

---

## EJERCICIOS TEMA 10: ADMINISTRACIÓN DE LINUX

### Consideraciones previas

La documentación a entregar será un fichero pdf con los **comandos que has escrito** de los puntos indicados en cada práctica. No hace falta que captures pantallas.

Hay que hacer toda la práctica mediante la consola de comandos.

Recordad la utilización del mandato **sudo** en ciertos comandos. Se antepone a un comando para ejecutarlo en modo administrador, con los mismos privilegios que si fuéramos el usuario root. Nos pedirá nuestra contraseña.

### Preparación Máquinas virtuales

Vamos a trabajar con las 2 máquinas virtuales creadas la quincena anterior (*Kepler* y *Einstein*). La máquina principal sobre la que trabajaremos será *Einstein*, *Kepler* sólo la utilizaremos para abrir una sesión remota desde *Einstein*.

---

## PRÁCTICA 1. Usuarios y Grupos

Para entregar, escribe los comandos que has escrito en los puntos 3, 10, 20, 24 y 25.

1. Arranca la máquina *Einstein*.
2. Abre un Terminal. Lo vamos a hacer todo con comandos. Nos va a servir para familiarizarnos con el entorno de trabajo de Linux. Al utilizar el mandato *sudo* nos pedirá nuestra contraseña en algunas ocasiones.
3. **Cambia la contraseña del usuario *root* para que sea *toor***
4. Abre una sesión con el usuario *root*.
5. Crea un directorio que cuelgue del directorio raíz llamado *casa*
6. Crea un grupo llamado *restringido*
7. Visualiza el GID de ese grupo (deberás hacer un *cat* del fichero correspondiente)
8. Sitúate en el directorio */etc/skel*. Este directorio contiene el esqueleto del directorio *\$HOME* de los nuevos usuarios.
9. Crea un nuevo directorio (dentro de */etc/skel*) llamado *Apuntes*. De esta manera los usuarios que se creen a partir de ahora lo tendrán en su *\$HOME*.
10. **Crea un usuario (con el mandato *useradd*) llamado *intruso*, y que se cree su directorio *\$HOME* de manera automática. El grupo principal al que pertenezca el usuario debe ser el grupo creado *resrtngido*, para ello utiliza el parámetro *-g* del comando de crear usuarios. (Utiliza *man* para cualquier duda).**
11. Introdúcele contraseña.
12. Edita el fichero */etc/passwd* para que el shell del usuario *intruso* sea el *csh*
13. Mediante un mandato vuelve a cambiar el shell del usuario *intruso* para que sea el *sh*.
14. Utiliza el mandato *chfn* para introducir la información relativa al usuario.
15. Utiliza el mandato *usermod* para dar fecha de expiración de la cuenta del usuario intruso el 30-06-2017
16. Compruébalo haciendo un *cat* del fichero correspondiente. ¿La información que sale es la que has introducido en el punto anterior? ¿Cómo se debe interpretar?
17. Abre una nueva terminal texto (con **Ctrl+Alt+F1**, o F2, o F3...)
18. Abre una sesión con el usuario intruso. Comprueba en qué directorio *\$HOME* se ha iniciado la sesión. Así como que aparece el directorio *Apuntes* en *\$HOME*.
19. Cierra la sesión.

20. Con el usuario *root* bloquea la cuenta de *intruso*. (Utiliza el mandato *passwd* con un parámetro)
21. Intenta entrar en el sistema de nuevo con el usuario *intruso*
22. Desbloquea la cuenta (Otra vez con el mandato *passwd* y un parámetro)
23. Crea un grupo llamado *sospechosos*
24. Añade al usuario *intruso* al grupo *sospechosos* (como grupo secundario).
25. Elimina el usuario *intruso* (junto con su directorio \$HOME)
26. Elimina el grupo *restringido*

## **PRÁCTICA 2. Permisos**

Para entregar, escribe los comandos que has escrito en los puntos 37, 38 y 40.

27. Debes tener abierta la sesión con el usuario habitual (alumno).
28. Crea el usuario *a\_juan*, de apellido *lópez*, con contraseña 123.
29. Crea un usuario denominado *a\_jose*, de apellido *ruíz* con contraseña 123.
30. Crea el usuario *p\_donluis*, de apellido *gómez*, con contraseña 123.
31. Crea el usuario *p\_donpedro*, de apellido *garcía*, con contraseña 123.
32. Crea el grupo *alumnos* y haz miembros de este grupo a los dos primeros usuarios creados.
33. Crea el grupo *profesores* y haz miembros de este grupo a los dos últimos usuarios creados.
34. Crea el grupo *alumprof* y haz miembros de este grupo a los 2 profesores y a los 2 alumnos.
35. Ahora vamos a crear los directorios que se van a compartir. Lo haremos en */datos*, con el usuario *root*. Crea los directorios *ejercicios*, *exámenes* y *entregas* dentro de */datos*.
36. Visualiza el usuario propietario, el grupo propietario y los permisos del directorio */datos* y de los 3 directorios que has creado dentro de este. (Ejecutando el mandato *ls -l*)
37. **Cambia el grupo propietario de */datos* para que sea *alumprof*.**
38. **Cambia los permisos del directorio */datos* para que quienes no pertenezcan al grupo *alumprof* no puedan tener ningún tipo de acceso a ese directorio. Con esto nos aseguramos que sólo *alumnos* y *profesores* pueden acceder al directorio */datos*. Basta con cambiar los permisos de *otros***

