Guião I

Laboratórios de Informática III

2021/2022

Objetivos

- Consolidação de conhecimentos essenciais da linguagem C, nomeadamente, operações sobre ficheiros, parsing de dados e gestão de memória;
- Consolidação do uso de ferramentas essenciais ao desenvolvimento de projetos em C, nomeadamente, compilação, linkagem e depuração de erros e de gestão de repositórios colaborativos;
- Preparação do trabalho que será realizado em fases posteriores do projeto.

Realização e avaliação do trabalho desenvolvido

- O trabalho desenvolvido por cada grupo será avaliado com base no ramo “guiao-1” do respetivo repositório GitHub, à data de 7 de novembro (23:59);
- O relatório do trabalho (máximo 3 páginas de conteúdo, ou seja, sem capas) terá de ser disponibilizado dentro da pasta “guiao-1” até 10 de novembro (23:59), em formato PDF, e o ficheiro correspondente terá de ter o nome “relatorio.pdf”. O relatório deverá centrar-se na resolução dos exercícios, identificando as estratégias seguidas e eventuais limitações;
- Esta fase do trabalho a realizar na unidade curricular de Laboratórios de Informática III terá um peso de 10% na avaliação final. A avaliação desta fase será composta por 50% do exercício 1 + 30% do exercício 2 + 20% do relatório;
- O trabalho terá de ser desenvolvido por todos os elementos do grupo de trabalho e todos deverão registar as suas contribuições individuais no respetivo repositório;
- O projeto terá de gerar o necessário ficheiro executável com base na preparação de um ficheiro “Makefile” (dentro da pasta “guiao-1”) e por invocação do comando “make”. Da mesma forma, deverá limpar todos os ficheiros desnecessários ao projeto através da execução do comando “make clean”;
- O executável deverá assumir a existência e ficheiros de entrada com nomes “users.csv”, “commits.csv” e “repos.csv” dentro de uma sub-pasta “entrada” da pasta “guiao-1” e deverá gerar os ficheiros com os registos validados dentro de uma sub-pasta “saida” (sem acentuação);
- O trabalho realizado por cada um dos grupos será parcialmente avaliado com base em testes de execução automática do programa executável, pelo que é imprescindível serem observados os nomes de todos os ficheiros e pastas identificados neste enunciado.

Descrição dos ficheiros necessários

- users.csv:
    - 3 milhões de registos (máximo)
    - public_repos, id, followers, following, public_gists: inteiro não negativo
    - follower_list, following_list: lista de inteiros não negativos
    - created_at: data e hora (no formato AAAA-MM-DD hh:mm:ss)
    - type: enumerado (Bot, Organization, User)
    - login: string
- commits.csv:
    - 10 milhões de registos (máximo)
    - repo_id, committer_id, author_id: inteiro não negativo
    - commit_at: data e hora (no formato AAAA-MM-DD hh:mm:ss)
    - message: string
- repos.csv:
    - 10 milhões de registos (máximo)
    - license, description, language, full_name, default_branch: string
    - created_at, updated_at: data e hora (no formato AAAA-MM-DD hh:mm:ss)
    - forks_count, open_issues, stargazers_count, owner_id, id, size: inteiro não negativo
    - has_wiki: Bool

Exercícios

1. Dados os ficheiros de dados disponíveis na BB, escreva um programa na linguagem C que leia cada um dos ficheiros (em formato CSV), descarte os registos que não respeitem os formatos dos respectivos campos (identificados acima) e gere ficheiros de saída contendo exclusivamente os registos válidos.

   Além de respeitar o formato, deverá também validar se os registos estão corretos, segundo o seguinte conjunto de normas:

   a. Os campos de data correspondem a uma data válida e não referem uma data no futuro nem prévia a 7 de Abril de 2005.
   b. Os campos *following* e *followers* apresentam exatamente o valor do comprimentos das listas *following_list* e *follower_list*, respetivamente.

   Os ficheiros de saída devem apresentar exatamente a mesma estrutura (ordem dos campos, delimitadores, etc) dos ficheiros de entrada. Os nomes dos ficheiros de saída terão de ser, respetivamente, “users-ok.csv”, “commits-ok.csv” e “repos-ok.csv”. O programa deverá ser invocado como “./guiao-1 exercicio-1”.

2. Utilizando agora os novos ficheiros gerados no exercício 1, escreva um programa em C que efetue o cruzamento de dados entre os vários ficheiros, filtrando dados inválidos. Desta forma, pretende-se que este novo programa produza ficheiros CSV, tendo em conta as seguintes filtragens:

    a. Remoção de commits que refiram utilizadores (*committer_id*, *author_id*) inexistentes;
    b. Remoção de *commits* cujo repositório não exista;
    c. Remoção de repositórios que refiram utilizadores (*owner_id*) inexistentes;
    d. Remoção de repositórios que não contenham qualquer *commit*.

    Os nomes dos ficheiros de saída terão de ser, respetivamente, “users-final.csv”, “commits-final.csv” e “repos-final.csv”. O programa deverá ser invocado como “./guiao-1 exercicio-2”.

    Nota: Assume-se como “existente” um elemento referenciado e corretamente registado no respetivo ficheiro.
    Exemplo: Supondo que existe um registo no ficheiro “commits.csv” com um valor X para o campo *commiter_id* e que não existe um registo em “users.csv” com campo *id* com esse valor, então o registo de commit deve ser descartado.