**Estrutura do Programa**

O programa é dividido em várias partes:

1. **Definição da estrutura Contato**: Define como um contato é representado.
2. **Prototipação das funções**: Declara as funções utilizadas no programa.
3. **Definição de constantes**: Define o arquivo onde os contatos serão salvos e o número máximo de contatos.
4. **Variáveis globais**: Define uma lista de contatos e o número atual de contatos.
5. **Função main**: Gerencia o fluxo principal do programa.
6. **Funções auxiliares**: Implementam as operações principais do programa (adicionar, pesquisar, excluir, carregar e salvar contatos).

**Estrutura do Contato**

typedef struct {

char nome[50];

char telefone[20];

} Contato;

Essa estrutura define um contato com dois campos: nome e telefone.

**Prototipação das Funções**

Antes de serem implementadas, as funções são declaradas:

void adicionarContato();

void pesquisarContato();

void excluirContato();

void carregarContatos();

void salvarContatos();

void exibirMenu();

void removeNovaLinha(char \*str);

int comparaStrings(const char \*str1, const char \*str2);

**Definição de Constantes**

#define ARQUIVO\_CONTATOS "contatos.txt"

#define MAX\_CONTATOS 100

Essas constantes definem o nome do arquivo onde os contatos são salvos e o número máximo de contatos que podem ser armazenados.

**Variáveis Globais**

Contato contatos[MAX\_CONTATOS];

int numeroDeContatos = 0;

Aqui são definidos um array de contatos e uma variável que rastreia o número atual de contatos.

**Função main**

A função main gerencia o fluxo principal do programa:

int main() {

carregarContatos();

int escolha;

do {

exibirMenu();

scanf("%d", &escolha);

getchar(); // Limpa o buffer do teclado

switch (escolha) {

case 1:

adicionarContato();

break;

case 2:

pesquisarContato();

break;

case 3:

excluirContato();

break;

case 4:

salvarContatos();

printf("Contatos salvos. Saindo...\n");

break;

default:

printf("Opção inválida, tente novamente.\n");

}

} while (escolha != 4);

return 0;

}

* **carregarContatos()**: Carrega os contatos do arquivo na inicialização do programa.
* **exibirMenu()**: Exibe o menu de opções para o usuário.
* **scanf("%d", &escolha)**: Lê a escolha do usuário.
* **getchar()**: Limpa o buffer do teclado para evitar problemas ao ler strings posteriormente.
* **Switch-case**: Executa a função correspondente à escolha do usuário.
* **Loop do-while**: Mantém o programa em execução até que o usuário escolha sair.

**Funções Auxiliares**

**Adicionar Contato**

void adicionarContato() {

if (numeroDeContatos >= MAX\_CONTATOS) {

printf("Catálogo cheio! Não é possível adicionar mais contatos.\n");

return;

}

Contato novoContato;

printf("Digite o nome: ");

fgets(novoContato.nome, 50, stdin);

removeNovaLinha(novoContato.nome);

printf("Digite o telefone: ");

fgets(novoContato.telefone, 20, stdin);

removeNovaLinha(novoContato.telefone);

contatos[numeroDeContatos++] = novoContato;

printf("Contato adicionado com sucesso!\n");

}

Essa função adiciona um novo contato à lista.

**Pesquisar Contato**

void pesquisarContato() {

char nome[50];

printf("Digite o nome para pesquisar: ");

fgets(nome, 50, stdin);

removeNovaLinha(nome);

for (int i = 0; i < numeroDeContatos; i++) {

if (comparaStrings(contatos[i].nome, nome) == 0) {

printf("Contato encontrado: %s - %s\n", contatos[i].nome, contatos[i].telefone);

return;

}

}

printf("Contato não encontrado.\n");

}

Essa função pesquisa um contato pelo nome.

**Excluir Contato**

void excluirContato() {

char nome[50];

printf("Digite o nome para excluir: ");

fgets(nome, 50, stdin);

removeNovaLinha(nome);

for (int i = 0; i < numeroDeContatos; i++) {

if (comparaStrings(contatos[i].nome, nome) == 0) {

for (int j = i; j < numeroDeContatos - 1; j++) {

contatos[j] = contatos[j + 1];

}

numeroDeContatos--;

printf("Contato excluído com sucesso.\n");

return;

}

}

printf("Contato não encontrado.\n");

}

Essa função exclui um contato pelo nome.

**Carregar Contatos**

void carregarContatos() {

FILE \*arquivo = fopen(ARQUIVO\_CONTATOS, "r");

if (arquivo == NULL) {

printf("Arquivo de contatos não encontrado. Criando um novo...\n");

return;

}

while (fread(&contatos[numeroDeContatos], sizeof(Contato), 1, arquivo)) {

numeroDeContatos++;

}

fclose(arquivo);

printf("Contatos carregados com sucesso.\n");

}

Essa função carrega os contatos do arquivo. Aqui, **fopen** abre o arquivo em modo de leitura, **fread** lê os contatos do arquivo, e **fclose** fecha o arquivo.

**Salvar Contatos**

void salvarContatos() {

FILE \*arquivo = fopen(ARQUIVO\_CONTATOS, "w");

if (arquivo == NULL) {

printf("Erro ao abrir o arquivo para salvar.\n");

return;

}

for (int i = 0; i < numeroDeContatos; i++) {

fwrite(&contatos[i], sizeof(Contato), 1, arquivo);

}

fclose(arquivo);

}

Essa função salva os contatos no arquivo. Aqui, **fopen** abre o arquivo em modo de escrita, **fwrite** escreve os contatos no arquivo, e **fclose** fecha o arquivo.

**Outras Funções Auxiliares**

**Exibir Menu**

void exibirMenu() {

printf("\n---- Catálogo de Contatos ----\n");

printf("1. Adicionar Contato\n");

printf("2. Pesquisar Contato\n");

printf("3. Excluir Contato\n");

printf("4. Salvar e Sair\n");

printf("Escolha uma opção: ");

}

Essa função exibe o menu de opções para o usuário.

**Remover Nova Linha**

void removeNovaLinha(char \*str) {

while (\*str) {

if (\*str == '\n') {

\*str = '\0';

return;

}

str++;

}

}

Essa função remove o caractere de nova linha de uma string, que é inserido por **fgets**.

**Comparar Strings**

int comparaStrings(const char \*str1, const char \*str2) {

while (\*str1 && (\*str1 == \*str2)) {

str1++;

str2++;

}

return \*(unsigned char \*)str1 - \*(unsigned char \*)str2;

}

Essa função compara duas strings. Se forem iguais, retorna 0.

**Explicação de getchar, fgets e fclose**

* **getchar**: Lê um único caractere do stdin. No contexto do programa, é usado para limpar o buffer do teclado após a leitura de um número com scanf para evitar problemas ao ler strings com fgets.
* **fgets**: Lê uma string do stdin (ou de um arquivo). No programa, é usado para ler o nome e o telefone do contato, garantindo que não haja overflow do buffer. Ele inclui o caractere de nova linha na string, por isso a necessidade da função removeNovaLinha.
* **fclose**: Fecha um arquivo aberto. No programa, é usado para fechar o arquivo de contatos após a leitura (carregarContatos) e a escrita (salvarContatos). Fechar arquivos é importante para liberar recursos e garantir que os dados sejam salvos corretamente.