**Trabalho banco de dados chave-valor**

**Integrantes: Nataniel, Izabella, Helen e Ícaro.**

O objetivo do presente documento é a descrição básica das funcionalidades do banco de dados chave-valor desenvolvido na disciplina do curso de Sistemas Operacionais.

O programa foi desenvolvido conforme orientações do Prof. Pedro Penna, que forneceu os requisitos e diretrizes mínimas do trabalho. Na sequência será documentado as características mínimas do trabalho:

1. Linguagem de programação
   1. O programa foi desenvolvido na linguagem C. Foi utilizada a IDE Visual Studio Code para escrita, compilação e teste do código.
2. Sistema de versionamento Git
   1. Foi utilizado o sistema de versionamento Git para controle de versões do projeto. O código fonte do programa foi hospedado no GitHub, e pode ser acessado pelo seguinte endereço:
      * + <https://github.com/Nataniel93/simpledb>
3. Sistema de compilação do projeto
   1. Para automatização da compilação do projeto, foi utilizado o Makefile. O arquivo contendo as diretivas necessárias no processo pode ser consultado no GitHub.
4. Implementação de teste
   1. Os testes do programa foram implementados na “main”, e executados por threads.

**Estrutura básica da solução**

Conforme já explicitado, foi desenvolvido um banco de dados chave-valor, com interação do usuário por meio da linha de comandos. Neste sentido, seguem explicações dos parâmetros necessários:

* sort-key: representa a “tag” do tipo de dado a ser inserido, tal como:

|  |  |  |
| --- | --- | --- |
| **Tag** | **Finalidade** | **Atributos** |
| 1 | Cadastro de pessoas | Nome, CPF, Email |
| 2 | Cadastro de livros | Autor, Ano, Título, Editora |
| 3 | Cadastro de veículos | Marca, Ano, Modelo, Cor |

Neste projeto, foi desenvolvido a lógica apenas para “Cadastro de pessoas”, sendo utilizada a “tag” 1.

* value: representa o dado a ser efetivamente cadastrado no banco de dados. Neste caso, foi utilizada a estrutura JSON que permite a inclusão de diversos atributos. Segue modelo de JSON utilizado, com os atributos de cadastro de pessoas:

{'nome':'joao','cpf':'123456789','email':'joao@joao.com.br'}

* Key: representa a chave de identificação dos dados inseridos no banco de dados.

No caso do cadastro de pessoas, a chave é o CPF.

Após definição dos itens acima, é possível citar as operações suportadas pela interface de linha de comandos:

--insert=<sort-key,value>

--search=<key>

--remove=<key>

--update=<key,sort-key,value>

Para alocação dos dados na memória, foi utilizada a estrutura de tabela hash com lista encadeada. O CPF foi utilizado na função de espalhamento, tendo em vista a alocação dos dados na posição específica da tabela. As colisões foram tratadas justamente pela utilização da lista encadeada. Sendo assim, caso uma determinada inserção tente ser alocado na mesma posição da tabela que já possui dados, será encadeada em um novo nó da lista. As estruturas utilizadas na tabela hash foram:

typedef struct no

{

Pessoa pessoa;

struct no \*proximo;

} No;

typedef struct

{

No \*inicio;

int tam;

} Lista;

typedef struct

{

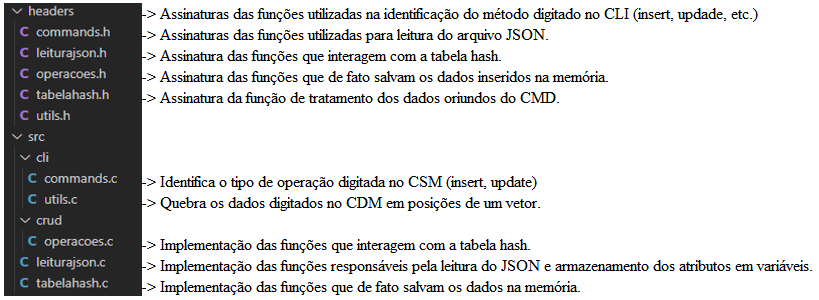
long unsigned cpf;

char nome[50];

char email[50];

} Pessoa;

Ademais, o código foi modularizado e classes específicas foram geradas:



Com o intuito de esclarecer o “caminhamento” dos dados pelo programa, foi elaborado o esquema a seguir:

