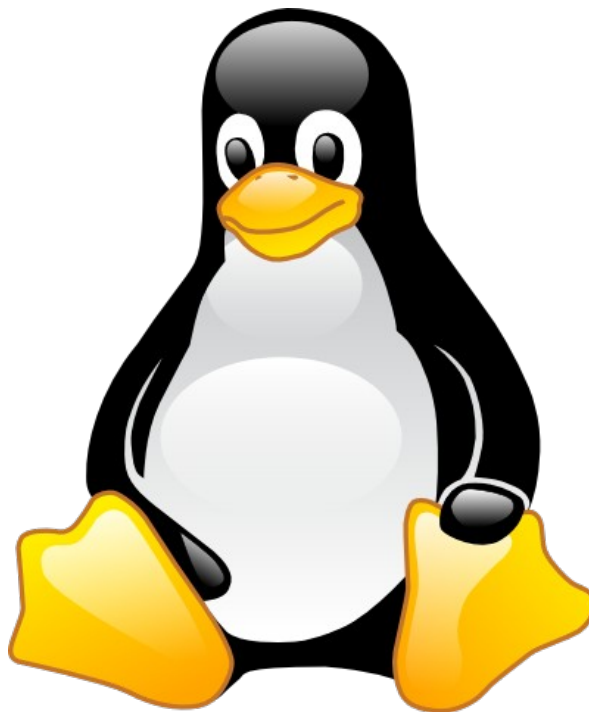


Linux System Administration 455



Aula 8 - 455



Aula 8 - 455

O outro modo é o Modo octal, que é o mais pedido na LPI:

- 1 - execução (x)
- 2 - gravação (w)
- 4 - leitura (r)

Aula 8 - 455

Qual seria essa permissão???

```
#chmod 764 arquivo
```

Resposta:

```
-rwxrW-r--
```

Ou seja, esses comandos fazem a mesma coisa:

```
#chmod 764 arquivo
```

```
#chmod u=rwx,g=rw,o=r arquivo
```

Aula 8 - 455

Além de ser uma variável, UMASK também é um comando:

```
# umask
```

Ela é armazenada nos arquivos profiles (perfis), ou seja, no `/etc/profile` e no `/home/"usuários"/.bashrc` que fica no home de cada usuário do sistema.

Aula 8 - 455

Seu valor padrão é 0022, todo usuário tem seu UMASK:

```
$ cat /home/leo/.bashrc | grep umask  
umask 0022
```

```
# cat /root/.bashrc | grep umask  
umask 0022
```

Se você não encontrar a umask no arquivo, acrescente-a no final do arquivo.

Exemplo:
umask 0022

Aula 8 - 455

Vejo a permissão que eu quero e tiro do valor da permissão total.

Permissão total é: 777

```
777
-755
-----
022
```

Então, temos permissão total ($rw\text{x} = 4+2+1 = 7$)

Aula 8 - 455

A regra é simples:

Sempre pego o valor total de permissão que posso ter, que é 777.

Depois pego o valor de umask que tenho, o padrão é 022.

Pego o valor total e subtraio dele o valor da umask, com isso chego no valor da permissão padrão para diretórios.

Aula 8 - 455

Quando falo permissão total, estou falando de rwx...

Em diretório estou falando:

- r - Posso listar o conteúdo do mesmo
- w - Posso criar arquivos dentro do mesmo
- x - Posso entrar nele para criar os arquivos ou listar...

Em arquivo estou falando:

- r - Posso ler o conteúdo desse arquivo
- w - Posso alterar o conteúdo desse arquivo
- x - Posso executar esse arquivo...

Aula 8 - 455

Baseado nisso, o sistema sempre irá tirar a permissão de execução dos arquivos criados.

Então repetindo a conta:

777 - Permissão total

022 - UMASK

755 - Permissão equivalente ao valor de umask 022 para diretório

-111 - Tirando o x que vale 1 das 3 pessoas

644 - Permissão equivalente ao valor de umask 022 para arquivos...

Aula 8 - 455

Vamos verificar se estou falando a verdade:

Em diretório = 755:

```
#cd /tmp
```

```
#mkdir teste
```

```
#ls -l
```

```
drwxr-xr-x 1 root root teste
```

Em arquivo = 644:

```
#touch arquivo
```

```
# ls -l
```

```
-rw-r--r-- 1 root root arquivo
```

Aula 8 - 455

Quem responde essas?

Baseado na permissão 777 para um diretório, qual seria o valor de UMASK para essa permissão?

Aula 8 - 455

Então vamos a conta:

777 - Permissão total

-777 - Permissão que eu quero

=====

000 - Valor de Umask

Aula 8 - 455

Sabendo que o valor de UMASK é 007 qual seria a permissão para um arquivo simples?