39. En la carpeta *ejercicios* los *profesores* van a poder leer y escribir ficheros y los *alumnos* sólo podrán leer. Para ello:
- Haz como grupo propietario de la carpeta *ejercicios* a *profesores*.
  - Modifica los permisos para que los miembros de *profesores* tengan permisos de lectura y escritura, y el resto de usuarios tengan permiso de sólo lectura (no te preocupes por los usuarios que no son *alumnos*, pues no deben tener ningún acceso al no ser miembros de *alumprof* y por consiguiente no haber podido acceder a */datos*).
40. En la carpeta *entregas* los *profesores* sólo van a poder leer y los *alumnos* sólo podrán escribir. Para ello:
- Haz como grupo propietario de la carpeta *entregas* a *profesores*.
  - Modifica los permisos para que los miembros de *profesores* tengan permisos de lectura, y el resto de usuarios tengan permiso de sólo escritura.
41. En la carpeta *exámenes* los *profesores* sólo van a poder escribir y los *alumnos* sólo podrán leer. Para ello:
- Haz como grupo propietario de la carpeta *exámenes* a *profesores*.
  - Modifica los permisos para que los miembros de *profesores* tengan permisos de escritura, y el resto de usuarios tengan permiso de sólo lectura.
42. Para probarlos vamos a necesitar tener abiertas 3 sesiones una con el usuario actual, otra con *p\_donpedro* y otra con *a\_jose*. Puedes utilizar las terminales texto o abrir 2 terminales más (además de la actual) en el entorno gráfico y mediante el mandato **su** te cambias de usuario en esas 2 terminales.
43. Intenta acceder con tu usuario actual a cualquiera de los 3 directorios (*ejercicios*, *entregas* y *exámenes*). No debes poder.
44. Con tu usuario actual intenta ver el contenido de esos 3 directorios.
45. Con el usuario *p\_donpedro* graba en el directorio *ejercicios* un nuevo fichero llamado *ejercicio1*.
46. Con el usuario *a\_jose* intenta grabar en el directorio *ejercicios* un nuevo fichero llamado *ejercicio2*. No debe dejarte.
47. ¿Puedes visualizar y copiar el fichero *ejercicio1* con el usuario *a\_jose*?
48. ¿Puedes visualizar y copiar el fichero *ejercicio1* con tu usuario?
49. Con el usuario *a\_jose* crea un fichero dentro del directorio *entregas* llamado *entrega\_ejer1*.

50. Con el usuario *p\_donpedro* visualiza la entrega.
51. ¿Puedes escribir algo en el directorio *entregas* con el usuario *p\_donpedro*?
52. ¿Puedes visualizar el contenido del directorio *entregas* con el usuario *a\_jose*?
53. ¿Puedes visualizar y copiar el fichero *entrega\_ejer1* con tu usuario?

### **PRÁCTICA 3. Procesos**

Para entregar, escribe los comandos que has escrito en los puntos 55, 56 y 63.

54. Ejecuta el comando *sleep 160* (aparentemente no hace nada, simplemente espera –duerme- durante 160 segundos).
55. **Mientras se está ejecutando, pásalo a segundo plano.**
56. **Ejecuta el comando *sleep 145* directamente en segundo plano.**
57. Ejecuta el comando *updatedb* (con sudo) directamente en segundo plano.
58. Teclea *jobs* y observa la información.
59. Trae a primer plano el proceso que has ejecutado en el paso 62 (*updatedb*).
60. Lanza el comando *sleep 150* directamente en segundo plano.
61. Ejecuta *top*
62. Ejecuta *ps*
63. **Mata el proceso lanzado en el punto 65 utilizando la orden *kill***
64. Ejecuta el *firefox*.
65. Ejecuta otra vez el *firefox* sin cerrar el anterior.
66. Con un solo mandato finaliza todas las instancias abiertas del *firefox*.

### **PRÁCTICA 4. Instalación de software**

Para entregar, escribe los comandos que has escrito en los puntos 69 y 71.

