Relatório - Omaggio

Gabriel Lima Pereira

8 de janeiro de 2025

Resumo

Este documento apresenta o relatório do trabalho final da disciplina de Estrutura de Dados I, referente a um gerenciador de tarefas usando os conceitos apresentados na disciplina. O gerenciador foi nomeado como Omaggio.

1 Introdução

O trabalho tem como objetivo o desenvolvimento de um gerenciador de tarefas feito em linguagem C, com requisitos referentes ao uso do conteúdo transmitido na disciplina. Devido à liberdade dada ao desenvolvimento e interpretação da problemática, tendo em vista o cumprimento dos requisitos necessários apresentados, se faz necessário um tópico para justificativa do Omaggio. Segue outro tópico destinado às personalizações do sistema.

2 Proposta

A proposta do software, como já citada anteriormente, é permitir a criação e manipulação de tarefas determinadas pelo usuário. Omaggio é um gerenciador que, completamente via terminal, possibilita que mais de um usuário tenha essa capacidade.

3 Funcionamento

3.1 Ambiente

Para que o programa funcione em um sistema operacional como o Linux é necessário ter os arquivos main.c, functions.c, functions.h, sort.c e sort.h em uma mesma pasta e ter o gcc (o compilador) instalado. Via terminal, digite o comando "gcc -o main main.c functions.c sort.c"para que o Omaggio seja compilado e, após a compilação, basta digitar o comando "./main". Para o Windows, tendo algum compilador (como o MinGW), basta executar o mesmo comando de compilação (isso já na pasta com os arquivos) e dar o comando main.exe.

3.2 Inicialização do Sistema

A inicialização do sistema começa com a execução do main(), que configura o ambiente necessário para o funcionamento do programa. A primeira ação é a chamada à função srand(time(NULL)), que define a semente para o gerador de números aleatórios, permitindo que a geração de identificadores únicos ocorra de forma randomizada. Em seguida, a função initializeUserList() é chamada para inicializar a lista de usuários, criando uma lista vazia. Esta será a lista usada ao longo da execução do programa para armazenar e gerenciar os dados dos usuários.

Após a inicialização da lista de usuários, o programa chama a função mainMenu(userList), que exibe o menu principal do sistema, permitindo ao usuário escolher entre as opções disponíveis, como login, registro e manipulação de usuários.

3.3 Menu Principal

A primeira opção permite que o usuário faça login no sistema. Para isso, o programa solicita o ID do usuário e usa essa informação para procurar o usuário na lista registrada. Se o ID for encontrado, o usuário é autenticado e o sistema entra no menu específico de tarefas do usuário. Caso o ID não exista, uma mensagem de erro é exibida informando que o usuário não foi encontrado.

A segunda opção exibe a lista de usuários cadastrados. O sistema chama a função que percorre todos os usuários e exibe suas informações, como o nome e o ID, permitindo que o administrador ou o próprio usuário visualizem os dados dos outros usuários registrados no sistema.

A terceira opção permite registrar um novo usuário no sistema. O programa chama a função responsável por adicionar esse novo usuário à lista de usuários do programa.

A quarta opção oferece a possibilidade de apagar um usuário do sistema. O programa solicita o ID do usuário a ser removido, busca esse usuário na lista e, se encontrado, o apaga. Caso o usuário com o ID fornecido não exista, uma mensagem de erro será exibida.

Por fim, a quinta opção é para encerrar o programa. Ao escolher essa opção, o sistema exibe uma mensagem de despedida e termina a execução do programa.

3.4 Menu de Usuário

No menu de usuário, a primeira opção permite listar todas as tarefas do usuário de maneira recursiva. O sistema exibe todas as tarefas associadas ao usuário, caso existam. Se a lista estiver vazia, uma mensagem informando que não há tarefas será exibida.

A segunda opção permite que o usuário insira uma nova tarefa. O programa solicita as informações necessárias para criar a tarefa, como nome e status (pendente ou concluída), e a adiciona à lista de tarefas do usuário. Dependendo do status da tarefa, ela será adicionada à fila de tarefas pendentes ou concluídas.

A terceira opção do menu do usuário oferece a funcionalidade de ordenar a lista de tarefas. O usuário pode escolher um algoritmo de ordenação (cbubble sort, selection sort, insertion sort, merge sort ou quick sort) para organizar suas tarefas de acordo com a ordem de prioridade (1 a 5).

A quarta opção exibe as tarefas pendentes do usuário. O sistema chama uma função que mostra as tarefas que ainda não foram concluídas, permitindo que o usuário tenha uma visão das tarefas que precisam ser feitas.

A quinta opção permite ao usuário visualizar suas tarefas concluídas. O sistema chama uma função que exibe as tarefas que já foram completadas, proporcionando uma visão das atividades finalizadas.

