



Universidade do Minho - Escola de Engenharia
Licenciatura em Engenharia Informática - 2º ano

Laboratórios de Informática III

Fase 1

Grupo 29

Ano Letivo 2022/2023

Diogo Gomes Matos - a100741

José Paulino Ribeiro Freitas - a96140

Paula Frederika Janovska Marques - a90088



Índice

Introdução	2
Procedimento	3
Estruturação dos Dados	3
Parsing	3
Catálogo	3
Interpretação de Comandos	3
Queries	4
• Query 1	4
• Query 2	4
• Query 3	4
Encapsulamento	5
Conclusão	6



Introdução

O presente relatório serve como suporte à Fase 1 do projeto da unidade curricular Laboratórios de Informática III (LI3) do ano letivo 2022/2023, que visa alcançar os seguintes objetivos:

- Consolidação de conhecimentos essenciais da linguagem C e de Engenharia de Software, nomeadamente, modularidade e encapsulamento, estruturas dinâmicas de dados, validação funcional e medição de desempenho (computacional, consumo de memória, etc);
- Consolidação do uso de ferramentas essenciais ao desenvolvimento de projetos em C, nomeadamente, compilação, linkagem, definição de objetivos de projeto com base nas suas dependências e depuração de erros, e de gestão de repositórios colaborativos.

O grande volume de dados e cálculos necessários para o desenvolvimento do projeto, na casa dos milhões, obrigou a um planeamento prévio da estrutura e funcionamento do mesmo. Assim, foi criado um programa capaz de ler e interpretar ficheiros de dados de grande extensão, recorrendo a estruturas capazes e eficientes que permitissem a fácil manipulação dos mesmos.



Procedimento

Estruturação dos Dados

A nossa estratégia de estruturação de dados consistiu na criação de 4 diferentes estruturas, responsáveis por armazenar os dados relativos a um condutor, utilizador, viagem e estatísticas, respetivamente (*driver*, *user*, *ride* e *stat*).

Para o caso dos condutores, utilizadores e viagens estas estruturas são inicializadas durante o processo de parsing dos dados e posteriormente armazenadas em Tabelas de Hash. Já no caso das estatísticas, a estrutura é inicializada de forma diferente para queries diferentes, isto é, armazena diferentes tipos de dados para diferentes situações e inclui, por isso, variáveis para todo o tipo de estatísticas e variáveis auxiliares usadas no seu cálculo.

Parsing

Responsável pela leitura e tratamento do dataset. Foram implementadas 3 tabelas de hash diferentes especializadas para cada um dos ficheiros de dados, sendo que possuem características distintas. Para tal, foram implementadas as estruturas mencionadas acima para os condutores, utilizadores e viagens que são inicializadas individualmente para cada linha de um ficheiro. Atualmente, não aproveitamos o parsing para fazer cálculos relevantes à realização das queries estatísticas, mas é algo que planeamos fazer no futuro para uma melhoria de eficiência.

Catálogo

Criamos um catálogo (*catalog*) para fácil acesso aos dados obtidos durante o processo de parsing, isto é, uma estrutura que inclui apontadores para todas as tabelas de hash criadas e que pode ser acessada globalmente, simplificando de forma significativa o acesso aos diferentes tipos de informação, necessária para calcular certas estatísticas (por exemplo o preço de uma viagem, que requer a classe do condutor correspondente).

Considerando as vantagens fornecidas pelo facto de esta estrutura poder ser acessada globalmente e a sua natureza simples e pouco significativa, decidimos que era justificável a sua falta de encapsulamento, sendo a única em todo o projeto que segue esta estratégia.

Interpretação de Comandos

Responsável pela leitura do ficheiro de input e pela sua interpretação. Cada linha do ficheiro é analisada individualmente, sendo recolhido o número da query a ser executada e os parâmetros que esta recebe, dados que são direcionados para a função de execução correspondente (*query1()*, *query2()*, etc). Só no final da execução bem sucedida de cada query é que é analisada a próxima linha do ficheiro e assim sucessivamente até este terminar.

