Trabalho Final CPD

Rafael Lacerda Busatta (00297033) e
Tomás Ruschel Canales da Trindade (00326448)

18 de Setembro 2022

1 Problema

O problema escolhido foi relacionado ao assunto de cinema. Utilizamos listas de filmes que possuem o título de cada filme, ano de lançamento, um identificador único, gênero e link para o site do Internet Movie Database(IMDB). A intenção do programa é facilitar para o usuário a pesquisa de filmes de acordo com seu gosto pelo gênero, trazendo informações extras através do link para sua respectiva página no site do Imdb. Lá o usuário poderá ter acesso a informações mais completas a respeito do filme, como avaliação dos usuários, nota da crítica, classificação indicativa, elenco completo, dados de produção e bilheteria, além de curiosidades e tempo de duração. Para resolução desse problema utilizamos um vetor de estruturas, uma árvore B+ e um arquivo binário com as estruturas dos filmes. As funcionalidades implementadas foram as de leitura de arquivos, ordenamento por index, ordenamento inverso por index, busca por index e pesquisa por ano.

2 Implementação

Inicialmente é feita a leitura de múltiplos arquivos .txt e .csv. Ao ler os arquivos, fazemos a inserção dos dados de cada filme em um vetor de estruturas. A estrutura utilizada é do tipo Movie, que armazena título, ano de lançamento, gênero(s), um identificador único do filme, e um identificador único do imdb que é utilizado para a impressão do link para o site. A cada filme adicionado no vetor de estruturas, criamos uma chave em um nodo com o índice do vetor na árvore. Ou seja, se fazemos a inserção de 1000 filmes em um vetor de 1000 estruturas, criamos uma árvore B+ com 1000 chaves, numeradas de 0 a 999. Foi definido no início do programa que cada nodo da árvore contém um valor fixo. Esse valor foi definido como sendo 4. Após, é percorrido o vetor de estruturas, adicionando cada campo no nó de índice correspondente na B+ Tree. A partir daí, qualquer chamada de busca, ordenamento e pesquisa será feita na árvore B+. Encrontramos bastante dificuldade na implementação da árvore B+ e em seus acessos, visto que seu código foi retirado do site Programiz[1]. Logo, a inserção das estruturas dos filmes na árvore apresentou-se como um desafio e, para solucioná-lo, tivemos que criar uma nova função que adiciona as estruturas após a criação dos nodos de índices da árvore. Para impressão em ordem, por index e por título, foram usados vetores de estrutura Movie, já que ele foi inicialmente criado em memória para inserção na árvore. Após a segunda execução do programa, já há a existência de um arquivo binário, de nome movies.dat na estrutura dos arquivos locais do projeto. Esse arquivo é constituído pelos dados dos filmes, os quais são utilizados para a criação do vetor de estruturas em memória sempre que existir, descartando a releitura dos arquivos fonte iniciais. A cada execução do programa é verificado se o arquivo movies.dat existe, e caso não exista, é feita a leitura dos arquivos.

Inicialmente se pensou em uma interface de usuário onde este pudesse escolher ler um arquivo pela primeira vez ou ler a partir de um arquivo binário já

exitente. Todavia, como a criação de um arquivo binário contendo as estruturas dos filmes visa facilitar e tornar mais eficiente a criação da Árvore B+, não faria sentido dar essa escolha ao usuário, visto que mesmo que ele optasse por ler um arquivo novo, caso esse arquivo já tivesse sido lido, o arquivo binário acabaria não sendo utilizado para a geração da árvore B+.

3 Guia de uso

O projeto foi desenvolvido no Visual Studio Code (VSCode), utilizando as extensões C/C++, C/C++ Themes, C/C++ Extension Pack, CMake Tools, todas da Microsoft, além da Code Runner, de Jon Han, Better C++ Syntax, de Jeff Hykin. A compilação foi via terminal, utilizando os seguintes comandos:
g++ -Wextra -std=c++17 -c main.cpp
g++ -Wextra -std=c++17 -c src/bPlusTree.cpp
g++ -Wextra -std=c++17 -c src/invertedIndex.cpp
g++ -Wextra -std=c++17 -o main main.o bPlusTree.o invertedIndex.o
./main
Assim, é gerado o arquivo main.exe o qual é executado com o último comando listado acima. A estrutura dos arquivos e diretórios do projeto possui o mesmo padrão do CodeBlocks, para facilitar sua importação, caso seja da preferência do usuário, todavia, um projeto novo deve ser criado por precaução, fazendo a importação dos arquivos.
Há uma interface de usuário para fazer a listagem de todos os filmes, pesquisa por id, pesquisa por ano e listagem dos filmes em ordem crescente e reversa. A lista possui pouco mais de 27 mil filmes, então foi definida a constante MAXARRAY, no início do programa principal (main.cpp) com o número de filmes que queremos utilizar dessa lista.
A busca por ano foi implementada utilizando arquivos invertidos, o qual o código é de autoria de JhaPrajjwal[2]. Foi incorporado ao menu principal do programa na main.cpp, onde é chamada a função InvertedIndex("data/movies.csv"), adicionando automaticamente o arquivo movies.csv. O submenu da aplicação ainda dá a opção de adicionar novos arquivos ao dicionário, busca query pelo ano, onde deve ser adicionado entre parênteses [e.g (1995)], e saída, a qual retorna ao menu principal da aplicação. A busca retorna a quantidade de filmes achados, lista numericamente cada um, informando o arquivo o qual ele pertence, a linha em que se encontra, que é o identificador do filme (pode-se pesquisar no arquivo movies.csv por este valor para verificar cada filmes encontrado e confirmar seu ano, conforme demonstrado em apresentação) e seu index, que é a posição em que se encontra na linha, visto que o dicionário separa todas as palavras em tokens.

4 Considerações Finais

As áreas que apresentaram maior desafio foram a implementação de árvores e de arquivos binários. Apesar de entender ambos na teoria, a prática se mostrou desafiadora. Durante o desenvolvimento do trabalho final cogitamos o uso de árvores Trie ou Patricia, mas, no fim, decidimos pelo uso da B+, já que acreditávamos ser a que teríamos mais facilidade de criação e manipulação. A utilização de arquivos invertidos foi outro ponto desafiador, devido a falta de conhecimento da linguagem C++, que inicialmente foi escolhida por ser melhor do que C, não termos familiaridade com outra linguagem, e esta ter sido utilizada durante todo o semestre no decorrer de outros trabalhos práticos. A impressão dos filmes através de linhas e index não era a esperada por nós, porém não obtivemos sucesso na no resgate das informações da linha em uma estrutura do tipo Movie para imprimir o filme corretamente. Por fim, apesar de termos resolvido o problema, não conseguimos fazer isso da melhor maneira, visto que não foi possivel salvar a árvore B+ em um arquivo binário, nem pesquisar ná arvore B+ pelo titulo do filme escolhido. Apesar de melhorarem o desempenho do programa, acreditamos que a implementação de árvores e arquivos binários se mostraram prejudicial à resolução do problema. Já que, por conta da falta de familiaridade com esses, não foi possível a implementação de um programa completo, como desejávamos. Porém, compreendemos que é por meio dessas dificuldades que aprendemos a lidar com coisas novas, e esperamos que no futuro possamos fazer melhor e usar o que aprendemos aqui para desenvolver programas mais eficientes.

5 Referências

1. https://www.programiz.com/dsa/b-plus-tree 2. https://github.com/JhaPrajjwal/Inverted-Index