

Simulador de Sistema de Arquivos

Por Mateus Ferrareze Malgarin

Palavras-chave: Arquivos. Diretórios. Journaling. Simulação. Operações.

Resumo

Neste projeto, apresentamos a implementação de um sistema de arquivos simulado, que inclui suporte a operações de criação, renomeação, cópia e deleção de arquivos e diretórios. Utilizamos uma estrutura de dados eficiente para gerenciar o sistema de arquivos e implementamos um sistema de journaling para garantir a consistência e integridade dos dados. Os resultados demonstram a eficácia do sistema em realizar operações básicas e a importância do journaling na manutenção da consistência do sistema de arquivos.

Introdução

Um Sistema de Arquivos é uma estrutura que um sistema operacional usa para controlar como os dados são armazenados e recuperados em um dispositivo de armazenamento, como um disco rígido, SSD ou dispositivo removível. Ele organiza os dados em uma hierarquia de arquivos e diretórios, permitindo que os usuários e os aplicativos acessem e gerenciem esses dados de maneira eficiente. O propósito principal do Journaling em sistemas de arquivos é aumentar a confiabilidade e a integridade dos dados. Ele ajuda a garantir que, em caso de falha do sistema, como quedas de energia ou travamentos, o sistema de arquivos possa ser restaurado a um estado consistente rapidamente, minimizando a perda de dados e o tempo de recuperação. Existem diferentes tipos de journaling, cada um com seu próprio método de registros e alterações, como o write-ahead logging, neste método todas as operações de modificação são registradas no journal antes de serem aplicadas ao sistema de arquivos principal ; o log-structured, sistemas de arquivos estruturados em log organizam todos os dados e metadados como um log contínuo ; ordered journaling, este método garante que todas as alterações de dados são registradas no journal antes das alterações nas estruturas de metadados do sistema de arquivos.

Metodologia

O simulador foi desenvolvido em Java, em modo shell, que interage com o usuário por meio de uma interface de linha de comando, recebendo comandos do usuário, processando-os e exibindo os resultados diretamente no terminal ou console. Foram utilizadas as seguintes estruturas de dados para o projeto: Map(Hash-Map), utilizado para armazenar os arquivos; Set(HashSet), utilizado para armazenar os diretórios e List(ArrayList), utilizado na classe journal para armazenar uma lista de entradas de operações realizadas no sistema de arquivos. Partindo para as classes, temos a

'FileSystemSimulator', que é o núcleo do sistema de arquivos, responsável por gerenciar arquivos e diretórios. Suas principais funções incluem criação de diretórios e arquivos com seus conteúdos, listando-os de um diretório principal criado, renomeação de diretórios, subdiretórios e arquivos, assim como seus caminhos, cópia de arquivos para novos caminhos, remoção de diretórios e arquivos, e todos os seus conteúdos inseridos e salvar e carregar o estado do sistema, utilizando o journal para reconstruir o estado, registrando todas essas operações. A classe 'File' e a classe 'Directory', responsáveis pela manipulação dos arquivos e diretórios, dando nome, conteúdo, lista de arquivos e subdiretórios, acesso e modificação, assim como adicionar e remover arquivos ou diretórios. A classe 'Journal', que é crucial para o funcionamento seguro e consistente do sistema de arquivos, registrando operações e permitindo recuperação dos dados em caso de falhas, suas principais funções são: adicionar entradas, retornar a lista destas, limpar entradas dadas, salvar e carregar o journal de/para um arquivo respectivamente.

Resultados e Discussão

Ao executar a classe 'Main', o sistema de arquivos simulado realiza diversas operações sobre diretórios e arquivos, produzindo a saída, mostrando o diretório principal criado e seu subdiretório com seus arquivos, conteúdos e caminhos específicos, mostra a cópia, renomeação, listagem, remoção dos arquivos e diretórios, apresentando toda a manipulação e simulação do sistema de arquivos, de forma estrutural passo a passo e eficiente.

Conclusão

Os testes e resultados obtidos indicam que o FileSystemSimulator é uma ferramenta funcional e eficiente para simular operações básicas de um sistema de arquivos, demonstrando claramente a importância do journaling para manter a integridade e a consistência do sistema.

Referências

TECNOLÓGICO, C. **UNIVERSIDADE FEDERAL DE SANTA CATARINA**. Disponível em: <https://lisha.ufsc.br/teaching/os/ine5355-2008-1/work/g1/journaling_filesystems_text.pdf>. Acesso em: 15 jun. 2024.

O que é : File System Journaling. Disponível em: <<https://artigos.valuehost.com.br/glossario/o-que-e-file-system-journaling/>>. Acesso em: 15 jun. 2024.

Anexos

<https://github.com/MateusFerrareze/FileSystemSimulator.git> Link do repositório do projeto.

