Arquivos, Pastas e Permissões

Esta lista foca em **chmod**, **chown**, **cp**, **mv**, **rm** e **nano**.

**1. Qual comando chmod (numérico) torna o arquivo script.sh executável pelo dono, legível pelo grupo e nada para outros (r-x r-- ---)?**

a) chmod 755 script.sh

b) chmod 740 script.sh

c) chmod 540 script.sh

d) chmod 140 script.sh

**2. O que significa a permissão chmod 644 relatorio.txt?**

a) Dono pode ler/escrever; Grupo pode ler/escrever; Outros podem ler/escrever.

b) Dono pode ler/escrever; Grupo pode ler; Outros podem ler.

c) Dono pode ler/executar; Grupo pode ler; Outros podem ler.

d) Dono pode ler/escrever; Grupo e Outros não podem fazer nada.

**3. Você quer remover a permissão de escrita (w) do *grupo* no arquivo config.ini. Qual comando simbólico faz isso?**

a) chmod g-w config.ini

b) chmod o-w config.ini

c) chmod u-w config.ini

d) chmod a-w config.ini

**4. Qual comando muda o *usuário dono* para "ana" e o *grupo dono* para "devs" no arquivo main.py?**

a) chmod ana:devs main.py

b) usermod -g devs ana

c) chown ana:devs main.py

d) chown main.py ana:devs

**5. Como você muda APENAS o *grupo dono* do arquivo logo.png para "marketing"?**

a) chown :marketing logo.png

b) chgrp marketing logo.png

c) chown marketing logo.png

d) Ambas 'a' e 'b' estão corretas.

**6. Qual é a diferença entre cp arquivo.txt /tmp/ e mv arquivo.txt /tmp/?**

a) cp copia o arquivo para /tmp/; mv move o arquivo para /tmp/ (ele deixa de existir na origem).

b) cp move o arquivo; mv copia o arquivo.

c) cp renomeia o arquivo; mv move o arquivo.

d) Não há diferença, ambos movem o arquivo.

**7. O que o comando mv relatorio\_v1.doc relatorio\_final.doc (executado no mesmo diretório) faz?**

a) Cria uma cópia chamada relatorio\_final.doc.

b) Move relatorio\_v1.doc para dentro do arquivo relatorio\_final.doc.

c) Renomeia relatorio\_v1.doc para relatorio\_final.doc.

d) Dá um erro, pois mv só move entre diretórios.

**8. Você tem um diretório "antigo" cheio de arquivos. Qual comando o apagará junto com todo o seu conteúdo?**

a) rmdir antigo

b) rm antigo

c) rm -r antigo

d) del antigo

**9. Você quer aplicar permissões (750) a um diretório "Projeto" e a *todos* os arquivos e subdiretórios dentro dele. Qual flag você usa?**

a) -a (all)

b) -f (force)

c) -R (recursive)

d) -c (changes)

**10. Qual comando abre o arquivo lista.txt em um editor de texto simples no terminal?**

a) cat lista.txt

b) nano lista.txt

c) vi lista.txt

d) Ambas 'b' e 'c' estão corretas (mas nano foi o ensinado).

**Cenário: Organizando o Servidor da Agência**

**(Setup: Crie um usuário bruno e um grupo designers antes de começar os passos abaixo.)**

**1. Crie um diretório chamado Portfolio:** \_\_\_\_ \_\_\_\_

**2. Entre no diretório Portfolio:** \_\_\_\_ \_\_\_\_

**3. Crie um arquivo para o cliente "XPTO" e um script de upload:** \_\_\_\_\_ \_\_\_\_

**4. Veja as permissões padrão dos arquivos que você criou:** \_\_\_\_

**5. Torne o script upload.sh executável *apenas para você* (o dono):** \_\_\_ \_\_\_ \_\_\_

**6. Mude o arquivo xpto.jpg para que o dono e o grupo possam ler/escrever, mas outros não possam fazer nada:** \_\_\_ \_\_\_ \_\_\_

**7. Mude o *dono* do arquivo xpto.jpg para o usuário bruno:** \_\_\_ \_\_\_ \_\_\_

**8. Mude o *grupo dono* de ambos os arquivos (xpto.jpg e upload.sh) para designers:** \_\_\_ \_\_\_ \_\_\_

**9. Verifique as permissões, donos e grupos para confirmar suas mudanças:** \_\_

**10. Crie um diretório chamado Uploads:** \_\_\_ \_\_\_

**11. Mova o upload.sh para dentro do diretório Uploads:** \_\_\_ \_\_\_ \_\_\_

**12. Crie uma cópia do xpto.jpg chamada xpto\_backup.jpg:** \_\_\_ \_\_\_ \_\_\_

**13. Renomeie o arquivo xpto.jpg original para xpto\_final.jpg:** \_\_\_ \_\_\_ \_\_\_

**14. Volte para o diretório anterior (seu /home):** Comando: \_\_\_ \_\_\_

**15. Apague o diretório Portfolio e todo o seu conteúdo (pastas, backups, tudo):** \_\_\_ \_\_\_ \_\_\_