# ADMINISTRACIÓN DE USUARIOS Y GRUPOS


IMPLANTACIÓN DE SISTEMAS OPERATIVOS
I.E.S. JACARANDÁ (BRENES,SEVILLA)
JOAQUÍN DELHOM VIANA

# Índice

# 1 Introducción

Buena parte de la administración de los sistemas GNU/Linux tiene que ver con la gestión de las tareas corrientes de cada día. Muchas de estas tareas están relacionadas con usuarios y grupos: añadirlos, eliminarlos, configurar sus entornos, etc. En un sistema pequeño, estas tareas se podrían realizar de vez en cuando, pero en un sistema con mucha actividad hay que ajustar las cuentas frecuentemente. En cualquier caso, debes saber como hacerlo.

Como administrador de sistemas, debes administrar a los usuarios de tu sistema. Tienes la opción de agregar o eliminar usuarios, además de añadir y eliminar grupos, y de modificar los derechos y permisos de usuarios y grupos. También tiene acceso a los archivos de inicialización predeterminados copiados en una cuenta de usuario cuando se crea por primera vez. Tienes la opción de decidir la manera en que se deben configurar inicialmente nuevas cuentas de usuario mediante la modificación de estos archivos.

## *1.1 Definiciones preliminares*

**USUARIO:** Persona que trabaja en el sistema (edita ficheros, ejecuta programas, etc.)

**PSEUDO-USUARIO:** Entidad que puede ejecutar programas o poseer ficheros (normalmente tienen identificadores de 0 a 100). En el sistema existen una serie de usuarios que no pueden iniciar sesión, pero que pueden poseer ficheros o ejecutar programas, por ejemplo el usuario www-data es el usuario propietario de los sitios web si tenemos un servidor web instalado en nuestro sistema.

## *1.2 Características de un usuario.*

Las características que definen a todo usuario de un sistema GNU/Linux son:

- Nombre de usuario, (también conocido como login, logname o username).
- Identificador de usuario (UID) → internamente el sistema identifica a un usuario por su UID, y no por su nombre. Esto es un número, sería como la matrícula del usuario, en principio este número es único.
- Grupos a los que pertenece (GID's) → todos los grupos a los que pertenece el usuario y que definirán los derechos que tiene el usuario dentro del sistema.

## *1.3 Ficheros de configuración*

A continuación se van a definir los ficheros más importantes relacionados con la configuración de usuarios, conocer estos ficheros es de vital importancia para poder solucionar rápidamente cualquier problema relacionado con gestión de usuarios. No te preocupes si te pierdes con alguno de ellos, ya que a lo largo del tema se irá haciendo referencia a ellos, aquí simplemente se expone su estructura y contenido para referencias futuras.

Los ficheros de configuración relacionados con la gestión de usuarios son los siguientes:

- **/etc/passwd** → Información de las cuentas de usuarios.

  Los permisos del fichero son rw_r__r__, es por esto por lo que las contraseñas se suelen guardar en el fichero **/etc/shadow** que tiene unos permisos más restrictivos y así nadie que

no sea root ni pertenezca al grupo root pueda ver las contraseñas.

Para editar el fichero lo podemos hacer con cualquier editor de texto (forma no segura) o con **/usr/sbin/vipw,** si utilizamos este último comando al terminar la edición hará una comprobación y nos avisará si hemos cometido algún error relacionado con el formato del archivo.

Cuando se agrega un usuario, una entrada (una nueva línea en el fichero) se crea en el archivo **/etc/passwd**. Cada entrada ocupa una línea que tiene varios campos separados por ":". Así cada línea, sería de la siguiente forma:

nombre:password:uid:gid:gecos:home:shell

- nombre → Nombre del usuario, login, logname o username.
- Password → Este campo define la contraseña, sus valores pueden ser:
  - ✔ Contraseña cifrada → Las shadow están desactivadas.
  - ✔ "*" o "!!" o "!" → La cuenta está desactivada o bloqueada.
  - ✔ "x" → Las shadow están activas, es decir, la contraseña cifrada se guarda en **/etc/shadow**
- uid → identificador del usuario. Es un número único para cada usuario.
- gid → identificador del grupo primario al que pertenece.
- Gecos → campo de información referente al usuario (nombre, telefóno, …) Si utilizamos herramientas automáticas (adduser o la interfaz gráfica) le asigna unos valores, pero podemos personalizar esto como queramos. Normalmente los diferentes campos se separan por ",".
- home → Ruta del directorio "HOME" del usuario.
- Shell → Intérprete de órdenes que se ejecutará al entrar al sistema. Ej. /bin/bash o /bin/sh

◆ **/etc/shadow** → Contraseñas encriptadas e información de "envejecimiento" de las cuentas.

