

Propósitos de los grupos en Linux:

1. Control de acceso: Permiten asignar permisos a varios usuarios de manera conjunta.
 2. Facilitar la administración: Es más fácil gestionar permisos para un grupo que para cada usuario individualmente.
 3. Seguridad: Limitan el acceso a ciertos archivos o comandos solo a miembros autorizados.
-

Tipos de grupos en Linux:

1. Grupo primario (Primary Group):
 - Cada usuario tiene un grupo primario asignado al crearse su cuenta.
 - Cuando un usuario crea un archivo, este pertenece a su grupo primario por defecto.
 - Ejemplo: El usuario `juan` puede tener como grupo primario `usuarios`.
2. Grupos secundarios (Secondary Groups):
 - Un usuario puede pertenecer a varios grupos secundarios.
 - Útiles para dar permisos adicionales sin cambiar el grupo primario.
 - Ejemplo: El usuario `juan` puede estar en los grupos `desarrolladores`, `sudo` y `ftp`.
 - Usuario Root también denominado superusuario o administrador del sistema, cuenta con todos los privilegios sobre el sistema, teniendo acceso total al mismo, tanto de administración, actualización y mantenimiento del sistema. Se identifica con su UID 0. Su directorio es `home/root`
 - Usuarios de sistemas: No son usuarios físicos, sino que se generan al instalar el sistema o cualquier programa o servicio. No tienen todos los privilegios, pero sí algunos especiales. no son usuarios que puedan entrar al sistema y no pueden iniciar sesión. Para estos usuarios se reserva el UID desde el 1 al 999, salvo para el `ubody` que se le asigna el último UID posible, el 65.534

3. Usuarios Normales: Son los usuarios que entran al sistema introduciendo nombre de usuario y contraseña.

para entrar a un usuario es con USUARIO Y CONTRASEÑA O SSH

- **/etc/passwd**
 - Cada una de sus líneas de este fichero de texto representa la información de un usuario.
 - La información se organiza en campos separados por el carácter ":".
 - Los campos de cada línea son:
 - **<nombre_usuario>:x:<UID>:<GID>:<información_usuario>:<director io_personal>:<shell_de_inicio>**
 - **profesor:x:1000:1000:profesor,,,:/home/profesor:/bin/bash**
- **/etc/shadow**
 - Almacena las contraseñas encriptadas de cada usuario.
 - Sólo se puede acceder a este fichero como root o con privilegios de administrador (sudo)
 - En versiones antiguas de Linux, la contraseña del usuario se guardaba en el fichero /etc/passwd en el lugar de la "x", pero en versiones más recientes las contraseñas se almacenan en este fichero con diferentes tipos de encriptación:

Algoritmo MD5 Sin encriptación

Algoritmo SHA 256 256 bits

Algoritmo SHA 512 512 bits

- En este fichero se encuentran los grupos de usuarios definidos en el sistema
- Cada línea contiene la siguiente información:
 - **<nombre_grupo>:<contraseña_grupo_x=notiene>:<GID>:<lista_usuarios_separados_por_comas>**
- Existen otros ficheros importantes en la gestión de usuarios y grupos, como son:

Fichero	Función
/etc/gshadow	Almacena las contraseñas de los grupos y sólo el root puede visualizarlo.
/etc/default/useradd	Contiene valores por defecto a la hora de añadir un usuario con el comando <i>useradd</i>
/etc/adduser.conf	Guarda los valores por defecto que se le dan a un nuevo usuario cuando se crea con el comando <i>adduser</i>
/etc/deluser.conf	Guarda los valores por defecto cuando se elimina un usuario con el comando <i>deluser</i>
/etc/shells	Contiene una lista de los shells válidos

Comandos Útiles:

SUDO: Permite ejecutar comandos como si fuéramos el superusuario o ROOT, para utilizar es necesario que el usuario deba estar configurado con privilegios root o administrador, para eso debe pertenecer al grupo "SUDO" o configurarlo en el fichero /etc/sudoers.

Comando SU: Cambia de usuario o nos permite ser el superusuario. Si no indicamos ningún usuario, nos cambiaremos al super usuario root.

Sintaxis:

Comando	Función
<code>su <nombre_usuario></code>	Cambia de usuario a <nombre_usuario> pero no carga su configuración personal
<code>su - <nombre_usuario></code>	Cambia de usuario a <nombre_usuario> pero sí carga su configuración personal
<code>su</code>	Cambia al superusuario o root, pero no carga la configuración de root
<code>su -</code>	Cambia al superusuario o root, pero sí carga la configuración de root

- **Comando `exit` o (Ctrl + D)**
 - Sale o termina la sesión del usuario con el que estamos logueados en el sistema.

```
Añadiendo el nuevo grupo 'antonio' (1001) ...
Añadiendo el nuevo usuario 'antonio' (1001) con grupo 'antonio' ...
Creando el directorio personal '/home/antonio' ...
Copiando los ficheros desde '/etc/skel' ...
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña tiene menos de 8 caracteres
Vuelva a escribir la nueva contraseña:
Las contraseñas no coinciden.
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para antonio
Introduzca el nuevo valor, o presione INTRO para el predeterminado
Nombre completo []: Antonio García
Número de habitación []:
Teléfono del trabajo []:
Teléfono de casa []:
Otro []:
chfn: el nombre contiene caracteres ilegales (no ASCII): «Antonio García»
¿Es correcta la información? [S/n] S
usuario@usuario-VirtualBox:~$ grep "antonio" /etc/passwd
antonio:x:1001:1001:Antonio García,,,:/home/antonio:/bin/bash
usuario@usuario-VirtualBox:~$
```

Comando	Funciones
<code>adduser [opciones] <usuario_nuevo></code>	Crea un nuevo usuario
<code>adduser <usuario_existente> <grupo_existente></code>	Añade un usuario existente a un grupo existente

Opciones	Función	Ejemplo
<code>--ingroup <grupo_existente></code>	Sirve para añadir un nuevo usuario, pero en lugar de crearle un grupo con su mismo nombre lo añade al grupo existente que se le indica	<code>adduser usu3 --ingroup usuario</code>
<code>--home <directorio_personal></code>	Esto hace que el nuevo usuario tenga como carpeta personal la que le indiquemos	<code>adduser usu4 --home /home/otros</code>

- **Comando `deluser`**

- Elimina un usuario del sistema, pero dicho usuario no puede estar conectado en ese momento al sistema.
- Sintaxis:
 - `deluser [opciones] <nombre_usuario>`
 - Opciones:
 - `--remove-home`
 - Borra el fichero personal del usuario
 - `--remove-all-files`
 - Elimina todos los ficheros que pertenecen al usuario

Siempre es mejor utilizar un usuario que no contenga todos los privilegios de administrador para evitar errores, entonces sudo espacio y el comando. (siempre trabajar con usuario con permisos de administrador)

- **Comando `Usermod`**

- Modifica datos de un usuario existente.
- Sintaxis
 - `usermod [opciones] <nombre_usuario>`

Opciones	Función
<code>-d <nombre_directorio></code>	Cambia el directorio personal del usuario
<code>-g <nombre_grupo></code>	Cambia el grupo primario del usuario
<code>-u <número_uid></code>	Cambia el UID del usuario
<code>-l <nuevo_nombre></code>	Cambia el nombre del usuario