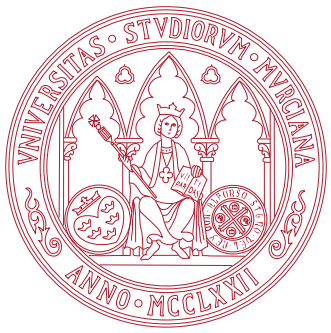

Universidad de Murcia

Facultad de Informática

Departamento de Ingeniería y Tecnología de Computadores

Área de Arquitectura y Tecnología de Computadores

PRÁCTICAS DE I.S.O.

2º de Grado en Ingeniería Informática

Boletín de Prácticas 5 – Gestión de usuarios en Linux

Curso 2017/2018

Índice

1. Introducción

El objetivo de esta práctica es trabajar con la gestión de los usuarios. Así vamos a crear y configurar cuentas de usuarios utilizando órdenes en modo texto. A pesar de la existencia de herramientas gráficas para estos propósitos, la utilización de estas órdenes es preferida por los administradores de sistema dada la mayor rapidez que ofrecen a la hora de gestionar determinadas situaciones (por ejemplo cuando hay que crear muchos usuarios a la vez). Asimismo, es interesante que veas la importancia de los siguientes puntos:

- UID y GID's de la cuenta, y forma de asignación.
- La contraseña y su política de envejecimiento, longitud, etc.
- Los ficheros asociados al directorio `$HOME`.
- Los ficheros relacionados con la gestión de usuarios.

2. Objetivos

Al terminar este boletín deberás ser capaz de:

- Crear cuentas de usuario.
- Controlar qué usuarios hay conectados al sistema y qué labores están realizando.
- Modificar parámetros de una cuenta de usuario: asignar nuevos grupos, cambiar el intérprete de órdenes asignado, etc.
- Gestionar las restricciones de tiempo asignadas a la contraseña y a la cuenta.
- Crear cuentas de usuario restrictivas y conocer su utilidad y aplicación.
- Crear y modificar grupos de usuario.
- Distinguir entre grupo primario, grupos secundarios y grupo activo.
- Eliminar cuentas de usuario y de grupo.

3. Órdenes utilizadas

A lo largo de este boletín, algunas de las órdenes que vamos a ver son:

- `su`: cambio de usuario.
- `who`: listar usuarios contectados.
- `useradd`, `usermod` y `userdel`: creación, modificación y eliminación de cuentas de usuario.
- `newusers`: crea nuevos usuarios, indicados estos en un fichero con el mismo formato que el fichero `/etc/passwd`.
- `passwd`: asigna o cambia la contraseña de un usuario.
- `chfn`: cambia la información `GECOS` de un usuario.
- `finger`: muestra la información de un usuario.
- `chsh`: cambia el shell asignado a un usuario.
- `chage`: asigna los valores de envejecimiento de la contraseña y la cuenta de un usuario.

- `groupadd`, `groupmod` y `groupdel`: creación, modificación y eliminación de grupos.
- `id` y `groups`: identificación del usuario y grupos a los que pertenece.
- `newgrp`: cambia de grupo activo.

4. Gestión de usuarios

Las cuentas de usuario son un punto importante en la administración de un sistema pues permiten que los usuarios accedan al sistema y puedan trabajar. Tareas para la gestión de usuarios:

- Añadir/Eliminar cuentas de usuario al sistema.
- Modificar/Controlar propiedades de las cuentas de usuario.
- Crear/eliminar/modificar grupos de usuarios.

4.1. Definición de usuario

Un usuario es una entidad que ejecuta programas o posee ficheros. Normalmente es una persona que trabaja en el sistema (entra al sistema, edita ficheros, ejecuta programas, etc.), pero también puede ser un pseudo-usuario, como veremos al final del boletín.

4.2. Características de un usuario

Las características principales de un usuario son:

- Nombre del usuario (logname o username): nos permite identificarlo de una forma sencilla.
- Identificador de usuario (UID): es un número que identifica al usuario. Internamente el sistema usa el UID para identificar al usuario, y no su nombre.
- Grupos a los que pertenece (GUIDs). Como veremos más delante en este boletín, un grupo está formado por la colección de usuarios que comparten una función similar.

Tal como se vió en clase de teoría, el UID de un usuario y los grupos a los que pertenece marcarán los permisos que tiene este usuario para acceder a ficheros o recursos.

4.3. El superusuario

En sistemas operativos del tipo Unix, el usuario `root`, también llamado *supersusuario*, es el nombre convencional de la cuenta de usuario que posee todos los derechos. El usuario `root` puede hacer muchas cosas que están vedadas a un usuario común, tales como hacer modificaciones en la configuración del sistema, instalar programas, modificar configuraciones de servidores, administrar usuarios, etc. El superusuario siempre tiene como UID el valor 0.

