Universidade do Minho
Licenciatura em Engenharia e Gestão de Sistemas de Informação

Relatório Projeto

Algoritmos e Estruturas de Dados 2

janeiro 2022

PL1
Gonçalo Fernandes Pereira, nº96550
José Pedro Costa Vilaça, nº97992
Mariana Magalhães Araújo, nº97245

Índice

Introdução

No âmbito da Unidade Curricular de Algoritmos e Estruturas de Dados 2 do 2º Ano de LEGSI foi nos proposto a realização de um sistema de estacionamentos para uma empresa de gestão de parque automóvel. Com a realização deste trabalho pretendemos desenvolver um sistema que permita a criação de um parque com pisos e lugares, guardar a informação sobre o parque e sobre cada viatura estacionada, estacionar viaturas, retirar viaturas, calcular o preço a pagar com base no tempo de ocupação e visualizar a ocupação do parque.

Estratégia Adotada

Para a resolução deste projeto inicialmente começamos por definir a estrutura do nosso parque. Este será constituído por pisos e cada piso terá um determinado número de lugares. Este será armazenado num array bidimensional de caracteres, uma vez que cada elemento será constituído por “L”, no caso do lugar se encontrar livre, ou “O”, caso esteja ocupado. Esta foi a estrutura decidida dado que nos facilita na funcionalidade de visualizar a ocupação do parque.

De seguida para guardar os dados relativos a cada carro optamos por criar uma struct que irá guardar a matrícula, o piso e o lugar em que está estacionado, e a data de entrada. Para esta última utilizamos também uma struct que irá guardar o dia, mês, ano, hora, minutos e segundos em que o carro entrou.

Para armazenar os carros estacionados optamos também por um array cuja memoria será alocada dinamicamente consoante o número de carros estacionados. Quando um carro for estacionado este será adicionado à lista e quando sair, será removido da lista.

Para realizar o estacionamento do carro optamos por este ser estacionado no próximo lugar que se encontrar disponível.

Desta forma conseguimos realizar todas as funcionalidades que pretendemos para este projeto.

Solução Proposta

O nosso projeto é constituído por 3 ficheiros, sendo estes o “main.c”, “funções.c” e “carro.h”. O primeiro é constituído pela main do nosso programa, o segundo por todas as funções que foram definidas para a realização deste projeto e o último contem as structs utilizadas para armazenar os dados.

Criar parque

Para criar o parque o utilizador começa por indicar o número de pisos e lugares, alocamos a memoria necessária e por fim preenchemos o array com o caracter “L”, uma vez que inicialmente todos os lugares se encontram disponíveis.

Mostrar Parque

Para mostrar o parque percorremos o array e imprimimos os caracteres que este contem.

Lugar Disponível

Esta função permite-nos determinar qual o próximo lugar disponível, ou seja, o piso e o lugar que se encontra livre. Para isso a função recebe como parâmetro o endereço de memoria da variável que irá armazenar o número do lugar e irá retornar o número do piso.

Para o determinar iremos percorrer o array do parque até encontrar um elemento que esteja preenchido com o elemento “L” e guardamos nas respetivas variáveis o piso e o lugar.

Desta forma, ao estacionar um carro sabemos qual o piso e o lugar onde este deve ser colocado.

Disponível

Esta função irá nos indicar se o parque ainda tem algum lugar disponível para estacionar um carro. Para isso irá percorrer o array do parque e, no caso de um elemento ser “L”, isto é, o lugar estar livre, irá somar 1 a um contador que é inicializado em 0.

No final, a função irá retornar o número de lugares disponíveis.

Assim, caso não tenha lugares disponíveis irá retornar 0, que corresponde ao valor lógico falso. Caso tenha lugares disponíveis, irá retornar o número de lugares, que irá corresponder ao valor lógico verdade, permitindo assim saber se tem ou não lugares disponíveis e o número de lugares.

Adicionar Carro

Através desta função podemos estacionar um carro. Irá receber como argumento a lista atual de carros e a variável contador, que nos indica o número de carros que estão atualmente estacionados.

Inicialmente declaramos as variáveis lugar e piso que irão armazenar o lugar e o piso que está livre, isto é, onde o carro deve ser estacionado.

De seguida aloca memória para o novo carro e adiciona o novo carro a lista, com os respetivos valores (matrícula, piso, lugar e data de entrada)

Por fim adicionamos 1 ao contador, uma vez que adicionamos um carro a lista e retornamos a nova lista de carros, isto é, a lista com o carro que estacionamos.

Mostrar Preço

Esta função permite-nos calcular o preço que uma determinada pessoa deve pagar pelo tempo que esteve estacionado no parque.

Para isso fazemos a diferença entre a data de saída, ou seja, a data atual, e a data de entrada do carro.

Essa diferença é convertida para segundos e arredondada para o inteiro seguinte, isto é, no caso de ter estado 0,5 segundos no parque, o valor é arredondado para 1, que corresponde a uma hora no parque, logo irá pagar o valor de 1 hora.

Retirar Carro

Utilizamos esta função para remover um carro do nosso parque.

Esta função irá receber como parâmetro a matrícula do carro que pretendemos remover. Percorremos a lista de carros e comparamos a matrícula de cada carro com a do carro que pretendemos remover e, quando encontrar a correspondente, irá colocar no array do parque o lugar onde este se encontrava como disponível, ou seja, como “L” e indicar o preço a pagar.

Por fim, de forma a retirar o carro da lista, copiamos os elementos (carros) que se encontram nas posições a seguir ao carro que removemos para a posição anterior, de forma que o último endereço de memoria fique livre. De seguida diminuímos o contador em 1, dado que retiramos um carro e realocamos a memoria para o número de carros atuais.

Ver Informações

Esta função tem como objetivo verificar se uma matrícula já se encontra atualmente estacionada no parque. Para isso percorremos o array do nosso parque e verificamos se alguma matrícula dos carros estacionados corresponde à matrícula que pretendemos verificar. Em caso afirmativo retorna 1, isto é, verdadeiro, em caso negativo retorna 0, ou seja, falso.

Conclusão

Com a realização deste trabalho, conseguimos aplicar todos os conhecimentos adquiridos na unidade curricular de Algoritmos e Estruturas de Dados 2. Ao longo deste projeto enfrentamos algumas dificuldades que só foram ultrapassadas com a troca de opiniões e ideias entre os elementos do grupo. Este trabalho foi muito importante pois permitiu-nos descobrir como funciona todo o processo de criação de um programa.

Todas as funcionalidades propostas foram implementadas no sistema.

Bibliografia

- StackOverflow - https://stackoverflow.com
- Materiais disponibilizados na UC
- Programiz - https://www.programiz.com/c-programming