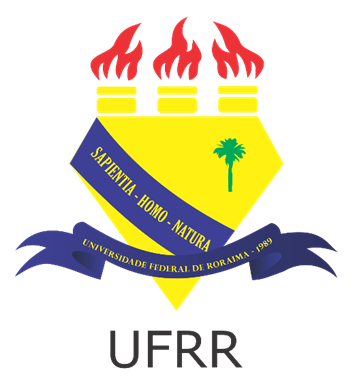
****

**UNIVERSIDADE FEDERAL DE RORAIMA**

**CIÊNCIA DA COMPUTAÇÃO - GRADUAÇÃO**

**RELATÓRIO DO PROJETO:  
Criação de um Sistema de Arquivos Virtual de 8 bits (Toy File System)**

**Boa vista – Roraima**

**2025**

**MARCUS VINÍCIUS MAIA DOS SANTOS**

**FILIPE GABRIEL TOMAZ BRITO**

**RELATÓRIO DO PROJETO:**

**Criação de um Sistema de Arquivos Virtual de 8 bits (Toy File System)**

Relatório técnico apresentado ao Prof. Dr. Herbert Oliveira Rocha, como requisito de obtenção de nota parcial na disciplina DCC403 - SISTEMAS OPERACIONAIS I.

**Boa vista – Roraima**

**2025**

**RESUMO**

Este relatório descreve o processo de estudo, desenvolvimento, implementação e testes de um sistema de arquivos virtual denominado Toy File System (ToyFS). O objetivo principal é construir um sistema de arquivos básico, utilizando a biblioteca FUSE (Filesystem in Userspace), que opere inteiramente em memória, com suporte a operações essenciais como leitura, escrita, criação e remoção de arquivos e diretórios. O projeto também incluiu a análise de sistemas de arquivos de exemplo como o Gogislenefs e o Hellofs, servindo como base para a compreensão e estruturação do sistema desenvolvido. O ToyFS foi montado e testado tanto em ambiente local quanto simulado em um pendrive virtualizado.

1. **introdução**

Sistemas de arquivos são componentes fundamentais de qualquer sistema operacional, responsáveis por organizar, armazenar, acessar e manipular dados em dispositivos de armazenamento. Este trabalho teve como proposta o desenvolvimento de um sistema de arquivos simplificado com funcionalidades básicas, focando no aprendizado prático do uso da biblioteca FUSE e da estruturação interna de sistemas de arquivos. Para isso, inicialmente foram estudados exemplos didáticos como o Gogislenefs e o Hellofs. Em seguida, foi projetado e implementado o ToyFS, com o diferencial de operar em memória, possuir hierarquia de diretórios e permitir montagem via imagem de disco simulada em um pendrive. Ao final, foram realizados testes funcionais das operações implementadas.

1. **Análise de Sistemas de Arquivos Existentes**

**2.1. Hellofs**

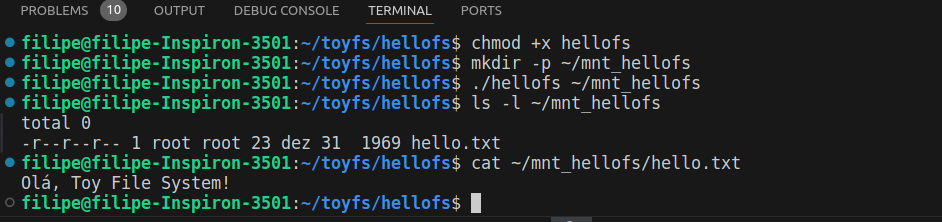
O sistema Hellofs é um exemplo didático de sistema de arquivos implementado com FUSE. Ele simula um diretório virtual que contém um único arquivo chamado "hello", cujo conteúdo é uma string fixa ("Olá, Toy File System!\n").

Estrutura utilizada:

* FUSE Operations: getattr, readdir, open, read
* Armazenamento em memória (constante estática)

Funcionalidades:

* Listagem de diretório raiz
* Leitura do arquivo "hello"



**2.2. Gogislenefs**

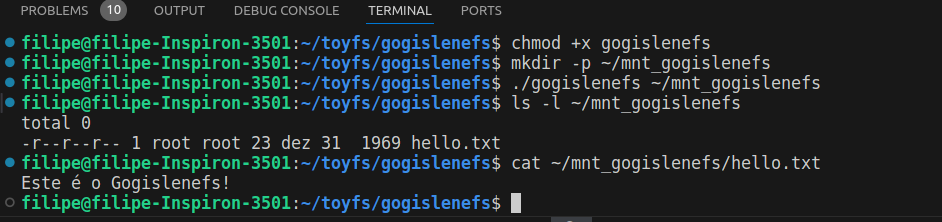
O Gogislenefs é um exemplo um pouco mais completo, que implementa a leitura e escrita de arquivos diretamente em memória. Ele armazena conteúdo em buffers e simula um pequeno sistema de arquivos volátil.