No es recomendable utilizar el usuario `root` para una simple sesión de uso habitual, ya que pone en riesgo el sistema al garantizar acceso privilegiado a cada programa en ejecución. Es preferible utilizar una cuenta de usuario normal y utilizar la orden `su` para acceder a los privilegios de `root` en caso de ser necesario, como veremos en la siguiente sección.

4.4. La orden `who`

La orden `who` muestra un listado de los usuarios con sesiones abiertas. Por ejemplo:

```
$ who
jcamara  pts/0        2017-09-18 16:59 (77.210.244.55)
jgines   pts/3        2017-09-18 18:17 (185.154.58.181)
javiercm pts/4        2017-09-18 19:25 (132.red-79-154-79.dynamicip.rima-tde.net
```

En este ejemplo, vemos como hay 3 usuarios conectados: el usuario `jcamara` está conectado a la terminal `pts/0`, `jgines` a la `pts/3` y `javiercm` a la `pts/4`. Igualmente, podemos ver la fecha y hora de inicio de cada conexión, así como la dirección IP desde donde se conectaron.

4.5. Las órdenes `id` y `groups`

La orden `id` muestra el UID del usuario cuyo nombre se le pasa como parámetro, así como la relación de grupos a los que pertenece. Por ejemplo:

```
$ id javiercm
uid=503(javiercm) gid=501(profesor) grupos=501(profesor),526(webmaster)
```

En este ejemplo, vemos cómo el usuario de nombre `javiercm` tiene UID=503. Este usuario pertenece a dos grupos: `profesor`, con GID=501, y `webmaster`, con GID=526; siendo `profesor` su grupo activo en este momento.

La orden `groups` muestra simplemente la relación de grupos a los que pertenece el usuario cuyo nombre se le pasa como parámetro. Por ejemplo:

```
$ groups javiercm
javiercm : profesor webmaster
```

Tanto en la orden `id` como en la orden `groups`, si no se indica ningún usuario ni grupo, respectivamente, se mostrará la información correspondiente al usuario que ejecuta las órdenes.

Al final del boletín, en la gestión de grupos, veremos todos estos conceptos con más detalle.

4.6. La orden `su`

La orden `su` ejecuta un nuevo shell, con los identificadores de usuario (UID) y grupo (GID) del nuevo usuario indicado, de tal manera que podremos trabajar en el sistema como si efectivamente fuésemos dicho usuario. Se operará con estos nuevos identificadores hasta salir con la orden `exit` del shell que se ha lanzado. Por defecto, se utilizará el usuario `root`.

Algunas opciones de la orden `su` son:

- Si se ejecuta sin opciones, el valor de la mayoría de las variables de entorno no varía, excepto las variables `HOME` y `SHELL`. En caso de haber cambiado a un usuario distinto del root, también cambian las variables `USER` y `LOGNAME`. Por ejemplo:

  ```
  login: javiercm
  password: ****

  $ su
  password: ****
  # echo $USER
  javiercm
  # echo $LOGNAME
  javiercm
  # exit
  ```

- -l : se cambian todas las variables de entorno, como si se hubiera hecho un login de nuevo. (En algunas versiones esta opción ha sido eliminada y han dejado sólo la siguiente). Por ejemplo:

```
login: javiercm
password: ****

$ su -l
password: ****
# echo $USER
root
# echo $LOGNAME
root
```

- - : equivalente a -l.

El superusuario puede usar esta orden para actuar fácilmente como otro usuario y así comprobar la configuración de su cuenta, resolver algún problema que pueda tener el usuario, etc.

Cuando esta orden la ejecuta un usuario normal, se le solicitará la contraseña del usuario al que quiere cambiar.

4.7. La orden **whoami**

La orden `whoami` muestra el nombre del usuario efectivo que realmente soy en cada instante, o sea, el que corresponde con EUID, tal como se vió en teoría. En el siguiente ejemplo podemos apreciar, usando `whoami`, cómo tiene lugar un cambio de usuario efectivo tras utilizar la orden `su`. Por otro lado, la orden `who` siempre nos muestra el nombre del usuario que inició la sesión en la máquina:

```
login: javiercm
Password: ********

$ who
javiercm pts/3        2017-09-20 09:39 (eduroam_um-6-228.inf.um.es)

$ whoami
javiercm

$ su -l alumno
Password: ********

# who
javiercm pts/3        2017-09-20 09:39 (eduroam_um-6-228.inf.um.es)

# whoami
alumno
```

4.8. Herramientas de gestión de usuarios

A continuación se describen las órdenes más importantes que se utilizan para la creación, modificación y borrado de cuentas de usuarios:

- `useradd`: para crear nuevas cuentas de usuario. Su sintaxis es:

```
useradd [opciones] usuario
```

Las opciones más importantes de esta orden son:

