

**Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação
Disciplina de Organização de Arquivos
Profa. Dra. Cristina Dutra de Aguiar Ciferri
PAE Jaqueline Joice Brito (Turma A)
PAE Lucas de Carvalho Scabora (Turma B)**

Segundo Trabalho Prático

Turma B

A solução deste trabalho prático deve ser proposta exclusivamente pelo grupo com base nos conhecimentos adquiridos ao longo das aulas. Consulte as notas de aula e o livro texto quando necessário.

Grupo do Trabalho

Este trabalho prático deve ser feito pelo mesmo grupo do primeiro trabalho prático. Entretanto, alunos que tiveram problemas com seus grupos no primeiro trabalho prático devem procurar a docente para que as devidas providências sejam tomadas.

Descrição do Trabalho

Considere os seguintes dados a respeito de um *tweet*:

TEXT (i.e., texto do *tweet* – tipo char)

USER (i.e., nome do usuário que postou o *tweet* – tipo char)

COORDINATES (i.e., coordenadas do usuário que postou o *tweet* – tipo char)

FAVORITE_COUNT (i.e., quantidade de marcações como favorito – tipo inteiro)

LANGUAGE (i.e., língua – tipo char)

RETWEET_COUNT (i.e., quantidade de vezes em que foi retweetado – tipo inteiro)

VIEWS_COUNT (i.e., quantidade de visualizações do *tweet* – tipo long)

Considere o tipo de arquivo de dados com **campos de tamanho variável com delimitadores entre campos e registros de tamanho variável com indicadores de tamanhos**.

Implemente um programa que ofereça as funcionalidades descritas a seguir.

[1] Ofereça uma interface por meio da qual seja possível fornecer os dados de *tweets*.

[2] Permita a gravação de registros a partir de dados de entrada do teclado até que uma condição de parada seja detectada. A condição de parada deve ser determinada pelo grupo.

[3] Permita a recuperação dos dados, de todos os registros, armazenados no arquivo de dados, mostrando os dados de forma organizada na saída padrão para permitir a distinção dos campos e registros. Deve-se permitir a visualização dos registros, um por vez.

[4] Permita a recuperação dos dados de todos os registros com base em um valor do campo USER por meio da busca sequencial.

[5] Permita a recuperação dos dados de um ou mais registros com base no valor do campo FAVORITE_COUNT por meio de um índice secundário fortemente ligado. A duplicação de chaves deve ser obrigatoriamente tratada por meio de listas invertidas. Também considere que o índice somente será criado depois que o arquivo de dados contiver, pelo menos, 10 registros.

[6] Permita a recuperação dos dados de um ou mais registros com base no valor do campo LANGUAGE por meio de um índice secundário fortemente ligado. A duplicação de chaves deve ser obrigatoriamente tratada por meio de listas invertidas. Também considere que o índice somente será criado depois que o arquivo de dados contiver, pelo menos, 10 registros.

[7] Permita a recuperação dos dados de um ou mais registros com base nos valores dos campos FAVORITE_COUNT e LANGUAGE simultaneamente, usando os índices secundários fortemente ligados criados nos itens [5] e [6]. Por exemplo, essa funcionalidade deve permitir consultas do tipo: “Encontre todos os registros de dados tal que FAVORITE_COUNT seja 10 e que LANGUAGE seja português”. Para oferecer suporte para essa funcionalidade, o grupo deve implementar eficientemente a operação cosequencial *matching*.

[8] Permita a recuperação dos dados de um ou mais registros com base nos valores dos campos FAVORITE_COUNT ou LANGUAGE simultaneamente, usando os índices secundários fortemente ligados criados nos itens [5] e [6]. Por exemplo, essa funcionalidade deve permitir consultas do tipo: “Encontre todos os registros de dados tal que FAVORITE_COUNT seja 10 ou que LANGUAGE seja português”. Para oferecer suporte para essa funcionalidade, o grupo deve implementar eficientemente a operação cosequencial *merging*.

[9] Permita a remoção lógica de um registro usando como base o valor do campo FAVORITE_COUNT. Desde que pode o mesmo autor pode estar presente em mais do que um registro, o registro a ser removido deve ser determinado da seguinte forma. Primeiramente, usando o índice secundário criado no item [5], devem ser recuperados todos os registros que satisfaçam à condição de busca. Esses registros devem ser exibidos na tela. A partir dessa exibição, o usuário deverá escolher apenas um registro a ser removido.

[10] Permita a inserção de registros adicionais, com reaproveitamento dinâmico do espaço de registros logicamente removidos. Deve ser adotada a estratégia de alocação *best-fit* com tratamento da fragmentação interna.

Restrições

As seguintes restrições têm que ser garantidas no desenvolvimento do trabalho.

[1] Os arquivos de dados devem ser gravados em disco no **modo binário**. O modo texto não deve ser usado.

[2] Os dados do registro descrevem os nomes dos campos, os quais não podem ser alterados. Ademais, todos os campos devem estar presentes na implementação, e nenhum campo adicional pode ser incluído. O tamanho de cada campo deve ser determinado pelo grupo.

[3] Os dados a serem inseridos no arquivo e utilizados nas buscas devem refletir dados do mundo real. Dica: procure na internet por geradores de bancos de dados sintéticos (ou desenvolva o seu).

[4] Todos os campos devem representar campos de chave secundária, ou seja, os campos podem armazenar valores repetidos de dados.

