Comandos de gestión de usuarios, grupos y permisos

Linux

Ángel González M.

Gestión de usuarios y grupos

Ya sabemos que Linux es un sistema multiusuario y por lo tanto distingue diferentes usuarios. Cada usuario recibe una cuenta que incluirá toda la información necesaria (nombre de usuario, directorio inicial, etc.).

Además de las cuentas dadas a personas, existen cuentas especiales definidas por el sistema que tienen privilegios especiales. La más importante es la cuenta raíz (administrador), con el nombre de usuario root.

Normalmente, los usuarios normales están restringidos, de forma que los permisos de los ficheros en el sistema están preparados para que no puedan borrar o modificar ficheros en directorios compartidos por todos los usuarios.

Estas restricciones desaparecen para root. El usuario root puede leer, modificar o borrar cualquier fichero en el sistema, cambiar permisos y pertenencias, etc. Por lo tanto, podemos deducir que la gestión de los usuarios solamente puede realizarla el usuario root.

Ruta donde se guardan los usuarios del sistema

ver el fichero /etc/passwd

>cat /etc/passwd

La información que el sistema mantiene acerca de cada usuario es la siguiente:

• Nombre de usuario: es un identificador único dado a cada usuario del sistema. Es la cadena de caracteres con la que el usuario se identifica al entrar en el sistema. Se pueden utilizar letras, dígitos y los caracteres _ (guión bajo) y . (punto). Ejemplo: simmd.

• User ID o UID: es un número único dado a cada usuario del sistema. Su número debe ser mayor que el del último usuario creado en el sistema.

• Group ID o GID: número identifica el grupo al que pertenece el usuario. El número ha de ser el mismo para todos los usuarios que formen el grupo. Cada usuario puede pertenecer a uno o más grupos definidos por el administrador del sistema. Aunque la importancia real de las relaciones de grupo es la relativa a los permisos de ficheros.

• Clave: el sistema almacena la contraseña del usuario encriptada /etc/shadow. El comando passwd nos permitirá asignar y cambiar las claves de los usuarios.

• Nombre completo: puede ser el nombre real del usuario, su número de teléfono, su dirección, etc. Es decir, guarda información real sobre el sistema.

• Directorio inicial: es el directorio al que accede el usuario al entrar en el sistema. Cada usuario debe tener su propio directorio inicial, normalmente situado bajo /home. En principio será el único directorio en el que el usuario podrá guardar su información personal, programas, etc. Ejemplo: /home/simmd.

• Intérprete de inicio: es el intérprete de comandos que arranca para el usuario cuando se conecta al sistema. Ejemplos: /bin/bash, /bin/tcsh.

Ángel González M.

Ruta donde se guardan los password de los usuarios

ver el fichero /etc/shadow

>cat /etc/shadow

Nota: se guardan de forma encriptada

Añadir un usuario nuevo

> sudo useradd angel

ver el usuario creado (angel) en el panel de control y en los ficheros de linux.

Otra forma de crear usuarios nuevos

>sudo adduser angel2

adduser es más amigable que crea carpetas de inicio de la cuenta y otras configuraciones (por ejemplo, automáticamente carga las estadísticas del sistema y las notificaciones de inicio de sesión), mientras que **useradd** sólo crea el usuario.

chfn: Cambiar el nombre de un usuario

chfn: permite cambiar el nombre completo del usuario:

>chfn –f

Chage: Plazos y límites de uso del usuario

Muestra los plazos y límites fijados al usuario1.
>chage -l usuario1

Colocar un plazo para la contraseña del usuario. En este caso dice que la clave expira el 31 de diciembre de 2020.

>chage -E 2019-12-31 usuario1:

Ángel González M.

Ver quien soy

>whoami

passwd → (password: contraseña)

Cambia las contraseñas de cuentas de usuario.

Los usuarios normales solo pueden cambiar la contraseña de su propia cuenta y el superusuario puede cambiar todas.

La sinapsis del comando sería:

>passwd [opciones] [USUARIO]

Ángel González M.

passwd (continuación)

Opciones:
-a, --all → informa del estado de las contraseñas de todas las cuentas
-d, --delete → borra la contraseña para la cuenta indicada
-e, --expire → fuerza a que la contraseña de la cuenta caduque
-h, --help → muestra este mensaje de ayuda y termina
-k, --keep-tokens → cambia la contraseña sólo si ha caducado
-i, --inactive INACTIVO → establece la contraseña inactiva después de caducar a INACTIVO
-l, --lock → bloquea la contraseña de la cuenta indicada
-n, --mindays DÍAS_MIN → establece el número mínimo de días antes de que se cambie la contraseña a DÍAS_MIN
-q, --quiet → modo silencioso
-r, --repository REP → cambia la contraseña en el repositorio REP
-R, --root CHROOT_DIR → directory to chroot into
-S, --status → informa del estado de la contraseña la cuenta indicada
-u, --unlock → desbloquea la contraseña de la cuenta indicada
-w, --warndays DÍAS_AVISO → establece el aviso de caducidad a DÍAS_AVISO
-x, --maxdays DÍAS_MAX → establece el número máximo de días antes de cambiar la contraseña a DÍAS_MAX
Si se especifica nombre-usuario, se cambiará la contraseña de dicho usuario (para esto se debe ser root), sino, la del usuario que ejecuta el comando. La mecánica de cambio de contraseña tiene 3 pasos:
- Ingresar la contraseña antigua.
- Ingresar la contraseña nueva.
- Repetir la contraseña nueva para confirmar.

Ángel González M.

chequear la sintaxis del fichero passwd

Para chequear la sintaxis correcta el formato de fichero de ‘/etc/passwd’ y la existencia de usuarios.

>pwck

Ver en qué carpeta estoy

>pwd

Volver a mi carpeta base (carpeta personal)

temenos varias alternativas:

>cd $home

>cd ~

>cd /home/juan/

>cd

Ver los usuarios conectados al sistema (que han iniciado sesión)

>users

Ángel González M.

Borrar usuarios (userdel y deluser)

>userdel

>deluser

La diferencia entre ambos es que el funcionamiento de deluser es mas sencillo, lo que hace al ejecutarse es atender a las configuraciones establecidas en el fichero /etc/deluser.conf y otras opciones que se le pasen al ejecutar el comando.

Ángel González M.

userdel: Borrar usuarios

Para borrar un usuario desde el modo consola:

>userdel [-r] usuario

Si utilizamos la opción –r también eliminaremos el directorio home del usuario o directorio inicial.

Una forma de deshabilitar una cuenta de usuario sin tener que borrarla es escribir ! en el campo clave del usuario en el fichero /etc/shadow o /etc/passwd.

Ruta donde se guardan los grupos creados

>cat /etc/group

Fichero de configuración por defecto de nuevos usuarios

> *cat /etc/adduser.conf*

Añadir nuevos grupos

>addgroup x

>groupadd x

groupadd: Añadir nuevos grupos

groupadd: permite añadir un nuevo grupo. Sintaxis:

>groupadd [-g GID] [-f]

addgroup: permite añadir un nuevo grupo. Sintaxis:

-f obliga al sistema a informar si se producen errores (por ejemplo cuando el grupo que queremos crear ya existe).

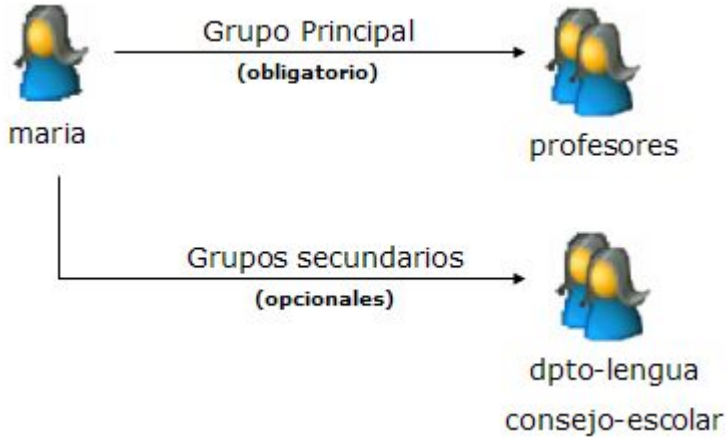
Si no especificamos un GID, el sistema asigna el menor GID que corresponde a este grupo.

Ambos comandos son correctos, pero el estándar POSIX es groupadd

Asignar un usuario a un grupo

para grupo primario

>usermod -g grupo usuario



para grupo secundario

>usermod -G grupo usuario

Todos los usuarios pertenecen al menos a un grupo que es el grupo principal del usuario, también llamado grupo primario del usuario, pero pueden pertenecer a más grupos. En caso de que pertenezcan a más grupos, éstos serán grupos secundarios.

groupdel: borrar un grupo

groupdel: borra el grupo cuyo nombre indiquemos junto a la orden.