Los permisos de este fichero son rw_r_____, por lo que solo el usuario root y aquellos que pertenezcan al grupo root podrán ver su contenido. Es por esto que se suelen guardar aquí las contraseñas en lugar de en el fichero **/etc/passwd**.

Este fichero guarda para cada una de las cuentas del sistema, la **contraseña encriptada** junto con su información de **envejecimiento.**

En el campo "password" del fichero **/etc/passwd** aparecerá una "x" indicando que las "shadow" están activas. Por defector están activas y se actualizan automáticamente.

Cada una de las líneas sigue este formato:

nombre:password:changed:minlife:maxlife:warn:inactive:expired:unused

- **nombre** → Nombre del usuario, evidentemente el usuario debe existir en **/etc/passwd.**
- **Password** → Contraseña encriptada. Antiguamente venían encriptadas con el algoritmo

MD5, actualmente suelen estar encriptadas con el algoritmo SHA1 que es más seguro.

- **Changed** → Fecha del último cambio de contraseña. Si intentamos leer directamente este dato en el fichero nos resultará un tanto extraño, ya que viene expresado en tiempo Unix, pero para cambiarlo utilizaremos herramientas que nos permiten utilizar fechas normales. Si este valor está a 0 quiere decir que tendremos que cambiar la contraseña en el próximo inicio de sesión.
- **Minlife** → Nº de días que han de pasar para poder cambiar la contraseña.
- **Maxlife** → Nº de días máximo que puede estar con la misma contraseña sin cambiarla.
- **Warn** → Cuántos días antes de que la contraseña expire (maxlife) será informado sobre ello el usuario, indicándole que tiene que cambiarla.
- **Inactive** → Nº de días después de que la contraseña expire que la cuenta se deshabilitará de forma automática si no ha sido cambiada.
- **Expired** → Fecha en la que la cuenta expira y se deshabilita de forma automática.

◆ **/etc/group** → Definición de los grupos y usuarios miembros de los mismos.

Las líneas de este fichero siguen el siguiente formato:

nombre:x:gid:lista de usuarios

- nombre → nombre del grupo
- gid → identificador del grupo
- lista de usuarios que pertenecen al grupo, separados por “,”

  Ejemplo: aso:x:519:pilar,alvaro,juan,eduardo,aso01,aso02,aso03

◆ **/etc/gshadow** → Contraseñas encriptadas de los grupos. Esto sirve para poner contraseñas a los grupos con la idea de que un usuario pueda utilizar un grupo sin pertenecer a él pero conociendo la contraseña.

◆ **/etc/login.defs** → Definiciones de cuentas de inicio de sesión predeterminadas para usuarios.

# 2 Gestión de usuarios

## *2.1 Añadir un nuevo usuario al sistema.*

Para entender mejor lo que hacen las herramientas que veremos en los puntos siguientes, vamos a definir a continuación todos los pasos que habría que seguir para dar de alta a un usuario de forma manual:

1. Decidir el nombre de usuario, el UID y los grupos a los que va a pertenecer (grupo primario y grupos secundarios).
2. Introducir los datos en los ficheros **/etc/passwd** y **/etc/group** (poniendo como contraseña “*” para bloquear la cuenta y que no se pueda utilizar hasta que terminemos el proceso).
3. Asignar un password a la nueva cuenta. Hacer esto manualmente es algo complejo, ya que

habría que encriptar la contraseña e introducirla en el fichero **/etc/shadow**, por lo que siempre lo haremos con el comando **passwd** que veremos más adelante.

4. Establecer los parámetros de "envejecimiento" de la cuenta en el fichero **/etc/shadow**.
5. Crear el directorio "HOME" del nuevo usuario y establecer el propietario y grupo correspondientes y los permisos adecuados. Esto es importante y suele quedar en el olvido, ten en cuenta que sólo podemos crear un directorio en /home con el usuario root, por lo que si no cambiamos nada el nuevo usuario no tendrá permisos sobre el directorio creado.
6. Copiar los ficheros de inicialización (.bash_profile, .bashrc, …) que veremos más adelante y se encuentran en el directorio **/etc/skel**.
7. Establecer otras configuraciones: quotas, mail, permisos para imprimir,etc. De esto solo veremos este año las quotas en el tema de sistemas de ficheros.
8. Probar la nueva cuenta. Este paso también es muy importante y se suele olvidar. Un buen administrador siempre comprobará antes las cosas para evitar fallos innecesarios.

