

Algoritmos e Estruturas de Dados II

Aula 20 – Alocação Ligada

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

2025

Alocação de Blocos na Memória Secundária

- Organização Interna de Arquivos
- Acesso à Memória por Blocos (*seeks*)
- 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 (ordenação pelas chaves)
 - Indexada
 - Árvores B / B+
 - *Hashing*
- Para cada estratégia analisaremos complexidadeⁱ.

ⁱ Complexidade em termos de número de *seeks*

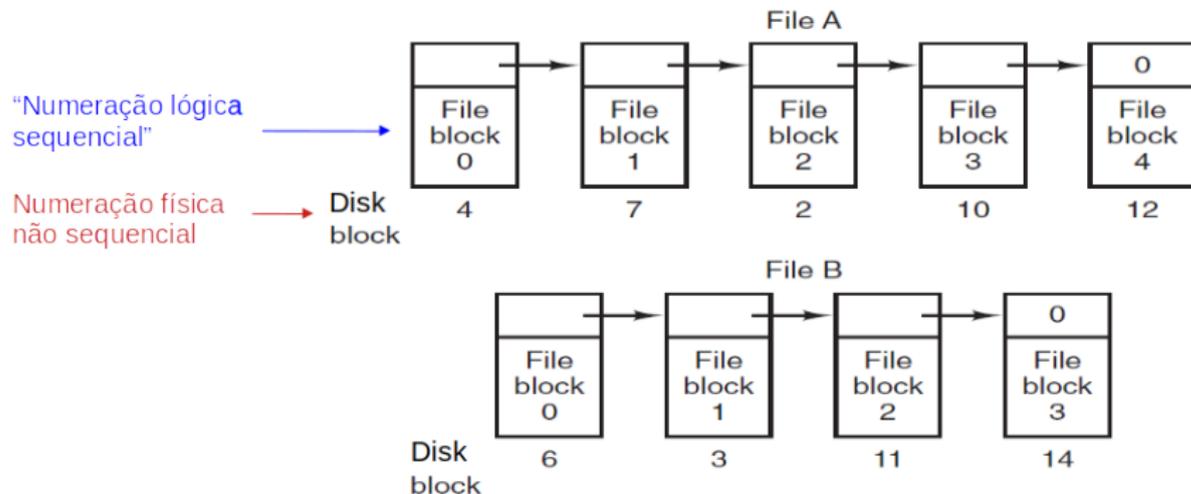
Alocação de Blocos na Memória Secundária

- Organização Interna de Arquivos
- Acesso à Memória por Blocos (*seeks*)
- 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 (ordenação pelas chaves)
 - Indexada
 - Árvores B / B+
 - *Hashing*
- Para cada estratégia analisaremos complexidadeⁱ.

ⁱ Complexidade em termos de número de *seeks*

Alocação por Lista Ligada

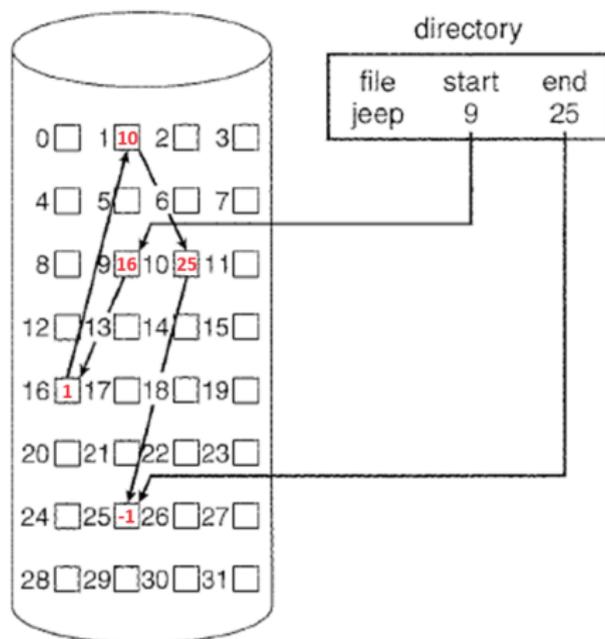
- Cada arquivo é uma lista ligada de blocos



(TANEMBAUM, 2015)

Alocação por Lista Ligada

- Cada arquivo é uma lista ligada de blocos



(SILBERCHATZ et al, 2009)

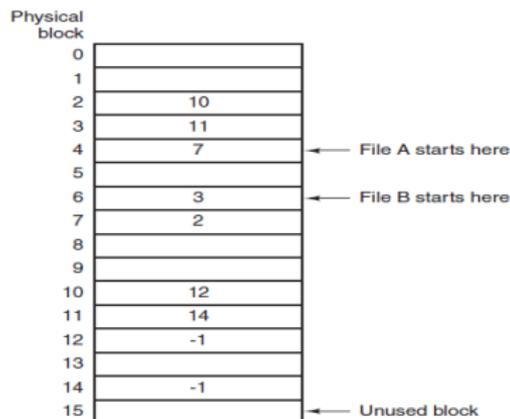
Alocação por Lista Ligada

- **Uso de espaço:**
 - **Vantagem:** resolve o problema de fragmentação externa
 - **Desvantagem:** perda de espaço com os ponteiros
- **Leitura sequencial:** $O(b)$
- **Leitura aleatória / Busca:** $O(b)$
- **Inserção:** $O(1)$ – assumindo que sei onde inserir
- **Remoção:** $O(1)$ para remoção de fato, assumindo que sei onde remover, senão $O(b)$ para buscar
- **Modificação:** $O(1)$ em campo não chave, $O(b)$ se for no campo chave (encontrar o local para a inserção)

