

## 1. Conceptos Básicos sobre Usuarios en Linux

En Linux, los usuarios tienen ciertas características y permisos que determinan qué acciones pueden realizar en el sistema. Los tipos de usuarios más comunes son:

- **Usuario root:** El usuario administrador del sistema, con acceso completo a todos los archivos y configuraciones.
- **Usuarios estándar:** Cualquier otro usuario creado en el sistema, con permisos limitados según lo que se les haya asignado.

## 2. Comandos Básicos para Gestionar Usuarios

### 2.1 Ver usuarios del sistema

Para ver todos los usuarios que existen en el sistema, puedes usar:

```
bash
Copiar
cat /etc/passwd
```

Este archivo contiene información sobre cada usuario. Cada línea tiene el siguiente formato:

```
ruby
Copiar
usuario:x:UID:GID:comentarios:/home/usuario:/bin/bash
```

Donde:

- `usuario` es el nombre del usuario.
- `UID` es el identificador único del usuario.
- `GID` es el identificador del grupo principal del usuario.
- `/home/usuario` es el directorio home del usuario.
- `/bin/bash` es el intérprete de comandos (shell).

### 2.2 Crear un usuario

Para crear un nuevo usuario, usamos el comando `useradd` seguido del nombre del usuario:

```
bash
Copiar
sudo useradd nombre_usuario
```

Para asegurarte de que el usuario tenga un directorio home (carpeta personal) y un shell adecuado (bash por ejemplo), puedes agregar la opción `-m` (para crear el home) y `-s` (para especificar el shell):

```
bash
Copiar
sudo useradd -m -s /bin/bash nombre_usuario
```

### 2.3 Establecer una contraseña para el usuario

Para asignar una contraseña a un usuario, se utiliza el comando `passwd`:

```
bash
Copiar
sudo passwd nombre_usuario
```

Esto te pedirá que ingreses y confirmes una nueva contraseña.

### 2.4 Modificar un usuario

Para cambiar la información de un usuario, puedes usar `usermod`. Algunos ejemplos de uso:

- **Cambiar el nombre de usuario:**

```
bash
Copiar
sudo usermod -l nuevo_nombre_usuario antiguo_nombre_usuario
```

- **Cambiar el directorio home de un usuario:**

```
bash
Copiar
sudo usermod -d /nuevo/directorio/home nombre_usuario
```

- **Añadir un usuario a un grupo:**

```
bash
Copiar
sudo usermod -aG nombre_grupo nombre_usuario
```

### 2.5 Eliminar un usuario

Para eliminar un usuario con su directorio home y archivos:

```
bash
Copiar
sudo userdel -r nombre_usuario
```

Sin la opción `-r`, el usuario será eliminado, pero sus archivos en el directorio home permanecerán.

### 2.6 Ver información de un usuario

Para ver información detallada sobre un usuario específico, como su directorio home, grupo y otros detalles, utiliza:

```
bash
Copiar
id nombre_usuario
```

### 3. Grupos en Linux

Los grupos permiten organizar a los usuarios en conjuntos con permisos similares. Un usuario puede pertenecer a uno o más grupos.

#### 3.1 Ver los grupos de un usuario

Puedes ver los grupos a los que pertenece un usuario con el siguiente comando:

```
bash
Copiar
groups nombre_usuario
```

#### 3.2 Crear un grupo

Para crear un nuevo grupo:

```
bash
Copiar
sudo groupadd nombre_grupo
```

#### 3.3 Añadir un usuario a un grupo

Para agregar un usuario a un grupo, puedes usar el comando `usermod`:

```
bash
Copiar
sudo usermod -aG nombre_grupo nombre_usuario
```

#### 3.4 Eliminar un grupo

Para eliminar un grupo (sin eliminar a los usuarios asociados):

```
bash
Copiar
sudo groupdel nombre_grupo
```

### 4. Permisos de Archivos y Directorios

Cada archivo y directorio en Linux tiene permisos asociados para el propietario, el grupo y los demás usuarios.

#### 4.1 Ver los permisos de un archivo

Puedes ver los permisos de los archivos con el comando `ls -l`:

```
bash
Copiar
ls -l archivo
```

El resultado tendrá un formato como:

```
css
Copiar
```

```
-rw-r--r-- 1 usuario grupo 1234 mar 27 12:00 archivo.txt
```

- El primer carácter muestra el tipo de archivo (- para archivos normales, d para directorios).
- Los siguientes 9 caracteres muestran los permisos (lectura *r*, escritura *w*, ejecución *x*):
  - Los primeros 3 caracteres son para el propietario.
  - Los siguientes 3 son para el grupo.
  - Los últimos 3 son para otros usuarios.

#### 4.2 Cambiar permisos de archivos

Para cambiar los permisos de un archivo, usa `chmod`:

```
bash
Copiar
chmod 755 archivo
```

Donde:

- 7 es para el propietario (lectura, escritura, ejecución).
- 5 es para el grupo (lectura y ejecución).
- 5 es para otros usuarios (lectura y ejecución).

También puedes usar la notación simbólica:

```
bash
Copiar
chmod u+x archivo      # Añadir permiso de ejecución al propietario
chmod g-w archivo      # Eliminar permiso de escritura al grupo
chmod o=r archivo      # Establecer permisos solo de lectura para otros
```

#### 4.3 Cambiar el propietario de un archivo

Para cambiar el propietario y/o el grupo de un archivo, usa `chown`:

```
bash
Copiar
sudo chown nuevo_usuario:nuevo_grupo archivo
```

#### 4.4 Cambiar el grupo de un archivo

Para cambiar el grupo de un archivo sin afectar al propietario:

```
bash
Copiar
sudo chown :nuevo_grupo archivo
```

## 5. Permisos de Acceso a la Terminal (sudo)

Los usuarios pueden ser configurados para utilizar comandos con privilegios de administrador mediante el uso del comando `sudo`.

### 5.1 Añadir un usuario al grupo `sudo`

Para permitir que un usuario ejecute comandos como administrador (root), añádelo al grupo `sudo` (o `wheel`, dependiendo de la distribución):

```
bash
Copiar
sudo usermod -aG sudo nombre_usuario
```

### 5.2 Eliminar un usuario del grupo `sudo`

Para eliminar un usuario de este grupo y evitar que pueda usar `sudo`:

```
bash
Copiar
sudo deluser nombre_usuario sudo
```

## 6. Gestionar sesiones de usuario

### 6.1 Ver usuarios conectados al sistema

Para ver qué usuarios están actualmente conectados, usa el comando:

```
bash
Copiar
who
```

### 6.2 Cerrar sesión de un usuario

Para terminar la sesión de un usuario, puedes usar el comando `pkill` seguido del nombre de usuario:

```
bash
Copiar
sudo pkill -KILL -u nombre_usuario
```

## 7. Archivos de Configuración Importantes

- **/etc/passwd**: Información básica sobre los usuarios.
- **/etc/shadow**: Contiene las contraseñas cifradas de los usuarios.
- **/etc/group**: Información sobre los grupos.
- **/etc/sudoers**: Define los usuarios que pueden usar `sudo` y sus permisos.