## *2.2 Contraseñas*

Para cambiar la contraseña de un usuario o establecer una nueva si no la tenía, utilizaremos el comando **passwd**:

**passwd [nombre de usuario]** → asignar o cambiar contraseña a un usuario. Si lo utilizamos sin parámetros entiende que queremos cambiar la contraseña del usuario que ejecuta el comando. Cada usuario podrá cambiar su contraseña, mientras que el usuario root podrá asignar o cambiar la de cualquier usuario.

La elección de las contraseñas es un tema más importante de lo que parece, tener en cuenta que una mala contraseña puede comprometer la seguridad de todo el sistema. Como anécdota os contaré que algunas canciones del cantante David Bisbal vieron la luz antes del tiempo porque tenía como contraseña de su correo electrónico "almeria" y una fan fue haciendo pruebas y la averiguó.

Consejos para elegir una buena contraseña:

- No utilizar:
  - Tu nombre o parte de él, o de alguien cercano a ti.
  - Números significativos para ti o alguien cercano a ti.
  - Algún nombre, número, lugar, gente, etc., asociado a tu trabajo.
  - Palabras que estén en un diccionario.
  - Nombre de gente famosa, lugares, películas, relacionadas con publicidad, etc.
- Consejos:
  - Introducir 2 o más caracteres extras, símbolos especiales o de control.
  - Escribir mal las palabras.
  - Utilizar mayúsculas y minúsculas, pero no de forma evidente, es decir no poner la primera mayúscula y el resto en minúsculas.

- Concatenar, embeber o mezclar 2 o más palabras, o partes de palabras.
- Usar caracteres poco comunes, como por ejemplo $,&,#,^.
- Utilizar una frase fácil de recordar y coger sólo la primera letra de cada palabra, como por ejemplo: Mi perro se llama Pancho → MpslP. Es fácil de recordar pero difícil de averiguar, sobretodo si le añadimos números y símbolos.

Las contraseñas se deben cambiar cuando:

- Se sospecha que alguien la ha podido conocer o averiguar.
- Un usuario se marcha del trabajo. En este caso hay que cambiar todas las contraseñas que conozca.
- Un administrador del sistema se va. En este caso hay que cambiar TODAS las contraseñas.
- Se despide a un usuario o a un administrador.
- Se sospecha que alguien ha conseguido el fichero con las contraseñas.
- Un intruso ha conseguido entrar en el sistema.
- Periódicamente, se debe forzar a que los usuarios cambien sus contraseñas, incluido el administrador. Aquí hay que encontrar un punto justo, ya que si obligas a los usuarios a cambiar las contraseñas muy a menudo, acabarán utilizando contraseñas inseguras y si nunca se cambian es muy fácil que alguien acabe averiguando alguna.

**REALIZAR EL BOLETÍN DE EJERCICIOS 1 DE ADMINISTRACIÓN DE USUARIOS**

## *2.3 Herramientas para crear/modificar cuentas de usuario*

Las herramientas automáticas para la creación de cuentas de usuario suelen realizar todas las tareas básicas del proceso, a excepción de las específicas (quotas o impresión, etc.). Nosotros como administradores de sistemas evitaremos las excesivamente automáticas que no nos permiten la configuración de algunos detalles. A continuación vamos a ir viéndolas una a una:

- **adduser** → esta herramienta nos realiza una serie de preguntas y crea de forma automática el usuario, está es una de la que nosotros no utilizaremos habitualmente, ya que no nos permite modifar algunos parámetros que nos interesan. Toma los valores por defecto de /etc/default/useradd y de /etc/login.defs
- **useradd** → Esta será la herramienta que más utilizaremos nosotros, para conocerla a fondo deberéis utilizar el comando **man**, pero aquí vamos a definir los parámetros más habituales.
  - ✔ -b directorio_base → Por defecto el home de los usuarios se crea en /home, con esta opción podemos decir que el home del usuario se cree en otro directorio. Por ejemplo, si utilizamos *useradd -b /usuarios pepito* se crearía el usuario pepito y su home sería /usuarios/pepito. NOTA: el directorio base indicado debe existir previamente. El directorio del usuario sólo se creará si se utiliza también la opción -m.
  - ✔ -c comentario → Todo lo que pongamos en comentario se añadirá al campo gecos del fichero /etc/passwd del usuario correspondiente. Si utilizamos espacios el comentario

deberá ir entre comillas.

