Documentação - Trabalho 2 de Banco de Dados

João Pedro Abensur Amoedo[1], **Isabelle Nascimento Serique**[1]

[1]Instituto de Computação – Universidade Federal do Amazonas (UFAM)
Av. Gen. Rodrigo Octávio 6200, Coroado
69080-900 – Manaus – AM

1. A estrutura de cada arquivo de dados

O arquivo de dados organizado por hash é composto de Buckets, Blocos e Registros (Artigos). O arquivo é organizado por bytes em Hexadecimal, para facilitar a leitura e o entendimento dos programadores em relação a como o arquivo está sendo escrito, se está correto. O arquivo contém 300000 Buckets, sendo o tamanho destes sendo de 4100 bytes. Cada bloco contém o tamanho de 4096 bytes, permitindo que cada bucket possua apenas um bloco, sendo 4 bytes reservados para o valor do valor hash que será utilizado para encontrar o bucket, seu bloco e seus artigos. Entretanto, os blocos foram implementados de forma dinâmica, otimizando a quantidade de artigos que podem armazenar, que pode variar de acordo com os valores nos campos dos artigos.

2. Quais fontes formam cada programa

O programa ”upload” é gerado pelos códigos fontes upload.cpp, auxiliares.cpp e auxiliares.hpp, assim como o programa ”findrec” é gerado pelos códigos fontes findrec.cpp, auxiliares.cpp e auxiliares.hpp. O header ”auxiliares” se trata de funções auxiliares de conversão que facilitaram a leitura dos dados da planilha, assim como a escrita dos dados no arquivo organizado por hash. Os códigos fontes principais são upload.cpp e findrec.cpp, que tratam da leitura e escrita do arquivo de dados, e busca no arquivo por ID, respectivamente.

3. As funções que cada fonte contém

A fonte upload.cpp utiliza duas funções principais: Upload() e insereArtigo(). A fonte findrec.cpp utiliza as funções buscaArtigo() e ShowArtigoFind(). O header auxiliares.hpp declara as funções: decHex(), bytesHex(), hexInt(), hexVal(), hexString(), intBytes(), shortBytes(), stringBytes(), ArtigoBytes() e stringArtigo().

4. Quem desenvolveu cada fonte/função

O grupo é composto por dois integrantes: João Pedro Abensur Amoedo, que desenvolveu as fontes e funções respectivas: upload.cpp e findrec.cpp. Juntamente com Isabelle Nascimento Serique, que desenvolveu as fontes e funções respectivas: auxiliares.cpp e auxiliares.hpp para facilitar as conversões para bytes e hexadecimal, e vice versa, tornando a leitura do arquivo de dados mais compreensível.

5. Qual o papel de cada função

Abaixo estão definidas os parâmetros e retornos de cada função em cada fonte do código fonte.

UPLOAD:

- upload(): Cria o arquivo organizado em hash com os dados do csv de artigos.
- insereArtigo(): recebe o ponteiro do arquivo, o id do bucket e os dados do artigo, escrevendo os dados no arquivo no espaço disponível no bucket.

FINDREC:

- buscaArtigo(): recebe o ID de um arquivo e obtém o valor ID em hexadecimal para buscar no arquivo, se o valor for encontrado executa a função ShowArtigoFind para buscar e mostrar o artigo encontrado, caso não encontre um ID correspondente a função finaliza e um erro é lançado.
- ShowArtigoFind(): recebe o valor ID de um arquivo em hexadecimal e mostra no terminal o resultado encontrado pela consulta do artigo: ID, Titulo, Ano, Autores, Citações, Atualizações e Snippet.

AUXILIARES:

- decHex(): recebe valor decimal e retorna o hexadecimal correspondente.
- bytesHex(): recebe um byte e retorna o hexadecimal correspondente.
- hexInt(): recebe um valor hexadecimal e retorna o decimal inteiro correspondente.
- hexVal(): recebe um valor hexadecimal de um digito e retorna seu valor ascii.
- hexString(): recebe dados em hexadecimal, monta uma string com os dados e retorna a string construida.
- shortBytes(): recebe um inteiro de 16 bits e retorna o ponteiro de array de bytes correspondente.
- stringBytes(): recebe uma string e retorna o ponteiro de array de bytes correspondente.
- ArtigoBytes(): recebe uma instância de artigo (struct) e retorna um ponteiro do array de bytes correspondente.
- stringArtigo(): recebe uma string (linha de dado artigo) e retorna uma instância de artigo (struct).