

WUOLAH



postdata9

www.wuolah.com/student/postdata9



32573

sesion1.pdf

Módulo I (actualizado)



2º Sistemas Operativos



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.





**KEEP
CALM
AND
ESTUDIA
UN POQUITO**

Sesión 1:

Herramientas de administración básicas



Descarga la app de Wuolah desde tu store favorita

Índice:

1. Obtención de privilegios de root en el laboratorio

2. Administración de usuarios y grupos en Linux

2.1 Usuario administrador del sistema en Linux: root

2.2 Gestión de usuarios

2.2.1 Creación de cuentas de usuario

2.2.2 Cambio de contraseña

2.2.3 Parámetros de configuración de una cuenta

2.3 Gestión de grupos

2.4 Usuarios y grupos especiales

3. Organización del sistema de archivos y gestión básica de archivos

3.1 Organización común en sistemas de archivos tipo Linux.

3.2 Acceso a información del SO relativa a sistemas de archivos.

4. Preguntas de repaso

1. Obtención de privilegios de root en el laboratorio

Para la realización de este módulo, vamos a necesitar de muchas operaciones que requieren privilegios, por tanto, vamos a realizar una serie de pasos para entrar en el kernel, los cuales son los siguientes:

1. Copiar los ficheros de la ruta que se especifica a abajo a /tmp:

```
$ cp /fenix/depar/lsi/UML/*.gz /tmp
```

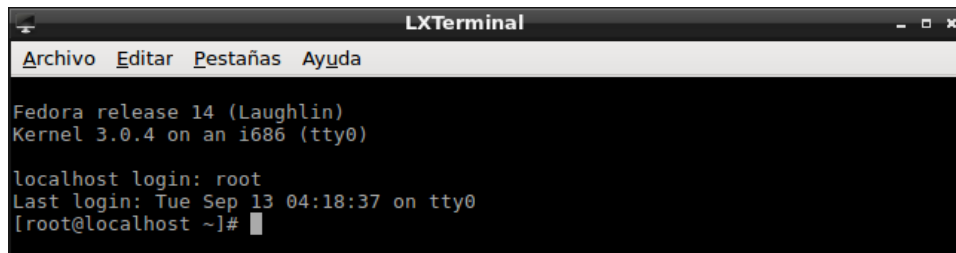
2. Descomprimir los ficheros copiados en /tmp:

```
$ gunzip /tmp/*.gz
```

3. Ejecutar en /tmp el siguiente archivo:

```
$ ./kernel32-3.0.4 ubda=./Fedora14-x86-root_fs mem=1024m
```

4. Una vez hemos ejecutado lo anterior, entramos como usuario **root**:



```
LXTerminal
Archivo Editar Pestañas Ayuda

Fedora release 14 (Laughlin)
Kernel 3.0.4 on an i686 (tty0)

localhost login: root
Last login: Tue Sep 13 04:18:37 on tty0
[root@localhost ~]#
```

Actividad 1.1 Repaso de *scripts* de bash

Crea un script de bash que automatice todos los pasos vistos en este punto y que guardarás preferiblemente en tu directorio home. Al entrar de nuevo en el sistema sólo tendrás que ejecutar el script para empezar a trabajar en modo root.

```
$ vi inibash.sh
```

```
#!/bin/bash
```

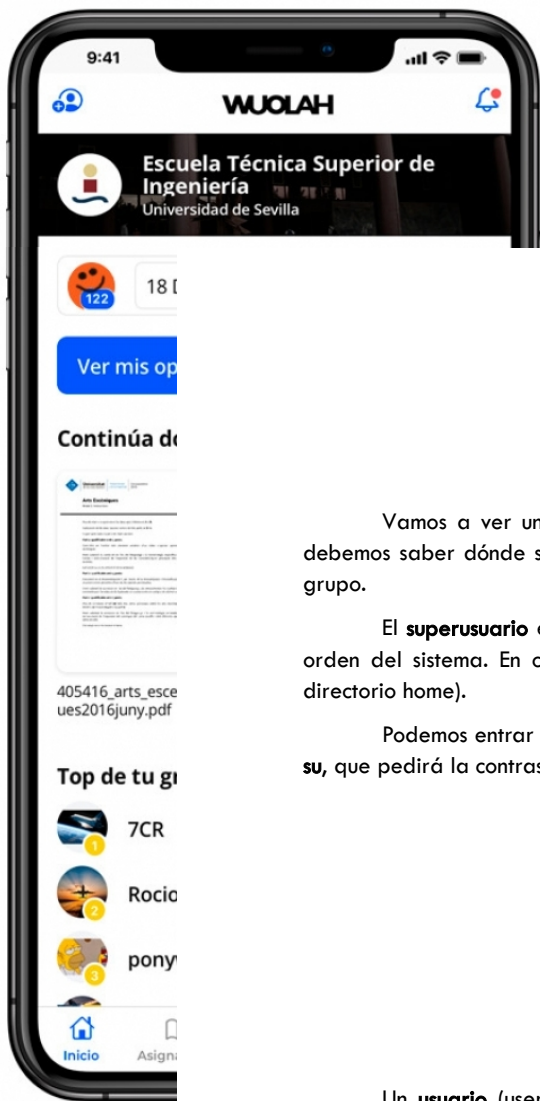
```
cp /fenix/depar/lsi/UML/*.gz /tmp
```

```
gunzip /tmp/*.gz
```

```
cd /tmp
```

```
./kernel32-3.0.4 ubda=./Fedora14-x86-root_fs mem=1024m
```

```
$ chmod u+x inibash.sh
```



