



CENTRO DE CIÊNCIA E TECNOLOGIA
LABORATÓRIO DE CIÊNCIAS MATEMÁTICAS
UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE

Ordenação Externa: Ordenação por Intercalação

Disciplina: Estrutura de Dados II

Prof. Fermín Alfredo Tang Montané

Curso: Ciência da Computação

Ordenação Interna vs Ordenação Externa

O princípio de Intercalação

- Todos os algoritmos estudados previamente são algoritmos de **ordenação interna**. O que significa que todos os dados precisam estar em memória principal durante o processo de ordenação.
- A **ordenação externa** é um tipo de ordenação, que lida com grandes volumes de dados e que permite que porções desses dados sejam armazenados na memória secundária durante o processo de ordenação.
- O princípio base para realizar ordenação externa é o de intercalação (***merging***). A ordenação externa começa ordenando blocos de dados na memória principal e escrevendo esses blocos ordenados em arquivos na memória secundária. Uma vez que todos os dados passam por essa fase de ordenação interna, inicia-se o processo de construir um único arquivo ordenado mediante a combinação de arquivos seguindo o princípio de intercalação.

Ordenação Interna

O princípio de Intercalação

- O princípio de intercalação pode também ser aplicado na ordenação interna. Neste contexto, o algoritmo recebe o nome de **MergeSort**.
- Este método de ordenação interna será tratado separadamente em outra aula.

Ordenação Externa

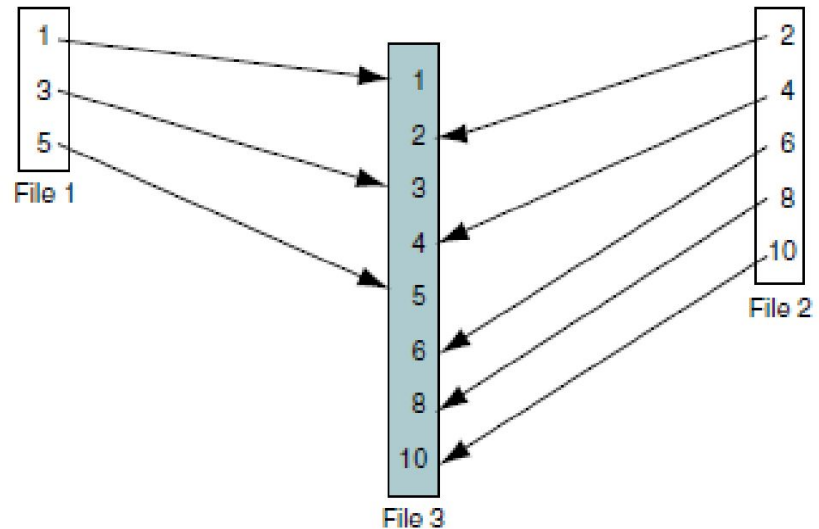
O princípio de Intercalação

- O conceito de intercalação (*merge*) pode ser definido como a combinação de dois blocos de dados ordenados em um único bloco de dados ordenado.

Ordenação Externa

Intercalação de arquivos

- No contexto de ordenação de arquivos. A intercalação (*merge*) é um processo que combina dois arquivos ordenados, com base em um certo campo chave, em um único arquivo ordenado no mesmo campo chave.
- No exemplo, os arquivos ordenados *File1* e *File2* são combinados (*merged*) no arquivo *File3*.
- Para combinar os arquivos, compara-se o primeiro registro do *File1* com o primeiro registro do *File2* e escreve-se o menor, o 1, no *File3*.
- Depois, compara-se o segundo registro do *File1* e o primeiro do *File2* e escreve-se o menor, o 2, no *File3*.
- O processo continua até que todos os dados sejam escritos em ordem no arquivo *File3*.



Ordenação Externa

Intercalação de arquivos

- O algoritmo descreve o processo de intercalação de arquivos.

Algorithm mergeFiles

Merge two sorted files into one file.

Pre input files are sorted

Post input files sequentially combined in output file

```
1 open files
2 read (file1 into record1)
3 read (file2 into record2)
4 loop (not end file1 OR not end file2)
  1 if (record1 key <= record2 key)
    1 write (record1 to file3)
    2 read (file1 into record1)
    3 if (end of file1)
      1 set record1 key to infinity
    4 end if
  2 else
    1 write (record2 to file3)
    2 read (file2 into record2)
    3 if (end of file2)
      1 set record2 key to infinity
    4 end if
  3 end if
5 end loop
6 close files
end mergeFiles
```

