# TEMA 12: ADMINISTRACIÓN Y CONFIGURACIÓN EN LINUX.


1. USUARIOS

2. GRUPOS

3. ADMINISTRACIÓN DE USUARIOS Y GRUPOS

   3.1 COMANDOS PARA LA ADMINISTRACIÓN DE USUARIOS Y GRUPOS

4. PERMISOS EN LINUX.

   4.1 COMANDO PARA GESTIONAR LOS PERMISOS

5. COMANDOS PARA CAMBIAR EL PROPIETARIO Y EL GRUPO DE UN FICHERO

6. FICHEROS DE CONFIGURACIÓN DEL SISTEMA

   6.1 Fichero /etc/passwd

   6.2 El fichero /etc/shadow

   6.3 Fichero /etc/group

   6.4 Otros ficheros de configuración

# TEMA 12: ADMINISTRACIÓN Y CONFIGURACIÓN EN LINUX.

## 1. USUARIOS

Ya sabemos que Linux es un sistema multiusuario y por lo tanto distingue diferentes usuarios. Podemos distinguir tres tipos:

1.- Usuarios normales: son los usuarios habituales del sistema que consumen sus recursos. Cada usuario recibe una *cuenta* que incluirá toda la información necesaria (nombre de usuario, directorio inicial, etc.).

2.- Usuarios del sistema: Son incorporados por el propio sistema operativo cuando este se instala. El sistema los utiliza para realizar tareas administrativas. Estos usuarios no pueden hacer login en el sistema.

3.- Usuario root: Es una cuenta especial definida por el sistema con privilegios especiales, tiene control total sobre el sistema. Su directorio personal se encuentra en /root.

Normalmente, el acceso a los recursos por parte de los usuarios normales está muy restringido. Estas restricciones desaparecen para el root. El usuario root o superusuario puede leer, modificar o borrar cualquier fichero en el sistema, cambiar permisos y pertenencias, etc.

El administrador del sistema es, normalmente, un usuario individual responsable de mantener el sistema en ejecución correctamente. La experiencia ha demostrado que un único usuario puede mantener la consistencia del sistema mucho mejor que varios usuarios.

El usuario root tiene su propia contraseña (password), que sólo debería ser conocida por él mismo y posiblemente por un administrador de reserva. Tiene, igualmente, privilegios especiales que incluyen el acceso completo a todos los ficheros y recursos del sistema, determinando quien y en qué condiciones puede acceder a dicho sistema.

Teniendo en cuenta que el usuario root tiene tareas tan importantes como, añadir o suprimir usuarios, instalar y mantener o desinstalar software del sistema, crear ficheros de dispositivos, añadir o suprimir usuarios, etc. El identificador root debe ser utilizado con sumo cuidado. Una mala utilización del root puede acarrear un gran daño a los programas y a los ficheros del sistema.

El shell proporciona un prompt especial, # (almohadilla), para recordarte que has entrado al sistema como superusuario y que, por tanto, tienes privilegios que permiten acceso de lectura, escritura y ejecución a todos los ficheros y directorios.

La información asociada a los usuarios en un sistema Linux se guarda en el fichero **/etc/passwd** y las contraseñas y datos afines en **/etc/shadow**. En el último punto del tema veremos con detalle estos ficheros.

## 2. GRUPOS

Cada usuario pertenece a un grupo primario, además puede pertenecer a uno o más grupos secundarios.

Cada fichero tiene un usuario propietario (inicialmente es el usuario que crea el fichero) y pertenece a un grupo (inicialmente será el grupo primario del usuario que crea el fichero).

Los grupos permiten otorgar los mismos privilegios a un conjunto de usuarios.

La información de los grupos, sus miembros y passwords están en **/etc/group** y **/etc/gshadow** respectivamente. En el último punto del tema veremos con detalle estos ficheros.

## 3. ADMINISTRACIÓN DE USUARIOS Y GRUPOS

El usuario con más privilegios en Linux es aquel cuyo login es root. Este es el único con derechos suficientes para crear o eliminar a otros usuarios, además de acceder a todo el sistema de ficheros sin ninguna restricción.

En Linux además existen grupos de usuarios también administrados por root o por un usuario designado por este. El root será el encargado de crear y eliminar grupos, también será el encargado de asignar y/o dar de baja a los usuarios en los grupos. Como veremos más adelante el root podrá designar a un usuario ordinario como administrador de un grupo, si esto ocurriera, ese usuario podrá también añadir y eliminar usuarios a su grupo.

