

Sistema *fakerational* com filesystem em B+ tree

1 Descrição Geral

Neste exercício, o você deverá implementar o seu próprio sistema *fakerational*. Esse sistema simula comandos básicos de um terminal Unix/Linux, utilizando como estrutura de dados principal uma **B+ tree** implementada do zero.

Além da funcionalidade do sistema, será exigida uma **avaliação empírica de desempenho** das operações da árvore B+, com comparação aos limites assintóticos teóricos.

2 Comandos Suportados

O sistema deve interpretar e executar os seguintes comandos:

- **ls**
Lista os arquivos e diretórios presentes no diretório atual, em ordem lexicográfica.
- **cd <nome>**
Acessa o diretório <nome>, se ele existir.
- **cd ..**
Retorna ao diretório pai. Caso esteja na raiz, permanece nela.
- **mkdir <nome>**
Cria um novo diretório com o nome especificado.
- **touch <nome>**
Cria um novo arquivo com o nome especificado.
- **rm <nome>**
Remove um arquivo ou diretório existente (não é necessário implementar remoção recursiva).

3 Requisitos

3.1 Implementação da Árvore B+

- A árvore B+ deve ser implementada do zero, sem uso de bibliotecas externas.
- Deve possuir grau mínimo configurável e manter as propriedades de balanceamento.
- Os nós internos armazenam chaves e ponteiros para filhos.
- Os nós folha armazenam os dados (arquivos ou diretórios) e devem estar duplamente encadeados.

3.2 Modelagem de Arquivos e Diretórios

- Arquivos são elementos terminais e não contêm filhos.
- Diretórios contêm uma nova instância de árvore B+, que armazena seus filhos.
- O diretório raiz deve ser criado automaticamente ao iniciar o sistema.

3.3 Simulação de Terminal

- O sistema deve simular um terminal interativo com um **prompt** que reflita o caminho atual.
- Comandos devem ser lidos do usuário e processados sequencialmente.
- Erros como acessar diretórios inexistentes ou remover elementos ausentes devem ser tratados com mensagens adequadas.

4 Avaliação Experimental

Você deverá implementar um conjunto de testes automatizados para mensurar os tempos médios das seguintes operações da árvore B+:

- Inserção
- Busca
- Remoção

Essas operações devem ser testadas com diferentes tamanhos de entrada, como $n = 10^4, 10^5, 10^6$, e os tempos devem ser comparados às complexidades assintóticas teóricas esperadas para árvores B+ de grau d :

- Inserção: $\mathcal{O}(\log_d n)$
- Busca: $\mathcal{O}(\log_d n)$
- Remoção: $\mathcal{O}(\log_d n)$

Gráficos e Relatório

Inclua na entrega um gráfico ou relatório comparando os resultados experimentais com as curvas teóricas de crescimento logarítmico. O uso de bibliotecas como `matplotlib`, `seaborn` ou similares é permitido.

5 Exemplo de Execução

```
fakerational:~$ mkdir projetos
fakerational:~$ cd projetos
fakerational:~/projetos$ touch experimento.txt
fakerational:~/projetos$ mkdir resultados
fakerational:~/projetos$ ls
experimento.txt
resultados/
fakerational:~/projetos$ cd ..
fakerational:~$ rm projetos
fakerational:~$ ls
(nada)
```

6 Entrega

O projeto deverá conter:

- Código-fonte modular, separado por:
 - Implementação da árvore B+
 - Interpretador de comandos
 - Scripts de benchmark
- Arquivo `README.md` com instruções de uso.
- Gráficos e relatório contendo a análise empírica de desempenho.
- Código comentado e organizado.
- Gráfico do conjunto de árvores geradas (visualização do sistema de arquivos)

7 Critérios de Avaliação

- Correção e completude da B+ Tree.
- Funcionalidade dos comandos e manutenção da hierarquia.
- Robustez do sistema frente a entradas inválidas.
- Qualidade e coerência da análise empírica.
- Clareza, modularidade e legibilidade do código.