67. Instala desde el Centro de Software de Ubuntu el programa de terminal ***guake terminal***
68. Actualiza la lista de paquetes de los repositorios mediante el comando *apt-get*
69. **Mediante la utilidad *apt-get* instala el paquete *virtualbox***
70. Abre Firefox y accede a la página de descargas de Google Chrome. Selecciona la descarga para la plataforma Linux y guarda el paquete *google-chrome-*

*stable\_current\_amd64.deb* en el directorio descargas. En el cuadro de diálogo selecciona *Guardar Archivo* (NO INSTALAR).

71. Desde una terminal de comandos instala con el mandato correspondiente el paquete descargado. (OJO, si te da error de dependencias añade el parámetro `--force-depends`)

## **PRÁCTICA 5. Programación de tareas**

Esta práctica consiste en programar tareas para que se ejecuten en días y horas determinados. Para poder probarlo habrá que ajustar el reloj del sistema minutos antes de que se ejecuten. Puesto que trabajamos en máquinas virtuales con las *guestadditions* instaladas, habrá que ajustar el reloj del sistema de la máquina host.

Para entregar, escribe los comandos que has escrito en los puntos 72 y 74.

72. **Mediante el programa cron planifica una tarea para que todos los días a las 21,00 se apague el sistema.** Para entregar muestra el contenido de la línea que has añadido al fichero correspondiente.
73. Mediante cron planifica una tarea para que cada lunes a las 8,00 se actualice la lista de repositorios (*apt-get update*). Ese comando genera una salida, almacénala en tu directorio \$HOME con el nombre de *actualizacion*.
74. **Mediante cron planifica una tarea para que cada media hora se almacene en un fichero llamado *conexión* ubicado en tu directorio \$HOME los usuarios conectados al sistema. No se debe sobrescribir el fichero, sino añadir al final.**
75. Mediante cron planifica una tarea para que los días 1, 8, 15, 21 y 28 de cada mes se descargue automáticamente de Internet el fichero [http://www.aemet.es/documentos\\_d/enportada/20180407160748\\_p54tespl.pdf](http://www.aemet.es/documentos_d/enportada/20180407160748_p54tespl.pdf) y lo deje en el directorio Descargas de tu \$HOME (mandato *wget*).

## **PRÁCTICA 6. Demonios**

Para entregar, escribe los comandos que has escrito en los puntos 76 y 78.

76. **Comprueba el estado del servicio de red llamado *networking***
77. Vamos a detener un servicio, en este caso el que controla el gestor de sesiones gráficas. Cada distribución de Linux lleva un gestor de sesiones asociado (GDM en GNOME, KDM en KDE o Lightdm que es el que lleva Ubuntu con Unity). Detén el servicio *lightdm* pero antes salva lo que tengas abierto en la sesión gráfica.
78. **Abre el terminal 1 (teclas Ctrl+Alt+F1) y vuelve a arrancar el servicio.**
79. Con las teclas Ctrl+Alt+F7 podrás volver al entorno gráfico.

## **PRÁCTICA 7. Inicio de sesión**

Al iniciar una sesión en Linux se ejecutan una serie de procesos y scripts, los cuales analizan usuario y contraseña y configuran lo que van a ser el shell de la nueva sesión.

A grandes rasgos, y dependiendo de la distribución Linux, los pasos son los siguientes:

- a) Se ejecuta login, el cual pide nombre de usuario y contraseña.
- b) Se comprueba si la contraseña es correcta.
- c) Se analiza y toman valores de los ficheros */etc/passwd* y */etc/shadow*
- d) Se ejecuta el script */etc/bash.bashrc*  
Este configura el prompt, variable PS1  
Carga los alias por defecto
- e) Se ejecuta el script *.bashrc* que hay en el directorio *\$HOME* del usuario que ha iniciado la sesión: *~/.bashrc*  
A diferencia del anterior, éste sólo se ejecuta para el usuario que ha iniciado la sesión. Aquí podría el usuario definir sus alias particulares o redefinirse el prompt (variable PS1)
- f) Se ejecuta el script */etc/profile*  
También es común a todos los usuarios.  
Aquí se fija, entre otras cosas, el valor de *umask* (022), las variables generales \$UID, \$USER, \$HOST, ...  
También se fija el PATH
- g) Si existe se ejecuta el script */etc/profile.local*

Si hay que hacer alguna modificación en `/etc/profile` se recomienda hacerla en `/etc/profile.local` y no tocar el `/etc/profile`, pues este puede ser modificado en una actualización de versión.

