

Usuarios y Grupos

Usuarios y grupos en Unix

Cuentas de Usuario

Las cuentas de usuario en Unix se dividen en dos partes.

En **nombre de usuario**(Login) y en **contraseña**(password).

Las cuentas son creadas por el administrador de sistema(ROOT). Los usuarios pertenecerán como mínimo a un grupo de usuarios, ya que como mínimo tendrán asignados un grupo primario.

Las cuentas de usuario también ofrecen al usuario una ruta para almacenar sus documentos y su perfil. Esta ruta es **/home/nombre de usuario**. Se le denomina carpeta home del usuario. También se le genera un **shell** que le permite ejecutar aplicaciones.

Con el comando **top** se visualizan todos los procesos en ejecución en el sistema.

Cuando creas un nuevo archivo, el propietario de este es el usuario que lo ha creado y el grupo será el grupo de su mismo nombre.

Una cuenta de usuario permite el acceso a un equipo tanto presencia y tanto por red.

Los sistemas Unix codifican a los usuarios mediante el **uid**, un número que es único de cada usuario. Los usuarios creados parte desde el 1000.

Los menores a 100 son usuarios propios del sistema.

La información de los usuarios se almacena en el archivo **/etc/passwd**. El archivo solo puede ser modificado por el usuario root.

Las contraseñas de los usuarios están encriptadas y se almacenan en el archivo **/etc /shadow**.

El usuario root se representa en el terminal con **#** y los no root con **\$**.

Grupos de usuarios

Los grupos de usuarios en Unix se utilizan para administrar los permisos de los usuarios de una manera más fácil.

Todos los usuarios pertenecen a un grupo de usuario, este se denomina **grupo primario**; Si el usuario pertenece a más grupos, estos se denominan **grupos secundarios**.

Los grupos pueden contener varios usuarios, pero solo podrán contener usuarios.

Estos se denominan mediante **gid**.

La asignación de estos parte desde el 1000 para los grupos creados por el usuario, mientras que los menores de 100 se reservan para grupos especiales del sistema.

Grupos predeterminados

Usuarios y Grupos

Grupos	Descripción
adm	Grupo de administración que permite accesos a archivos de registro y comandos como <i>sudo</i> y <i>su</i>
users	Grupo de usuarios estándar
nobody	Sin privilegios
root	Administración sin restricciones sobre todo el sistema
tty	Aporta privilegios sobre algunos dispositivos, como <i>/dev/tty</i>
lpadmin	Confiere privilegios sobre dispositivos de puerto paralelo

Administración(comandos) de grupos y usuarios

Crear usuarios	useradd [opciones] nombre	-g : grupo principal -d : carpeta home del usuario -m :Crear carpeta home -s :Interprete de comandos
Establecer contraseña	passwd nombre	
Modificación de usuarios	usermod[opciones] nuevo nombre	-g : grupo principal -d : carpeta home del usuario -m :Crear carpeta home -s :Interprete de comandos
Eliminar usuario	userdel -r usuario	
Crear grupos	groupadd nombre-grupo	
Modificacion de grupos	groupmod -g nuevo-gid -n nuevonombre nombregupo	
Añadir usuario a grupo	adduser usuario grupo	
Quitar usuarios de un grupo	deluser usuario grupo	
Modificar propietario	chown[ops]nuevoPropietario fichero	-R :aplicar cambios a contenido de los dirs -h :afecta al enlace simbólico
Modificar el grupo de un archivo	chgrp [ops] nuevo_Grupo	-R :afecta al contenido de los dirs

Usuarios y Grupos

Modificar contraseña

passwd [ops] contraseña nueva

d : deja en blanco la contraseña
e : hace expirar la passwd
i : establece días de inactividad después de que expire la passwd
l : bloquea la cuenta de usuario
u : desbloquea la cuenta
x : establece tiempo para cambiar la passwd
w : antelación para avisar de caducidad de la passwd
-R : Opción recursiva para un directorio

