

DESENVOLVIMENTO DE UM SIMULADOR DE SISTEMA DE ARQUIVOS EM JAVA COM SUPORTE A JOURNALING

Autor 1: Pedro Henrique Câmara Matos

github: <https://github.com/phcmprogrammer/FileSystemSimulator.git>

Palavras-chave: Simulador de Sistema de Arquivos, Java, Journaling, Operações de Arquivos, Integridade de Dados

Resumo

Este trabalho apresenta o desenvolvimento de um simulador de sistema de arquivos em Java com suporte a journaling, implementando funcionalidades básicas de manipulação de arquivos e diretórios. O simulador permite a criação, renomeação, listagem e exclusão de arquivos e diretórios, bem como a cópia de arquivos entre diretórios. Através do uso de journaling, o simulador garante a integridade dos dados, registrando todas as operações realizadas. Este relatório descreve a metodologia utilizada, os resultados obtidos e as conclusões alcançadas, proporcionando uma compreensão prática do funcionamento de um sistema de arquivos.

Introdução

Sistemas de arquivos são componentes cruciais de sistemas operacionais, responsáveis por organizar e armazenar dados de forma eficiente e segura. Este projeto visa desenvolver um simulador de sistema de arquivos em Java, com suporte a journaling para garantir a integridade dos dados. Journaling é uma técnica que registra mudanças antes de aplicá-las, permitindo a recuperação do sistema em caso de falhas. O objetivo é fornecer uma ferramenta educativa que permita a visualização e compreensão das operações básicas de um sistema de arquivos e a importância do journaling.

Metodologia

Para o desenvolvimento do simulador, utilizou-se a linguagem de programação Java devido à sua robustez e ampla adoção. O simulador foi projetado para executar operações básicas de arquivos e diretórios: criação, exclusão, renomeação, listagem e cópia de arquivos. Cada operação é registrada em um arquivo de journal, permitindo a recuperação em caso de falhas. A metodologia seguiu os seguintes passos:

1. Definição dos Requisitos: Identificação das operações essenciais para um sistema de arquivos básico.
2. Projeto do Sistema: Estruturação das classes principais (FileSystemSimulator, Directory, File, Journal) e seus métodos.
3. Implementação: Desenvolvimento do código Java, com atenção especial ao manejo do journaling.
4. Teste e Validação: Execução de comandos no simulador para validar o funcionamento correto das operações.
5. Documentação: Criação deste relatório para documentar o processo e os resultados.

Resultados e Discussão

O simulador desenvolvido suporta todas as operações especificadas, incluindo a criação, exclusão, renomeação, listagem e cópia de arquivos e diretórios. O comando help foi adicionado para auxiliar o usuário na utilização do simulador. A seguir, são apresentados exemplos de comandos e suas saídas:

- **Criação de Diretórios:**

```
> mkdir /dir1  
> mkdir /dir1/subdir1
```

- **Listagem de Diretórios:**

```
> ls /  
  
Contents of directory /:  
- /dir1  
  
> ls /dir1  
  
Contents of directory /dir1:  
- /dir1/subdir1
```

- **Criação de Arquivos:**

```
> createFile /dir1 file1.txt  
> createFile /dir1/subdir1 file2.txt
```

- **Renomear Diretório:**

```
> renameDir /dir1/subdir1 /dir1/subdir2
```

- **Renomear Arquivo:**

```
> renameFile /dir1/subdir2 file2.txt file2_renamed.txt
```

- **Copiar Arquivo:**

```
> copyFile /dir1 /dir1/subdir2 file1.txt
```

- **Apagar Arquivo:**

```
> deleteFile /dir1 file1.txt
```

- **Apagar Diretório:**

```
> rmdir /dir1/subdir2
```

Os resultados demonstram a correta funcionalidade do simulador, permitindo a visualização prática das operações de um sistema de arquivos e a importância do journaling para a integridade dos dados.

- **Funcionamento do Journaling**

O journaling é uma técnica fundamental para garantir a integridade dos dados em sistemas de arquivos. No simulador desenvolvido, todas as operações realizadas são registradas em um arquivo de journal, permitindo a recuperação do estado do sistema em caso de falhas.

