Nome: Eder Oliveira Dias
Data: 09/05/2024

Documentação do primeiro trabalho da disciplina estrutura de dados

Prof: Patricia Cockhorn Costa

Introdução:

O problema pedia para criar uma lista de usuários em que cada usuário tinha uma lista de com várias playlists, sendo que cada playlist deveria ter uma lista de músicas. A primeira coisa que pensei fazer foi ler os arquivos amizade.txt e playlist.txt dados.

O problema pedia que se fizesse uma nova lista de playlists para cada usuário, sendo que cada nova playlist representasse um autor/banda único da lista antiga e que tivesse suas músicas numa lista de músicas dentro da playlist, criando no final um arquivo played-refatorada.txt como essas novas playlists. Depois o problema pedia a criação de um arquivo similaridades.txt com a quantidade de músicas iguais entre dois amigos.

Implementação:

Eu resolvi criar três bibliotecas de listas, com cada lista lidando com uma camada do problema e usando a main para ler os arquivos pedidos e carregar minhas listas. A primeira lista representada pela struct tListaL tinha um usuário em cada nó,uma matriz de nomes com os nomes dos amigos do usuário e uma lista de playlist(ListaP), que é a segunda lista usada representada pela struct tLista, e em cada nó da playlist tinha seu nome e uma lista das músicas, que é a terceira lista usada usada representada pela struct tListaM. Assim meu programa tinha na camada superior a listaL e as demais nas camadas inferiores. Na main eu carreguei os dados pedidos na ListaL, os amigos, playlist,usuários etc. Depois fiz uma biblioteca refatorar.h para cumprir a primeira parte do problema. Nela se encontra a função refatoralistaL que faz as novas listaP para cada usuário e um struct como uma matriz de palavras para registarem os autores/bandas únicos de cada lista de ListaP, de forma a ajudar a criar a nova listaP. E tive que prestar atenção em liberar as listaP antigas.

Para resolver a segunda parte do problema foi mais simples. Tive que usar a struct com os amigos criada no início para contar a quantidade de músicas iguais em cada conta do usuário e criar ao arquivo similaridades.txt.. Assim terminando de resolver a segunda parte do problema criei a função liberalista para liberar todas as memórias que tinha alocado para as três listas antes.

Conclusão:

Foi um trabalho legal de fazer, aprendi muito de lista encadeada com ele. As minhas principais dificuldades foi liberar a memória que aloquei. O formato das músicas foi difícil de trabalhar por causa de nomes de bandas com hifens. Também foi complicado resolver o problema da ordem de apresentar os dados nos arquivos de similaridades.txt. Achei que se a documentação do programa fosse mais clara a esses problemas tivesse menos trabalho.

Bibliografia:

https://chatgpt.com/c/cb7cf7ce-92c3-45fc-a38b-36c5b9af627b