



Universidade do Minho

Licenciatura em Engenharia Informática

Laboratórios de Informática III

Trabalho prático - Fase 1
Grupo 13

Gonçalo Pereira (A96849) José Moreira (A95522)
Santiago Domingues (A96886)

Ano Letivo 2022/2023

Índice

1	Introdução	3
2	Problemas e soluções	4
2.1	Ficheiros, parsing e catálogos de dados	4
2.2	Batch e queries	4
2.3	Modularidade e encapsulamento	5
3	Desempenho e custo computacional	6
4	Conclusão	7

Capítulo 1

Introdução

Apresentamos, assim, o relatório referente à Fase 1 do projeto a desenvolver na Unidade Curricular de Laboratórios de Informática III. Todo este documento serve como suporte ao código desenvolvido e aborda assuntos como: linguagem de programação C, modularidade e encapsulamento, volumes de dados, problemas e respetivas soluções, memória, entre outros. Com uma abordagem geral, o projeto sugerido pelos docentes desafia ao desenvolvimento de uma aplicação, em C, que tenha a capacidade de guardar, manipular e realizar "queries" sob o conteúdo de ficheiros extensos ".csv". Com isto, é esperado um projeto fiável que, enfrentando grandes quantidades de dados, consiga responder às exigências sem que haja perda de dados e desempenho. Nesta fase do trabalho prático, foi proposto o desenvolvimento das seguintes etapas: parsing dos dados de entrada (divididos por três ficheiros ".csv"), modo de operação batch, catálogos de dados para os três ficheiros e 1/3 das queries propostas completamente funcionais. Em consequência, o atual projeto visa desenvolver as capacidades na área de Engenharia de Software, refinando as capacidades nos temas de desempenho de código, capacidade no uso de ferramentas como compilação, linkagem, depuração de erros (debug), desenvolvimento de Makefile e gestão de repositórios comuns (pelo uso do GitHub) e, de uma forma óbvia, consolidação de conhecimentos essenciais dentro da linguagem de programação C. Desta forma, inicia-se a discussão detalhada de todos os parâmetros decisivos na realização da primeira fase do projeto da presente UC.

Capítulo 2

Problemas e soluções

2.1 Ficheiros, parsing e catálogos de dados

Inicialmente, iniciou-se uma discussão em grupo sobre as estruturas de dados que acolheriam os ficheiros ".csv": "drivers.csv", "users.csv" e "rides.csv". Uma vez que todos os ficheiros possuem semântica e composição idênticas, decidiu-se criar as estruturas DRIVER, USER e RIDE, representando, cada uma delas, uma linha (exceto o cabeçalho) do ficheiro ".csv" correspondente. Desta forma, cada uma das estruturas é composta pelos parâmetros presentes nos ficheiros sendo especialmente representados pelos tipos de dados char* e int. Porém, a discussão continuou com o objetivo de se decidir de que forma é que essas estruturas seriam armazenadas "conjuntamente", de acordo com o seu ficheiro de origem. De forma a resolver este problema, desenvolveu-se as estruturas DRIVERS, USERS e RIDES que são, tal como o nome indica, organizações coletivas das estruturas respetivas anteriormente referidas, ou seja, catálogos que, neste caso, são representadas por arrays dinâmicos. Cada um destes arrays possui um apontador para o tipo de estrutura que acolhe, ou seja, DRIVER*, USER* e RIDE*, um valor inteiro length e um valor inteiro size. Desta forma, o grupo conseguiu ultrapassar a sua primeira dificuldade e, de um certo modo, resolver, com sucesso, dois dos quatro desafios lançados nesta primeira fase do projeto: parsing dos dados de entrada e catálogos de dados.

2.2 Batch e queries

Por outro lado, começou a surgir a necessidade de manipular todas as estruturas de dados que já se encontravam guardadas em memória, de forma a responder-se aos outros dois desafios propostos no guião do projeto: modo de operação batch e realização de queries (1/3). Deste modo, o grupo empenhou-se em desenvolver código que se responsabilizasse por aceder ao ficheiro de texto onde estão especificadas as queries e realizar a query respetiva aquando a leitura individual da linha referente presente nesse mesmo ficheiro. Nesta primeira fase, o grupo decidiu desenvolver o código solução para as queries 4, 5 e 6, não precisando assim de desenvolver algum tipo de estrutura que otimizasse a realização das mesmas, uma vez que a complexidade linear implementada no processo não representa um perigo ao desempenho do programa. Porém, em ponderação conjunta, percebe-se a necessidade de implementação de outro tipo de estruturas que ofereçam uma grande otimização no desempenho do programa durante uma segunda fase deste mesmo projeto. Apesar de não se encontrarem, atualmente, implementadas, todas estas estruturas referidas possuirão a devida ponderação e serão implementadas numa fase mais avançada do trabalho prático.

