Relatório do trabalho final da disciplina de Estrutura de Dados

Acadêmicos: Felipe Weiss

Leonardo Valério Anastácio

Lucas Litter Mentz

Observação: Programa desenvolvido em Linux x86\_64, kernel 4.9.10.

Não garantimos funcionamento do programa quando executado em Windows.

Em nosso trabalho, fazemos uso da ABB implementada pelo professor Gilmário para criarmos um sistema de busca de dados em um arquivo txt que contém diversos registros contendo matrícula, nome, telefone, salário, idade e departamento.

O tratamento do arquivo é feito dentro dos arquivos cliente.h e cliente.c, referente às funções com o qual o usuário pode interagir. No cliente.h está a declaração das structs info, usada pela árvore binária de busca, que contém a matrícula, que é única para cada pessoa bem como a linha onde essa matrícula se encontra no arquivo txt usado. Também se encontra neste arquivo as declarações das funções usadas pelo usuário na hora de inserir, buscar e remover alguma matrícula da árvore.

Ao iniciar o programa, é criado uma nova árvore binária de busca usando a função criaABB, implementada pelo professor Gilmário, logo após é aberto o arquivo “arq.txt”, que é onde se encontra todos os dados das pessoas cadastradas no sistema. Depois é feito uma tentativa para abrir o arquivo “abb\_index.bin”, que é o arquivo no qual é salvo a árvore binária de busca. Se não for possível abrir o arquivo, então a ABB é construída usando-se o arquivo “arq.txt” usando a função controiArvoreIndices. Esta função passa pelo arquivo inteiro, começando do início dele, pegando apenas a matrícula da pessoa bem como a linha onde está matrícula se encontra dentro do arquivo “arq.txt”. A linha é capturada com a ajuda de um contador auxiliar que é incrementado sempre que uma nova linha é lida. Essas informações – linha e matrícula – são salvas dentro uma variável do tipo info e então são “colocadas” dentro da ABB através da função insereABB.

Se o arquivo binário já existir, então, através de um laço while, é passado por todas as linhas do arquivo binário, salvando a linha dentro de uma váriavel do tipo info que é então “colocada” dentro da ABB através da função insereABB. Por fim o arquivo binário é fechado.

Logo após toda a inicialização da árvore é mostrado ao usuário um menu onde se encontram as opções disponíveis para uso:

1. A primeira função é a de busca na árvore. Para esta função é usada a função buscaABB que já se encontra implementada nos arquivos disponibilizados pelo professor. Caso a função retorne FRACASSO, é mostrada uma mensagem indicando que a matrícula não se encontra dentro da árvore, caso contrário, é usado a função fseek para pularmos diretamente para a linha dentro do arquivo onde se encontra a matrícula, usando de uma variável do tipo info dentro da árvore que indica a linha, então é copiado todo o contetúdo da linha para um vetor de char, por fim é mostrado na tela a linha onde se encontra a informação bem como a informação em si.
2. A segunda função disponível é inserção de uma nova matrícula. Para tal, primeiramente é encontrado a menor matrícula possível que ainda não foi inserida na árvore, usando de um laço do … while e de um contador, bem como da função de buscaABB. Depois é feito uma varredura pelo arquivo, indo do início dele até o primeiro ponto onde a nova informação possa ser colocada, ou seja, indo até a primeira linha disponível para ser preenchida, isto é feito usando-se um laço while até encontrarmos o final do arquivo (EOF). Depois é chamado a função inserirMatricula, onde primeira temos o arquivo sendo aberto no modo append, onde as novas informações são postas no final do arquivo, logo depois o usuário digita todas as informações pertinentes à nova matrícula a ser efetuada. Depois de ter salvo todas as informações dentro do vetor de char cadastro, ele é inserido ao final do arquivo, usando a função fprintf e logo após é inserido na ABB um novo nó contendo a matrícula nova bem como a linha na qual ele foi inserido dentro do arquivo, por fim o arquivo é fechado e reaberto no modo leitura.
3. A terceira opção disponível para o usuário é a de remoção de uma matrícula do arquivo. A remoção é feita pedindo ao usuário o número da matrícula a ser excluida, depois é feita uma busca na árvore para confirmar se aquela matrícula existe no sistema. Caso a matrícula não exista, é mostrado ao usuário uma mensagem informando que a matrícula não foi encontrada, caso contrário, é mostrado uma mensagem com todas as informações pertinentes a matrícula em questão e então é perguntado ao usuário se ele realmente deseja excluir aquela matrícula do sistema, em caso afirmativo, a árvore é destruída por completo, então é criada uma árvore nova, por fim é chamada a função removerMatricula. Esta função cria uma arquivo temporário, chamada tmp.dat em modo de escrita (“w”), depois é transferido todas as informações contida no arquivo txt original para o arquivo temporário, exceto a linha referente a matrícula que deve ser removida, isto é feito através de um laço de repetição while que tem como condição de parada quando o final do arquivo dor atingido. Depois os arquivos “arq.txt” e “tmp.dat” são fechados, o arquivo arq.txt é então removido e o tmp.dat é renomeado para arq.txt e esse arquivo é novamente aberto no ponteiro \*arquivo para devolver o ponteiro à main da forma que o recebeu. No final da função a árvore é recriada novamente, agora seguindo o novo arq.txt que já não contém mais a matrícula que o usuário pediu para remover. Isso é feito pois todos as matrículas da linha após a deletada estariam desatualizadas.

0. Sair.

Enquanto o usuário não escolhe a opção 0 (sair) o menu é repetido após executar a chamada solicitada.