

Algoritmos e Estruturas de Dados II

Aula 18 – Arquivos - Acesso à Memória Secundária

Prof. Luciano A. Digiampietri
digiampietri@usp.br
@digiampietri

2025

Arquivos: Acesso à Memória

Por que se preocupar com alocação de arquivos na memória secundária?

Arquivos: Acesso à Memória

Por que se preocupar com alocação de arquivos na memória secundária?

Porque há alguns desafios específicos relacionados à memória secundária (em especial o tempo de acesso).

Arquivos: Armazenamento Não Volátil

- Quando estamos falando em arquivos, normalmente estamos falando de armazenamento em memória secundária:

Arquivos: Armazenamento Não Volátil

- Quando estamos falando em arquivos, normalmente estamos falando de armazenamento em memória secundária:
 - HD (discos rígidos - *hard drives*)

Arquivos: Armazenamento Não Volátil

- Quando estamos falando em arquivos, normalmente estamos falando de armazenamento em memória secundária:
 - HD (discos rígidos - *hard drives*)
 - SSD (*solid state disk*)

Arquivos: Armazenamento Não Volátil

- Quando estamos falando em arquivos, normalmente estamos falando de armazenamento em memória secundária:
 - HD (discos rígidos - *hard drives*)
 - SSD (*solid state disk*)
 - CD-ROM
 - DVD
 - Pen-drives
 - Fita magnética
 - ...

Arquivos: Armazenamento Não Volátil

- Quando estamos falando em arquivos, normalmente estamos falando de armazenamento em memória secundária:
 - HD (discos rígidos - *hard drives*)
 - SSD (*solid state disk*)
 - CD-ROM
 - DVD
 - Pen-drives
 - Fita magnética
 - ...

HD x Memória Principal

- Capacidade de Armazenamento:

- Tipo de Armazenamento:

HD x Memória Principal

- Capacidade de Armazenamento:
 - HD - muito alta, a um custo relativamente baixo

- Tipo de Armazenamento:

HD x Memória Principal

- Capacidade de Armazenamento:
 - HD - muito alta, a um custo relativamente baixo
 - RAM - capacidade limitada, custo relativamente bem mais caro

- Tipo de Armazenamento:

HD x Memória Principal

- Capacidade de Armazenamento:
 - HD - muito alta, a um custo relativamente baixo
 - RAM - capacidade limitada, custo relativamente bem mais caro

- Tipo de Armazenamento:
 - HD - não volátil

HD x Memória Principal

- Capacidade de Armazenamento:
 - HD - muito alta, a um custo relativamente baixo
 - RAM - capacidade limitada, custo relativamente bem mais caro
- Tipo de Armazenamento:
 - HD - não volátil
 - RAM - volátil

Memória Primária x Memória Secundária

Velocidade de Acesso

	Velocidade Típica		Capacidade Típica
	1 ns	Registradores	< 1 KB
Memória volátil (10^{-9} s)	2 ns	Cache	4 MB
	10 ns	Memória Principal	1-64 GB
Memória não volátil (10^{-3} s)	10 ms	Memória Secundária	> 1 TB

HD x SSD

Os SSDs costumam ser “apenas” de 10 a 20 vezes mais rápidos do que os HDs.

SSDs possuem um custo por byte maior (de 2 a 4 vezes).

Breve História sobre os Discos

Em 1956 houve o lançamento das primeiras unidades de disco.

IBM 350 (1956)



Breve História sobre os Discos

Em 1956 houve o lançamento das primeiras unidades de disco.

IBM 350 (1956)

- 170 cm de altura, quase uma tonelada



Breve História sobre os Discos

Em 1956 houve o lançamento das primeiras unidades de disco.

IBM 350 (1956)

- 170 cm de altura, quase uma tonelada
- 3,75 MB de capacidade



Breve História sobre os Discos

Em 1973 a IBM lançou o que é considerado o pai dos HDs modernos.



