

sobre o site | contato

Pesquisar

Início | Hardware | Software | Redes | Linux | Dicas | Tutoriais | Web | Notícias | Blog | Dúvidas

Notícias

- 10/9/2013 06 de outubro: vem aí a edição 2013 do Video Games Live
- 19/8/2013 Cielo tem nova edição de seu programa de trainees - Salário inicial é de R\$ 4,5 mil
- 19/8/2013 Confirmado: o Windows 8.1 chega em 18 de outubro
- 13/8/2013 Brasil perde apenas para os Estados Unidos em malwares para internet banking
- 06/8/2013 Microsoft abre inscrições para programa de estágio no Brasil

Blog

- 13/8/2013 Dica: documentário on-line alerta para riscos de se usar o celular ao dirigir
- 13/7/2013 Alcatel anuncia novos smartphones Android para o mercado brasileiro
- 10/7/2013 Um terminal Linux on-line gratuito, mas que se "autodestrói" em 35 minutos
- 02/7/2013 Adrift, o fantástico curta-metragem criado por um animador da Pixar
- 26/6/2013 Build 2013: Microsoft apresenta as novidades do Windows 8.1

facebook

Find us on Facebook

InfoWester

Like

13,762 people like InfoWester.

Facebook social plugin

publicidade

publicidade

Experiência Alphaville

Alphaville.com.br/Experiência

Lotes Limitados Condições Especiais Experiências Inesquecíveis! Acesse

Anúncios Google Linux windows Linux ubuntu Unix linux Debian linux

Início » GNU/Linux »

Entendendo e usando permissões no Linux

10

31

5

Tweet

Like



publicidade



Introdução

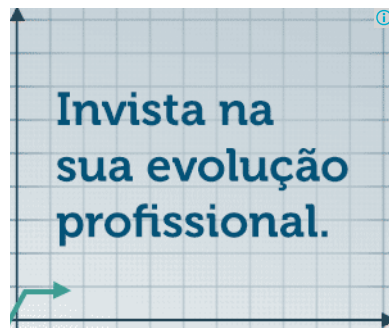
As permissões são um dos aspectos mais importantes do Linux (na verdade, de todos os sistemas baseados em Unix). Elas são usadas para vários fins, mas servem principalmente para proteger o sistema e os arquivos dos usuários. Manipular permissões é uma atividade interessante, mas complexa ao mesmo tempo. 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.

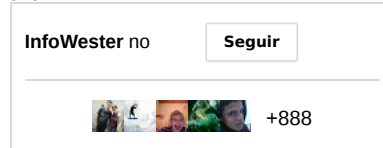
Entendendo as permissões

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,



google+



publicidade



Imperdível FIAT Completo!
Novo Palio Attractive 1.0 Flex 2014 4P DH AC ABS Air Bag só R\$32.990
ofertas.fiat.com.br



Vestibular UNINOVE
Processo Seletivo para Graduação presencial e EAD. Inscreva-se já!
www.UniNove.br

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. Entretanto, outros caracteres podem aparecer para indicar outros tipos de arquivos, conforme mostra a tabela abaixo:

d => diretório
b => arquivo de bloco
c => arquivo especial de caractere
p => canal
s => socket
- => arquivo "normal"

Repare agora que no restante da string ainda há 9 caracteres. Você já sabe o que significa o primeiro. Os demais são divididos em três grupos de três, cada um representado 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 dividindo-a em 4 partes:

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 (**---**).

Linha 2:

```
-rw-rw-r-- ... 1 wester ..... 280232 Dec .. 16 22:41... notas.txt
```

- é um arquivo (**-**);
- o proprietário pode alterá-lo, gravá-lo, mas não executá-lo. Repare que como este arquivo não é executável, a permissão de execução aparece desabilitada (**rw-**);
- o grupo tem permissões idênticas ao proprietário (**rw-**);
- o usuário somente tem permissão de ler o arquivo, não pode alterá-lo (**r--**).

A tabela abaixo mostra as permissões mais comuns:

--- => nenhuma permissão;
r-- => permissão de leitura;
r-x => leitura e execução;
rw- => leitura e gravação;
rwX => leitura, gravação e execução.

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

```

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 Vídeos
alecrim@alecrim-eee:~$
  
```

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.

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

Se você não conhece o sistema binário deve estar se perguntando o que esse "monte" de 0 e 1 tem a ver com os números de 0 a 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.

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

Permissões **rw-----** no arquivo *notas.txt* com o comando **chmod 600**

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:

-----	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

rw-rw-rw- 777

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

Finalizando

Como você viu, é muito mais prático utilizar o `chmod` com o método numérico. Mas você pode ter ficado confuso com todo esse esquema de permissão. Mas não se sinta culpado por isso (e também não ponha toda a culpa na ineficiência do autor para explicar o assunto :D). A questão é que nos sistemas baseados em Unix, permissões são um dos aspectos mais complexos existentes. Tal complexidade é equivalente à eficiência do uso de permissões. Por isso, a melhor maneira de entender as permissões é treinando. Sendo assim, ao trabalho! Treine, crie permissões e veja seus resultados. Boa aprendizagem!

LIVROS SUGERIDOS:

- [Linux: referência para leigos](#)
- [Manual completo do Linux](#)
- [Comandos do Linux - Consulta rápida](#)
- [Outros livros sobre Linux](#)

Via Shopping UOL

[Acesse a segunda parte deste artigo clicando aqui.](#)

Escrito por [Emerson Alecrim](#) - Publicado em 08_02_2004 - Atualizado em 05_02_2010

[Anúncio Google](#) [Linux server root](#) [Sistema linux](#) [Linux de](#) [Pasta de arqui](#)

 [Voltar](#)

 [Página inicial](#)

 [Ir para o topo](#)

Mais em [Hardware](#) • [Software](#) • [Redes](#) • [GNU/Linux](#) • [Dicas](#) • [Tutoriais](#) • [Web Design](#) • [Notícias](#) • [Blog](#) • [Dúvidas](#)

Feed RSS

[InfoWester no Facebook](#)

[InfoWester no Twitter](#)

[InfoWester no Google+](#)

[Newsletter](#)

[Contato](#)

[Acompanhe o site](#)

Política de privacidade

[Condições de uso do site](#)

[Perguntas mais frequentes](#)

[Que site é este?](#)

[Quem faz este site?](#)

[Assessorias / Agências](#)

[Anuncie aqui](#)

Search

Custor

Conteúdo disponível sob uma licença Creative Commons.
Saiba mais nas [condições de uso dos textos](#).

InfoWester 2013 | *Conhecimento tecnológico ao seu alcance*

Responsável: Emerson Alecrim | Layout: [Erika Sarti](#) | No ar desde 2001