Cauã Daniel | 1ºDS

**Tabela de permissões de arquivos e pastas do Linux**

As permissões dentro de um sistema Linux protegem os arquivos e diretórios do acesso individuo ou não autorizado de algum usuário. São as permissões que impedem que algum software malicioso cause estragos no nosso sistema. Vamos entender algumas coisas sobre permissões.

O princípio da segurança do nosso sistema de arquivos Linux é definir o acesso aos arquivos por donos, grupos e usuários.

Dono, Grupo e usuário e suas definições:

|  |  |
| --- | --- |
| **Dono** | É o usuário no qual criou o arquivo ou diretório. |
| **Grupo** | Permite que os usuários do grupo tenham a cesso ao arquivo/diretório. |
| **Outros usuários** | Usuários que não são donos ou não pertencem ao grupo do arquivo/diretório. |

Tipos de permissão

Dentro do sistema Linux nós temos 3 tipos de permissões básicas:

|  |  |
| --- | --- |
| **r** | Permissão de leitura para arquivos e permissão pra listar conteúdo dos diretórios. |
| **w** | Permissão para gravação de arquivos e permite criar arquivos e diretórios caso for um diretório. |
| **x** | Permite execução de um arquivo (se for um arquivo executável) e caso for um diretório permite o acesso. |

Nós conseguimos ver as permissões utilizando o comando “**ls -l”** As 3 letras (rwx) são agrupadas da seguinte forma:

**drwxr-xr– root root teste**

Da segunda a quarta letra (rwx) dizem qual é a permissão de acesso ao dono do arquivo. Sendo assim ele tem a permissão de ler (**r – read**), gravar (**w – write**) e executar (**x – execute**) o arquivo computadores.

Da quinta a sétima letra (**r-x**) diz qual é a permissão de acesso ao grupo do arquivo. Neste caso todos os usuários que pertencem ao grupo root tem a permissão de ler (**r**), e executar (**x**) o arquivo teste. Lembre-se que quanto um usuário é criado, um grupo com o nome do usuário também é criado. Isso explica o fato do “root root”, que indica que o arquivo pertence ao usuário root e ao grupo root.

Da oitava a décima letra (**r–**) diz qual é a permissão de acesso para os outros usuários. Neste caso todos os usuários que não são donos do arquivo teste tem a permissão somente para ler (**r**) o arquivo.

O primeiro root indica o nome do dono do arquivo.

O segundo root indica o nome do grupo que o arquivo pertence.

A palavra teste indica o nome da pasta.

Da esquerda para a direita nós temos as permissões do *dono*, *grupo e outros:*

|  |  |
| --- | --- |
| **Nome** | **Permissão** |
| Dono | rwx |
| Grupo | r-x |
| Outros | r-x |

Vendo a tabela, podemos entender que o dono tem acesso total ao diretório, o *grupo* tem acesso de **leitura** e **acesso** (acesso pois é um diretório), e **outros usuários** também têm permissão de **leitura** e **acesso**.

Etapas para o acesso ao arquivo/diretório

Para o acesso a um arquivo ou diretório da seguinte forma:

1. Verifica-se primeiro se o usuário que está acessará o arquivo é o dono, caso seja as permissões destinadas ao dono são aplicadas.

2. Se não for dono do arquivo/diretório, verifica-se o grupo do arquivo e do usuário caso pertençam ao mesmo grupo as permissões referentes ao grupo são aplicadas.

3. Se não pertencer ao grupo é verificado as permissões de acesso para os outros usuários e aplicadas.

Alterar permissões

Alterar permissões de acesso a um arquivo ou diretório é bem simples, basta usar o comando **chmod**.

Exemplo de sintaxe:

**chmod [opções] [permissões] <arquivo/diretório>**

O sistema entende com as indicações de usuários, grupo e outros usuários pelas letras **u**, **g** e **o** respectivamente e os símbolos **+** e **-** para adicionar ou remover as permissões.

Pequeno exemplo:

Ferrini-PC:~$ chmod g+x arquivo.txt

Ferrini-PC:~$ ls -l arquivo.txt

-rwxr-xr—1 Ferrini Ferrini 0 Set 17 21:57 arquivo.txt

Podemos ver que a permissão de execução daquele arquivo foi concedida ao grupo no qual o arquivo pertence.

Permissões octal

Nós também podemos alterar as permissões de acesso utilizando o modo octal, que é um conjunto de oito números onde cada número define um tipo de acesso diferente, ao invés de utilizar +x, -x, etc.

Tabela com as permissões em octal

|  |  |
| --- | --- |
| Octal | Permissão |
| 0 | *Nenhuma (-rwx)* |
| 1 | *Execução (x)* |
| 2 | *Escrita (w)* |
| 3 | *Escrita + Execução (wx)* |
| 4 | *Leitura (r)* |
| 5 | *Leitura + Execução (rx)* |
| 6 | *Leitura + Escrita (rw)* |
| 7 | *Todas (+rwx)* |

Exemplo:

Ferrini-PC:~$ chmod 701 arquivo.txt

Ferrini-PC:~$ ls -l

-rwxr-xr—1 Ferrini 0 Set 17 21:57 arquivo.txt

O primeiro número é destinado ao dono, o segundo ao grupo e o terceiro aos outros usuários, logo nesse exemplo nós demos todas as permissões ao dono, nenhuma ao grupo e execução para outros usuários.

Alterar Dono ou Grupo

Para alterarmos o dono ou grupo do arquivo/diretório também é bem simples de se fazer. Nós utilizamos o comando **chown** para esta ação.

Exemplo de sintaxe:

chown [opções] <dono/grupo> <diretório/arquivo>

Exemplo:

Ferrini-PC:~$ sudo chown root arquivo.txt

Ferrini-PC:~$ ls -l

-rwx-----x 1 root Ferrini 0 Set 17 21:57 arquivo.txt

Como é possível perceber trocamos o dono do arquivo para o usuário **root**.

Algumas opções de comando:

|  |  |
| --- | --- |
| **Opção** | **Função** |
| -R | inclui os subdiretórios caso seja um diretório. |
| -c | Exibe o resultado |

Para alterarmos o grupo do arquivo ou diretório a sintaxe é a seguinte.

**chown [opções] dono:grupo <arquivo/diretório>**

Exemplo:

Ferrini-PC:~$ sudo chown root:root arquivo.txt

Ferrini-PC:~$ ls -l

-rwx-----x 1 root Ferrini 0 Set 17 21:57 arquivo.txt

Definição de comandos:

* -R: Recursivo, altera as permissões de todo o conteúdo da pasta, opcional.
* u: Permissões para o dono da pasta (usuário)
* g: Permissões para o grupo (grupo)
* o: Permissões para os demais (outros)
* +: Adiciona a permissão
* -: Remove a permissão
* r: Permissão de leitura
* w: Permissão de gravação
* x: No caso dos arquivos indica permissão de escrita e, no caso da pasta, permissão para ver o conteúdo.