

# Protótipo de um sistema gerenciador de arquivos

Caroline de Oliveira Braga

<sup>1</sup>Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ)  
Unidade Descentralizada de Petrópolis – Petrópolis – RJ – Brasil

braga.coliveira@gmail.com

**Resumo.** *Este relatório visa explicar e mostrar como utilizar o protótipo de um sistema gerenciador de arquivos desenvolvido em linguagem C para a disciplina de algoritmos e estruturas de dados I.*

## 1. Introdução

O sistema gerenciador de arquivos desenvolvido trabalha com arquivos de texto em formato “.txt”. Ao armazenar esses arquivos, o conteúdo de cada um é dividido em blocos de duzentos bytes até atingir o uso total da memória disponível, que consiste em cem blocos, totalizando 20.000 bytes.

O sistema pode ser inicializado junto com a inserção de arquivos, via linha de comando do terminal e, após a inicialização, ele exibirá um menu contendo as operações disponíveis. O gerenciador suporta as seguintes operações: inserção de arquivos, remoção de arquivos, busca de um nome de arquivo no sistema e contagem da ocorrência de um termo nos arquivos do sistema.

## 2. Metodologia

Todo o programa foi desenvolvido em linguagem C, com seu código fonte subdividido em três arquivos: arquivos.h, que é a biblioteca, main.c, que contém a função main e, arquivos.c, onde estão implementadas as funções do gerenciador de arquivos e o menu.

A biblioteca arquivos.h contém as declarações das structs utilizadas, de algumas variáveis e das seguintes funções, presentes no arquivos.c: cadastrar, descadastrar, busca, inserir, remover, contador\_ocorrencias, ocorrencias\_tratadas, ocorrencias\_arquivo e menu.

O arquivo main.c recebe, por meio da função main, os nomes de arquivos digitados na linha de comando do terminal e os insere no sistema, além de cadastrar suas informações numa lista encadeada. Se a memória já estiver ocupada, o programa exibe uma mensagem de erro. A última atribuição da função main é apresentar o menu de operações ao usuário.

O menu pergunta ao usuário, por meio do terminal, qual operação ele deseja executar ou se deseja sair do programa. Sendo 1 para realizar inserção, 2 para remoção, 3 para buscar um arquivo no sistema, 4 para buscar um termo nos arquivos do sistema e 5 para sair do programa.

A função cadastrar armazena o texto do arquivo no gerenciador, além de cadastrar suas informações no sistema. A função descadastrar remove as informações de um determinado arquivo do sistema. A função busca procura o nome de um arquivo e as informações a ele associadas no sistema, além de imprimir no terminal as informações de

todos os arquivos armazenados e a lista encadeada associada ao arquivo escolhido pelo usuário. A função inserir é responsável por inserir diretamente o texto num bloco de memória do gerenciador. A função remover remove o texto de um arquivo dos blocos de memória do sistema.

A função contador\_ocorrencias conta as ocorrências de um termo em um bloco da memória. A função ocorrencias\_tratadas verifica se o termo procurado pelo usuário está presente na interseção de dois blocos de um mesmo arquivo. A função ocorrencias\_arquivo reúne as informações obtidas pelas funções contador\_ocorrencias e ocorrencias\_tratadas para descobrir o total de ocorrências do termo em um arquivo.

### 3. Exemplos de uso

O programa pode ser utilizado por qualquer pessoa que tenha acesso a um computador, pois não exige conhecimentos técnicos nem acesso à internet. A sua utilização está dividida em inserção, remoção, busca no sistema e contador de ocorrências de um termo.

#### 3.1. Inserção

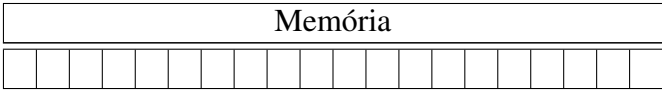
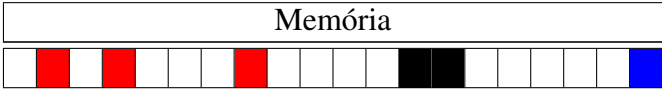


Table 1. Inserção de um arquivo.





**Table 4. Remoção de um arquivo.**

### 3.3. Busca de arquivo

Procura o nome de um arquivo e as informações a ele associadas no sistema, ou seja, na lista encadeada de informações. Também imprime no terminal as informações de todos os arquivos armazenados e a lista encadeada associada ao arquivo escolhido pelo usuário. Considerando que os arquivos "test.txt" e "t.txt" estão cadastrados no sistema e o usuário buscou o arquivo "t.txt", ele receberia o seguinte retorno na tela:

nome: test.txt	início: 5	blocos: 10
nome: t.txt	início: 2	blocos: 2

**Table 5. Exemplo**

O arquivo t.txt está nas seguintes posições da memória:

2 3

Caso o arquivo não esteja cadastrado no sistema, o programa apresentará uma mensagem informando que o arquivo não foi encontrado e imprimirá as informações dos demais arquivos do sistema. Imaginando que os arquivos do exemplo anterior continuam no sistema, porém o usuário buscou o arquivo "abc.txt", ele obteria o seguinte resultado:

nome: test.txt	início: 5	blocos: 10
nome: t.txt	início: 2	blocos: 2

**Table 6. Exemplo**

Arquivo não encontrado.

### 3.4. Busca de termo

O programa busca e contabiliza, para cada arquivo do sistema, as ocorrências de um termo inserido pelo usuário e imprime na tela. Imaginando que o gerenciador está armazenando os arquivos "test.txt" e "t.txt", o usuário ao realizar uma busca obteria algo como:

arquivo = test.txt	ocorrências = 5
arquivo = t.txt	ocorrências = 2

**Table 7. Exemplo**

## 4. Conclusão

O protótipo de um sistema gerenciador de arquivos consegue realizar as tarefas propostas, é de fácil utilização e consiste numa aplicação prática de conteúdos estudados na disciplina algoritmos e estruturas de dados I, como os conceitos de modularização e otimização de recursos, além de estruturas de dados como listas encadeadas.

A opção por alocação dinâmica para armazenar os dados de cada arquivo inserido no sistema permite melhor uso dos recursos do computador, segundo [Jayme Luiz Szwarcfiter 2010], pois assim não existem espaços de memória pré-alocados que acabam sem utilização pelo programa. Além disso, a escolha por cem blocos de memória foi pensada para garantir uma usabilidade do programa, pois se poucos blocos fossem reservados, rapidamente o gerenciador chegaria a sua capacidade máxima, impedindo novas inserções.

## **References**

Jayme Luiz Szwarcfiter, L. M. (2010). *Estruturas de dados e seus algoritmos*. LTC, 3th edition.