- `-d HOME_DIR`: El nuevo usuario se creará utilizando `HOME_DIR` como su directorio personal. Por defecto, si no se indica esta opción, se creará un directorio personal usando el nombre del usuario.
- `-g GROUP`: El grupo principal del usuario será `GROUP`. Si no se especifica esta opción, dependiendo de la configuración establecida por defecto en el sistema, o bien el grupo principal del usuario será un nuevo grupo con el mismo nombre del usuario, o bien, será un grupo por defecto.
- `-G GROUP [,GROUP2,...[,GROUPN]]`: Para indicar una lista de grupos secundarios a los que también pertenece el usuario.
- `-m`: Para forzar la creación del directorio personal, en caso de que éste no exista previamente. Si no se especifica esta opción, la creación o no del directorio personal dependerá de la configuración establecida por defecto en el sistema.

A la hora de establecer para una nueva cuenta sus características básicas iniciales (el shell por defecto que va a tener el usuario, si se le va a crear un directorio personal y donde se crearía en tal caso, las restricciones de tiempo de la contraseña, etc.) esta orden utiliza los valores por defecto que se encuentran en los ficheros `/etc/default/useradd` y `/etc/login.defs`. Estos valores se pueden modificar utilizando esta misma orden con la opción `-D`, junto a la opción correspondiente al valor por defecto que queremos modificar (ver página del manual para más información).

- `usermod`: para modificar las características de una cuenta de usuario ya existente. Su sintaxis es:

```
usermod [opciones] usuario
```

Las opciones más importantes de esta orden son:

- `-g GROUP`: El grupo principal del usuario será ahora `GROUP`. Este grupo debe existir previamente.
- `-G GROUP [,GROUP2,...[,GROUPN]]`: Para modificar la lista de grupos secundarios a los que también pertenece el usuario.
- `-s SHELL`: El shell inicial del usuario será ahora `SHELL`.

- `userdel`: para eliminar cuentas (por defecto no borra el directorio personal).
- `newusers`: para crear las cuentas de usuarios indicadas en un fichero de texto, cuyo nombre se pasa como parámetro, que tiene el mismo formato que `/etc/passwd`.

4.9. Ficheros de configuración

La información para las cuentas de usuario se guarda en varios ficheros de configuración:

- `/etc/passwd`: Información sobre las cuentas de los usuarios definidos en el sistema.
- `/etc/shadow`: Contraseñas encriptadas e información de envejecimiento de las cuentas de usuario.
- `/etc/group`: Información sobre los grupos definidos y los usuarios miembros de los mismos.
- `/etc/gshadow`: Contraseñas encriptadas asignadas a los grupos.

4.9.1. Fichero `/etc/passwd`

Este fichero contiene la lista de los usuarios definidos en el sistema. Algunas de sus características más importantes:

- Lo usan muchos procesos para obtener la información de los usuarios. Por ejemplo, lo consultan desde el proceso de entrada hasta la orden `ls`.
- El usuario propietario es el `root` y el grupo `root`.
- Sus permisos son `rw-r--r--`. Estos permisos no se deben cambiar, por seguridad.
- Con la orden `pwck` podemos verificar la integridad de los ficheros `/etc/passwd` y `/etc/shadow`.

El formato de cada una de sus líneas es:
`nombre:password:uid:gid:gecos:home:shell`

- `nombre`

  Es el nombre asignado al usuario (logname o username) y nos permite que sepamos quién es cada uno.

- `password`

  Tradicionalmente este campo guardaba la contraseña encriptada, pero en la actualidad guarda una `x` indicando que la contraseña se almacena en el fichero `/etc/shadow` por mayor seguridad.

- `uid`

  Es el identificador asignado al usuario. Para los pseudo-usuarios se reserva desde el 1 al 999, mientras que los valores a partir de 1000 se utilizan para usuarios normales, y el UID 0 está reservado para el usuario root. En el caso de que dos usuarios compartan el mismo UID, serán considerados el mismo para el sistema. Esta compartición sólo debería usarse en casos muy concretos, y con precaución. Por ejemplo, puede ser útil para tener a dos usuarios, con nombre y clave de acceso diferente, como administradores del sistema (UID=0).

- `gid`

  Es el identificador del grupo primario al que pertenece el usuario.

- `gecos`

  Contiene diversa información que identifica al usuario: nombre completo, teléfono, despacho, etc. Podemos usar la orden `chfn` para cambiar esta información

- `home`

  Contiene el path del directorio de trabajo del usuario. Cuando el usuario se autentica y entra al sistema, este es el directorio donde se le posiciona. En principio, es el único directorio donde el usuario podrá crear sus ficheros. El propietario ha de ser el usuario y el grupo propietario suele ser el grupo primario del usuario, aunque esto último se podría cambiar.

