ou análises
- **Frio**: para dados estruturados ou não estruturados que são acessados com pouca frequência.
  - Motivos de conformidade legal
- Fonte da imagem: G2



Estrutura do Disco

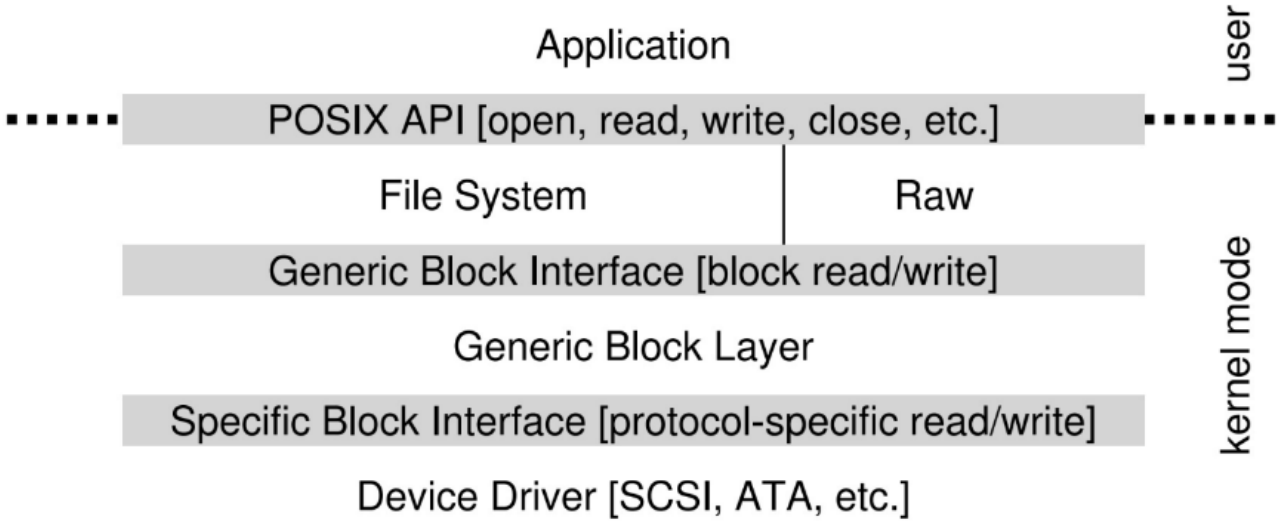
- Um disco pode ser subdividido em **partições**.
- Discos ou partições podem ser protegidos contra falhas através de **RAID**.
  - Partições: minidiscos ou fatias.
  - Disco ou partição pode ser utilizado no estado **bruto** – sem um sistema de arquivos – ou **formatado** com um sistema de arquivos.
- A entidade contendo o sistema de arquivos é conhecida como **volume**. Cada volume rastreia informações sobre esse sistema em um **diretório do dispositivo** ou uma **tabela de volumes**, indicando seu conteúdo.
- Fonte da imagem: diskinternals.com



Fonte da imagem: History of hard disk drives @ Wikipedia

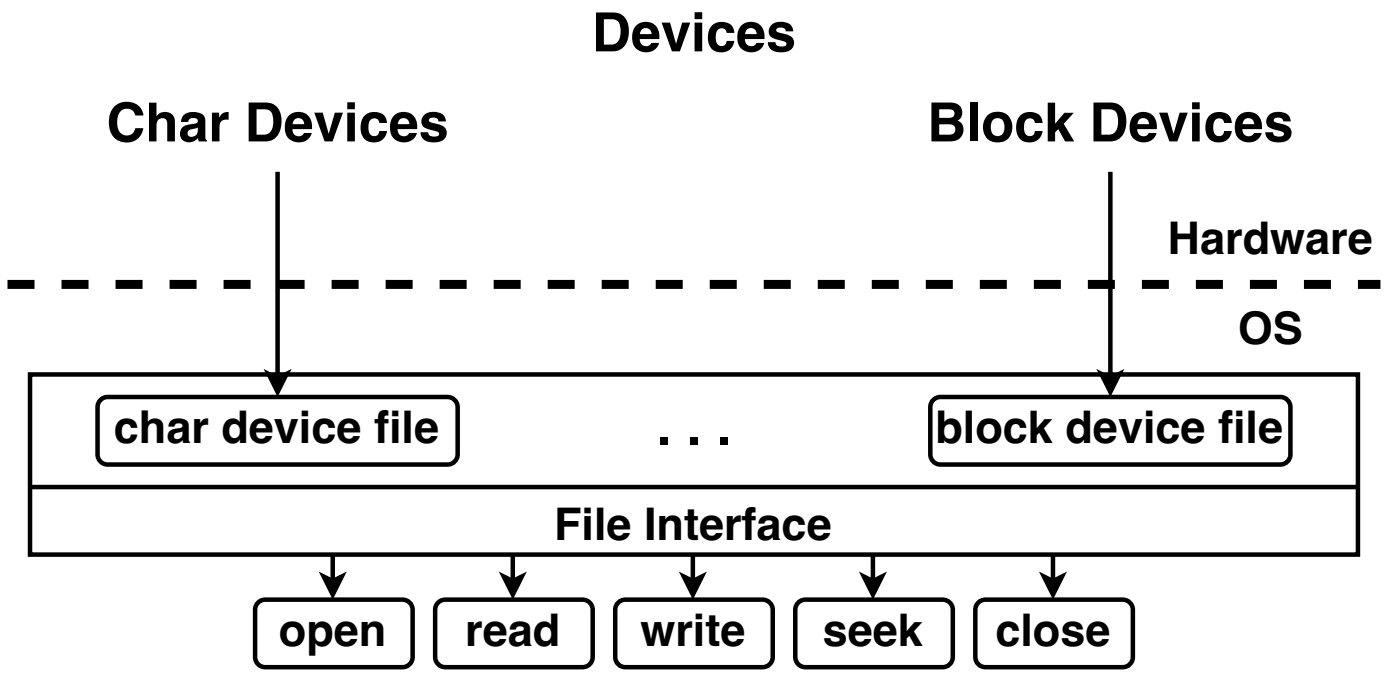
Interface de I/O para Aplicações

- Chamadas de sistema encapsulam comportamentos de dispositivos
- Drivers escondem diferenças entre controladores
- Novos dispositivos usando protocolos existentes não exigem mudanças no SO



Fonte da Imagem: Anotações de JungJae Lee

Representação de Dispositivos de I/O no Linux



Linux abstrai um dispositivo de I/O como um arquivo especial (diretório `/dev`).