2.3 Modularidade e encapsulamento

Como já referido anteriormente, a modularidade e o encapsulamento apresentam-se como características fulcrais no desempenho perfeito de todo o trabalho prático. O desenvolvimento do projeto com estes temas em mente permite que o código seja mais flexível, seguro e fiável, uma vez que há um controlo exemplar dos acessos a cada porção do mesmo. Assim, sendo uma das grandes preocupações do grupo, este tema foi, provavelmente, o tema que necessitou de mais atenção. De forma a desenvolver todo o código em módulos, encapsulando-o de uma forma correta, decidiu-se criar vários ficheiros .c responsáveis apenas por cada uma das tarefas individuais atribuídas. Deste modo, surgiram os ficheiros responsáveis por guardar cada estrutura individualmente: `driver.c`, `user.c` e `ride.c`; ficheiros responsáveis por guardar estruturas coletivas: `cat_drivers.c`, `cat_users.c` e `cat_rides.c`; ficheiros de queries e auxiliares às mesmas: `queries.c` e `queries_aux.c`; ficheiros compostos por funções úteis ao desenvolvimento de todo o código: `util.c`; e ficheiros de carácter mais global, responsáveis por reunir, funcionalmente, todos os outros: `parsing.c` e `main.c`. Por outro lado, para cada um destes ficheiros (à exceção do ficheiro `main.c`), foi desenvolvido um header file, ou seja, `file_name.h`, responsável por acolher cada função desenvolvida no .c referente e responsável, também, por incluir, através de `#include`, todos os ficheiros com funções úteis ao mesmo. Em último lugar, em cada um dos ficheiros responsáveis pelas estruturas (`driver.c`, `cat_drivers.c`, `user.c`, `cat_users.c`, `ride.c`, `cat_rides.c`), foram escritas várias funções de carácter `get` que permitem que seja feito o acesso seguro aos diversos campos das estruturas guardadas nos mesmos (`DRIVER`, `DRIVER`, `USER`, `USERS`, `RIDE`, `RIDES`).

Capítulo 3

Desempenho e custo computacional

De uma forma geral, o projeto desenvolvido até à data mostra-se eficaz em termos de desempenho e custo computacional. Como já foi referido anteriormente, a realização das queries escolhidas pelo grupo (4, 5 e 6) não pedem um grande esforço computacional da máquina e conseguem ser realizadas em tempos muito baixos sem que haja uma grande quantidade de memória a ser utilizada em simultâneo. Deste modo, de uma forma mais objetiva, o grupo decidiu testar minuciosamente os tempos de execução de cada query e o uso de RAM e de CPU aquando a execução do programa. Para isto, foi usada a máquina ASUS TUF Gaming F15 com as seguintes especificações: processador Intel Core i9-11900H, placa gráfica NVIDIA GeForce RTX 3060, armazenamento 1 TB SSD e memória RAM 16GB DDR4. Todos os testes foram feitos com o cabo de alimentação ligado e foram então obtidos os seguintes resultados: query 4 - 0.052789 segundos, query 5 - 0.124625 segundos, query 6 - 0.063750 segundos. Em termos de memória RAM, houve uma ligeira subida de 5.2 GB para 5.8 GB (31.4% para 35.1%) e, relativamente a desempenho de CPU, notou-se a utilização a praticamente 100% (98%) de uma thread (entre 16).

Capítulo 4

Conclusão

Concluindo, o grupo mostra-se bastante satisfeito com todo o trabalho desenvolvido e explicado neste relatório do projeto. Em primeiro lugar, todos os desafios lançados no guião, referentes à primeira fase do projeto, foram realizados na totalidade. Por outro lado, o grupo mostra-se feliz em poder afirmar que todos esses desafios foram realizados nunca esquecendo os critérios propostos. Houve uma aprendizagem significativa em termos de programação na linguagem C, os conceitos de modularidade e encapsulamento foram abordados da devida forma e todos os métodos de trabalho com estruturas de dados, acessos a memória, etc. foram devidamente utilizados. Como referido na introdução do respetivo relatório, o uso de ferramentas como compilação, linkagem, debug e gestão de repositórios comuns forneceram as devidas capacidades que permitiram que o trabalho fosse realizado com sucesso e, de uma forma mais importante, em grupo. Deixando todas as expectativas altas para a realização da Fase 2 do projeto, dá-se como terminado o relatório da primeira fase do projeto a desenvolver na UC de Laboratórios de Informática III, no presente ano letivo de 2022/2023.