

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

`$ sudo passwd root`

4. Abre una sesión con el usuario root.

`$ su root`

5. Crea un directorio que cuelgue del directorio raíz llamado casa

`# mkdir casa`

6. Crea un grupo llamado restringido

`# addgroup restringido`

7. Visualiza el GID de ese grupo (deberás hacer un cat del fichero correspondiente)

`# cat /etc/group`

8. Sitúate en el directorio `/etc/skel`. Este directorio contiene el esqueleto del directorio `$HOME` de los nuevos usuarios.

`# cd /etc/skel`

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`.

`# mkdir Apuntes`

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 restringido, para ello utiliza el parámetro `-g` del comando de crear usuarios. (Utiliza `man` para cualquier duda).

`# useradd -g restringido -m intruso`

11. Introdúcele contraseña.

`# passwd intruso`

12. Edita el fichero /etc/passwd para que el shell del usuario intruso sea el csh

`nano /etc/passwd` (editar ultimo campo del usuario)

13. Mediante un mandato vuelve a cambiar el shell del usuario intruso para que sea el sh.

`# usermod -s /bin/sh intruso`

14. Utiliza el mandato chfn para introducir la información relativa al usuario.

`# chfn -f "Intruso garcia lopez" -h 1234567 intruso`

`# cat etc/passwd`

`intruso:x:1001:1001:Intruso garcia lopez,,1234567:/home/intruso:sh`
(el ultimo campo es /bin/sh corregido el punto 13)

15. Utiliza el mandato usermod para dar fecha de expiración de la cuenta del usuario intruso el 30-06-2018

`# usermod -e 2018-06-30 intruso`

`intruso:6U/ITHAeG$Vg78whQUMKTsviNktUHxvtlfzRi.P4UEk/88CNeI6NPqTbM2pmZF7.Qq9/2IwZuIhydCYewx5EkWCxJGxdCtU.:17648:0:99999:7::17712:`

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?

`# cat /etc/shadow`

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)

`# passwd -l intruso`

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)

`# passwd -u intruso`

23. Crea un grupo llamado sospechosos

`# addgroup sospechosos`

24. Añade al usuario intruso al grupo sospechosos (como grupo secundario).

`# gpasswd -a intruso sospechosos`

Alumno: Llorenç Arbiol Barelles

25. Elimina el usuario intruso (junto con su directorio \$HOME)

deluser --remove-home intruso

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.

adduser a_juan (Siendo root abre dialogo y pregunta todos los datos de usuario para introducir)

29. Crea un usuario denominado a_jose, de apellido ruíz con contraseña 123.

adduser a_jose

30. Crea el usuario p_donluis, de apellido gómez, con contraseña 123.

adduser p_donluis

31. Crea el usuario p_donpedro, de apellido garcía, con contraseña 123.

adduser p_donpedro

32. Crea el grupo alumnos y haz miembros de este grupo a los dos primeros usuarios creados.

addgroup alumnos

adduser a_juan alumnos

adduser a_jose alumnos

33. Crea el grupo profesores y haz miembros de este grupo a los dos últimos usuarios creados.

addgroup profesores

adduser p_donluis profesores

adduser p_donpedro profesores

34. Crea el grupo alumprof y haz miembros de este grupo a los 2 profesores y a los 2 alumnos.

addgroup alumprof

adduser a_juan alumprof

```
# adduser a_jose alumprof
```

```
# adduser p_donluis alumprof
```

```
# adduser p_donpedro alumprof
```

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.

```
# mkdir -p /home/datos/{ejercicios,exámenes,entregas}
```

(-p es para si no existe directorio lo crea. El entrecasillado es para crear varios directorios a la vez con un mismo comando.)

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)

```
root@alumno-VirtualBox:/home# ls -l /home/datos/
total 12
drwxr-xr-x 2 root root 4096 abr 30 07:12 ejercicios
drwxr-xr-x 2 root root 4096 abr 30 07:12 entregas
drwxr-xr-x 2 root root 4096 abr 30 07:12 exámenes
```

37. Cambia el grupo propietario de /datos para que sea alumprof.

```
# chgrp alumprof /home/datos/
```

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

```
# chmod o-rwx /home/datos/
```

39. En la carpeta ejercicios los profesores van a poder leer y escribir ficheros y los alumnos sólo podrán leer. Para ello:

a) Haz como grupo propietario de la carpeta ejercicios a profesores.

```
# chgrp profesores /home/datos/ejercicios/
```

b) Modifica los permisos para que los miembros de profesores tengan permiso 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).

```
# chmod g+rw,u-w /home/datos/ejercicios/
```

40. En la carpeta entregas los profesores sólo van a poder leer y los alumnos sólo podrán escribir. Para ello:

a) Haz como grupo propietario de la carpeta entregas a profesores.

```
# chgrp profesores /home/datos/entregas/
```

b) Modifica los permisos para que los miembros de profesores tengan permisos de lectura, y el resto de usuarios tengan permiso de sólo escritura.

```
# chmod g+r,o+w /home/datos/entregas/
```

