

**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 Fábio Felix Dias (Turma A)
PAE Anderson Chaves Carniel (Turma B)**

**Segunda Parte do Trabalho Prático (Parte II)
Valor: 70%**

A solução para este 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.

O trabalho deve ser feito considerando-se o mesmo grupo da Parte I do trabalho prático, com exceção dos grupos cujos alunos foram autorizados pela docente para participarem de outros grupos.

Descrição do arquivo de dados

Cada registro do arquivo de dados deve conter dados relacionados a:

Turma A. Dados dos domínios governamentais registrados no Registro.br (coletado em <http://dados.gov.br/dataset/dominios-gov-br>), de forma que cada registro indique um domínio governamental.

- Campos de tamanho fixo: *ticket* (número do ticket vinculado ao registro da URL no cadastro do Registro.br), *documento* (número do documento informado ao Registro.br no cadastro da URL), *dataHoraCadastro* (data e hora de quando a URL foi cadastrada), *dataHoraAtualiza* (data e hora de quando a URL foi modificada).
- Campos de tamanho variável: *dominio* (URL cadastrada no Registro.br), *nome* (nome do Órgão ou Entidade responsável pela URL), *cidade* (cidade cadastrada no Registro.br), *uf* (nome do estado por extenso).

Turma B. Dados cadastrais de companhias abertas no Brasil (coletado em <http://dados.gov.br/dataset/cias-abertas>), de forma que cada registro indique uma companhia aberta.

- Campos de tamanho fixo: *CNPJ* (CNPJ de cadastro da companhia), *dataRegistro* (data em que a companhia foi registrada sob o CNPJ), *dataCancelamento* (data em que a companhia foi fechada), *CNPJauditor* (CNPJ da empresa que presta auditoria para a companhia).
- Campos de tamanho variável: *nomeSocial* (nome no qual foi registrado o CNPJ), *nomeFantasia* (nome fantasia da companhia), *motivoCancelamento* (razão pela qual a companhia foi fechada), *nomeEmpresa* (nome da empresa que presta auditoria).

Organização do arquivo de dados

Deve ser considerada a *organização híbrida de campos e registros*, da seguinte forma:

Turma A.

- Campos de tamanho fixo e campos de tamanho variável. Para os campos de tamanho variável, deve-se usar o método *indicador de tamanho*.
- Registros de tamanho variável. Para os registros de tamanho variável, deve-se usar o método *delimitadores entre registros*.

Turma B.

- Campos de tamanho fixo e campos de tamanho variável. Para os campos de tamanho variável, deve-se usar o método *delimitadores entre campos*.
- Registros de tamanho variável. Para os registros de tamanho variável, deve-se usar o método *delimitadores entre registros*.

Importante.

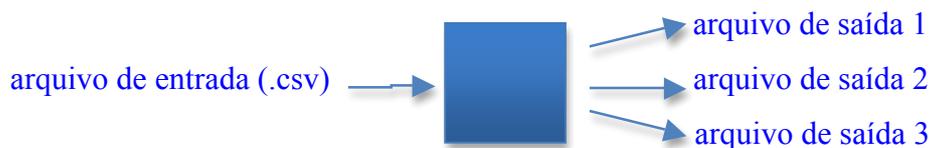
A forma de armazenamento da organização híbrida deve ser respeitada. Isso significa que os grupos devem garantir o armazenamento correto requerido pela organização híbrida.

Programa

Implemente um programa em C que ofereça as seguintes funcionalidades.

[1] Ofereça uma interface por meio da qual o usuário possa escolher a funcionalidade a ser realizada.

[2] Permita a leitura de vários registros obtidos a partir de um arquivo de entrada e a gravação desses registros em três arquivos de dados de saída. O arquivo de entrada será fornecido juntamente com a especificação do projeto, enquanto que o arquivo de dados de saída deverá ser gerado como parte deste trabalho prático. Note que essa funcionalidade é a mesma que a solicitada na Parte I do Trabalho Prático. Porém, na Parte II do Trabalho Prático, devem ser gerados três arquivos de dados de saída. Ou seja,



Note que os arquivos de saída gerados na funcionalidade 2 serão iguais.