**Siempre que se añada un usuario** al sistema se creará un grupo con su mismo nombre, llamado **grupo primario**. Durante la creación o posteriormente, se podrá incorporar el usuario a otros grupos secundarios o incluso cambiar su grupo primario.

Al dar de alta un usuario también se puede crear un directorio base para el mismo con el nombre de su login (o con el nombre que se indique durante su creación). Este directorio se coloca por defecto en el directorio /home excepto para root, cuyo directorio base es /root.

### 3.1 COMANDOS PARA LA ADMINISTRACIÓN DE USUARIOS Y GRUPOS

#### Comando useradd

El comando useradd permite añadir nuevos usuarios al sistema, además de establecer la información por defecto de los nuevos usuarios que se añadan.

Sintaxis: useradd [opciones] [login]

Modificadores:

-m crea un directorio en /home con el mismo nombre que el usuario.

-d crea un directorio en /home con otro nombre diferente al del usuario.

-e permite indicar cuando expira la cuenta usando el formato yyyy-mm-dd

-g permite indicar el grupo principal al que pertenece el usuario

-G permite indicar otros grupos a los que pertenece el usuario.

-D nos devuelve los valores por defecto del comando useradd

Ejemplos:

| | |
|---|---|
| **useradd noelia** | **crea el usuario noelia pero no crea /home/noelia** |
| **useradd -m carlota** | **crea el usuario carlota con directorio /home/carlota.** |
| **useradd –d /home/dire director** | **crea usuario director con carpeta de trabajo /home/dire** |
| **useradd –e 2012-05-05 –m pedro** | **crea usuario pedro con carpeta de trabajo /home/pedro y su cuenta expira el 05/05/12.** |
| **useradd –D** | **muestra las opciones por defecto que se aplicarán a los usuarios nuevos.** |

## Comando passwd

El comando passwd permite cambiar el password de un usuario. También puede bloquear, desbloquear y deshabilitar una cuenta. Si se invoca sin argumentos se asume el usuario actual.

Sintaxis: passwd [opciones] [login]

Modificadores:

| | |
|---|---|
| -d | deshabilita la cuenta de usuario eliminando su passwd. |
| -l | bloquea la cuenta del usuario poniendo un signo ! delante de su password en el fichero /etc/shadow. |
| -u | desbloque la cuenta indicada. |
| -n min | nº de días que tiene que pasar para poder cambiar la contraseña. |
| -x max | nº de días que durara la contraseña sin ser modificada. Pasado ese tiempo es obligatorio cambiarla. |
| -w warn | nº días de aviso. |
| -S | devuelve un informe de la contraseña indicando L si esta bloqueada, NP si no hay contraseña y PS si es correcta. También se indica la fecha de ultimo cambio de contraseña, edad máxima de contraseña, periodo de aviso y periodo de inactividad. |
| -a | Se usa junto con S y devuelve un informe de todas las contraseñas. |

-e El usuario esta obligado a cambiar contraseña la próxima vez que inicie sesión.

Ejemplos:

| | |
|---|---|
| **passwd noelia** | **pide una contraseña para noelia y se la asigna.** |
| **passwd -d javier** | **deshabilita la cuenta del usuario javier eliminando su password** |
| **passwd -l director** | **bloquea la cuenta del usuario director poniendo un signo ! delante de su password en el fichero /etc/shadow** |
| **passwd -u director** | **desbloquea la cuenta del usuario director.** |

Ejercicio:

Poner contraseña a los usuarios: director, carlota y pedro

Crear usuario javier con directorio de trabajo javier y ponerle contraseña.

Cerrar sesión y entrar como cualquier usuario de los creados. Volver a entrar como root.

## Comando su

La orden su permite cambiar de usuario. Cuando se invoca, nos pide la palabra clave o password del usuario al que queremos cambiar. Si a su no le pasamos como parámetro ningún nombre de usuario, asumirá que deseamos convertirnos en el administrador del sistema o super usuario (root). Obviamente, si no conocemos la password del usuario, la orden fallará. La opción – se emplea para indicar a su que se tomen los parámetros de inicio (directorio de arranque, variables de entorno, etc) definidos para el usuario al que nos convertimos. Por defecto estos parámetros no se toman.