>groupdel nombreGrupo

groupmod: Modificar el GID

groupmod: permite modificar el GID y el nombre del grupo. Sintaxis:

>groupmod [-g] [-n]

id: muestra UID y GID del usuario y los grupos a los que pertenece el usuario conectado al sistema. Sintaxis:

#id

#id usuario

Ángel González M.

Visualizando los grupos guardados en el sistema

muestra todos los grupos a los que pertenece el usuario.

>groups

Chequear la sintaxis correcta del fichero group

Chequear la sintaxis correcta y el formato del fichero ‘/etc/group’ y la existencia de grupos.

>grpck

Permisos en los sistemas de archivos

Los sistemas UNIX o compatibles POSIX, incluyendo sistemas basados en Linux y Mac OS X, poseen un sistema simple para el manejo de permisos sobre archivos individuales. POSIX especifica también un sistema de listas de control de acceso (ACLs), pero sólo está implementado por ciertos sistemas de archivos y sistemas operativos.

Las variantes de DOS (incluyendo los productos de Microsoft MS-DOS, Windows 95, Windows 98, y Windows Me) no implementan ningún sistema de permisos. Existe un atributo de "solo lectura" que puede ser asignado o quitado de cualquier archivo por cualquier usuario.

Microsoft Windows NT y sus derivados (incluyendo Windows 2000 y Windows XP), así como VMS y OpenVMS usan listas de control de acceso (ACLs) para administrar un conjunto más complejo y variado de permisos.

Ángel González M.

Permisos sobre archivos

Cada usuario es dueño de su directorio personal y será dueño también de los archivos que incluya en él.

Un usuario en Linux podrá configurar permisos en sus archivos. Por ello, distinguiremos por un lado tres categorías de usuarios, y por otro los tipos de permisos que cada uno de ellos puede tener sobre un archivo y/o directorio.

Categorías de usuarios

- Dueño del archivo (u).
- Grupo dueño (g), formado por todos los usuarios que son miembros de un grupo asociado al archivo.
- Resto de usuarios (o), todos los usuarios que no son ni el dueño ni miembros del grupo dueño.

Tipos de permisos

- Lectura (r de Read, leer): para un archivo permite leer su contenido, para un directorio permite que se muestren los archivos que contiene.
- Escritura (w de Write, escribir): para un archivo permite que se modifique su contenido, para un directorio permite agregar y quitar archivos.
- Ejecución (x de eXecute, ejecutar): para un archivo permite su ejecución, para un directorio permite que el usuario lo recorra (que entre y pase por él) – si no tiene permiso de lectura, aunque pueda entrar no podrá ver el contenido.

Ver permisos de ficheros

>ls -l

el modificador -l nos muestra mucha información, entre ella los permisos de un fichero/directorio en particular

- El primer carácter indica el tipo de archivo: “d” si es directorio, “-“ si es un archivo regular, “l” si es un enlace simbólico.
- Los siguientes nueve caracteres indican los permisos para el dueño, el grupo dueño y otros (rwxrwxrwx); si aparece un guión, indica que el permiso correspondiente no está habilitado.
- El siguiente número indica el número de vínculos.
- Nombre del dueño y nombre del grupo dueño.
- Tamaño en bytes.
- Fecha de la última modificación.
- Nombre del archivo.

Ángel González M.

Permisos en linux

Los permisos de sistemas UNIX se dividen en tres clases, conocidas como usuario, grupo y otros (con frecuencia abreviado UGO, por sus siglas en inglés, User, Group, Others).

De hecho, los permisos en Unix son una forma simplificada de listas de control de acceso (ACLs).

Existen muchas formas para representar los esquemas de permisos Unix:

Ángel González M.

Permisos usando notación simbólica

El primer carácter indica el tipo de archivo:

'-' denota un archivo regular
'd' denota un directorio
'b' denota un archivo especial de bloques
'c' denota un archivo especial de caracteres
'l' denota un enlace simbólico
'p' denota una tuberia
's' denota un socket de dominio
Cada clase de permisos se representa por tres caracteres.

- El primer conjunto de caracteres representa la clase de usuario.
- El segundo conjunto representa la clase de grupo.
- El tercer y último conjunto de tres caracteres representa la clase del resto.