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



2. Administración de usuarios y grupos en Linux

2.1 Usuario administrador del sistema en Linux: root

Vamos a ver una serie de comandos, de los cuales debemos saber en qué fichero se almacenan; debemos saber dónde se crea la carpeta de usuario, dónde se almacena la contraseña, la información del grupo.

El **superusuario** es aquel que tiene siempre todos los privilegios sobre cualquier archivo, instrucción u orden del sistema. En cualquier sistema UNIX, se identifica con el usuario **root** (grupo root y /root como directorio home).

Podemos entrar al sistema como usuario root o, si ya se ha iniciado con otro usuario, realizar la orden **su**, que pedirá la contraseña de root.

```
paula@postdata9:~$ sudo su
[sudo] contraseña para paula:
root@postdata9:/home/paula#
```

2.2 Gestión de usuarios

Un **usuario** (user) es el que trabaja en el sistema mediante una cuenta a la que accede con una identificación. En Linux, se caracteriza por:

- **username:** nombre del usuario;
- **UID (User Identifier):** es un número entero que representa al usuario en el SO, es el identificador con el que trabaja el sistema. UID de root es 0.
- **GID (Group Identifier):** es un número entero que representa el grupo principal al que pertenece el usuario. GID de root es 0.

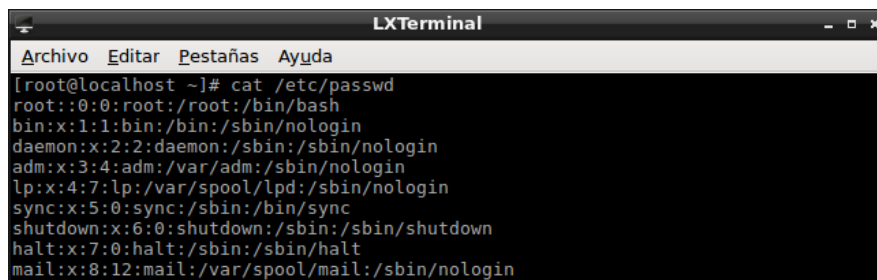
Un usuario puede pertenecer a más de un grupo; estos grupos se conocen como suplementarios.

Toda esta información es almacenada por el sistema operativo en varios archivos, los cuales son los siguientes:

- **/etc/passwd**
- **/etc/shadow**
- **/etc/group**

/etc/passwd

Contiene una lista de todos los usuarios del sistema y cierta información sobre ellos. La información aparece con la siguiente estructura:



```
[root@localhost ~]# cat /etc/passwd
root::0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
```

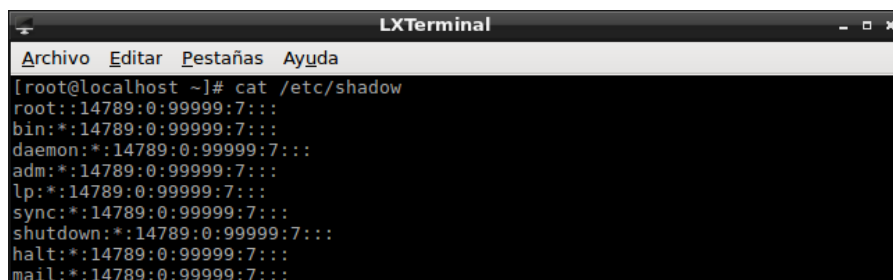
<nombre : contraseña : UID : GID : descripción : home : shell>

- **nombre** es el nombre del usuario;
- **contraseña** nos indica si el usuario tiene una contraseña asignada.
 - x: tiene una contraseña asignada;
 - si está vacío, no tiene una contraseña asignada.
- **UID** identificador del usuario. Este campo no puede estar vacío, ya que es necesario para que el sistema identifique al usuario;
- **GID** identificador del grupo al que pertenece;
- **descripción** es una descripción opcional, como el nombre completo del usuario, o alguna característica importante. Puede ir vacía;
- **home** es el directorio home del usuario;
- **shell** asociado al usuario.
 - Podemos modificar el shell del usuario a uno de los shells permitidos que se encuentren en /etc/shells. Podemos cambiarlo manualmente en este archivo o con la orden **chsh**.
 - Para denegar la entrada al sistema a un usuario, en este campo el shell se establece a /bin/false ó /sbin/nologin.