Estrutura utilizada:

* Lista de arquivos em memória
* Ponteiros para buffers de conteúdo
* Implementação de funções de escrita e leitura com FUSE

Funcionalidades:

* Criação e leitura de arquivos
* Escrita de conteúdo temporário



1. **Desenvolvimento do Toy File System**

**3.1. Requisitos**

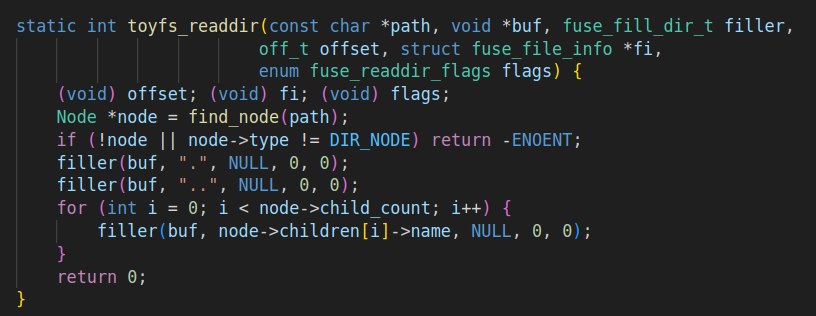
O ToyFS foi projetado com os seguintes requisitos mínimos:

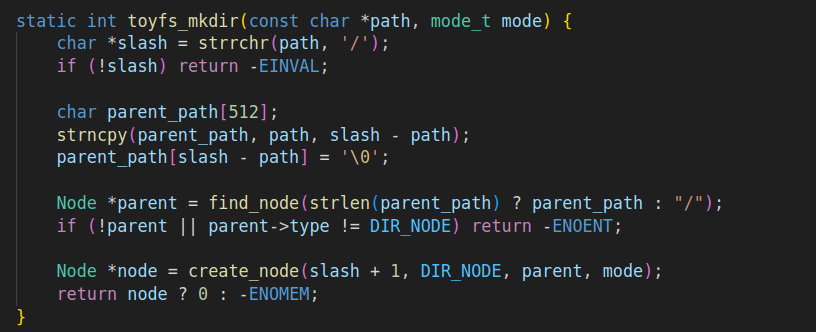
* Operações básicas: leitura, escrita, criação e remoção de arquivos e diretórios.
* Estrutura de diretórios hierárquica.
* Armazenamento temporário em memória.
* Utilização da biblioteca FUSE em C.
* Montagem a partir de uma imagem simulada em disco.

**3.2. Estrutura Interna**

O sistema de arquivos ToyFS foi estruturado com as seguintes abstrações:

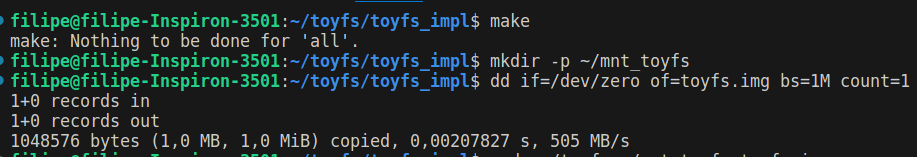
* Estrutura toyfs\_file\_entry para armazenar arquivos e diretórios
* Campos: nome, tipo (arquivo/diretório), ponteiro para conteúdo (caso arquivo), filhos (caso diretório), timestamps e permissões (parcial)
* Armazenamento como árvore em memória
* Indexação simples com listas ligadas



****

**3.3. Ferramenta de Formatação**

Foi implementado o script toyfs\_format.c que inicializa a estrutura raiz do sistema de arquivos, salvando os dados iniciais no arquivo de imagem.



1. **Conclusão**

O desenvolvimento do Toy File System permitiu explorar na prática os conceitos fundamentais de sistemas de arquivos, como estrutura hierárquica, gerenciamento de arquivos em memória e interface com o FUSE. A partir da análise dos sistemas de exemplo, foi possível aplicar os conceitos em um projeto funcional, que suporta operações básicas e montagem simulada via imagem de disco. O sistema atendeu com sucesso aos requisitos propostos, demonstrando corretamente as funcionalidades implementadas. Trabalhos futuros podem incluir a implementação de persistência real em disco, sistema de permissões completo e estrutura de blocos e inodes.

1. **Repositório**

<https://github.com/marcusv0/FinalProject_OS_UFRR_Desc_3_2025>

1. **Referências**

FUSE: Filesystem in Userspace – <https://github.com/libfuse/libfuse>

Projeto Hellofs – <https://github.com/libfuse/libfuse/tree/master/example>

Manuais do Linux: fuse(4), mount(8), dd(1), mkdir(1), cat(1)