

Atividade 06

Valor: 3,0 pontos na Primeira Nota

Primeira Chamada: 22/10/2025

Segunda Chamada: 29/10/2025

Questão 01 - 0,5 ponto - Deve ser feita em sala.

Nesta questão, o nome do *script* será *cinco_diretorios.sh*. Ao ser executado, o *script* deve realizar as seguintes tarefas:

1. Criar um diretório chamado *cinco*.
2. Criar cinco subdiretórios *cinco/dir1* até *cinco/dir5*.
3. Em cada subdiretório, faça quatro arquivos, *arq1.txt* até *arq4.txt*:
 1. O arquivo *arq1.txt* deve ter uma linha contendo apenas o dígito 1.
 2. O arquivo *arq2.txt* deve ter duas linhas, cada uma contendo o dígito 2.
 3. O arquivo *arq3.txt* deve ter três linhas, cada uma contendo o dígito 3.
 4. O arquivo *arq4.txt* deve ter quatro linhas, cada uma contendo o dígito 4.

Você não pode repetir chamadas ao *mkdir* 6 vezes e executar 20 comandos um atrás do outro para criar os 20 arquivos. Você deve obrigatoriamente utilizar laços aninhados para criar a estrutura.

Questão 02 - 1,0 ponto - Deve ser feita em sala.

Faça um *script* chamado *ordenar_linhas.sh* que receba como parâmetro o caminho de um diretório que só tem arquivos de texto como conteúdo. O *script* deve imprimir em ordem crescente uma lista com os arquivos de acordo com a quantidade de linhas de cada arquivo. Lembrando que o arquivo com mais linhas não é necessariamente o arquivo com tamanho maior em *bytes*.

Com base no enunciado, a função do *script* é ordenar arquivos de texto por sua contagem de linhas. Para criar um ambiente de teste eficaz, você precisa de um diretório que contenha múltiplos arquivos de texto com diferentes quantidades de linhas.

Aqui está a descrição de como criar esse ambiente de teste usando comandos básicos do Linux:

Correção

O ambiente será um diretório chamado `testes_linhas` contendo três arquivos (`a.txt`, `b.txt`, `c.txt`) com diferentes contagens de linhas.

1. **Crie o Diretório de Teste:** Crie um diretório chamado `testes_linhas` onde você irá colocar os arquivos de texto:

Bash

```
mkdir testes_linhas
```

2. **Crie o Arquivo 1 (Menos Linhas):** Crie um arquivo com **3 linhas**.

Bash

```
echo -e "Linha 1\nLinha 2\nLinha 3" > testes_linhas/a.txt
```

3. **Crie o Arquivo 2 (Número Intermediário de Linhas):** Crie um arquivo com **5 linhas**.

Bash

```
echo -e "Linha A\nLinha B\nLinha C\nLinha D\nLinha E" > testes_linhas/b.txt
```

4. **Crie o Arquivo 3 (Mais Linhas):** Crie um arquivo com **7 linhas**.

Bash

```
seq 1 7 > testes_linhas/c.txt
```

(Alternativamente, você pode usar `echo -e "...\\n..."` como nos passos anteriores).

5. **Verifique a Estrutura:** Confirme se os arquivos foram criados e a contagem de linhas está correta:

Bash

```
wc -l testes_linhas/*
```

A saída deve mostrar: 3, 5 e 7 linhas para os respectivos arquivos.

Agora, o diretório `testes_linhas` está pronto para ser passado como parâmetro para o seu script. O resultado esperado do script deve ser a lista de arquivos ordenada por linha: `a.txt`, `b.txt`, `c.txt`.

Questão 03 - 1,5 pontos - Apresentada no dia 29/10/2025

Você vai desenvolver o *script* `latencia.sh`. Esse *script* vai receber como parâmetro o nome de um arquivo de texto, contendo um endereço IP por linha.

O *script* deve usar o comando *ping* para enviar dez pacotes ICMP para cada endereço do arquivo, calculando o valor médio do tempo de resposta. Ao final, deve imprimir uma lista de IP ordenada do menor para o maior tempo médio de resposta, informando além do endereço, o tempo de resposta médio. **Observação:** não pode usar *awk*.

```
# os IPs abaixo são fantasia.
$ cat enderecos_ip.txt
8.8.8.8
192.168.1.1
208.67.222.222
104.26.4.156
$ ./latencia.sh enderecos_ip.txt
192.168.0.1 11.1ms
208.67.222.222 55.4ms
8.8.8.8 94.0ms
104.26.4.156 23.7ms
```