

# Permissões de arquivos (Linux)

## Descrição de projeto

O uso de comandos do Linux para configurar autorizações.

## Verifique os detalhes do arquivo e do diretório

Serão exploradas as permissões da pasta *projects* e os arquivos que ela contém. O comando `ls -la` lista detalhadamente todos os arquivos e diretórios do diretório atual, incluindo os ocultos (que começam com `.`)

## Descreva a sequência de permissões

A cadeia de 10 caracteres em permissões de arquivos no Linux representa o tipo de arquivo e os níveis de acesso para o dono, o grupo e outros usuários. O primeiro caractere indica o tipo de arquivo, podendo ser um diretório (`d`), um arquivo comum (`-`), ou um link simbólico (`l`). Os nove caracteres seguintes são divididos em três grupos de três, representando, respectivamente, as permissões do dono, do grupo e de outros usuários. Cada conjunto pode conter `r` para leitura (read), `w` para escrita (write) e `x` para execução (execute). Por exemplo, na cadeia `drwxr-x---`, o `d` indica que é um diretório, `rw` mostra que o dono pode ler, escrever e executar, `r-x` significa que o grupo pode ler e executar, mas não modificar, e `---` indica que outros usuários não têm permissão de acesso.

```
researcher2@a4e1c3713755:~$ cd projects
researcher2@a4e1c3713755:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Feb  7 19:48 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb  7 21:12 ..
-rw--w---- 1 researcher2 research_team  46 Feb  7 19:48 .project_x.tx
drwx--x--- 2 researcher2 research_team 4096 Feb  7 19:48 drafts
-rw-rw-rw- 1 researcher2 research_team  46 Feb  7 19:48 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Feb  7 19:48 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Feb  7 19:48 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Feb  7 19:48 project_t.txt
researcher2@a4e1c3713755:~/projects$
```

## Alterar permissões de arquivos

O comando `chmod o-w` e `chmod g-r` modifica as permissões dos arquivos, removendo a permissão de escrita para outros usuários (`o-w`) e a permissão de leitura para o grupo (`g-r`). Isso significa que, após a alteração, outros usuários não poderão mais modificar o arquivo `project_k.txt`, e os membros do grupo não poderão mais ler o arquivo `project_m.txt`. O dono do arquivo continua com suas permissões inalteradas. Esse tipo de ajuste é útil para restringir o acesso e proteger o conteúdo do arquivo contra edições não autorizadas por usuários externos ao dono.

```
researcher2@a4e1c3713755:~/projects$ chmod o-w project_k.txt
researcher2@a4e1c3713755:~/projects$ chmod g-r project_m.txt
researcher2@a4e1c3713755:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Feb  7 19:48 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb  7 21:12 ..
-rw--w---- 1 researcher2 research_team  46 Feb  7 19:48 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Feb  7 19:48 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Feb  7 19:48 project_k.txt
-rw----- 1 researcher2 research_team  46 Feb  7 19:48 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Feb  7 19:48 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Feb  7 19:48 project_t.txt
researcher2@a4e1c3713755:~/projects$
```

## Alterar permissões em um arquivo oculto

O comando `chmod u-w, g-w, g+r .project_x.txt` altera as permissões do arquivo oculto `.project_x.txt`, removendo a permissão de escrita do dono (`u-w`) e do grupo (`g-w`), ao mesmo tempo em que concede a permissão de leitura para o grupo (`g+r`). Após essa mudança, o dono do arquivo não poderá mais modificá-lo, assim como os membros do grupo, mas estes ainda poderão ler seu conteúdo. Outros usuários não são afetados por essa

alteração.

```
researcher2@a4e1c3713755:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@a4e1c3713755:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Feb  7 19:48 .
drwxr-xr-x 3 researcher2 research_team 4096 Feb  7 21:12 ..
-r--r----- 1 researcher2 research_team  46 Feb  7 19:48 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Feb  7 19:48 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Feb  7 19:48 project_k.txt
-rw----- 1 researcher2 research_team  46 Feb  7 19:48 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Feb  7 19:48 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Feb  7 19:48 project_t.txt
researcher2@a4e1c3713755:~/projects$
```

## Alterar permissões de diretório

O comando `chmod g-x .[*]` remove a permissão de execução para o grupo em todos os arquivos ocultos do diretório atual. No Linux, arquivos ocultos começam com um ponto (.), e a expressão `.[*]` garante que tanto arquivos simples quanto aqueles com múltiplos pontos no nome sejam incluídos. Ao remover a permissão de execução (`g-x`), membros do grupo deixam de poder executar arquivos ocultos que sejam scripts ou programas e, no caso de diretórios ocultos, perdem a capacidade de acessá-los diretamente. Isso é útil para restringir o uso de arquivos sensíveis sem afetar as permissões do dono ou de outros usuários.

```
researcher2@a4e1c3713755:~/projects/drafts$ chmod g-x .[*]
researcher2@a4e1c3713755:~/projects/drafts$ ls -la
total 8
drwx----- 2 researcher2 research_team 4096 Feb  7 19:48 .
drwxr--r-x 3 researcher2 research_team 4096 Feb  7 19:48 ..
researcher2@a4e1c3713755:~/projects/drafts$
```

## Resumo

Experiência prática no uso de comandos básicos do shell Linux Bash para:

- examinar permissões de arquivo e diretório,
- alterar permissões em arquivos visíveis e ocultos e
- alterar permissões em diretórios.