1 Permisos e seguridade en Linux

En linux, os permisos son algo moi importante. Todos os arquivos teñen dono, pertencen a un grupo e contan con proteccións enfocadas a lectura, escritura e execución.

- Permisos Linux en Manuais

1.1 Sumario

1.2 Explicación dos permisos en linux

Podemos utilizar a orden [ls] para ver os permisos actuais dos arquivos:

Podemos ver un exemplo executando **ls -l** nun directorio onde existe unha carpeta e un arquivo:

```
$ ls -l
drwxr-xr-x  2  usuario  usuario  4096  2011-01-31 12:00 datos
-rw-r--r--  1  usuario  usuario    28  2011-01-31 11:00 notas.txt
```

Na saída proporcionada pola orden **ls -l** podemos ver unha primeira columna (neste caso, na primeira fila, vemos que está formada por **drwxr-xr-x**) que se pode dividir en dúas partes:

- O primeiro carácter nos indica o tipo de arquivo ("-" para un arquivo como tal, "d" para un directorio, "b" para un dispositivo e "l" para un enlace)
- Os restantes 9 (**rwxr-xr-x** no noso exemplo) son os que nos falan dos permisos:

Posición	Para
Segunda a Cuarta (Ex. rwx)	Usuario dono do arquivo

Posición	Para
Quinta a Sétima (Ex. r-x)	Grupo dono do arquivo
Oitava a Décima (Ex. r-x)	Demais usuarios do equipo

- Na terceira columna podemos ver quen é o usuario dono do arquivo ou carpeta.
- Na cuarta columna podemos ver quen é o grupo dono do arquivo ou carpeta.
- Os permisos sería así:
    - **r** significa lectura.
    - **w** significa escritura.
    - **x** significa execución.
    - **-** significa que non ten o permiso que iría nesa posición.

1.3 chmod

chmod (*change mode*, cambiar modo en inglés) é unha chamada ao sistema e o seu comando asociado no sistema operativo Linux que permite cambiar os permisos de acceso dun arquivo ou o directorio.

Exemplos:

```
# Acceso total ao dono e lectura e escritura aos demais:
$ chmod 766 file.txt
# Acceso total ao dono e ao grupo e elimina todos os permisos aos demais usuarios:
$ chmod 770 file.txt
# Lectura e escritura ao dono, escritura e execución ao grupo e lectura e execución ao resto:
$ chmod 635 file.txt
# Agrega permisos de lectura a todos os usuarios
$ chmod +r arch.txt
# Agrega permisos de escritura ao dono
$ chmod u+w arch.txt
# Elimina o permiso de execución a todos os usuarios
$ chmod ?x arch.txt
# Establece os permisos de lectura e escritura ao dono e elimina todos os permisos aos demais usuarios
$ chmod u=rw, go= arch.txt
```

1.4 chown

O comando [chown] permite cambiar o propietario do ficheiro en sistemas UNIX. Exemplos:

```
# Poñer a xan como propietario de arquivo1 e arquivo2
$ chown xan arquivo1 arquivo2

# Poñer a maria como propietaria do directorio /var/datos e de todos
# os seus subdirectorios e arquivos
$ chown -R maria /var/datos
```

1.5 chgrp

O comando [chgrp] permite cambiar o grupo propietario dun ficheiro.

```
# Poñer a proxecto1 como propietario de arquivo1 e arquivo2
$ chgrp proxecto1 arquivo1 arquivo2

# Poñer a maria como usuaria propietaria e poñer a proxecto1 como grupo propietario
# do directorio /var/datos e de todos os seus subdirectorios e arquivos
$ chown -R maria:proxecto1 /var/datos
```

1.6 umask

Os permisos que por defecto son asignados aos nosos arquivos ao seren creados veñen pola orden umask.
O valor a asignar como ***umask*** atópase configurado no arquivo /etc/profile. O que indica é o número a ser "restado" do total de permisos 777 (para directorios) ou 666 (para ficheiors). Por exemplo, unha umask de 022 implica que os directorios que se creen teñan permisos por

defecto de 755 e os arquivos que se creen teñan permisos por defecto 644.
Cada usuario no ficheiro /home/usuario/.profile (este último valor prevalece sobre o xeral).