Lê o próximo registro do arquivo

Enquanto não atingir o fim de ambos arquivos

Compara os registros

Escreve o menor registro no File3

Lê o próximo registro do arquivo

Se o arquivo chega ao fim

Ordenação Externa

Definição

- A **ordenação externa** trata o caso em que é preciso ordenar um grande volume de dados os quais se encontram armazenados em memória secundária.
- Trata-se de grandes arquivos que não podem ser copiados de uma vez na memória principal. Apenas, blocos menores desses arquivos são copiados na memória principal, ordenados e copiados de volta na memória secundária em arquivos auxiliares.
- Esta primeira etapa na qual todos os dados são lidos por blocos, ordenados e copiados de volta em arquivos auxiliares é chamada de etapa de ordenação.
- Uma segunda etapa consiste em combinar os arquivos auxiliares em um único arquivo ordenado usando o **princípio de intercalação (Merge)**.

Ordenação Externa

Fase de Ordenação

- Na **fase de ordenação** define-se a quantidade de registros que podem ser armazenados por vez na memória principal.
- Depois inicia-se a leitura de um bloco de registros de tamanho apropriado, por vez. Realiza-se a ordenação destes registros e escreve-se o bloco de registros ordenado, novamente na memória secundária, em arquivos auxiliares.

Os registros ordenados por blocos poderiam ser salvos em um único arquivo como mostrado na figura.

1
2
3
4
5
14
16
17
6
7
9
13
23
25
29
8
11
19
26
27

File 3

No entanto, para poder aplicar o princípio de intercalação, é necessário salvar em pelo menos dois arquivos.

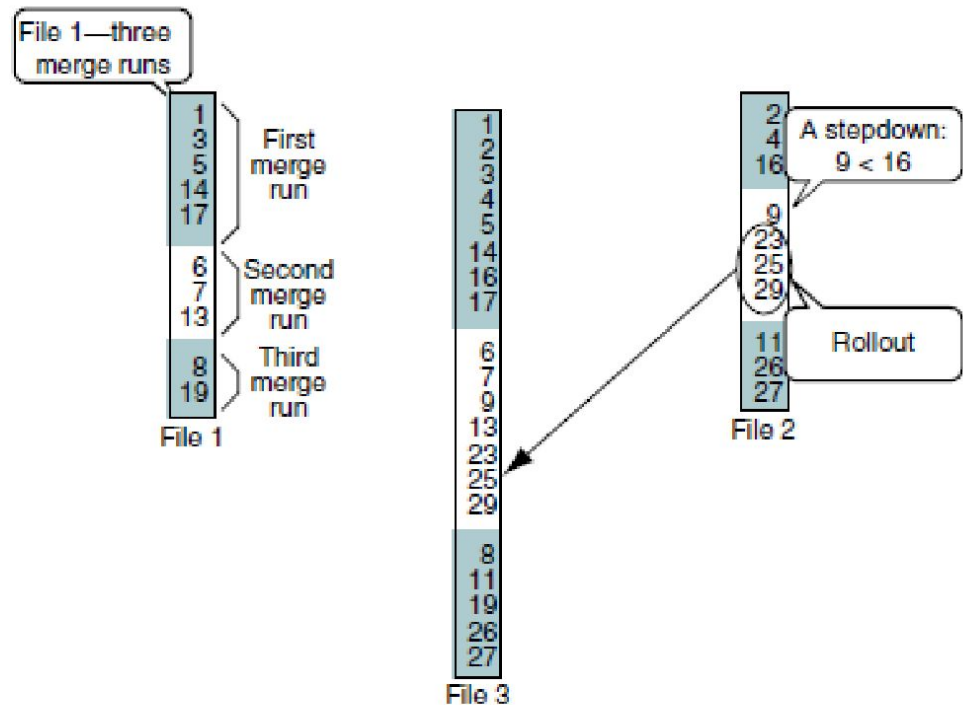
1
2
3
4
5
14
16
17
8
11
19
26
27

6
7
9
13
23
25
29

Ordenação Externa

Intercalação de arquivos

- Na ordenação externa, após a fase de ordenação, são gerados arquivos de saída formados por vários blocos de dados ordenados ou *merge runs*.
- Entre cada bloco de dados ordenados existe uma quebra de sequência chamada de *stepdown*.



