Giullio Emmanuel da Cruz Di Gerolamo

RA: 790965

Frequência F5

**Arquivo FilaCadastral.cpp**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <conio.h>

#define BUFFER 64

/\* Estrutura da lista declarada para armazenar nossos dados. \*/

typedef struct lista {

char \*nome;

int idade;

struct lista \*proximo;

} Dados;

/\* Prototipo das funcoes de manuseio dos dados. \*/

Dados \*inicia\_dados(char \*nome, int idade);

Dados \*insere\_dados(Dados \*dados, char \*nome, int idade);

void exibe\_dados(Dados \*dados);

void busca\_dados(Dados \*dados, char \*chave);

Dados \*deleta\_dados(Dados \*dados);

int checa\_vazio(Dados \*dados);

/\* Prototipo das funcoes do menu.\*/

void insere(void);

void exibe(void);

void busca(void);

void deleta(void);

/\* Inicializa a estrutura de dados principal. \*/

Dados \*principal = NULL;

/\* Cria a nova lista apontando o proximo no para NULL. \*/

Dados \*inicia\_dados(char \*nome, int idade) {

Dados \*novo;

novo = (Dados \*)malloc(sizeof(Dados));

novo->nome = (char \*)malloc(strlen(nome)+1);

strncpy(novo->nome, nome, strlen(nome)+1);

novo->idade = idade;

novo->proximo = NULL;

return novo;

}

/\* Como a lista nao esta mais vazia, apontamos o proximo no para lista anterior. \*/

Dados \*insere\_dados(Dados \*dados, char \*nome, int idade) {

Dados \*novo;

novo = (Dados \*)malloc(sizeof(Dados));

novo->nome = (char \*)malloc(strlen(nome)+1);

strncpy(novo->nome, nome, strlen(nome)+1);

novo->idade = idade;

novo->proximo = dados;

return novo;

}

/\* Percorre todos os campos da lista e imprime ate o ponteiro proximo chegar em NULL. \*/

void exibe\_dados(Dados \*dados) {

fprintf(stdout, "Cadastro:\n\n");

fprintf(stdout, "------------------------\n");

for (; dados != NULL; dados = dados->proximo) {

fprintf(stdout, "Nome: %s\n", dados->nome);

fprintf(stdout, "Idade: %d\n", dados->idade);

fprintf(stdout, "------------------------\n ");

}

printf("Pressione uma tecla para continuar.");

getch();

}

/\* Percorre cada ponta comparando o nome com a chave. \*/

void busca\_dados(Dados \*dados, char \*chave) {

int achou = 0;

fprintf(stdout, "Cadastro:\n\n");

for (; dados != NULL; dados = dados->proximo) {

if (strcmp(chave, dados->nome) == 0) {

fprintf(stdout, "------------------------\n");

fprintf(stdout, "Nome: %s\n", dados->nome);

fprintf(stdout, "Idade: %d\n", dados->idade);

fprintf(stdout, "------------------------\n");

achou++;

}

}

if (achou == 0)

fprintf(stdout, "Nenhum resultado encontrado.\nPressione uma tecla para continuar.\n");

else

fprintf(stdout, "Foram encontrados %d registros. \nPressione uma tecla para continuar.\n", achou);

sleep(1);

getch();

}

/\* Deleta o ultimo registro inserido. \*/

Dados \*deleta\_dados(Dados \*dados) {

Dados \*novo;

novo = dados->proximo;

free(dados->nome);

free(dados);

fprintf(stdout, "O ultimo registro inserido foi deletado com sucesso.\\n");

sleep(1);

return novo;

}

/\* a pena checa se a lista e NULL ou nao. \*/

int checa\_vazio(Dados \*dados) {

if (dados == NULL) {

fprintf(stdout, "Lista vazia!\\n");

sleep(1);

return 1;

} else

return 0;

}

/\* Obtem os dados necessarios para chamar as funcoes de manuseio de dados. \*/

void insere(void) {

char \*nome;

int idade;

nome = (char \*)malloc(BUFFER);

fprintf(stdout, "\n\nDigite o Nome: \n----> ");

scanf("%s", nome);

fprintf(stdout, "\n");

fprintf(stdout, "Digite a Idade: \n----> ");

scanf("%d", &idade);

fprintf(stdout, "\n");

if (principal == NULL)

principal = inicia\_dados(nome, idade);

else

principal = insere\_dados(principal, nome, idade);

}

void exibe(void) {

if (!checa\_vazio(principal))

exibe\_dados(principal);

}

void busca(void) {

char \*chave;

if (!checa\_vazio(principal)) {

chave = (char \*)malloc(BUFFER);

fprintf(stdout, "Digite o nome para buscar: \\n--> ");

scanf("%s", chave);

busca\_dados(principal, chave);

getch();

}

}

void deleta(void) {

if (!checa\_vazio(principal))

principal = deleta\_dados(principal);

}

int main(void) {

char escolha;

do {

system("cls");

fprintf(stdout, "\n Cadastro de Pessoas\n\n");

fprintf(stdout, "Escolha uma opcao: \n\n");

fprintf(stdout, "1 - Insere Dados\n");

fprintf(stdout, "2 - Exibe Dados\n");

fprintf(stdout, "3 - Busca Dados\n");

fprintf(stdout, "4 - Deleta Dados\n");

fprintf(stdout, "5 - Sair\n\n");

scanf("%c", &escolha);

switch(escolha) {

case '1':

insere();

break;

case '2':

exibe();

break;

case '3':

busca();

break;

case '4':

deleta();

break;

case '5':

exit(0);

break;

default:

fprintf(stderr,"Digite uma opcao valida!\\n");

sleep(1);

break;

}

//getchar(); /\* E para impedir sujeira na entrada da escolha. Nao lembro de nada melhor tambem. \*/

}

while (escolha > 0); /\* Loop Principal. \*/

return 0;

}

**Prints da execução**

Graphical user interface, text

Description automatically generated

A picture containing text, screenshot, monitor, indoor

Description automatically generated

A picture containing text, screenshot, monitor, indoor

Description automatically generated

A picture containing text, screenshot, monitor, indoor

Description automatically generated

Graphical user interface

Description automatically generatedA picture containing text, screenshot, monitor, indoor

Description automatically generated