

Controle de usuários, grupos e permissões

O **usuário** root é aquele que tem o maior nível de privilégio no Linux. Ele consegue instalar, acessar qualquer pasta, remover qualquer arquivo, etc.

`adduser nomedousuario` → Adiciona um usuário.

Vai ser pedido o cadastramento de uma senha.

Somente o root pode acrescentar um usuário ou grupo ao sistema. Logo você tem que fazer a elevação de privilégio utilizando o comando `sudo`, para você agir como administrador.

`sudo adduser nomedousuario`

Toda vez que criamos um usuário, é criado um grupo com o mesmo nome e uma pasta pessoal para esse usuário.

`su nomedousuario` → Troca de usuário.

Vai ser pedida a senha.

`sudo su` → Troca para o usuário root.

`passwd nomedousuario` → Altera a senha do usuário.

`lastlog` → Exibe informações de login de todos os usuários do sistema.

`last` → Exibe uma listagem de entrada e saída do usuário no sistema.

`logname` → Exibe o nome do usuário atual logado no sistema.

`id` → Exibe todos os identificadores do usuário. Ele mostra o usuário e os grupos a que esse usuário pertence.

O arquivo `passwd`, dentro da pasta `etc`, exibe todos os usuários.

`cat /etc/passwd`

`userdel nomedousuario` → Remove um usuário.

`userdel -r nomedousuario` → Remove um usuário e a pasta pessoal dele.

Para esses comandos é necessário usar o sudo para elevação de privilégio.

Grupos permitem organizar os usuários e definir as permissões de acesso a arquivos e diretórios de forma mais fácil.

Esses grupos são criados pelos programas para que você tenha privilégio de acesso aos arquivos desses programas e até privilégios de acesso ao próprio sistema.

Existe um arquivo chamado group, na pasta etc, que contém todos os grupos do sistema:

`cat /etc/group`

`groups` → Exibe todos os grupos de um usuário, que ele pertence.

`addgroup` → Cria um grupo.

Somente o root pode acrescentar um grupo ao sistema.

Ex.: `sudo addgroup nomedogruppo`

`adduser usuario grupo` → Adiciona um usuário a um grupo.

Você precisa do comando sudo para fazer isso.

Ex.: `sudo adduser usuario grupo`

`gpasswd -a usuario grupo` → Também adiciona um usuário a um grupo.

Você precisa do comando sudo para fazer isso.

`gpasswd -d usuario grupo` → Remove um usuário de um grupo.

Você precisa do comando sudo para fazer isso.

`groupdel grupo` → Remove um grupo.

Você precisa do comando sudo para fazer isso.

Se aparecer a mensagem: Usuário não está no arquivo sudoers, significa que o usuário não é um usuário administrador. O sudoers é o arquivo de administradores.

Permissões em arquivos e diretórios servem para restringir acessos como: leitura, escrita e execução, onde:

r - read (leitura)

w - write (escrita)

x - execution (execução)

ls -lh → Verifica permissões em um diretório. Ele exibe detalhes sobre determinado diretório exibindo, além do nome dos arquivos que existem no diretório, outros dados como data de criação, por quem foi criado, entre outros, e também as permissões.

Ex.: drwxr-xr-x

O d identifica que se trata de um diretório. Se fosse um arquivo seria um - ;

O primeiro grupo de 3, identifica as permissões do dono;

O segundo grupo de 3 identifica as permissões do grupo;

O terceiro grupo de 3 identifica as permissões dos outros, ou seja, aqueles que nem estão no grupo.

chmod → Muda a permissão de um arquivo ou diretório.

Como opção do comando chmod, nós temos um modo que se chama octal.

(Não existe apenas o modo octal, mas esse é o modo mais fácil para se alterar permissões.)

A máscara octal é composta por números sob a base 8, ou seja, de 0 a 7, onde:

o primeiro dígito representa o dono do ficheiro/diretório(u);

o segundo dígito representa o grupo (p);

o terceiro dígito representa os outros (o).

As permissões são especificadas para cada grupo.

User (owner)			Group			Other		
r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1

Exemplo:

Supondo que eu deseje colocar somente a permissão de execução para o usuário (dono) e nenhuma permissão para o grupo e para os outros.

```
chmod 100 arquivo.txt
```

E se eu quisesse somente a permissão de escrita (gravação) para o usuário (dono) e nenhuma permissão para o grupo e para os outros?

```
chmod 200 arquivo.txt
```

E se eu quisesse somente a permissão de leitura para o usuário (dono) e nenhuma permissão para o grupo e para os outros?

```
chmod 400 arquivo.txt
```

Agora, se eu quisesse escrita (gravação) e execução, você vai somar os números. (2 + 1) Isso para o usuário (dono) e nenhuma permissão para o grupo e para os outros.

```
chmod 300 arquivo.txt
```

E se, no mesmo caso, eu quisesse a permissão de leitura também? Soma tudo. (4 + 2 + 1)

```
chmod 700 arquivo.txt
```

E o mesmo serve para se você quiser dar essas permissões para o grupo e os outros.

Se eu quiser todas as permissões pro usuário (dono), somente leitura e execução e, pros outros, somente leitura:

```
chmod 754 arquivo.txt
```

Se eu quero todas as permissões para todas as estruturas:

```
chmod 777 arquivo.txt
```