Merging Files Example

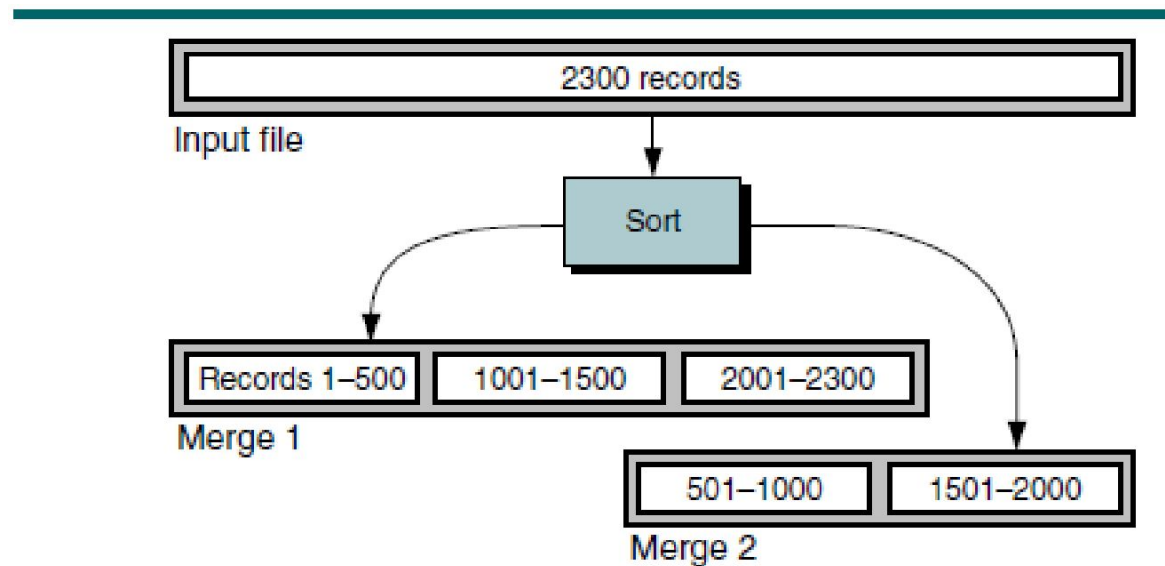
Ordenação Externa

Fase de Ordenação - Exemplo

- A figura ilustra um exemplo, com 2300 arquivos. Onde assume-se que o tamanho do registro e a memória disponível somente permitem que 500 registros sejam lidos e ordenados de cada vez.
- Uma vez ordenados, os registros são armazenados em dois arquivos auxiliares.

O primeiro bloco de registros é armazenado em no primeiro arquivo. Enquanto o segundo bloco de registros é armazenado no segundo arquivo.

O processo continua ordenando os blocos e alternando o arquivo de destino até esgotar os registros.



Sort Phase in an External Sort

Ordenação Externa

Fase de Intercalação (*Merge Phase*)

- Após a fase de ordenação da ordenação externa segue a **fase de intercalação**.
- Os arquivos auxiliares, compostos por vários blocos de registros ordenados, são combinados pelo princípio de intercalação.
- Embora o objetivo final seja produzir um único arquivo ordenado contendo todos os registros, o processo de intercalação é realizado em várias etapas.
- Em cada etapa, os blocos ordenados nos arquivos auxiliares são lidos e combinados formando blocos ordenados maiores que são salvos em novos arquivos de saída.
 - Para seguir aplicando o princípio de intercalação, é necessário salvar em pelo menos dois arquivos de saída.
- Para executar esta etapa foram propostos diferentes conceitos de intercalação. Entre os mais representativos temos:
 - Intercalação natural; Intercalação Balanceada e Intercalação Polifásica.