Aula 8 - 455

Vamos a conta:

777 - Permissão total

-007 - Valor de Umask

=====

770 - Permissão equivalente ao valor de umask 007 para diretórios

-111 - Tirando o x (1) pois ele pediu arquivo simples

=====

660 - Permissão para um arquivo simples!

Resposta = 660

Aula 8 - 455

Então uso o comando chown:

```
# cd /tmp
```

```
# touch arquivo_publico.txt
```

```
# ls -l arquivo_publico.txt
```

```
-rw-r--r-- 1 root root 0 2004-11-16 14:14 arquivo_publico.txt
```

```
# chown root.selecaobrasileira arquivo_publico.txt
```

ou

```
# chgrp selecaobrasileira arquivo_publico.txt
```

Ficando assim:

```
# ls -l
```

```
-rw-r--r-- 1 root selecaobrasileira 0 2004-11-16 14:14 arquivo_publico.txt
```

Onde a sintaxe do comando sempre será:

chown dono.grupo nome(arquivo ou diretório)

Aula 8 - 455

Vamos fazer em terminais em modo texto, isto é, 4 shells (TTY1, TTY2, TTY3 e TTY4).

Para acessar o terminal modo texto 1 (tty1) faça: CTRL + ALT + F1

Para acessar o terminal modo texto 2 (tty2) faça: CTRL + ALT + F2

Para acessar o terminal modo texto 3 (tty3) faça: CTRL + ALT + F3

Para acessar o terminal modo texto 4 (tty4) faça: CTRL + ALT + F4

Para voltar ao modo gráfico: CTRL + ALT + F7

Aula 8 - 455

No primeiro terminal estaremos como root.

Lembrando que vamos usar 4 terminais:

- 1 - root
- 2 - Usuário debian
- 3 - Usuário gnu
- 4 - Usuário tux

Os usuários debian, gnu e tux serão criados ainda.

Aula 8 - 455

Então no primeiro terminal como root, vamos criar um diretório público:

```
#cd /mnt
```

```
#mkdir documentos
```

Dando um `ls -ld` nesse diretório veremos que o dono do mesmo é root e seu grupo tb é root:

```
# ls -ld documentos/
```

```
drwxr-xr-x  2 root root 4096 Nov 16 14:42 documentos/
```

Aula 8 - 455

Criando o grupo diretoria para "setar" no diretório documentos:

```
# groupadd diretoria
```

Lembrando que é apenas um exemplo! Pode ser qualquer nome de grupo!

Verificando se o mesmo foi criado:

```
# cat /etc/group | grep diretoria
```

```
diretoria:x:1001:
```

Vimos que ele foi criado e que não tem nenhum usuário que pertence a esse grupo.

Ou seja, a linha dele está vazia:

```
diretoria:x:1001: "Não tem nenhum usuário aqui!!!"
```

Aula 8 - 455

Agora, vamos no diretório /mnt/documentos para definir que o grupo desse diretório será diretoria e não mais root:

```
# chown root.diretoria documentos/
```

```
# ls -ld documentos/
```

```
drwxr-xr-x    2  root  diretoria  4096  2004-11-16  15:28  
documentos/
```

Aula 8 - 455

Resumindo o que fizemos até aqui:

Criamos o diretório /mnt/documentos

```
#mkdir /mnt/documentos
```

Adicionamos um grupo no sistema que irá ser público.

```
#groupadd diretoria
```

E setamos que o grupo do diretório documentos será diretoria.

```
#chown root.diretoria documentos/
```

Aula 8 - 455

Agora nosso próximo passo é criar os usuários que irão acessar esse diretório...

Que no nosso caso será o usuário debian e o gnu!
O usuário tux será o que **NÃO TERÁ ACESSO!!!**

Ou seja, o tux vai ser criado mas não faz parte do grupo diretoria!!!

Aula 8 - 455

Adicionando o Debian:

```
# adduser debian
```

E o usuário GNU agora:

```
# adduser gnu
```

E o usuário TUX agora:

```
# adduser tux
```


Aula 8 - 455

Verificando se os mesmos existem no passwd:

```
# cat /etc/passwd | grep debian
```

```
debian:x:1002:1002:Usuário
```

```
Exemplo Aula,x,x,x,x:/home/debian:/bin/bash
```

```
# cat /etc/passwd | grep gnu
```

```
gnu:x:1003:1003:Usuário
```

```
Exemplo Aula,x,x,x,x:/home/gnu:/bin/bash
```

```
# cat /etc/passwd | grep tux
```

```
tux:x:1006:1006:Usuário
```

```
Exemplo Aula,x,x,x,x:/home/tux:/bin/bash
```

Aula 8 - 455

Mas os mesmos estão com seus respectivos grupos:

```
# groups debian  
debian : debian
```

```
# groups gnu  
gnu : gnu
```

```
# groups tux  
tux : tux
```

