

Boletín de actividades

Administración básica del sistema (Linux II)

Introducción

Antes de empezar, es fundamental tener en cuenta que el sistema operativo se rige por parámetros de configuración almacenados en **ficheros de texto**. Estos ficheros se encuentran en la carpeta `/etc` del sistema.

Teniendo esto claro, para establecer una determinada configuración en el sistema podemos:

- 1) **Modificar manualmente** el fichero con un editor de texto.
- 2) **Utilizar una aplicación**, desde la terminal o desde la interfaz gráfica, que lo modifique por nosotros.

Es importante recordar que independientemente del método utilizado, el resultado será el mismo: una modificación de los parámetros de configuración contenidos en un fichero de texto. Cada método tiene sus inconvenientes:

Modificación manual del fichero de texto:

- Una línea mal escrita en el fichero puede dejar el sistema inutilizable.
- Puede requerir el reinicio del servicio asociado a la configuración para aplicar los nuevos parámetros.

Utilizando una aplicación:

- En el caso de una aplicación gráfica, no tiene porque disponer de todas las configuraciones disponibles. Además, hay sistemas que no disponen de interfaz gráfica.
- En el caso de una aplicación de terminal, requiere el uso de comandos.

Actividad 1. Modificar el nombre del equipo

En esta primera actividad vamos a hacer la misma tarea de administración con los distintos métodos descritos anteriormente. En este caso, cambiar el nombre al equipo.

Apartado 1. Modificación manual.

Modifica el nombre del equipo **a través de su fichero de configuración**. El nuevo nombre debe ser `ubuntu-primerApellido`.

Puedes consultar [este artículo de somebooks](#).

Apartado 2. Usando una aplicación gráfica.

Modifica el nombre del equipo **a través de la interfaz gráfica**. El nuevo nombre debe ser `ubuntu1-primerApellido`.

Puedes consultar el apartado *Cambiar host name en modo gráfico* de [comoinstalarlinux](#).

Apartado 3. Usando una aplicación de terminal.

Modifica el nombre del equipo **a través de los comandos de la terminal**. El nuevo nombre debe ser `ubuntu16-primerApellido`.

Puedes consultar el apartado *Cambiar host name con la línea de comandos* de [comoinstalarlinux](#).

Administración de usuarios.

Actividad 1. Creación.

La creación de usuarios en Ubuntu se puede realizar con dos comandos: `useradd` o `adduser`. A continuación veremos sus diferencias.

Apartado A. Creación (`useradd -d -m`)

Crea un nuevo usuario `usuario1` utilizando el comando `useradd`. Especifica el directorio personal que utilizará el usuario (`/home/usuario1`) y que éste se cree si todavía no existe.

Puedes consultar las opciones del comando con `man useradd` o `useradd --help`.

Apartado B. Contraseña (`passwd`)

El `usuario1` fue creado correctamente pero ¿de qué contraseña dispone? Por defecto los usuarios se crean en estado bloqueado y no se desbloquean hasta introducirles una contraseña.

Establece una contraseña para el `usuario1` utilizando el comando `passwd`.

Apartado C. Creación (`adduser`)

Crea un nuevo usuario `usuario2` y un nuevo usuario `usuario3` utilizando el comando `adduser`.

Este comando realiza algunas tareas de forma automática y nos muestra información sobre ellas. Ten en cuenta:

1. Los únicos datos necesarios son el nombre de usuario y la contraseña.
2. Por defecto se crea un directorio personal con el mismo nombre de usuario en la carpeta `/home`.

Entonces, ¿qué comando usar para crear usuarios?

Los dos son válidos. En general, `adduser` nos proporcionará un usuario totalmente funcional de forma rápida.

Actividad 2. Eliminación.

Dejar cuentas que no se utilizan en un sistema puede convertirse en un problema de seguridad, por ello es importante conocer cómo se eliminan. El principal comando para la eliminación de cuentas de usuario es `userdel`.

Apartado A. Eliminación (`userdel`)

Elimina al `usuario3` utilizando el comando `userdel`.

Comprueba si el usuario puede iniciar sesión y su carpeta personal está disponible.

Apartado B. Seguridad.

Por defecto, el comando `userdel` no borra el directorio personal del usuario. Al eliminar un usuario, una práctica común es mover su carpeta personal a una carpeta "almacén" por si es necesario recuperar sus archivos en el futuro.

Mueve la carpeta personal del `usuario3` a `/almacen/archivosPersonales`

Apartado C. Eliminación completa.

Si sabemos de antemano que no queremos conservar los ficheros personales de un usuario, podemos indicar su eliminación a través de una opción `-r` del comando `userdel`. En cualquier caso, siempre podremos borrar la carpeta personal como si se tratara de otra cualquiera del sistema.

Elimina la carpeta `usuario3` de `/almacen/archivosPersonales`.

