

Projeto de Programação

Licenciatura Engenharia Informática

Departamento de Engenharia
Informática e de Sistemas

Ano Letivo 2022/2023

Rodrigo da Silva Calção - 2019134575



**Instituto Superior
de Engenharia**

Politécnico de Coimbra

Conteúdo

1.Objetivos E Introdução.....	4
2.Desenvolvimento.....	5
2.1 Funcionamento do Programa	5
2.3 Estruturas de dados implementadas	6
2.3 Explicação do código e justificação das opções tomadas	7
2.4 Considerações Finais.....	11

1.Objetivos E Introdução

O objetivo do trabalho é a implementação de um programa, em linguagem de programação C99, que fizesse a gestão das linhas do sistema de mobilidade do Metro Mondego. O objetivo é o utilizador poder observar e manipular: as linhas e as paragens.

Para tal teve ser criado um array de estruturas dinâmico chamado “all_paragens”, onde são guardadas todas paragens existentes e a Lista Ligada Lista_P que guarda as nossas linhas. O utilizador tem permissão para adicionar novas paragens/linhas, novas paragens a linhas já existentes, eliminar paragens e eliminar paragens de uma linha, e visualizar todas as paragens/linhas.

2.Desenvolvimento

2.1 Funcionamento do Programa

O programa começa e verifica se existe algum ficheiro chamado “linhasmetro2.bin” e o “paragensmetro.bin”, os quais tem armazenados a informação das linhas que foram criadas em programas anteriores e o das paragens..

Após a estas informações serem importadas, é apresentado o **welcome** que imprime uma mensagem de boas-vindas ao utilizador, seguido pelo **menu1** que permite ao utilizador navegar pelas linhas, pelas paragens e se este desejar fazer alterações.

2.2 Fim do programa

O programa termina quando o utilizador introduz na consola o número 9, após isto o programa guarda a informação das paragens e das linhas num ficheiro binário. Sendo depois libertada toda a memória dinâmica usada no programa.

2.3 Estruturas de dados implementadas

```
11 typedef struct info_p{
12     char nome[10];
13     char codigo[5]; // pq é alfanumérico
14     int n_linhas; // número de linhas a que pertence
15
16 }Paragens;
```

Figura 1 - Estrutura Paragens

A Figura 1, representa a estrutura Paragens, usada pelo array de estruturas dinâmico **all_paragens** para armazenar todas as informações de cada paragem.

```
31 typedef struct info_linha{
32     char nome[15]; // NOME DA LINHA
33     char** array_paragens; // array dinamico guarda paragem por linha
34     struct info_linha *prox;
35     int n_paragens; // numero de paragens que tem
36 }Linhas;
```

Figura 2 - Estrutura Linhas

A Figura 2, representa a estrutura Linhas, esta estrutura é usada pela lista ligada **L** para fazer a ligação entre todas as linhas e com as respetivas informações de cada linha.

2.3 Explicação do código e justificação das opções tomadas

O programa ficou dividido em 5 ficheiros: o main.c, o struct.h, o struct.c, e func_aux.h, func_aux.c.

❖ main():

```
14 int main() {
15
16     int total_p; // numero de paragens armazenadas
17     lista_L L;
18     Inicializa_L( listaL: &L);
19     le_bin_L(&L);
20     char *lista_arquivos[] = { [0]: "C:\\Users\\rodri\\Documents\\isec\\2\\P\\TP\\linha1.tx
21                               [1]: "C:\\Users\\rodri\\Documents\\isec\\2\\P\\TP\\linha2.tx
22                               [2]: "C:\\Users\\rodri\\Documents\\isec\\2\\P\\TP\\linha3.tx
23                               [3]: "C:\\Users\\rodri\\Documents\\isec\\2\\P\\TP\\linha4.tx
24                               [4]: "C:\\Users\\rodri\\Documents\\isec\\2\\P\\TP\\linha5.tx
25 };
26     int num_arquivos = sizeof(lista_arquivos) / sizeof(lista_arquivos[0]);
27     //ler_varios_arquivos(lista_arquivos, num_arquivos,&L,&total_p);
28     Paragens* all_paragens;
29     all_paragens = malloc( Size: sizeof(Paragens) * total_p);
30     /.../
61     all_paragens = le_bin_p(&total_p);
62     welcome(); // mensagem de apresentação
63     menu1(all_paragens,total_p,&L);
64     free_Linhas( listaL: &L);
65     return 0;
66 }
```

Figura 3 - Ficheiro main.c

A main() é a função principal, onde são chamadas todas as funções dos outros ficheiros e onde corre todo o programa.

