



Estruturas de Dados em Arquivos: Fundamentos



Prof. Paulo H. R. Gabriel

Créditos

- ▶ **Parte desse material foi baseado nas aulas de:**
 - ▶ Moacir Ponti Jr.
 - ▶ Thiago A. S. Pardo
 - ▶ Leandro C. Cintra
 - ▶ Maria C. F. de Oliveira



Memória Secundária

- ▶ Até agora, estudamos estruturas de dados que trabalham com informação em memória principal
 - ▶ RAM
- ▶ Diferente da memória RAM, a memória secundária não é acessada diretamente pela CPU
 - ▶ Dados acessado da memória secundária são carregados primeiro na RAM e, então, é enviados para o processador



Memória Secundária: Arquivos



Memória Secundária

- ▶ HD
- ▶ Disquetes
- ▶ Fitas magnéticas
- ▶ CD, DVD e Blu-ray
- ▶ Pendrives, SSD



Alguns Termos

- ▶ **Arquivo:** estrutura de dados em um sistema de arquivos, mapeando *nomes* para *objetos*
- ▶ **Estrutura de arquivo** (*file structure*): um padrão para se organizar dados num arquivo
 - ▶ Incluindo leitura, escrita e modificação



Terminologia de acesso a arquivos

- ▶ **Armazenamento volátil:** armazenamento que perde o conteúdo quando não alimentado por energia
- ▶ **Armazenamento não-volátil:** armazenamento que retém o conteúdo quando não alimentado por energia
- ▶ **Dados persistentes:** informação que é retida mesmo após a execução de um programa que a cria



Comparações

- ▶ Custo (aprox.):

- ▶ HD: 1,0 GB – R\$ 0,20

- ▶ (SSD: R\$ 2,50)

- ▶ RAM: 1,0 GB – R\$ 40,00

- ▶ Velocidade (aprox.):

- ▶ HD: 160 MB/s

- ▶ (SSD: 500 MB/s)

- ▶ RAM: 17 GB/s

- ▶ Logo: podemos armazenar muito mais, mas perdemos em eficiência...



Objetivos das Estruturas de Arquivos

- ▶ **Minimizar o número de acessos ao disco**
 - ▶ Idealmente obter e processar a informação em um único acesso ao disco
- ▶ **Maximizar a quantidade de informações recuperadas ou processadas em um acesso**
 - ▶ Agrupando informações relacionadas
- ▶ **De forma independente da tecnologia**
 - ▶ $\text{Tempo de acesso} = \text{Número de acessos} \times \text{Tempo de 1 acesso}$
 - ▶ Deve-se ter cautela para não projetar uma estrutura de arquivo muito dependente da tecnologia atual



Estruturas de Arquivos

- ▶ Estruturas de dados eficientes em memória são muitas vezes inviáveis em disco
- ▶ Um dos problemas em se obter uma estrutura de dados adequada é a constante necessidade de alterações em arquivos
- ▶ O ideal é evitar sequências de acesso



Estruturas de Arquivos

- ▶ É preciso um método que recupere esse mesmo registro em poucos acessos
- ▶ Para isso, devemos agrupar informações para permitir recuperar o máximo de informação em uma única operação de acesso
 - ▶ Por exemplo, ao consultar um pedido de um cliente e buscar suas informações pessoais (nome, endereço, telefone, CPF, etc.), é preferível obter todas as informações de uma vez ao invés de procurar em vários lugares



Arquivos Físicos e Lógicos

- ▶ Um arquivo sempre é físico do ponto de vista do armazenamento
 - ▶ É um conjunto de bytes armazenados e rotulados com um nome
- ▶ Para um aplicativo a noção é diferente: só é possível acompanhar o fluxo de bytes de leitura e escrita no arquivo
- ▶ Os bytes podem ser originários de um arquivo físico, do teclado ou outros dispositivos
 - ▶ Assim, a “linha de comunicação” aberta pelo sistema operacional para o programa é chamado de arquivo lógico
 - ▶ O programa é geralmente limitado a 20 conexões com arquivos



Arquivos Físicos e Lógicos

- ▶ No código fonte, uma instrução liga o arquivo físico a uma variável lógica
- ▶ Uma vez ligado, é preciso declarar o que desejamos fazer com o arquivo
- ▶ Em geral, temos duas opções: abrir um arquivo existente, *ou* criar um novo arquivo, apagando qualquer conteúdo anterior no arquivo físico
 - ▶ Após abrir um arquivo estaremos posicionados no início do arquivo e portanto prontos para escrever ou ler.