- ✔ -d directorio_home → Con esta opción podemos especificar la ruta completa del home del usuario si no queremos que sea el que se elige por defecto. Por ejemplo, si utilizamos *useradd -d /empleados/contabilidad/jefe pepito*, se creará el usuario pepito y su directorio home será /empleados/contabilidad/jefe. NOTA: el comando useradd solo intenta crear el último directorio de la ruta, por lo que todo lo anterior debe existir previamente. Igual que en la opción -b, el home solo se creará si también se especifica la opción -m.
- ✔ -e Expire_date → la cuenta expirará en la fecha indicada por expire_date. El formato de la fecha será YYYY-MM-DD (2012-10-31 sería 31 de octubre de 2012)
- ✔ -g grupo → el usuario tendrá como grupo principal el grupo indicado en grupo. Se puede indicar tanto mediante el gid como mediante el nombre del grupo. Si esta opción no se indica el sistema crea un grupo con el mismo nombre que el usuario y ese es el grupo principal del usuario.
- ✔ -G grupo1[,grupo2,....[,grupoN]] → Con esta opción podemos indicar los grupos secundarios a los que pertenezca el usuario, podremos indicar tantos como queramos separados por ",". Obviamente los grupos deberán estar creados previamente.
- ✔ -m → Esta opción crea el home del usuario si no existe. Los ficheros y directorios que existan en el directorio /etc/skel serán copiados al home del usuario. NOTA: sin esta opción, pongamos lo que pongamos NO se creará el directorio home del usuario.
- ✔ -o → Por defecto linux no permite la creación de más de un usuario con el mismo uid. Con esta opción nos permitirá crear dos o más usuarios con el mismo uid. Evidentemente carece de sentido si no utilizamos la opción -u.
- ✔ -p contraseña → Establece la contraseña del usuario a lo indicado en contraseña, pero OJO el parámetro espera que la contraseña esté encriptada, por lo que si no la hemos encriptado previamente, no utilizaremos este parámetro. Normalmente estableceremos la contraseña con el comando passwd.
- ✔ -s shell → Nos permite establecer la shell que utilizará el usuario. Por defecto la shell establecida es /bin/sh, pero nosotros habitualmente utilizaremos /bin/bash que mejora la anterior. De hecho la herramienta gráfica de creación de usuarios establece /bin/bash como shell a los nuevos usuarios.
- ✔ -u uid → nos permite establecer el uid del usuario, si no ponemos nada el sistema nos asignará el siguiente disponible. Normalmente en los sistemas linux se utilizan para los usuarios uid's a partir del 1000, reservando para cuentas del sistema los anteriores. El uid del usuario root es 0.

- **usermod** → La herramienta anterior nos permite crear usuarios con unas determinadas características, esta en cambio nos permite modificar las características de cuentas de usuario ya creadas. Entendiendo las opciones de la herramienta anterior y con la ayuda del comando **man** os resultará muy sencillo descubrir las opciones de esta.
- **userdel** → eliminar cuentas de usuarios (por defecto no borra el directorio HOME del usuario, buscar en el manual la opción que os permite borrarlo).

- **newusers** → crea cuentas de usuarios utilizando la información introducida en un fichero de texto, que ha de tener el formato del fichero **/etc/passwd** (¡OJO! No copia los ficheros de inicialización. Ejemplo:
  - ✔ Creamos un fichero llamado "usuarios" (podría llamarse de cualquier otra forma) con las siguientes líneas:

    usuario1:contraseña1:1500:1000:Usuario Apellido1 Apellido2:/home/usuario1:/bin/bash

    usuario2:contraseña2:1501:1000:Usuario Apellido1 Apellido2:/home/usuario2:/bin/bash
  - ✔ A continuación ejecutamos: *newusers usuarios*
  - ✔ Esto crearía el usuario **usuario1** con la contraseña **contraseña1** con uid **1500** gid **1000** en el campo gecos **Usuario Apellido1 Apellido2** crearía el home **/home/usuario1** y le asignaría al usuario la shell **/bin/bash.** Y lo mismo para el usuario **usuario2** pero con los datos detallados en su línea. Si hubiera 50 líneas bien hechas crearía los 50 usuarios.
  - ✔ Este comando es útil para copiar usuarios de otro sistema.
- **users-admin:** esto lanzaría la herramienta gráfica de creación de usuarios. También la podemos lanzar desde la interfaz gráfica.

