

# Algoritmos e Estruturas de Dados II

## Ordenação Externa III

Prof. Ricardo J. G. B. Campello

## Intercalação Multi-Dispositivos

- ◆ O modelo de ordenação externa visto anteriormente assume que se dispõe de um único canal para L/E em um único dispositivo
- ◆ O processamento interno se dá usualmente de forma quase instantânea quando comparado ao tempo de acesso, que representa praticamente todo o tempo de ordenação
- ◆ Uma primeira forma de evitar esse gargalo (*bottleneck*) é utilizar 2 dispositivos independentes, um para leitura e outro para escrita
- ◆ Nesse caso, pode-se ler e escrever em paralelo, o que representa um ganho de 2 vezes no tempo

2

## Intercalação Multi-Dispositivos

- ◆ De forma mais geral, quando se dispõe de  $2m$  dispositivos, pode-se utilizar  $m$  dispositivos para leitura e  $m$  dispositivos para escrita
- ◆ Para tanto, Merge-Sort Externo deve ser modificado
- ◆ Ao invés de 2 arquivos de origem (leitura) e 2 arquivos de destino (escrita) se alternando nessas funções, passa-se a trabalhar com  $m$  arquivos de origem e  $m$  arquivos de destino em  $2m$  dispositivos
- ◆ Se usarmos múltiplos buffers de L/E independentes em memória interna, pode-se reduzir significativamente os tempos de acesso

3

## Intercalação Multi-Dispositivos

- ◆ Segundo (Aho, Hopcroft & Ullman, 1983), a redução do tempo total  $t$  de L/E pode ser de até  $2m$  vezes, ou seja,  $t$  pode ser reduzido para algo da ordem de  $t/2m$ 
  - OBS: essa estimativa simplificada é otimista...
- ◆ Adicionalmente, o no. de passagens requeridas é também reduzido
- ◆ De fato, após cada passagem o tamanho da rodada aumenta em  $m$  vezes. Com rodadas iniciais unitárias, são necessárias  $i$  passagens, onde  $m^i \geq n$ , sendo  $n$  o número de registros
- ◆ Logo,  $O(\log_m n) = O(\log n / \log m)$  passagens são suficientes
- ◆ Isso significa uma redução da ordem de  $\log m$  passagens

4

## Intercalação Multi-Dispositivos

- ◆ Como exemplo, lembramos que Merge-Sort Externo demanda em torno de 1450 acessos de 330Mb para ordenar um arquivo de 40Gb composto de registros de 1Kb quando se dispõe de 1Gb de RAM
  - 6 passagens pelos arquivos
- ◆ Com  $m = 3$ ,  $6/\log_2(3) = 4$  passagens são suficientes
  - porém, isso demanda 6 buffers, ou seja, blocos de 165Mb
  - isso implica em torno de 1940 acessos de 165Mb
    - ◆ mas acessos em parte paralelizados

5

## Intercalação Multi-Dispositivos

- ◆ Exemplo: Ordenar por ordem alfabética das chaves 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
- ◆ Assumindo que temos  $2m = 6$  fitas magnéticas independentes e que a memória primária só tem espaço para 3 registros, tem-se após a organização inicial nas 3 fitas de origem:

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

6

## Intercalação Multi-Dispositivos

- ◆ Na primeira passagem pelas fitas as rodadas de tamanho 3 são intercaladas em rodadas de tamanho 9 direcionadas de forma alternada para as fitas de saída:

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

- ◆ Mais uma passagem (rodadas de 27 registros) e o processo termina com todos os registros ordenados em uma única fita. As duas outras ficam vazias já que temos apenas 22 registros

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

7

## Intercalação Multi-Dispositivos

- ◆ O procedimento anterior é chamado **Intercalação Balanceada Multi-Caminhos** (*multi-way balanced merging*) via Múltiplos Dispositivos
- ◆ Como em qualquer intercalação multi-caminhos, existe um preço computacional a pagar: a busca interna pela menor dentre  $m$  chaves
- ◆ Em outras palavras, cada passo do procedimento de fusão não demanda mais apenas a comparação de apenas 2 chaves
- ◆ É possível encontrar a menor chave via:
  - $m - 1$  comparações  $\Rightarrow$  tempo  $O(m)$ ;
  - Implementação de uma fila de prioridade (heap)  $\Rightarrow$  tempo  $O(\log m)$

8

## Intercalação Multi-Dispositivos

- ◆ Nivio Ziviani sugere que a abordagem de heap passa a compensar a partir de  $m \geq 8$  (Ziviani, 2004), pág. 131.
- ◆ De qualquer forma, os ganhos computacionais decorrentes do aumento no número  $m$  de dispositivos de armazenamento externo tenderiam a inexistir a partir de um dado limite
- ◆ A partir desse limite o gargalo passaria a ser o tempo de fusão realizado em memória interna, que cresce na ordem  $O(m)$  ou  $O(\log m)$ , dependendo da abordagem adotada para busca da menor chave

9

## Exercícios

- ◆ Seja um arquivo de 120Gb composto de registros de 500 bytes cada. Supondo que se dispõe de 600Kb de RAM disponível:
  - Qual o máximo no. de registros em cada rodada inicial que se pode constituir para minimizar o no. de passagens requeridas pelo algoritmo de Intercalação Balanceada Multi-Caminhos?
  - Qual é esse no. mínimo de passagens assumindo que se dispõe de  $2m = 6$  dispositivos ( $m = 3$ )?
  - Qual a quantidade  $b$  de registros que pode ser transferida em cada acesso assumindo que a RAM disponível será dividida igualmente em buffers independentes para cada dispositivo?
  - Quantos acessos de L/E são realizados pelo algoritmo nesse caso?

10

## Exercícios

- ◆ Escolha uma seqüência não ordenada contendo 29 números (chaves) e assuma que temos  $2m = 6$  fitas magnéticas independentes e uma memória primária capaz de ordenar no máximo apenas 2 registros:
  - Ilustre em detalhes o procedimento de intercalação balanceada multi-caminhos dessa seqüência mostrando o conteúdo de cada dispositivo após cada passagem
    - ◆ Destaque quais são as rodadas em cada um dos passos

11

## Bibliografia

- ◆ **A. V. Aho, J. E. Hopcroft & J. Ullman, *Data Structures and Algorithms*, Addison Wesley, 1983.**
- ◆ **N. Ziviani, *Projeto de Algoritmos*, Thomson, 2a. Ed., 2004.**
- ◆ **R. Sedgewick, *Algorithms in C: Parts 1-4: Fundamentals, Data Structures, Sorting, Searching*, Addison-Wesley, 3<sup>rd</sup> Ed., 1997.**

12