Ordenação Externa

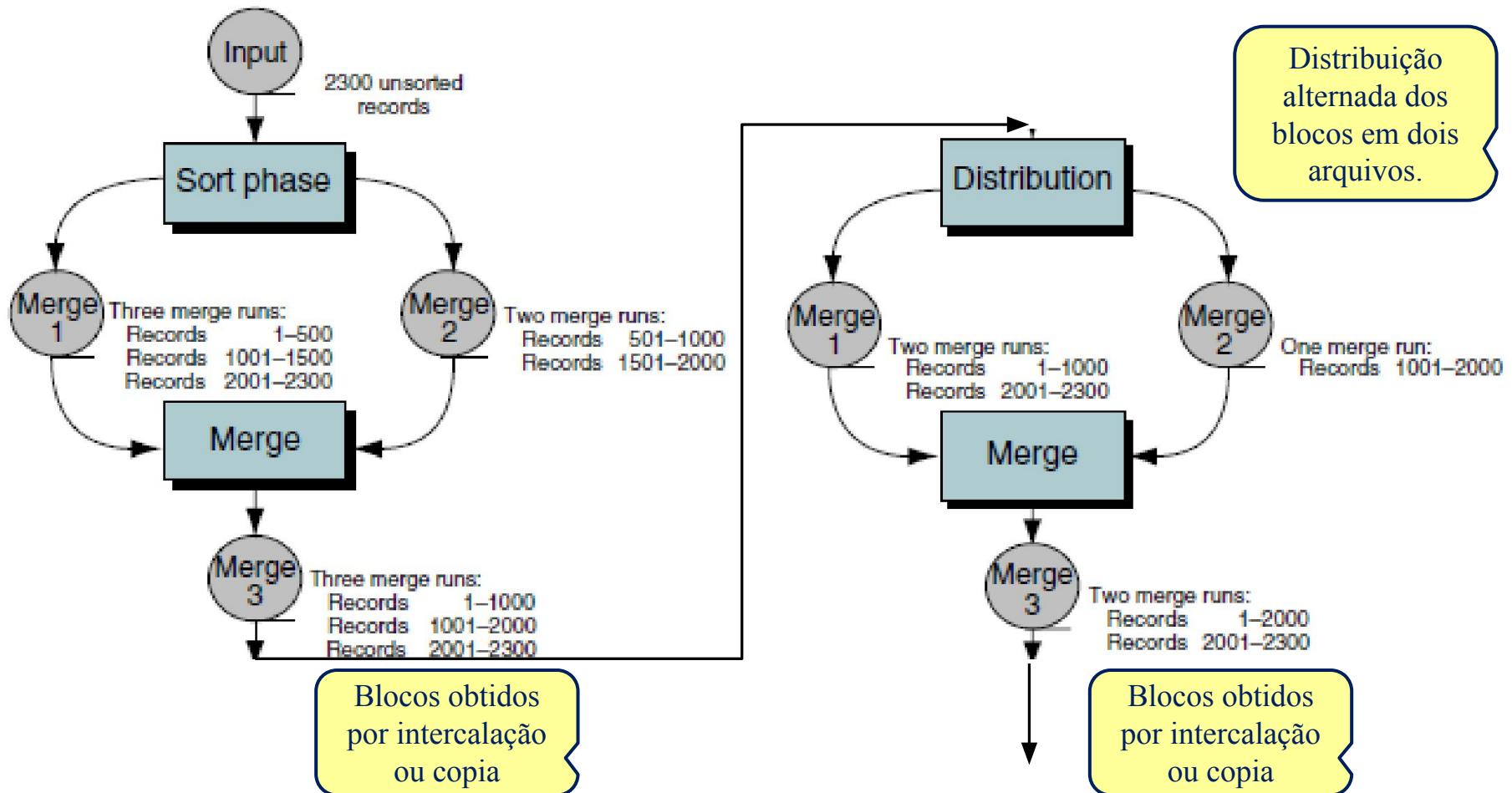
Intercalação Natural

- A intercalação natural combina um **número constante de arquivos de entrada** em um **único arquivo de saída**.
- Blocos de registros ordenados são combinados em blocos ordenados maiores. O arquivo combinado pode ter um único bloco ordenado ou vários blocos identificados pelas quebras na ordenação (*stepdown*).
- No primeiro caso, o processo de ordenação termina.
- No segundo caso, realiza-se uma **etapa de distribuição**, que consiste em redistribuir os blocos ordenados nos arquivos de saída para uma nova etapa de intercalação.
- A etapa de distribuição é um processo bastante ineficiente uma vez que as operações de leitura e escrita de registros contam como as mais demoradas no processamento de dados.

Ordenação Externa

Exemplo - Intercalação Natural (Parte1)