Cada uno de los tres caracteres representa los permisos de lectura, escritura y ejecución respectivamente:
'r' si el bit de lectura está asignado, '-' en caso contrario.
'w' si el bit de escritura está asignado, '-' en caso contrario.
'x' si el bit de ejecución está asignado, '-' en caso contrario.
Existen bit especiales como el 's' 'S' y 't'

Ángel González M.

Bits especiales

s: SUID (setuid) Se utilizan principalmente para permitir a los usuarios del sistema ejecutar binarios con privilegios elevados temporalmente para realizar una tarea específica. Ejemplo whereis passwd ; ls -l /usr/bin/passwd

t: El sticky bit implementa restricciones dentro de un directorio a la hora de borrar contenido dentro de él. Cuando el sticky bit está activado los usuarios podrán acceder al contenido y modificarlo (siempre que tengan permisos) pero nunca van a poder borrar nada. Solo pueden borrar el contenido,el propietario o root. Esta función es aplicable normalmente a los directorios. El sticky bit se aplica al tercer grupo de permisos, al grupo otros.

Ejemplo de notación simbólica

"-rwxr-xr-x" para un archivo regular que tiene todos los permisos asignados para su propietario y solo permisos de lectura y ejecución para el grupo de usuarios del archivo y el resto de los usuarios. Ningún usuario, salvo el propietario, puede modificar los contenidos del archivo.

"crw-rw-r--" para un archivo especial de caracteres que tiene permisos de lectura y escritura para su propietario y grupo de usuarios y solo permiso de lectura para el resto de los usuarios.

"dr-x------" para un directorio que tiene permisos de lectura y ejecución únicamente para su propietario.

Asignar permisos sobre ficheros y directorios: chmod

chmod

ver los modificadores

chmod [a,u,g,o][+,-][r,w,x]

a= all,
u= user,
g= group
o= *other*
+ = poner
- = quitar
r= lectura
w=escritura
x=ejecución

Ejemplo de asignación de permisos

chmod a+r paco Da a todos los usuarios acceso de lectura al archivo paco.
chmod +r paco Igual al anterior. Si no se indica a, u, g u o por defecto se toma a.
chmod og-x paco Quita permisos de ejecución de paco a todos los usuarios excepto al propietario.
chmod u+rwx paco Permite al propietario leer, escribir y ejecutar el archivo paco.
chmod o-rwx paco Quita permisos de lectura, escritura y ejecución del archivo paco a todos los usuarios menos al propietario y a los usuarios del grupo.
chmod og-r paco luis Quita permisos de lectura de los archivos paco y luis a todos los usuarios excepto al propietario.

Otra forma de asignar permisos usando notación octal

Con la notación octal de tres dígitos, cada número representa un componente distinto del conjunto de permisos: clase de usuario, clase de grupo y clase del resto respectivamente.

Cada uno de estos dígitos es la suma de los bits que lo componen (véase también sistema de numeración binario). El peso de cada bit en un dígito es el siguiente:
El bit de lectura suma 4 al total.
El bit de escritura suma 2 al total.
El bit de ejecución suma 1 al total.

Estos valores nunca producen una combinación ambigua: cada suma representa un conjunto específico de permisos.

Cód	Binario	Permisos efectivos
0	0 0 0	- - -
1	0 0 1	- - x
2	0 1 0	- w -
3	0 1 1	- w x
4	1 0 0	r - -
5	1 0 1	r - x
6	1 1 0	r w -
7	1 1 1	r w x

Ángel González M.

Ejemplo asignación permisos usando notación octal

He aquí los ejemplos de la sección Notación simbólica de más arriba en su notación octal:

"-rwxr-xr-x" se representa como 755 en notación octal de tres dígitos.
"-rw-rw-r--" se representa como 664 en notación octal de tres dígitos.
"-r-x------" se representa como 500 en notación octal de tres dígitos.

// Dar todos los permisos al usuario y ninguno ni al grupo ni al resto
>chmod 700 examen.txt

// Dar al usuario y al grupo permisos de lectura y ejecución y ninguno al resto
>chmod 550 examen.txt

// Dar todos los permisos al usuario y lectura y ejecución al grupo y al resto
>chmod 755 /usr/bin/games/tetris

// Dar todos los permisos al usuario y de lectura al resto, sobre todos los archivos
>chmod 744 *

// Cambiar permisos a todos los archivos incluyendo subcarpetas
>chmod -R 744 *

Ángel González M.

