



LINUX

PERMISSÕES DE ARQUIVOS E DIRETÓRIOS

INTRODUÇÃO

- ▶ Permissões protegem o sistema e os arquivos dos usuários;
- ▶ Somente o super-usuário (root) tem ações irrestritas no sistema;
- ▶ Cabe ao root determinar o que cada usuário pode executar, criar, modificar, etc. (normalmente através de permissões).

Entendendo as permissões

ls -l (para listar um diretório e suas permissões)

```
drwx----- ... 2 user ..... 512 Jan ... 29 23:30 .. Arquivos/  
-rw-rw-r-- ... 1 user ..... 280232 Dec .. 16 22:41... notas.txt
```

Primeiro Caractere	
Caractere	Significado
d	Diretório
b	arquivo de bloco
c	arquivo especial de caractere
p	canal
s	socket
-	arquivo "normal"

Entendendo as permissões

```
drwx----- ... 2 user ..... 512 Jan ... 29 23:30 .. Arquivos/  
-rw-rw-r-- ... 1 user ..... 280232 Dec .. 16 22:41... notas.txt
```

3 Partes de 3 Caracteres	
Partes	Significado
1ª parte	Permissões do Proprietário
2ª parte	Permissões do Grupo ao qual usuário pertence
3ª parte	Permissões para os demais usuários

Significados dos Caracteres	
Caractere	Significado
r	Permissão de Leitura (read)
w	Permissão de Gravação (write)
x	Permissão de Execução (execution)
-	Permissão Desabilitada

Entendendo as permissões

- ▶ A tabela abaixo mostra as permissões mais comuns:

Resumo das Permissões	
Conjuntos	Significado
---	Nenhuma permissão
r--	Permissão de leitura
r-x	leitura e execução
rw-	leitura e gravação
rwX	leitura, gravação e execução.

Configurando permissões com chmod (change mode)

- ▶ Há duas maneiras de configurar as permissões através deste comando:
 - ▶ Simbolicamente
 - ▶ Numericamente

Configurando Permissões: Simbolicamente

Exemplos:

```
chmod u+w teste.old
```

```
chmod g+rw teste.old
```

```
chmod g=rwx teste.old
```

Lista 1		Lista 2	
Símbolo	Significado	Símbolo	Significado
u	Usuário	r	Leitura
g	Grupo	w	Gravação
O	Outro	x	execução

Operador	Significado
+	Adicionar permissão
-	Remover permissão
=	Definir permissão

Configurando Permissões: Numericamente

Exemplo:

`chmod 600 notas.txt`

Permissão	Binário	Decimal
---	000	0
--x	001	1
-w-	010	2
-wx	011	3
r--	100	4
r-x	101	5
rw-	110	6
rwX	111	7

Permissão	Decimal
-----	000
r-----	400
r--r--r--	444
rw-----	600
rw-r--r--	644
rw-rw-rw-	666
rwX-----	700
rwXr-x---	750
rwXr-xr-x	755
rwXrwXrwX	777

Exercício

- ▶ Crie arquivos e diretórios. Em seguida, teste a combinação de permissões com **chmod**. Isso lhe ajudará muito no entendimento deste recurso.
- ▶ Comando de criar diretório:
 - ▶ mkdir
- ▶ Comando de criar arquivo de texto:
 - ▶ touch arquivo.txt

Permissões Padrão (umask)

- ▶ Máscara de criação de arquivos (umask) é responsável por controlar as permissões padrão dos arquivos.
- ▶ Quando essa instrução é utilizada, ela pode, por exemplo, mostrar o valor 07. Neste caso, acrescenta-se um 0 à esquerda, transformando o valor em 007 (sempre que houver uma situação como essa, deve-se interpretar como se houve mais um zero à esquerda). Mas isso não significa que o arquivo em questão tenha sido criado sem nenhuma permissão para o proprietário e para o grupo, e todas permissões para os demais. Na verdade, significa justamente o contrário.

Permissões Padrão (umask)

- ▶ Transformando 007 em binário:
 - ▶ $007 \rightarrow 000\ 000\ 111$
- ▶ Alinhando a este valor o equivalente a todas permissões
 - ▶ $777 \rightarrow 111\ 111\ 111$
- ▶ Ao somar:
 - ▶ $000\ 000\ 111$
 - ▶ $111\ 111\ 111 +$
 - ▶ $111\ 111\ 000$ (o resultado é 770)

Permissões Padrão (umask)

- ▶ Ou seja, para ter 770 como permissão padrão, se usa umask 007, o oposto.
- ▶ Exemplo:
 - ▶ **umask 027 pasta**
- ▶ Faz com que todo arquivo criado nessa pasta receba a permissão 750. (porém, a permissão de execução não é atribuída para arquivos que não sejam executáveis).

Alterando a propriedade de um arquivo/diretório (chown)

- ▶ O procedimento para alterar a propriedade de um arquivo consiste em primeiro informar quem será o novo proprietário e então indicar o arquivo ou diretório.
 - ▶ **chown novo_proprietario arquivo.txt**
- ▶ Em muitos casos você não terá permissão para executar essa transferência, por motivos de segurança, o root que poderá fazê-lo.
- ▶ Para isso utilize o comando **sudo** seguido da instrução.
 - ▶ **sudo chown novo_proprietario arquivo.txt**
- ▶ Em poucas palavras, o *sudo* é uma espécie de lista de autorização para que o seu e outros usuários possam executar tarefas de poder administrativo. Se seu usuário tiver autorização para isso, o *sudo* pedirá que você informe sua senha na primeira vez que o comando for utilizado.

Alterando o grupo de um arquivo/diretório (chgrp)

- ▶ Também é possível alterar o grupo de um arquivo ou diretório. Para isso, pode ser necessário identificar o grupo no qual o arquivo pertence. Tal procedimento pode ser feito com o comando
 - ▶ **ls -lg nome do arquivo/diretório**
- ▶ Aqui novamente pode ser necessário usar o comando sudo antes da instrução
 - ▶ **sudo chgrp novo_grupo arquivo.txt**
- ▶ É importante citar que você deve ser membro do grupo do qual o arquivo pertencia para poder transferí-lo.

Exercício

- ▶ Crie um diretório chamado GSO e atribua a ele a permissão padrão 644.
- ▶ Em seguida, crie dois arquivos e um novo diretório dentro de GSO. Verifique se receberam a permissão padrão. (ls -l)
- ▶ Altere a permissão de um dos arquivos para `rwxr-x---`.