

COMANDOS LINUX PARA GESTIÓN DE USUARIOS Y GRUPOS

1.-PERMISOS.....	3
2.-CAMBIO DE PERMISOS.	5
3.-CAMBIANDO LOS PROPIETARIOS DE LOS FICHEROS.....	6
4.-GESTIÓN DE USUARIOS.....	7
4.1.Archivos implicados en la gestión de usuarios.....	7
4.2.-Altas de usuarios haciendo uso del comando useradd	8
4.3.-Directorio home del usuario. Creación manual	9
4.4.-Baja de usuarios utilizando el comando userdel	10
4.5.-Modificación de información administrativa de usuarios.....	10
5.-GESTIÓN DE GRUPOS	12
5.1.-Archivos implicados en la gestión de grupos.....	12
5.2.-Alta de grupos con el comando groupadd.....	12
5.3.-Modificación de información de grupos	13
6.-OTROS COMANDOS RELACIONADOS DIRECTA O INDIRECTAMENTE CON LA GESTIÓN DE USUARIOS Y GRUPOS	14
7.-ALGUNOS FICHEROS DE CONFIGURACIÓN DEL ENTORNO DE LOS USUARIOS 15	
8.-PERSONALIZACIÓN DE LA PANTALLA DE LOGIN EN MODO TEXTO.....	16

1.-PERMISOS

A la hora de establecer los permisos para los ficheros podemos distinguir tres entidades fundamentales: el usuario propietario del fichero, el grupo propietario (el usuario propietario no tiene porqué pertenecer a éste) y el resto de usuarios. Para cada una de estas entidades se establecen tres permisos diferentes (lectura, escritura y ejecución). Puesto que es suficiente un bit para reflejar cada uno de los dos posibles estados de un permiso (si se tiene o no se tiene permiso) se requieren nueve bits para representar los mismos (como veremos más adelante hay dos bits adicionales con propósitos específicos).

¿Cómo podemos ver los permisos de los ficheros?

Los permisos se pueden ver en la primera columna que se obtiene al utilizar el comando ls con la opción -l. El primer carácter de esa columna indica el tipo de fichero, que puede ser:

- : Fichero normal.
- **b** : Fichero especial de bloques
- **c** : Fichero especial de carácter.
- **d** : Directorio.
- **l** : Enlace simbólico.
- **p** : Tubo o pipe.
- **s** : Socket.

el resto de caracteres de la columna se agrupan en tres secciones de tres caracteres cada una. En cada sección se especifican los permisos en este orden: lectura-escritura- ejecución (rwx), la ausencia de un permiso se representa por el guión (-). La sección más a la izquierda especifica los permisos para el usuario, la sección central, lo hace para el grupo propietario, y por último, la sección de la derecha se refiere a los permisos del resto de usuarios.

Ejemplo:

```
- r w x r w - r - -
```

La línea anterior indica que se trata de un fichero normal, con permisos de lectura, escritura y ejecución para el usuario propietario, lectura y escritura para el grupo propietario y lectura para el resto de usuarios.

¿Qué significa cada uno de los permisos?

El significado de cada permiso depende del tipo de fichero al cual se aplica. Para el caso de ficheros normales tenemos lo siguiente:

r (lectura): Permite ver el contenido del fichero, bien con cualquier editor o haciendo uso de filtros como es el caso del fichero cat. En el primer caso no es posible hacer cambios al contenido del fichero si no está habilitado el permiso de escritura.

w (escritura): Permite modificar el contenido de un fichero. Con cualquier editor (siempre que esté habilitado el permiso de lectura), o bien redireccionando la salida de cualquier programa a éste.

x (ejecución): Permite la ejecución del fichero, si ello es posible, es decir, si se trata realmente de un fichero ejecutable o de un guión del shell (para poder ejecutar un fichero debe existir, además del preceptivo permiso de ejecución, permiso de lectura del mismo).

Para los directorios la interpretación se da a continuación:

r (lectura): Permite ver el contenido del directorio.

w (escritura): Permite modificar el contenido de un directorios, es decir, añadir y/o eliminar ficheros y directorios (para poder modificar un directorio es imprescindible, tener además, permiso de ejecución).

x (ejecución): Permite convertir el directorio en el directorio activo.

2.-CAMBIO DE PERMISOS.

