

Aula __: Permissões de acesso a arquivos e diretórios

1 Objetivos

- Expor os conceitos associados as permissões de acesso a arquivos e diretórios.
- Explorar as permissões em nível do usuário proprietário(*owner*);
- Explorar as permissões em nível de grupos e outros;

2 Conceito de permissões de acesso a arquivos e diretórios

As permissões de acesso a arquivos e diretórios permitem proteger o sistema de arquivos do Linux do acesso indevido por pessoas ou programas não autorizados.

O princípio da segurança está baseado no conceito de **usuário dono ou proprietário, grupo e outros usuários**. Um arquivo sempre possui um usuário que é o seu **dono** (*owner*). Quando um usuário cria um arquivo ou diretório ele passa a ser o seu dono. No entanto, o arquivo poderá ser repassado a outro usuário.

O **grupo** (*group*) permite atribuir permissões de acesso a arquivos e diretórios comuns a um grupo de usuários. Os **outros** são usuários que não são donos nem pertencem ao grupo do arquivo ou diretório.

Nota: É bom lembrar que um usuário do sistema é identificado pelo seu *nome de login* ao qual existe associado um número chamado UID (userID). Uma coleção de usuários pode pertencer a um grupo. Um grupo também possui um nome e um identificador numérico (GID)

As permissões podem do tipo:

- **leitura** (*Read*) para arquivos, ou no caso de diretório listar seu conteúdo (por exemplo com `ls`);
- **escrita** (*Write*) no arquivo, ou no caso de diretório a criação de arquivos ou sub-diretórios dentro dele;
- **execução** (*eXec*) de arquivo (caso seja executável) ou de entrar para dentro do diretório (por exemplo com `cd`).

Nota: O acesso a um arquivo/diretório é feito verificando primeiro se o usuário que acessará o arquivo é o seu dono, caso seja, as permissões de dono do arquivo são aplicadas. Caso não seja o dono do arquivo/diretório, é verificado se ele pertence ao grupo correspondente, caso pertença, as permissões do grupo são aplicadas. Caso não pertença ao grupo, são verificadas as permissões de acesso para os outros usuários que não são donos e não pertencem ao grupo correspondente ao arquivo/diretório.

3 Verificando as permissões de acesso a arquivos e diretórios

1. Entrar em um terminal;
2. Verificar qual é o *diretório corrente* usando o comando:
`pwd`
3. Confirme o seu login name:
`whoami`
4. Criar um diretório *TestePermissoes* no diretório de entrada e entrar para o mesmo;
5. Criar no diretório corrente um sub-diretório chamado **dir** com o comando:
`mkdir dir`

6. Criar no diretório corrente um arquivo chamado **arq** com o comando:

```
touch arq
```

7. Listar o conteúdo do diretório corrente com o comando:

```
ls -l
```

8. Criar um link simbólico

```
ln -s arq ptr1
```

A saída do comando `ls -l` terá o seguinte formato:

```
-rw-r--r-- 1 aluno aluno    0 2010-04-06 13:01 arq
drwxr-xr-x 2 aluno aluno 4096 2010-04-06 13:01 dir
lrwxrwxrwx 1 aluno aluno    3 2010-04-06 13:02 ptr1 -> arq
```

Na saída acima, o primeiro dos 10 caracteres mostrados na coluna da esquerda identificam o tipo (- arquivo regular; d diretório; l link;) e os outros 9 mostram as permissões para o dono, grupo e outros.

Por exemplo, analisando a primeira linha acima, concluímos que o dono do arquivo **arq** (aluno) possui permissão de leitura e escrita (r w -) sobre o mesmo, o grupo (aluno) possui permissão de leitura (r - -) e os outros possuem permissão de leitura (r - -).

Quais as permissões para o diretório **dir**? O que há de diferente em relação ao arquivo **arq**?

4 Continuando a verificação de permissões

1. Mudar para o diretório `/home`, usando o comando:

```
cd /home
```

2. Listar o conteúdo com o comando:

```
ls -l;
```

3. Verifique quantos usuários diferentes possuem diretórios pessoais abaixo do `/home` e os grupos aos quais pertencem;

4. Quais as permissões dos diretórios pessoais abaixo do `/home`?

5. Mude para um dos diretórios pessoais abaixo do `/home` e tente criar um arquivo com o comando `touch`. O que acontece?