Cambiar permisos sobre ficheros y directorios

El valor por defecto en cuanto a permisos de ficheros y directorios es:

- ficheros 644
- directorios 755

Se pueden cambiar estos permisos por defecto con el comando umask

>umask 022 cambia esto

para que este cambio quede aplicado hay que guardarlo en el .bashrc y reiniciar sesión

Ángel González M.

chown: Cambiar el propietario de un fichero

cambiar el propietario

>chown usuario fichero

Este comando sólo lo puede emplear el actual propietario de los mismos. Los nombres de propietario que admite Linux son los nombres de usuario, que están almacenados en el fichero /etc/passwd.

chgrp: Cambiar el grupo al que pertenece un fichero

cambiar el grupo

>chgrp grupo fichero

Los grupos de usuarios están almacenados en el fichero /etc/group.

Ejemplos con chmod y chown

chmod ugo+rwx directory1: colocar permisos de lectura ®, escritura (w) y ejecución(x) al propietario (u), al grupo (g) y a otros (o) sobre el directorio 'directory1'.
chmod go-rwx directory1: quitar permiso de lectura ®, escritura (w) y (x) ejecución al grupo (g) y otros (o) sobre el directorio 'directory1'.
chown user1 file1: cambiar el dueño de un fichero.
chown -R user1 directory1: cambiar el propietario de un directorio y de todos los ficheros y directorios contenidos dentro.
chgrp group1 file1: cambiar grupo de ficheros.
chown user1:group1 file1: cambiar usuario y el grupo propietario de un fichero.
find / -perm -u+s: visualizar todos los ficheros del sistema con SUID configurado.
chmod u+s /bin/file1: colocar el bit SUID en un fichero binario. El usuario que corriendo ese fichero adquiere los mismos privilegios como dueño.
chmod u-s /bin/file1: deshabilitar el bit SUID en un fichero binario.
chmod g+s /home/public: colocar un bit SGID en un directorio –similar al SUID pero por directorio.
chmod g-s /home/public: deshabilitar un bit SGID en un directorio.
chmod o+t /home/public: colocar un bit STIKY en un directorio. Permite el borrado de ficheros solamente a los dueños legítimos.
chmod o-t /home/public: deshabilitar un bit STIKY en un directorio.

Ángel González M.

chattr: Atributos especiales en ficheros

chattr +a file1: permite escribir abriendo un fichero solamente modo append.
chattr +c file1: permite que un fichero sea comprimido / descomprimido automaticamente.
chattr +d file1: asegura que el programa ignore borrar los ficheros durante la copia de seguridad.
chattr +i file1: convierte el fichero en invariable, por lo que no puede ser eliminado, alterado, renombrado, ni enlazado.
chattr +s file1: permite que un fichero sea borrado de forma segura.
chattr +S file1: asegura que un fichero sea modificado, los cambios son escritos en modo synchronous como con sync.
chattr +u file1: te permite recuperar el contenido de un fichero aún si este está cancelado.
lsattr: mostrar atributos especiales.

Ángel González M.

Envío de mensajes a usuarios

Ángel González M.

mesg: Envio de mensajes

mesg : Digamos que es el botón que ‘mutea‘ o activa los mensajes del comando write en una terminal (pts). Suponiendo que 2 usuarios conversan por terminal, el usuario A tiene abierta 2 terminales (pts2 y pts4) y quiere dejar de recibir mensajes por pts4:

$mesg n < /dev/pts/4
Si queremos volver a recibir mensajes por esa terminal utilizamos la opción: y
talk : Conversación bidireccional mediante consola entre 2 usuarios conectados a la misma o diferente máquina.

write: Envío de mensajes

write : Comando que nos permite enviar mensajes a la consola de un determinado usuario, conectado al mismo sistema.

$write usuario < texto

$write usuario
Escribo aquí lo que
quiera que le llegue y luego cierro.
Control-D

Ángel González M.

talk: Envio de mensajes

$talk usuario [ttyn | pts/n]

$talk usuario@maquina
wall : Útil para enviar un mensaje a todos los usuarios conectados a un sistema Linux.

wall: Envio de mensajes

$wall < archivo_texto

$echo “Hola a todos” | wall
[Opciones]:

Si queremos quitar el banner del mensaje (solo root puede hacerlo): -n
Si queremos usar una cuenta a tras, por ejemplo para apagar el sistema: -t tiempo

Ángel González M.