COMANDO chmod

A la hora de utilizar este comando disponemos de dos opciones: sintaxis numérica y sintaxis no numérica.

Este comando puede aplicarlo el superusuario (root) para cambiar los permisos de cualquier fichero. En cambio los usuarios normales únicamente pueden cambiar los permisos de los ficheros que les pertenezcan. Esto se ha de tener muy en cuenta a la hora de utilizar el comando con la opción -R.

2.1.-COMANDO chmod. SINTAXIS NUMÉRICA

```
chmod [-R] XYZ fichero1 [ fichero2 ...]
```

XYZ: son tres dígitos octales que especifican los permisos que deseamos establecer para el fichero.

Ejemplo:

```
chmod 761 mifichero.txt
```

Establece para el fichero mifichero.txt, los siguientes permisos:

- Para el usuario propietario: rwx (7(8=111(2)
- Para el grupo propietario: rw- (6(8=110(2)
- Para el resto de usuarios: --x (1(8=001(2) La opción -R se utiliza para forzar recursivamente el cambio de permisos para los ficheros especificados que sean directorios.

3.-CAMBIANDO LOS PROPIETARIOS DE LOS FICHEROS.

COMANDO chown

Hemos visto que todo fichero tiene un propietario y un grupo propietario, y que cada una de estas entidades tiene asociados una serie de permisos. Ahora vamos a ver cómo es posible hacer un cambio de propiedad para los ficheros.

El comando destinado a tal efecto obedece a la siguiente sintaxis:

```
chown [-R] nuevoUsuarioProp.nuevoGrupoProp fichero1 [ fichero2 ...]
```

Interpretación de los parámetros:

nuevoUsuarioProp: es el nombre del nuevo usuario propietario del fichero o ficheros.

nuevoGrupoProp: nombre del nuevo grupo propietario del fichero o ficheros (nuevoUsuarioPropietario no tiene porqué pertenecer a este grupo). Si omitimos este parámetro, el sistema asume que el nuevo grupo propietario es el grupo en el cual se encuentra nuevoUsuarioPropietario).

-R: si alguno de los ficheros especificados es un directorio, cambia el propietario de manera recursiva para el contenido del mismo.

Únicamente el superusuario (root) puede cambiar los propietarios (usuario y grupo) de cualquier fichero, los usuarios únicamente pueden hacerlo para los ficheros de los cuales sean propietarios. Esto se ha de tener muy en cuenta a la hora de utilizar la opción -R.

Ejemplos:

Si el directorio undirectorio es propiedad del usuario pepe y pertenece al grupo pepes y ejecutamos el comando:

```
chown paco.pepes undirectorio
```

el directorio pasará a ser propiedad del usuario paco y seguirá perteneciendo al grupo pepes.

Si en cambio se hubiese ejecutado el comando:

```
chown -R pepe.pacos undirectorio
```

entonces, tanto el directorio y su contenido hubiesen cambiado de grupo propietario, que sería pacos, siempre y cuando el comando lo hubiera ejecutado root o el usuario propietario del directorio y de todo su contenido.

4.-GESTIÓN DE USUARIOS

Se puede decir que la gestión de usuarios tiene dos fines fundamentales: § Garantizar que la información de cada usuario esté libre de accesos indebidos § Adaptar el entorno de trabajo de cada usuario a sus necesidades específicas.

A la hora de gestionar usuarios hemos de tener en cuenta que cada usuario tiene asociados, entre otros, los siguientes elementos:

Un nombre de usuario y una contraseña que le servirán para iniciar una sesión en el sistema.

Un identificador numérico único (uid), asignado, en la mayoría de los casos, por el administrador.

Un directorio de trabajo (HOME).

Uno o varios ficheros de configuración de su entorno de trabajo, ubicados en su directorio HOME.

Un shell.

Un grupo, en el cual se encontrará al entrar al sistema.

La gestión de usuarios incluye tareas como dar de alta usuarios, eliminarlos y modificar sus propiedades (es decir, alterar el valor de cualquiera de sus elementos asociados).

Para dar de alta un usuario tenemos dos opciones. Una de ellas consiste en modificar manualmente los archivos del sistema implicados en el proceso. La otra, con menos riesgos de cometer errores, consiste en hacer uso de alguna herramienta del sistema.