6. Mude para o diretório `/etc` e liste seu conteúdo com os comandos:

```
cd /etc
```

7. Verifique as permissões do arquivo `passwd` dentro do `/etc` usando `ls`;

8. Abra o arquivo `passwd` com o editor `gedit` e tente salvar. O que acontece? Saia do editor com `:q!`

Nota: O administrador do sistema (usuário `root`), possui permissão completa sobre o sistema de arquivo do Linux”

5 Mudando as permissões de acesso a arquivos e diretórios

O comando `chmod` (*change mode*) permite alterar as permissões de acesso a arquivos e diretórios. Há duas formas de utilizar o comando `chmod`: utilizando parâmetros no **formato octal** ou com o **formato simbólico**.

chmod: formato octal

No formato octal passa-se como parâmetro para o `chmod` três algarismos octais, os quais definem as permissões para o **dono**, **grupo** e **outros**, conforme a tabela abaixo:

	leitura	escrita	execução	octal
-	-	-	-	0
-	-	-	x	1
-	w	-	-	2
-	w	x	-	3
r	-	-	-	4
r	-	-	x	5
r	w	-	-	6
r	w	x	-	7

Por exemplo, o comando:

```
chmod 770 arq
```

atribui permissões de leitura, escrita e execução para o dono e grupo do arquivo **arq** e nenhuma permissão para os outros.

Vamos fazer alguns exercícios:

1. Mudar para o diretório `TestePermissoes`, com o comando `cd`;
2. Criar com `gedit` um arquivo `MeuShellScript.sh` e colocar as linhas:

```
#!/bin/bash
echo Alo Mundo
echo Estou testando permissoes
```

3. Verifique as permissões deste arquivo com `ls -l`. Você tem direito a executar este programa? Tente executá-lo:
`./MeuShellScript.sh`
4. Coloque direito de execução, leitura e escrita somente para o dono retirando estes direitos para o grupo e para outros:
`chmod 700 MeuShellScript.sh`
5. Confirme estes direitos usando o `ls -l`
6. Execute o programa:
`./MeuShellScript.sh`
7. Retire o direito de leitura e execução e tente ler e depois executar o arquivo:
`chmod 200 MeuShellScript.sh`
`cat MeuShellScript.sh`
`./MeuShellScript.sh`
8. Recoloque o direito de leitura e execução mas RETIRE o de escrita e tente acrescentar alguma coisa no arquivo usando o `gedit`. Substitua o `XXX` adequadamente:
`chmod XXX MeuShellScript.sh`
9. Mude as permissões do diretório **dir** deixando unicamente direito de leitura e escrita para o dono:
`chmod XXX dir`
10. Tente mudar para o diretório **dir** usando `cd`. O que acontece?
11. Tente copiar algo para dentro do diretório:
`cp MeuShellScript.sh dir`
12. Tentar entrar para o didretório `dir` usando `cd`;
13. Coloque permissões de escrita no diretório **dir** de forma que possa novamente entrar neste diretório.
`chmod 777 dir`

6 Usando modo simbólico e testando permissões para o usuário dono

1. Entrar novamente para o diretório `TestePermissoes`:
2. Listar as permissões do diretório `dir`:
`ls -ld dir`
3. Criar um diretório teste e remover o direito de entrar para o mesmo com `cd` em nível de usuário proprietário:
`chmod u-x teste`

4. Tentar entrar para o diretório teste com *cd*. O que acontece?
cd teste
5. Colocar novamente a permissão de execução:
chmod u+x teste
6. Retirar a permissão de escrita:
chmod u-w teste
7. Mas mesmo sem este direito você pode entrar e sair do diretório:
cd teste
cd ..
8. No entanto não pode colocar nada lá dentro:
touch dir/alfa.txt
9. Recoloque o direito de escrita:
chmod u+w teste
10. Entre para o diretório
cd teste
11. Crie um arquivo com gedit chamado beta.txt e coloque algum texto dentro. Salve e saia.
12. Retire a permissão de leitura:
chmod u-r beta.txt
13. Faça um comand cat para ler o conteúdo para a tela:
cat beta.txt
14. Coloque novamente a permissão:
chmod u+r beta.txt
15. Retire a permissao de escrita do usuário:
chmod u-w beta.txt
16. Com o gedit edite o arquivo e tente salvar.
17. Recoloque a permissão de escrita.
chmod u+w beta.txt
18. Copie o comando ls para o seu diretório:
cp /bin/ls .
19. Liste com:
ls -l
20. Mude o nome do comando:
mv ./ls ./MeuLS
21. Execute o comando com:
./MeuLS
22. Retire o direito de execução deste comando:
chmod u-x ./MeuLS
23. Tente executar novamente o comando
./MeuLS

7 Permissões de acesso a arquivos e diretórios para grupos e outros

Nos exercícios que se seguem exploraremos o uso de permissões em relação a grupos e outros. Na notação simbólica podemos alterar permissões de grupo e de outros fazendo:

chmod g+x alfa.txt #acrescenta direito de escrita para o grupo

chmod o-w alfa.txt #retira o direito de escrita para outros

chmod go-rw alfa.txt #retira o direito de leitura e escrita para grupos e outros

chmod a-x alfa.txt #retira o direito de execução para TODOS

chmod ug=rw alfa.txt #Atribui direito de escrita e leitura e remove o de execução para proprietário e grupo

Usando a notação octal podemos forçar as permissões:

chmod 777 alfa.txt #acrescenta direito de leitura, escrita e execução para usuário, grupo e outros.

