

TRABALHO PRÁTICO

Metro Mondego

Instituto Superior de Engenharia de Coimbra
Licenciatura em Engenharia Informática
Programação 22/23

João Miguel Baptista Duarte
Nº 2020122715

Índice

1.	Introdução	3
2.	Estruturas de Dados.....	4
2.1	Estrutura Paragens.....	4
2.2	Estrutura Linhas	4
2.3	Estrutura paragensLinhas	5
3.	Ficheiros.....	6
4.	Organização do programa.....	7
4.1	Paragens.c.....	7
4.2	Linhas.c	7
4.3	Data.c	7
5.	Estratégias utilizadas.....	8
6.	Conclusão.....	9

1. Introdução

O presente trabalho prático tem como objetivo desenvolver um programa em linguagem C para a gestão do sistema de mobilidade do metro Mondego.

O ambiente utilizado para o desenvolvimento do projeto foi o CLion.

Cada linha do metro é constituída por várias paragens. O programa é capaz de gerir as paragens, permitindo a adição, visualização e remoção das mesmas. Para além disto deve possibilitar a adição de novas linhas e atualização das linhas existentes, assim como a visualização completa das linhas.

Ao longo deste relatório serão apresentadas as estruturas de dados utilizadas, a organização do programa e as justificativas para as escolhas feitas.

2. Estruturas de Dados

2.1 Estrutura Paragens

```
typedef struct{  
    char nome[100];  
    char codigo[5];  
    char linhaqptence[100];  
} Paragens;
```

Estrutura que permite armazenar o nome e o código alfanumérico que identifica a paragem e também as linhas a que pertence a paragem.

2.2 Estrutura Linhas

```
typedef struct Linha linha, *plinha;  
struct Linha{  
    char nome[100];  
    plinha proxLinha;  
    paragensLinhas* PLparagem;  
};
```

Estrutura que representa uma linha do sistema. Esta estrutura possui uma variável nome, que é responsável por guardar o nome da linha.

Um ponteiro do tipo plinha (um ponteiro para a própria estrutura), que aponta para a próxima linha na lista ligada.

E um ponteiro do tipo paragensLinhas (PLparagem) que aponta para a estrutura paragensLinhas que contém a informação sobre as paragens associadas a essa linha.

2.3 Estrutura paragensLinhas

```
typedef struct ParagensLinhas paragensLinhas;  
struct ParagensLinhas{  
    char nome[100];  
    char codigo[5];  
    paragensLinhas* proxparagem;  
    char paragemqpertence[100];  
};
```

Estrutura que representa uma paragem associada a uma linha. Esta estrutura possui uma variável nome que guarda o nome da paragem. Uma variável código que guarda o código da paragem e um ponteiro do tipo paragensLinhas* (proxparagem) que aponta para a próxima paragem na lista ligada de paragens associadas à linha. Possui também uma variável que guarda as paragens que pertencem à linha.

3. Ficheiros

No programa existem quatro funções destinadas à manipulação de ficheiros. Uma que está responsável pela escrita dos dados das paragens num ficheiro binário, outra responsável por fazer a leitura desses mesmos dados, outra destinada a escrever os dados das linhas e das respetivas paragens e por fim, uma para ler esses dados. Existe ainda uma função para obter o total de paragens que estão no ficheiro

Esta parte foi a que me causou mais dificuldade na realização do trabalho prático, não estando implementada na totalidade.

Não consegui fazer a função correta para ler os ficheiros de texto e então optei por não implementar dado que ia ser código inutilizável.

4. Organização do programa

4.1 Paragens.c

Contém o conjunto de funções relacionadas com as paragens, seja o registo, a visualização ou eliminação. As funções de registar e eliminar paragem operam sobre o array de estruturas “Paragens” através de memória dinâmica, cada vez que se adiciona uma paragem e se elimina uma paragem.

Contém também uma função responsável por gerar o código alfanumérico, único, respetivo de cada paragem criada.

4.2 Linhas.c

Contém o conjunto de funções relacionadas com as linhas. Uma função responsável por adicionar uma nova linha, juntamente com as suas paragens. Uma função para visualizar a lista de linhas e as respetivas paragens. Outra função que permite eliminar uma paragem específica de uma determinada linha. E por fim três funções responsáveis por atualizar a linha, é possível adicionar uma paragem no início, no fim ou ordenadamente.

4.3 Data.c

Contém a implementação das funções relacionadas com a manipulação de ficheiros. Possui uma função que tem o objetivo de guardar dados num ficheiro binário e outra função para ler os dados desse mesmo ficheiro. Possui também uma função responsável por ler de um ficheiro texto uma determinada linha e as respetivas paragens.

5. Estratégias utilizadas

Toda a informação das paragens está armazenada num array de estruturas dinâmico, no entanto para fazer a ligação de paragens com as linhas, optei por copiar a informação do array de estruturas para uma lista ligada de paragens que pertence às linhas.

Para adicionar paragens às linhas, optei por criar uma função para cada situação, ou seja, existe uma função para adicionar paragens no fim, no início e também numa posição à escolha do utilizador, bastando apenas inserir o código da paragem anterior à que se quer adicionar.

Para manuseamento dos ficheiros, optei por ter quatro funções, duas que são destinadas a guardar e ler os dados apenas do array de estruturas das paragens. As outras duas funções são utilizadas para guardar e ler os dados respetivos às listas ligadas (linhas e respetivas paragens).

6. Conclusão

A realização deste trabalho foi extremamente importante para aprofundar os conhecimentos sobre a manipulação de estruturas de dados como arrays dinâmicos e listas ligadas e também leitura e escrita de ficheiros.

Foi também importante para desenvolver as minhas capacidades de investigação e resolução de problemas.

Penso que realizei um bom trabalho, dado que o programa é capaz de fazer praticamente todas as funcionalidades necessárias e pedidas no enunciado (excetuando a leitura do ficheiro de texto e o cálculo de percursos em linhas diferentes).