Si desde consola tecleamos la orden umask , vamos a obtener una salida de 4 dígitos en donde el primero de ellos es siempre "0" (hasta el momento este dígito no se utiliza, aunque se planea relacionarlo con los bits de seguridad SUID o GUID en un futuro).

1.7 chkconfig

O comando [chkconfig] tamén pode usarse para activar ou desactivar servicios. Se utiliza o comando *chkconfig --list*, poderá ver unha lista dos servicios do sistema e se están arrancados (on) ou detidos (off) nos niveis de execución 0-6.

1.8  Niveis de Execución en Linux

Nos sistemas GNU/Linux especifícanse ata 7 niveis execución (tamén coñecidos como *runlevels*). Na maioría das distribucións, a especificación dos niveis de execución son como sigue:

- **Nivel de execución 0: *Halt*.** Este nivel de execución se encarga de deter todos os procesos activos no sistema, enviando á placa nai unha interrupción para o completo apagado do equipo.
- **Nivel de execución 1: *Single*.** Nivel de execución monousuario, sen acceso a servicios de rede. Este nivel é regularmente utilizado en tarefas de mantemento do sistema, e o usuario que executa é *root*.
- **Nivel de execución 2:** Ao igual que o nivel de execución monousuario, pero con funcións de rede e compartición de datos mediante nfs.
- **Nivel de execución 3:** Sistema multiusuario, con capacidades plenas de rede, sen entorno gráfico. Este nivel de execución é o recomendado para sistemas de servidor, xa que evita a carga innecesaria de aplicacións consumidoras de recursos.
- **Nivel de execución 4:** Nivel especificado, pero non se utiliza.
- **Nivel de execución 5:** Ao igual que o nivel de execución 3, pero con capacidades gráficas. Ideal para entornos de escritorio.
- **Nivel de execución 6: *Reboot*.** Este nivel de execución se encarga de deter todos os procesos activos no sistema, enviando á placa nai unha interrupción para o reinicio do equipo.

Os guións de execución de mandatos específicos de cada nivel de execución atópanse almacenados nas carpetas /**etc**/**rc{0..6}.d**/, sendo únicamente enlaces simbólicos cara os guións específicos das distintas aplicacións.
É dicir, cando eliximos ao noso sistema en certo nivel de execución (por exemplo, 3), os guións *init* que se executarán son os que se atopen na carpeta /**etc**/**rc3.d**/. O nome do guión de execución componse básicamente de tres elementos: se o nome do guión comeza cunha letra **"K"**, se está especificando que para ese nivel de execución específico o guión dará de baixa (matar, ou *kill*) os procesos iniciados por dito guión. Polo contrario, se o nome do guión comeza pola letra **"S"** dito guión iniciará (*start*) dita tarefa. Ademais, estes guións se executan en orden alfabético, polo que hai que determinar cal é o orden de execución mediante a inclusión dun número de dous díxitos despois da primeira letra do nome do guión. O terceiro elemento do nome do guión especifica o nome do servicio ou tarefa coa que se vai a traballar.

Exemplos:

- Agregar o servicio web(***httpd***) pero non configurar o arranque:

```
$ chkconfig --add httpd
```

- Agregar o servicio web (***httpd***) e facer que arrnaque nos niveis 2, 3, 4 e 5:

```
# Agregamos o servizo
$ chkconfig --add httpd

# Facemos que arranque nos niveis 2, 3, 4 e 5
$ chkconfig --level 2345 httpd on
```

- Eliminar o servicio web(***httpd***) do listado de servicios:

```
$ chkconfig --del httpd
```

- Deshabilitar o servicio web(***httpd***) no nivel de execución 5:

```
$ chkconfig --level 5 httpd off
```

- Listar os servizos e o seu estado

```
$ chkconfig --list
```