

Implementação de Ordenação Externa com K-way Merge Sort em C

Jefferson Korte Junior¹, Igor Carvalho Marchi¹

¹Curso de Ciência da Computação – Universidade Tecnológica Federal do Paraná (UTFPR)
Santa Helena – PR – Brasil

jefferson.2024@alunos.utfpr.edu.br, Igorcmarchi@gmail.com

Abstract. Neste trabalho, foi feito um programa em C que usa um método chamado K-way Merge Sort para organizar dados que não cabem na memória. O processo tem duas partes: primeiro, ele cria blocos com os dados já em ordem usando o Selection Sort; depois, junta esses blocos para formar um arquivo final todo organizado. O programa também gera arquivos com números aleatórios para testar e foram feitos testes básicos para ver se tudo estava funcionando direitinho.

Resumo. Este trabalho apresenta a implementação de um algoritmo de ordenação externa usando K-way Merge Sort. O programa em C cria um arquivo com números aleatórios, separa em blocos menores, ordena cada bloco na memória e depois junta tudo em um único arquivo final já ordenado. Fizemos alguns testes básicos para verificar se o programa funcionava corretamente.

1. Introdução

Quando temos muitos dados que não cabem todos na memória, fica difícil ordenar de uma vez só. Para isso existem as técnicas de ordenação externa, que usam a memória apenas para partes menores dos dados. Neste trabalho, nosso objetivo foi implementar em C o algoritmo K-way Merge Sort, que funciona criando blocos ordenados e depois juntando esses blocos em um único arquivo final.

2. Fundamentação Teórica

O algoritmo de ordenação externa baseado em Merge Sort funciona em duas fases: na primeira, os dados são divididos em blocos menores que cabem na memória, ordenados internamente e gravados em arquivos temporários; na segunda, ocorre a intercalação desses blocos até gerar um único arquivo final ordenado.

Para ordenar os blocos na memória, pode-se escolher qualquer algoritmo de ordenação interna. Neste trabalho, foi utilizado o *Selection Sort*, pela sua simplicidade de implementação, mesmo sabendo que não é o mais eficiente.

3. Metodologia e Implementação

O programa desenvolvido em C foi dividido em três etapas principais:

- **Geração do arquivo de entrada:** criação de um arquivo `entrada.txt` com números inteiros aleatórios.

- **Criação dos blocos:** leitura do arquivo de entrada em blocos de tamanho definido, ordenação interna de cada bloco com `Selection Sort` e gravação em arquivos temporários (`bloco0.txt`, `bloco1.txt`, etc.).
- **Intercalação:** abertura dos blocos criados e intercalação K-way, produzindo um arquivo final `saida.txt` com todos os números ordenados.

Além disso, o programa possui um menu que permite ao usuário visualizar o arquivo de entrada, os blocos intermediários e a saída final.

4. Resultados

O programa foi testado com $n = 150$ registros e tamanho de bloco igual a 10. Nesse caso, foram gerados 15 blocos intermediários e, ao final, o arquivo `saida.txt` continha todos os números em ordem crescente.

Durante os testes, foi possível verificar que blocos menores aumentam a quantidade de arquivos gerados, enquanto blocos maiores tornam a ordenação interna mais custosa, confirmando o equilíbrio necessário entre processamento e operações de disco.

5. Conclusão

A implementação do K-way Merge Sort em C funcionou corretamente, produzindo arquivos de saída ordenados. O trabalho permitiu entender melhor como funciona a ordenação externa e a importância de dividir o problema em etapas (geração, ordenação e intercalação).

Referências

- [1] Aula do Dr. prof Frank Carlos Pesquisa e Ordenação de Dados. PDF - (Metodos de ordenacao)
- [2] Video aula assistida no canal - <https://www.youtube.com/watch?v=ZTdT8yn48slist=PL5TJqBvpXQv4l7nH-08fMfyI7aDFNWfC>