Breve Histórico: Fitas Magnéticas

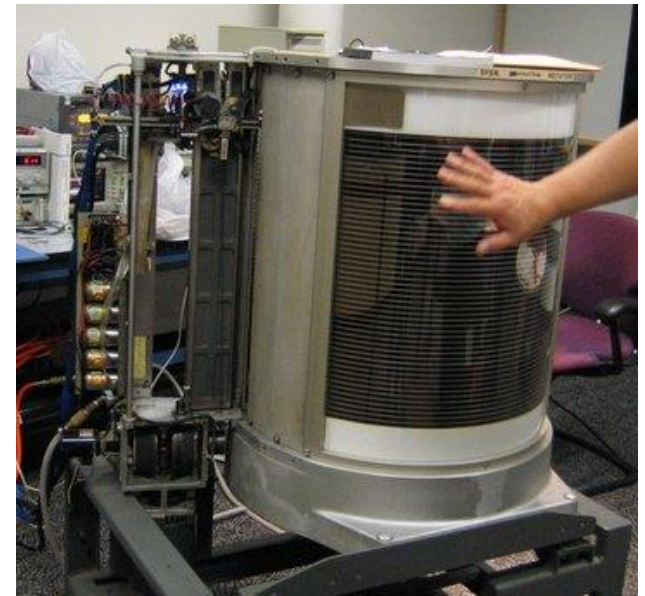
- ▶ Os primeiros trabalhos com arquivos presumiam o armazenamento em *fitas acesso sequencial*
 - ▶ Porém, o tamanho dos arquivos cresceu muito e inviabilizou esse tipo de acesso
- ▶ São, no entanto ainda usadas principalmente para *backup*
 - ▶ Vida útil de até 100 anos



Fonte: <https://duncanstephen.net/>

Breve Histórico: *Hard Disk*

- ▶ Com os discos:
 - ▶ Foram adicionados índices aos arquivos
 - ▶ Tornaram possível manter uma lista de chaves e ponteiros para acesso aleatório
 - ▶ Mais uma vez o crescimento dos arquivos de índice tornou difícil a sua manutenção
- ▶ Necessidade de
- ▶ *estruturas de dados mais eficientes*