- `shell`

  Contiene la ruta del intérprete de órdenes usado para ese usuario. Se ejecuta automáticamente cuando el usuario entra al sistema. El valor de este campo se puede cambiar con la orden `chsh`.

Un ejemplo de línea de este fichero sería:
`pilar:x:1008:100:Pilar Gonzalez Ferez,3.45:/home/pilar:/bin/bash`

4.9.2. Fichero `/etc/shadow`

Las contraseñas encriptadas de los usuarios se guardan en el fichero `/etc/shadow` y no en el `/etc/passwd` debido a que el fichero `/etc/shadow` es más seguro que el `/etc/passwd`, pues sus permisos son `----------`, siendo el usuario propietario el `root` y el grupo propietario el `root`.

Este fichero guarda, para cada usuario, además de su contraseña encriptada, la información de envejecimiento de su cuenta y contraseña. El formato del fichero es:

```
nom:pass:changed:minlife:maxlife:warn:inactive:expired:unused
```

En las dos siguientes secciones veremos con detalle cómo se gestiona el contenido de este fichero.

4.10. Contraseña de los usuarios

En el campo `pass` del fichero `/etc/shadow` puede haber una de las siguientes opciones:

- Una contraseña encriptada.
- '*' o '!!': La cuenta está bloqueada y no se puede usar.

Cuando se crea un usuario nuevo, por ejemplo con la orden `useradd`, sin asignarle contraseña, su cuenta se encontrará bloqueada inicialmente, con lo que este usuario no podrá entrar al sistema. Sin embargo, aunque una cuenta esté bloqueada, sí puede ejecutar procesos que haya programado con herramientas de planificación. Para cambiar la contraseña de una cuenta, o asignársela inicialmente, se utiliza la orden `passwd` de esta manera:

- `passwd <nombre_usuario>`: la utiliza el `root` para asignar/cambiar la contraseña a un usuario.
- `passwd -e <nombre_usuario>`: la utiliza el `root` para obligar al usuario a cambiar su contraseña la próxima vez que entre al sistema.
- `passwd`: la utiliza cualquier usuario para cambiar su contraseña.

La herramienta `passwd` hace un chequeo, previamente a realizar el cambio efectivo de contraseña, para asegurar que la contraseña elegida cumpla ciertos parámetros de seguridad (longitud mínima, validez, etc.).

Periódicamente, se debe forzar a que los usuarios cambien sus contraseñas, incluido el administrador. Como veremos en la siguiente sección, una forma automática para llevar a cabo este mantenimiento de la seguridad general de las contraseñas es mediante la información de envejecimiento que se asocia a las cuentas, pues con su gestión se permitirá forzar que los usuarios cambien su contraseña cada cierto tiempo.

4.11. Restricciones de tiempo

Para las cuentas de los usuarios se pueden establecer restricciones de tiempo o envejecimiento respecto a la validez de su cuenta y de su contraseña. Estas restricciones se expresan mediante ciertos valores que se guardan en el fichero `/etc/shadow` para cada usuario:

- `changed`: fecha del último cambio de contraseña.
- `minlife`: número de días que han de pasar para poder cambiar la contraseña después de hacer un cambio.
- `maxlife`: número de días máximo que puede estar con la misma contraseña sin cambiarla después de haber hecho un cambio.

- `warn`: cuántos días antes de que la contraseña expire (`maxlife`) será informado sobre ello, indicándole que tiene que cambiarla.
- `inactive`: número de días que la contraseña seguirá siendo válida después de que expire. El usuario debería cambiarla la próxima vez que entre. Transcurrido este tiempo la cuenta se deshabilitará.
- `expired`: fecha en la que la cuenta expira y se deshabilita de forma automática.

Los valores los establece el administrador con las órdenes `chage` o `passwd`. El fichero `/etc/login.defs` tiene los valores por defecto.

Ejemplo: supongamos que el usuario `pilar` cambia su contraseña el 1 de marzo, y el administrador ejecuta la siguiente orden ese mismo día:

```
# chage -M 20 -W 6 -I 5 -E 2011-10-30 pilar
```

Los tiempos quedarían fijados de la siguiente manera:

- El 14 de marzo pilar recibirá el primer aviso para que cambie su contraseña.
- El 20 de marzo pilar debería haber cambiado su contraseña.
- Si no cambia la contraseña, como se ha fijado el tiempo de inactividad, la cuenta aún no se bloqueará.
- Si el 25 de marzo no la ha cambiado, la cuenta será bloqueada.
- La cuenta expira, y por tanto se bloqueará, el 30 de octubre.

4.12. Mensajes

El fichero `/etc/issue` contiene un mensaje que se mostrará cuando nos conectemos a un terminal antes de pedirnos el login. En el mensaje se pueden incluir caracteres de control que se sustituirán por su correspondiente valor cuando se muestre el mensaje. Por ejemplo, se puede incluir «`\l`», que se sustituirá por el nombre de la terminal de texto (tty2, tty3, etc.) en la que se muestre el mensaje.