Actividad 3. Ficheros de configuración.

Al igual que el resto de configuraciones del sistema, la configuración de los usuarios también se almacena en ficheros de texto. Los ficheros más importantes que intervienen en la gestión de usuarios son: `passwd` y `shadow` ambos en la carpeta `/etc`.

Abre ambos ficheros con un editor de texto e indica la información que contienen.

Actividad 4. Inicio de sesión (`su`)

Cuando se dispone de varios usuarios puede ser necesario cambiar de uno a otro para realizar distintas operaciones. Para hacerlo a través de la terminal se dispone del comando `su`.

Cámbiate al usuario `root`, al `usuario1` y luego al `usuario2`. Por último vuelve a tu usuario.

Administración de grupos.

Los grupos en Linux tienen el mismo propósito que en otros sistemas operativos: gestionar de forma más eficiente el acceso a los recursos del sistema. Asignando un grupo a un recurso se puede permitir o denegar el acceso a los usuarios simplemente añadiéndolos o sacándolos de dicho grupo.

Actividad 5. Administración de grupos.**Apartado A. Grupos de un usuario (groups)**

Lo primero que nos puede interesar como administradores es conocer los grupos a los que pertenece un usuario.

Muestra los grupos a los que pertenece el `usuario1` y el `usuario2`.

Estos usuarios pertenecen a un grupo que tiene el mismo nombre que el propio usuario. En Ubuntu, sino se indica nada al respecto, se hace pertenecer a los usuarios a un grupo con el mismo nombre. En otras distribuciones se hace pertenecer a todos los usuarios por defecto a un grupo *users*.

Un usuario no tiene porque pertenecer a un único grupo. Puedes comprobar por ejemplo los grupos a los que pertenece el usuario administrador.

Apartado B. Grupos del sistema.

Los grupos que hay en el sistema se almacenan en el fichero de configuración `/etc/group`.

Abre el fichero y comprueba los grupos con los que cuenta actualmente el sistema.

Apartado C. Creación y eliminación (groupadd y groupdel)

Los comandos *groupadd* y *groupdel* permiten crear y eliminar un grupo respectivamente.

Crea tres nuevos grupos `grupo1`, `grupo2` y `grupo3`. A continuación, elimina el `grupo3`.

Apartado D. Añadir un usuario a un grupo (usermod -aG)

Para añadir un usuario a un grupo se suele utilizar el comando *usermod*.

Añade el `usuario1` al `grupo1` utilizando el comando *usermod*. A continuación, añade el `usuario2` al `grupo2`.

Apartado E. Cambio de grupo primario (usermod -g)

Aunque se haya añadido a otro grupo, el `usuario1` tiene como grupo primario el grupo `usuario1` asignado en su creación. El grupo primario se utiliza por ejemplo para determinar el grupo propietario de un fichero creado por un usuario.

Haz que el `usuario1` tenga como grupo primario el grupo `grupo1` utilizando el comando *usermod*.

Apartado F. Quitar a un usuario de un grupo (gpasswd -d)

Para sacar a un usuario de un grupo se utiliza el comando *gpasswd*. Este comando también permite incluir un usuario en un grupo.

Saca al `usuario2` del grupo `grupo2` utilizando el comando *gpasswd*.

Actividad G. Creación de un usuario administrador.

Cuando un usuario utiliza el comando `sudo`, puede ejecutar órdenes que permiten administrar el sistema. Para proporcionar a un usuario la capacidad de usar `sudo`, es suficiente con incluirlo en el grupo también llamado `sudo`.

Crea un nuevo usuario denominado `admin` y hazlo administrador del sistema.

Administración y políticas de contraseña.**Actividad 6. Contraseñas.****Apartado A. Bloqueo y desbloqueo (`passwd -l -u`)**

El comando `passwd` permite establecer la contraseña de un usuario pero también bloquear o desbloquear una determinada cuenta con las opciones `-l` y `-u` respectivamente.

Bloquea la cuenta del `usuario2`. A continuación, desbloquéala.

Apartado B. Políticas de contraseña (`chage -l -d -M`)

El comando `chage` permite establecer restricciones sobre las contraseñas de los usuarios de forma similar a las llevadas a cabo con las directivas de Windows.

Realiza las siguientes operaciones:

- Comprueba la política actual de contraseña del `usuario1`.
- Configura al `usuario1` para que tenga que cambiar su contraseña en el siguiente inicio de sesión.
- Configura al `usuario2` para que tenga que cambiar su contraseña cada 30 días.

Administración de permisos.