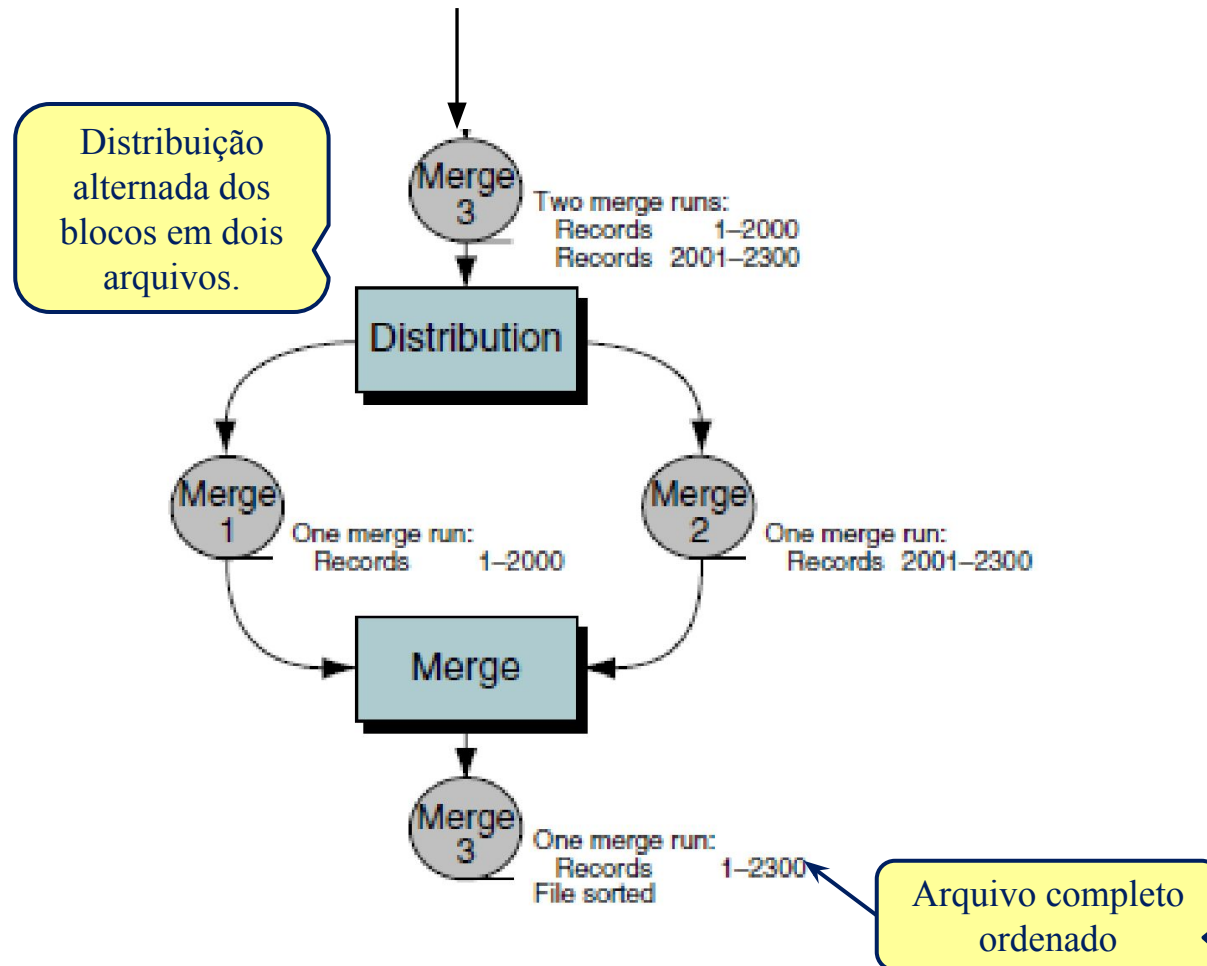
- Intercalação natural de duas vias: 2 arquivos de entrada e 1 arquivo de saída.



Ordenação Externa

Exemplo - Intercalação Natural (Parte2)

- Intercalação natural de duas vias: 2 arquivos de entrada e 1 arquivo de saída.