El mensaje del día `/etc/motd`, está pensado para mostrarse siempre que un usuario entra en el sistema. Por defecto está vacío. Puedes modificar este fichero para incluir mensajes recordando cosas tales como que se limpien los directorios usados, no se haga un consumo excesivo del disco, o notificar la fecha de la próxima copia de seguridad, por ejemplo.

Hay que tener en cuenta que todo esto no se aplica cuando en modo gráfico abrimos una consola, ya que realmente no hacemos un login en el sistema.

4.13. Ficheros de inicialización

Los ficheros de inicialización son guiones shell que hacen tareas de inicialización como fijar variables, ejecutar funciones, fijar los alias, ... Cada usuario tiene los suyos en su directorio `$HOME` que puede personalizar. Estos ficheros son:

- `.bash_profile`: Se ejecuta cuando un usuario inicia su sesión, ya sea de manera local o remota desde otro ordenador. Su función es configurar el entorno básico de intercomunicación con el sistema.
- `.bashrc`: Una vez que se está conectado a una máquina, este fichero de inicialización se ejecutará cada vez que se abra un nuevo terminal de intérprete de ordenes, o incluso si desde dentro de un shell se inicia una nueva instancia de intérprete de ordenes, por ejemplo tecleando `/bin/bash`.
- `.bash_logout`: Se ejecuta al salir del sistema el usuario, es decir, al cerrar la sesión.

En el directorio `/etc/skel/` hay unas versiones por defecto de estos ficheros de inicialización.

Es importante tener en cuenta que cuando se utiliza la orden `useradd` para crear un usuario, se copian automáticamente estos ficheros de inicialización al directorio personal del nuevo usuario. Sin embargo, la orden `newusers` no realiza esta copia por sí misma, por lo que, tras utilizarla, será necesario copiar estos ficheros a mano.

Obviamente, estos ficheros se pueden personalizar según las necesidades del sistema.

4.14. Intérprete de órdenes

En el último campo del fichero `/etc/passwd` se establece el intérprete de órdenes que se ejecuta al entrar al sistema. Este intérprete podrá escogerse de entre los que aparezcan en el fichero `/etc/shells`, donde se indican los shells permitidos. Si se elimina un shell de este fichero, este shell quedaría como prohibido, con lo que no se podrá elegir como nuevo, pero los usuarios que ya lo tienen asignado lo seguirán usando sin problemas.

Algunos puntos a considerar sobre la elección del intérprete de ordenes:

- Con la orden `chsh` el usuario puede cambiar su shell.
- Si un usuario no tiene asignado ningún intérprete de órdenes, se usará el shell por defecto `/bin/sh`.
- Si se desea que el usuario no pueda entrar al sistema se le debe asignar `/bin/false` o `/sbin/nologin`.
- Se puede indicar que el shell de un usuario es un fichero ejecutable específico. De esta manera, cuando este usuario entre al sistema se ejecuta ese fichero ejecutable, y, al finalizar la ejecución, el usuario sale del sistema automáticamente, sin llegar a hacer un login propiamente dicho.

4.15. Cuentas restringidas

Las cuentas restringidas permiten limitar las acciones de los usuarios en el sistema, haciendo que tengan determinadas restricciones. Existen básicamente dos tipos de cuentas restringidas:

1. Aquellas en las que se asigna como shell un fichero ejecutable que hace una tarea y al terminarla el usuario finaliza su sesión automáticamente. El usuario asociado tiene que tener los permisos necesarios para poder hacer la tarea asignada. Dos ejemplos podrían ser:
   - Usuario para realizar las copias de seguridad: como shell tiene asignado un ejecutable (guión shell) que hace esta tarea.
   - Usuario para apagar el equipo: como shell tiene asignado simplemente la orden `shutdown -h now`.
2. Las que usan el shell restringido `/bin/rbash`[1]. 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 `$SHELL` o `$PATH`.
   - Especificar órdenes que contengan una '/' (e.d. con ruta).
   - Usar la redirección.
   - Usar la orden interna `exec` para reemplazar el shell por otro programa.

   En cualquier caso, hay que tener en cuenta que si queremos mantener a un usuario en un entorno restringido, además de asignarle este shell restringido, es necesario limitar los ficheros que puede ejecutar. Para ello hay que llevar a cabo dos acciones:

[1] El shell restringido `/bin/rbash` es equivalente a ejecutar `/bin/bash -r`. Este shell es realmente un enlace simbólico a `/bin/bash` que por defecto no existe, hay que crearlo con la orden (`ln -s /bin/bash /bin/rbash`).