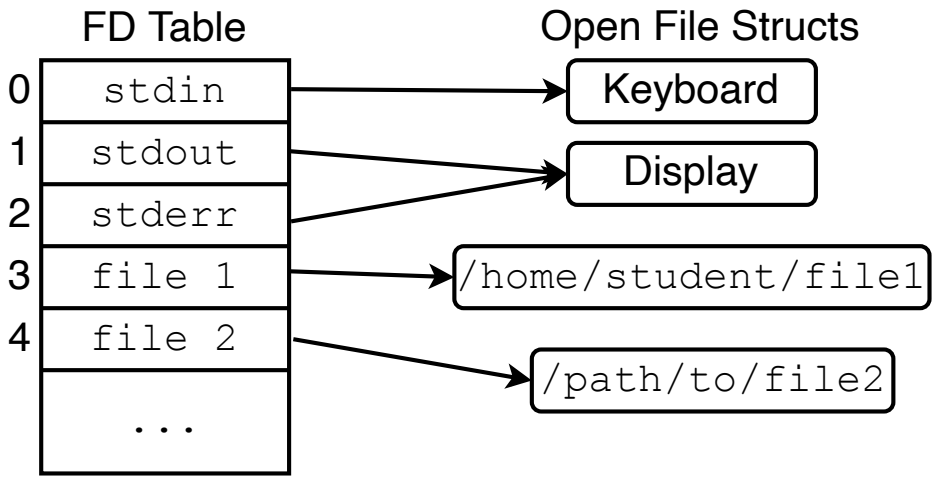
Fonte da Imagem: OS Team - OS OER

Arquivos

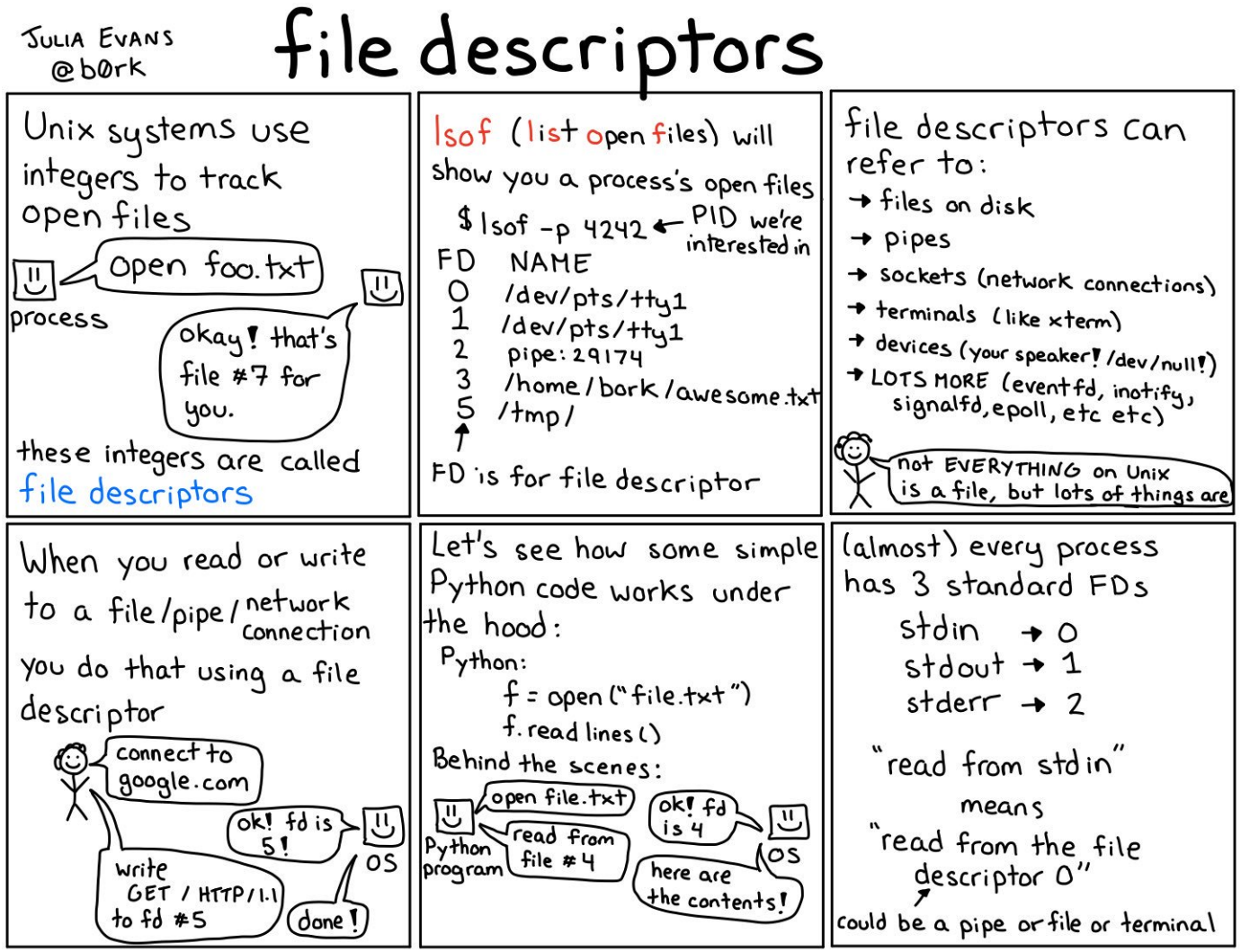
- Arquivos são abstrações comuns do SO para organizar dados e armazená-los de maneira persistente.
  - Persistência: SO deve manter dados mesmo que haja cortes de energia ou falhas no sistema.
- *File Descriptors*: SO representa arquivos via números inteiros chamados *descriptors*
  - Arquivos: *named streams of bytes*
  - Abstração: arquivos, diretórios, dispositivos de I/O, acesso de rede, etc.
- Operações: `open`, `close`, `read`, `write`