## 2.4 Restricciones de tiempo

Como hemos visto anteriormente, para las cuentas de los usuarios se pueden establecer restricciones de tiempo o envejecimiento respecto a la validez de la cuenta o de la contraseña. Los valores se guardan en el fichero **/etc/shadow** de la forma que hemos definido en puntos anteriores.

Los valores los establece el administrador con las órdenes **chage** (del inglés change age) o con **passwd** (sí, el mismo que nos sirve para establecer las contraseñas.

El fichero **/etc/login.defs** tiene los valores por defecto.

La utilización del comando **chage** es:

chage [parámetros] [nombre_de_usuario]

Los parámetros que podemos utilizar con el comando **chage** son:

- -d ult_día → fecha del último cambio de contraseña.
- -m min_días → establece el número de días que han de pasar para poder cambiar la contraseña al número indicado en min_días.
- -M max_días → establece el número máximo de días que puede estar con la misma contraseña sin cambiarla al número indicado en max_días.
- -W warn_días → establece el número días antes de que la contraseña expire que será avisado el usuario al número indicado en warn_días.
- -I inac_días → establece el número de días después de que la contraseña expire que la cuenta se deshabilitará de forma automática si la contraseña no ha sido cambiada a inac_días.
- -E fecha → se establece la fecha en la que la cuenta expirará. La fecha se indicará en el formato YYYY-MM-DD.

- EJEMPLO: supongamos que el usuario **joaquín** cambia su contraeña el 1 de marzo, y el administrador ejecuta la siguiente órden:
  - chage -M 20 -W 6 -I 5 -E 2012-11-12 joaquín
  - Los tiempos quedarían fijados de la siguiente manera:
    - El 14 de marzo joaquín recibirá el primer aviso para que cambie su contraseña.
    - El 20 de marzo joaquín debería haber cambiado su contraseña. Si no la cambia, como se ha fijado el tiempo de inactividad, la cuenta aún no se bloqueará.
    - Si el 25 de marzo joaquín no ha cambiado su contraseña, la cuenta será bloqueada.
    - La cuenta expira, y por tanto se bloqueará, el 12 de noviembre de 2012.

## *2.5 Ficheros de inicialización*

Cada vez que un usuario inicia sesión, se crean dos secuencias de comandos de perfil (se ejecutan una serie de comandos para inicializar algunas configuraciones), una de sistema, que es la misma para cada usuario y una de inicio de sesión que se personaliza de acuerdo con las necesidades de cada usuario. De la misma forma, cuando el usuario sale de su sesión, se ejecuta una secuencia de comandos de salida de sesión de usuario. Además, cada vez que se genera una shell (cada vez que abrimos un terminal), incluida la shell de inicio de sesión, se ejecuta una secuencia de comandos de shell de usuario. Existen diferentes tipos de secuencias de comandos utilizadas para diferentes shells. La shell que se utiliza de manera más común es la shell BASH, aunque existen otras como SH, TCSH, Z, etc. Las diferencias entre ellas son pequeñas y no merece la pena detallarla ahora mismo.

Las secuencias de comandos mencionadas en el anterior párrafo se guardan en unos ficheros ocultos que deben encontrarse en el home de cada usuario. Estos ficheros son:

- Se ejecuta al hacer login, establece entre otras cosas la variable PATH, las variables de entorno, umask y funciones de inicialización. Cualquier comando que añadamos a este fichero, se ejecutará al hacer login. Según la shell que utilicemos tenemos:
  - **.bash_profile** → para la shell Bourne Again Shell (bash)
  - **.profile** → para Bourne Shell (sh)
  - **.login** → para C Shell (csh)
- Cada vez que se ejecuta un shell. Cualquier comando que añadamos a este fichero, se ejecutará cada vez que iniciemos una shell (recuerda que al iniciar el sistema también se lanza una shell, incluso aunque utilicemos el entorno gráfico). Según la shell que utilicemos tenemos:
  - **.bashrc** → en Bourne Again Shell (bash)
  - **.cshrc** → en C Shell (csh)
- Al salir del sistema el usuario (al finalizar sesión):
  - **.bash_logout** → en Bourne Again Shell (bash)