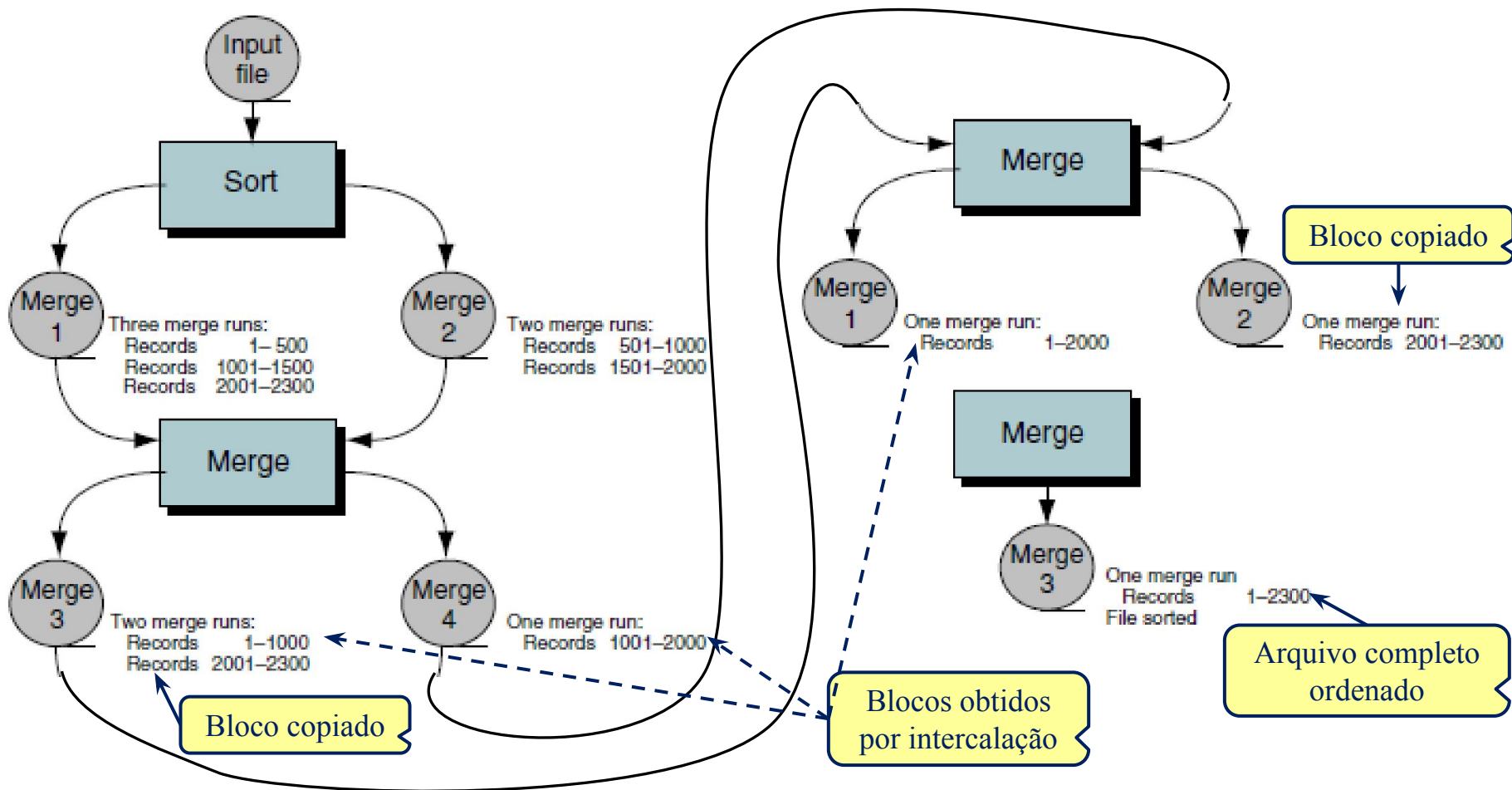
Ordenação Externa

Intercalação Balanceada

- A intercalação balanceada usa um **número constante de arquivos de entrada** e o **mesmo número de arquivos de saída**.
- Qualquer número de arquivos de entrada pode ser usado, embora mais de quatro seja incomum.
- Blocos de registros ordenados são combinados em blocos ordenados maiores, salvos em múltiplos arquivos de saída. Os arquivos de saída pode ter um único bloco ordenado ou vários blocos identificados pelas quebras na ordenação (*stepdown*).
- No primeiro caso, restará apenas uma etapa de intercalação. Enquanto que no segundo serão necessárias várias etapas.
- Este conceito de intercalação não necessita de uma **etapa de distribuição**.

Exemplo - Intercalação Balanceada

- Intercalação balanceada de duas vias: 2 arquivos de entrada e 2 de saída.



Ordenação Externa

Exemplo - Intercalação Balanceada

- No exemplo, observe que quatro arquivos são necessários na intercalação balanceada de duas vias.
- A primeira fase de intercalação:
 - Mistura o primeiro bloco do arquivo 1 com o primeiro bloco do arquivo 2, em um bloco maior que é escrito no arquivo 3.
 - Depois mistura o segundo bloco do arquivo 1 com o segundo bloco do arquivo 2, e o escreve no arquivo 4.
 - Neste ponto, todas os blocos do arquivo 2, foram processados. No entanto, existe um bloco no arquivo 1, que não foi processado, ele será copiado no arquivo 3 (*rollout*). Esta escrita pode ser considerada um esforço desperdiçado.
- A segunda fase de intercalação segue de maneira análoga.
- Observe que uma deficiência da intercalação balanceada surge quando um bloco não é combinado. O bloco é copiado repetidamente em diferentes arquivos até ser de fato combinado com outro bloco.