Sintaxis: su [opciones] [login]

Ejemplo:

| | |
|---|---|
| **su noe** | **cambiamos del usuario actual al usuario noe.** |

## Comando groupadd

Nos permite crear grupos.

Sintaxis: groupadd grupo

Ejemplo:

**groupadd SOM**

**groupadd jefes**

**groupadd empleados**

Ejercicio:

Crea el usuario ana con directorio de trabajo ana y grupo principal SOM.

## Comando gpasswd

El administrador del sistema puede establecer una contraseña para un grupo y nombrar un administrador para el mismo. Dicho administrador podrá ser un usuario cualquiera del sistema. El administrador de un grupo podrá añadir o excluir usuarios de su grupo.

Sintaxis: gpasswd grupo

Modificadores:

-A para nombrar al administrador del grupo

-a para añadir usuarios al grupo

-d para eliminar usuarios del grupo

Ejemplos:

| | |
|---|---|
| **# gpasswd som** | **el root establece una contraseña para el grupo som** |
| **# gpasswd –A profesor som** | **el root nombre al usuario profesor administrador del grupo som** |
| **$ gpasswd –a alumno1 som** | **el administrador del grupo som añade al usuario alumno1 a su grupo** |
| **$ gpasswd –d alumno1 som** | **el administrador del grupo som elimina al usuario alumno1 a su grupo** |

## Comando groups

Este comando permite saber los grupos a los que pertenece un usuario.

Sintaxis: groups nombreusuario

Ejemplo:

**groups ana**

## Comando userdel

El comando userdel permite eliminar definitivamente un usuario del sistema.

Sintaxis: userdel [opciones] <login>

Modificadores:

-r Además de borrar el usuario borra también su directorio en /home. Si no se usa este modificador el directorio se mantiene

Ejemplo:

**userdel ana** **elimina usuario ana pero no su directorio.**

**userdel -r javier** **elimina al usuario javier y borra su directorio base.**

## Comando groupdel

El comando groupdel permite eliminar definitivamente un grupo del sistema. Podremos eliminar a un grupo aunque tengamos a usuarios y recursos que pertenezcan al mismo, siempre y cuando no se trate de un grupo primario de algún usuario dado de alta en el sistema.

Sintaxis: groupdel [opciones] grupo

Ejemplo:

**groupdel SOM**

## Comando usermod

El comando usermod se emplea para modificar algunas propiedades de los usuarios como: el login, el directorio base, el shell que se inicia al conectarse, los grupos a los que pertenece, la fecha de expiración de la cuenta, etc. También bloquea y desbloquea una cuenta.

Sintaxis: usermod [opciones] <login>

Modificadores:

-g para indicar el grupo principal de un usuario.

-G para indicar grupos secundarios de un usuario.

-e para indicar cuando expira la cuenta de un usuario

-l para cambiar el login de un usuario.

Ejemplos:

| | |
|---|---|
| **usermod –g empleados noelia** | **cambia el grupo ppal de noelia a empleados** |
| **usermod -G jefes noelia** | **señala como grupo secundario de noelia a jefes** |
| **usermod -e 2007-13-05 noelia** | **indica que la cuenta de noelia expirará el 13 de mayo de 2007.** |
| **usermod –l noe noelia** | **cambia el login de noelia a noe** |
| **usermod –G grupo1, grupo,2 grupo3 noelia** | **añade al usuario Noelia a los grupos: grupo1, grupo2 y grupo3** |

## Comando chfn

El comando chfn permite cambiar la información de contacto de un usuario. Esta incluye aspectos como: el nombre completo, la oficina de trabajo y los teléfonos. Se almacena en el fichero de usuarios /etc/passwd.

Sintaxis: chfn [opciones] [login]

Ejemplo:

| | |
|---|---|
| **chfn noe** | **comprobar cambios en /etc/passwd.** |

## Comando id

El comando id, imprime dado un usuario, sus identificadores de usuario y de grupo principal (gid y uid) así como del resto de los grupos a los que pertenece. Sin argumentos se asume el usuario actual.

Sintaxis: id [opciones] [login]

Ejemplo:

**id pepe**

**obtenemos: uid=502(pepe) gid=502(pepe) groups=502(pepe),100(users)**

Comprobar que obtenemos con id pedro

## Comando chage

El comando chage nos permita gestionar la información sobre la caducidad de las contraseñas.