Modificar permisos

chmod[ops] permisos nuevos

Modificación de permisos

Octal

Administrar los permisos mediante números

Las equivalencias serían

- **r**(lectura) vale 4
- **w** (escritura) vale 2
- **x**(ejecución) vale 1

Simbólica

chmod [destinatarios] [tipo de mod] [permiso] archivo

Destinatarios

- **u** : propietario
- **g** : grupo
- **o** : otros
- **a** : todos
-

Tipo de modificación

- **+** : se añaden a los permisos actuales
- **=** : se establecen los nuevos
- **-** : se restan al valor actual

Permisos

- **r** : lectura
- **w** : escritura
- **x** : ejecución
- **s** : set-uid/set-gid
- **t** : sticky-bit

Usuarios y Grupos

Ejemplo:

Por ejemplo, si quisiéramos deshabilitar para el grupo y otros el permiso de lectura:

```
chmod go-r prueba.txt.
```

grep es un comando de línea de comandos utilizado en sistemas operativos basados en Unix y Linux. Su función es buscar patrones de texto dentro de uno o varios archivos.

La sintaxis básica del comando grep es la siguiente:

```
grep [opciones] PATRÓN [ARCHIVO]
```

Donde:

- Opciones: son argumentos que modifican el comportamiento del comando. Algunas opciones comunes son -i (ignora mayúsculas y minúsculas), -r (busca de forma recursiva en subdirectorios) y -n (muestra los números de línea).
- PATRÓN: es el texto que se desea buscar. Puede ser una palabra, una cadena de caracteres o una expresión regular.
- ARCHIVO: es el archivo en el que se desea buscar. Si no se especifica ningún archivo, grep buscará en la entrada estándar.

Algunos ejemplos de uso de grep son:

Buscar una palabra en un archivo:

```
grep palabra archivo.txt
```

Buscar una palabra en varios archivos:

```
grep palabra archivo1.txt archivo2.txt
```

Buscar una palabra en todos los archivos de un directorio (de forma recursiva):

```
grep -r palabra directorio/
```

Buscar una palabra ignorando mayúsculas y minúsculas:

```
grep -i palabra archivo.txt
```

Usuarios y Grupos

find es un comando de línea de comandos que se utiliza en sistemas operativos basados en Unix y Linux para buscar archivos

`find [ruta] -name [patrón]`: Busca archivos y directorios por nombre. ruta es la ubicación donde se desea buscar y patrón es el nombre o patrón de nombres que se desea buscar.

`find [ruta] -type [tipo]`: Busca archivos y directorios por tipo. tipo puede ser f para archivos regulares, d para directorios, l para enlaces simbólicos, entre otros.

`find [ruta] -mtime [días]`: Busca archivos modificados hace días días. ruta es la ubicación donde se desea buscar.

`find [ruta] -size [tamaño]`: Busca archivos por tamaño. tamaño puede ser n para archivos de exactamente n bloques de tamaño, n+c para archivos de más de n bloques de tamaño, -n para archivos de menos de n bloques de tamaño, entre otros.

`find [ruta] -exec [comando] { } \;`: Ejecuta un comando para cada archivo o directorio encontrado. comando es el comando que se desea ejecutar, y { } es el marcador de posición para el archivo o directorio encontrado.

El comando **umask** en Ubuntu es utilizado para establecer los permisos predeterminados de los archivos y directorios creados por un usuario. El valor de umask se resta de los permisos máximos que se establecen en un archivo o directorio. Por lo tanto, cuanto mayor sea el valor de umask, menor serán los permisos predeterminados que se aplicarán a los nuevos archivos y directorios creados.

El valor de umask se representa generalmente en octal (base 8) y se puede establecer utilizando el comando umask seguido del valor de umask deseado. Por ejemplo, si se desea establecer un umask de 022, se puede utilizar el siguiente comando:

```
umask 022
```

Esto establecerá permisos predeterminados para archivos nuevos de 644 (rw-r--r--) y para directorios nuevos de 755 (rwxr-xr-x).

El valor predeterminado de umask en Ubuntu es 002, lo que significa que los nuevos archivos tendrán permisos de lectura y escritura para el propietario y el grupo, y permisos de lectura para otros usuarios, mientras que los nuevos directorios tendrán permisos de lectura, escritura y ejecución para el propietario y el grupo, y permisos de lectura y ejecución para otros usuarios.