

Permissões, Tudo no Linux é um Arquivo!

Prof. Me Robson Ferreira Lopes

rferreira@ifsp.edu.br

IFSP Câmpus Guarulhos

Introdução

No sistema operacional GNU/Linux trata todas as partes do sistema com arquivos, ou seja, dispositivos de hardwares, a comunicação entre os processos e todas as tarefas do sistema operacional são realizados por arquivos.

O sistema de arquivos associado ao Linux mantém para cada arquivo informações do dono do arquivo, do grupo do dono, do tipo de arquivo, as permissões, tamanho, data e hora que foram modificados ou criados, etc.

Cada arquivo possui três permissões básicas r (read – Leitura), w (write – escrita) e x (execute – execução) e tem o formato que iremos analisar a seguir

Execute os comandos a abaixo:

```
#echo permissoes dos arquivos > exemplo
#mkdir <diretório>
#ln -s /root/exemplo /tmp/linkexemplo
#ls -l | grep exemplo > tipos.txt
#ls -l | grep diretório >> tipos.txt
#ls -l /tmp | grep linkexemplo >> tipos.txt
#ls -l /dev/psaux >> tipos.txt
#ls -l /dev/sda >> tipos.txt
#cat tipos.txt
```

Formato do arquivo

-	rwx	rwx	rwx	1	root	root	1	Abr 27 14:32	Exemplo
1	2	3	4	5	6	7	8	9	10

Campo 1 – Tipos de Arquivos

- Arquivos Comum – São arquivos de texto, arquivos de dados e arquivos binários;

d Diretório – São arquivos que contém nomes de outros arquivos que estão armazenados, ou organizados em grupos

l links – são arquivos que fazem referencia a outros arquivos dentro do sistema de arquivos

c caractere – são arquivos que representam dispositivos físicos (Hardware)

b bloco – são arquivos que representam dispositivos físicos que podem ser acessados em blocos de bytes, por exemplo HD

s sockets – são arquivos de comunicação entre os processos

| (pipe) – São arquivos também utilizados na comunicação entre processos

Campos 2,3 e 4

Nos campos 2,3 e 4 temos as permissões para o dono do arquivo, grupo do dono e dos outros usuários respectivamente. Essas permissões são entendidas por um numero de 12 bits onde obviamente cada bit pode assumir 0 ou 1 indicando ausência ou presença de permissão

000 000 000 000

Bits de Atributo Bits de Proteção

Os bits de atributos são permissões especiais para arquivos e diretórios esses serão estudados posteriormente.

Bits de Proteção

São os bits que definem as pemições básicas de cada arquivo sendo:

000 000 000
Dono Grupo Outros

Podemos representar esses bits simbolicamente por:

rwX rwX rwX
Dono Grupo Outros

Assim cada grupo é formado por 3 bits podem ser representados através do sistema octal de numeração

r(4)	w(2)	x(1)	Total	Permissão	Descrição
0	0	0	0	---	Nenhuma
0	0	1	1	--x	Somente execução
0	1	0	2	-w-	Somente Escrita

0	1	1	3	-wx	Escrita e Execução
1	0	0	4	r--	Somente Leitura
1	0	1	5	r-x	Leitura e Execução
1	1	0	6	rw-	Leitura e escrita
1	1	1	7	rwX	Total

Atribuição das permissões é feita por meio do comando `chmod`

#man `chmod`

Sintaxe

`chmod <permissão> <arquivo>`

Exemplos:

#`chmod u+rw exemplo`

#`ls -l exemplo`

Atribui para o dono a permissão de leitura e escrita

#`chmod g-w exemplo`

#`ls -l exemplo`

Retira do grupo a permissão de escrita

#`chmod o-wx exemplo`

#`ls -l exemplo`

Retirar dos outros usuários a permissão de escrita e execução

#`chmod 644 exemplo`

#`ls -l exemplo`

Quando usamos notação octal podemos atribuir as três permissões de uma só vez onde no exemplo estamos atribuindo leitura e escrita para o dono, leitura para o grupo e leitura para outros usuários.

Campo 5

Quantidade de sub-diretórios do arquivo

#`mkdir ifsp`

#`mkdir -p ifspgru/{ads,tai}`

#`ls -l`

Campo 6

Usuário dono do arquivo

Para alterar o dono de um arquivo usamos o comando `chown`

Sintaxe

`chown <novo dono> <arquivo>`

exemplo

```
#ls -l exemplo
```

```
#chown sysadmin exemplo
```

```
#ls -l exemplo
```

Campo 7

Grupo do dono do arquivo

Podemos alterar o grupo do dono do arquivo através do comando `chgrp`

Sintaxe

`chgrp <novo grupo dono> <arquivo>`

Exemplo:

```
#ls -l exemplo
```

```
#addgroup usuarios
```

```
#chgrp usuarios exemplo
```

```
#ls -l exemplo
```

Campo 8

Tamanho do Arquivo

```
#echo <escreva seu nome> > nome.txt
```

```
#wc -m nome.txt
```

```
#ls -l nome.txt
```

```
#hexdump nome.txt
```

Campo 9

Data e hora de Criação

Campo 10

Nome do arquivo com até 255 letras

Mascara de criação de arquivos

No Linux por padrão um diretório pode ter permissão máxima 777 totais para todos os grupos de permissões e um arquivo pode ter permissão máxima 666 sendo leitura e

escrita para todos os grupos de permissões essas regras ajudam o Linux a ser um sistema seguro pois nunca um arquivo poderá ser criado por nenhum usuário com a permissão de execução direto.

Umask é a máscara usada pelo Linux para criação de arquivos e diretório que por padrão é 022 assim cada arquivo ou diretório criado com essa mascara terá a permissão máxima subtraída da mascara sendo assim cada arquivo nasce com a permissão leitura e escrita para o dono e somente leitura para o grupo e outros usuários e cada diretório é criado com permissão total para o dono e permissão de leitura e execução para o grupo e outros usuários.

A máscara fica armazenada no arquivo `/etc/profile`

Exemplo:

```
#umask
```

- visualiza a mascara atual

```
#umask 002
```

```
#touch teste
```

```
#ls -l
```

- Verifique a permissão inicial do arquivo teste

```
#mkdir grupo
```

```
#ls -l
```

- Verifique a permissão inicial do diretório grupo