

RELATÓRIO TRABALHO DE IALG

Integrantes : Vinicius Passos Oliveira, Marlon Silva Araújo

O trabalho em questão é um programa de gerenciamento de universidades, desenvolvido em C++. O programa permite ao usuário realizar operações como exibir, adicionar, remover e editar informações sobre universidades.

O programa utiliza as bibliotecas "iostream" para lidar com a entrada e saída de dados, "fstream" para lidar com arquivos, a "cstring" para manipular char e usar a função "strcpy" e a "string" para lidar com strings

Temos um registro (struct) com o nome de "Universidade" que é responsável por pegar as informações de cada universidade que será lida pelo nosso programa, os dados no arquivo estão armazenados na seguinte ordem :

- Identificador : variável do tipo inteiro que serve como identificador de cada universidade
- Nome : uma variável do tipo char responsável por ler/armazenar o nome de cada universidade
- quantidadeCursos: variável do tipo inteiro que mostra a quantidade de cursos de cada universidade
- cidade : variável do tipo char que mostra a cidade em que está a universidade
- posição : variável do tipo inteiro que mostra a posição da universidade no ranking das melhores do país

Temos a função do tipo "void" chamada "imprimirUniversidades", que é passada como parâmetro um vetor constante do tipo "Universidade". Optamos por criar uma função própria para imprimir as variáveis da struct para que o código fique menos bagunçado e para evitar erros na hora de imprimir.

Logo após temos uma função de tipo "void" chamada "exibirUniversidades", tem como parâmetro um vetor constante do tipo Universidade e a variável tamanho que armazena a quantidade de universidades. Usamos uma condicional para ver se o tamanho é igual a zero, se for é porque não tem nenhuma universidade cadastrada, se for maior que zero ele cai no else e imprime todas as universidades que tem dentro do vetor. Utilizamos a função "imprimirUniversidades" dentro de uma repetição e passamos o vetor como parâmetro para que ele imprima as informações de todas as universidades.

A função "stringParaInt" que converte os dados do vetor para o tipo inteiros para que possamos manipulá-los de forma correta, durante nossas pesquisas descobrimos que existem funções prontas que já fazem isso dentro do C++, mas para evitar o uso de bibliotecas que não foram ensinadas optamos por escrever o código e criar uma função para ele.

A função "particionar" e a função "quickSort" implementa o algoritmo de ordenação eficiente do nosso código, utilizando o método de particionamento de Hoare.

A função carregarUniversidadesBinaria é chamada para carregar os dados das universidades a partir do arquivo binário para o vetor de universidades, retornando o tamanho atual do vetor, se o arquivo binário ainda não tiver sido criado é chamada a função "carregarUniversidades" que tem o mesmo objetivo, só que carrega os dados pro vetor a partir do arquivo CSV.

A função salvarUniversidades salva as informações das universidades em um arquivo CSV. A função percorre cada universidade no vetor e escreve seus atributos no arquivo CSV, separando-os por vírgulas.

A função "adicionarUniversidade" permite adicionar uma nova universidade ao vetor de universidades. usuário insere as informações da universidade, como identificador, nome, quantidade de cursos, cidade e posição. As informações são lidas a partir da entrada padrão com a string "pegar" e em alguns casos é necessário converter para o tipo inteiro utilizando a função "stringParaInt". Em seguida, a universidade é adicionada ao vetor de universidades na posição tamanho e o contador tamanho é incrementado. Por fim, uma mensagem é exibida informando que a universidade foi adicionada com sucesso.

A função "lerPosicaoUniversidade" permite ao usuário pesquisar uma universidade com base no identificador ou no nome.

A função removerUniversidade permite remover uma universidade do vetor de universidades com base em um identificador fornecido. Primeiro, a função verifica se há universidades cadastradas. Em seguida, solicita ao usuário que digite o identificador da universidade a ser removida. Se a universidade for encontrada, ela é removida do vetor, deslocando os elementos seguintes para preencher o espaço vago. Caso contrário, uma mensagem informando que a universidade não foi encontrada é exibida.

A função editarUniversidade permite editar os detalhes de uma universidade existente no vetor de universidades. Primeiro a função verifica se há universidades cadastradas, em seguida, solicita ao usuário que digite o identificador da universidade que deseja editar. Se a universidade for encontrada, são solicitadas as novas informações e os valores correspondentes são atualizados no objeto universidade dentro do vetor. Caso a universidade não seja encontrada, uma mensagem informando que a universidade não foi encontrada é exibida.

A função "criaArquivoBinarioARRAY" é responsável por criar e escrever um arquivo binário a partir de um vetor do tipo Universidade.

A função "carregarUniversidadesBinario" lê as informações de um arquivo binário e armazena elas em um vetor do tipo Universidade.

A função "salvarUniversidadesBinario" e "salvarUniversidadesCSV" é responsável por salvar as alterações que foram feitas no programa, cada uma salvando com seu respectivo tipo (Binário e CSV).

A função main do programa exibe um menu onde o usuário tem várias opções de escolha.

Inicialmente é declarado um vetor dinâmico de 100 posições e uma variável que armazena o nome do arquivo CSV.

Depois é criada a variável tamanho universidades que executa a função carregarUniversidadesBinario e retorna o tamanho do vetor.

Se o arquivo binário ainda não tiver sido criado(o que vai acontecer quando você executar o programa pela primeira vez) , ele vai ler a função carregarUniversidades que lê do arquivo CSV.

O programa entra em um loop do-while que exibe um menu com opções para o usuário escolher apertando um dos números correspondentes, dentro do loop utilizamos um switch case em vez de usar if else, considerando que são várias opções e assim o código fica mais clean.

Conclusão:

O trabalho ao todo foi muito desafiador, mesmo abordando questões que já "conhecíamos" ainda tivemos que correr atrás de novas informações e jeitos de se fazer. carregar as informações dos arquivos para o registro, como tratar esses dados e fazer com que o programa armazene-os de forma correta nas duas situações diferentes de arquivos (binário e csv) com certeza foi a parte mais desafiadora desse projeto.

Foi um projeto que agregou muito no nosso conhecimento e deu uma noção de como funciona realmente o mundo da programação.