[5] Todo código fonte deve ser documentado. A documentação inclui, dentre outros, a documentação de procedimentos, de funções, de variáveis, de partes do código fonte que realizam tarefas específicas. Ou seja, o código fonte deve ser documentado tanto em nível de rotinas quanto em nível de variáveis e blocos funcionais. Ademais, os integrantes do grupo devem constar como comentário no início do código (i.e., NUSP e nome de cada integrante do grupo). Não será atribuída nota ao aluno cujos dados não constarem no código fonte.

[6] A interface pode ser feita em modo texto (terminal) ou modo gráfico e deve ser funcional.

[7] A implementação deve ser realizada usando a linguagem de programação C. As funções das bibliotecas `<stdio.h>` devem ser utilizadas para operações relacionadas à escrita e leitura dos arquivos. A implementação não deve ser feita em qualquer outra linguagem de programação. O programa deverá compilar no GCC versão 4.8.2 ou superior. O trabalho será corrigido no Linux.

Fundamentação Teórica

Conceitos e características dos diversos métodos para representar os conceitos de campo e de registro em um arquivo de dados podem ser encontrados nas transparências de sala de aula e também nas páginas 96 a 107 do livro *File Structures (second edition)*, de Michael J. Folk e Bill Zoellick.

Material para Entregar

[1] Código Fonte do Programa. Enviar por email para o aluno PAE da disciplina o arquivo compactado contendo o código fonte do programa, e as bibliotecas necessárias para a execução do programa. A mensagem deve ser enviada para o em-mail labbdciferri@gmail.com e deve ter como assunto: [Organização de Arquivos] Trabalho Prático 2 – Turma B. Deve constar no corpo da mensagem o NUSP e nome de cada integrante do grupo. Não será atribuída nota ao aluno cujos dados não constarem no corpo da mensagem. O código fonte do programa deve ser entregue **apenas por email**.

[2] Documentação Externa. A documentação externa deve conter a descrição dos principais conceitos usados no trabalho prático, as decisões de projeto e as suas justificativas (ex.: estruturas de dados e algoritmos usados), assim como qualquer outra consideração adicional assumida no desenvolvimento do trabalho prático. Em detalhes, a documentação externa deve possuir:

- CAPA, com as seguintes informações: o nome da instituição, o nome do curso, o nome da disciplina, o nome do professor responsável, o nome do trabalho prático, o nome dos participantes e os respectivos números USP, e a data de entrega do trabalho prático.
- ÍNDICE, listando os nomes das seções que compõem o trabalho prático e as suas respectivas páginas de início.
- SEÇÃO 1: Descrição dos campos dos registros, contendo os nomes dos campos, os tamanhos dos campos e um desenho que mostra visualmente a estrutura dos registros. A escolha dos tamanhos dos campos deve ser justificada.
- SEÇÕES 2 a N: Quaisquer outras decisões de projeto. Em detalhes, a documentação referente a essas seções deve conter a descrição dos principais conceitos usados no trabalho prático, incluindo desenhos que facilitem a compreensão das estruturas de dados, as decisões de projeto e as suas justificativas, assim como qualquer outra consideração adicional assumida no desenvolvimento do trabalho prático. Todas as funcionalidades do programa devem ser descritas em detalhes.
- SEÇÃO N+1: Cópias de telas da interface.
- SEÇÃO N+2: Baterias de testes, as quais devem gerar informações que permitam acompanhar a execução do programa
- REFERÊNCIAS BIBLIOGRÁFICAS, caso necessário.

A documentação externa deve ser entregue **impressa** e por **email**.

[3] Checklist.

Enviar por e-mail um arquivo zip contendo:

- Código fonte do programa.
- Makefile para a compilação do programa.
- Bibliotecas necessárias para a execução do programa.
- Documentação externa em formato PDF.

Entregar de forma impressa:

- Documentação externa em formato PDF.

Critério de Correção

Critério de Avaliação do Trabalho

Na correção do trabalho, serão ponderados os seguintes aspectos.

- [1] Qualidade da documentação (interna e externa) entregue. MAIOR PESO
- [2] Corretude da execução do programa. MAIOR PESO
- [3] Qualidade da interface.

Adicionalmente,

- A não execução de um programa devido a erros de compilação implica que a nota final do trabalho prático será igual a zero (0).
- A ausência da documentação interna implica que haverá uma diminuição expressiva na nota do trabalho prático.
- A ausência da documentação externa implica que haverá uma diminuição expressiva na nota do trabalho prático.
- A inserção de palavras ofensivas nos arquivos e em qualquer outro material entregue implica que a nota final do trabalho prático será igual a zero (0).
- Em caso de cola, as notas dos trabalhos envolvidos será igual a zero (0).
- Devem ser exibidos avisos ou mensagens de erro quando apropriado.

Critério de Avaliação dos Integrantes

Na data da segunda avaliação da disciplina, será realizada uma avaliação individual sobre o trabalho desenvolvido. Essa avaliação será composta de 2 perguntas sobre o projeto. Cada pergunta respondida de forma incorreta acarretará no decréscimo de 25% da nota do integrante do grupo no projeto. A realização dessa avaliação é obrigatório para cada um dos integrantes do grupo. Integrantes que não realizarem essa avaliação terão a nota do projeto igual a 0,0.

Data de Entrega do Trabalho

Na data especificada na página da disciplina.

Bom Trabalho !