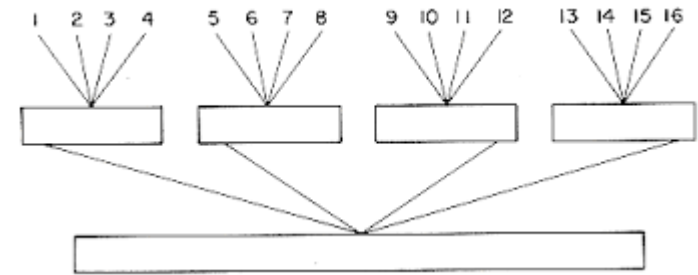


Pesquisa e Ordenação de Dados

Unidade 4.1:

Ordenação Externa



Ordenação Externa

- Podemos classificar os métodos de ordenação em dois grandes grupos:
 - Métodos de ordenação interna
 - Aqueles nos quais a lista de registros pode ser mantida inteiramente na memória principal durante a ordenação

Ordenação Externa

- Podemos classificar os métodos de ordenação em dois grandes grupos:
 - Métodos de ordenação interna
 - Aqueles nos quais a lista de registros pode ser mantida inteiramente na memória principal durante a ordenação
 - Métodos de ordenação externa
 - Aqueles nos quais há mais registros a serem classificados do que é possível manter na memória principal em qualquer momento



Ordenação Externa

- Fatores que determinam as diferenças das técnicas de ordenação externa, comparadas à ordenação interna:
 - O custo para acessar um item é algumas ordens de grandeza maior. Este custo se refere principalmente à transferência de dados entre a memória interna e externa;
 - Restrições relacionadas ao método de acesso aos dados (fitas e discos);
 - Os métodos de ordenação externa são dependentes do estado atual da tecnologia. A variedade de tipos de unidades de memória externa torna os métodos dependentes de vários parâmetros que afetam seu desempenho.

Ordenação Externa

- Fatores que determinam as diferenças das técnicas de ordenação externa, comparadas à ordenação interna:
 - O custo para acessar um item é algumas ordens de grandeza maior.

Este c
memó

– Restri
disco

– Os m
tecnol

os metodos dependentes de varios parametros que afetam seu desempenho.

	L3	DRAM	SSD	HDD
Read Latency	~20 ns	60 ns	25,000 ns	10,000,000 ns
Write Latency	~20 ns	60 ns	300,000 ns	10,000,000 ns

Intercalações balanceada de N caminhos

- Algoritmo que ordena os registros por meio da intercalação de N caminhos ou fontes de dados (arquivo ou fita), todos com tamanhos semelhantes
 - Feita com vários arquivos temporários (caminhos) em um mesmo disco
- Etapas:
 - 1) Classificação das partições
 - distribuição de blocos de registros ordenados por N caminhos balanceados
 - 2) Intercalação
 - intercalações sucessivas dos N caminhos