[3] Para cada arquivo de dados de saída, crie um índice primário sobre o campo *ticket* (Turma A) ou sobre o campo **CNPJ** (Turma B). Considere que o índice somente deve ser criado depois que os arquivos de dados de saída estiverem criados. Ou seja,

- para o arquivo de saída 1, crie um índice primário 1 sobre o campo *ticket* (Turma A) ou sobre o campo **CNPJ** (Turma B)

- para o arquivo de saída 2, crie um índice primário 2 sobre o campo *ticket* (Turma A) ou sobre o campo **CNPJ** (Turma B)
- para o arquivo de saída 3, crie um índice primário 3 sobre o campo *ticket* (Turma A) ou sobre o campo **CNPJ** (Turma B)

Note que os arquivos de índice gerados na funcionalidade 3 serão iguais.

[4] Permita a remoção lógica de registros dos arquivos de dados de saída. Essa funcionalidade deve ser desenvolvida da seguinte forma. Primeiramente, o usuário deve indicar um valor para o campo *ticket* (Turma A) ou para o campo **CNPJ** (Turma B). Na sequência, para cada arquivo de dados de saída deve-se: (i) buscar o registro desejado usando-se o índice primário correspondente; (ii) realizar a remoção lógica do registro desejado no arquivo de dados; (iii) realizar a remoção física da entrada correspondente no arquivo de índice; e (iv) mostrar uma mensagem informando o sucesso ou não da remoção.

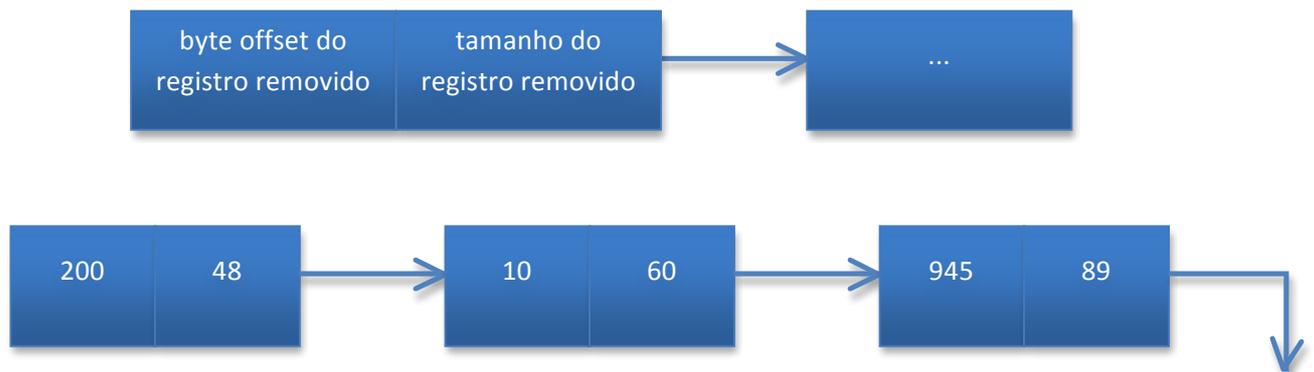
[5] Permita a inserção de registros adicionais, com reaproveitamento dinâmico do espaço de registros logicamente removidos. As seguintes estratégias de alocação devem ser usadas:

- para o arquivo de saída 1, deve-se usar a estratégia de alocação *first-fit*, sem ordenação da lista de registros removidos, com tratamento da fragmentação interna, e sem realizar a junção de espaços vazios adjacentes
- para o arquivo de saída 2, deve-se usar a estratégia de alocação *best-fit*, com ordenação crescente da lista de registros removidos, com tratamento da fragmentação interna, e sem realizar a junção de espaços vazios adjacentes
- para o arquivo de saída 3, deve-se usar a estratégia de alocação *worst-fit*, com ordenação decrescente da lista de registros removidos, com tratamento da fragmentação interna, e sem realizar a junção de espaços vazios adjacentes.