Breve História sobre os Discos

Em 1973 a IBM lançou o que é considerado o pai dos HDs modernos.

- 10 MB de capacidade



HD - Estrutura

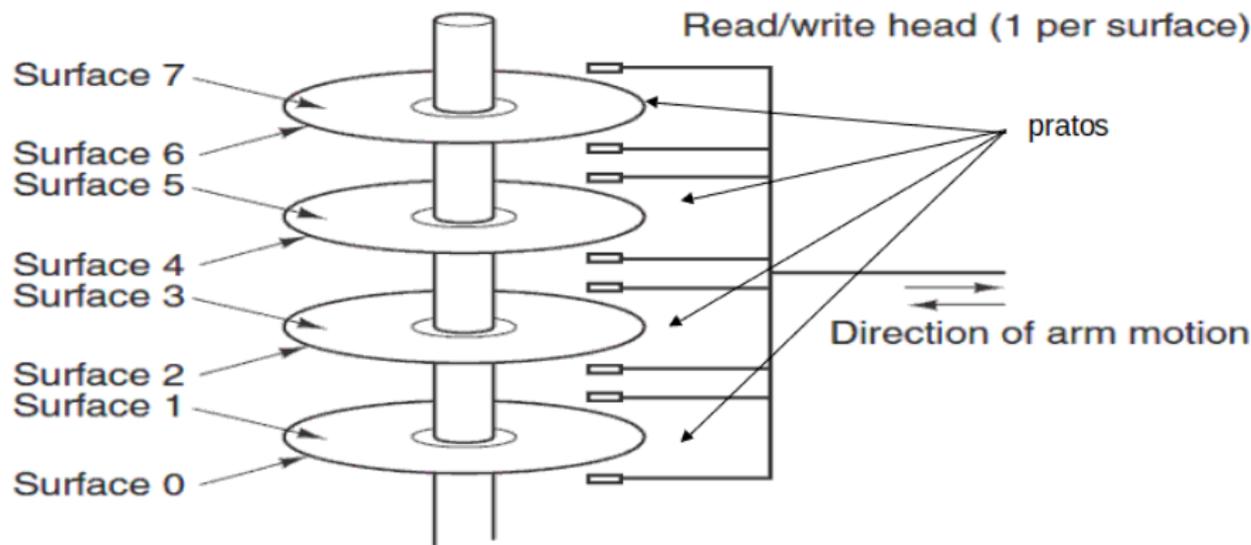
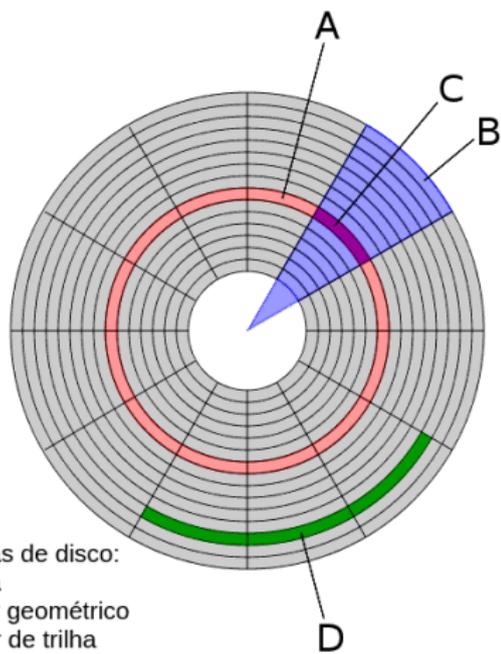


Figure 1-10. Structure of a disk drive.

(TANEMBAUM & BOS, 2015)

Conjunto de pratos, cabeças de leitura e escrita (uma por superfície), braço.

