

# 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?**

- a) chmod 755 script.sh
- b) chmod 740 script.sh
- c) chmod 540 script.sh
- d) chmod 140 script.sh - Essa

**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. - Essa
- 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 - Essa
- 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` - Essa

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. - Essa

**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). - Essa

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`. - Essa

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` - Essa

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) - Essa
- 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 - Essa
- 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:** mkdir Portfolio

**2. Entre no diretório Portfolio:** cd Portfolio

**3. Crie um arquivo para o cliente "XPTO" e um script de upload:** touch xpto.txt | touch upload.sh

**4. Veja as permissões padrão dos arquivos que você criou:** ls -l

**5. Torne o script upload.sh executável apenas para você (o dono):** chmod u+x upload.sh

**6. Mude o arquivo xpto.jpg para que o dono e o grupo possam ler/escrever, mas outros não possam fazer nada:** chmod 660 xpto.jpg

**7. Mude o dono do arquivo xpto.jpg para o usuário bruno:** sudo chown bruno xpto.jpg

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

**9. Verifique as permissões, donos e grupos para confirmar suas mudanças:** `ls -l`

**10. Crie um diretório chamado Uploads:** `mkdir Uploads`

**11. Mova o upload.sh para dentro do diretório Uploads:** `mv upload.sh Uploads`

**12. Crie uma cópia do xpto.jpg chamada xpto\_backup.jpg:** `sudo cp xpto.jpg xpto_backup.jpg`

**13. Renomeie o arquivo xpto.jpg original para xpto\_final.jpg:** `sudo mv xpto.jpg xpto_final.jpg`

**14. Volte para o diretório anterior (seu /home):** `cd /home`

**15. Apague o diretório Portfolio e todo o seu conteúdo (pastas, backups, tudo):**  
`rm -r Portfolio | y`