

Comandos en Linux

Manejo de usuarios y grupos

whoami

Es un comando del tipo Unix, proviene de la concatenación de las palabras en inglés ¿Who am I? que significa, ¿Quién soy?.

Muestra el nombre de usuario efectivo del usuario actual cuando se invoca, que se entiende como el nombre del usuario en sesión.

Sintaxis:

- *whoami [opciones]*

id

Se usa para averiguar los nombres de usuarios y grupos y los ID numéricos (UID o ID de grupo) del usuario actual o de cualquier otro usuario en el servidor.

Sintaxis:

- *id [OPTION]... [USER]*

Parámetros:

- g : Muestra solo el ID de grupo efectivo.
- G : Muestra todos los ID de grupo.
- n : Muestra el nombre en lugar del número.
- r : Muestra una identificación real en lugar de números.
- u : Muestra solo el ID de usuario efectivo.
- g: Muestra el GID actual.

Ejemplos:

- *id -nu*
Lista el nombre del usuario.
- *id -u user1*
Muestra el id del usuario "user1".

useradd

Para crear usuarios en el sistema. SE deben pasar por parámetros todas las opciones del usuario.

Los valores por defecto que tomará useradd, se configuran en el archivo /etc/default/useradd.

Sintaxis:

- *useradd [opciones] nombre - usuario.*

Parámetros:

- c: Permite agregar un nombre completo tipo comentario para mayor claridad en su identificación.
- d HOME_DIR: Esta opción nos permite cambiar el directorio por defecto del usuario, que suele ser /home/.
- e FECHA: Permite seleccionar la fecha en la que se deshabilitará la cuenta. Debe introducirse siguiendo el formato AAAA-MM-DD.
- f DÍAS: Nos permite seleccionar el número de días a partir de la fecha de expiración de la contraseña para deshabilitar la cuenta. Con un valor de -1, no se deshabilitará.
- g GRUPO: Permite añadir al usuario a un grupo como grupo principal. El grupo debe existir con anterioridad para poder añadirlo. Podemos introducir el grupo por su nombre o por su identificador de grupo (GID).
- G GRUPO1[,GRUPO2,...[,GRUPOn]]]: Similar a la opción anterior, pero permite introducir varios grupos separados por comas. Los grupos deben existir con anterioridad para poder añadirlo, y hacen referencia a los grupos secundarios.

-m: Crea el directorio personal (home) del usuario si no existe.
-M: No crea el directorio de inicio (home) del usuario.
-n: No crea un grupo privado para el usuario.
-r: La cuenta se convierte en cuenta del sistema, con identificador de usuario (UID) menor a 500 y sin directorio personal.
-p CONTRASEÑA: Establece una contraseña de usuario. Hay que tener en cuenta que al pasar la contraseña como opción, esta irá en claro. La contraseña se puede crear posteriormente con el comando `passwd`.
-s SHELL: Permite modificar la shell de inicio de sesión del usuario, con `useradd` por defecto es `/bin/sh` y con `adduser` por defecto es `/bin/bash`.
-u UID: Permite especificar el identificador de usuario (UID), que deberá ser mayor a 499 y único.
-k SKEL_DIR: The skeleton directory (El directorio esqueleto) contiene archivos y directorios que se copiarán en el directorio de inicio del usuario cuando `useradd` crea el directorio de inicio. Esta opción solo es válida cuando se especifica la opción `-m`. Si la opción `-k` no está configurada, el directorio esqueleto está definido por la variable `SKEL` en el archivo `/etc/default/useradd`. Por defecto el directorio esqueleto se guarda en `/etc/skel`.

Ejemplos:

- `sudo useradd user1`
Crear usuario "user1" no incluye la creación del directorio home.
- `sudo useradd -m user1`
Crear usuario "user1" (incluido directorio home del usuario `/home/user1`).
- `useradd user1 -g comercial`
Crear el usuario "user" y lo agrega al grupo "comercial" siendo este su grupo principal.
- `useradd user1 -g comercial -G ventas, staff`
Agrega el usuario "user1" tal que "comercial" será su grupo principal, y tanto "ventas" como "staff" serán sus grupos secundarios.
- `useradd -c "Julia Funes" julia`
Se crea el usuario "julia" cuyo nombre será "Julia Funes".
- `useradd -m user1 -p cl4v3`
Se crea el usuario "user1" con su directorio principal de nombre igual que el nombre del usuario y de contraseña el valor "cl4v3".

adduser

Nos permite crear cuentas de usuario.

El comando `adduser` es la versión modificada del comando `useradd` y facilita la posibilidad de especificar algunos parámetros, como por ejemplo, del usuario a crear y el directorio de inicio (home) para asociarlo.