HD - Estrutura



Estruturas de disco:
(A) Trilha
(B) Setor geométrico
(C) Setor de trilha
(D) Unidade de alocação

- Disco: conjunto de pratos empilhados
- Superfícies: organizadas em trilhas
- Trilhas: divididas em setores
- Cilindro: conjunto de trilhas na mesma posição
- Um setor é a menor porção endereçável do disco

Imagem: <https://pt.wikipedia.org/wiki/Ficheiro:Disk-structure2.svg>

HD - Leitura do Disco

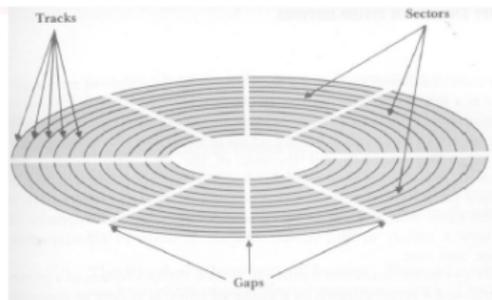
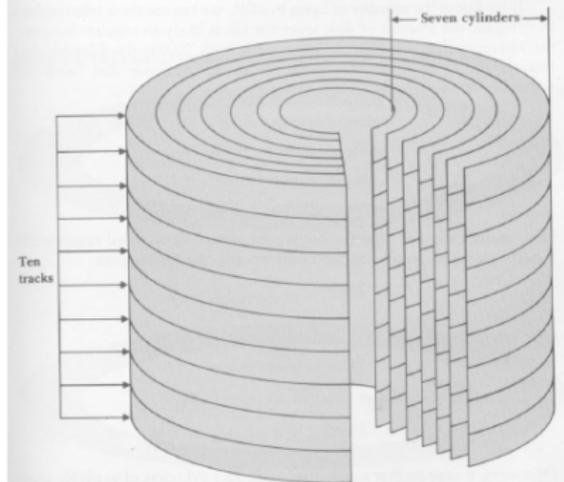


FIGURE 3.3 Schematic illustration of disk drive viewed as a set of seven cylinders.



Como o computador lê alguma informação do HD?

1. Identifica em que setor/trilha/superfície está a informação
2. Movimenta o braço de leitura para o cilindro correto (para poder acessar a trilha correta)
3. Rotaciona o prato para posicionar a cabeça de leitura sobre o setor correto
4. Faz a leitura de um certo número de bytes

HD - Leitura do Disco

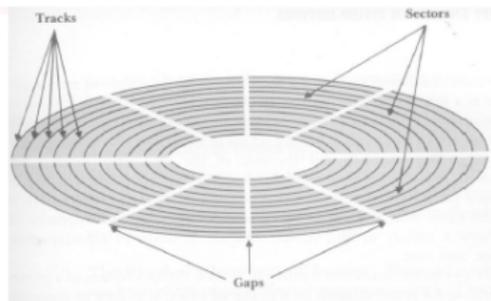
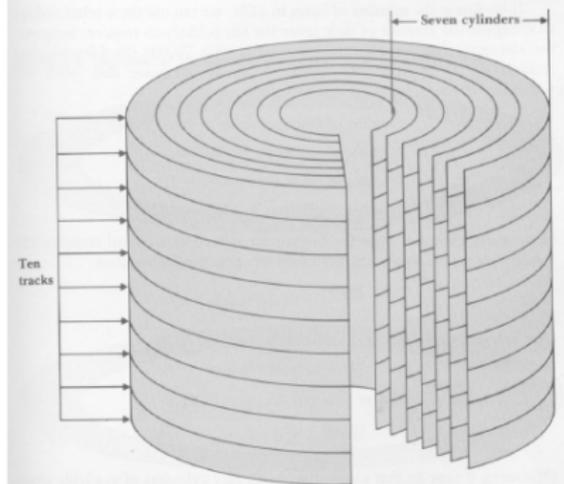


FIGURE 3.3 Schematic illustration of disk drive viewed as a set of seven cylinders.



Como o computador lê alguma informação do HD?