8 Recordando a modificação das permissões para usuário

Na aula passada trabalhamos a modificação de permissões para o proprietário do arquivo. Vamos recordar um pouco:

1. Logar no sistema;
2. Criar um diretório:
`mkdir teste`
3. Entrar para o diretório:
`cd teste`
4. Criar um arquivo e inserir dados. Salvar e sair:
`pico alfa.txt`
5. Conferir as permissões:
`ls -l`
6. Retirar as permissões de leitura para o usuário proprietário:
`chmod u-r alfa.txt`
7. Conferir as permissões:
`ls -l`
8. Tentar ler o arquivo:
`cat alfa.txt`
9. Acrescentar novamente as permissões agora com notação octal:
`chmod 644 alfa.txt`
10. Conferir e ler:
`ls -l`
`cat alfa.txt`

9 Preparando um ambiente para criação de usuários e grupos

1. Criar uma máquina virtual fazendo:
`vstart -xterm=gnome M1`

Nota: Nesta máquina você pode ser superusuário e fazer o que você quiser!

2. Verifique quem você é e onde você está:
`whoami`
`pwd`
3. Crie uma conta para *joao*, *maria* e *lara*. Coloque o mesmo nome como senha.
`adduser joao`
`adduser maria`
`adduser lara`

Nota: Fazer ENTER para todas as perguntas

4. Confira o diretório `home` para verificar os diretórios criados para estas contas.
`ls -l /home`
5. Crie dois grupos:
`addgroup projx`
`addgroup projy`
6. Adicione os usuários aos grupos
`addgroup joao projx`
`addgroup maria projx`
`addgroup joao projy`
`addgroup lara projy`

10 Mudando e testando permissões de outros

1. Logar como *joao* fazendo:
login joao
2. Criar um arquivo *alfa.txt* com o editor *pico* e coloque algum texto dentro do arquivo. Verificar permissões, proprietário e grupo. :
pico alfa.txt
ls -l
3. Retirar o direito de leitura e escrita, para outros, de *alfa.txt*
chmod o-rx alfa.txt
4. Sair da conta *joao*
exit
5. Logar como *maria* e conferir quem é e onde está:
login maria
whoami
pwd
6. Como *maria* entrar no diretório de entrada do usuário *joao*:
cd /home/joao
7. Fazer um *cat* no arquivo *alfa.txt*:
cat alfa.txt

Nota: Por que não foi possível esta operação? Neste momento você está acessando o arquivo como *other* (outros)

8. Saia da conta *maria* e entre novamente como *joao*. Dê permissão para que outros possam ler e escrever:
chmod o+rw alfa.txt
9. Ainda como *joao*, retire os seus direitos como proprietário:
chmod u-rw alfa.txt
10. Tente acessar o conteúdo do arquivo:
cat alfa.txt

Nota: Situação interessante onde o próprio dono do arquivo não tem acesso sobre o seu arquivo mas outros usuários podem acessá-lo

11. Saia da conta *joao* e logue como *maria*
exit login maria
12. Vá até o diretório de *joao*, edite e salve algum texto em *alfa.txt*;
13. Saia da conta e logue como *joao* novamente;
14. Volte ao acesso normal para o arquivo *alfa.txt*:
chmod u+rw alfa.txt

11 Trabalhando permissões de grupos

Nota: Um usuário pode pertencer a vários grupos mas somente um é o primário. O grupo primário é associado ao arquivo no momento de sua criação. É possível que o usuário mude seu grupo primário para um grupo ao qual ele já pertence

1. Logar como *joao*;
2. Verificar a quais grupos você pertence e qual é o grupo primário:
id
3. Criar um arquivo:
ls -l
4. Mudar seu grupo primário para *projx*:
newgrp projx

5. Conferir o seu grupo ativo:
id
6. Criar um outro arquivo e conferir o grupo associado:
touch delta.txt
ls -l
7. Colocar direitos de escrita e leitura para o grupo e remover os direitos de outros:
chmod g+rw,o-rw delta.txt
8. Sair de *joao* e logar como *maria*:
exit
login maria
9. Ir ao diretório de *joao*:
cd /home/joao
10. Acessar o arquivo *delta.txt*
cat delta.txt
11. Escrever em *delta.txt*:
pico delta.txt
12. Tente retirar o direito do grupo:
chmod g-rw delta.txt

Nota: <i>maria</i> não é o proprietário! Mesmo pertencendo ao grupo <i>projx</i> , <i>maria</i> não consegue modificar o acesso
--

13. Saia da conta de *maria* e logue como *joao*;
14. Mude o grupo associado ao arquivo *delta.txt*:
chgrp projy delta.txt
15. Saia da conta *joao*, logue como *maria* e tente acessar novamente *delta.txt*;
16. Saia da conta *maria* e logue como *lara* e tente acessar o arquivo. Não esqueça de mudar o grupo ativo de *lara* para *projy*. Confira com *id*.