

Nome: Gabriel Rucker

RA: 191043214

1 Relatório de Implementação

A metodologia de implementação utilizada para a aplicação de inserção de referências utilizou o conceito de árvores B para a indexação das chaves primárias, de forma que os registros pudessem ser armazenados em um dispositivo de armazenamento secundário. A implementação em si foi baseada em uma fonte externa e adaptada para os requisitos da aplicação em questão. A linguagem utilizada para a implementação foi C++.

2 Classes utilizadas na implementação

2.1 Classe Arquivo

Utilizada para as operações de leitura e escrita nos arquivos utilizados no decorrer do programa.

2.2 Classes CabecalhoDados e CabecalhoIndice

Representam os cabeçalhos de ambos os arquivos, para inicializá-los com os valores apropriados.

2.3 Classe ArquivoDados

Esta classe é utilizada para fazer referência ao arquivo de dados, onde as informações a respeito dos registros estarão armazenadas. Aqui é possível inserir objetos de referências bibliográficas na árvore B, inserir e remover nós da árvore, além de recuperar dados dos registros armazenados.

2.4 Classe ArquivoIndice

Esta classe é utilizada para fazer referência ao arquivo de índices, onde as referências aos registros estarão armazenadas. Aqui é possível inserir uma chave e um índice no arquivo de índices, remover uma chave, obter informações referentes à árvore B localizada no arquivo de índices, obter

um determinado índice do arquivo de dados a partir de uma chave, além de outras funções privadas à classe para auxiliar as operações mencionadas anteriormente.

2.5 Classe BTreeNode

Uma estrutura de árvore B foi utilizada para armazenar os índices das chaves primárias, as quais, no caso da aplicação de referências bibliográficas, se referem ao nome do livro/artigo. Dessa forma, a inserção ocorre por meio de um índice adicionado ao arquivo de índices, bem como a adição de um objeto referência bibliográfica e seus atributos ao arquivo de dados. A busca, por sua vez, ocorre por meio da obtenção do índice e em seguida a recuperação dos dados a partir do índice obtido. A alteração pode ser feita em quaisquer atributos, exceto a chave primária, de forma que para alterar uma determinada chave é preciso excluir um registro para então adicioná-lo novamente. Por fim, a exclusão de determinado registro é semelhante à inserção, obtendo-se o índice referente ao registro para então removê-lo do arquivo de dados.

Quanto ao armazenamento da árvore B, utilizou-se uma lista duplamente encadeada em um arquivo binário, de forma que os nós pudessem ser acessados a partir do arquivo semelhante a uma *heap*. Em relação a ordem da árvore B e o número de chaves em cada nó, também foram criadas funções para verificar tanto o *underflow* quanto o *overflow* nos nós, sendo que neste último caso é necessário realizar o *split* do nó e a redistribuição das chaves.

2.6 Classe NoReferencia

Representam um nó de referência bibliográfica, utilizado para ser inserido em uma lista e percorrer outros nós de referência.

2.7 Classe Referencia

Classe utilizada para a instanciação do objeto referência bibliográfica com seus atributos em si, no caso nome, autor, edicao, local, editora e ano.

2.8 Classe Util

Classe auxiliar para facilitar a manipulação dos arquivos, realizar verificações e melhorar a experiência do usuário.

3 Conclusão

A aplicação cumpriu com o requisito de inserir, buscar, alterar e excluir um registro do arquivo. Os índices utilizados, por sua vez, foram sendo criados conforme a inserção e recuperação dos dados para a memória.

Repositório contendo a implementação:

https://github.com/Gabrielrucker/Atividade_org_de_arquivos

Fontes Utilizadas:

<https://github.com/eduardocembranel/Persistent-B-Tree-on-disk->