4.1. Archivos implicados en la gestión de usuarios

4.1.1.-Archivo /etc/passwd

El archivo **/etc/passwd** es el que contiene toda o casi toda la información administrativa de los usuarios. Este archivo es de sólo lectura para todos los usuarios, el único usuario que tiene permiso para modificarlo es root. Se trata de un archivo de texto, formateado en campos. En cada línea se define un usuario, y como separador de campos se usa el carácter ':'. La estructura y significado de los campos se muestra a continuación.

Primer campo: contiene el nombre del usuario, este es su identificador simbólico (también llamado login), el empleado para acceder al sistema.

Segundo campo: tiene que ver con la contraseña. Si se usa el método de contraseñas en la sombra, cosa que vamos a suponer, la contraseña se encuentra encriptada en el archivo **/etc/shadow** (no visible para los usuarios), y aquí figurará una x.

Tercer campo: es el uid del usuario.

Cuarto campo: gid del grupo al que de forma predeterminada pertenece el usuario. Al igual que el uid, el gid es un número entero único que identifica a un grupo.

Quinto campo: es un comentario, lo más habitual es poner el nombre real del usuario, aunque podemos poner lo que queramos.

Sexto campo: es la ruta absoluta del directorio de trabajo del usuario. Los directorios de trabajo de los usuarios se deben colocar dentro del directorio /home y su nombre es recomendable que coincida con el nombre de usuario.

Séptimo campo: es la ruta del shell que se ejecutará cuando el usuario acceda al sistema. Los shells disponibles en el sistema se encuentran en el directorio /bin. En realidad podemos indicar la ruta de cualquier archivo ejecutable.

4.1.2.-Archivo /etc/shadow

El archivo **/etc/shadow** es el archivo donde se guardan las contraseñas encriptadas. Se trata, al igual que el fichero **/etc/passwd**, de un fichero de texto con una línea para cada usuario, y con campos delimitados por el carácter :. Nosotros únicamente vamos a ver los dos primeros campos:

Primer campo: contiene el nombre del usuario.

Segundo campo: es el destinado a contener la contraseña. Cabe señalar las siguientes particularidades:

- Si el usuario no tiene contraseña, aparecerá vacío.
- Si el usuario tiene contraseña esta figurará encriptada.
- Cuando se bloquea la cuenta del usuario con el comando **passwd** (consultar más abajo), si el campo estaba vacío (el usuario no tenía contraseña), el sistema le coloca el carácter !, en caso contrario, delante de la contraseña encriptada se insertará el carácter !.

4.1.3.-Archivo /etc/group Este archivo lo vemos en el apartado correspondiente a gestión de grupos.

4.2.-Altas de usuarios haciendo uso del comando useradd

Esta es la forma más rápida de dar de alta un usuario. La sintaxis del comando es la siguiente:

```
useradd [-c comentario] [-d directorio_home] [-g grupo_inicial] [-G grupo2[,...]] [-m] [-p contraseña] [-s shell] [-u uid] nombre_usuario
```

Significado de las opciones:

-c comentario: el texto que formara parte del campo comentario en **/etc/passwd**.

-d directorio_home: el directorio de trabajo del usuario.

-g grupo_inicial: el grupo al que pertenecerá el usuario cuando acceda al sistema. Este grupo esta definido en **/etc/group**.

-G grupo1[,...]: lista de grupos en los que el usuario se encontrara incluido, si bien su pertenencia se hará efectiva de forma explicita por el propio usuario con el comando newgrp. (Aunque con fines administrativos y de compartición de recursos un usuario puede formar parte de varios grupos, en un momento determinado sólo puede pertenecer a uno de ellos). Estos grupos deben de estar definidos previamente en el archivo **/etc/group**.

-m : indica que si no existe el directorio home del usuario, se cree. Además se copiarán en él los archivos y subdirectorios del directorio **/etc/skel**.

-p contraseña : es la contraseña encriptada. Si no se especifica esta opción se bloquea la cuenta.