41. En la carpeta exámenes los profesores sólo van a poder escribir y los alumnos sólo podrán leer. Para ello:

a) Haz como grupo propietario de la carpeta exámenes a profesores.

```
# chgrp profesores /home/datos/examenes/
```

b) Modifica los permisos para que los miembros de profesores tengan permisos de escritura, y el resto de usuarios tengan permiso de sólo lectura.

```
# chmod g+w,u-w /home/datos/examenes
```

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 su

43. Intenta acceder con tu usuario actual a cualquiera de los 3 directorios (ejercicios, entregas y exámenes). No debes poder.

OK

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.

```
p_donpedro@alumno-VirtualBox:/home/alumno$ nano /home/datos/ejercicios/ejercicio1
```

46. Con el usuario a_jose intenta grabar en el directorio ejercicios un nuevo fichero llamado ejercicio2. No debe dejarte.

```
nano /home/datos/ejercicios/ejercicio1
```

```
[ 1 línea leída (aviso: no hay permiso de escritura) ]
```

47. ¿Puedes visualizar y copiar el fichero ejercicio1 con el usuario a_jose?

SI

48. ¿Puedes visualizar y copiar el fichero ejercicio1 con tu usuario?

NO

49. Con el usuario a_jose crea un fichero dentro del directorio entregas llamado

entrega_ejer1.

OK

50. Con el usuario p_donpedro visualiza la entrega.

SI

51. ¿Puedes escribir algo en el directorio entregas con el usuario p_donpedro?

NO

52. ¿Puedes visualizar el contenido del directorio entregas con el usuario a_jose?

SI

53. ¿Puedes visualizar y copiar el fichero entrega_ejer1 con tu usuario?

NO

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.

Ctrl+Z

56. Ejecuta el comando sleep 145 directamente en segundo plano.

\$ Sleep 145 &

57. Ejecuta el comando updatedb (con sudo) directamente en segundo plano.

\$ sudo updatedb &

58. Teclea jobs y observa la información

```
alumno@alumno-VirtualBox:~$ jobs
[1]-  Detenido                  sleep 160
[2]   Ejecutando                sleep 145 &
[3]+  Detenido                  sudo updatedb
```

59. Trae a primer plano el proceso que has ejecutado en el paso 57 (updatedb).

```
alumno@alumno-VirtualBox:~$ bg 3
[3]+ sudo updatedb &
[2] Hecho sleep 145
```

60. Lanza el comando sleep 150 directamente en segundo plano.

\$ sleep 150 &

61. Ejecuta top

62. Ejecuta ps

```

PID  USUARIO  PR  NI  VIRT  RES  SHR  S  %CPU  %MEM  HORA+  ORDEN
929  root     20   0  474028 42020 6708 R   1,0   4,1   1:19.81 Xorg
1833 alumno   20   0 1293708 136660 40692 S   1,0  13,5   4:42.59 compiz
6863 alumno   20   0  669520 35156 27708 S   0,7   3,5   0:01.31 gnome-termi
1105 alumno   20   0  123368 416 416 S   0,3   0,0   0:09.27 VBoxClient
1781 alumno   20   0  864356 14976 9380 S   0,3   1,5   0:06.02 unity-setti
6939 alumno   20   0  49080 3748 3104 R   0,3   0,4   0:00.01 top
1  root     20   0  185520 3504 2244 S   0,0   0,3   0:02.00 systemd
2  root     20   0  0 0 0 S   0,0   0,0   0:00.00 kthreadd
4  root     0 -20  0 0 0 S   0,0   0,0   0:00.00 kworker/0:0
6  root     20   0  0 0 0 S   0,0   0,0   0:00.35 ksoftirqd/0
7  root     20   0  0 0 0 S   0,0   0,0   0:00.37 rcu_sched
8  root     20   0  0 0 0 S   0,0   0,0   0:00.00 rcu_bh
9  root     rt    0  0 0 0 S   0,0   0,0   0:00.00 migration/0
10 root     0 -20  0 0 0 S   0,0   0,0   0:00.00 lru-add-dra
11 root     rt    0  0 0 0 S   0,0   0,0   0:00.02 watchdog/0
12 root     20   0  0 0 0 S   0,0   0,0   0:00.00 cpuhp/0
[4] Hecho sleep 150 0 S 0,0 0,0 0:00.00 kdevtmpfs

[5]+ Detenido top
alumno@alumno-VirtualBox:~$ ps
  PID TTY          TIME CMD
 6869 pts/0    00:00:00 bash
 6882 pts/0    00:00:00 sleep
 6939 pts/0    00:00:00 top
 6942 pts/0    00:00:00 ps
alumno@alumno-VirtualBox:~$ kill %4
bash: kill: %4: no existe ese trabajo
alumno@alumno-VirtualBox:~$ kill -l
 1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGBUS     8) SIGFPE     9) SIGKILL    10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE    14) SIGALRM    15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD   18) SIGCONT    19) SIGSTOP    20) SIGTSTP
21) SIGTTIN    22) SIGTTOU    23) SIGURG     24) SIGXCPU    25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF   28) SIGWINCH   29) SIGIO      30) SIGPWR
31) SIGSYS     34) SIGRTMIN  35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
```

63. Mata el proceso lanzado en el punto 60 utilizando la orden kill

\$ kill -9 6882

(9) SIGKILL (6882) del PS sleep

Alumno: Llorenç Arbiol Barelles

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.