Alocação por Lista Ligada

Alocação por listas ligadas com uso de uma *File Allocation Table* (FAT) em memória

- $t[i]$ armazena a posição do próximo bloco



(TANEMBAUM, 2015)

Alocação por Lista Ligada

File Allocation Table (FAT)

directory entry

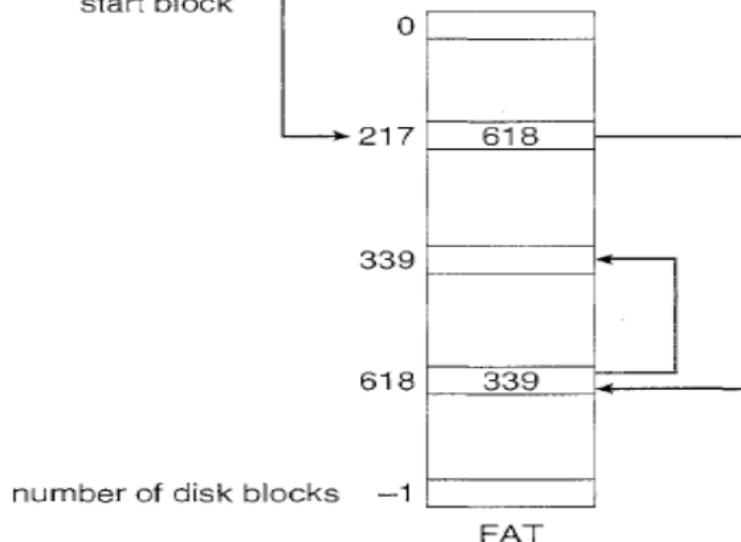
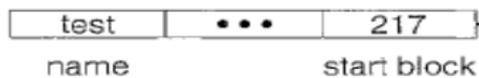


Figure 11.7 File-allocation table.

Alocação por Lista Ligada

Alocação por listas ligadas com uso de uma *File Allocation Table* (FAT) em memória

- **Vantagens:**

- Economiza espaço nos blocos de dados (que só conterão dados e não ponteiros) \rightarrow preciso de menos blocos \rightarrow b menor impacta nas velocidades dependentes de b
- Para uma leitura aleatória (dado um deslocamento em relação ao início do arquivo), o encadeamento (para achar o bloco certo) é seguido apenas sobre a tabela (que está toda em memória) $\rightarrow O(1)$

Alocação por Lista Ligada

Alocação por listas ligadas com uso de uma *File Allocation Table* (FAT) em memória

- **Vantagens:**

- Economiza espaço nos blocos de dados (que só conterão dados e não ponteiros) \rightarrow preciso de menos blocos \rightarrow b menor impacta nas velocidades dependentes de b
- Para uma leitura aleatória (dado um deslocamento em relação ao início do arquivo), o encadeamento (para achar o bloco certo) é seguido apenas sobre a tabela (que está toda em memória) $\rightarrow O(1)$

Daria para fazer Busca Binária?

Alocação por Lista Ligada

Alocação por listas ligadas com uso de uma *File Allocation Table* (FAT) em memória

- **Desvantagens:**
 - Não escalável para grandes discos
Ex: para um disco de 1TB e blocos de 1KB, a tabela ocuparia 4GB de memória (se cada endereço ocupar 4B [32 bits])
- Sistema de arquivos FAT32, por exemplo, impõe tamanho máximo de disco de 2TB

Alocação Sequencial

Complex.	Sequencial		Ligada
	Não Ordenado	Ordenado	Ordenado
Busca	$O(b)$	$O(\log(b))$	$O(b)$
Inserção	$O(1)$ se houver espaço, ou $O(b)$	$O(b)$	$O(1)$ ou $O(b)$ (custo da busca)
Remoção ⁱⁱ	$O(b)$ (custo da busca), ou $O(1)$	$O(\log(b))$ ⁱⁱ	$O(b)$ (custo da busca), ou $O(1)$
Leitura Ordenada	$\omega(b)$ ⁱⁱⁱ	$O(b)$, $O(1)$ (leitura toda de uma vez)	$O(b)$
Mínimo/Máximo	$O(b)$, ou $O(1)$ se as chaves estiverem no cabeçalho	$O(1)$	$O(1)$
Modificação	$O(1)$, ou $O(b)$ se precisar buscar	$O(1)$, $O(\log(n))$ (buscar), ou $O(b)$ (modificação de chave)	$O(1)$, ou $O(b)$ se precisar buscar ou modificar a chave

ⁱⁱConsiderando o uso de um bit de validade

ⁱⁱⁱDepende do algoritmo de ordenação [externo]

Referência

- Slides baseados no material da profa. Ariane Machado Lima - ACH2024
- ELMASRI, R.; NAVATHE, S. B. Fundamentals of Database Systems. 4 ed. Ed. Pearson-Addison Wesley. Cap 13 (até a seção 13.7), 2004
- 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 20 – Alocação Ligada

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

2025