Ordenação Externa

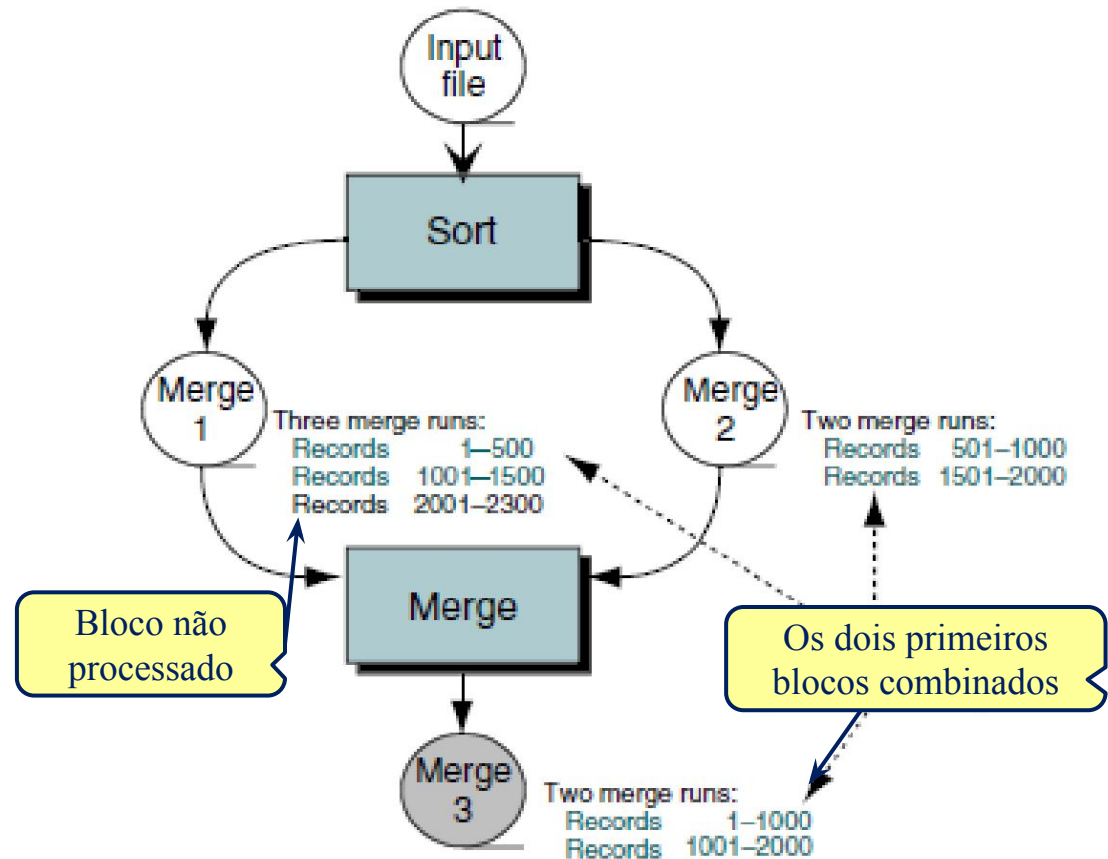
Intercalação Polifase

- A intercalação polifase usa um **número constante de arquivos de entrada** que são misturados em **um arquivo de saída**.
- Os arquivos de entrada são reutilizados, imediatamente, assim que todas as suas entradas forem completamente processadas. Os arquivos de entrada se tornam arquivos de saída. Enquanto o arquivo de saída se torna um arquivo de entrada.
- Os arquivos com blocos de dados ainda não processados continuam servindo como arquivos de entrada na próxima intercalação.
- Este método de intercalação evita fazer cópias de blocos que não foram processados.

Ordenação Externa

Exemplo - Intercalação Polifase (Parte1)

- Os dois primeiros blocos (merge runs) dos arquivos de entrada Merge1 e Merge2 são processados, intercalados e escritos no arquivo de saída Merge3.
- O arquivo Merge1 segue como arquivo de entrada.
- O arquivo Merge2, teve todos seus dados processados, logo se torna arquivo de saída.
- Já o arquivo Merge3, se torna arquivo de entrada.