Sintaxis: chage usuario

Modificador:

-l obtenemos parámetros actuales

| | |
|---|---|
| **chage –l pedro** | **podemos obtener los parámetros actuales de tiempo de la contraseña del usuario pedro** |

## Comando sudo

En algunos sistemas, como puede ser Ubuntu, existe una orden denominada sudo, que permite ejecutar una orden como el usuario root.

Sintaxis: sudo

Ejemplos:

| | |
|---|---|
| **sudo passwd root** | **Siendo un usuario ordinario nos permite establecer la contraseña para root.** |
| **sudo gedit /etc/shadow** | **Ejecuta gedit /etc/shadow como si lo ejecutará el root.** |

## Comando pwck

Es conveniente ejecutar este comando si hemos modificado manualmente el fichero de **/etc/passwd**, ya que va a realizar un chequeo del fichero para localizar posibles errores de formato, así como posibles errores de inconsistencia (usuarios duplicados, etc).

Sintaxis: pwck

### Comando grpck

Es conveniente ejecutar este comando si hemos modificado manualmente el fichero de **/etc/group**, ya que va a realizar un chequeo del fichero para localizar posibles errores de formato e inconsistencia avisándonos de ello.

Sintaxis: grpck

## 4. PERMISOS EN LINUX.

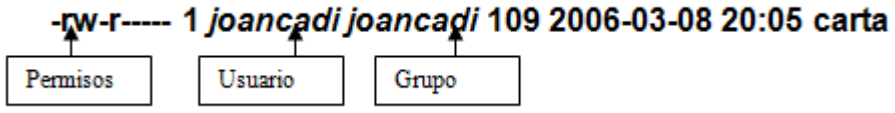
La gestión de permisos en Linux (proceso de autorización para uso de recursos) es bastante distinta de la que usa Windows. Linux usa un esquema de permisos bastante más simple, ya que los recursos sólo admiten 3 tipos de usuarios, y básicamente 3 tipos de permisos. Sin embargo, aunque es un sistema muy simple, con una buena planificación permite desarrollar políticas de seguridad lo suficientemente buenas para el uso cotidiano de un sistema.

**En Linux cada recurso pertenece a un usuario y a un grupo,** y se le pueden establecer permisos en tres niveles:

- Permisos de propietario: persona que crea el fichero (inicialmente, ya que el root podrá cambiar al propietario, por lo que si esto sucede el propietario no será el usuario que creo ese fichero).
- Permisos de grupo: grupo al que pertenece el fichero.
- Permisos del resto de usuarios.

El administrador o root, es el encargado de crear usuarios, de darles de baja y de establecer los privilegios que cada uno de ellos tendrá en el sistema.
El root para facilitarse las tareas de administración, creara grupos de usuarios con características similares a los que les concederá o no ciertos permisos.

Cuando listamos el contenido de un directorio la información que obtenemos es algo parecido a esto:




La columna permisos, consta de 10 caracteres y se divide de la siguiente manera:

**1º carácter**: Nos indica que tipo de fichero es, se vio en el primer tema de Linux.

**2º 3º 4º carácter**. Es el primer trío de permisos, y nos indican los permisos que el usuario propietario del fichero tiene sobre ese fichero (U).

**5º 6º 7º carácter**. Es el segundo trío de permisos, y nos indica los permisos que los usuarios que pertenecen al grupo al que pertenece el fichero tienen sobre el fichero (G).

**8º 9º 10º carácter**. Es el tercer trío de permisos, y nos indica los permisos que los otros (O) tienen sobre ese fichero. Otros se refiere a cualquier usuario que no sea el usuario propietario del fichero (U) ni pertenezca al grupo al que pertenece el fichero (G).

| USUARIO (U) | | | GRUPO (G) | | | OTROS (O) | | |
|---|---|---|---|---|---|---|---|---|
| r | w | x | r | w | X | r | w | x |
| READ | WRITE | EXECUTE | READ | WRITE | EXECUTE | READ | WRITE | EXECUTE |

Para cada uno de estos tres tríos vemos que existen tres tipos de permisos fundamentales:

**r**: read (lectura). El usuario que tenga este permiso podrá si es un directorio listar los recursos almacenados en él, y si es cualquier otro tipo de fichero podrá leer su contenido.

**w**: write (escritura). Todo usuario que posea este permiso para un fichero podrá modificarlo. Si se posee para un directorio se podrán crear y borrar ficheros en su interior.