Arquivos

- O padrão POSIX descreve 3 descritores (numerados 0, 1, 2) para cada processo:
  - `0` : *Standard input*, `stdin` (e.g., entrada do teclado)
  - `1` : *Standard output*, `stdout` (e.g., imprimir na tela ou no terminal)
  - `2` : *Standard error*, `stderr` (e.g., imprimir mensagem de erro no terminal)



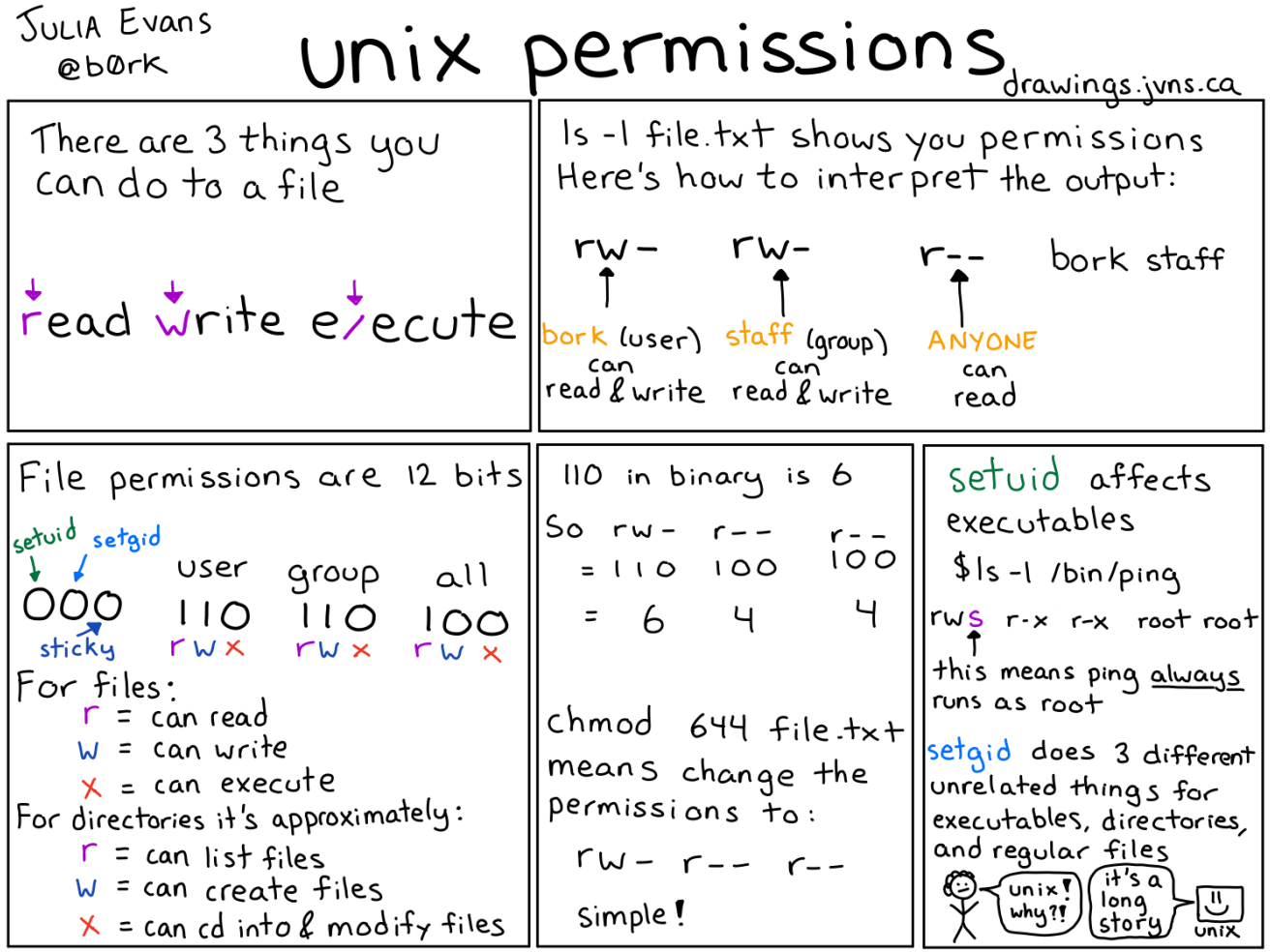
Fonte da Imagem: OS Team - OS OER



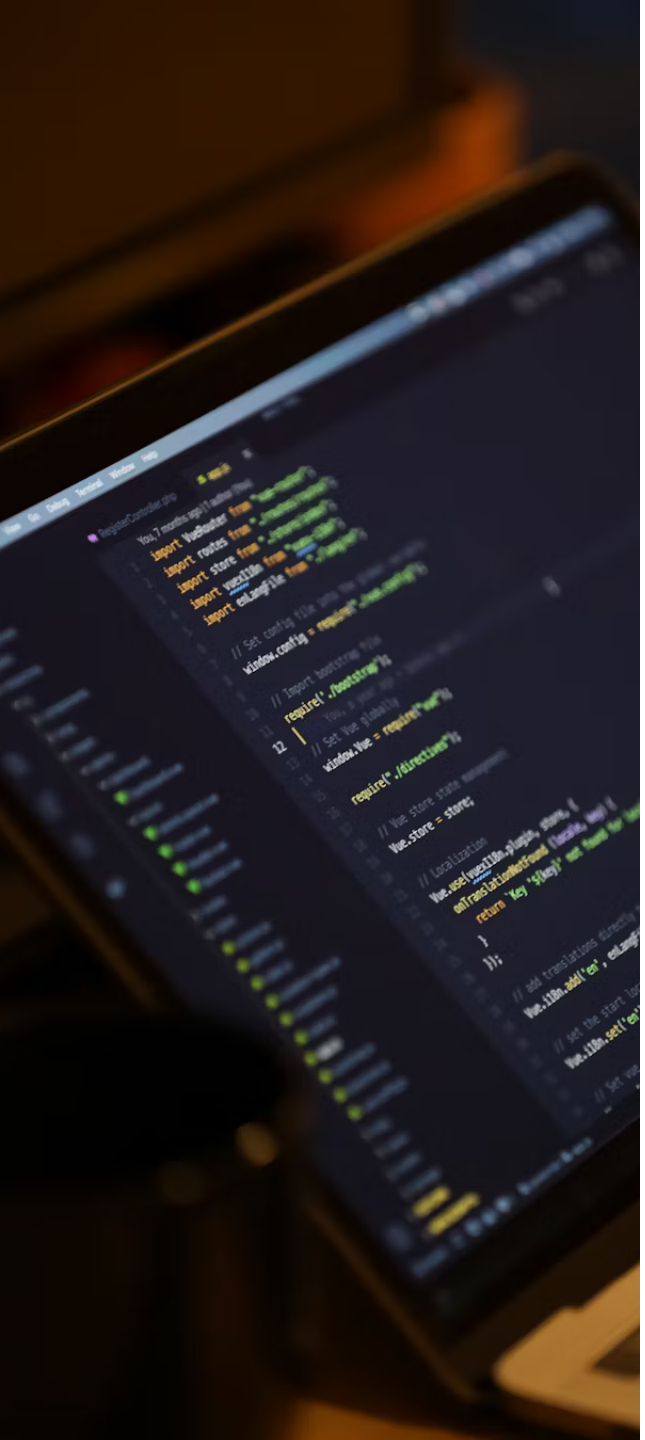
Fonte da Imagem: Julia Evans

Permissões de Acesso

- Quem tem permissão para fazer o quê?
- O sistema controla o acesso a objetos por sujeitos.
- **Objeto:** qualquer coisa que precise ser protegida: por exemplo, uma região de memória, um arquivo, um serviço.
  - Com operações diferentes dependendo do tipo de objeto.
- **Sujeito:** entidade ativa que utiliza os objetos, ou seja, um processo.
  - Threads dentro de um processo compartilham as mesmas permissões de acesso.
  - O sujeito pode também ser o próprio objeto, por exemplo, terminar uma thread ou um processo.

