



Permissões chmod

Prof. Ricardo
Prof. Wesley

As permissões são um dos aspectos mais importantes do Linux (na verdade, de todos os sistemas baseados em Unix). Elas são usadas para vários fins, mas servem principalmente para proteger o sistema e os arquivos dos usuários. Manipular permissões é uma atividade interessante, mas complexa ao mesmo tempo. Mas tal complexidade não deve ser interpretada como dificuldade e sim como possibilidade de lidar com uma grande variedade de configurações, o que permite criar vários tipos de proteção a arquivos e diretórios.

Como você deve saber, somente o super-usuário (*root*) tem ações irrestritas no sistema, justamente por ser o usuário responsável pela configuração, administração e manutenção do Linux. Cabe a ele, por exemplo, determinar o que cada usuário pode executar, criar, modificar, etc. Naturalmente, a forma usada para especificar o que cada usuário do sistema pode fazer é a determinação de permissões. Sendo assim, neste artigo você verá como configurar permissões de arquivos e diretórios, assim como modificá-las.

Entendendo as permissões

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

As linhas acima representam um comando digitado (*ls -l*) para listar um diretório e suas permissões. O primeiro item que aparece em cada linha (**drwx-----** e **-rw-rw-r--**) é a forma usada para mostrar as permissões do diretório *Arquivos* e do arquivo *notas.txt*. É esse item, que recebe o nome de *string*, que vamos estudar. Um ponto interessante de citar é que o Linux trata todos os diretórios como arquivos também, portanto, as permissões se aplicam de igual forma para ambos. Tais permissões podem ser divididas em quatro partes para indicar: tipo, proprietário, grupo e outras permissões. O primeiro caractere da string indica o tipo de arquivo: se for "**d**" representa um diretório, se for "-" equivale a um arquivo.

Você já sabe o que significa o primeiro. Os demais são divididos em três grupos de três, cada um representando o proprietário, o grupo e todos os demais, respectivamente. Tomando a linha 2 do exemplo (**-rw-rw-r-**), desconsiderando o primeiro caractere e dividindo a string restante em 3 partes, ficaria assim:

rw- => a primeira parte significa permissões do proprietário
rw- => a segunda parte significa permissões do grupo ao qual o usuário pertence
r-- => a terceira parte significa permissões para os demais usuários

Vamos entender agora o que significa esses caracteres (**r**, **w**, **x**, **-**). Há, basicamente, três tipos de permissões: **leitura**, **gravação** e **execução**. Leitura permite ao usuário ler o conteúdo do arquivo mas não alterá-lo. Gravação permite que o usuário altere o arquivo. Execução, como o nome diz, permite que o usuário execute o arquivo, no caso de ser executável. Mas acontece que as permissões não funcionam isoladamente, ou seja, de forma que o usuário tenha ou permissão de leitura ou de gravação ou de execução.

As permissões funcionam em conjunto. Isso quer dizer que cada arquivo/diretório tem as três permissões definidas, cabendo ao dono determinar qual dessas permissões é habilitada para os usuários ou não. Pode ser que uma determinada quantidade de usuários tenha permissão para alterar um arquivo, mas outros não, por exemplo. Daí a necessidade de se usar grupos. No caso, a permissão de gravação desse arquivo será dada ao grupo, fazendo com que todo usuário membro dele possa alterar o arquivo. Note que é necessário ter certo cuidado com as permissões. Por exemplo, do que adianta o usuário ter permissão de gravação se ele não tem permissão de leitura habilitada?

Agora que já sabemos o significado das divisões da string, vamos entender o que as letras **r**, **w**, **x** e o caractere - representam:

r => significa permissão de leitura (read);

w => significa permissão de gravação (write);

x => significa permissão de execução (execution);

- => significa permissão desabilitada.