**x**: execute (ejecución). Este permiso para el caso de los ficheros permitirá ejecutarlos desde la línea de comandos y para los directorios, el usuario que lo posea tendrá acceso para realizar el resto de las funciones permitidas mediante los otros permisos (lectura y/o escritura). Si un usuario no tiene permiso de ejecución en un directorio, directamente no podrá entrar en el mismo, ni pasar por el.

## 4.1 COMANDO PARA GESTIONAR LOS PERMISOS

### Comando chmod.

Para cambiar los permisos de un recurso se utiliza el comando chmod.

Sintaxis: chmod [opciones] <permisos> <ficheros>

Las formas de expresar los nuevos permisos son diversas, se puede emplear la representación numérica o utilizando caracteres.

Utilizando caracteres, la orden chmod se usa así:

Chmod (letra del trío a cambiar) (+ - o =) (permisos) fichero

Donde:

**Letra de trío a cambiar puede ser:**

- U - Usuario (1º trío)
- G - Grupo (2º trío)
- O - Otros (3º trío)
- A - All, todos los tríos.

**El carácter puede ser:**

• +añade el permiso indicado a los que había

• - Quita el permiso indicado a los que había.

• = pone estos permiso exactamente igual que indicamos ahora mismo.

**Permisos puede ser cualquier combinación de:**

• r para leer

• w para escribir

• x para ejecutar

Utilizando la representación numérica, simplemente hay que poner chmod (representación numérica) fichero

Ejemplo:

| | |
|---|---|
| **chmod u+x clase.txt** | **añade el permiso de ejecución al dueño** |
| **chmod g=rx program.sh** | **asigna exactamente los permisos de lectura y ejecución al grupo sobre el fichero programa.sh** |
| **chmod go-w profile** | **elimina el permiso de escritura en el grupo y en otros** |
| **chmod a+r,o-x *.ts** | **adiciona el permiso de lectura para todos los usuarios y elimina el de ejecución para otros** |
| **chmod -R o+r apps/** | **adiciona el permiso de lectura a otros para un directorio de forma recursiva (incluyendo todo su contenido)** |

Ejercicio:

Entrar en el sistema como usuario director y comprobar que estamos en su directorio de trabajo /home/dire.

Crear un fichero llamado profesores con la siguiente información:

Pedro:informatica

Noelia:informatica

Mario:historia

Mila:filosofia

Listar los ficheros y comprobar los permisos que tiene profesores.

Cambiar a usuario noe y comprobar si puede listar /home/dire.

Cambiar permisos del directorio dire para que noe pueda leerlo.

¿Qué permiso hay que darle al usuario noe?

Intentar hacer un cambio en fichero profesores como usuario noe. ¿qué ocurre?.

Hacer cambios necesarios para que noe pueda modificar profesores.

## 5. COMANDOS PARA CAMBIAR EL PROPIETARIO Y EL GRUPO DE UN FICHERO

### Comando chown

El comando chown se utiliza para cambiar el dueño y el grupo de un fichero. Existe también el comando chgrp que se emplea de forma similar pero para cambiar el grupo solamente. El dueño de un fichero solo lo puede cambiar el usuario root mientras que el grupo además de root, lo puede cambiar el propio dueño, siempre que pertenezca al nuevo grupo.

El propietario y el grupo de un fichero se puede comprobar con un ls -l

Sintaxis:

**chown [opciones] <dueño>[.grupo] <ficheros>**

**chown [opciones] <grupo> <ficheros>**

Opción:

-R en los directorios cambia el dueño y/o el grupo recursivamente.

Ejemplos:

| | |
|---|---|
| **chown pepe.pepe tesis/** | **cambia el propietario de tesis a pepe y el grupo de tesis al grupo pepe** |
| **chown -R root /tmp/oculto** | **cambia todos los ficheros que estén en el directorio oculto y coloca como propietario al usuario root** |

Ejercicio:

Como root, cambia el propietario del fichero profesores a noe

### Comando chgrp

Para cambiar de grupo a un fichero o grupo de ficheros es necesario utilizar la orden chgrp, cuya sintaxis es:

chgrp nuevo-grupo fichero(s)

Ejemplo:

| | |
|---|---|
| **chgrp usuarios agenda pruebas** | **cambia los ficheros agenda y pruebas al grupo usuarios.** |

para realizar este cambio, los ficheros nos deben pertenecer. Lógicamente el grupo “usuarios” tiene que estar dado de alta en la máquina.