- **.logout** → en C Shell (csh)

Estos ficheros se guardan en el directorio **/etc/skel** y se copian cuando se crea una cuenta al directorio HOME asignado al usuario (obviamente esto solo se hará si hemos utilizado una herramienta que cree el home del usuario). Por lo tanto si queremos cambiar una configuración de inicio para todos los usuarios que creemos a partir de la modificación, solo tendremos que modificar estos ficheros en el directorio **/etc/skel**. Tener muy en cuenta que las modificaciones que hagamos en el fichero **/etc/skel** sólo se aplicarán a los usuarios que creemos tras las modificaciones, en ningún momento esto afectará a los usuarios ya creados. De la misma forma, si quisiéramos que se creara en el home de todos los usuarios que creemos un directorio o un fichero, solo tendríamos que crearlo dentro del directorio **/etc/skel**.

## 2.6 Asignar o cambiar de intérprete de órdenes

Como definimos anteriormente, en el último campo del fichero **/etc/passwd** se establece el intérprete de órdenes (shell) que se ejecuta al entrar al sistema para un usuario. Evidentemente, se puede cambiar en cualquier momento la shell que utiliza un usuario, lo podríamos hacer manualmente modificando el fichero /etc/passwd, aunque disponemoss de herramientas que nos permiten evitar errores de edición del fichero como **chsh** (change shell) o **usermod**. Ahora bien, hay que tener en cuenta que por defecto ni todas las shells posibles están instaladas ni todas están permitidas, la lista de shell permitidas en el sistema se indican en **/etc/shells (¡**Ojo**!** Si se prohíbe un shell, no se podrá elegir con **chsh**, pero los usuarios que ya lo tenían asignado lo seguirán usando sin problemas). Cualquier usuario puede utilizar **chsh** para cambiar la shell que utiliza por una de las permitidas en el fichero **/etc/shells** y evidentemente el usuario root puede cambiar la de cualquier usuario.

Si quisiéramos prohibir una shell podemos eliminar la línea que contiene dicha shell o simplemente añadirle delante el símbolo # (este símbolo en la mayoría de ficheros de configuración de GNU/Linux indica que lo que hay a continuación es un comentario y no lo tiene en cuenta) en el fichero **/etc/shells**.

Si se desea que un usuario no pueda entrar al sistema se le puede asignar **/bin/false** o **/sbin/nologin**. Esto es de gran utilidad para configurar usuarios que son necesarios para otros servicios (como SAMBA, apache, ftp, etc) pero no deben poder iniciar sesión en el sistema.

También podemos establecer como shell de un usuario un fichero ejecutable, cuando dicho usuario entre al sistemas se ejecuta (el comando que hayamos puesto como shell), y al finalizar la ejecución, el usuario sale del sistema (realmente no llega a hacer login).

## 2.7 Cuentas restrictivas

Las cuentas restrictivas permiten limitar las acciones que los usuarios pueden realizar en el sistema, haciendo que tengan determinadas restricciones. Se pueden crear de dos formas:

- Asignando como shell un fichero ejecutable que realiza una tarea determinada y al terminarla el usuario sale del sistema, por ejemplo:
  - Usuario para realizar las copias de seguridad. Como shell le pondríamos un script (pequeño programa de comandos GNU/Linux, lo veremos en segundo) que realiza esta tarea.

    - Usuario para que se apague el sistema, le pondríamos como shell halt por ejemplo. Ojo, el usuario debe tener permisos para ejecutar la tarea asignada, por ejemplo, en este caso solo el usuario root puede apagar el sistema.

- Usando el shell restrictivo **/bin/rbash**. Realmente esta shell es un enlace simbólico (blando) a **/bin/bash**. Este intérprete se comporta como un intérprete normal, salvo que el usuario no puede hacer determinadas tareas, como:
    - Cambiar de directorio.
    - Establecer o modificar los valores de las variables SHELL o PATH.
    - Especificar nombre de órdenes que contengan /.
    - Usar la redirección.
    - Utilizar la orden interna **exec** para reemplazar el shell por otro programa

    A estos usuarios se les tiene que limitar los ficheros que pueden ejecutar, copiándolo a un directorio y que su **PATH** sea sólo ese directorio. En otro caso, con un **PATH** "normal", es casi como si no tuviesen restricciones.

