

TRABALHO PRÁTICO PARTE 1

PRAZO PARA ENTREGA: 22/11/2021

1. Introdução

O TikTok é um aplicativo bastante popular que permite aos usuários compartilhar vídeos curtos. O aplicativo para Android pode ser baixado pela Google Play. A base de dados disponível em <https://www.kaggle.com/shivamb/35-million-tiktok-mobile-app-reviews> [1] reúne informações de aproximadamente 3,5 milhões de *reviews* de usuários do aplicativo na loja da Google. Esses dados podem ser encontrados no link supracitado, no arquivo *tiktok_app_reviews.csv*. Cada registro do arquivo é composto das seguintes informações:

- `review_id`: string identificadora única
- `review_text`: o *review* postado pelo usuário
- `upvotes`: número de votos favoráveis ao *review*
- `app_version`: string indicando a versão do aplicativo a que o *review* se refere
- `posted_date`: data e hora de postagem

2. Objetivos

O objetivo principal deste trabalho é dar ao aluno a oportunidade de manipular arquivos através de uma base real de dados. Espera-se que o aluno seja capaz de:

- manipular adequadamente arquivos em modo texto;
- definir estruturas de dados para representar as entidades;
- gravar e manipular registros em arquivos binários.

3. Desenvolvimento

As etapas para o desenvolvimento do trabalho são descritas abaixo.

Etapas 1: Processamento dos dados

Sua primeira tarefa será fazer a leitura dos dados do arquivo *tiktok_app_reviews.csv* de forma correta, armazenando-os em estruturas de dados apropriadas. Você é livre para escolher a forma de representação dos seus dados, mas lembre-se de que você será avaliado em função da qualidade das suas escolhas. Na sequência, você deverá salvar esses conjuntos de registros em um arquivo binário intitulado *tiktok_app_reviews.bin*.

Etapa 2: Importação de Registros e Módulo de Teste

Desenvolver uma função que deverá importar para a memória principal conjuntos de N registros aleatórios do arquivo binário, armazenados em um arranjo de registros. Deve ser implementado também um módulo de testes simples, que realiza algumas operações para garantir que as funções básicas do programa entregam resultados corretos. Esse módulo deve conter as seguintes funções:

- `acessaRegistro(i)`: acessa diretamente o i-ésimo registro do arquivo binário e o imprime na tela. O valor de i deve ser fornecido pelo usuário.
- `testeImportacao()`: importa N registros aleatórios do arquivo binário. Para essa importação, a função deve perguntar ao usuário se ele deseja exibir a saída no console ou salvá-la em um arquivo texto. Se a opção for a saída em console, deve-se utilizar N = 10. Para a saída em arquivo, utilize N = 100.

Note que a função `testeImportacao()` é apenas uma função de teste. O programa deve ser capaz de importar registros do arquivo para quaisquer valores de N, sem erros e sem gerar exceções. Essa opção deve estar disponível a partir de um menu no console, que permitirá escolher executar o módulo de testes ou a importação para um dado valor de N.

4. Relatório parcial

Você deverá confeccionar um relatório simples de no máximo uma página sobre o trabalho desenvolvido. Este relatório deve conter, obrigatoriamente, os seguintes itens:

- detalhamento das atividades realizadas por cada membro do grupo;
- decisões de implementação;
- toda e qualquer referência utilizada no desenvolvimento do trabalho.

Note que esse relatório, embora simples, deve ser formal e bem redigido. A divisão de tarefas do trabalho se aplica somente à implementação. Todo o grupo é responsável pelo relatório (escrita e revisão).

5. Exigências

O trabalho deverá, obrigatoriamente, atender aos seguintes requisitos:

- Implementação em C ou C++
- O projeto deve ser compilável e executável via linha de comando. Não conte com a presença de IDEs como Code::Blocks ou Visual Studio. Caso seu grupo opte por utilizar algum ambiente de desenvolvimento, certifique-se de que o projeto enviado possa também ser facilmente compilado em um sistema operacional Linux sem esses ambientes instalados. Forneça instruções claras e precisas de compilação e execução pela linha de comando. **Recomenda-se a utilização de algum Makefile**

ou **script para a compilação**. Caso o grupo julgue necessário, é possível solicitar que o professor verifique as instruções de compilação **antes** do prazo final de envio.