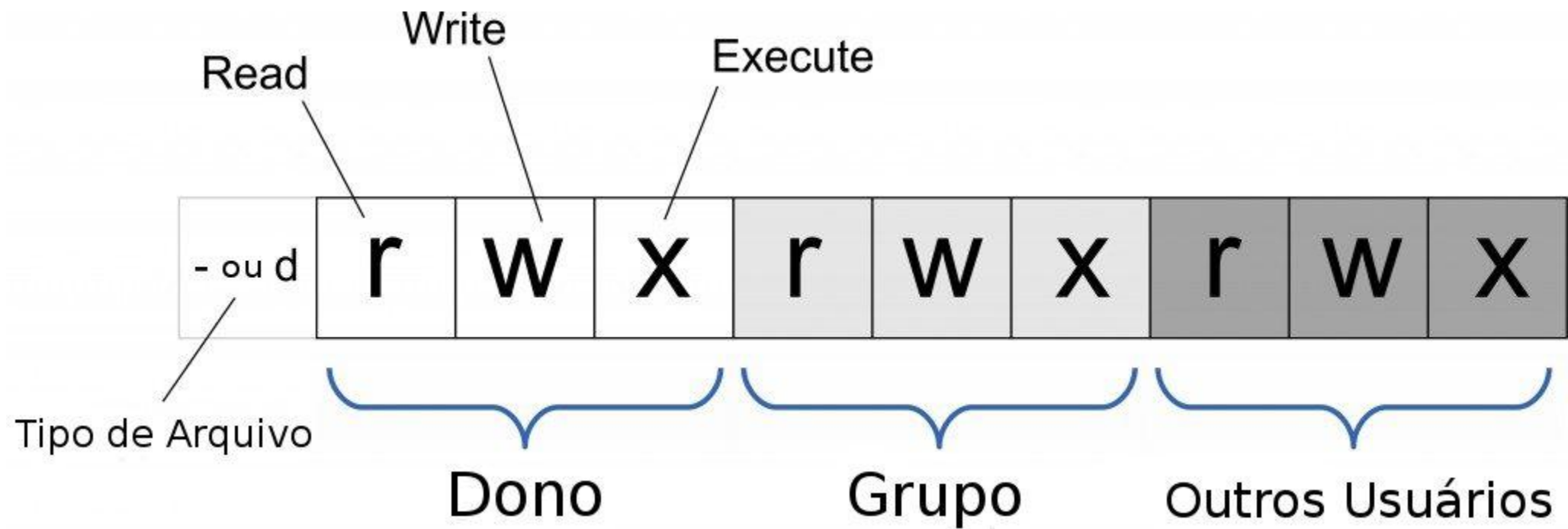
A ordem em que as permissões devem aparecer é **rwX**. Sendo assim, vamos entender a string do nosso exemplo:

Linha 1:

drwx..... 2 wester 512 Jan ... 29 23:30 .. Arquivos/

- é um diretório (**d**);
- o proprietário pode alterá-lo, gravá-lo e executá-lo (**rwX**);
- o grupo não pode alterá-lo, gravá-lo, nem executá-lo (---);
- os demais usuários não podem alterá-lo, gravá-lo, nem executá-lo (---).

```
ricardo@ricardo-VirtualBox:~$ ls -l
total 108
drwxrwxr-x 2 ricardo ricardo 4096 jun  8 14:36 letim
drwxrwxr-x 3 ricardo ricardo 4096 jul 19 07:26 letimDS2021
drwxrwxr-x 2 ricardo ricardo 4096 jun  8 14:30 2etim
drwxrwxr-x 2 ricardo ricardo 4096 jun  8 14:30 3etim
drwxr-xr-x 2 ricardo ricardo 4096 mai 17 07:36 'Área de Trabalho'
drwxrwxr-x 2 ricardo ricardo 4096 jun 15 14:37 arquivos
-rw-rw-r-- 1 ricardo ricardo 2589 jul 27 14:33 arquivoteste
drwxr-xr-x 2 ricardo ricardo 4096 mai 24 07:42 Documentos
drwxr-xr-x 2 ricardo ricardo 4096 mai 17 07:36 Downloads
drwxrwxr-x 2 ricardo ricardo 4096 jun  7 07:23 ds
drwxr-xr-x 2 ricardo ricardo 4096 mai 17 07:36 Imagens
drwxrwxr-x 2 ricardo ricardo 4096 jun  7 07:23 info
-rw-rw-r-- 1 ricardo ricardo  98 jun 13 16:29 listanomes
-rw-rw-r-- 1 ricardo ricardo  81 jun 14 07:30 listaordenada
drwxr-xr-x 2 ricardo ricardo 4096 mai 17 07:36 Modelos
drwxr-xr-x 2 ricardo ricardo 4096 mai 17 07:36 Música
-rw-rw-r-- 1 ricardo ricardo  81 jun 14 07:28 nomes
drwxrwxr-x 2 ricardo ricardo 4096 mai 31 07:22 nutella
-rw-rw-r-- 1 ricardo ricardo  354 jun 13 16:10 poema
-rw-rw-r-- 1 ricardo ricardo  355 jun 14 07:20 poeta
drwxr-xr-x 2 ricardo ricardo 4096 mai 17 07:36 Público
drwxrwxr-x 3 ricardo ricardo 4096 jul 19 07:32 raiz
drwxrwxr-x 2 ricardo ricardo 4096 jul 19 07:35 teste
drwxrwxr-x 2 ricardo ricardo 4096 mai 31 07:22 toddy
-rw-rw-r-- 1 ricardo ricardo  35 jul 27 14:29 txt
drwxrwxr-x 3 ricardo ricardo 4096 jun  7 07:40 vestibulinho
drwxr-xr-x 2 ricardo ricardo 4096 mai 17 07:36 Vídeos
```

PERMISSÕES POSSÍVEIS

DECIMAL	BINÁRIO			DESCRIÇÃO
	R (LEITURA)	W (ESCRITA)	X (EXECUTAR)	
0	0	0	0	Nenhuma permissão
1	0	0	1	Permissão apenas para Executar
2	0	1	0	Permissão apenas para Gravar
3	0	1	1	Permissão para Gravar e Executar
4	1	0	0	Permissão apenas para Ler
5	1	0	1	Permissão para Ler e Executar
6	1	1	0	Permissão para Ler e Gravar
7	1	1	1	Permissão total (Ler, Gravar, Executar)

Configurando permissões com chmod

Nos tópicos anteriores você deve ter tido pelo menos uma noção do que são permissões e de sua importância no Linux. Chegou a hora de aprender a configurar permissões e isso é feito através do comando **chmod** (de *change mode*). Um detalhe interessante desse comando é que você pode configurar permissões de duas maneiras: simbolicamente e numericamente. Primeiramente veremos o método simbólico.

Para ter uma visão mais clara da forma simbólica com o chmod, imagine que tais símbolos se encontram em duas listas, e a combinação deles gera a permissão:

Lista 1

Símbolo

u => usuário

g => grupo

O (letra 'o' maiúscula) => outro

a => todos

Lista 2

Símbolo

r => leitura

w => gravação

x => execução

Para poder combinar os símbolos destas duas listas, usam-se os operadores:

+ (sinal de adição) => adicionar permissão

- (sinal de subtração) => remover permissão

= (sinal de igualdade) => definir permissão

Para mostrar como essa combinação é feita, vamos supor que você deseje adicionar permissão de gravação no arquivo *teste.old* para um usuário. O comando a ser digitado é:

chmod u+w teste.old

O "u" indica que a permissão será dada a um usuário, o sinal de adição (+) indica que está sendo adicionada uma permissão e "w" indica que a permissão que está sendo dada é de gravação. Caso você queira dar permissões de leitura e gravação ao seu grupo, o comando será:

chmod g+rw teste.old


Agora, vamos supor que o arquivo *teste.old* deverá estar com todas as permissões disponíveis para o grupo. Podemos usar então:

chmod g=rwx teste.old

Repare que o arquivo teste.old tem permissões rwx para o grupo

Dica: crie arquivos e diretórios. Em seguida, teste a combinação de permissões com chmod. Isso lhe ajudará muito no entendimento deste recurso.

```
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
alecrim@alecrim-eee:~$ chmod g=rwx teste.old
alecrim@alecrim-eee:~$ ls -l
total 856
-rw-r--r-- 1 alecrim alecrim 21731 2010-01-27 19:36 apipad.jpg
-rw-r--r-- 1 alecrim alecrim 252223 2010-01-27 19:34 apipad.png
drwxr-xr-x 2 alecrim alecrim 4096 2009-12-30 10:00 Desktop
drwxr-xr-x 2 alecrim alecrim 4096 2009-01-20 19:02 Documentos
-rw-r--r-- 1 alecrim alecrim 556688 2009-01-13 22:44 Firefox_wallpaper.png
drwxr-xr-x 4 alecrim alecrim 4096 2009-01-24 15:00 Imagens
drwxr-xr-x 2 alecrim alecrim 4096 2009-01-11 19:53 Modelos
drwxr-xr-x 2 alecrim alecrim 4096 2009-01-11 19:53 Músicas
drwxr-xr-x 2 alecrim alecrim 4096 2009-01-11 19:53 Pública
-rw-rwxr-- 1 alecrim alecrim 197 2009-01-11 19:51 teste.old
drwxr-xr-x 3 alecrim alecrim 4096 2009-01-12 15:59 Videos
alecrim@alecrim-eee:~$
```



Usando chmod com o método numérico

Usar o chmod com valores numéricos é uma tarefa bastante prática. Em vez de usar letras como símbolos para cada permissão, usam-se números. Se determinada permissão é habilitada, atribui-se valor 1, caso contrário, atribui-se o valor 0. Sendo assim, a string de permissões **r-xr....** na forma numérica fica sendo **101100000**. Essa combinação de 1 e 0 é um número binário. Mas temos ainda que acrescentar a forma decimal (ou seja, números de 0 a 9). Para isso, observe a tabela abaixo:

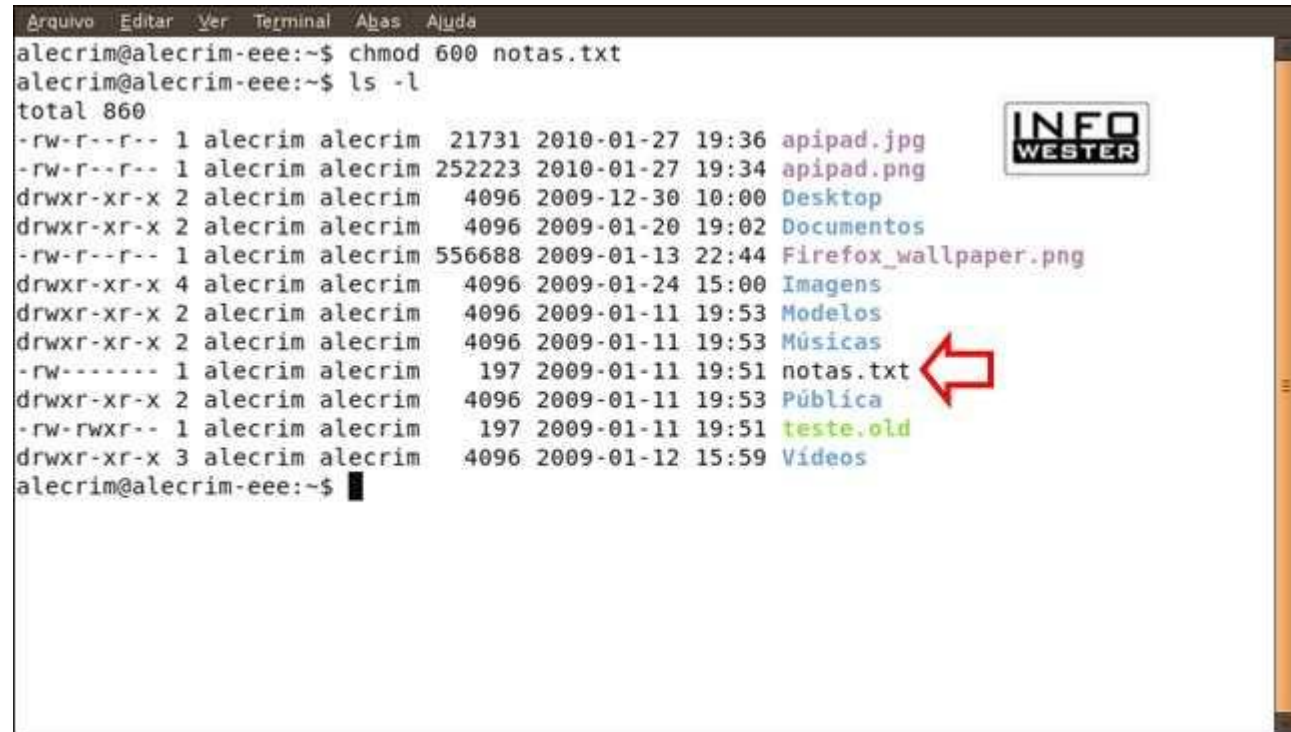
Permissão	Binário	Decimal
---	000	0
--x	001	1
-w-	010	2
-wx	011	3
r--	100	4
r-x	101	5
rw-	110	6
rwX	111	7

Como o sistema binário somente trabalha com os números 0 e 1 (decimal trabalha com os números de 0 a 9, ou seja, é o sistema de numeração que utilizamos no nosso cotidiano), ele precisa de uma sequência para representar os valores. Sendo assim, na tabela acima, a coluna *Binário* mostra como são os valores binários dos números de 0 a 7 do sistema decimal. **(aprendemos a trabalhar com binário no início do ano)**

Chegou a hora então de relacionar a explicação do parágrafo acima com a coluna *Permissão*. Para exemplificar, vamos utilizar a permissão **rw-**, cujo valor em binário é **110**, que por sua vez, em decimal corresponde ao número **6**. Então, em vez de usar **rw-** ou **110** para criar a permissão, simplesmente usa-se o número **6**. Repare que, com o método numérico, usamos somente um dígito para representar uma permissão, em vez de três. Assim sendo, a string de permissões **r--r--r--** pode ser representada por **444**, pois **r--** em decimal é igual a quatro.

Observe o exemplo abaixo:

chmod 600 notas.txt



```
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
alecrim@alecrim-eee:~$ chmod 600 notas.txt
alecrim@alecrim-eee:~$ ls -l
total 860
-rw-r--r-- 1 alecrim alecrim 21731 2010-01-27 19:36 apipad.jpg
-rw-r--r-- 1 alecrim alecrim 252223 2010-01-27 19:34 apipad.png
drwxr-xr-x 2 alecrim alecrim 4096 2009-12-30 10:00 Desktop
drwxr-xr-x 2 alecrim alecrim 4096 2009-01-20 19:02 Documentos
-rw-r--r-- 1 alecrim alecrim 556688 2009-01-13 22:44 Firefox_wallpaper.png
drwxr-xr-x 4 alecrim alecrim 4096 2009-01-24 15:00 Imagens
drwxr-xr-x 2 alecrim alecrim 4096 2009-01-11 19:53 Modelos
drwxr-xr-x 2 alecrim alecrim 4096 2009-01-11 19:53 Músicas
-rw----- 1 alecrim alecrim 197 2009-01-11 19:51 notas.txt
drwxr-xr-x 2 alecrim alecrim 4096 2009-01-11 19:53 Pública
-rw-rwxr-- 1 alecrim alecrim 197 2009-01-11 19:51 teste.old
drwxr-xr-x 3 alecrim alecrim 4096 2009-01-12 15:59 Videos
```

The screenshot shows a terminal window with a menu bar (Arquivo, Editar, Ver, Terminal, Abas, Ajuda) and a title bar (alecrim@alecrim-eee). The user runs 'chmod 600 notas.txt' and then 'ls -l'. The output shows a list of files and directories with their permissions, owner, group, size, date, and time. A red arrow points to the line for 'notas.txt', which now has permissions '-rw-----'.

Acima, estão sendo dadas as permissões `rw-----` ao arquivo *notas.txt*, pois **6** equivale a **rw-** e **0** equivale a `---`. Como zero aparece duas vezes, forma-se então o valor 600. Faça o comando acima com um arquivo de teste e depois digite `ls -l notas.txt` para ver o que aparece (*notas.txt* deve ser substituído pelo arquivo que você está usando). A tabela abaixo mostra uma lista de configurações bastante utilizadas:

<code>-----</code>	000
<code>r-----</code>	400
<code>r--r--r--</code>	444
<code>rw-----</code>	600
<code>rw-r--r--</code>	644
<code>rw-rw-rw-</code>	666
<code>rwX-----</code>	700
<code>rwXr-x---</code>	750
<code>rwXr-xr-x</code>	755
<code>rwXrwxrwx</code>	777

Fonte:

<https://www.infowester.com/linuxpermissoes.php>

As três últimas permissões da tabela são comumente usadas para programas e diretórios.