

# ENIGMA - Linux File System

Miller Raycell, Acadêmico de Ciência da Computação, UFRR e Rodrigo Andrade, Acadêmico de Ciência da Computação, UFRR

**Resumo**—Sistema de arquivos é uma parte crucial de qualquer sistema operacional, ele é responsável pela manipulação dos arquivos, sem o mesmo, a organização e gerenciamento dos arquivos não seria possível, o seguinte relatório apresentará a implementação de um sistema de arquivo simples para Linux.

**Abstract**—File system is a crucial part of any Operation System, the file system is in charge of manipulate the archives, without it, files organization would not be capable, this article shows how to implement a simple file system for Linux.

## I. INTRODUÇÃO

SISTEMAS de arquivos são implementados em larga escala, cada sistema operacional possui um sistema de arquivos que satisfaz a sua necessidade, Windows utiliza o NTFS, Linux em geral usa da família EXT, o supracitado ENIGMA é um sistema de arquivos para Linux.

Boa Vista, Roraima  
27 de Julho de 2019

## II. O QUE É UM SISTEMA DE ARQUIVOS

Primeiramente deve-se fazer a definição de arquivos, segundo [1] um arquivo é uma unidade lógica criada por processos, os arquivos são necessários quando se necessita realizar a persistência de dados.

A manipulação dos arquivos é responsabilidade específica do sistema operacional, e é tão importante que ganhou um nome específico, que é conhecido como sistema de arquivos.

Um sistema de arquivos é projetado para armazenar dados de forma não volátil, possui características obrigatórias como um namespace (uma metodologia de organização), metadados (fornece a base lógica), API (fornece acesso a chamadas de sistemas que manipulam arquivos e diretórios), segurança (garante que usuários acessem somente seus arquivos) e um software (sistema de arquivos virtual do linux e os drivers de dispositivo específicos) para implementar o sistema de arquivos.

### A. Esquema dos sistemas de arquivos

Discos podem ser particionados e armazenar diversos sistemas de arquivos em cada uma de suas partições. Ao iniciar o computador é ativado o MBR que contém no seu fim uma tabela de partição que informa o início e fim de cada partição, a bios lê e executa o MBR que por sua vez localiza a partição ativa e lê o bloco de inicialização e inicia o SO. Cada partição além do bloco de inicialização possui um superbloco que contém parâmetros chaves do sistema de arquivos que incluem um número mágico para determinar qual o tipo do sistema de arquivos, também encontramos informações de blocos disponíveis por meio de mapa de bits ou lista de

ponteiros, seguido de i-nodes um arranjo estruturado para cada arquivo com informações sobre o mesmo, depois temos o diretório raiz e por fim os demais diretórios e arquivos.

### B. I-nodes

I-nodes associam arquivos a uma estrutura index-node que lista os atributos e endereços dos blocos no disco. O uso do i-node é vantajoso pois precisa estar na memória apenas quando o arquivo correspondente estiver aberto, sendo assim se cada i-node ocupa  $n$  bytes e  $k$  arquivos estiverem abertos, então o espaço total de memória é  $kn$  bytes. A utilização de i-nodes requer um arranjo na memória cujo tamanho é proporcional ao número máximo de arquivos que podem estar abertos ao mesmo tempo.

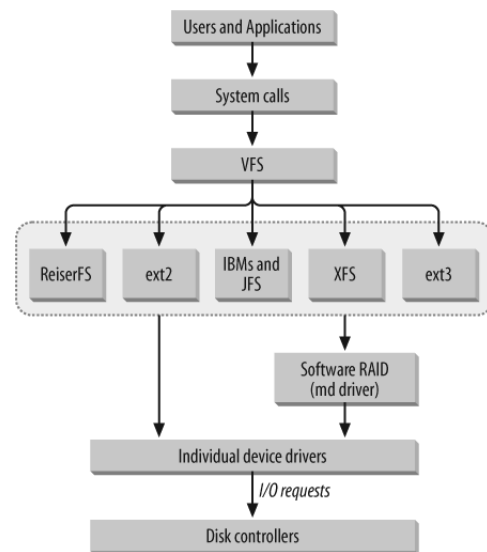


Figura 1. Representação de um sistema de arquivos

### C. Diretórios

Ao abrir um arquivo para que possa ser realizada a leitura o SO utiliza o caminho informado pelo usuário para encontrar a entrada de diretório de disco, a partir dessa informação é possível localizar os blocos de disco por meio do número do i-node. De toda maneira a principal função do sistema de diretórios é mapear o nome do arquivo em ASCII na informação necessária para que os dados sejam encontrados.