Queries

- Query 1

Com a implementação da estrutura de estatísticas, a primeira query foi relativamente simples de realizar. A maior dificuldade centrou-se precisamente na criação de tais estatísticas, nomeadamente na contagem do dinheiro gasto/ganho por um utilizador/conductor. Foi necessário aceder à informação do conductor que fez a viagem, nomeadamente à classe do seu veículo, que naturalmente não se encontra na tabela de hash das viagens, mas na dos condutores. Foi este tipo de situações que nos levou a introduzir um apontador para o catálogo de dados na estrutura de estatísticas, que é utilizada como transportadora de argumentos em funções predefinidas da livreria *glib* que apenas permitem a passagem de um apontador, como acontece em *stat_build()*, por exemplo.

Durante a execução da query em si, foi só preciso ter em atenção se a conta do utilizador/conductor pedida é do tipo *active* ou não (*inactive*), o que foi resolvido com uma simples condição de comparação.

- Query 2

Esta query foi a primeira a obrigar-nos a desenvolver um sistema mais eficiente, devido à sua necessidade de lidar com uma grande quantidade de dados e fazer a sua ordenação.

Neste sentido, a nossa abordagem passou por calcular uma estatística em específico para a totalidade dos utilizadores/condutores numa só passagem, em vez da totalidade das estatísticas para um utilizador/conductor como acontece na query 1. Esta abordagem foi essencial para obter um melhor desempenho e não percorrer a totalidade da tabela de hash das viagens para cada conductor.

Os dados calculados são armazenados numa temporária tabela de hash auxiliar que é depois convertida numa lista ligada do tipo *glist* e ordenada com *g_list_sort()*, utilizando uma função de comparação que tem em conta todos os parâmetros enumerados no enunciado (avaliação média, viagem mais recente e ID). A decisão do uso de uma tabela de hash neste contexto deveu-se à necessidade constante de efetuar uma procura pelo conductor atual para cada linha analisada, de forma a incrementar as suas estatísticas caso este já exista ou acrescentá-lo/acrescentá-las à tabela caso contrário.

- Query 3

A query 3 segue exatamente o mesmo raciocínio da precedente, mas tendo em conta os utilizadores em vez dos condutores e calculando a distância total de cada um deles em vez da avaliação média. Os critérios de ordenação também mudam ligeiramente, sendo o primeiro a distância total e o último o *username* dos utilizadores. Tendo a query 2 feita, foi portanto bastante simples a adaptação para a terceira.



Encapsulamento

Tivemos em conta o processo de encapsulamento ao longo do desenvolvimento do projeto, desde logo pela criação das estruturas de dados, onde separamos as estatísticas dos restantes para uma melhor organização. Por uma questão de simplicidade e eficiência, começamos por usar estruturas declaradas globalmente e por reutilizar os valores alocados no parsing dos dados e não uma cópia dos mesmos, o que requer novas alocações e libertações de memória. Este método, embora bastante eficiente, torna-se difícil de escalar e manter com o crescimento do projeto e o encapsulamento de todas as estruturas acabou por ser implementado.



Conclusão

Apesar das dificuldades, consideramos que esta fase foi bem conseguida. Após vários testes com os exemplos dados, e a verificação dos testes automáticos, verificamos que todas as queries davam os resultados expectáveis. Conseguimos também obter zero perdas de memória.

As dificuldades que encontramos centraram-se principalmente na tentativa de maximizar a eficácia e desempenho do programa. Apesar de acharmos ter conseguido um resultado satisfatório, acreditamos poder melhorar este aspeto na fase 2 do trabalho. Outro obstáculo que encontramos foi o facto de não termos entendido na totalidade como aplicar o encapsulamento no início do desenvolvimento do projeto, obstáculo esse que também pensamos ter conseguido ultrapassar com sucesso.

A resolução desta fase provocou o interesse, estudo e desenvolvimento de estruturas dinâmicas e eficientes de dados e da gestão de memória que necessitamos de assegurar. Conseguimos desafiar os nossos conhecimentos já existentes e por sua vez compreender melhor os novos conceitos que exploramos.