

Permissões em arquivos e diretórios no Linux

Olá, nesse artigo iremos falar a respeito das permissões básicas em arquivos e diretórios no **Linux**.

1 - O comando **<ls -l>** lista arquivos e diretórios em detalhes:

```
ls -l
```

Saída:

```
total 0
drwxr-xr-x  2 dbahero  hero  64 Nov 12 09:39 Blog
-rw-r--r--  1 dbahero  hero   0 Nov 12 09:39 teste.sh
```

Ao separar o resultado da saída acima por colunas, nós temos a seguinte estrutura:

Permissões	Número de links	Proprietário	Grupo a que pertence	Tamanho em bytes	Mês	Dia	Hora:Minuto	Nome do arquivo
drwxr-xr-x ¹⁾	2	dbahero	hero	64	Nov	12	09:39	Blog
-rw-r--r-	1	dbahero	hero	0	Nov	12	09:39	teste.sh

Podemos notar o seguinte formato para as permissões:

Formato Padrão	Permissão
r	Leitura
w	Escrita
x	Execução

Esse formato de permissão também têm uma divisão em colunas através do separador **<hífen>**:

```
-rw-r--r--
```

Abaixo a representação de cada coluna:

Coluna 1	Coluna 2	Coluna 3
User	Group	Others

A **<Coluna 1>**²⁾ representa as permissões para quem é proprietário do arquivo ou diretório, já a **<Coluna 2>**³⁾ as permissões para quem pertence aquele mesmo grupo e a **<Coluna 3>**⁴⁾ as permissões para todos os outros usuários que não são proprietários e não pertencem aquele grupo.

Nós temos também a representação em formato **<Octal>**, que é a representação numérica de cada permissão:

Formato Padrão	Formato Octal	Permissão
r	4	Leitura
w	2	Escrita
x	1	Execução

O comando **<chmod>** é o responsável por aplicar, alterar ou remover as permissões de um determinado arquivo ou diretório:

Aplicar permissão de leitura a todos os usuários em um determinado arquivo ou diretório:

```
chmod +r <arquivo_ou_diretorio>
```

Remover permissão de escrita em um determinado arquivo ou diretório:

```
chmod -w <arquivo_ou_diretorio>
```

Quando omitimos as colunas ao aplicar ou remover uma permissão, o comando **<chmod>** é aplicado a todos (**User, Group, Others**).

Abaixo temos um exemplo indicando a quem devemos aplicar ou revogar as permissões:

```
chmod u+x teste.sh
```

No exemplo acima nós passamos o argumento **<u>**⁵⁾, seguido do símbolo de adição **<+>**⁶⁾, o tipo de permissão **<x>**⁷⁾ e o nome do arquivo.

Já no exemplo abaixo nós passamos os argumentos **<go>**⁸⁾, seguido do símbolo de subtração **<->**⁹⁾, o tipo de permissão a ser revogada **<w>**¹⁰⁾ e o nome do arquivo:

```
chmod go-w teste.sh
```

O mesmo vale para utilização do formato **<Octal>**¹¹⁾:

Formato Padrão	Formato Octal	Permissão
r	4	Leitura
w	2	Escrita
x	1	Execução

No exemplo abaixo é usado o formato octal para representação das permissões e a quem deve-se aplicar cada uma:

```
chmod 654 teste.sh
```

O formato acima fornece permissões de leitura/escrita **<6>** para quem é proprietário, leitura/execução **<5>** para quem pertence aquele grupo e permissão de somente leitura **<4>** para todos os outros.

Percebam que os valores utilizados são o resultado da soma de cada de tipo de permissão:

Formato Octal	Permissão	Tradução
6	r+w	leitura/escrita
5	r+x	leitura/execução
4	r	leitura

Uma vez que se compreende o formato octal, torna-se mais simples de aplicar/revogar as permissões

em arquivos e diretórios.

Dica:

```
Todo diretório no Linux necessita de permissão de execução para que possamos acessá-lo/navegá-lo
```

E como fazemos para alterarmos um proprietário de um arquivo ou diretório?

Assim como **<chmod>** é utilizado para permissões, nós temos o **<chown>** que é utilizado para alterarmos o proprietário e grupo de um arquivo ou diretório:

Listar com detalhes o arquivo **<teste.sh>**

```
ls -l teste.sh
```

Saída:

```
-rw-r--r--. 1 dbahero hero 0 Nov 12 11:23 teste.sh
```

No exemplo acima, nós temos o arquivo **<teste.sh>** que pertence ao usuário **<dbahero>** e ao grupo **<hero>**

Para alterar o grupo podemos utilizar o **<chown>** da seguinte maneira:

```
chown dbahero:oinstall teste.sh
```

Listar novamente o arquivo **<teste.sh>** com detalhes:

```
ls -l teste.sh
```

Saída com o resultado da alteração do grupo:

```
-rw-r--r--. 1 dbahero oinstall 0 Nov 12 11:25 teste.sh
```

Para alteração do proprietário, executamos da seguinte forma:

```
chown oracle:oinstall teste.sh
```

Listar novamente o arquivo **<teste.sh>** com detalhes:

```
-rw-r--r--. 1 oracle oinstall 0 Nov 12 11:30 teste.sh
```

Entendemos então que a utilização do comando **<chown>**, têm o seguinte formato:

```
chown <usuario>:<grupo> nome_arquivo_ou_diretorio
```

- Até a próxima.

Permissões em arquivos e diretórios no Linux

— Autor: [Leonardo Lopes](#) 14/11/2023 19H:46

1)

A letra <**d**> no início, indica que é um diretório.

2)

Proprietário

3)

Grupo

4)

Outros

5)

User

6)

Adição de permissão

7)

Execução

8)

Group, Others

9)

Remoção/Revogação de permissão

10)

Escrita

11)

Numérico

From:

<https://blog.dbahero.com/> -

Permanent link:

<https://blog.dbahero.com/doku.php?id=permissaoarquivodiretorio>

Last update: **14/11/2023 19H:46**

