JOÃO PEDRO CORREIA

DOCUMENTAÇÃO

OBJETIVO

O objetivo do algoritmo é ler um arquivo onde tem a matrícula e o nome do estudante, e armazena-los em uma árvore binária. Também deve proporcionar para o usuário operações para manipular a árvore, podendo inserir, remover, printar, buscar e salvar a base de dados.

Para realizar a atividade em questão eu fiz apenas uma biblioteca, ela é responsável pelas operações que são feitas na árvore. Os arquivos usados são arvore.c, arvore.h e main.c, nos arquivos ‘arvore’ é onde se encontra o código fonte de manipulação da árvore binária. No ‘main.c’ é o código principal, nele é chamado a biblioteca ‘arvore.h’, faz as leituras e gravações no arquivo.

ESTRUTURAS USADAS

arvore.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// ESTRUTUTURA DA ARVORE BINARIA
struct ArvoreBinaria {
  char *matricula;
  char *nome_aluno;
  struct ArvoreBinaria *esq;
  struct ArvoreBinaria *dir;
};
```

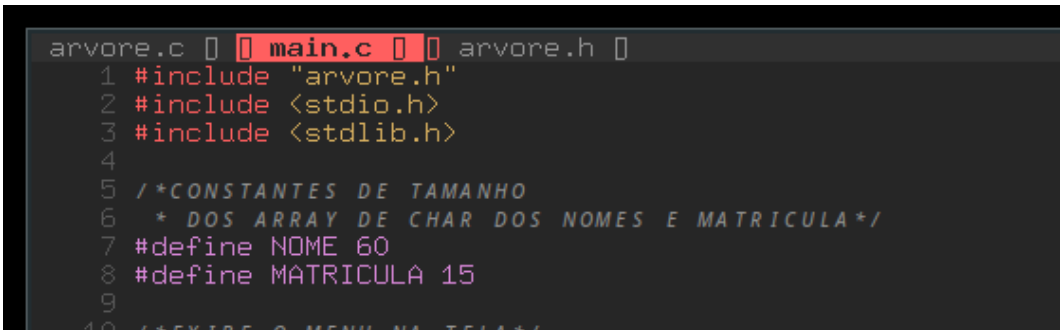
Na estrutura da árove binária tem 4 ponteiros, dois para char, onde vão ser armazenados o nome do alino e sua respectiva matricula, os outros dois são ponteiros para nós filhos.

arvore.h

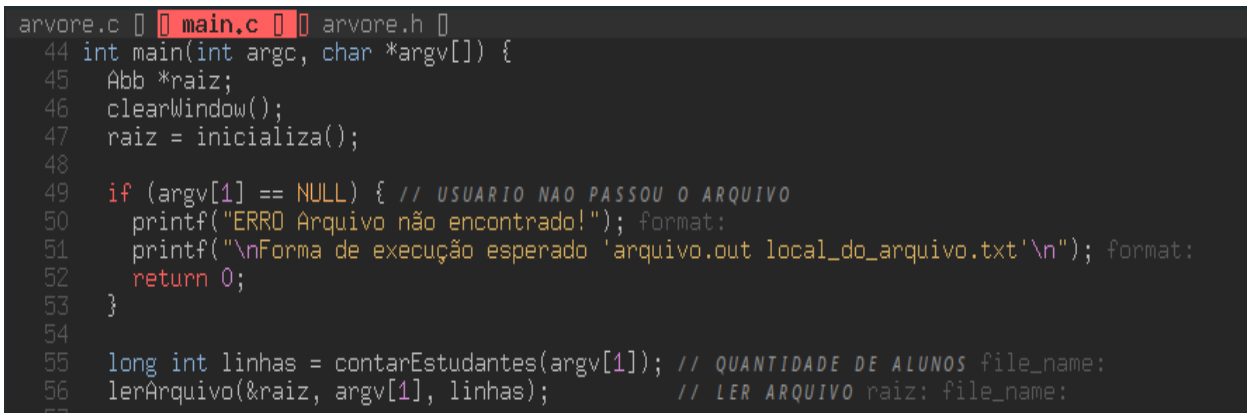
```c
#include "arvore.c"

typedef struct ArvoreBinaria Abb;

/*INSERE UM NOH NA ARVORE BINARIA*/
struct ArvoreBinaria *inserirNo(struct ArvoreBinaria **nodo, char *matricula,
                                char *nome);

/*REMOVE UM NOH DA ARVORE*/
struct ArvoreBinaria *removeNo(struct ArvoreBinaria **nodo, char *matricula);

/*PROCURA UM NO DA ARVORE*/
struct ArvoreBinaria *procurarNo(struct ArvoreBinaria *nodo, char *matricula);