Estratégia Geral

1) Classificação das partições

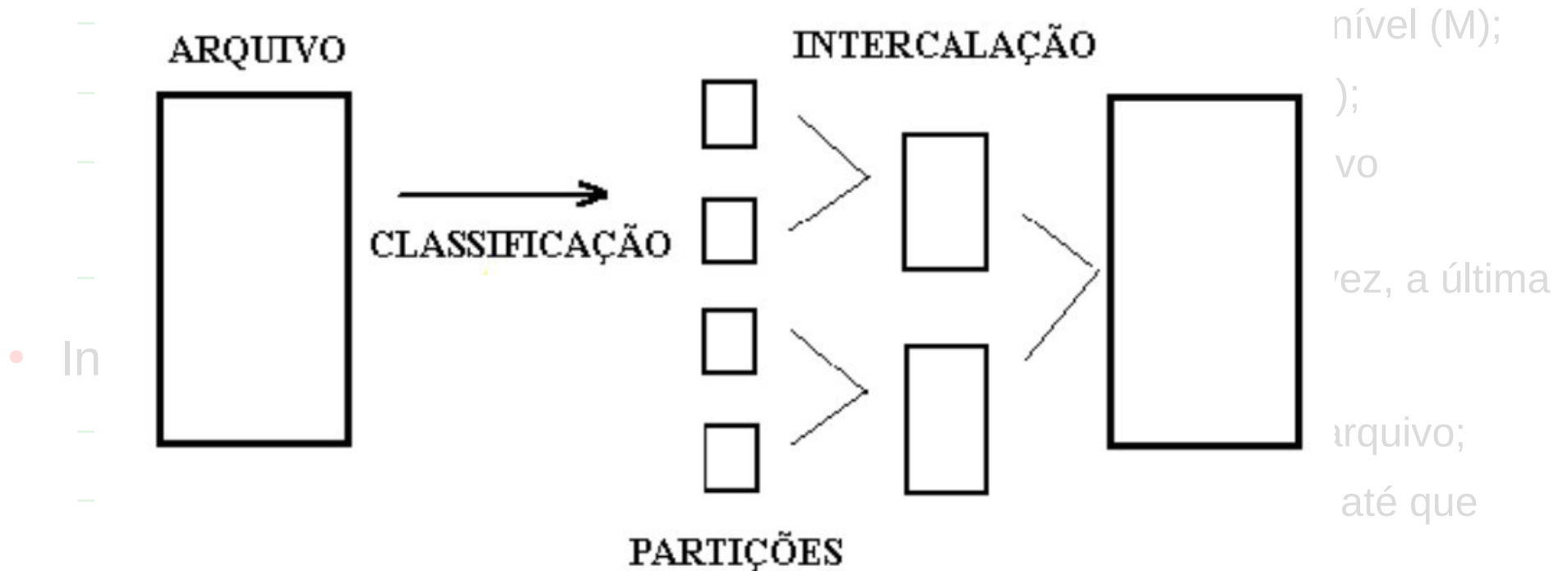
- Dividir o arquivo em blocos do tamanho da memória interna disponível (M);
- Ordenar cada bloco na memória interna (quicksort, mergesort, etc);
- Gravar os registros ordenados em uma partição nova de um arquivo existente (ou em um novo arquivo);
- Todas as partições classificadas conterão M registros, exceto, talvez, a última.

2) Intercalação

- Intercale os blocos ordenados, fazendo várias passadas sobre o arquivo;
- A cada passada são criados blocos ordenados cada vez maiores, até que todo o arquivo seja ordenado.

Estratégia Geral

- Classificação das particões



Exemplo

- Considere o seguinte arquivo com 22 registros:

I N T E R C A L A C A O B A L A N C E A D A

- Objetivo:
 - ordenar os 22 registros e colocá-los num arquivo de saída;
- Os registros são lidos sequencialmente, um após o outro;
- A memória interna tem capacidade para apenas **três** registros;
- Podemos manter abertos **6** arquivos de saída (**F**)
- Metade dos arquivos (**F/2**) será usada para leitura (entrada) → será uma intercalação de **3 caminhos**;
- A outra metade (**F/2**), para escrita (saída);

Exemplo

- **Etapa 1:** classificação das partições

I N T E R C A L A C A O B A L A N C E A D A

arq 1: I N T A C O A D E

arq 2: C E R A B L A

arq 3: A A L A C N

arquivos de entrada

Exemplo

- **Etapa 2:** intercalação dos blocos
 - O primeiro registro de cada arquivo de saída é lido e trazido para a memória interna;
 - O registro com a menor chave é então retirado e gravado em um novo arquivo de saída, sendo em seguida substituído na memória interna pelo próximo registro lido de sua respectiva fita; Repete-se o processo;
 - Quando o terceiro registro de um dos blocos é lido, aquele arquivo fica inativo até que o terceiro registro dos demais arquivos também tenham sido lidos e escritos no arquivo de saída, formando um novo bloco de 9 registros;
 - Repete-se o processo para o próximo conjunto de blocos, intercalando-os em um novo arquivo de saída.
 - No final de cada passada, o conjunto de partições de saída torna-se o conjunto de entrada para a próxima fase.

Exemplo

- Etapa 2: intercalação dos blocos
 - O primeiro registro de cada arquivo de saída é lido e trazido para a memória interna;
 - O registro com a menor chave é então retirado e gravado em um novo arquivo de saída, sendo em seguida substituído na memória interna pelo próximo registro lido de sua respectiva fita; Repete-se o processo;
 - Quando o terceiro registro de um dos blocos é lido, aquele arquivo fica inativo até que o terceiro registro dos demais arquivos também tenham sido lidos e escritos no arquivo de saída, formando um novo bloco de 9 registros;
 - Repete-se o processo para o próximo conjunto de blocos, intercalando-os em um novo arquivo de saída.
 - No final de cada passada, o conjunto de partições de saída torna-se o conjunto de entrada para a próxima fase.

Exemplo

- **Etapa 2:** intercalação dos blocos
 - Primeira passada:

arq 1: **I**NT ACO ADE

arq 2: **C**ER ABL A

arq 3: **A**AL ACN

} arquivos de entrada

arq 4: **A**ACEILNRT

} arquivos de saída

Exemplo

- **Etapa 2:** intercalação dos blocos
 - Primeira passada:

arq 1: INT ACO ADE

arq 2: CER ABL A

arq 3: AAL ACN

} arquivos de entrada

arq 4: AA**CEIL**NR**T**

} arquivos de saída

Exemplo

- **Etapa 2:** intercalação dos blocos
 - Primeira passada:

arq 1: INT ACO ADE

arq 2: CER ABL A

arq 3: AAL ACN

} arquivos de entrada

arq 4: AACEILNRT

} arquivos de saída

Exemplo

- **Etapa 2:** intercalação dos blocos
 - Primeira passada:

arq 1: I N T A C O A D E

arq 2: C E R A B L A

arq 3: A A L A C N

} arquivos de entrada

arq 4: A A C E I L N R T

} arquivos de saída

Exemplo

- **Etapa 2:** intercalação dos blocos
 - Primeira passada:

arq 1: INT ACO ADE

arq 2: CERR ABL A

arq 3: AALL ACN

} arquivos de entrada

arq 4: AACERILNRT

} arquivos de saída

Exemplo

- **Etapa 2:** intercalação dos blocos
 - Primeira passada:

arq 1: I N T A C O A D E

arq 2: C E R A B L A

arq 3: A A L A C N

} arquivos de entrada

arq 4: A A C E I L N R T

} arquivos de saída

Exemplo

- **Etapa 2:** intercalação dos blocos
 - Primeira passada:

arq 1: I N T A C O A D E

arq 2: C E R A B L A

arq 3: A A L A C N

} arquivos de entrada

arq 4: A A C E I L N R T

} arquivos de saída

Exemplo

- **Etapa 2:** intercalação dos blocos
 - Primeira passada:

arq 1: I N T A C O A D E

arq 2: C E R A B L A

arq 3: A A L A C N

} arquivos de entrada

arq 4: A A C E I L N R T

} arquivos de saída

Exemplo

- **Etapa 2:** intercalação dos blocos
 - Primeira passada:

arq 1: INT ACO ADE

arq 2: CER ABL A

arq 3: AAL ACN

} arquivos de entrada

arq 4: ACEILNRT

} arquivos de saída

Exemplo

- **Etapa 2:** intercalação dos blocos
 - Primeira passada:

arq 1: INT ACO ADE

arq 2: CER ABL A

arq 3: AAL ACN

} arquivos de entrada

arq 4: ACEILNRT

arq 5: AAABCC LNO

} arquivos de saída

Exemplo

- **Etapa 2:** intercalação dos blocos
 - Primeira passada:

arq 1: INT ACO ADE

arq 2: CER ABL A

arq 3: AAL ACN

} arquivos de entrada

arq 4: AA CEILNRT

arq 5: AA ABCCLNO

arq 6: AADE

} arquivos de saída

Exemplo

- **Etapa 2:** intercalação dos blocos
 - Segunda passada:

arq 1: A A A A A A A B C C C D E E I L L N N O R T
arq 2:
arq 3:

} arquivo de saída

arq 4: A A C E I L N R T
arq 5: A A A B C C L N O
arq 6: A A D E

} arquivos de entrada

Exemplo

- Etapa 2: intercalação dos blocos

- Segunda passada:

Reciclamos o Arq1

arq 1: A A A A A A B C C C D E E I L L N N O R T

arq 2:

arq 3:

arquivo de saída

arq 4: A A C E I L N R T

arq 5: A A A B C C L N O

arq 6: A A D E

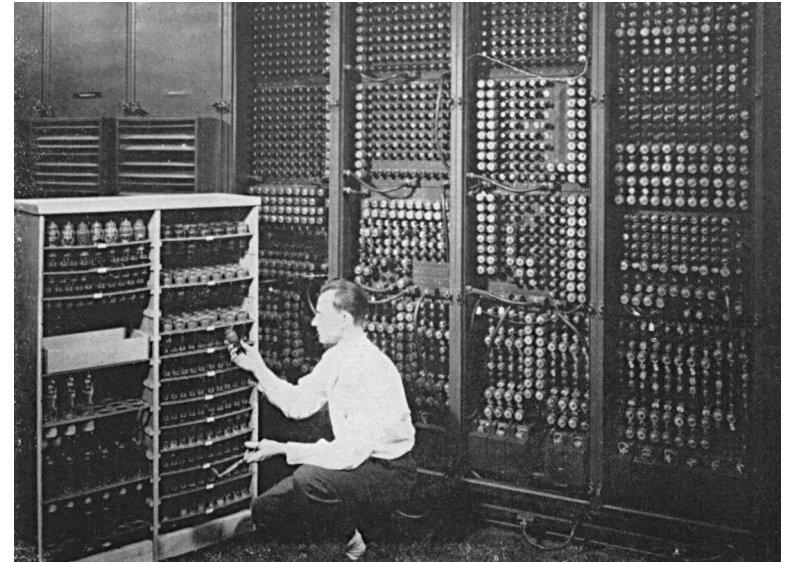
arquivos de entrada

Exemplo 2

- Vamos ordenar o seguinte vetor

30	21	43	3	9	82	15
----	----	----	---	---	----	----

- Com a seguinte máquina
 - Não temos memória para tantos itens



Exemplo 2

- Vamos ordenar o seguinte vetor:

30	21	43	3	9	82	15
----	----	----	---	---	----	----

Arq 1: 21 30 43

Arq 2: 3 9 82

Arq 3: 15

} arquivos de entrada

Arq 4: 3 9 15 21 30 43 82

} arquivo de saída

Exemplo 2

- Vamos ordenar o seguinte vetor:

30	21	43	3	9	82	15
----	----	----	---	---	----	----

Arq 1: 21 30 43

Arq 2: 3 9 82

Arq 3: 15

} arquivos de entrada

Arq 4: 3 9 15 21 30 43 82

} arquivo de saída

Exemplo 2

- Vamos ordenar o seguinte vetor:

30	21	43	3	9	82	15
----	----	----	---	---	----	----

Arq 1: 21 30 43

Arq 2: 3 9 82

Arq 3: 15

} arquivos de entrada

Arq 4: 3 9 15 21 30 43 82

} arquivo de saída

Exemplo 2

- Vamos ordenar o seguinte vetor:

30	21	43	3	9	82	15
----	----	----	---	---	----	----

Arq 1: 21 30 43

Arq 2: 3 9 82

Arq 3: 15

} arquivos de entrada

Arq 4: 3 9 15 21 30 43 82

} arquivo de saída

Análise

- A base da análise é o número de “passadas” pelos registros (passar todos os registros do arquivo fazendo distribuição ou intercalação)
- O tempo de ordenação em memória principal é pouco significativo
- Estimativa: $passadas = \log_m \left(\frac{N}{b} \right)$
 - Em que
 - N = número total de registros
 - b = tamanho do bloco ordenado em memória principal
 - m = quantidade de fontes usadas na intercalação (qtde de arq temporários; caminhos)
- Importante para estimar quantos arquivos temporários serão utilizados, qual o melhor tamanho de bloco, etc, a fim de minimizar o custo da operação