- Copiar a un directorio concreto los ficheros ejecutables que permitimos usar al usuario.
- Establecer ese directorio concreto como el único valor de la variable PATH.

De esta manera el usuario solamente podrá hacer uso de los ejecutables que le hayamos copiado en ese directorio. En otro caso, con un PATH normal, conformado por la lista habitual de directorios (`/bin`, `/usr/bin`, ...), el usuario podría saltarse las restricciones, ejecutando órdenes que están en estos otros directorios. Por ejemplo, podría ejecutar la orden `chsh` para cambiar este intérprete a uno sin restricciones, o incluso ejecutar directamente la orden `bash` para tener acceso a un shell sin restricciones.

5. Gestión de grupos

5.1. Definición de grupo de usuarios

Los grupos son colecciones de usuarios que comparten ficheros o recursos del sistema. De esta manera permiten garantizar unos permisos para un conjunto de usuarios sin repetirlos individualmente.

Características de un grupo:

- Nombre del grupo o `groupname`.
- Identificador del grupo (GID): Internamente el sistema identifica al grupo por este número.

El fichero de configuración es `/etc/group`, cuyo formato es:

```
nombre:x:gid:lista_de_usuarios_separados_por_comas
```

Un ejemplo de la definición de un grupo en este fichero, sería la línea de texto:

```
iso:x:1019:jcuenca,lfmaimo,piernas,meacacio
```

5.2. Tipos de grupos y actuación

Para cada usuario podemos encontrar dos tipos de grupos:

- Primario: el grupo especificado para este usuario en `/etc/passwd`.
- Secundarios: los otros grupos a los que pertenece el usuario, indicados en `/etc/group`.

Cuando un usuario crea un fichero, se establece como grupo propietario el grupo activo del usuario en ese momento. Por otro lado, al determinar los permisos que un usuario tiene sobre un fichero, p.e. para leerlo o modificarlo, se usan todos los grupos a los que pertenece dicho usuario.

El grupo activo de un usuario suele ser el que tiene definido como primario, salvo que se cambie con la orden `newgrp`. Esta orden realmente lo que hace es lanzar un nuevo shell con el nuevo grupo como grupo activo.

5.3. Órdenes para la gestión de grupos

Las ordenes más importantes relacionadas con la gestión de grupos son:

- `groupadd grupo`: para crear un nuevo grupo.
- `groupdel grupo`: para eliminar un grupo

- `gpasswd`: para administrar la información contenida en los ficheros `/etc/group` y `/etc/gshadow`.

  En relación a esta orden, tenemos que considerar que cada grupo puede tener usuarios administradores, usuarios miembros y, además, puede contar con una contraseña de acceso. La lista de usuarios miembros y administradores de un grupo la establece el usuario `root` de esta manera:

  - `gpasswd -A usuarios grupo`: para definir la lista de usuarios (nombres de usuarios separados por comas) que serán administradores del grupo.
  - `gpasswd -M usuarios grupo`: para definir la lista de usuarios miembros del grupo.

  Si un usuario es definido como administrador de un grupo podrá utilizar a partir de ese momento esta orden para realizar diversas labores de gestión del grupo (obviamente, el usuario `root` siempre puede realizarlas, en cualquier caso):

  - `gpasswd -a usuarios grupo`: para añadir un usuario al grupo.
  - `gpasswd -d usuarios grupo`: para sacar a un usuario del grupo.
  - `gpasswd grupo`: para asignar una contraseña a dicho grupo. Si un grupo tiene contraseña, un usuario que la conozca podrá usar ese grupo como grupo activo, a pesar de no pertenecer él. En tal caso, cuando este usuario ejecute la orden `newgrp`, le será solicitada dicha contraseña.

- `groups [usuario]`: para consultar los grupos a los que pertenece un usuario.
- `id [usuario]`: para listar el identificador de un usuario y los grupos a los que pertenece.
- `grpck`: para chequear la consistencia de los ficheros de grupos.

El fichero `/etc/gshadow` contiene la información de seguridad de los grupos (grupo, contraseña, y también la relación de usuarios administradores y miembros).

6. Pseudo-usuarios y pseudo-grupos

Tal como dijimos al principio del boletín, un usuario es una entidad que ejecuta programas o posee ficheros. Normalmente es una persona que trabaja en el sistema (entra al sistema, edita ficheros, ejecuta programas, etc.), pero también puede ser un pseudo-usuario. Un pseudo-usuario ejecuta programas o posee ficheros para que determinados servicios funcionen, pero NO está asociado a una persona. Por ejemplo, el pseudo-usuario `apache` ejecuta los servicios necesarios de un servidor Web (en concreto, el demonio `httpd`).

7. Ejercicios

7.1. Creación de usuarios

1. Crea el usuario `iso1` con la orden `useradd` y comprueba si puede entrar en el sistema. Una vez creado el usuario, resuelve las siguientes cuestiones:

   1.1 ¿Crea el directorio `$HOME`?

