

Relatório T1 - Sistemas Avançados de Banco de Dados

Guilherme Carbonari Boneti
PPGInf
UFPR
guilherme.boneti@ufpr.br

I. OBJETIVO

Este relatório descreve a implementação do trabalho prático **T1 - Cuckoo Hashing** da disciplina de Desempenho de SGBD. O objetivo deste trabalho foi implementar a inclusão e exclusão de valores em uma tabela hash de endereçamento aberto, utilizando duas tabelas hash com diferentes funções hash e o algoritmo de tratamento de colisão inspirado no Cuckoo Hashing.

II. O PROBLEMA

O problema proposto consiste em gerenciar a inclusão e exclusão de chaves inteiras em duas tabelas hash de tamanho fixo ($m = 11$). A primeira tabela utiliza a função hash $h1(k) = k \bmod m$, e a segunda tabela usa a função hash $h2(k) = \lfloor m \cdot (k \cdot 0.9 - \lfloor k \cdot 0.9 \rfloor) \rfloor$. Quando ocorre uma colisão em uma das tabelas, as chaves são movidas de uma tabela para a outra, conforme a lógica do Cuckoo Hashing, sem o uso de rehashing.

Além disso, foi solicitado o desenvolvimento de algoritmos de busca, inclusão e exclusão. A busca deve localizar uma chave em uma das duas tabelas. A inclusão sempre começa na tabela T1, com tratamento de colisão que envolve mover chaves entre T1 e T2. A exclusão deve remover a chave de T1 ou T2, mantendo o sistema funcional para futuras operações de busca.

III. SOLUÇÃO

A solução foi implementada em linguagem C, com três arquivos principais e um *Makefile* para facilitar a compilação. As funções implementadas seguem o enunciado, garantindo o correto funcionamento dos algoritmos de busca, inserção e exclusão de chaves.

- **hash.[c,h]**: Contém a implementação das tabelas hash e as funções principais para inserção, busca e exclusão de chaves.
- **utils.[c,h]**: Inclui funções auxiliares para leitura de entrada e geração da saída formatada.
- **myth.c**: Arquivo principal que utiliza as funções definidas para realizar as operações de acordo com os casos de teste fornecidos.

O *Makefile* foi configurado para compilar todos os arquivos e gerar o executável **myht**, conforme solicitado no enunciado. A compilação é realizada com o comando `make`, sem a necessidade de parâmetros adicionais.

IV. RESULTADOS

Foram executados os casos de teste com entradas de inclusão e exclusão de chaves, e os resultados mostraram que o algoritmo de Cuckoo Hashing foi capaz de gerenciar as colisões entre as duas tabelas, conforme esperado. A saída gerada inclui as chaves armazenadas, a tabela em que se encontram (T1 ou T2) e suas respectivas posições.

V. CONCLUSÃO

A estrutura desenvolvida permite a inclusão, busca e exclusão de chaves, simulando o comportamento do algoritmo de Cuckoo Hashing. O código foi devidamente comentado para facilitar a leitura e análise, e as funcionalidades foram organizadas em diferentes arquivos para promover a modularidade do sistema.