Los campos que deben ir necesariamente rellenos son: el nombre, el UID y la ruta /home.

/etc/shadow

Almacena información sobre las contraseñas de los usuarios encriptadas y sobre los cambios de las contraseñas. La información aparece con la siguiente estructura:



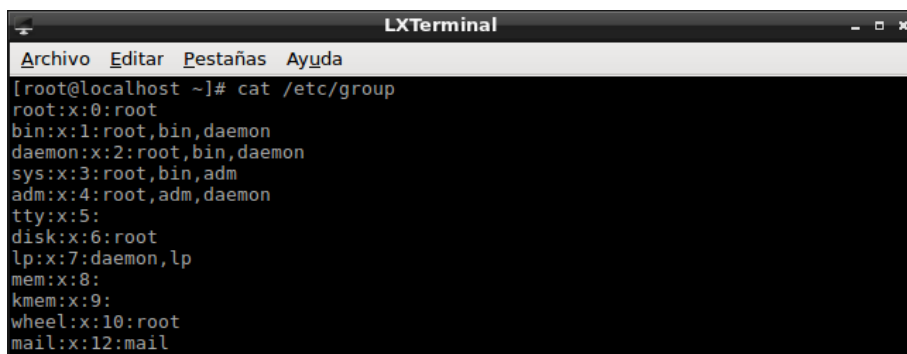
```
[root@localhost ~]# cat /etc/shadow
root:!:14789:0:99999:7:::
bin:!:14789:0:99999:7:::
daemon:!:14789:0:99999:7:::
adm:!:14789:0:99999:7:::
lp:!:14789:0:99999:7:::
sync:!:14789:0:99999:7:::
shutdown:!:14789:0:99999:7:::
halt:!:14789:0:99999:7:::
mail:!:14789:0:99999:7:::
```

<nombre : contraseña : cambio : mínimo : máximo : aviso : inactivo : vencimiento>

- **nombre** es el nombre del usuario;
- **contraseña** es la contraseña encriptada. Cuando la desencriptemos, obtendremos la propia del usuario;
- **cambio** es la fecha del último cambio de contraseña;
- **mínimo** es el mínimo número de días que han de pasar para poder cambiar la contraseña;
- **máximo** es el número de días máximo que puede estar con la misma contraseña sin cambiarla;
- **aviso** es el número de días antes de que expire (maxlife) la contraseña, en el que se le avisa al usuario de que su contraseña debe ser cambiada;
- **inactivo** es el número de días después de que la contraseña expire en la que esa cuenta es deshabilitada;
- **vencimiento** es la fecha en la que expira la cuenta y se deshabilita de forma automática.

/etc/group

Contiene la información de los grupos y usuarios miembros de dichos grupos.



```
LXTerminal
Archivo Editar Pestañas Ayuda
[root@localhost ~]# cat /etc/group
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon
tty:x:5:
disk:x:6:root
lp:x:7:daemon,lp
mem:x:8:
kmem:x:9:
wheel:x:10:root
mail:x:12:mail
```

<nombre : contraseña : GID : miembros>

- **nombre** es el nombre del grupo;
- **contraseña** nos indica si tiene una contraseña asignada;
 - x si tiene una contraseña asignada;
 - si está vacío no tiene una contraseña asignada;
- **GID** identificador del grupo, con el que el sistema lo reconoce;
- **miembros** muestra los usuarios que pertenecen a ese grupo.

Manualmente:

Para crear un usuario de forma manual, debemos:

1. Saber el **nombre** del usuario y los **grupos** a los que va a pertenecer.
2. Introducir los datos en **/etc/passwd** y **/etc/group** según hemos visto antes, con una x en el campo de la contraseña (se la asignaremos después).
3. Asignarle una **contraseña**.
4. Establecer los **parámetros** de envejecimiento de la cuenta, los parámetros de shadow.
5. Crear el directorio **/home** del usuario con el usuario, grupos y permisos correspondientes.
6. Copiar los archivos de inicialización del directorio **/etc/skel**:
 1. **.bash_profile**: se ejecuta en cada inicio de sesión del usuario y en él se indica la configuración del principio de la sesión;
 2. **.bashrc**: se indican los programas o scripts a ejecutar y se ejecuta cada vez que se ejecuta una shell;
 3. **.bash_logout**: se ejecuta cuando el usuario sale del sistema y se indica la configuración cuando salimos de la sesión;
7. Establecer otras facilidades, ejecutar cualquier tarea de inicialización propia del sistema.
8. Probar la cuenta nueva.