- **Exemplo de Conteúdo do Journal:**

Após executar os comandos acima, o arquivo journal.txt terá registros como:

```
CREATE_DIRECTORY /dir1
```

```
CREATE_DIRECTORY /dir1/subdir1
```

```
CREATE_FILE /dir1 file1.txt
```

```
CREATE_FILE /dir1/subdir1 file2.txt
```

```
RENAME_DIRECTORY /dir1/subdir1 /dir1/subdir2
```

```
RENAME_FILE /dir1/subdir2 file2.txt file2_renamed.txt
```

```
COPY_FILE /dir1 /dir1/subdir2 file1.txt
```

```
DELETE_FILE /dir1 file1.txt
```

```
DELETE_DIRECTORY /dir1/subdir2
```

- **Recuperação do Sistema:**

Em caso de falha do sistema, o arquivo de journal pode ser utilizado para restaurar o estado do sistema de arquivos. A partir do journal, é possível reexecutar as operações na ordem em que foram registradas para reconstruir o sistema de arquivos até o último estado consistente.

Por exemplo, se o sistema falhar após a execução de alguns comandos, o conteúdo do arquivo de journal pode ser processado novamente para garantir que todas as operações até o ponto de falha sejam reaplicadas, restaurando o sistema de arquivos ao seu estado correto.

Conclusão

O desenvolvimento deste simulador de sistema de arquivos em Java com suporte a journaling atingiu os objetivos propostos, proporcionando uma ferramenta educacional eficaz para entender as operações básicas de um sistema de arquivos e a importância do journaling. O simulador permite a execução de operações de criação, renomeação, listagem, exclusão e cópia de arquivos e diretórios, registrando todas as operações em um arquivo de journal. Futuras melhorias podem incluir a adição de funcionalidades avançadas, como controle de permissões e suporte a diferentes tipos de arquivos.

Referências

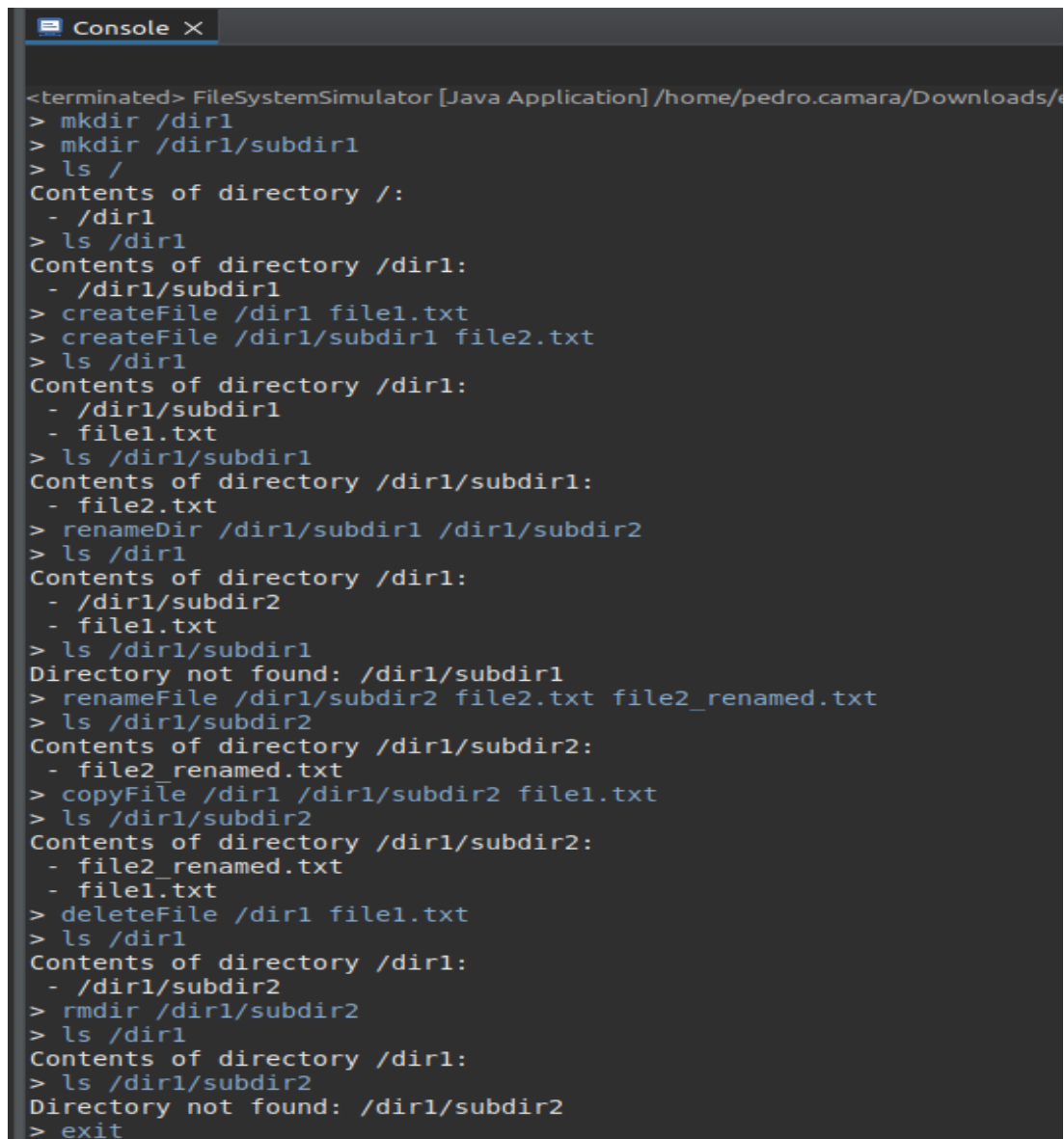
Tanenbaum, A. S., & Bos, H. (2014). Modern Operating Systems. Pearson.
Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts. Wiley.