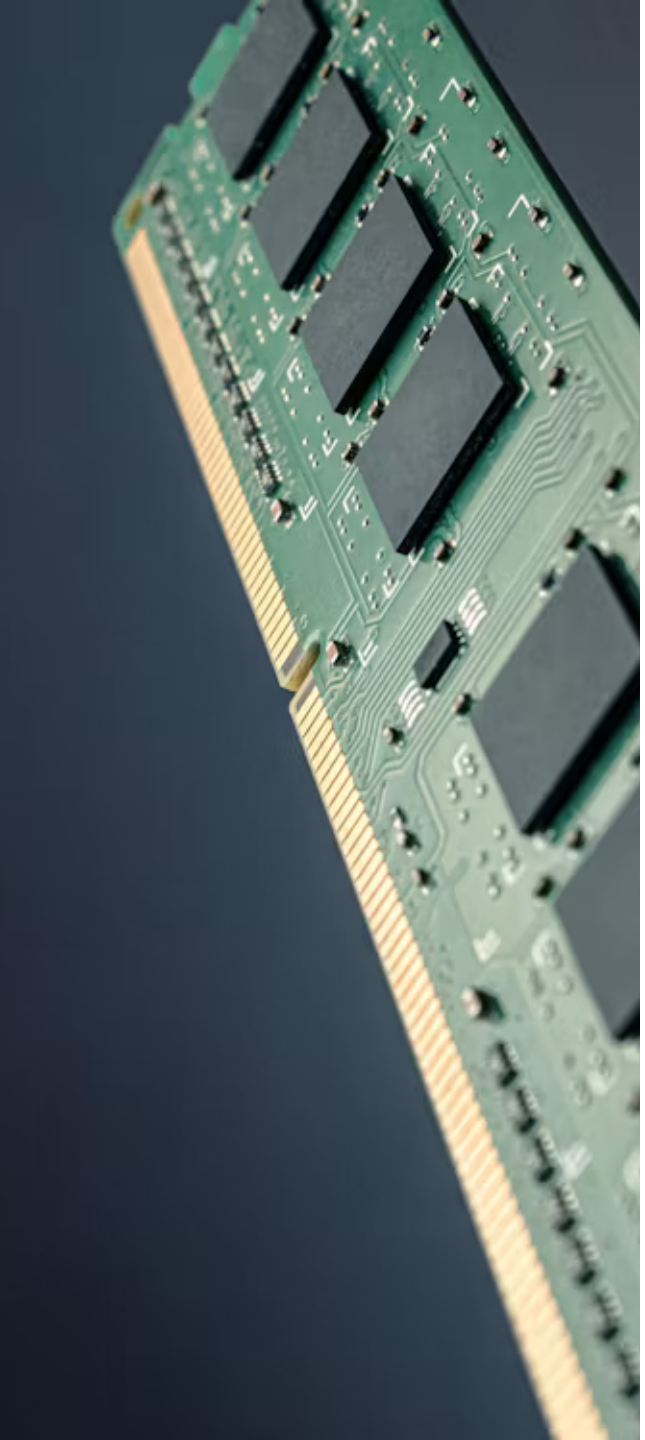


Fonte da Imagem: Julia Evans



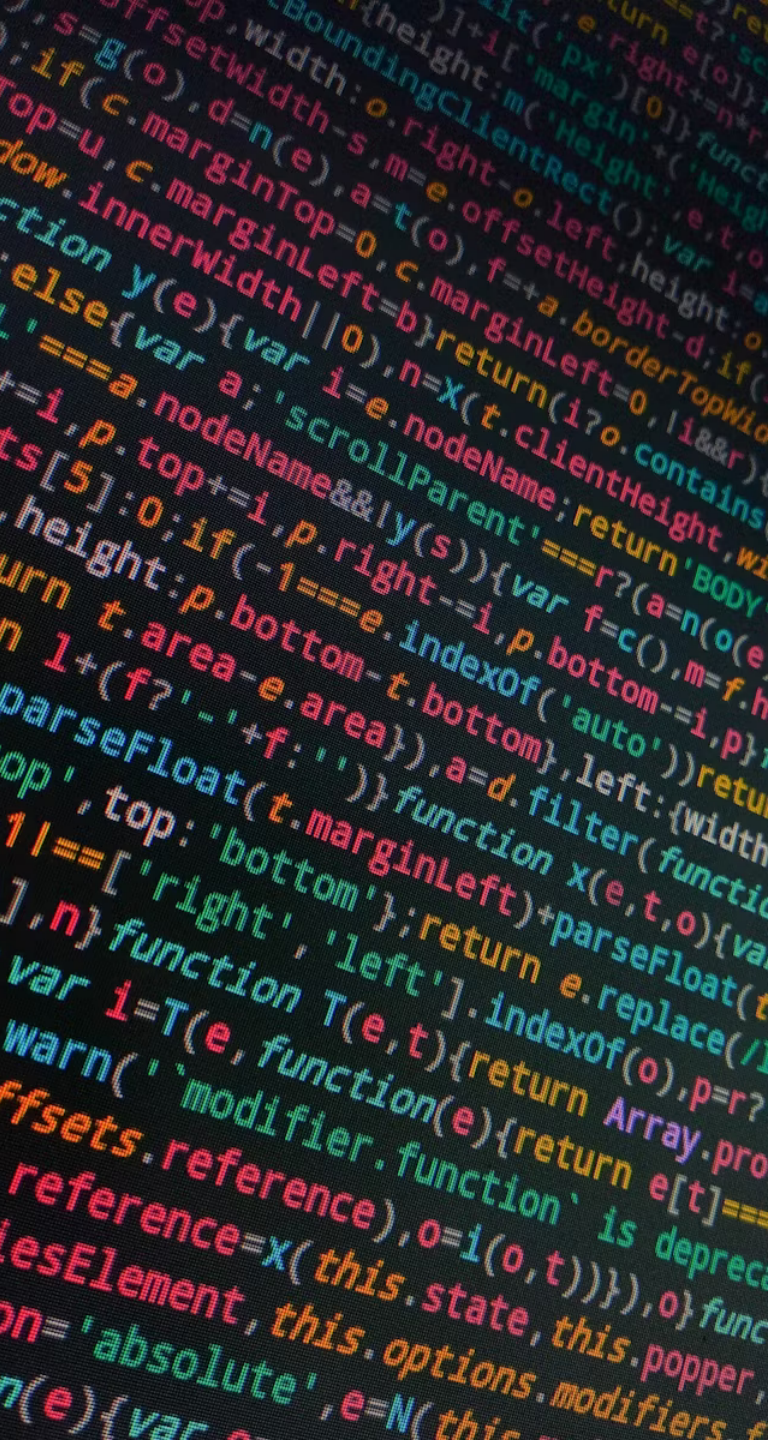
Uma Perspectiva do Programador sobre os Dados

- Dados = variáveis
- Operações: declarar/ler/escrever
- Variáveis são armazenadas na memória, portanto, dependendo da linguagem, você também pode:
  - Alocar memória
  - Desalocar memória



Memória Volátil

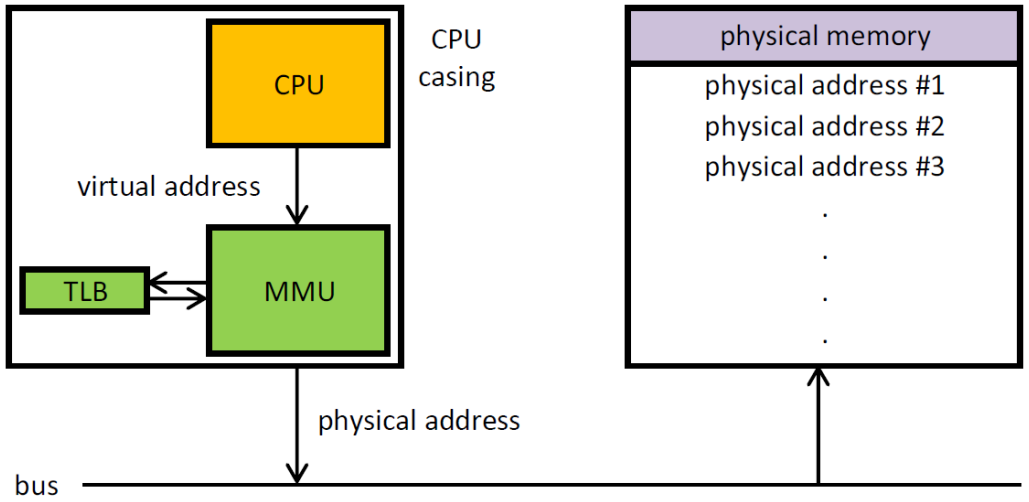
- A DRAM (Memória Dinâmica de Acesso Aleatório) é frequentemente utilizada como dispositivo de armazenamento em massa.
  - Tecnicamente não se qualifica como armazenamento secundário devido à sua volatilidade, mas pode apresentar sistemas de arquivos e ser utilizada como um armazenamento secundário extremamente rápido.
- Unidades RAM (conhecidas por diversos nomes, incluindo discos RAM) são apresentadas como dispositivos de bloco brutos, geralmente formatados com um sistema de arquivos.