1. Identifica em que setor/trilha/superfície está a informação
2. Movimenta o braço de leitura para o cilindro correto (para poder acessar a trilha correta)
3. Rotaciona o prato para posicionar a cabeça de leitura sobre o setor correto
4. Faz a leitura de um certo número de bytes

Os passos 2 e 3 são mecânicos, conhecidos como *SEEK*.

HD - Seeking

- Movimento de posicionar a cabeça de leitura/escrita sobre a trilha/setor desejado
- O conteúdo de todo um cilindro pode ser lido de uma vez
- É o movimento **mais lento** de toda operação de leitura/escrita
- Esse movimento deve ser minimizado, na medida do possível

Acesso Sequencial x Acesso Aleatório

- Apesar do custo de um *seek*, o acesso é direto (aleatório ou *randômico*), pois não é necessário ler/passar por todos os dados anteriores
- Em contraposição, fitas demandam acesso sequencial, ou seja, é preciso passar por todo o trecho de fita anterior ao que contém o dado desejado

HD - Arquivo

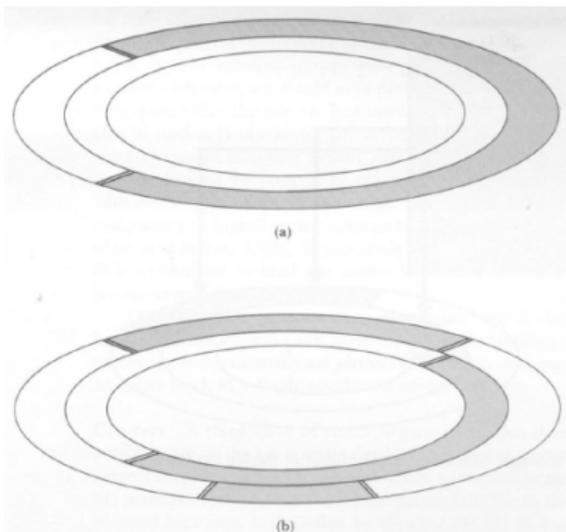


FIGURE 3.6 File extents (shaded area represents space on disk used by a single file).

Os arquivos não são estáticos e podem não estar armazenados de forma contígua pelo disco.

- Seus pedaços podem estar espalhados nos setores (que estarão de alguma forma ligados, mais ou menos como se fosse uma lista ligada)

HD - Arquivo

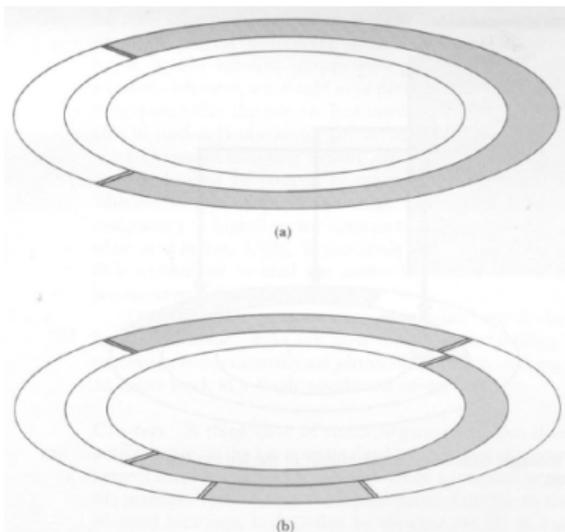


FIGURE 3.6 File extents (shaded area represents space on disk used by a single file).

Os arquivos não são estáticos e podem não estar armazenados de forma contígua pelo disco.

