

MC202EF - Estruturas de Dados

Lab 05

(O conteúdo necessário para realizar este laboratório vai até a unidade:
Árvore de busca AVL)

PED: Márcio de Carvalho Saraiva
PAD: Anderson, Mateus, Victor
Professor: Lehilton Lelis Chaves Pedrosa

02 de outubro de 2016

1. Problema

Zezinho é um rapaz pouco organizado e mantém todos os seus arquivos dentro de um único diretório no seu computador. Por isso, é importante filtrar e manipular rapidamente todos os arquivos na pasta. Por meio de um interpretador de comandos simplificados, Zezinho realiza operações sobre arquivos. Você deverá implementar cada um dos comandos sobre uma estrutura de arquivos eficiente.

2. Entrada

Cada linha contém um comando, que pode ser:

touch <FILENAME>

Cria um arquivo de nome <FILENAME>. O nome não contém espaços e tem no máximo 20 caracteres.

ATENÇÃO: No sistema de arquivos implementado por Zezinho, pode haver mais de um arquivo com o mesmo nome!

rm <EXPRESSION>

Remove todos arquivos cujos nomes correspondem a <EXPRESSION>.

ls <EXPRESSION>

Lista, um por linha, em ordem alfabética (de acordo com a tabela ASCII), todos os arquivos existentes atualmente no diretório que correspondam a <EXPRESSION>.

Nos comandos acima, <EXPRESSION> tem um dos seguintes formatos:

1. **PREFIX***: corresponde a qualquer arquivo cujo nome começa com PREFIX, por exemplo, **ab*** faz corresponde a todos os arquivos que começam com as letras ab, como: abismo, abacate, ab. Sempre que não houver arquivo correspondente, deve-se imprimir uma linha com **nao existem arquivos com essa expressao**
ATENÇÃO: PREFIX pode ser vazio, isto é, apenas * .

2. **FILENAME**: o nome do arquivo é FILENAME; se não existir arquivo com esse nome, deve imprimir uma linha com **nao existem arquivos com essa expressao**.

Exemplo de entrada:

```
touch casa
touch carro
touch livro
touch lixo
touch luxo
ls livro
touch luxo
rm ca*
rm livro
ls li*
ls carro
ls luxo
```

3. Saída

Um conjunto de linhas com a saída dos comandos ls (ou mensagens **nao existem arquivos com essa expressao**, quando os arquivos que serão listados não existirem).

A saída para o exemplo será:

```
livro
lixo
nao existem arquivos com essa expressao
luxo
luxo
```

4. Dicas

- Lembrem-se de liberar o espaço após o uso.
- Implemente uma função para cada um dos comandos separadamente.

5. Avaliação

5.1 As notas desse laboratório serão calculadas da seguinte maneira:

- 7 pontos proporcionais proporcionais à quantidade de acertos dos casos de teste do sistema.
- 3 pontos referentes à qualidade de código: legibilidade, algoritmo, memory leak, boas práticas de programação (comentários no código, escolha do nomes para variáveis, reutilização de funções que possam melhorar a apresentação do código, etc...)

5.2 Critérios avaliados neste laboratório:

Além de passar nos casos de teste, é obrigatório:

- Implementar o conjunto dinâmico baseado em Árvore AVL.
- Criar funções que verificam cada uma das condições para rotacionar (rotação à esquerda e rotação à direita) os arquivos.

6. Entrega

A submissão de código deve ser feita no Run.codes em no máximo 10 tentativas até o dia 14/10/16 às 23:59:59.