

Trabalho Prático II

Seminários I

Valor: 25 pontos

Entrega: quarta-feira dia 27/Abril (via SGA)

Questão 1: (5 pontos)

Utilizando o comando **cat**, mostre como é possível concatenar o conteúdo de múltiplos arquivos textuais. Por exemplo, concatenar os arquivos `introducao.txt`, `desenvolvimento.txt` e `conclusao.txt` em um único arquivo chamado `texto.txt`.

Questão 2: (5 pontos)

Suponha que durante a execução de um programa, um arquivo `log.txt` foi gerado, contendo todas as mensagens de avisos (WARNING) e erros (ERROR), além de outras informações sobre a execução.

Exemplo de `log.txt`:

INFO: Lendo configurações

WARNING: Arquivo de configurações não encontrado. Utilizando configurações padrões.

INFO: Sistema configurado

ERROR: Exceção encontrada durante execução. O Sistema será encerrado.

Responda:

- a) Utilizando o comando **grep**, mostre como filtrar apenas as mensagens de aviso (WARNING).
- b) Utilizando o comando **grep**, mostre como filtrar apenas as mensagens de erro (ERROR).

Questão 3: (5 pontos)

Pipelines oferecem um conceito muito poderoso e versátil em ambientes Linux, permitindo encadear programas para compor um programa maior e mais complexo. Em um encadeamento de comandos por pipeline, a saída de um comando serve como entrada para o comando seguinte. O script `wordfreq.sh`, apresentado a seguir, realiza a contagem da frequência das palavras contidas na entrada.

wordfreq.sh

```
tr -s '[:punct:][:space:]' '\n' | tr '[:upper:]' '[:lower:]' | sort | uniq -c | sort -bnr
```

O script `wordfreq.sh` é composto por cinco comandos encadeados em um pipeline. O primeiro comando quebra todo o conteúdo textual da entrada em uma lista de palavras separadas por quebra de linha. O segundo comando converte todas as palavras da lista para minúscula. O terceiro comando ordena a lista de palavras. O quarto comando realiza a contagem da frequência de cada palavra da lista. Por fim, o último comando ordena as palavras por sua frequência.

Com esse script complexo, é possível fazer a contagem da frequência das palavras apenas utilizando o comando

```
cat <file> | sh wordfreq.sh
```

ou então

```
echo <text> | sh wordfreq.sh
```

Por exemplo, o comando a seguir
echo 'hello world, hello again' | sh wordfreq.sh
produz
2 hello
1 world
1 again

Responda:

Mostre como estender os comandos dos dois exemplos anteriores, utilizando um **pipeline** para o comando **grep**, filtrando a contagem de uma única palavra. Por exemplo, filtrar apenas a contagem da palavra 'hello' no exemplo anterior.

Questão 4: (5 pontos)

Implementar um script para realizar backup de pastas. O script deve receber o caminho de uma pasta como parâmetro de linha de comando, por exemplo:

```
sh ./backup.sh pasta/minhaPasta
```

O script deve criar uma pasta local chamada backup, verificar se o diretório passado por parâmetro existe e caso existir, copiar a pasta e todo seu conteúdo para a pasta backup. Se o caminho não existir, uma mensagem deve ser exibida.

Observações: no comando if, **-f <arq>** verifica se <arq> é um arquivo existente, enquanto **-d <dir>** verifica se <dir> é um diretório existente.

Questão 5: (5 pontos)

Implementar um script de uma calculadora. O script calc.sh deve receber um primeiro parâmetro que informa qual operação (“+”, “-”, “p”, “d”) e uma sequência de números. O operador deve ser verificado utilizando comandos de if, para em seguida chamar realizar a operação adequada.

Exemplo:

```
sh ./calc.sh + 10 2 3
```

Retornando o valor “15” em tela.

```
sh ./calc.sh d 10 2
```

Retornando o valor “5” em tela.