/*VERIFICA SE O NODO ESTA VAZIO*/
int vazio(struct ArvoreBinaria *nodo);

/*RETORNA A MATRICULA NO NODO*/
char *getMatricula(struct ArvoreBinaria *nodo);

/*RETORNA O NOME NO NODO*/
char *getNome(struct ArvoreBinaria *nodo);

/*RETORNA O NODO DA ESQUERDA*/
Abb *noEsq(struct ArvoreBinaria *node);

/*RETORNA O NODO DA DIREITA*/
Abb *noDir(struct ArvoreBinaria *node);

/*LIMPA TODA A ARVORE*/
void freeArvore(struct ArvoreBinaria *arvore);

/*INICIALIZA*/
struct ArvoreBinaria *inicializa();
```

Essas são ass funções responsávei por manipular a árvore como um todo ou algum nó em específico. Todas estão presentes no arquivo ‘main.c’. Seus códigos fonte são encontrados dentro do arquivo ‘arvore.c’.

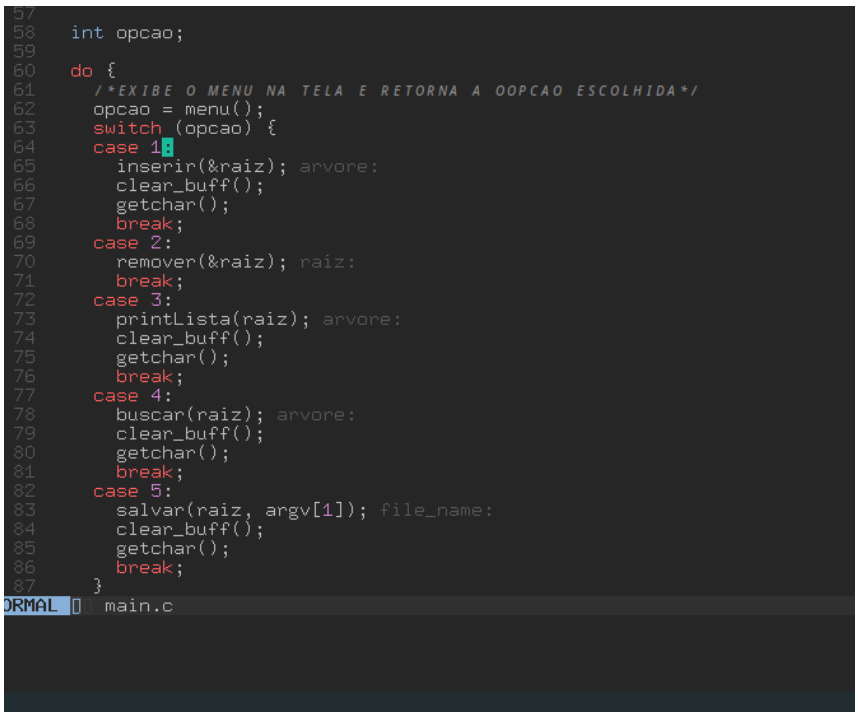
ESTRATÉGIA



Esse é o topo do arquivo ‘main.c’, é chamado as bibliotecas ‘stdio.h’, ‘stdlib.h’ e ‘arvore.h’. É definido duas constantes com pré-processamento também, a constante NOME representa o tamanho que um array onde vai armazenar o nome, a outra constante representa o tamanho do array onde vai ser armazenado a matricula. Os arrays de nome e matrícula vão ser ponteiros para char.



No inicio da função ‘main’ é esperado um parâmetro, que nesse caso é o nome do arquivo. Posteriormente o arquivo é aberto para contar a quantidade de estudantes nele e posteriormente é feito a abertura e é inserido os alunos na árvore binária da variavel ‘raiz’.



Após o armazenamento dos dados presente no arquivo na árvore, é exibido o menu de opções para o usuário e podendo escolher 6 opção que são indicadas, elas são selecionadas pelos valores discretos de 0 à 5.
Quando o programa é encerrado é chamado uma função para liberar toda a memória alocada pelos nós da árvore e suas variáveis.

COMPILAR E EXECUTAR
Para compilar basta digitar o seguinte comando:
→ **gcc main.c**

Na execução do programa é exigido que passe um arquivo como parâmetro, nesse caso arquivo de alunos:
→ ./**<arquivo.out> lista_matricula_alunos_2023.txt**

Após executar esses comando você deve ver a seguinte mensagem na tela:

Leitura de arquivo concluida
1 - Inserir
2 - Remover
3 - Printar arvore
4 - Buscas elemento
5 - Salvar base de dados
0 - Sair

Option: