



Universidade de São Paulo
Departamento de Ciências de Computação
Instituto de Ciências Matemáticas e Computação
SCC0215 - Organização de Arquivos
Professor: Robson Leonardo Ferreira Cordeiro

Trabalho de Organização de Arquivos - Parte II

Alisson Mateus de Oliveira Magalhães - N° USP: 8066287

Ítalo Tobler Silva - N° USP: 8551910

Pedro Henrique Siqueira de Oliveira - N° USP: 8922006

São Carlos, 14 de Junho de 2016

Sumário

1	Introdução	2
2	Detalhes do Sistema	3
2.1	Estrutura da Árvore-B*	3
2.2	Decisões sobre a Árvore-B*	3
3	Conclusões	4
4	Bibliografia	5

1 Introdução

Este trabalho tem como objetivo a criação de um catálogo virtual de filmes (como os de uma locadora), onde os registros são armazenados em um arquivo externo. O programa possui uma interface com o usuário, onde ele pode optar por:

- Listar todo o acervo de filmes;
- Buscar um filme específico pelo ID do filme;
- Sair do programa.

Optamos por organizar o arquivo utilizando registros de tamanho variável e campos híbridos (3 campos de tamanho fixo, referentes ao ID, ano e duração do filme, e 4 campos de tamanho variável, referentes ao título, descrição, país e gênero). Tomamos essa decisão, pois o tamanho dos campos ID, ano e duração do filme não ultrapassam 4 bytes, já os outros campos possuem maior variação em seus tamanhos.

Nesta segunda parte do trabalho, complementamos o nosso sistema incluindo um arquivo de índice com campo o campo ID do filme. Isso facilita nas buscas pelos filmes, tornando-as mais eficientes. Este arquivo de índice é implementado na forma de uma árvore-B*, que é uma variação do índice árvore-B, a qual implementa a redistribuição durante a inserção. Nossa árvore-B* tem ordem 7 e taxa de preenchimento de $2/3$.

Na seção a seguir iremos mostrar detalhes pertinentes da implementação desta segunda parte do trabalho.

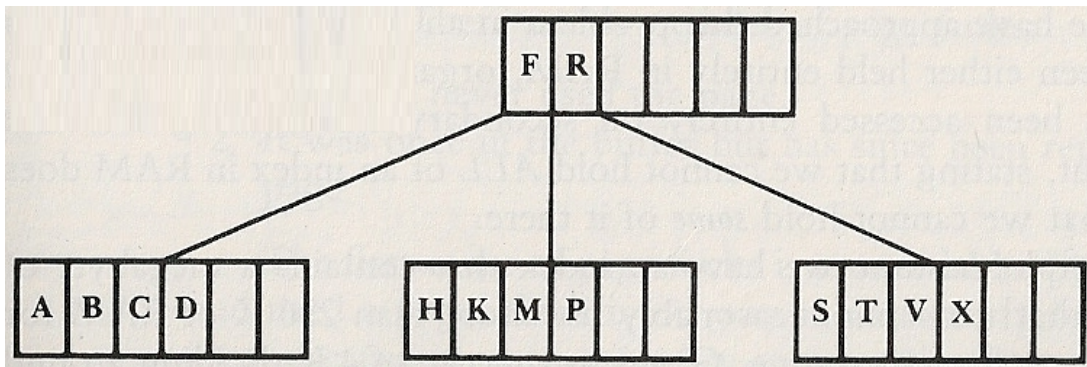
2 Detalhes do Sistema

Apresentaremos agora alguns detalhes do nosso programa, pertinentes a esta segunda parte do projeto.

2.1 Estrutura da Árvore-B*

Nossa árvore-B* é de ordem 7 e taxa de preenchimento de $2/3$. Desta forma temos, no máximo, 6 chaves por página e, no mínimo, 4 chaves. A estrutura visual desta árvore pode ser vista na figura a seguir:

Figura 1: Exemplo de Árvore-B* de Ordem 7



2.2 Decisões sobre a Árvore-B*

A página da árvore-b* possui um *unsigned char* indicando quantos campos estão ocupados. Isso quer dizer que só funciona para árvores de ordem máxima 256 por conta do overflow.

Ainda sobre as páginas da árvore-b*, temos um vetor de **INDEXELEMENT** armazenando os elementos de índice e um vetor de inteiros armazenando os RRNs das páginas no arquivo de índice. **INDEXELEMENT** é uma estrutura que armazena um *unsigned int*, que é o ID, e um *long int*, que é o *offset* no arquivo de dados.

Para mais detalhes sobre implementação, basta verificar os comentários no código.

3 Conclusões

Com este trabalho foi possível aprender mais à fundo os conceitos de indexação e de árvores B, ferramentas importantes para dinamizar a busca em grandes arquivos. Técnicas que têm como finalidade a eficiência nas buscas, como a indexação, são de fundamental importância, pois isso torna os grandes sistemas de bancos de dados mais eficientes e, conseqüentemente, menos custosos.

4 Bibliografia

- **File Structures**; Folk, M.J. and Zoellick, B.; 1992; Addison-Wesley Publishing Company.