❖ Func_aux.h:

```
func_aux.c  func_aux.h  struct.h  main.c
1 //
2 // Created by rodri on 14/04/2023.
3 //
4
5 #ifndef TP_FUNC_AUX_H
6 #define TP_FUNC_AUX_H
7 #include "struct.h"
8 void ler_varios_arquivos(char **lista_arquivos, int num_arquivos, lista_L* lista_L, int* total_p); //função lê varios ficheiros.txt
9
10 void welcome(); // mensagem de boas-vindas ao utilizador
11
12 int menu1(Paragens* all_paragens, int total_p, lista_L* L); // menu principal
13
14 void menu_paragem(Paragens* all_paragens, int total_p); // menu paragem
15
16 void menu_linha(Paragens* all_paragens, int total_p, lista_L* L); // menu linha
17
18 void menu_percurso(lista_L* L); // menu
19
20 void guarda_bin_p(Paragens* all_paragens, int total_p); // escreve a info num fich binario de todas as paragens
21
22 void guarda_bin_l(lista_L* L); // escreve num fich binario a informação de cada linha
23
24 Paragens* le_bin_p(int* total_p); // le fich binario de paragens
25
26 void le_bin_l(lista_L* L); // le fich binario de linhas
27
```

Figura 4 - Ficheiro func_aux.h

Na figura 3, vemos o *header* “func_aux.h”, este foi criado com o intuito de ter todas as funções que não alterassem as linhas nem as paragens, e ajudassem no bom funcionamento do programa.

struct.h:

```
1 //
2 // Created by rodri on 14/04/2023.
3 //
4 #ifndef TP_STRUCT_H
5 #define TP_STRUCT_H
6 #include "stdio.h"
7 #include "string.h"
8 #include "stdlib.h"
9
10
11 typedef struct info_p{
12     char nome[15];
13     char codigo[5]; // pq é alfanumérico
14     int n_linhas; // número de linhas a que pertence
15 }Paragens;
16
17 typedef Paragens* lista_P, Pparagem; // lista paragens
18
19
20 char* gera_cod(); // gera um cod random
21
22 Paragens regista_paragem(Paragens* all_paragens,int total_p); //regista-se a info de uma nova paragem, numa estrutura P
23
24 void adiciona_paragem(Paragens *P,int* total_p ); // adiciona a estrutura P,no fim do array de estruturas de todas as paragens
25
26 Paragens* elimina_paragem(Paragens *P, int* total_p);
27
28 void imprime_paragens(Paragens* P, int total_p); // imprime todas as paragens
```

Figura 5.1- Ficheiro struct.h

Aqui encontramos a estrutura Paragens e as funções que a manipulam, aqui encontramos as funções que permitem adicionar novas paragens, eliminar paragens e visualizar todas as paragens.

```

31 typedef struct info_linha{
32     int n_paragens; // numero de paragens que tem
33     char nome[15]; // NOME DA LINHA
34     char** array_paragens; // array dinamico guarda paragem por linha
35     struct info_linha *prox;
36 }Linhas;
37
38 typedef Linhas* lista_L;
39
40 void Inicializa_L (lista_L* listaL); // atribui o valor NULL a lista_L
41
42 void Insere_L(lista_L* listaL, char* nome,char** array_paragens, int tam);// cria uma nova linha
43
44 void imprime_L(lista_L* listaL);//imprime as linhas
45
46 void preenche_aux_paragens(int total_p,char** aux_array_paragens,Paragens* all_paragens,int np);// preenche uma array dinamico auxiliar com as paragens
47 // que deixamos guardar na linha
48
49 void acrescentar_paragens_linha(lista_L* listaL,char** array_paragens, int tam); //acrescenta paragens a uma linha
50
51 void delete_paragens_linha(lista_L*aux, int np);// elimina paragens de uma linha
52
53 void atualizar_linha(int total_p, Paragens* all_paragens,lista_L* listaL);//esta função permite adicionar ou eliminar paragens a uma linha
54
55 lista_L* devolve_linha(char* nlinha,lista_L* listaL); //verifica se a linha existe
56
57 void free_Linhas(lista_L* listaL);//liberta a memoria das Linhas

```

Figura 5.2- Ficheiro struct.h

Nesta figura, encontramos a estrutura Linhas, e as funções que dependem desta estrutura, como as funções que permitem atualizar uma linha (acrescentar ou eliminar paragens a uma linha), criar uma linha.

2.4 Considerações Finais

Neste trabalho desenvolvi um programa usando gestão dinâmica de memória, tentando aperfeiçoar os conceitos de lista ligada e array de estrutura dinâmica.

Também foi percebi melhor a diferença entre ficheiros de texto e ficheiros binários, reconhecendo as suas diferenças e vantagens e desvantagens quer de um quer de outro.