Ejercicio:

Cambia el grupo del fichero profesores al grupo jefes.

# 6. FICHEROS DE CONFIGURACIÓN DEL SISTEMA

## 6.1 Fichero /etc/passwd

El fichero /etc/passwd es un fichero de texto que contiene información sobre los usarios del sistema.

Cuando el usuario teclea su identificación y su contraseña, el sistema operativo busca en /etc/passwd. Si la información es correcta, Linux ejecuta el Shell para ese usuario, quien a su vez se encarga de ejecutar el fichero .profile del usuario en cuestión. A continuación sale el prompt del sistema ($) para un usuario normal y # para el root.

Cada línea del fichero contiene información acerca de un único usuario; el formato de cada línea es:

*nombre:clave encríptada:UID:GID:nombre completo:dir-personal:interprete*

Cada campo está separado por dos puntos. Un ejemplo puede ser:

*Crispin:TBe1jvLU$QWNBxcwdVM:506:506:Javi Ruiz:/home/Crispin:/bin/bashs*

**El primer campo**, «Crispin», es el **nombre de usuario**.

El **segundo campo**, TBe1jvLU$QWNBxcwdVM, es la **clave encriptada**. Las claves no se almacenan en un formato legible, son encriptadas utilizándose a sí mismas como clave. En otras palabras, solo si se conoce la clave, esta puede ser desencriptada. Es una forma de encriptación conocida como «de un solo sentido». El mecanismo es bastante seguro, en realidad la contraseña nunca se desencripta, lo que sucede al hacer un login (acceder a el sistema identificándose mediante el nombre de usuario y la contraseña) es que la contraseña se vuelve a encriptar y se compara con la que tenemos almacenada. Si las dos coinciden, se reconoce al usuario y se le permite entrar.

Una forma más segura de almacenar las contraseñas es el sistema «shadow password» («claves en sombra») en la que la información de las claves se relega al fichero /etc/shadow.

En este caso en el campo de la contraseña de /etc/passwd se sustituye por una x, de esta manera tendremos:

*Crispin:x:506:506:Javi Roman:/home/Crispin:/bin/bashs*

El problema de /etc/passwd es que es legible por todo el mundo, mientras que /etc/shadow no lo es, por lo que suministra un grado extra de seguridad. Las claves shadow suministran otras funciones como la expiración de claves.

Si necesitamos deshabilitar una cuenta temporalmente, basta con colocar un asterisco (*) delante de la clave encriptada:

*Crispin:*TBe1jvLU$QWNBxcwdVM:506:506:Javi Ruiz/home/Crispin:/bin/bashs*

El asterisco (*) no es un carácter válido para un campo encriptado. Si después necesitamos usar esta cuenta de nuevo, solamente tendremos que retirar el asterisco.

**El tercer campo** «506», **es el UID** (Número de Identificación de Usuario). Este debe ser único para cada usuario.

**El cuarto campo**, «506», **es el GID** (Número de Identificación de Grupo). Este usuario pertenece al grupo numerado 506. La información de grupos, se almacena en el fichero /etc/group.

**El quinto campo** es el **nombre completo del usuario**. «Javi Ruiz». Este campo es opcional rellenarlo. Es útil en sitios donde tengamos muchos usuarios para la identificación de los mismos.

**El sexto campo** es el directorio personal del usuario (/home/Crispin). No es necesario que el directorio inicial de un usuario tenga el mismo nombre que el del nombre de usuario. Sin embargo, ayuda a identificar el directorio.

**Y el último campo es intérprete de comandos** que usará para esa sesión (/bin/bash).

## 6.2 Fichero /etc/shadow

En este fichero se guardan las contraseñas de los usuarios así como su configuración.

Contiene una línea por cada una del fichero /etc/passwd. Cada línea tiene una serie de campos separados por el carácter (**:).** Cada campo significa lo siguiente:

**El primer campo** es el nombre de usuario.

**El segundo campo** contiene la contraseña encriptada. En aquellos usurarios que son propios del sistema contiene una x.

**El tercer campo** es el número de días transcurridos entre el 1 de enero de 1970 y el día en que la contraseña fue modificada por ultima vez.

**El cuarto campo** es el número mínimo de días requeridos para que la contraseña pueda ser cambiada por el usuario.

**El quinto campo** es el número máximo de días que una contraseña es válida. Pasado ese periodo el usuario es forzado a cambiarla.

