

Universidade Federal do Amazonas  
Instituto de Computação  
Bancos de Dados I – 2019/02  
Professor: Altigran Soares da Silva ([alti@icomp.ufam.edu.br](mailto:alti@icomp.ufam.edu.br))  
Monitor: Ronny Guimarães ([ronny@icomp.ufam.edu.br](mailto:ronny@icomp.ufam.edu.br))  
**Trabalho Prático 2 – 23 de setembro de 2019**  
**Data da Entrega: 23 de outubro de 2019**  
**Data da Entrega com Adiamento: 25 de outubro de 2019**

## 1. Apresentação

Este trabalho consiste na implementação de programas para armazenamento e consulta em dados armazenados em memória secundária utilizando as estruturas de arquivo e índice estudadas em sala de aula. Os programas devem fornecer suporte para a inserção, assim como diferentes meios de busca, seguindo as técnicas apresentadas nas aulas de organização e indexação de arquivos.

O trabalho deve ser implementado na linguagem C++ utilizando as bibliotecas padrão de chamadas de sistema disponíveis no Linux.

Os dados para testes e avaliação dos programas implementados serão fornecidos em um arquivo CSV formatado que servirá como entrada de dados.

## 2. Descrição do trabalho

O arquivo de dados deverá armazenar registros de dados sobre artigos científicos publicados em conferências. A estrutura deste arquivo será a seguinte:

Campo	Tipo	Descrição
ID	inteiro	Código identificador do artigo
Título	alfa 300	Título de artigo
Ano	inteiro	Ano de publicação do artigo
Autores	alfa 150	Lista dos autores do artigo
Citações	inteiro	Número de vezes que o artigo foi citado
Atualização	data e hora	Data e hora da última atualização dos dados
Snippet	alfa 1024	Resumo textual dos dados do artigo

Os seguintes programas devem ser implementados:

- upload <file>*: Programa que fará a carga inicial dos dados de teste que irá criar um banco de dados composto pelos seguintes arquivos:
  - Arquivo de dados organizado por hashing
  - Arquivo de índice primário usando uma B+ Tree
  - Arquivo de índice secundário usando um outra B+ Tree
- findrec <ID>*: Busca no arquivo de dados por um registro com o ID informado, se existir, e retornar os campos do registro, a quantidade de blocos lidos para encontrá-lo e a quantidade total de bloco do arquivo de dados;
- seek1 <ID>*: Devolve o registro com ID igual ao informado, se existir, pesquisando através do arquivo de índice primário, mostrando todos os campos, a quantidade de blocos lidos para encontrá-lo no arquivo de índice e a quantidade total de blocos do arquivo de índice primário;
- seek2 <Titulo>*: Mostra os dados do registro que possua o Título igual ao informado, se existir, pesquisando através do arquivo de índice secundário, informando a quantidade de blocos lidos para encontrá-lo no arquivo de índice e a quantidade total de bloco do arquivo de índice secundário;

### 3. Registro das Equipes

Para registrar a equipe, os membros deverão enviar um e-mail para o monitor **Ronny Guimarães** ([ronny@icomp.ufam.edu.br](mailto:ronny@icomp.ufam.edu.br)) com os **nomes** e **e-mails** dos membros da equipe até as 23/09/2019 do dia 30/09/2019.

**Atenção:** (1) só serão aceitos e-mails enviados do domínio `icomp.ufam.edu.br`; (2) os e-mails de todos os integrantes devem ser do domínio `icomp.ufam.edu.br`; (3) os e-mails deve ser enviados no máximo até o dia **30/09/2019 às 23:59**; (4) As equipes devem ter 2 ou 3 integrantes; (5) **Não serão aceitas mudanças nos integrantes da equipe depois do envio do e-mail.**

### 4. O que entregar

Um link para uma Imagem Docker disponibilizada no Docker Hub (<https://hub.docker.com/>). No container gerado a partir dessa imagem, devem ser disponibilizados os seguintes arquivos no diretório `/tp2`:

- a) Os arquivos-fonte dos programas comentados.
- b) Os programas prontos para serem testados. Os nomes dos programas devem seguir as instruções indicadas no item 2 deste trabalho.
- c) A documentação do projeto dos programas deve ser disponibilizada em um único arquivo PDF nomeado **TP2.pdf**, registrando as todas as decisões de projeto tomadas, incluindo:
  - a. A estrutura de cada arquivo de dados e índices;
  - b. Quais fontes formam cada programa;
  - c. As funções que cada fonte contém;
  - d. Quem desenvolveu cada fonte/função;
  - e. Qual o papel de cada função;

### 4. Como entregar

O link da imagem no Docker Hub deve ser enviado para o e-mail [ronny@icomp.ufam.edu.br](mailto:ronny@icomp.ufam.edu.br) até às **23:59 do dia 23/10/2018** (no caso de adiamento, até às **23:59 do dia 25/10/2018**). No dia 26/10/2018, para fins de verificação, será postada no classroom uma lista das equipes que enviaram link.

### 5. Avaliação

Execução: Execução correta: 30%; Saída legível: 15%; Estilo de programação: Código bem estruturado: 15%; Documentação: Código legível: 15%, Descrição das estruturas de dados e principais decisões: 25%

### 6. Comentários Gerais

Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar. Seja ético, desenvolva seu trabalho, não copie de outras equipes e nem da Internet. O monitor e o professor poderão pedir defesa do trabalho apresentado.

### 7. Sobre o Docker

Caso não esteja familiarizado com a tecnologia de containers e com o Docker, seguem alguns links para ajudar:

- <https://www.redhat.com/pt-br/topics/containers/what-is-docker>
- <https://blog.geekhunter.com.br/docker-na-pratica-como-construir-uma-aplicacao/>
- <https://woliveiras.com.br/posts/comandos-mais-utilizados-no-docker/>
- <http://flaviosilveira.com/2017/criando-seu-container-com-dockerfile/>
- <https://medium.com/trainingcenter/docker-dockerhub-pull-e-push-nas-suas-imagens-57dffa0232ad>
- <https://docs.docker.com/>