

Trabalho Prático

Metro do Mondego

Programação

Paulo Guilherme Lopes Sá - 2021142819 - a2021142819@isec.pt

2022/2023

Índice

Main // main.c	3
"sair"	3
Paragens // metroparagens.c	3
"ver paragens"	3
"adicionar paragem"	3
"eliminar paragem"	3
Linhas // metroLinhasParagens.c	4
"adicionar linha"	4
"ver linhas"	4
"eliminar linha"	4
"adicionar no a linha"	4
"eliminar no a linha"	4
"localizar"	5
"calcular percurso"	5
Ficheiro // metroFicheiros.c	5
"Guardar os dados"	5
"inserir linha"	5

Main // main.c

Na main tem como componente principal um loop que só acaba quando o utilizador escreve “sair”, dentro desse loop têm if e else para determinar o comando do utilizador.

"sair"

Sai do programa e guarda o que está em memória para um ficheiro.

Paragens // metroparagens.c

As paragens são um array dinâmico de uma estrutura que tem como parâmetro interno nome da paragem número de linhas que a paragem está inserida e o código que é unico para cada paragem como pedido, em relação à inserção de paragens na memória do programa tem 3 opções: "ver paragens", "adicionar paragem" e "eliminar paragem" .

“ver paragens”

Ver paragens mostra o nome de todas as paragens que existem na memória do programa como o número de linhas que ela está inserida e o código da paragem.

“adicionar paragem”

No comando adicionar paragem é feito uma realocação até mesmo na primeira inserção porque o ponteiro está em “NULL” por isso o realloc funciona como um malloc, a paragem vai para o final de array dinâmico já que não era pedido uma ordenação do array por isso optei por não o ordenar.

“eliminar paragem”

Em eliminar a paragem optei por pôr o último elemento do array em lugar da paragem que ia ser eliminada e fazer uma realloc para diminuir o array em um elemento da estrutura. quando o array não tem um elemento faz free da memória dinâmica e return “NULL”, mas a ação(eliminar paragem) só tem efeito quando a variável da estrutura que tem significa o número de linhas que a paragem está inserida está a “0”.

Linhas // metroLinhasParagens.c

As linhas não basicamente uma lista ligada simples que cada nó tem um ponteiro para outra lista liga duplamente ligada que são as paragens. Em relação à inserção de paragens na memória do programa tem 6 opções: "adicionar linha", "ver linhas" e "eliminar linha", "adicionar no a linha", "eliminar no a linha" e "localizar".

“adicionar linha”

No comando adicionar linha é criado um espaço de memória dinâmico e inserida a informação e a ponteiro de primeira linha fica com o ponteiro para o início da linha e devolvida um ponteiro para esse espaço de memória que começa a ser o início da lista de linhas se o utilizador quiser inserir paragem tem que utilizar o comando "adicionar no a linha".

“ver linhas”

Ver linhas mostra as linhas e logo em seguida as paragens da linha não retorna nada em modificar.

“eliminar linha”

Como não era pedido, o utilizador só pode eliminar linhas quando não estiver associada com nenhuma paragem, caso não esteja o utilizador pode eliminar uma linha do fim início ou meio.

“adicionar no a linha”

O utilizador pode escolher se quer inserir a paragem no fim no início ou no meio mas se escolher no a linha tem que ter pelo menos 2 paragem para poder inserir se ele se enganar a por o código da paragem vai para a página inicial.

“eliminar no a linha”

Há a possibilidade de apagar uma paragem de uma linha menos que a linha já não tenha paragens o utilizador tem que introduzir o código e o nome da linha.

"localizar"

O utilizador introduz o código da paragem que quer procurar e a função diz as linhas que o código está para poupar tempo o código não vai até ao fim das linhas mas sim até encontrar o número de vez que ele está inserido esta informação está na estrutura das paragens.

"calcular percurso"

Calcular percurso calcular o percurso que está em uma linha ou duas em uma linha vai percorrer as linhas até achar a paragem de origem e depois vai para a frente ou para trás em busca de paragem de destino até encontrar se não encontrar vai procurar em 2 linhas primeiro vai a traz as linhas que a paragem de origem e de destino estão inseridas depois vai passar por cada linha e calcular se existe uma paragem em comum se achar essa paragem calcula o percurso de origem até paragem em comum e da paragem em comum até a de destino.

Ficheiro // metroFicheiros.c

"Guardar os dados"

Um ficheiro binário que tem a quantidade de paragens para saber quantos ler depois de a iniciar o programa depois de guardas as paragens guarda as linhas e cada linha tem o número de paragens que está inserida nela por isso é mais fácil de restaurar os dados.

"inserir linha"

O utilizador escreve o nome do Ficheiro com .txt no final e o programa lê a primeira linha e as paragens se a paragem não estiver inserida na memória do programa não é inserir e avisa o utilizador que essa paragem foi pulada mas mesmo assim não dá erro e cria a linha com as paragens que tem referência no espaço de memória se o utilizador escreveu mal o nome da paragem tb é corrigido automaticamente.