    Recordar que la variable PATH es la que contiene todos los directorios donde el sistema buscará los comandos ejecutables.

# 3 Grupos

Linux utiliza los grupos como medios para organizar los usuarios. En muchos sentidos, los grupos hacen similares entre sí a sus usuarios. En particular, están definidos en ficheros de configuración similares, tienen nombres similares a nombres de usuarios y se representan internamente mediante números (puesto que son cuentas). Sin embargo, los grupos no son cuentas, sino medios para organizar colecciones de cuentas, en gran parte como medida de seguridad. Cada fichero de un sistema Linux está asociado con un grupo específico, pudiéndose asignar varios permisos a los miembros de cada grupo. Por ejemplo, puede que los miembros del grupo (como una facultad de una universidad) tengan permiso para leer un fichero, pero puede que a otros (como los estudiantes) no se les permita dicho acceso. Como Linux proporciona acceso a la mayoría de los dispositivos de hardware (como puertos serie, unidades externas, etc) a través de ficheros, puede utilizar también este mismo mecanismo para controlar el acceso hardware.

Cada grupo puede tener desde ningún usuario hasta tantos como tenga el sistema. La pertenencia a los grupos se controla mediante el fichero **/etc/group**. Además, cada usuario posee un grupo por defecto o primario. El grupo primario del usuario se define en la configuración del usuario del fichero **/etc/passwd**. Cuando los usuarios acceden al sistema, se define su pertenencia a su grupo primario. Cuando los usuarios crean ficheros o inician programas, esos ficheros y programas en ejecución se asocian con un único grupo, el grupo actual. Un usuario puede acceder a fichero que pertenecen a otros grupos siempre que el usuario pertenezca a dicho grupo y que ello esté permitido por los permisos de acceso del grupo. Sin embargo, para ejecutar programas o crear ficheros con un grupo distinto del primario, el usuario deberá ejecutar el comando **newgrp** para cambiar la pertenencia al grupo actual. Esto lo podrá hacer si también pertenece al grupo al que se quiere cambiar o si el grupo tiene asignada una contraseña y el usuario la conoce. Este cambio es temporal

y sólo se mantiene mientras no cerremos la shell actual.

### 3.1 Características de un grupo

- Nombre del grupo o groupname
- Identificador del grupo (GID) → internamente el sistema identifica al grupo por este número.

### 3.2 Tipos de grupos

- **Primario** → el grupo especificado en el fichero **/etc/passwd**. Como se explica en el punto 3 todos los ficheros y directorios que cree el usuario, así como los programas que ejecute serán con este grupo efectivo.
- **Secundarios** → los otros grupos a los que pertenece el usuario, se indican en /**etc/group**

### 3.3 Herramientas para la gestión de grupos

- **groupadd** grupo → crea un nuevo grupo, se pueden ver sus opciones que son muy sencillas con el comando **man**.
- **groupmod** grupo → permite modificar un grupo existente.
- **groupdel** grupo → eliminar un grupo existente.
- **gpasswd** grupo → asignar una contraseña a un grupo. Si un grupo tiene contraseña, un usuario que la conozca podrá trabajar con ese grupo, a pesar de no pertenecer a él. Al ejecutar la orden **newgrp** grupo, introducirá la contraseña y pasará a ser su grupo primario de forma temporal (hasta que cambie de shell o vuelva a ejecutar newgrp con otro grupo).
- **gpasswd** -a user grupo → permite añadir el usuario user al grupo grupo.
- **groups** [usuario] → grupos a los que pertenece un usuario. Si lo ejecutamos sin parámetros, muestra los grupos a los que pertenece el usuario que ha ejecutado el comando.
- **id** [usuario] → lista el identificador del usuario y los grupos a los que pertenece. Sin parámetros lo mostrará para el usuario que ejecuta el comando.

**REALIZAR EL BOLETÍN DE EJERCICIOS 2 DE ADMINISTRACIÓN DE USUARIOS Y GRUPOS.**

# 4 Bibliografía

**Libros:**

Linux. Sexta edición. Richard Petersen. Ed. McGrawHill

Linux Profesional Institute Certification. Guía de estudio – Exámenes 101 y 102.

**Otros:**

Transparencias de la asignatura Administración de Sistemas Operativos de la Universidad de Murcia de la profesora Mª Pilar González Férez. http://www.ditec.um.es/aso/