

Aula de Exercícios/Laboratório 2: Aplicação do sistema **octal** para definir permissões de acesso a arquivos e diretórios no Linux

Professor: Evandro Cantú

evandro.cantu@ifpr.edu.br

1 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.

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`).

2 Alterando as permissões de acesso a arquivos e diretórios

1. Logue em um terminal em modo texto;
2. Verificar qual é o *diretório corrente* usando o comando:

```
pwd
```

3. Criar no diretório corrente um sub-diretório chamado **dir** com o comando:

```
mkdir dir
```

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

```
touch arq
```

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

```
ls -l
```

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

```
...  
-rw-r--r-- 1 aluno aluno          0 2009-09-24 16:20 arq  
drwxr-xr-x 2 aluno aluno       4096 2009-09-24 16:20 dir  
...
```

Na saída acima, o primeiro dos 10 caracteres mostrados na coluna da esquerda identifica o tipo (- arquivo; 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** (cantu) possui permissão de leitura e escrita (r w -) sobre o mesmo, o grupo (professores) 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**?

3 Testando as permissões de acesso a arquivos e diretórios

O comando **chmod** (*change mode*) permite alterar as permissões de acesso a arquivos e diretórios. uma duas formas de utilizar o comando **chmod** é utilizando parâmetros no **formato octal**.

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.

1. Mudar para o diretório de entrada, com o comando *cd*;
2. Criar no diretório pessoal mais dois sub-diretórios, chamados **dir2** e **dir3**;
3. Criar no diretório pessoal mais dois arquivos, chamados **arq2** e **arq3**;
4. Verifique as permissões dos arquivos criados;
5. Mude as permissões de **arq2** para que o dono, o grupo e os outros tenham acesso completo ao arquivo;
6. Mude as permissões de **arq3** para que o somente o dono tenha acesso completo ao arquivo e o grupo e os outros tenham somente acesso para leitura;
7. Mude as permissões do diretório **dir2**, retirando todas as permissões, inclusive as do dono;
8. Tente mudar para o diretório **dir2**. O que acontece?
9. Mude as permissões do diretório **dir2** de forma que possa novamente entrar neste diretório.