## III. ENIGMA

O ENIGMA file system, é uma implementação simples de um sistema de arquivos com funcionalidades básicas para Linux, o mesmo foi desenvolvido usando a linguagem C e

utilizando as bibliotecas próprias do Linux para sistemas de arquivos.

Para realizar a incorporação do ENIGMA se recomenda kernel versão 4, gcc versão 7.4.0 e make versão 4.1, os testes do sistema foram realizados no Ubuntu 18.04 com versão do kernel 4.18.

#### IV. IMPLEMENTAÇÃO DO ENIGMA

As operações desenvolvidas para o ENGIMA foram, criação de diretórios, criação de um usuário root, criação, leitura e escrita de arquivos. A função de criação de usuário implementadas foram baseadas na estrutura I-NODE, ou seja, o usuário root é um modelo de um I-NODE, pois através dessa representação para se fazer a manipulação das estrutura de usuários.

A função que realiza a criação de diretórios também usa a estrutura I-NODE, na própria função são feitas as verificações de estabilidade da criação de pastas, as verificações analisam se os espaços de endereçamento estão corretos, se há espaço suficiente para realizar a dita criação no diretório especificado, essas análises são importantes para se ter certeza da correteza das operações, manipular arquivos de forma errada pode desencadear inúmeros problemas no sistema operacional.

A função de escrita no arquivo realiza a escrita de uma variável atômica, para evitar problemas de sincronização, no arquivo, para que dessa forma se verifique a quantidade de vezes ocorreu a operação de escrita no arquivo.

A leitura de arquivos é uma operação simples, cada linha é armazenada em um vetor e se após os testes que realizam a verificação se o arquivo é válido, se a informação é correta, se o arquivo pode ser lido tiverem todos dados resultados positivos, um contador que identifica a quantidade de linhas é incrementado.

Quando se cria um sistema de arquivos, há a necessidade de se incorporar um superbloco de memória, no qual terá as indicações do que o sistema opera, quais os usuários, entre outros. No ENIGMA foi criado apenas um usuário que é o root, logo ele possui todas as permissões dentro do sistema de arquivos, e através do superbloco o kernel terá as informações que necessita para executar as operações desejadas.

Devido o ENIGMA ser uma simplificação de um sistema de arquivos real, como o EXT, suas funcionalidades são reduzidas a implementação própria no código, que foi apresentado, dentro do próprio código se realiza a criação de um diretório exemplo e dois arquivos contendo a quantidade de vezes que ambos foram acessados.

Assim que o módulo do ENIGMA for retirado do kernel, todas as informações criadas no mesmo serão escoadas junto com o sistema de arquivos. O ENIGMA não possui suporte a Journaling, que seria a criação de um arquivo log, com todas as operações que foram realizadas dentro do sistema de arquivos, a vantagem do Journaling é a certeza das operações realizadas.

#### V. CONCLUSION

Através deste trabalho podemos reafirmar a importância da utilização do sistema de arquivos no sistema operacional, pois somente por meio deste é possível localizar e manipular dados.

O enigma mostrou-se como um sistema de arquivos funcional e simples que cumpre seu papel, permitindo que os dados sejam acessados e manipulados.

#### ACKNOWLEDGMENT

Os autores gostariam de agradecer primeiramente ao Prof. Dr. Herbert Rocha pela oportunidade de realizar esse trabalho e pela paciência ao nos acompanhar durante a execução do mesmo. Também gostaríamos de agradecer aos nossos colegas de laboratório André Leandro e Markus Kaul, pelo apoio e suporte enquanto necessitávamos de tempo para realizar o trabalho.

#### REFERÊNCIAS

- [1] Tanenbaum, Andrew S and Bos, Herbert *Modern operating systems*, Pearson, 2015.
- [2] Tweedie, Stephen C and others *Journaling the Linux ext2fs filesystem*, Durham, North Carolina, 1998