Na sexta opção, o usuário pode marcar uma tarefa pendente como concluída. O programa move a tarefa da fila de pendentes para a lista de concluídas, conforme o status da tarefa. Isso mantém o controle das tarefas que já foram feitas. É importante levar em conta que foi definido um limite de quantas tarefas concluídas a lista comporta. Na ocasião, o limite foi definido como dez.

Por fim, a sétima opção permite que o usuário busque uma tarefa específica pelo nome. O sistema realiza uma busca binária na lista de tarefas do usuário para encontrar a tarefa desejada e exibir suas informações. Se a tarefa não for encontrada, uma mensagem de erro será exibida. Vale ressaltar que o sistema ordena a lista de tarefas usada para ordenação por ordem alfabética, assim permitindo a busca binária.

4 Personalização

A única função que foi adicionada além do que foi pedido foi a responsável por deletar usuário. A funcionalidade está disponível no menu principal, acessada pela escolha da opção correspondente. Esta funcionalidade é implementada para permitir a exclusão de um usuário e todos os dados associados a ele, como suas listas de tarefas.

4.1 Deletar Usuário

Quando o usuário seleciona a opção de apagar um usuário, o programa solicita a entrada do identificador único do usuário que deve ser removido. Após receber o ID, a função deleteUser(userList, key) é chamada. Essa função busca na lista encadeada de usuários pelo nó correspondente ao ID fornecido. Uma vez localizado, a função remove o nó correspondente, ajustando os ponteiros da lista.

A exclusão de um usuário aciona a liberação de memória alocada dinamicamente para o armazenamento de seus dados, como as tarefas associadas a ele e a lista principal que possui tais tarefas. A função verifica se a fila de tarefas pendentes, lista de concluída e a lista de tarefas duplamente encadeada do usuário estão preenchidas. Se estiverem, funções auxiliares são chamadas para liberar a memória alocada por essas listas, garantindo que não haja vazamentos de memória.

Essa funcionalidade é uma personalização julgada como relevante no sistema, pois permite o gerenciamento eficiente da lista de usuários e suas informações e também a desalocação segura das informações deles.

5 Justificativa

Tendo em vista a liberdade dada ao desenvolvimento do trabalho (uma vez cumprido os requisitos propostos), o programa foi implementado de modo que todos os requisitos fossem cobertos, menos o responsável pela pilha de reversão.

A estrutura respectiva à tarefa teve seus campos definidos com base no que seria relevante para o funcionamento do programa. Não foram implementados campos como deadline pois foram julgados como desnecessários, uma vez que a lógica requerida para a manipulação de atributos assim seria de maior complexidade, enquanto que com os campos já escolhidos, se conseguiu alcançar um resultado de modo a manipular as listas usando tais campos.

No que diz respeito às listas, as estruturas foram desenvolvidas para que todas apontassem para as mesmas tarefas, dispensando a necessidade de alocações adicionais para a cópia de tarefas em diferentes estruturas. Essa solução, além de mais viável em termos de memória, auxilia a desalocação das listas, uma vez que basta a liberação da primeira lista (a encadeada normal) para que todas as tarefas sejam liberadas e, dessa forma, as demais listas tenham seus nós apagados.

Quanto ao cumprimento de tarefas pendentes, não foi desenvolvido um método que cumprisse uma tarefa específica (dado o identificador único dela) pois isso iria contra o conceito da fila dessas tarefas pendentes. Foram pensadas possíveis metodologias que respeitassem essa estrutura e que se ligassem à ordem de prioridade, todavia, foi decidido não implementá-las.

A desalocação de usuário foi julgada como relevante, tanto para a demonstração do conhecimento necessário para o feito quanto para preservação da memória em tempo de execução.

A busca binária foi implementada de modo que, uma vez dado o nome da tarefa, a lista responsável pela ordenação fosse ordenada em ordem alfabética antes da busca, permitindo a mesma. Caso não houvesse a prévia ordenação, não seria possível executá-la, pois a lista estaria sendo organizada pela ordem de prioridade.

Por fim, o nome do gerenciador foi escolhido para que houvesse uma homenagem à banda Tribalistas. Durante o desenvolvimento do trabalho, ao ser levantada a hipótese da escolha de um nome, estava tocando no ambiente a música Velha Infância, da mesma banda. Omaggio significa algo como "tributo", "presente"em italiano, e assim foi definido o nome.

6 Conclusão

O Omaggio implementa o gerenciamento de tarefas de modo a cumprir quase todos os requisitos apresentados, deixando apenas a reversão como uma funcionalidade não implementada. O gerenciamento de tarefas é funcional, assim como permite ter cadastros de mais de um usuário. Segue como anexo no arquivo .zip um diário de bordo utilizado no decorrer do desenvolvimento do trabalho, contendo registros escritos deste.