Aula 8 - 455

Na prática, vamos acrescentar o grupo diretoria para esses usuários:

Adicionando usuário debian ao grupo diretoria

```
# gpasswd -a debian diretoria
```

Adicionando usuário gnu ao grupo diretoria

```
# gpasswd -a gnu diretoria
```

Agora vamos verificar novamente os grupos desses usuários:

```
# groups gnu
```

```
gnu : gnu diretoria
```

Os dois agora fazem parte do grupo diretoria:

```
# cat /etc/group | grep diretoria
```

```
diretoria:x:1001:debian,gnu
```

Aula 8 - 455

Façam o teste tentando criar um arquivo dentro do diretório documentos com algum usuário (debian ou gnu)

Lembrando que para criar arquivos vazios é o comando abaixo (tentando criar como usuário debian):

```
$ touch /mnt/documentos/teste.txt
```

Não é possível! Permissão negada! :(

Tentando com o usuário gnu:

```
$ touch /mnt/documentos/teste.txt
```

```
touch: cannot touch `/mnt/documentos/teste.txt': Permissão negada
```

Aula 8 - 455

Resumindo, faremos da seguinte maneira:

```
# chmod 770 documentos/
```

```
# ls -ld documentos/
```

```
drwxrwx--- 2 root diretoria 4096 2004-11-16 15:28 documentos/
```

Podemos também usar as letras:

```
# chmod u=rwx,g=rwx,o=rwx documentos/
```

```
# ls -ld documentos/
```

```
drwxrwx--- 2 root diretoria 4096 2004-11-16 15:28 documentos/
```

Aula 8 - 455

Tentem acesso esse diretório público com o usuário tux, vá no terminal do tux.

```
$ cd /mnt/documentos/
```

```
bash: cd: /mnt/documentos/: Permissão negada
```

Pronto!!! Agora tenho os 4 terminais que eu havia falado no início:

1 - root

2 - debian

3 - gnu

4 - tux

Aula 8 - 455

```
$ cd /mnt/documentos/  
$ touch arq_debian.txt  
$ ls  
arq_debian.txt
```

Aula 8 - 455

Agora vamos testar isso com o gnu:

```
$ cd /mnt/documentos/
```

```
$ touch arq_gnu
```

```
$ ls
```

```
arq_debian.txt  arq_gnu
```


Aula 8 - 455

```
# cd /mnt/documentos/
```

```
# ls -l
```

```
-rw-r--r-- 1 debian debian 2004-11-16 16:47 arq_debian.txt
```

```
-rw-r--r-- 1 gnu gnu 2004-11-16 16:57 arq_gnu
```

Aula 8 - 455

```
# cd /mnt/  
# ls -ld documentos/  
drwxrwx--- 2 root diretoria 4096 2004-11-16 16:57 documentos/  
  
# chmod g+s documentos/  
# ls -ld documentos/  
drwxrws--- 2 root diretoria 4096 2004-11-16 16:57 documentos/
```

Aula 8 - 455

Então ficaria:

```
#chmod 2770 /mnt/documentos
```

Sendo que o 2 na frente diz que é Sgid Bit, e 770 o que já tínhamos setado antes.

Aula 8 - 455

Criando arquivo com o debian:

```
$ cd /mnt/documentos/  
$ touch arq2_debian.txt
```

E criando com o gnu:

```
$ cd /mnt/documentos/  
$ touch arq2_gnu.txt
```

Como root para ver se meu trabalho está bem feito.

```
# cd /mnt/documentos/  
# ls -l arq2*  
-rw-r--r--  1 debian diretoria 0 2004-11-16 17:16 arq2_debian.txt  
-rw-r--r--  1 gnu    diretoria 0 2004-11-16 17:16 arq2_gnu.txt
```

Aula 8 - 455

O valor de umask padrão do sistema e de todos usuários é 022....

O que equivale a permissão:

Umask= 0022

Permissão em Diretório = 755

Permissão em Arquivo = 644

Como cheguei a esses valores:

777 - Permissão Total

-022 - Valor de Umask

=====

755 - Permissão para Diretórios

-111 - Tirando o x de execução de arquivos

=====

644 - Permissão para arquivos

Lembrando que o UMASK é o valor da permissão padrão para novos arquivos e diretórios criados por qualquer usuário do sistema.

Aula 8 - 455

O Red Hat e o Fedora acabam já setando a UMASK certa para usuários mortais!

Umask = 002

Permissão em Diretório = 775

Permissão em Arquivo = 664

Como cheguei a esses valores:

777 - Permissão Total

-002 - Valor de Umask

=====

775 - Permissão para Diretórios

-111 - Tirando o x de execução de arquivos

=====

664 - Permissão para arquivos

Aula 8 - 455

Então eu tenho que definir a UMASK 007 para os usuários....