**El sexto campo** es el número de días de antelación con el que el sistema avisa al usuario que su contraseña va a expirar.

**El séptimo campo** es el número de días de inactividad que el sistema permite al usuario. Si el usuario no se conecta al sistema en ese intervalo de días, el sistema no le permitirá la conexión.

**El octavo campo** son los días desde el 1 de enero de 1970 a partir de cual el nombre de presentación no podrá ser utilizado.

**El noveno campo** no se utiliza.

## 6.3 Fichero /etc/group

El fichero /etc/group contiene información acerca de los grupos que existen en el sistema. El formato de cada línea es:

*nombre de grupo:clave:GID:otros miembros*

Algunos ejemplos de grupos pueden ser:

root:*:0:
usuarios:*:100:mdw,larry
invitados:*:200:
otros:*:250:kiwi

El primer grupo, root, es un grupo especial del sistema reservado para la cuenta root. El siguiente grupo, usuarios, es para usuarios normales. Tiene un GID de 100. Los usuarios mdw y larry tienen acceso a este grupo.

Los usuarios pueden pertenecer a mas de un grupo, añadiendo sus nombres de usuario a otras líneas de grupo en /etc/group. El comando groups lista a que grupos se tiene acceso.

El tercer grupo, invitados, es para usuarios invitados, y otros es para «otros» usuarios. El usuario kiwi tiene acceso a este grupo.

Como se puede ver, el campo clave de /etc/group raramente se utiliza. A veces se utiliza para dar una clave para acceder a un grupo.

Se pueden usar el comando groupadd para añadir grupos a su sistema. Normalmente es más sencillo añadir líneas a /etc/group uno mismo, puesto que no se necesitan más configuraciones para añadir un grupo. Para borrar un grupo, solo hay que borrar su entrada de /etc/group o usar el comando correspondiente.

## 6.4 Otros ficheros de configuración:

### 6.4.1 Fichero /etc/fstab

En Linux los dispositivos físicos de la máquina en general y los de almacenamiento de información, en particular, son manipulados a través de ficheros especiales ubicados en el directorio /dev. Los discos duros, las particiones de estos, las unidades de disquete, los CD-ROM o los dispositivos de almacenamiento USB son ejemplos de estos con los cuales interactuamos constantemente. Pero trabajar directamente sobre los dispositivos representados de esa forma casi nunca es conveniente ni resulta cómodo, por lo que usualmente se incorporan al sistema de ficheros tradicional.

Esta acción se conoce como ``montar'', que en definitiva es asociar el dispositivo a un directorio determinado.

Como se explicó anteriormente las particiones de los discos en Linux se montan en directorios como /, /home y /usr. El sistema tiene un fichero llamado /etc/fstab en el cual se especifican donde y en que forma se montan los diferentes

dispositivos. Si mostramos este fichero:

$ cat /etc/fstab

Obtenemos:

```
# /etc/fstab: static file system information.
# <file system>   <mount point>   <type>   <options>          <dump>   <pass>
proc              /proc           proc     defaults           0        0
/dev/hd3          /               ext3     defaults,errors=r  0        0
                                           emount-ro
/dev/hda7         none            swap     sw                 0        0
/dev/hdc          /media/cdrom0   udf,     user,noauto        0        0
                                  iso9660
```

Cada línea en este fichero describe un dispositivo, con seis campos cada uno de ellos indicando los siguientes aspectos:

**1.- Nombre del dispositivo o etiqueta**. Especifica el nombre del dispositivo a montar. Ejemplos: /dev/hda1, /dev/sdc1, /dev/fd0, etc.

**2.- Directorio donde se monta**. Ejemplos: /, /mnt/floppy, /tmp, etc.

**3.- Sistema de ficheros**. Especifica el tipo de sistema de ficheros. Ejemplos: ext2, msdos, nfs, swap, iso9660, auto, etc.