- Seus pedaços podem estar espalhados nos setores (que estarão de alguma forma ligados, mais ou menos como se fosse uma lista ligada)

Assim, o tempo de acesso a uma informação, na prática, depende:

- da distribuição dos dados de um arquivo pelo disco
- da tecnologia do disco

HD - Tempo de Acesso

- Se acesso a disco é caro (em termos de tempo) e queremos escolher estruturas de dados que diminuam o tempo de acesso, precisamos poder calcular (ou estimar) o tempo de acesso de uma forma não muito complicada, pelo menos:
 - independente da distribuição e localização do arquivo em disco
 - independente da tecnologia

HD - Tempo de Acesso

Para simplificar os cálculos, podemos fazer a seguinte aproximação (pior caso):

- considera-se que é necessário um *seek* por “pedaço” de arquivo a ser lido:
 - considerando que eu não preciso ler o arquivo inteiro em um dado instante, pois posso estar interessado em apenas um “pedaço”
 - como há outros processos sendo executados, quando eu quiser ler outro “pedaço” do meu arquivo a cabeça de leitura do disco pode não estar no mesmo lugar onde parou a última leitura desse meu arquivo
- **tempo de acesso total** = número de acessos (*seeks*) * tempo de um acesso

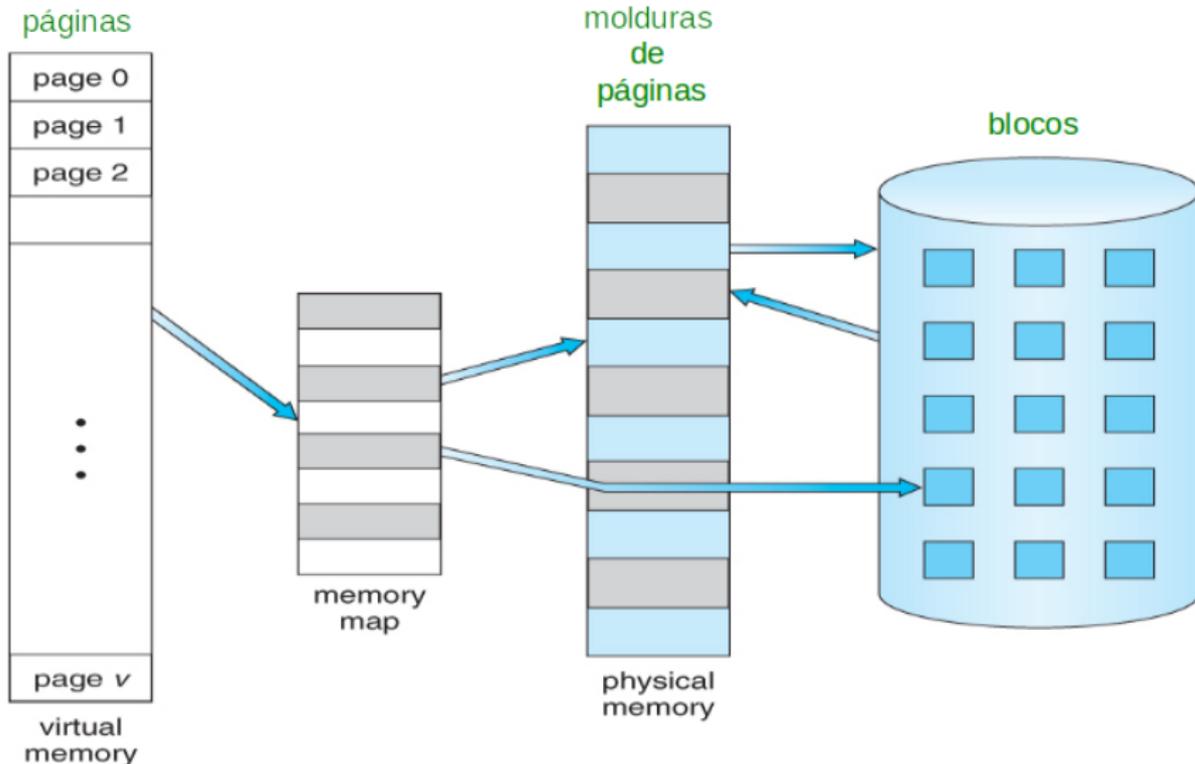
Paginação

- O disco é grande, mas o acesso é lento: fazer várias pequenas leituras no disco tornaria os programas inviáveis ...
- Solução: ler um “pedaço” razoável do disco, trazer para a memória principal, processá-lo lá conforme necessário, se for salvar reescrever o pedaço todo no disco novamente
- Ou seja, tenho um subconjunto do meu disco em memória principal
- O conteúdo desse “pedaço”, contendo i setores, será chamado de “página” e será armazenado fisicamente em um “pedaço” da memória chamado de “moldura de página”