Anexos

Listagem de comandos:

```
> help
Available commands:
mkdir [path]          - Create a directory
rmdir [path]          - Remove a directory
renameDir [oldPath] [newPath] - Rename a directory
ls [path]              - List contents of a directory (defaults to root if no path is specified)
createFile [dirPath] [fileName] - Create a file
deleteFile [dirPath] [fileName] - Delete a file
renameFile [dirPath] [oldName] [newName] - Rename a file
copyFile [sourceDirPath] [destDirPath] [fileName] - Copy a file
exit                  - Close the journal and exit the program
help                  - Show this help message
```

Execução/simulação de comandos:



```
<terminated> FileSystemSimulator [Java Application] /home/pedro.camara/Downloads/FileSystemSimulator.jar
> mkdir /dir1
> mkdir /dir1/subdir1
> ls /
Contents of directory /:
- /dir1
> ls /dir1
Contents of directory /dir1:
- /dir1/subdir1
> createFile /dir1 file1.txt
> createFile /dir1/subdir1 file2.txt
> ls /dir1
Contents of directory /dir1:
- /dir1/subdir1
- file1.txt
> ls /dir1/subdir1
Contents of directory /dir1/subdir1:
- file2.txt
> renameDir /dir1/subdir1 /dir1/subdir2
> ls /dir1
Contents of directory /dir1:
- /dir1/subdir2
- file1.txt
> ls /dir1/subdir1
Directory not found: /dir1/subdir1
> renameFile /dir1/subdir2 file2.txt file2_renamed.txt
> ls /dir1/subdir2
Contents of directory /dir1/subdir2:
- file2_renamed.txt
> copyFile /dir1 /dir1/subdir2 file1.txt
> ls /dir1/subdir2
Contents of directory /dir1/subdir2:
- file2_renamed.txt
- file1.txt
> deleteFile /dir1 file1.txt
> ls /dir1
Contents of directory /dir1:
- /dir1/subdir2
> rmdir /dir1/subdir2
> ls /dir1
Contents of directory /dir1:
> ls /dir1/subdir2
Directory not found: /dir1/subdir2
> exit
```