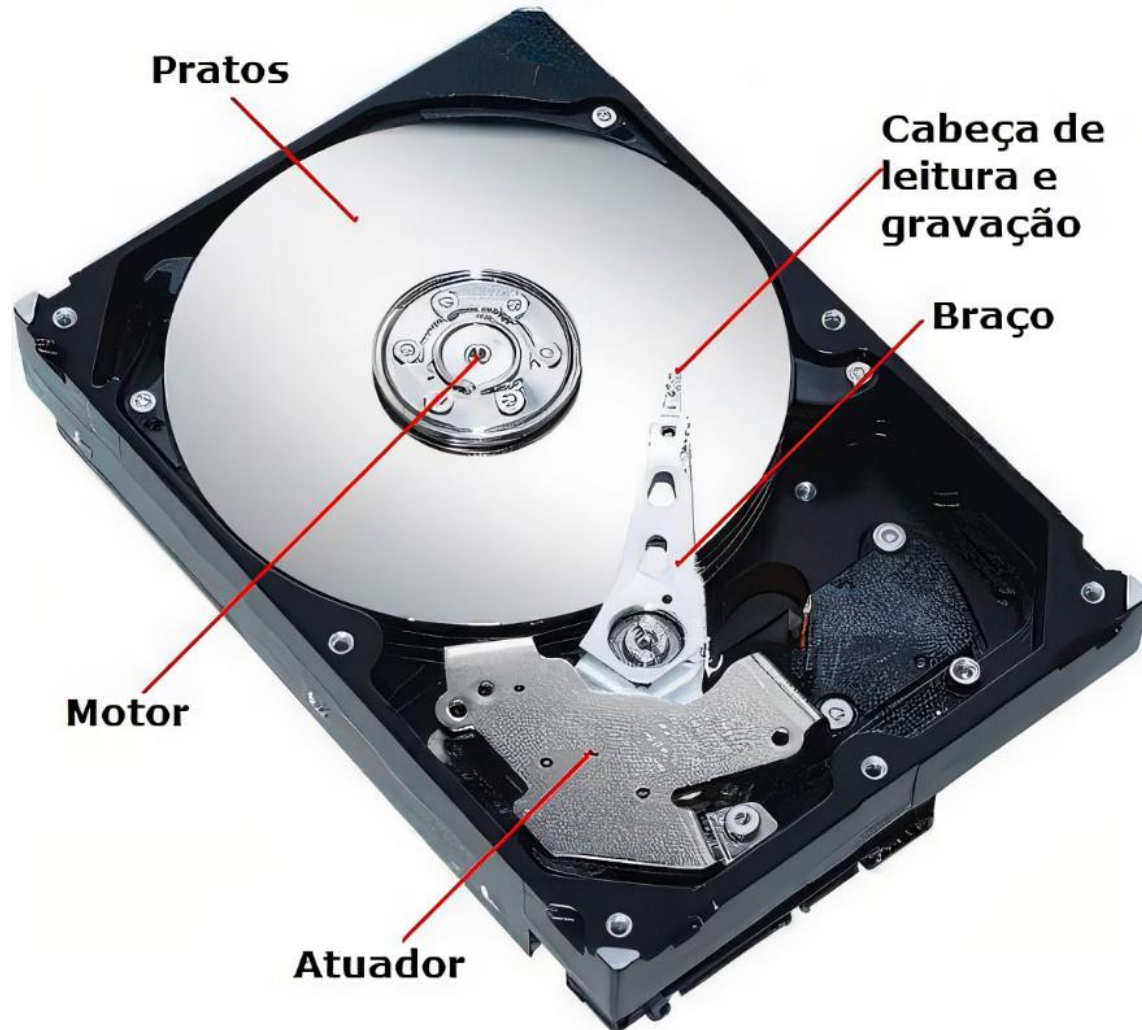
Fonte: Wikipedia (IBM 350)

HD: Componentes Importantes

- ▶ Discos (pode haver vários)
 - ▶ Substrato sólido (alumínio, vidro ou cerâmica)
 - ▶ Superfície magnética: filme magnético
- ▶ Cabeças de leitura/escrita: leem e escrevem nos discos enquanto eles giram (> 7000 RPM, em HDs comerciais)
 - ▶ Convertem entre sinais elétricos e pulsos magnéticos
- ▶ A informação lida é armazenada em um *buffer*, de onde é transferida para a memória



HD: Componentes Importantes



Fonte: [Blog do Disco Rígido \(memoriasecundaria.blogspot.com\)](http://memoriasecundaria.blogspot.com)

Em C...

- ▶ **Dois níveis de manipulação:**
 - ▶ Funções de baixo nível
 - ▶ Funções de alto nível
- ▶ **Observação:**
 - ▶ As funções de alto nível são implementadas em baixo nível, mantém área de memória (*buffer*) para manipulação dos bytes



O TAD Arquivo em C

► **No** `stdio.h`

```
#define FOPEN_MAX (20)

typedef struct _iobuf {
    char* _ptr;
    int _cnt;
    char* _base;
    int _flag;
    int _file;
    int _charbuf;
    int _bufsiz;
    char* _tmpfname;
} FILE;
```