Memória Virtual

- O Sistema Operacional (SO) gerencia se a informação (ou seja, a página que contém informação) já está em memória
 - Se não estiver, precisa carregá-la (em uma moldura de página, i.e., trecho da memória principal destinado a armazenar uma “página” - ou bloco - do disco)
 - Se não tiver moldura disponível, precisa descarregar alguma no disco antes
- Com isso os programas podem endereçar todo o disco como se ele estivesse em memória principal ([memória virtual](#))

HD - Arquivo



https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/9_VirtualMemory.html

Organização de Arquivos

- **Bloco**: unidade de transferência de dados entre memória principal e a memória secundária
 - Do ponto de vista do disco (*hardware*), o **bloco físico** é um setor (ex: 512 bytes); já o SO (*software*) define um **bloco lógico**, que corresponde a uma página, com tamanho igual a um ou mais setores (usualmente 4kb), que também é usado pelos gerenciadores de bancos de dados.
 - Usaremos o termo bloco de forma genérica como bloco lógico, a não ser quando especificado (isto é, o bloco tem o tamanho de uma página).

- **Cabeçalho do arquivo (descriptor)** pode conter informações como:
 - Descrição dos formatos dos campos de um registro
 - Códigos de tipos de registros para registros de tamanho variável
 - Primeiro e último bloco
 - Informações para determinar os endereços dos seus blocos
- Abrir um arquivo significa trazer para a memória o cabeçalho do arquivo (blocos contendo essas informações que ficarão em memória até o arquivo ser fechado)

Organização de Arquivos

- Pensando em arquivos de dados, é comum considerar que:
 - Nenhum registro é maior que um bloco
- Se R (tamanho fixo do registro, para simplificar) e B (tamanho do bloco) e $R \leq B$:
 - fator de blocagem $fb = \lfloor \frac{B}{R} \rfloor =$ número de registros inteiros que cabem em um bloco

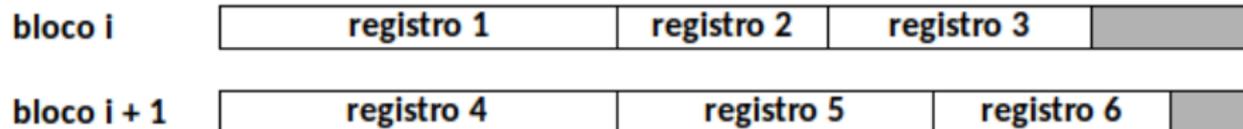
Organização de Arquivos

- **Organização espalhada** : os blocos são totalmente preenchidos; se um registro não cabe inteiramente na parte vazia do bloco, coloca o que couber e um ponteiro para o próximo bloco.



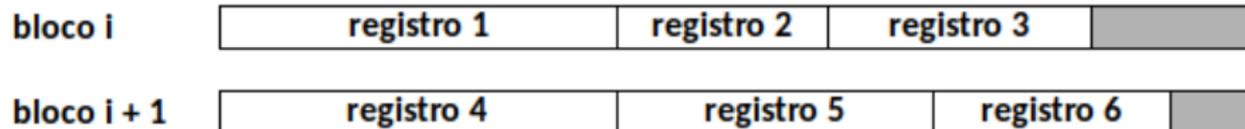
Organização de Arquivos

- **Organização não espalhada** : registros não podem ser divididos. Cada bloco pode conter até fb registros.