- **Nenhum trabalho será aceito após o prazo.**
- **O programa desenvolvido deve permitir que o usuário entre com o caminho do diretório que contém os arquivos de dados como um argumento na linha de comando.** Veja o Exemplo:

```
$ ./programa /diretorio/contendo/arquivos/entrada
```

- O programa deve procurar pela existência dos arquivos binários dentro da pasta. Se não existir, deve pré-processá-los (etapa 1) antes de seguir para o módulo de testes (etapa 2).
- Não é permitida a utilização de bibliotecas externas. O processamento dos arquivos deve ser implementado pelo grupo. Um dos objetivos desta etapa do trabalho é que vocês aprendam a trabalhar com arquivos. Caso haja dúvida quanto à utilização de alguma função ou biblioteca, entre em contato com o professor.
- Obviamente, todo código deve ser de autoria do grupo. Não é permitida a utilização de códigos de terceiros ou de outros grupos. É permitida a pesquisa por estratégias para a solução dos problemas (e as referências utilizadas nessas pesquisas devem constar do relatório), porém a apropriação de código alheio não será aceita. **Qualquer tentativa de plágio identificada resultará em nota zero. Os códigos fontes serão analisados pelo sistema Moss (<http://theory.stanford.edu/~aiken/moss/>)**

6. Entrega

O grupo deverá ser formado por **no mínimo 3** e **no máximo 4** alunos, e as responsabilidades de cada aluno devem ser documentadas e registradas. Não é permitido que algum integrante do grupo fique responsável somente pela confecção do relatório. Todos os integrantes devem contribuir com a implementação. A distribuição das responsabilidades deve ser feita de maneira uniforme, de modo que cada membro do grupo se envolva com o trabalho na mesma proporção que os demais.

Todos os itens abaixo devem ser entregues:

1. Código-fonte completo;
 - a. Deve ser submetido um link para um repositório git (github) contendo o código do trabalho;
 - b. **Não incluir o dataset na submissão via Google Classroom nem no github (nem o .csv nem o .bin).**
2. Relatório em Google Docs ou pdf atendendo ao especificado na Seção 4 deste documento.

7. Critérios de avaliação

O grupo será avaliado de acordo com os seguintes critérios:

- Execução correta do código (E);
- Desempenho do programa (D): evitar o consumo excessivo de memória, e o tempo de execução deve estar dentro de limites aceitáveis para o problema tratado;
- Atendimento ao que foi solicitado (A) (valor entre 0 e 1);
- Organização do código (O): seu código deve estar bem modularizado e bem documentado;
- Qualidade do relatório apresentado (R).

Cada membro do grupo será avaliado individualmente, tanto com relação aos detalhes de implementação que ficaram sob sua responsabilidade, quanto ao entendimento em alto nível de abstração do que foi feito pelo grupo como um todo. O entendimento teórico do conteúdo relacionado ao trabalho também será avaliado. A nota individual (M) será um valor de 0 a 1 que irá ponderar a nota da implementação.

A nota final de cada integrante será computada de acordo com a seguinte fórmula:

$$\text{Nota} = M \cdot (A \cdot (E + D) + O) + R$$

O critério (A) será considerado de acordo com a composição do grupo. O objetivo é evitar que algum aluno seja prejudicado pela desistência de outros membros do grupo. **Caso algum membro do seu grupo tranque ou abandone a disciplina, comunique o professor o quanto antes para que se possa discutir alternativas.**

Referências

[1] BANSAL, Shivam. 3.5 Million Tiktok Mobile App Reviews (2021). Disponível em: <https://www.kaggle.com/shivamb/35-million-tiktok-mobile-app-reviews> (acessado em 25 de outubro de 2021).