Crea un grupo del mismo nombre (nuevousuario), es decir que el identificador de usuario UID y el identificador de grupo GID serán iguales. Además se creará el directorio principal del usuario: `/home/nuevousuario` (por defecto).

Al ejecutar el comando, se le pedirá que completen cierta información sobre el usuario fundamentalmente contraseña, nombre, teléfono, entre otros.

La diferencia entre ambas es que `useradd` es un comando que ejecuta un binario del sistema, mientras que `adduser` es un script en perl que utiliza el binario `useradd`.

La mayor ventaja del comando `adduser` es que crea el directorio home (`/home/usuario/`) del usuario de manera automática, cosa que no hace `useradd` (hay que usar la opción `-m`). Sin embargo, como no es un comando del core de GNU/Linux, es posible que no funcione bien en todas las distribuciones que uses.

Sintaxis:

- `adduser [opciones] usuario grupo`

Parámetros:

--uid ID: Fuerza el nuevo identificador de usuario al número dado. `adduser` fallará si el UID ya está en uso.

Ejemplos:

- `passwd julia`
Cambia la contraseña de julia.
- `sudo adduser user1`
Se crea la cuenta de usuario "user1".

passwd

Cambia la contraseña del usuario.

Sintaxis:

- `passwd [OPCIONES] USUARIO`

Ejemplos:

- `passwd --maximum=90 julia`
La contraseña de julia expira en 90 días.
- `passwd -x 30 -w 5 -i 3 peter24`
Define que el usuario peter24 debe renovar su contraseña cada 30 días y cinco días antes de que se agote el plazo recibe una alerta. Si pasados los 30 días el usuario no ha renovado su contraseña, esta caduca y la cuenta de peter24 se bloquea al cabo de tres días.

usermod

Ofrece la posibilidad de editar cuentas ya creadas.

Sintaxis:

- `usermod [OPCIONES] USUARIO`

Ejemplos:

- `usermod -l pepe pedro`
Cambia el nombre del usuario "pedro" por "pepe".
- `usermod -L user1`
Se bloquea la contraseña del usuario "user1".
- `usermod -d /ruta/al/directorio/peter24 -m peter24`
Crear un nuevo directorio principal (-d DIRECTORIO) y mudar los archivos antiguos (-m).
- `usermod -aG comercial user1`
Añadir usuarios a un grupo (-a) y mantener a los demás miembros (-G). Se agrega el usuario "user1" al grupo "comercial".

deluser

Cierra la cuenta de usuario. Borra todas las entradas de una cuenta de usuario de los archivos de cuentas del sistema. Sólo se puede utilizar con derechos admin.

Este comando es un script de Perl con las funciones del programa de bajo nivel "userdel" en una forma más fácil de usar.

Sintaxis:

- `deluser [OPCIONES] USUARIO`

Ejemplos:

- `deluser user1`
Borra el usuario user1
- `deluser --remove-all-files user1`
Se trata de eliminar todos los archivos del usuario del Sistema. Si se tiene la intención de borrar todos los archivos del usuario en el directorio principal, entonces se utiliza el comando junto con la opción --remove-home.
- `userdel -r user1`
Borra el usuario user1 (incluido directorio home del usuario /home/user1).

groups

Permite conocer los grupos a los que pertenece un usuario en particular.

Sintaxis:

- *groups [OPCIÓN]... [NOMBREUSUARIO]*

Ejemplos:

- *groups user1*
Muestra los grupos a los que pertenece el usuario “user1”.

groupadd

Para crear un nuevo grupo.

Sintaxis:

- *groupadd [OPCIONES] GRUPO*

Ejemplos:

- *groupadd -g 1425 users*
Se crea el grupo “users” con el GID 1425.

groupmod

Modifica las características de un grupo.

Sintaxis:

- *groupmod OPCIONES GRUPO*

Ejemplos:

- *groupmod -n nuevo viejo*
Para cambiar el nombre del grupo “viejo” por “nuevo”

gpasswd

Agrega usuarios a un grupo.

Ejemplos:

- *gpasswd -a user1 comercial*
Agrega el usuario “user1” al grupo “comercial”.
- *gpasswd -d nombredeusuario nombredegrupo*
Para quitar el usuario “user1” del grupo “comercial”.

groupdel

Borra un grupo.

Ejemplos:

- *groupdel comercial*
Borra el grupo “comercial”

newgrp

Cambia el grupo de un usuario. Permite a los usuarios cambiar el ID de su grupo sin tener que cerrar e iniciar sesión de nuevo.

Sintaxis:

- *newgrp [-] [GRUPO]*

Manejo de permisos de archivo

chmod

Es la herramienta para cambiar los permisos de un archivo o directorio. Esta herramienta tiene dos modos de uso, el numérico y el simbólico.