Para cada registro a ser inserido, essa funcionalidade deve ser desenvolvida da seguinte forma. Primeiramente, o usuário deve indicar o registro a ser inserido. Na sequência, para cada arquivo de dados de saída deve-se: (i) inserir o registro desejado no arquivo de dados; (ii) inserir sua entrada correspondente no arquivo de índice; e (iii) mostrar uma mensagem informando o sucesso ou não da remoção.

[6] Permita a visualização de estatísticas sobre os arquivos de índice. Essa funcionalidade deve exibir uma tabela que mostra, para cada índice primário, a quantidade de entradas que ele possui. Também deve ser permitida a exibição dos registros dos três índices simultaneamente, possibilitando a visualização das semelhanças e das diferenças existentes entre eles. Caso a visualização dos índices seja textual, os dados devem ser mostrados de forma organizada na saída padrão para permitir a distinção dos campos e registros, sendo que deve-se permitir a visualização dos registros, um por vez. O grupo também pode optar por fazer uma visualização gráfica dos índices. Por exemplo, a visualização gráfica pode ser feita por meio de tabelas, uma tabela para cada índice.

[7] Permita a visualização de estatísticas sobre as listas de registros removidos. Essa funcionalidade deve exibir uma tabela que mostra, para cada arquivo de dados, a quantidade de elementos presentes em sua lista de registros removidos. Adicionalmente, a funcionalidade também deve permitir a visualização gráfica de cada lista de registros removidos de cada arquivo de dados, contendo informações sobre o *byte offset* e o tamanho do registro removido, bem como o *byte offset* do próximo registro. Por exemplo, uma visualização gráfica interessante é:



Restrições

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

[1] Os arquivos de dados e os arquivos de índice 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. Essa escolha deve ser explicada na **documentação externa**.

[3] Devem ser exibidos avisos ou mensagens de erro sempre que apropriado.

[4] Os dados devem ser escritos e lidos campo a campo. Pode-se usar também a serialização (memcpy).

[5] Os arquivos de índice devem ser mantidos em disco, mas devem ser carregados em memória primária e serem usados em memória primária durante a execução do programa. Nesse sentido, devem ser implementadas funcionalidades que garantam a reescrita correta do índice em disco.

[6] 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.

[7] Todo código fonte deve ser documentado. A **documentação interna** 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.

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

[9] 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 para Linux.

[9] O programa deve ser acompanhado de uma **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.
- SEÇÕES 1 a N: Quaisquer 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. Por exemplo, na Parte I do trabalho, deve ser feita a descrição dos campos dos registros do arquivo de dados, 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. A escolha dos tamanhos dos campos, bem como dos campos propriamente ditos, deve ser

justificada. Coloque aqui também os sistemas operacionais que foram usados e como o programa deve ser compilado e executado.

- SEÇÃO N+1: Cópias de telas da interface, por meio das quais é possível entender o funcionamento do programa.
- SEÇÃO N+2: Baterias de testes, as quais devem gerar informações que permitam acompanhar a execução do programa.
- REFERÊNCIAS BIBLIOGRÁFICAS, caso necessário.

Material para Entregar

Arquivo compactado. Deve ser preparado um arquivo .zip contendo:

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

Instruções de entrega.

[1] Enviar o arquivo compactado para o e-mail labbdciferri@gmail.com, com o seguinte assunto: [Organização de Arquivos] Trabalho Prático 2017 – Turma X – Parte Y. 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.

[2] Entregar pessoalmente a documentação externa impressa em horário e local definido na página da disciplina.

Critério de Correção

Critério de avaliação do trabalho. Na correção do trabalho, serão ponderados os seguintes aspectos.

- Qualidade da documentação (interna e externa) entregue. **MAIOR PESO**

- Corretude da execução do programa. **MAIOR PESO**
- Qualidade da interface.

Restrições adicionais sobre o critério de correção.

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

Critério de avaliação dos integrantes. Podem ser incluídas uma ou mais perguntas a respeito do trabalho na prova.

Data de Entrega do Trabalho

Na data especificada na página da disciplina.

Bom Trabalho !