Umask=007 (Lembre-se que os outros usuários que não pertencem ao grupo não terão nem permissão de leitura, isto é, o diretório é público só para quem está no grupo diretoria)

Permissão em Diretório = 770

Permissão em Arquivo = 660

Como cheguei a esses valores:

777 - Permissão Total

- 007 - Valor de Umask

=====

770 - Permissão para Diretórios

- 111 - Tirando o x de execução de arquivos

=====

660 - Permissão para arquivos...

Aula 8 - 455

O valor de umask está setado no arquivo .bashrc do home dos usuários.

```
#cat /home/debian/.bashrc | grep umask
```

O umask no Fedora fica no /etc/bashrc

Aula 8 - 455

```
# vi /home/debian/.bashrc  
umask 007
```

No arquivo a linha do umask vem comentada, então retire a cerquilha # antes da linha.

Para que a alteração entre em vigor faça:
\$ source /home/debian/.bashrc

Aula 8 - 455

Criando um terceiro arquivo como Gnu:

```
$ cd /mnt/documentos/  
$ touch arq3_gnu.txt
```

Como debian agora:

```
$ cd /mnt/documentos/  
$ touch arq3_debian.txt
```

Verificando com o root:

```
# cd /mnt/documentos/  
# ls -ls arq3*
```

```
-rw-rw---- 1 debian diretoria Nov 19 11:16 arq3_debian.txt  
-rw-rw---- 1 gnu    diretoria Nov 19 11:18 arq3_gnu.txt
```

Aula 8 - 455

Com o gnu:

```
$ touch arq4_gnu.txt
```

Com o debian:

```
$ touch arq4_debian.txt
```

Como root:

```
# ls -l
```

```
-rw-rw---- 1 debian  diretoria Nov 19 11:16 arq3_debian.txt
-rw-rw---- 1 gnu     diretoria Nov 19 11:18 arq3_gnu.txt
-rw-rw---- 1 debian  diretoria Nov 19 11:19 arq4_debian.txt
-rw-rw---- 1 gnu     diretoria Nov 19 11:19 arq4_gnu.txt
```

Aula 8 - 455

Então o usuário debian muito amigo...

```
debian@matrix:documentos$ ls
arq3_debian.txt arq3_gnu.txt arq4_debian.txt arq4_gnu.txt
debian@matrix:documentos$ rm arq3_gnu.txt
rm: remove regular empty file `arq3_gnu.txt'? y
```

E o Gnu para se vingar...

```
gnu@matrix:documentos$ ls
arq3_debian.txt arq4_debian.txt arq4_gnu.txt
debian@matrix:documentos$ rm arq3_debian.txt
rm: remove regular empty file `arq3_debian.txt'? y
```

Aula 8 - 455

Antes tínhamos assim:

```
# ls -ld /mnt/documentos/  
drwxrws--- 2 root diretoria 4096 Nov 19 11:34 /mnt/documentos/
```

Agora vamos acrescentar a stick bit que é a letra t:

```
# chmod o+t /mnt/documentos/  
# ls -ld /mnt/documentos/  
drwxrws--T 2 root diretoria 4096 Nov 19 11:34 /mnt/documentos/
```

Aula 8 - 455

```
# chmod 3770 /mnt/documentos
```

Sendo que $3 = 2$ (Sgidbit) + 1 (Stickbit)

E 770 as permissões normais que já tínhamos setado!

Aula 8 - 455

O debian tentando apagar o do gnu:

```
debian@matrix:documentos$ rm arq4_gnu.txt  
rm: remove regular empty file `arq4_gnu.txt'? y  
rm: cannot remove `arq4_gnu.txt': Operation not permitted
```

E gnu tentando apagar o do debian:

```
gnu@matrix:documentos$ rm arq4_debian.txt  
rm: remove regular empty file `arq4_debian.txt'? y  
rm: cannot remove `arq4_debian.txt': Operation not permitted
```

Aula 8 - 455

Exemplo prático:

Vou criar um grupo no qual os usuários que poderão reiniciar ou desligar o sistema estarão:
`# groupadd shutdown`

No meu caso, o usuário comum leo estará nesse grupo:
`# gpasswd -a leo shutdown`

Precisamos mudar o grupo do arquivo executável /sbin/shutdown:
`# chown root.shutdown /sbin/shutdown`

Aplicando a permissão especial SUID em modo octal, apenas o root terá poder de escrita nesse arquivo:
`# chmod 4750 /sbin/shutdown`

Agora, vamos criar um link do arquivo para o /bin. Lembrem-se que o usuário comum só executa arquivos que estão em /bin, por isso a necessidade do link. Criando o link:
`# ln -s /sbin/shutdown /bin/shutdown`