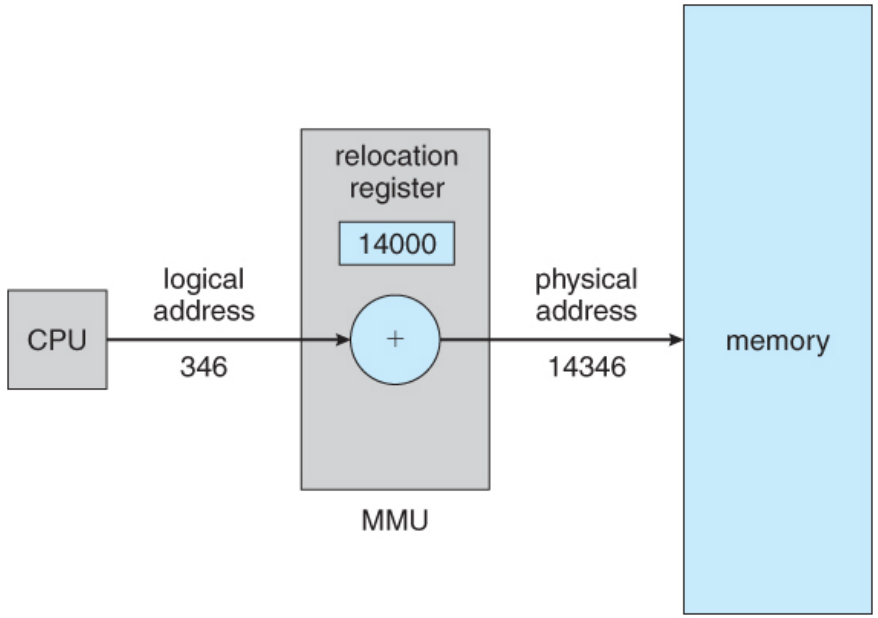


Quem gerencia a memória?

- Você (o programador) - C/C++
- A linguagem de programação - Python, Java
- Uma implementação de biblioteca - C/C++
- O sistema operacional - para todas as linguagens

Memory Management Unit (MMU)





Fonte das imagens: Wikipedia e Silberschatz et al., Operating System Concepts

- CPU não acessa diretamente endereços de memória física $\longrightarrow$ Requisita da MMU endereços virtuais
- MMU traduz endereços virtuais em endereços físicos (**extremamente rápida**)
- Kernel envolvido para tarefas complexas (por exemplo, decidir o que remover da memória)

Escalonamento de Disco

- O sistema operacional é responsável por utilizar o hardware de forma eficiente – no caso das unidades de disco rígido, isso implica em minimizar o tempo de acesso e maximizar a largura de banda do disco.
- Minimizar o tempo de busca.
- Tempo de busca é aproximadamente proporcional à distância de busca.
- A largura de banda (*bandwidth*) do disco é definida como o número total de bytes transferidos dividido pelo tempo total entre o primeiro pedido de serviço e a conclusão da última transferência.

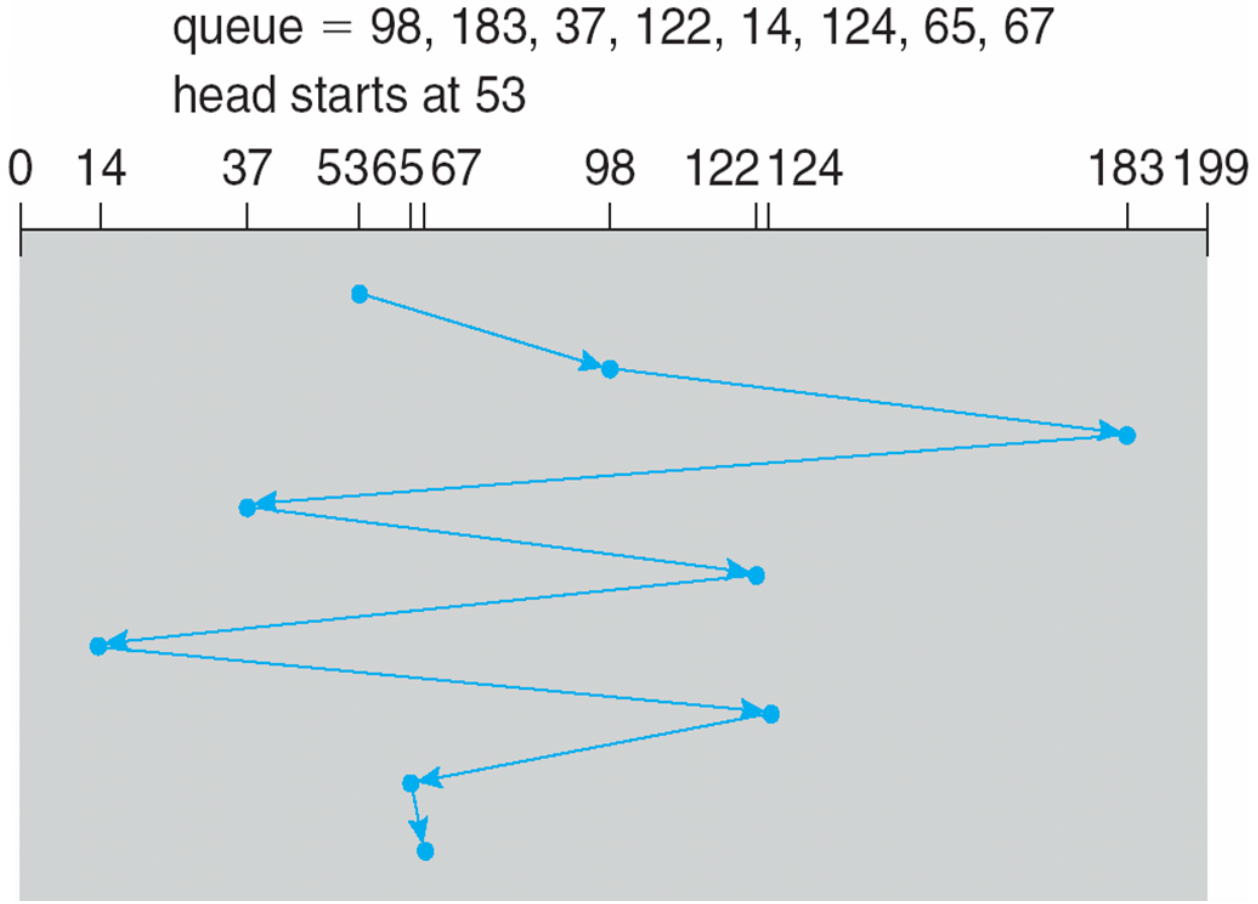
Escalonamento de Disco (cont.)