Para conocer los permisos establecidos sobre ficheros y directorios se utiliza la orden `ls -l`, la cuál lista el contenido de una carpeta mostrando información detallada de sus elementos.

```
[root@localhost datos]# ls -la
total 32
drwxr-xr-x. 5 maria usuarios 4096 jun 13 06:12 .
dr-xr-xr-x. 24 root root 4096 jun 13 05:59 ..
drwxrwx---. 2 maria jefes 4096 jun 13 06:11 documentos
drwxr-xr-x. 2 maria usuarios 4096 jun 13 06:11 programas
drwxr-xr-x. 2 maria usuarios 4096 jun 13 06:11 software
```

El primer conjunto de caracteres (por ejemplo `drwxrwx---` en la carpeta `documentos`) hace referencia a los permisos. Dentro este conjunto, el primer carácter puede ser una letra `d`, una letra `l` o un guión indicando que se trata de un directorio, un enlace o un fichero respectivamente. A partir de ahí, se describen los permisos asignados. Por ejemplo fíjate en la carpeta `programas`. La primera `d` indica que se trata de una carpeta y a continuación encontramos:

- `rw`, permisos para el usuario propietario (`u`)
- `r-x`, permisos para el grupo propietario (`g`)
- `---`, permisos para otros usuarios (`o`), el resto de usuarios del sistema.

En Linux cada fichero o directorio tiene un único usuario propietario y un único grupo propietario. En el caso de la carpeta `programas` su usuario propietario es `maria` y su grupo propietario es `usuarios`.

Las letras `r`, `w` y `x` tienen el siguiente significado dependiendo si se trata de un fichero o de un directorio:

	Fichero	Directorio
<code>r</code>	Se puede leer.	Se puede ver su contenido.
<code>w</code>	Se puede modificar.	Su contenido se puede modificar.
<code>x</code>	Se puede ejecutar como un programa.	Se puede usar <code>cd</code> para posicionarse dentro.

En caso de aparecer la letra, ese permiso está habilitado, si aparece un guión está deshabilitado.

En la carpeta `programas`, los permisos asignados son: `rw` `r-x` `---`, por lo que el usuario `maria` puede ver, modificar y colocarse dentro de la carpeta, los usuarios del grupo `usuarios` y el resto de usuarios del sistema sólo pueden ver y colocarse dentro de la carpeta.

Los comandos más importantes en la gestión de permisos son:

- `chown` para cambiar el usuario propietario.
- `chgrp` para cambiar el grupo propietario.
- `chmod` para especificar los permisos deseados.

Actividad 7. Caso práctico - Permisos estándar.

Un equipo ubicado en un aula requiere los siguientes usuarios y grupos. Créalos adecuadamente:

- Un usuario `alumno1` y `alumno2` que pertenecen a un grupo `alumnado`.
- Un usuario `profesor1` y `profesor2` que pertenecen a un grupo `profesorado`.

El equipo también necesita la siguiente estructura de carpetas y archivos. Créala:

- Una carpeta `/home/dir_compartido`.
- Dentro de ella, dos carpetas `dir_alumnado` y `dir_profesorado`.
- Otras dos dentro de éstas: `dir_alumno1` y `dir_alumno2` y `dir_profesor1` y `dir_profesor2`.
- Por último, dos archivos `archivo1` y `archivo2` en la carpeta `alumnado`.

Establece los permisos necesarios para cumplir las siguientes restricciones:

- a) La carpeta `dir_compartido` puede ser accedida por alumnos y profesores.
- b) La carpeta `dir_profesorado` puede ser accedida y gestionada por cualquier usuario profesor y nadie más.
- c) La carpeta `dir_alumnado` puede ser accedida y gestionada por cualquier usuario alumno. Los usuarios profesores pueden ver su contenido.
- d) Las carpetas de cada usuario pueden ser accedidas y gestionadas por el propio usuario y nadie más.
- e) El `usuario1` tiene todos los permisos sobre el `archivo1`, el resto de alumnos sólo pueden leerlo y los profesores no tienen ningún permiso.
- f) El `usuario2` tiene todos los permisos sobre el `archivo2`, el resto de alumnos y los profesores sólo pueden leerlo.

Actividad 8. Caso práctico - Listas de Control de Acceso.

Linux también permite definir Listas de Control de Acceso para determinar los permisos de los recursos.

Establece los permisos especificados en la actividad anterior utilizando Listas de Control de Acceso.

Actividad 21. Gestión de discos.

A continuación veremos como preparar un nuevo disco duro para que sea funcional en el equipo. Utiliza `fdisk` para particionar el disco, `mkfs` para formatear las particiones y `mount` para incorporar las particiones al sistema de ficheros.

- En primer lugar, añade físicamente un nuevo disco duro SATA a VirtualBox de 30Gb de capacidad.
- A continuación crea 6 particiones primarias de 5G cada una. La tabla de particiones del disco debe tener formato GPT (hasta 128 particiones, con formato MBR sólo se pueden crear 4 particiones primarias).
- Dale formato EXT4 a cada partición.
- Monta cada partición en una carpeta dentro del directorio `/mnt` (directorio del sistema destinado para montar dispositivos aunque también sería posible utilizar otra ubicación). Por ejemplo la partición 1 se montaría en la carpeta `/mnt/vol1` (la carpeta `vol1` debe ser previamente creada).