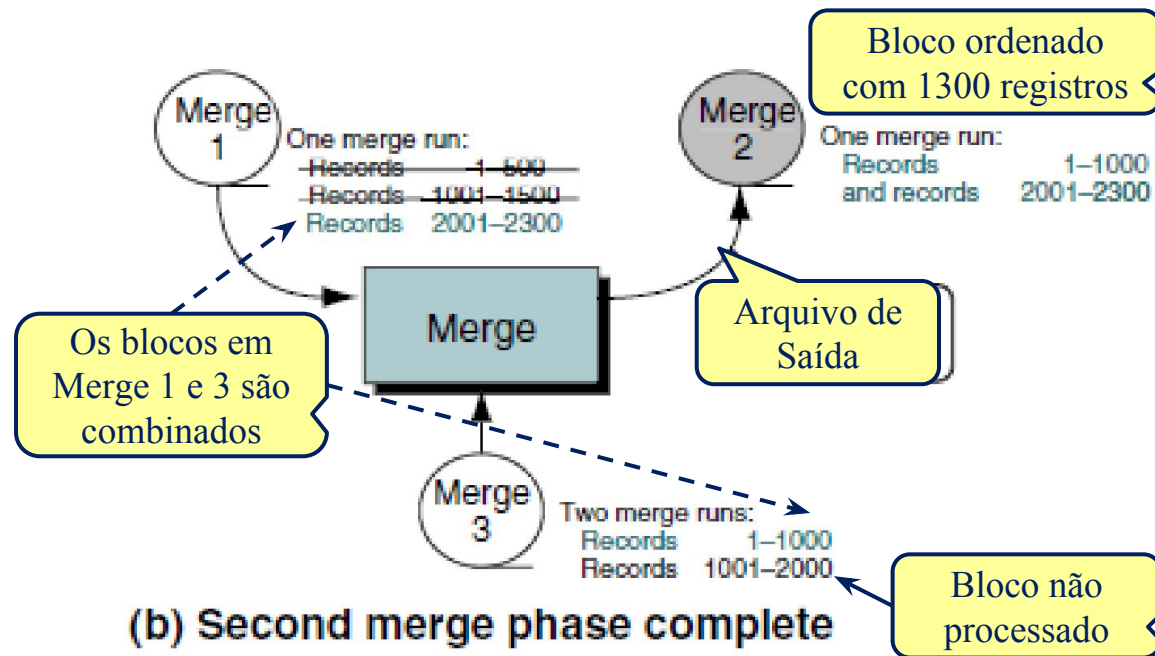


(a) First merge phase complete

Ordenação Externa

Exemplo - Intercalação Polifase (Parte2)

- Na segunda fase, os arquivos Merge1 e Merge3, servem como entrada. Já Merge2 serve como saída.
- Um único bloco ordenado é formado em Merge2.
- Merge1 teve todos os seus registros processados, mas existe um bloco não processado em Merge3.

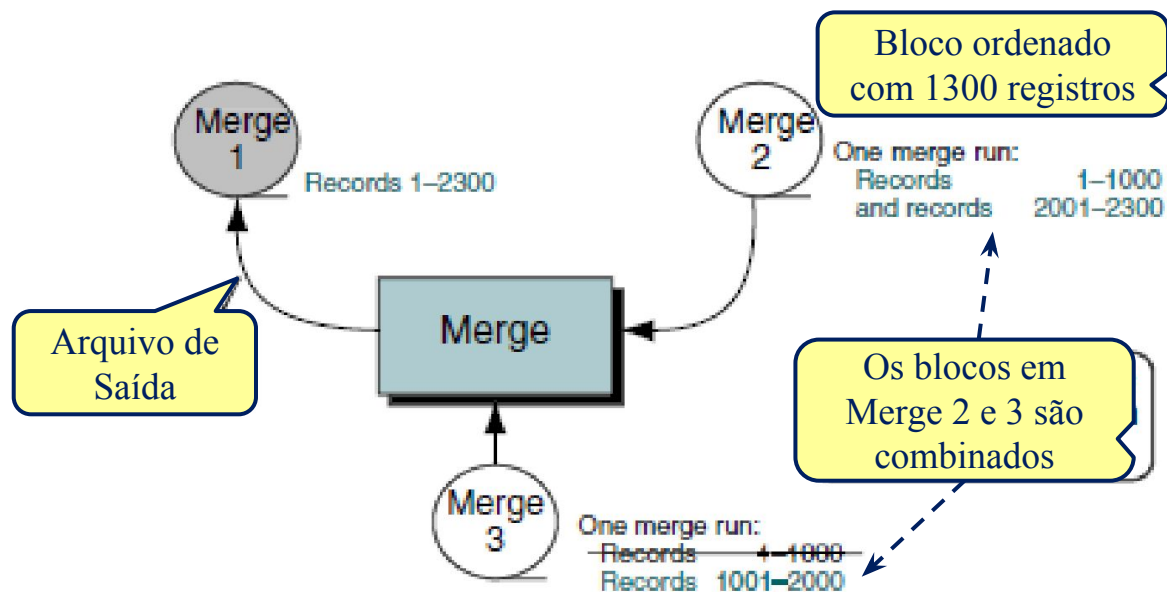


- Obs.** Blocos riscados foram processados na etapa anterior.

Ordenação Externa

Exemplo - Intercalação Polifase (Parte3)

- Na terceira fase, os arquivos Merge2 e Merge3, servem como entrada. Já Merge1 serve como saída.
- Um único bloco ordenado é formado em Merge1.
- O processo de ordenação termina.



(c) Thlrd merge phase; sort complete

- **Obs.** Blocos riscados foram processados na etapa anterior.

Referências

- Gilberg, R.F. e Forouzan, B. A. Data Structures_A Pseudocode Approach with C. Capítulo 12. Sorting. Segunda Edição. Editora Cengage, Thomson Learning, 2005.