- Existem diversas fontes de requisições de E/S do disco:
  - Processos do sistema
  - Processos do usuário
- Uma requisição de E/S inclui o modo de entrada ou saída, endereço do disco, endereço da memória e número de setores a serem transferidos.
- O SO mantém uma fila de requisições, por disco ou dispositivo.
  - Um disco inativo pode trabalhar imediatamente em uma requisição de E/S; um disco ocupado significa que o trabalho deve ser enfileirado.
- Note que os controladores de disco possuem buffers pequenos e podem gerenciar uma fila de requisições de E/S (de "profundidade" variável).

Escalonamento de Disco (cont.)

- No passado, o sistema operacional era responsável pelo gerenciamento da fila e pelo agendamento dos cabeçotes do disco rígido. Atualmente, isso está integrado aos dispositivos de armazenamento, controladores.
  - Atualmente, integrado aos dispositivos de armazenamento, controladores.
  - Apenas fornecem endereços lógicos de bloco (LBAs) e lidam com a ordenação das requisições.
- Diversos algoritmos existem para agendar a execução das requisições de E/S do disco.

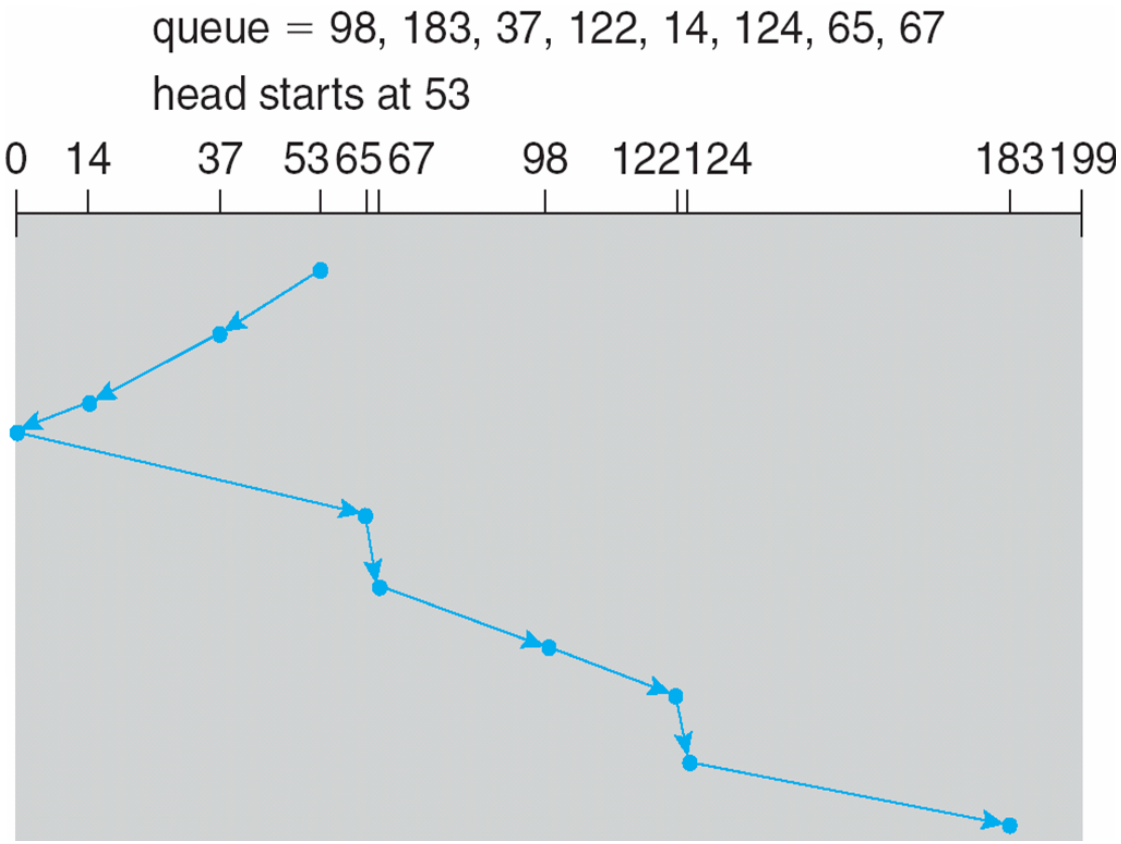
FCFS



Movimento total do cabeçote de 640 cilindros. Fonte: A. Silberschatz *et. al*, *Operating Systems Concepts*.