Recursos: Apartado 2.1.2. *fdisk* de los contenidos de la plataforma.

Actividad 22. Gestión de discos.

Las carpetas donde se han montado las particiones no se remontarán automáticamente al iniciar el sistema. Para no tener que montarlas cada vez que se inicie el equipo es necesario editar el fichero de configuración `/etc/fstab`.

Es suficiente con añadir una línea como ésta por cada punto de montaje:

```
UUID=b15ec9ce-7b59-45c6 /mnt/vol1 ext4 defaults 0 2
```

El UUID es un identificador único de la partición. Puedes conocer los identificadores de cada partición con el comando *blkid*. La opción 0 indica que la partición será montada en modo lectura/escritura y la opción 2 que debe montarse al iniciar el sistema.

Configura el sistema para que todas las particiones del nuevo disco se monten automáticamente al iniciar el equipo. Presta mucha atención a lo que escribes en el fichero de configuración, un archivo mal formado puede dejar el equipo inutilizable.

Actividad 23. Procesos

En la unidad anterior vimos que el comando *ps* nos podía proporcionar información sobre los procesos que se están ejecutando en el sistema. Si lo usábamos sin ninguna opción, nos daba información de los procesos que estaba ejecutando el usuario actual en la terminal actual. Normalmente nos interesará conocer los procesos que hay actualmente en el sistema independientemente del usuario o terminal desde la que se hayan iniciado, ¿qué opciones de *ps* puedo utilizar para obtener una información completa de los procesos?

Actividad 24. Procesos.

Obteniendo una visión completa de los procesos con *ps* es difícil localizar un determinado programa que se esté ejecutando en el equipo. El comando *grep* nos permite buscar las coincidencias de una cadena en un texto. ¿cómo podría utilizarse este comando en combinación con *ps* para localizar la información de, por ejemplo, el proceso *nginx* (un servidor web instalado por defecto en Ubuntu)?

Actividad 25. Procesos.

Otra opción muy interesante (y más amigable) para gestionar los procesos es el programa *htop*. Este programa no viene preinstalado en Ubuntu. Instálalo desde los repositorios oficiales y comprueba su funcionamiento.

Actividad 26. Servicios.

En sistemas *Linux*, lo que conocíamos como servicios en sistemas Windows, se les denomina

demonios. Recuerda que no son más que programas que se ejecutan en segundo plano y que normalmente se inician al arrancar el equipo.

En Linux, los servicios son gestionados por el primer proceso que se carga tras el núcleo (por lo tanto con PID 1) conocido como proceso de inicialización o más comunmente *init*. Este servicio se denomina *systemd* y puede ser gestionado con el comando *systemctl* el cuál nos permitirá iniciar, parar o ver el estado de todos los servicios del sistema.

Para hacer pruebas que no interfieran con el funcionamiento del sistema, instalaremos el servidor *openssh-server* disponible en los repositorios oficiales de Ubuntu. Al terminar la instalación, se iniciará un servicio (como cualquier otro servidor) a la espera de peticiones de clientes.

Para comprobar que el servicio se ha instalado, localiza su nombre exacto con el comando *systemctl* junto con *grep*.

Actividad 27. Servicios.

Ahora que conoces el nombre del servicio, utiliza *systemctl* con las opciones *status*, *stop*, *start*, *restart*, *enable* y *disable* para conocer el estado del servicio, pararlo, iniciarlo, reiniciarlo, habilitarlo y deshabilitarlo respectivamente.

Actividad 28. Programación de tareas.

Un usuario puede configurar una tarea para que se lleve a cabo en un determinado momento haciendo uso de la herramienta *Cron*. El principal comando que se utiliza para su configuración es *crontab*.

Para conocer las tareas programadas del usuario actual se utiliza la opción *-l*. Si quieres especificar las tareas programadas para un usuario puedes usar la opción *-u*.

Realiza las siguientes programaciones:

1. Crea un contenedor *dirHome.tgz* (con el comando *tar*) en el directorio */MisCopias* con los archivos de la carpeta */home* todos los viernes a las 12:03 a. m.
2. Sincroniza la carpeta */misCopias/misArchivos* con el directorio */home/usuario* (con el comando *rsync*) todas las mañanas a las 12 a.m.
3. Actualiza la lista de repositorios el primer día de cada mes a las 1:00 a.m.

Recursos: Apartado 4.5 . *Programación de tareas*, apartado 6.1.1. *El comando tar*, apartado 6.1.3. *rsync* de los contenidos de la plataforma.