Para encriptar la contraseña necesitamos recurrir a alguna herramienta del sistema. Si no queremos complicarnos la vida no pondremos nada y tras ejecutar el comando tenemos varias opciones:

- Editar el archivo `/etc/shadow` y quitar el bloqueo de la contraseña dejando el campo correspondiente vacío (eliminando el carácter `!`), el usuario no tendrá contraseña (esto mismo se puede hacer de otra forma como se indica en la opción siguiente).
- Usar el comando `passwd` con la opción `-u` para desbloquear la cuenta del usuario. :

```
passwd -u nombre_usuario
```

- Usar el comando `passwd` para fijar una contraseña para el usuario, de esta forma se desbloquea la cuenta y el usuario sólo puede acceder al sistema si conoce la contraseña.

```
passwd nombre_usuario
```

`-s shell`: indicamos el shell que se ejecutará cuando el usuario acceda al sistema

`-u uid` : número entero que identificará al usuario. Si no se especifica, el sistema asignará un número una unidad mayor que el id del usuario que tenga el uid más alto, si este es mayor de 99, o 100 en este último caso.

`nombre_usuario` : Para finalizar el comando pondremos el nombre del usuario (login).

En este comando el único parámetro obligatorio es el nombre del usuario `nombre_usuario`. Los datos necesarios para la creación de la cuenta que no se especifiquen en el comando se toman del archivo `/etc/default/useradd`. Estos valores predeterminados del sistema podemos verlos además de editando el citado archivo, usando el comando `useradd` con la opción `-D`. Asimismo, esos valores predeterminados se pueden cambiar editando el archivo o usando el comando `useradd` con la opción `-D` junto con otras opciones (para obtener la descripción de las mismas, se remite al lector a consultar páginas man).

Por último, el orden de uso de los parámetros no tiene porqué ser como aparece arriba, el único requisito es que el nombre del usuario (login) se escriba al final.

4.3.-Directorio home del usuario. Creación manual

El primer paso que daremos para crear el directorio personal de un usuario será abrir una shell como root porque en el directorio `/home` es el único usuario que tiene permiso de escritura.

A continuación copiaremos los archivos predeterminados del sistema al directorio `/home`, haciendo que el directorio copiado tenga el nombre del usuario:

Copiamos todos los archivos y directorios de `/etc/skel`:

```
cp -r /etc/skel /home/nombre_usuario
```

El siguiente paso consiste en hacer que el propietario del directorio que acabamos de crear y de todos sus archivos y subdirectorios pase a ser el nuevo usuario.

```
chown -R nombre_usuario.nombre_grupo /home/nombre_usuario
```

Y por último, cambiaremos los permisos del directorio para que nadie pueda leer ni escribir en él (los permisos de los archivos y subdirectorios que hemos copiado de `/etc/skel` podemos dejarlos como están, por eso ahora no usamos la opción `-R`).

```
chmod 700 /home/nombre_usuario
```

4.4.-Baja de usuarios utilizando el comando userdel

El comando `userdel` proporciona una forma rápida de eliminar usuarios del sistema, a diferencia del comando `useradd`, su sintaxis es muy sencilla.

```
userdel [-r] nombre_usuario
```

La opción `-r` indica que se borre, además del usuario el directorio HOME del mismo.

Se ha de tener en cuenta que este comando no recorre el sistema de ficheros en busca de ficheros y directorios propiedad del usuario. Si queremos eliminarlos tendremos que hacerlo nosotros.

4.5.-Modificación de información administrativa de usuarios

Al igual que para las altas y bajas tenemos dos opciones, modificación manual y automática.

4.5.1.-Modificación manual

Esta modificación implica efectuar manualmente la modificación de cualquiera de los datos, es decir, modificar el archivo `/etc/passwd` y/o el archivo `/etc/shadow` y/o `/etc/group`.

Los pasos a realizar serán más o menos en función del dato a cambiar, por ejemplo si modificamos el comentario o el shell, la única acción será modificar el campo oportuno en el archivo `/etc/passwd`, en cambio para cambiar su directorio HOME, aparte de la modificación del archivo anterior, se ha de crear el nuevo directorio, cambiar en éste el propietario y los permisos y, por último y muy importante, copiar todos los archivos y subdirectorios contenidos en su directorio HOME anterior, respetando la misma estructura de directorios y los permisos que tuvieran.

4.5.2.-Modificación utilizando el comando usermod

Este comando es muy parecido al comando `useradd`, su sintaxis es la siguiente:

```
usermod [-c nuevo_comentario] [-d nuevo_directorio_home [-m]] [-g nuevo_grupo_inicial] [-G nuevo_grupo2[,...]] [-p contraseña] [-s nuevo_shell] [-u nuevo_uid] nombre_usuario
```

La descripción de las distintas opciones y parámetros se realiza a continuación:

`-c comentario` : igual que en `useradd`.