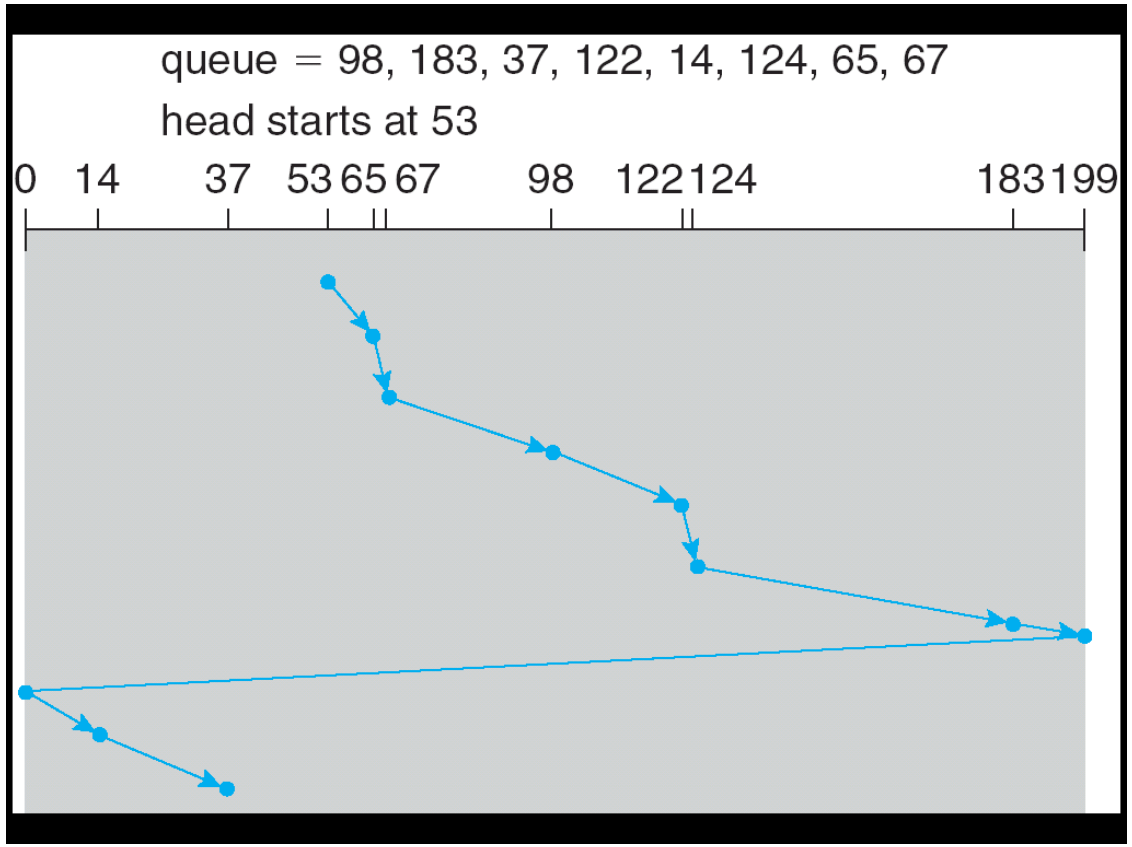
SCAN (algoritmo do elevador)

- O braço do disco inicia em uma extremidade do disco e move-se em direção à outra extremidade, atendendo às requisições até chegar ao final oposto, onde o movimento do cabeçote é invertido (repete a operação).
- As requisições forem densamente distribuídas uniformemente, a maior densidade estará na outra extremidade do disco e essas requisições aguardarão o tempo mais longo.
- Na figura: movimento total do cabeçote de 208 cilindros. Fonte: A. Silberschatz *et. al*, *Operating Systems Concepts*.



C-SCAN

- O cabeçote move-se de uma extremidade do disco à outra, atendendo às requisições durante o percurso.
- Ao atingir a outra extremidade, ele retorna imediatamente ao início do disco, sem atender nenhuma requisição na viagem de volta.
- Trata os cilindros como uma lista circular que se envolve do último cilindro para o primeiro.
- Fonte da imagem: A. Silberschatz *et. al*, *Operating Systems Concepts*.



Seleção de Algoritmo

- Como qualquer algoritmo de escalonamento, o desempenho depende muito da quantidade dos dados e dos tipos de solicitação
- SCAN e C-SCAN são menos prováveis de gerar problema de inanição, quando uma dada requisição nunca é atendida
- Métricas relevantes:
  - Tempo de busca (seek time): Tempo necessário para posicionar o cabeçote na trilha desejada
  - Tempo de latência rotacional (Latency time): Tempo necessário para atingir o início do setor a ser lido/escrito
  - Tempo para escrita/leitura efetiva dos dados

Conclusão

- **Considerações para Sistemas Operacionais:**
  - **Abstração do Hardware:** Como o SO fornece uma interface consistente para diferentes tipos de dispositivos de armazenamento.
  - **Gerenciamento de Espaço Livre:** Alocação e desalocação eficiente de espaço em disco.
  - **Particionamento:** Dividir discos em partições lógicas para organizar dados e permitir múltiplos sistemas operacionais.
  - **Interfaces de Discos:** Visão geral das tecnologias de conexão (SATA, NVMe, USB, Fibre Channel) e suas características
- **Gerenciamento de E/S e Desempenho:**: Como o sistema operacional organiza as requisições para minimizar o tempo de acesso ao disco (abordagem teórica).

Conclusão (cont.)

- **Conceitos Chave:**
  - **Latência:** Tempo de resposta do dispositivo de armazenamento.
  - **Throughput:** Taxa de transferência de dados.
  - **Redundância:** Duplicação de dados para proteção contra falhas.
  - **Escalabilidade:** Capacidade de aumentar a capacidade de armazenamento conforme necessário.

Material Adicional

- Paying for Cloud Storage is Stupid YouTube
- How do hard drives work? YouTube
- What is LTO TAPE?/FUJIFILM YouTube
- How computer memory works YouTube
- Registers and RAM: Crash Course Computer Science #6 YouTube
- Data Tiering in Heterogeneous Memory Systems

Dúvidas e Discussão