



---

TRABALHO DE SISTEMAS OPERACIONAIS

## Projeto B

### Simulador de sistemas de arquivos

Enzo Dornelles Italiano<sup>1</sup>

Henrique Souza Marcuzzo<sup>2</sup>

Matheus Henrique Batistela<sup>3</sup>

CAMPO MOURÃO

DEZEMBRO/2019

---

<sup>1</sup> enzoitaliano@alunos.utfpr.edu.br

<sup>2</sup> henriquemarcuzzo@alunos.utfpr.edu.br

<sup>3</sup> matheusbatistela@alunos.utfpr.edu.br



---

## SUMÁRIO

<b>INTRODUÇÃO E DESCRIÇÃO</b>	<b>2</b>
<b>MÉTODOS</b>	<b>2</b>
<b>RESULTADOS E DISCUSSÃO</b>	<b>2</b>
<b>CONCLUSÃO</b>	<b>4</b>
<b>REFERÊNCIAS</b>	<b>4</b>



## **1. INTRODUÇÃO E DESCRIÇÃO**

Este relatório tem como objetivo, escrever a documentação do projeto, incluindo explicação da lógica de códigos se necessários. Além disso será relatado todas as dificuldades encontradas durante o processo de desenvolvimento do projeto.

O projeto tinha o propósito de simular um sistema de arquivos em um disco virtual, melhorando o aproveitamento dos alunos. Para isso, seria necessário utilizar todos os conhecimentos adquiridos durante as aulas para que fosse possível cumprir todos os requisitos do projeto, sendo eles: criar um disco virtual e funções de um Shell para que fosse possível manipular o disco criado tanto internamente, como também fazendo uma ponte para funcionar com o disco real na cópia de arquivos.

## **2. MÉTODOS**

Por meio de pesquisas em livros de Sistemas Operacionais, e às explicações do conceito de um sistema de arquivos real, obtivemos a idéia da estrutura básica do nosso disco, onde basicamente iremos utilizar ponteiros dentro da FAT para nos indicar onde começava cada item que contém dentro do diretório atual.

Para não perdermos o caminho do diretório atual, também foi necessário criar um array de ponteiros que nos indicava os índices que já foram percorridos para chegar ao atual diretório.

Portanto a execução do projeto consistiu basicamente de índices que iriam nos auxiliar para a leitura e escrita nos blocos.

## **3. RESULTADOS E DISCUSSÃO**

### **3.1. REALIZAÇÃO**

As tarefas para realização do projeto foram bem definidas no começo da implementação, onde cada um iria implementar uma parte do problema o que tornou tudo mais organizado, fazendo com que o grupo conseguisse entregar, parcialmente, o que foi proposto no prazo previsto.



### 3.2. IMPLEMENTAÇÃO

Embora o código que se encontra disponível no GitHub pelo link [https://github.com/hmarcuzzo/project\\_B\\_SO](https://github.com/hmarcuzzo/project_B_SO), vale reforçar alguns pontos de como foi feita a implementação.

As variáveis que controlam o sistema de arquivo podem ser encontradas no início do código, podendo alterar configurações do disco se assim desejar, deixando para as funções apenas a manipulação destas variáveis.

Logo a primeira variável, que é na verdade uma constante, “DISK\_NAME”, define o nome que será dado para o nosso disco virtual.

O “DISK\_SIZE”, encontrado abaixo, define o tamanho total do disco virtual que poderá ser usado. E abaixo, o “BLOCK\_SIZE” define o tamanho de cada bloco que será usado para fazermos a leitura e escrita no disco.

O “MAX\_POINTERS”, resultado de uma mudança de estratégia, define quantos ponteiros para itens (diretórios ou arquivos) pode conter em um único arquivo. Utilizado na struct “Block” que define a estrutura de uma FAT com todos os campos necessários, tanto para diretórios, tanto para arquivos. A estrutura SuperBlock, funciona de forma similar, onde define a estrutura do superbloco do nosso disco.

O array “actualDir” é necessário para guardar o caminho até o diretório atual em que se encontra o Shell, começando sempre com o root “/”, e não podendo esse ser excluído deste array.

E por fim os enum’s definem os possíveis estados de um bloco da FAT, e também os retornos de funções como “true” ou “false”.

As outras funções basicamente consiste em manipular estas variáveis globais de forma com que o objetivo de cada comando seja atingido e posteriormente persistir esses blocos da FAT no disco para serem carregados na inicialização do nosso disco.

### 3.3. DIFICULDADES

A primeira grande dificuldade e também a única, foi fazer com que a função “fread” conseguisse ler os dados dos blocos da mesma forma que foram escritos, pois ao realizarmos às leituras, dados não compatíveis estavam sendo lidos.

Com esta dificuldade em nossos caminhos, fomos forçados a mudar de estratégia, adicionando os ponteiros de itens na própria FAT, para posteriormente persistir os blocos da FAT no disco, e na inicialização ser necessário apenas o carregamento da FAT em memória,



pois os dados dos arquivos e diretórios já estariam certos e não seria necessário o carregamento dos mesmos.

Porém esta dificuldade não conseguiu ser resolvida a tempo para a entrega do projeto e portanto nos impossibilitou de efetuarmos o trabalho na sua completude, pois uma vez que não foi possível carregar a FAT em memória após ter saído do disco, não importava se os dados estavam sendo persistidos, era necessário a reinicialização do disco para que conseguimos ter a FAT em memória e ao menos permitir a simulação de como seria com os dados persistidos.

#### **4. CONCLUSÃO**

Ao final todos puderam observar o cumprimento dos objetivos, visto que eles visavam a integração dos membros e a utilização dos conceitos estudados em sala de aula para melhorar o aproveitamento. Foi possível obter uma fixação melhor do conteúdo e até um aprimoramento do mesmo.

Enfim o objetivo do projeto também foi concluído parcialmente com a simulação de como funciona um sistema de arquivos, em um disco virtual, e atendendo em partes aos objetivos propostos pelo regulamento do projeto.

#### **5. REFERÊNCIAS**

1. FREAD. Disponível em: <http://www.cplusplus.com/reference/cstdio/fread/>. Acessado em Dezembro, 2019.
2. FWRITE. Disponível em: <http://www.cplusplus.com/reference/cstdio/fwrite/>. Acessado em Dezembro, 2019.
3. fread() function in C++. Disponível em: <https://www.geeksforgeeks.org/fread-function-in-c/>. Acessado em Dezembro, 2019.
4. MAZIEIRO, Carlos A.. Sistemas Operacionais: Conceitos e Mecanismos. Curitiba, 2019.
5. TANENBAUM, Andrew S.. Sistemas Operacionais Modernos. 4.ed. São Paulo: Pearson Education do Brasil, 2016.