`-d nuevo_directorio_home` : igual que en `useradd`.

`-m` : esta opción sólo puede aparecer acompañada de la anterior. Indica que se copien al nuevo directorio HOME, los archivos y subdirectorios del directorio HOME anterior, y, que si el nuevo directorio no existe se ha de crear.

`-g nuevo_grupo_inicial` : igual que en `useradd`. `-G nuevo_grupo2[,...]` : igual que en `useradd`.

`-l nuevo_nombre_usuario` : esta opción se utiliza para especificar el nuevo nombre de usuario.

`-p nueva_contraseña` : igual que en `useradd`.

`-s shell` : igual que en `useradd`.

`-u nuevo_uid` : igual que en `useradd`. `nombre_usuario` : nombre del usuario (login) al que se van a

aplicar los cambios.

5.-GESTIÓN DE GRUPOS

5.1.-Archivos implicados en la gestión de grupos

Esta operación implica la modificación del archivo `/etc/group`, al igual que el archivo `/etc/passwd`, se trata de un archivo de texto, con la información distribuida en campos separados por el carácter ':'. Su estructura se muestra a continuación:

Primer campo: identificador simbólico de grupo (nombre).

Segundo campo: contraseña del grupo (mismo sistema que en `/etc/passwd`), pero en este caso las contraseñas encriptadas se encuentran en `/etc/gshadow`.

Tercer campo: identificador numérico del grupo (gid).

Cuarto campo: lista de nombres de usuarios (login) que pertenecen al grupo. De tener que especificar más de un usuario, los login se separan entre sí con una coma.

Es preciso matizar, que para el grupo inicial del usuario, es decir, aquel con el que accederá al sistema, no es preciso especificar en este archivo el uid de dicho usuario, pues esa relación de pertenencia al grupo ya queda reflejada en el archivo `/etc/passwd`.

Para dar de alta un nuevo grupo simplemente añadiremos una línea con los datos del mismo.

5.2.-Alta de grupos con el comando `groupadd`

Al igual que sucedía con el comando `useradd`, este comando realiza todas las modificaciones necesarias de forma automática. Su sintaxis simplificada es la siguiente:

```
groupadd [-g gid] nombre_grupo
```

La descripción de las opciones que se han especificado se muestra a continuación, la del resto, como siempre, se puede encontrar en las páginas man.

`-g gid` : identificador numérico del grupo (gid). En Linux, si no se especifica, el sistema crea un grupo con gid una unidad mayor que el gid del grupo que lo tenga más alto, si este es mayor de 499, sino lo es, utilizará un gid igual a 500 (los gid 0 a 499 están reservados al sistema).

`nombre_grupo` : identificador simbólico del grupo.

5.2.1.-Baja de grupos utilizando el comando `groupdel`

La sintaxis de este comando es la siguiente:

```
groupdel nombre_grupo
```

Al igual que en el caso anterior, no se puede eliminar un grupo que sea grupo inicial de algún usuario, sin eliminar primero el usuario.

El comando no recorre el sistema de ficheros en busca de archivos propiedad del grupo. Si deseamos eliminarlos, hemos de hacerlo nosotros.

5.3.-Modificación de información de grupos

Al igual que en los casos anteriores tenemos dos opciones: hacer los cambios manualmente o utilizar un comando.

5.3.1.-Modificación manual de grupos

Dependiendo del tipo de modificación, habrá que modificar uno o más de los siguientes archivos:

- /etc/group
- /etc/passwd
- /etc/gshadow
- También en función del tipo de cambio, es posible que se deba cambiar el grupo propietario en los archivos y directorios propiedad del grupo objeto de la modificación.

Si por ejemplo cambiamos el gid de un grupo, tendremos que cambiar ese gid en el archivo **/etc/passwd** para aquellos usuarios cuyo grupo inicial sea el que acabamos de modificar. Además se ha de cambiar el grupo propietario en los archivos y directorios propiedad del grupo cuyo gid hemos modificado.

En el caso de cambiar el nombre del grupo, sólo es precisa la modificación en el archivo **/etc/group**.