`$ kill -9 7090`

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

`$ sudo apt-get update`

69. Mediante la utilidad apt-get instala el paquete virtualbox

`$ sudo apt-get install 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)

`$ sudo dpkg i /home/alumno/Descargas/google-chrome-stable_current_amd64.deb`

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.

`00 21 * * * root /sbin/shutdown -h now`

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 actualización.

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.

```
* /30 * * * root /sbin/who >> /home/conectados.txt
```

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_p54tesp1.pdf y

lo deje en el directorio Descargas de tu \$HOME (mandato wget).

```
* * 1,8,15,21,28 * * root cd /home/alumno/Descargas ;wget  
http://www.aemet.es/documentos\_d/enportada/20180407160748\_p54tesp1.pdf
```

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

```
$ sudo status 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.

```
$ sudo service lightdm start
```

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.

```
sudo nano /etc/bash.bashrc
```

Se añade

```
alias cls='clear'
```

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).

```
nano /home/alumno/.bashrc
```

Se añade “echo “Bienvenido a Linux” “

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.

Alumno: Llorenç Arbiol Barelles

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

```
$ sudo apt-get install webmin
```

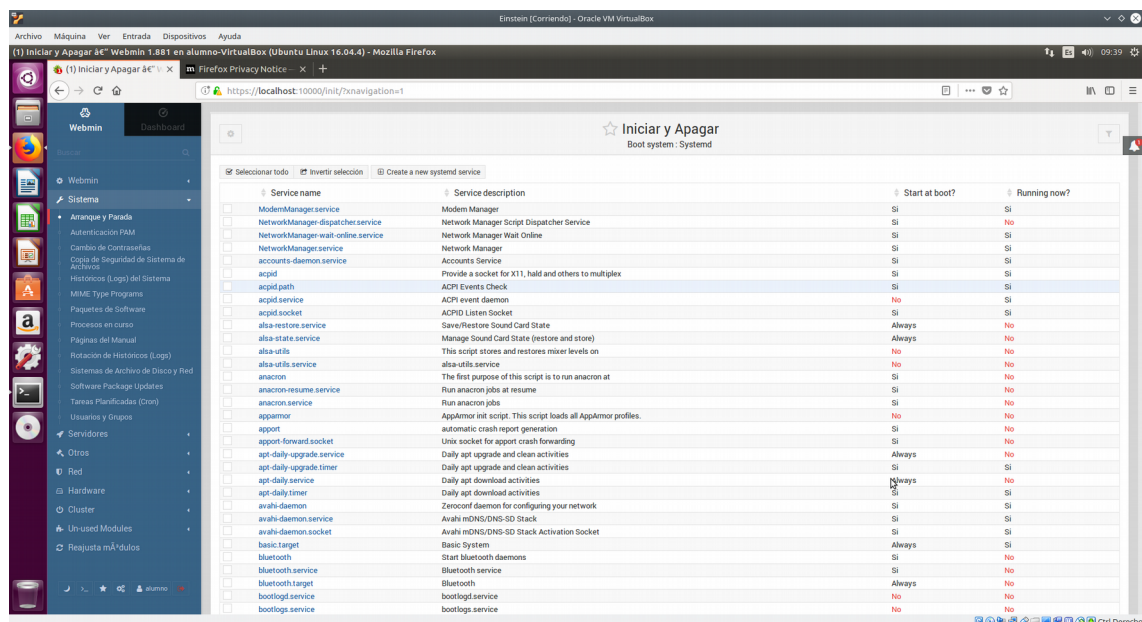
85. Abre al navegador y accede a

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.

Alumno: Llorenç Arbiol Barelles

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.

`$ sudo apt-get install openssh-server`

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.

`$ ssh 192.168.1.10`

95. Si no especificas nombre de usuario, entrarás con el mismo nombre de usuario que estás utilizando.

`$ ssh alumno@192.168.1.10`

96. Teclea logout para cerrar sesión.

97. Vuelve a entrar, pero esta vez con otro nombre de usuario. Teclea ssh usuario@dirección 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

`$ ssh alumno@192.168.1.10 -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.

`$ ssh alumno@192.168.1.10 -X gimp`

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.