- h) Por último se ejecuta el script `.profile` que hay en el directorio `$HOME` del usuario que ha iniciado la sesión: `~/.profile`

Igual que sucedía con el fichero `.bashrc`, el fichero `~/.profile` sólo se ejecuta para el usuario que ha iniciado la sesión, con lo cual las modificaciones que se hagan aquí sólo afectan al usuario.

Para entregar, escribe las líneas de comandos que has añadido en los scripts de los puntos 80 y 81. Indica también el script que has modificado.

80. Modifica el script correspondiente (mediante `nano`, `gedit` o `vi`) para añadir un nuevo alias para que al ejecutar el mandato `cls` limpie la pantalla como hace el mandato `clear`, de manera que puedan usarlo todos los usuarios que inicien sesión.
81. Modifica el fichero `profile` correspondiente de manera que sólo tu usuario lea un mensaje que diga “Bienvenido a Linux” cada vez que inicie la sesión (el mandato para sacar un mensaje en pantalla es `echo`).
82. Modifica el script correspondiente para cambiar la máscara de permisos en la creación de archivos **para todos los usuarios**. Fija una máscara de manera que los usuarios *otros* no tengan permiso de lectura en los nuevos ficheros y directorios que se creen.

## **PRÁCTICA 8. Webmin**

**Webmin** es una **herramienta web para la configuración de sistemas operativos GNU/Linux** que ofrece una interfaz muy simple y debido al hecho de ser utilizada desde el navegador web es compatible con cualquier escritorio o entorno gráfico. Se puede configurar gran cantidad de aspectos del sistema (usuarios, grupos, servicios, arranque, ...).

Para entregar, captura la pantalla del punto 89.



83. Para instalar webmin hay que hacer una serie de pasos antes de poder ejecutar el `apt-get install`:

a) En primer lugar vamos a editar el fichero de repositorios mediante el nano:

```
$ sudo nano /etc/apt/sources.list
```

b) Añadimos en el fichero las siguientes líneas:

```
deb http://download.webmin.com/download/repository sarge contrib
```

```
deb http://webmin.mirror.somersettechsolutions.co.uk/repository sarge contrib
```

c) Guardamos el fichero y salimos. Le hemos indicado dónde debe ir a buscar paquetes además de los repositorios que ya había.

d) Descargamos la clave CPG del repositorio para poder descargar paquetes desde ahí:

```
$ wget http://www.webmin.com/jcameron-key.asc
```

e) Instalamos la clave:

```
$ sudo apt-key add jcameron-key.asc
```

f) Actualizamos la lista de paquetes disponibles desde los repositorios:

```
$ sudo apt-get update
```

84. Instala el paquete *webmin* mediante el comando `apt-get`

85. Abre al navegador y accede a <https://localhost:10000>

86. Confirma la excepción de seguridad

87. Entra con el usuario *root* y su contraseña (habíamos puesto *toor*).

88. En el submenú *Webmin* cambia el idioma a Español-UTF8

89. **Entra en el menú *Sistema* apartado *Arranque y parada*. Comprueba los servicios que hay arrancados y se arrancan al iniciar. Captura la pantalla**

## **PRÁCTICA 9. Administración remota**

Para entregar, escribe los comandos que has escrito en los puntos 94 y 104.

90. Configura el adaptador de red de la máquina virtual para ponerlo en modo puente.

91. Visualiza la IP de tu equipo tecleando el mandato *ifconfig*

92. Comprueba que está instalado y arrancado el servicio Servidor SSH. Si no lo está, lo instalas.

93. Arranca la máquina *Kepler* y configura el adaptador de red virtual en modo puente.

94. **Conéctate desde *Kepler* a *Einstein* utilizando el cliente SSH.**
95. Si no especificas nombre de usuario, entrarás con el mismo nombre de usuario que estás utilizando.
96. Teclea *logout* para cerrar sesión.
97. Vuelve a entrar, pero esta vez con otro nombre de usuario. Teclea `ssh usuario@di.rec.ción.ip`.
98. Comprueba que estás en el otro equipo.
99. Intenta ejecutar una aplicación gráfica como *gedit*. No debe dejarte.
100. Cierra sesión en *Einstein* (*exit* o *logout*).
101. Vuelve a entrar en *Einstein*, esta vez tecleando el parámetro `-X` al final del mandato: `ssh di.rec.ción.ip -X`
102. Ya podemos ejecutar aplicaciones gráficas (si nos hemos conectado desde un terminal de la interfaz gráfica).
103. Instala en *Einstein* el programa *Gimp*.
104. **Ejecuta en *Kepler* el Gimp abriendo una sesión ssh con el parámetro `-X`. Al ejecutarse lo está haciendo en *Einstein*, sin utilizar recursos de *Kepler*, pero manejándolo como si estuviera instalado en esta última máquina.**
105. Cierra sesión con *logout*.