5.3.2.-Modificación de grupos con el comando groupmod

La sintaxis básica de este comando es:

```
groupmod [-g gid_nuevo] [-n nombre_nuevo] nombre_actual
```

Este comando únicamente realiza cambios en el archivo **/etc/group**, por lo tanto el resto de modificaciones implicadas en el cambio del gid de un grupo, se han de hacer manualmente (ver penúltimo párrafo del epígrafe anterior).

5.3.3.-Altas y bajas de usuarios en un grupo.

Para dar de alta un usuario en un grupo (no en el grupo de inicio de sesión), teclearemos la orden:

```
gpasswd -a login nombre_grupo
```

Para dar de baja un usuario en un grupo (no en el grupo de inicio de sesión), teclearemos la orden:

```
gpasswd -d login nombre_grupo
```

6.-OTROS COMANDOS RELACIONADOS DIRECTA O INDIRECTAMENTE CON LA GESTIÓN DE USUARIOS Y GRUPOS

Comando passwd

Se utiliza para modificar información administrativa de usuarios, su sintaxis es la siguiente:

```
passwd [-l] [-u [-d]] [nombre_usuario]
```

Este comando utilizado sin opciones produce como resultado el cambio de la contraseña para el usuario que lo ejecuta. Las opciones y parámetros, cuya descripción se da a continuación, sólo pueden utilizarse por root.

-l : bloquea la cuenta del usuario que se indique (el bloqueo supone que el usuario no podrá acceder al sistema, y se consigue precediendo la contraseña encriptada con el carácter '!').

-u : desbloquea la cuenta del usuario que se indique.

-d : elimina la contraseña de una cuenta del usuario indicado (el usuario podrá acceder al sistema sin especificar contraseña).

nombre_usuario : especifica el usuario al que se le aplicarán los cambios indicados en por alguna de las opciones anteriores, si únicamente se utiliza nombre de usuario, el resultado es el cambio de contraseña.

Comando su

Este comando permite ejecutar un shell con identificadores de grupo y de usuario distintos a los correspondientes al shell que se está ejecutando actualmente.

Si el superusuario desea acceder al sistema a través de un terminal que está siendo usado por otro usuario, en lugar de cerrar la sesión del usuario y entrar como root, lo que implicaría finalizar todos los procesos en curso, tiene la opción de iniciar un nuevo shell sobre el anterior. Una vez realizadas las tareas que han motivado el cambio de usuario, el nuevo shell se cierra con el comando exit.

Sintaxis:

```
su [-s shell] nombre_usuario
```

la opción `-s shell` permite ejecutar un shell distinto al que aparece en el archivo `/etc/passwd` para el usuario indicado en el comando.

Si el usuario indicado en el comando tiene contraseña, se pedirá al usuario que lo ejecuta si este no es root.

Comando newgrp

Sintaxis:

```
newgrp [grupo]
```

Cambia el grupo del usuario que ejecuta el comando. El grupo especificado debe ser un grupo a los que pertenezca el usuario. Para volver al grupo inicial, se ha de usar el comando `exit`.

7.-ALGUNOS FICHEROS DE CONFIGURACIÓN DEL ENTORNO DE LOS USUARIOS

Los ficheros de configuración del entorno de los usuarios se encuentran en su directorio **home**.

A continuación se indican algunos de ellos:

Los ficheros **.bash_profile** (en algunos sistemas será `.profile`) y **.basrc** son archivos que contienen scripts de shell, estos archivos se ejecutan cuando el usuario inicia sesión, y se utilizan para configurar determinados aspectos, como por ejemplo la definición de algunos alias.

El fichero **.bash_logout**, que al igual que los anteriores, contiene scripts de shell. Este archivo se ejecuta cuando el usuario cierra sesión.

El fichero **.bash_history** contiene la lista de órdenes tecleadas recientemente por el usuario. Estas órdenes son accesibles mediante la pulsación de los cursores del teclado.

8.-PERSONALIZACIÓN DE LA PANTALLA DE LOGIN EN MODO TEXTO.

El contenido del archivo **/etc/issue** se imprime en las pantallas de login de las consolas en modo texto del equipo. El contenido de **/etc/issue.net** se imprime en las pantallas de login de los terminales que se conectan vía telnet al sistema.

Por lo tanto la forma de personalizar las citadas pantallas es personalizando los ficheros correspondientes, es decir, escribir en los ficheros **/etc/issue** y **/etc/issue.net** el texto que deseamos que aparezca en la pantalla de login.