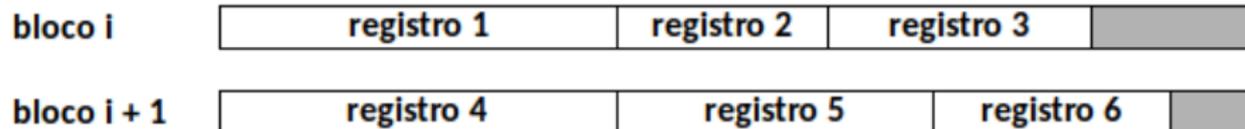
Organização de Arquivos

- **Organização não espalhada** : registros não podem ser divididos. Cada bloco pode conter até fb registros.
 - Se os registros tiverem tamanho fixo $= R$, blocos de tamanho B e taxa de blocagem $= fb$:
Perda de espaço em cada bloco: $B - (fb * R)$



Organização de Arquivos

- **Organização não espalhada** : registros não podem ser divididos. Cada bloco pode conter até fb registros.
 - Se os registros tiverem tamanho fixo = R , blocos de tamanho B e taxa de blocagem = fb :
Perda de espaço em cada bloco: $B - (fb \cdot R)$



- Organização **muito utilizada**, pois facilita a localização de registros.

Alocação de Blocos na Memória Secundária

- Leitura, escrita, buscas, etc., são realizadas por blocos.

¹ Complexidade em termos de número de *seeks*

Alocação de Blocos na Memória Secundária

- Leitura, escrita, buscas, etc., são realizadas por blocos.
- Os arquivos não são estáticos, eles crescem e diminuem

¹ Complexidade em termos de número de *seeks*

Alocação de Blocos na Memória Secundária

- Leitura, escrita, buscas, etc., são realizadas por blocos.
- Os arquivos não são estáticos, eles crescem e diminuem
- Estratégias de alocação de blocos no disco e organização de registros pelos blocos devem considerar esse fato:

¹ Complexidade em termos de número de *seeks*

Alocação de Blocos na Memória Secundária

- Leitura, escrita, buscas, etc., são realizadas por blocos.
- Os arquivos não são estáticos, eles crescem e diminuem
- Estratégias de alocação de blocos no disco e organização de registros pelos blocos devem considerar esse fato:
 - Sequencial não ordenado (*heap files*)
 - Sequencial ordenado (*sorted files*)
 - Por listas ligadas
 - Indexado
 - Árvores B / B+
 - *Hashing*

¹ Complexidade em termos de número de *seeks*

Alocação de Blocos na Memória Secundária

- Leitura, escrita, buscas, etc., são realizadas por blocos.
- Os arquivos não são estáticos, eles crescem e diminuem
- Estratégias de alocação de blocos no disco e organização de registros pelos blocos devem considerar esse fato:
 - Sequencial não ordenado (*heap files*)
 - Sequencial ordenado (*sorted files*)
 - Por listas ligadas
 - Indexado
 - Árvores B / B+
 - *Hashing*
- Para cada estratégia analisaremos a complexidade¹.

¹ Complexidade em termos de número de *seeks*

Referência

- Slides baseados no material da profa. Ariane Machado Lima - ACH2024
- Slides da Profa. Graça (ICMC) - [http://wiki.icmc.usp.br/index.php/SCC-203_\(gracan\)](http://wiki.icmc.usp.br/index.php/SCC-203_(gracan)) (Arquivos 8, 9 e 12)
- GOODRICH et al, Data Structures and Algorithms in C++. Ed. John Wiley & Sons, Inc. 2nd ed. 2011. Seção 14.2
- ELMARIS, R.; NAVATHE, S. B. Fundamentals of Database Systems. 4 ed. Ed. Pearson-Addison Wesley. Cap 13 (até a seção 13.7).
- TANEMBAUM, A. S. & BOS, H. Modern Operating Systems. Pearson, 4th ed. 2015

Algoritmos e Estruturas de Dados II

Aula 18 – Arquivos - Acesso à Memória Secundária

Prof. Luciano A. Digiampietri
digiampietri@usp.br
@digiampietri

2025