**4.- Opciones de montaje**. Especifica las opciones de montaje asociadas al sistema de ficheros. Van separados por comas, y algunas de ellas son:

- **auto** : indica que el dispositivo se monta siempre que se inicie el sistema. La opuesta es **noauto.**
- **rw:** indica que el dispositivo se monta con permisos de lectura y escritura.
- **ro:** indica que el dispositivo se monta con permisos de lectura solamente.
- **owner:** indica que el usuario conectado al sistema localmente en primer lugar tiene derechos a montar y desmontar el dispositivo (se adueña de este).
- **user** : indica que cualquier usuario puede montar y solo el mismo usuario podrá desmontar el dispositivo. La opción opuesta es **nouser.**
- **users** : indica que cualquier usuario puede montar y cualquiera también, puede desmontar el dispositivo.
- **exec** : indica que los binarios ejecutables almacenados en el dispositivo se pueden ejecutar. La opción opuesta es **noexec**.
- **async** : expresa que todas las operaciones de entrada y salida se hacen de forma asíncrona, o sea, no necesariamente en el momento en que se invocan. La opción opuesta es **sync**.

- **dev** : indica que se interprete como tal a los dispositivos especiales de bloques y de caracteres presentes en el dispositivo. La opción opuesta es **nodev.**
- **defaults :** es una opción equivalente a la unión de **rw, dev, exec, auto, nouser y async.**

Ejemplos: ro, rw, exec, auto, user, etc.

**5.- Frecuencia:** determinamos la frecuencia con la que se deben realizar copias de seguridad del sistema correspondiente por el comando dump.

**6.- Secuencia**: es usado por el comando fsck (File System Check, verificación del sistema de archivos) para determinar el orden en que se realizan los chequeos de los sistemas de archivos en el arranque. Si el campo es cero, este sistema de ficheros no se chequeará

En el fichero /etc/fstab siempre hay varias entradas especiales:

• Una línea para el sistema de archivos raíz.

• Una línea para el sistema de archivos /dev/pts.

• Una línea para el sistema de archivos /proc.

• Una línea para la partición swap. Estas particiones no son visibles en la estructura árbol, y el campo correspondiente al punto de montaje contiene la palabra clave swap.

Para montar y desmontar los dispositivos se emplean los comandos **mount y umount** respectivamente, estudiados en el tema 7.

### 6.4.2 Fichero /etc/mtab

En este fichero se mantiene una lista de los dispositivos montados.

El archivo /etc/mtab es actualizado por el programa mount cada vez que se montan o desmontan sistemas de archivos. He aquí una muestra de /etc/mtab:

```
/dev/sda3 / ext3 rw 0 0
none /proc proc rw 0 0
usbdevfs /proc/bus/usb usbdevfs rw 0 0
/dev/sda1 /boot ext3 rw 0 0
none /dev/pts devpts rw,gid=5,mode=620 0 0
/dev/sda4 /home ext3 rw 0 0
none /dev/shm tmpfs rw 0 0
none /proc/sys/fs/binfmt_misc binfmt_misc rw 0 0
```

El archivo /etc/mtab se utiliza para mostrar el estado de los sistemas de archivos montados actualmente. No se debería modificar manualmente.Cada línea representa un sistema de archivos que está actualmente montado y contiene los campos siguientes (de izquierda a derecha):

- La especificación del dispositivo

- El punto de montaje
- El tipo de sistema de archivos
- Si el archivo está montado como de sólo lectura (ro) o de sólo escritura (rw), junto con cualquier otra opción de montaje
- Dos campos sin utilizar llenos de ceros (para la compatibilidad con /etc/fstab)

## Comando df

Mientras que el uso de /etc/mtab le permite conocer los sistemas de archivos que se encuentran montados actualmente, hace muy poco más allá de eso. La mayoría de las veces un administrador estará quizás más interesado en un aspecto particular de los sistemas de archivos montados actualmente, la cantidad de espacio disponible en ellos.

Para esto, podemos utilizar el comando df. He aquí una muestra de la salida de df:

```
Filesystem          1k-blocks      Used    Available   Use%   Mounted on
/dev/sda3            8428196   4280980   3719084    54%        /
/dev/sda1            124427    18815     99188       16%        /boot
/dev/sda4            8428196   4094232   3905832    52%        /home
none                 644600        0         644600      0%         /dev/shm
```

Hay muchas diferencias obvias entre /etc/mtab que se ven inmediatamente:

- Se muestra un encabezado fácil de leer
- Con la excepción del sistema de archivos de la memoria compartida, solamente se muestran los sistemas de archivos basados en disco
- Tamaño total, espacio utilizado, espacio libre y el porcentaje de uso en números

El último punto es probablemente el más importante puesto que eventualmente todo administrador de sistemas tendrá que enfrentarse a un sistema que se encuentra sin espacio disponible en disco. Con df es fácil ver donde reside el problema.