   1.2 ¿Qué grupo primario le asigna?

   1.3 ¿Copia los ficheros de inicialización al directorio de trabajo del usuario?

   1.4 Finalmente, observa lo que ha escrito en `/etc/passwd` y `/etc/shadow`.

2. Ejecuta, como usuario `iso1`, las siguientes órdenes:

   2.1 Crea un fichero en tu directorio: `touch prueba.txt`

2.2 Con `su`, cambia el usuario propietario de dicho fichero usando la orden: `chown root prueba.txt`

2.3 Comprueba, con la orden `ls -l`, si se ha realizado el cambio.

3. Edita los ficheros `/etc/motd` y `/etc/issue` y cambia los mensajes que tienen. (Recuerda que estos mensajes sólo se visualizan en los terminales en modo texto).

4. En el directorio `/etc/skel`, están los ficheros de configuración iniciales que se copian a los directorios `$HOME` de los usuarios cuando se crean sus cuentas. Realiza las modificaciones que sean oportunas para que:

   4.1 Al crear un usuario, se le copie a su `$HOME` un fichero llamado "`horario`" que contenga lo siguiente «`Las salas de practicas están abiertas todos los días`». (Este fichero se copiará al `$HOME` del usuario, pero no se mostrará ni nada por el estilo).

   4.2 Cada vez que el usuario entre al sistema, se ha de ejecutar la orden `who` para saber quién hay conectado.

5. Crea ahora el usuario `iso2` con la orden `useradd` y comprueba si se le han copiado los ficheros creados en el ejercicio 4.

   Nota: no asignes contraseña al usuario con la opción "-p", ya que `useradd` espera recibir la contraseña encriptada. Por ello, al crear un usuario no se le asigna contraseña y se deja la cuenta bloqueada.

6. Usando la orden `passwd`, asígnale una contraseña al usuario `iso2`.

7. Entra al sistema con `iso2` y comprueba que se le ha copiado el fichero `horario` y que al entrar al sistema se le ejecuta la orden `who`.

8. Haz uso de la herramienta `newusers` y crea dos usuarios a la vez, llamados `iissoo1` e `iissoo2`. Esta herramienta recibe como entrada un fichero, con el mismo formato que `/etc/passwd`, con el listado de todos los usuarios que se desean añadir. En este caso, se puede asignar una contraseña a los nuevos usuarios, indicándola en texto plano en el fichero correspondiente.

   A continuación, entra al sistema con el usuario `iissoo1` y responde:

   8.1 ¿Crea el directorio `$HOME`?

   8.2 ¿Qué grupo primario le asigna?

   8.3 ¿Copia los ficheros de inicialización al directorio de trabajo del usuario?

7.2. Contraseñas y restricciones de tiempo para las cuentas y las contraseñas

9. Para el usuario `iso2`, establece los siguientes parámetros de tiempo:

   a) El número mínimo de días entre cambios de contraseña es 2.

   b) El usuario puede mantener la misma contraseña durante, como mucho, 60 días.

   c) Una semana antes de que su contraseña expire, el sistema debe empezar a informarle.

   d) Si 15 días después de haber expirado la contraseña aún no ha sido cambiada, la cuenta se debe bloquear.

   e) La cuenta no debe ser accesible a partir del 24 de diciembre del presente año.

10. Como usuario `iso2`, intenta cambiar la contraseña asignada. Cumpliendo las restricciones de tiempo, el sistema no te lo debe permitir.

7.3. Intérprete de órdenes

11. La orden `chsh` permite que un usuario cambie el shell que tiene asignado. Por otro lado, el fichero `/etc/shells` indica los shells que están permitidos en el sistema, es decir, que pueden ser asignadas a un usuario. Ten en cuenta que prohibir un intérprete de órdenes significa que a, partir de ese momento, no se podrá seleccionar, pero los usuarios que previamente lo tenían asignado seguirán usándolo sin problemas. Según esto, resuelve los siguientes ejercicios:

    a) Instala los paquetes `tcsh` y `ksh` (`dnf install tcsh`)(`dnf install ksh`) para disponer en el sistema de dos nuevos tipos de shell.

    b) Como administrador, "prohibe" el uso del shell `/bin/csh` y habilita el uso de `/bin/rbash` (si no está creada, se hará en el ejercicio 16).

    c) Como usuario `iso2`, intenta cambiarte el shell, seleccionando como nuevo `/bin/csh`.

    d) Como `iso2`, selecciona como nuevo shell `/bin/ksh`. A continuación, en un terminal, entra al sistema con este usuario y comprueba si te ha asignado el nuevo shell.

12. Como `root`, cambia el shell asignado a `iso1`, seleccionando `/bin/false`. No uses la orden `chsh` para resolver este ejercicio.

