

Permissões de arquivos

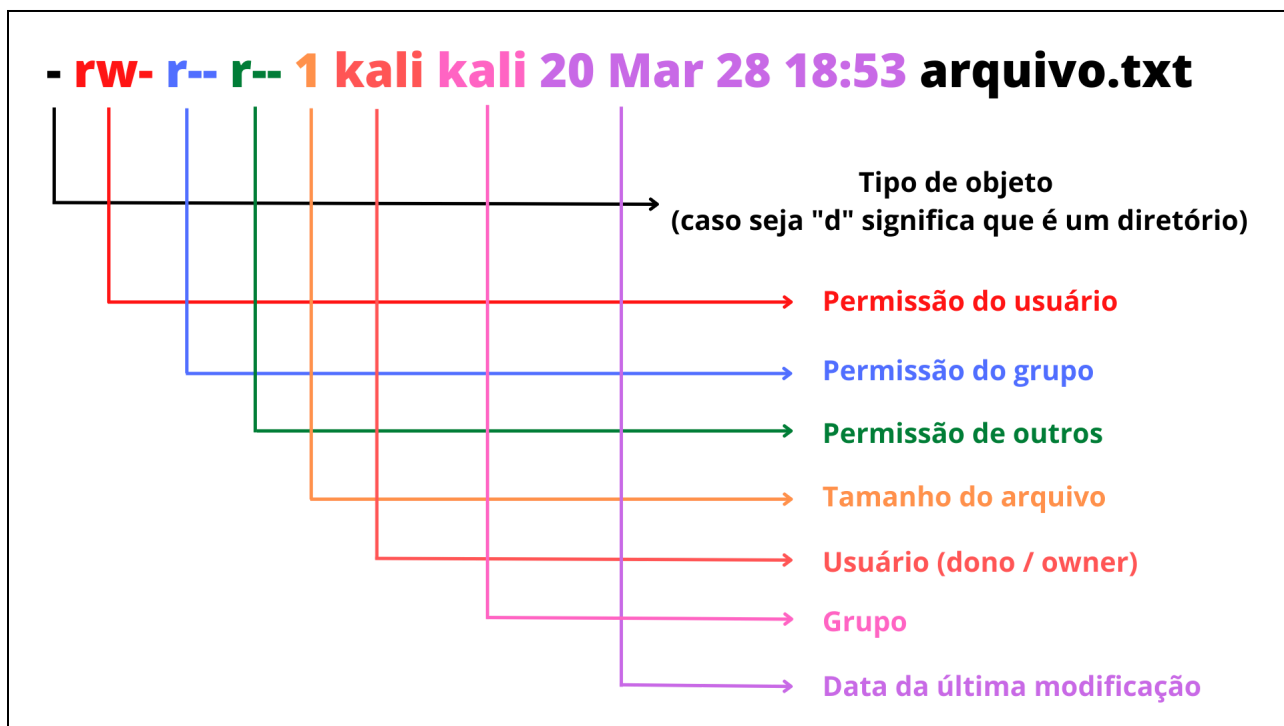
No Linux, cada arquivo ou diretório tem permissões que controlam **quem pode ler, escrever ou executar** o conteúdo.

Tipos de usuários para fins de permissão:

- **User**: o dono do arquivo.
- **Group**: grupo ao qual o arquivo pertence.
- **Others**: todos os demais usuários.

Tipos de permissões:

- **r (read)** → leitura → vale 4
- **w (write)** → escrita → vale 2
- **x (execute)** → execução → vale 1



Por: Alice Dantas

Comando *chmod* — altera permissões

Significa "**change mode**" (alterar modo).

```
chmod <tipo_de_usuario><+_ou_-><tipo_de_permissao> <arquivo>
```

Exemplos de alguns casos:

`chmod u+rw script.sh` → adiciona permissão ao usuário(u) de leitura(r), escrita(w) e execução(x) ao arquivo "script.sh".

`chmod g+rx script.sh` → adiciona permissão ao grupo(g) de leitura(r) e execução(x) ao arquivo "script.sh".

`chmod o-w script.sh` → remove permissão a outros(o) de escrita(w) ao arquivo "script.sh".

Também é possível usar notação numérica, como:

```
chmod 755 script.sh
```

- **7** = leitura (4) + escrita (2) + execução (1) → para o **usuário**
- **5** = leitura (4) + execução (1) → para o **grupo**
- **5** = leitura (4) + execução (1) → para os **outros**

Por: Alice Dantas

Comando *chown* — altera dono e grupo

Significa "**change owner**" (alterar dono).

`chown <usuario>:<grupo> <arquivo.txt>`

Exemplos de alguns casos:

`chown joao:devs arquivo.txt` → altera o **dono** do arquivo para o usuário joao e o **grupo** para devs.

`chown maria arquivo.txt` → altera **apenas o dono** do arquivo para maria (o grupo permanece o mesmo).

`chown :projeto arquivo.txt` → altera **apenas o grupo** do arquivo para projeto (o dono permanece o mesmo).