Simulador de Árvores B*

Objetivo

Construir um **simulador de árvores B* (B*Tree)**, conforme algoritmo visto em aula e definido em [KNUTH, 1998].

Todo o desenvolvimento, desde a análise e o projeto até a implementação, deverá ser realizado empregando orientação a objetos.

Será tolerada a utilização de qualquer linguagem de programação baseada em objetos (Java, C++, C#, etc.), recomendando fortemente que seja utilizado C++ ou Java.

Requisitos

1. ***Datafile***
   - 1.1. Tamanho total de 32MB
   - 1.2. Deve ser alocado previamente, completado com fillers (zeros)
   - 1.3. Deve, obviamente, garantir a persistência dos dados entre as execuções do simulador
2. **Tabela**
   - 2.1. O simulador deverá suportar somente uma tabela de dados com a seguinte estrutura:
     - 2.1.1. Código (PK) = 4B
     - 2.1.2. Descrição = até 200 caracteres
   - 2.2. O código é único: inserção de registros com código duplicado sinaliza erro
3. ***Datablocks***
   - 3.1. Tamanho = 2KB
   - 3.2. Total de 16.384 *datablocks* no *datafile*, endereçados de 0 a 16.383 (endereços seriam de 14 bits, mas podem ser arredondados para 2B)
4. ***Datablocks* da tabela**
   - 4.1. Deverão possuir uma tabela de endereçamento dos registros em seu *header*
   - 4.2. Cada entrada da tabela de endereçamento deverá indicar:
     - 4.2.1. Posição inicial do registro: 2B
     - 4.2.2. Tamanho do registro: 2B
5. ***RowId* dos registros de dados**
   - 5.1. RowId = [endereço do datablock] + [endereço do registro] = 2B + 2B = 4B
6. **B*Tree**
   - 6.1. Sobre a tabela de dados deverá ser implementado um índice secundário B*Tree externa
   - 6.2. A chave de busca (k) da B*Tree deve ser a coluna código do registro de dados = 4B
   - 6.3. Conforme o algoritmo B*Tree, a ocupação mínima de cada nó deverá ser de 2/3 (66%)
7. ***Buffer***
   - 7.1. 256 *frames* (1.024KB)
   - 7.2. Seleção de vítimas via algoritmo FIFO
   - 7.3. Operar com *write-back cache*: se o conteúdo do *datablock* tiver sido alterado, quando ele for escolhido como vítima pelo FIFO, seu conteúdo deve ser salvo no *datafile* (*flush*)
8. **Operações suportadas**
   - 8.1. insert(código,descrição)
   - 8.2. insert(n): insere n registros com valores aleatórios para código e descrição
   - 8.3. select(código)

8.4. select(descrição)
8.5. update(chave,nova_descrição)
8.6. delete(chave)

Entregáveis

Deverá ser entregue obrigatoriamente documentação digital composta por:

1. Documentação de análise e projeto em UML. Os diagramas a serem entregues são: Diagrama de Classes (incluindo a representação de atributos e métodos) e Diagramas de Seqüência.
2. Código fonte comentado.
3. Instruções detalhadas para instalação e testes.

Entregáveis

Deverá ser entregue obrigatoriamente documentação digital composta por:

1. Documentação de análise e projeto em UML. Os diagramas a serem entregues são: Diagrama de Classes (incluindo a representação de atributos e métodos) e Diagramas de Seqüência.
2. Código fonte comentado.
3. Instruções detalhadas para instalação e testes.

Critérios de avaliação

1. Modelagem: 0,5
    a. Diagrama de Classes
    b. Diagramas de Seqüência
2. Datafile: 0,5
3. Datablocks: 0,5
4. Buffer: 0,5
5. Operações:
    a. insert(código,descrição): 2,0
    b. insert(n): insere n registros com valores aleatórios para código e descrição: 0,5
    c. select(código): 1,0
    d. select(descrição): 0,5
    e. update(chave,nova_descrição): 0,5
    f. delete(chave): 2,0
6. Código fonte comentado: 0,5
7. Apresentação: 1,0
8. Interface gráfica que mostre a árvore: +1,0

Observações

Todos os integrantes do grupo deverão apresentar o trabalho no laboratório em data a ser definida pelo professor.

Em caso de plágio da web, de semestres anteriores ou de outra turma, cuja comprovação será realizada pelo professor junto ao grupo, também será atribuída nota ZERO!

Referência

[KNUTH, 1998] Knuth, Donald (1998), Sorting and Searching, The Art of Computer Programming, Volume 3 (Second ed.), Addison-Wesley, ISBN 0-201-89685-0. Section 6.2.4: Multiway Trees, pp. 481–491. Also, pp. 476–477 of section 6.2.3 (Balanced Trees) discusses 2-3 trees.