13. Con el usuario `iso1`, intenta entrar al sistema. ¿Puedes? ¿Por qué?

7.4. Información del usuario

14. La orden `chfn` permite que un usuario cambie la información que se tiene guardada sobre él en el fichero `/etc/passwd`. Estos datos se presentan cuando se usa la herramienta `finger`. Por ejemplo, al ejecutar "`finger pilar`" obtendremos:

```
Login: pilar                            Name: Pilar Gonzalez Ferez
Directory: /home/pilar                  Shell: /bin/bash
Office: 3.45, 868 88 76 58              Home Phone: 555555
```

    a) Entra al sistema con el usuario `iso2` y cámbiale esta información.

    b) Comprueba en qué campo del fichero `/etc/passwd` se almacenan los datos introducidos y qué formato se sigue para guardarlos.

7.5. Cuentas restringidas

15. Usando la herramienta `useradd`, crea un nuevo usuario, llamado `apagar`, que apague el sistema haciendo uso de la orden `/sbin/shutdown`. El usuario no debe iniciar ninguna sesión interactiva; simplemente, cuando se entre al sistema con este usuario, la máquina se apagará.

    La orden para apagar la máquina es "`/sbin/shutdown -h now`"

    Asígnale una contraseña y comprueba si apaga la máquina.

    Si no funciona como se esperaba, es muy probable que te estés equivocando al asignarle el UID a ese usuario. Piensa qué usuario es el único que puede ejecutar la orden `shutdown`. Esto te indicará el UID que tienes que asignarle al nuevo usuario.

    Es muy posible que este usuario sólo funcione en modo texto. Pruébalo, por tanto, en una consola de texto.

16. Shell restringido:

    a) Comprueba si existe el fichero `/bin/rbash`. En caso de que no exista, créalo como enlace simbólico al fichero `/bin/bash`.

    b) Permite que el shell restringido pueda ser usado.

c) Como administrador, asigna al usuario `iso1` dicho shell y entra al sistema con ese usuario para comprobar qué acciones puede o no realizar.
En la página de manual de `bash`, en la sección `RESTRICTED SHELL`, encontrarás una descripción detallada de lo que está prohibido para este nuevo tipo de shell.

7.6. Grupos de usuarios y pertenencia a grupos

17. Con `groupadd` crea un nuevo grupo, `admin`, y haz que el usuario `iso2` pertenezca al mismo usando la herramienta `usermod`.

18. Repite el ejercicio anterior, pero esta vez creando un nuevo grupo llamado `ssoo`.

19. Por defecto, `useradd` crea un grupo para el usuario con el mismo nombre. Crea un usuario `iso3` con `useradd` asignándole como grupo primario el grupo `admin` y haciendo que, además, pertenezca a los grupos `ssoo` y `users`.

20. Las órdenes `id` y `groups` permiten conocer los grupos a los que pertenece un usuario. Entra al sistema como el usuario `iso3` y realiza los siguientes ejercicios:

    a) Comprueba, con `groups` e `id`, cuál es el grupo activo del usuario.

    b) Crea un fichero ejecutando "`touch prueba`" y comprueba cuál es su grupo propietario.

    c) Con la orden `newgrp`, haz que el nuevo grupo activo sea `users`. Comprueba con `groups` o `id` que ha cambiado el grupo activo. Crea un fichero y observa cuál es el grupo asignado al mismo.

    d) Comprueba que `newgrp` realmente lo que hace es lanzar un nuevo intérprete de órdenes. Al ejecutar `exit` finalizará ese intérprete y volverá a tener como grupo activo su grupo primario.

    e) ¿Cómo sería el comportamiento si estuviera activado el bit `SGID` del directorio donde se crea el fichero? Por ejemplo, si tenemos el usuario `iso3` que pertenece a los grupos `users`, `admin` y `ssoo`, y tenemos el siguiente directorio:

    ```
    drwxrwsr-x  7 iso3 root 4096 ene 27 10:15 svnroot
    ```

    Sabiendo que el usuario `iso3` no pertenece al grupo `root`, si crea un fichero en el interior de ese directorio, ¿qué grupo es el propietario del fichero creado?

7.7. Borrando usuarios y grupos

21. Borra los usuarios `iso1` y `iso2` con la orden `userdel`. Elimina aquellos directorios `$HOME` que no se hayan borrado.

22. Usando la orden `userdel`, elimina los usuarios `iso3`, `iissoo1` e `iissoo2`, consiguiendo que se borre también su directorio `$HOME`.

23. Intenta ahora borrar el usuario `apagar` con `userdel`. ¿Qué ocurre? ¿Por qué? Fuerza el borrado del usuario, incluyendo el borrado de sus ficheros.

24. Borra los grupos `admin` y `ssoo` usando `groupdel`.