Modo numérico

0 = --- = sin acceso

1 = --x = ejecución

2 = -w- = escritura

3 = -wx = escritura y ejecución

4 = r-- = lectura

5 = r-x = lectura y ejecución

6 = rw- = lectura y escritura

7 = rwx = lectura, escritura y ejecución

Nos permite mediante cuatro octales diferentes permisos. Si se omite uno de ellos se entiende que es cero.

En el caso de directorios la x no representa ejecución, sino que se refiere a acceso al directorio en cuestión.

Modo simbólico

El modo simbólico es una combinación de letras y símbolos, que nos permitirán establecer los permisos para un archivo.

u : es para el usuario

g : para el grupo

o : es para los que no pertenecen al grupo

a : para todos

+ : añade un permiso

- : quita un permiso

= : asigna un permiso

r : lectura

w : escritura

x : ejecución

Sintaxis:

- *chmod [OPCIONES] ARCHIVO / DIRECTORIO*

Parámetros

-R: Se pueden ampliar los derechos de forma recursiva a las subdirectorios y a los archivos contenidos.

Ejemplos:

Modo numérico

- *chmod 400*

Lectura solo por el propietario

- *chmod 040*

Lectura para los pertenecientes al grupo

- *chmod 644*

Este es uno muy típico para archivos. Lectura y escritura por parte del propietario y lectura sólo para el resto aunque pertenezcan al grupo.

- *chmod 755*

Este es muy típico para directorios, porque da todos los permisos al propietario, mientras que al resto solo les da permiso de acceso y lectura.

Modo simbólico

- *chmod u=r,g=,o=*

Lectura solo por el propietario

- *chmod u=,g=r,o=*

Lectura para los pertenecientes al grupo.

- `chmod u=rw,g=r,o=r`
Este es uno muy típico para archivos. Lectura y escritura por parte del propietario y lectura sólo para el resto aunque pertenezcan al grupo.
- `chmod u=rwx,g=rx,o=rx`
Este es muy típico para directorios, porque da todos los permisos al propietario, mientras que al resto solo les da permiso de acceso y lectura.
- `chmod u+x`
Añadir el permiso de ejecución al propietario del archivo, dejando el resto de permisos como estuvieran

chown

Nos permite cambiar la autoría de cualquier archivo o directorio.

Sintaxis:

- `chown [OPTIONS] [USER][:GROUP] file1 [file2 | directory]`

Parámetros:

- R: para aplicar a directorios en forma recursiva.
- c: informa de los cambios con comentarios.

Ejemplos:

- `chown user1 archi1.txt`
Se le cambia la propiedad al archivo "archi1.txt" por el usuario "user1".
- `chown :group1 archi1.txt`
Se le cambia la propiedad al archivo "archi1.txt" por el grupo "group1".
- `chown user1:group1 archi1.txt`
Se le cambia la propiedad al archivo "archi1.txt" por el usuario "user1" y grupo "group1" simultáneamente.
- `chown user1 /testdir`
También se aplica a directorios utilizando el mismo formato que para archivos, es decir, ahora el directorio "testdir" tiene de propietario al usuario "user1".
- `chown -R nuevousuario *`
Cambia el directorio actual con todos sus archivos, y todos los subdirectorios con sus respectivos archivos contenidos.
- `chown -R 0:0 /etc`
También podemos además de usar el nombre de usuario y el nombre de grupo usar el formato numérico mediante el UID (identificador numérico del usuario) y el GID (identificador numérico de grupo). Asignará recursivamente a /etc y a sus subdirectorios y archivos el usuario con UID 0 y el grupo con GID 0 (usuario 0 = root y grupo 0=root).

umask

Se utiliza para establecer permisos por defecto en archivos nuevos. Los permisos de un archivo recién creado se establecen de acuerdo al programa que lo crea y a la máscara.

La máscara de permisos umask para sistemas Linux es habitualmente de 022, lo que elimina de la posibilidad de que otros usuarios y otros miembros del grupo primario del usuario eliminen ficheros del usuario.

Para determinar qué permisos se otorgan por defecto, se restara al número 6 de cada uno de los dígitos de la máscara para saber qué permisos se dan a los archivos normales y el 7 para los directorios.

Sintaxis:

- `umask máscara`

Ejemplos:

- `umask 022`
Corresponde 644 a los ficheros y 755 para los directorios.

getent

Obtiene las entradas de la base de datos administrativa (puede ser: passwd, group, hosts, services, protocols, o networks).

Sintaxis:

- *getent database [key ...]*

Ejemplos:

- *getent passwd*
Lista de los usuarios creados en Linux.
- *getent passwd user1*
Los detalles del usuario “user1”.
- *getent group*
Lista de todos los grupos del sistema Linux.