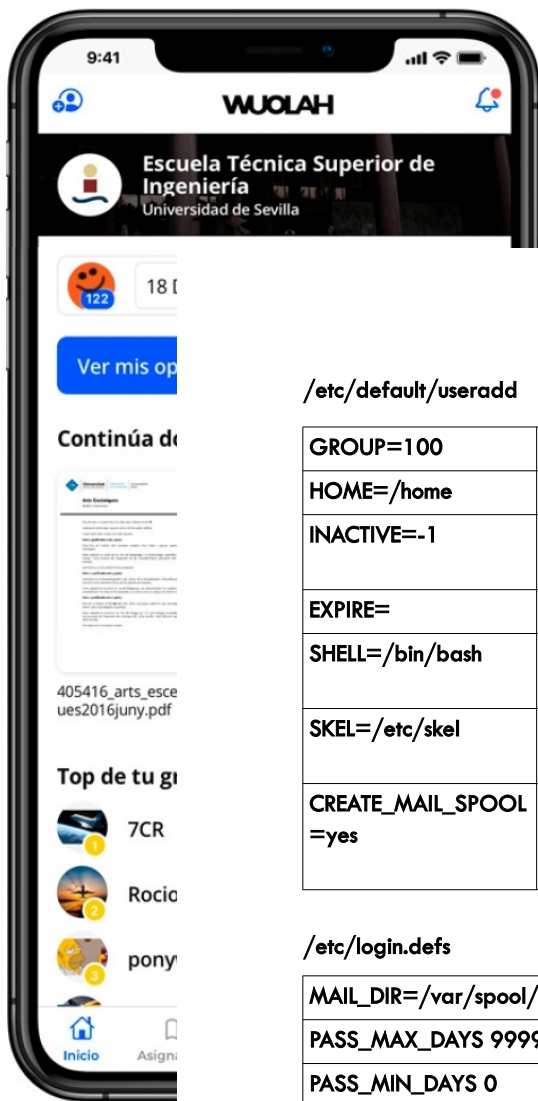
Automáticamente:

Todo esto se puede realizar de forma automática con:

- **\$ useradd <user_name>**
- **\$ adduser <user_name>**

Si ejecutamos las órdenes anteriores sin argumentos, nos muestra una lista con las opciones que podemos utilizar.

Al hacerlo de forma automática, se establecen unos valores por defecto, que se pueden consultar en los archivos **/etc/default/useradd** y **/etc/login.defs**.



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



/etc/default/useradd

GROUP=100	el GID por defecto es 100;
HOME=/home	el home por defecto es /home;
INACTIVE=-1	número de días que la cuenta debe permanecer inactiva después de su creación; al ser -1 supone que nunca está inactiva;
EXPIRE=	fecha(aaaa-mm-dd) en la que la cuenta debe expirar;
SHELL=/bin/bash	shell por defecto del usuario es /bin/sh (aunque puede variar según la distribución de Linux);
SKEL=/etc/skel	directorio desde donde se copiarán los archivos de perfil de usuario predeterminados al directorio de inicio del usuario;
CREATE_MAIL_SPOOL=yes	esta opción asegura que un nuevo usuario tendrá un directorio de su nombre de usuario en /var/main donde el proceso de correo puede almacenar mensajes de correo.

/etc/login.defs

MAIL_DIR=/var/spool/mail	directorio en el que se almacenan los mails;
PASS_MAX_DAYS 99999	número máximo de días que una contraseña puede ser usada;
PASS_MIN_DAYS 0	mínimo número de días permitido entre cambio de contraseñas;
PASS_MIN_LEN 5	longitud mínima de una contraseña aceptable;
PASS_WARN_AGE 7	número de días en el que se avisa antes de que caduque la contraseña;
UID_MIN 500	valor mínimo del UID por defecto;
UID_MAX 60000	valor máximo del UID por defecto;
GID_MIN 500	valor mínimo del GID por defecto;
GID_MAX 60000	valor máximo del GID por defecto;
CREATE_HOME yes	crea el /home por defecto. Con la opción -m en useradd, podemos deshabilitarlo;
USERGROUPS_ENAB yes	si no existen miembros de un grupo, esto permite a userdel eliminar el grupo de usuarios;
ENCRYPT_METHOD SHA512	para encriptar la contraseña.

Otras órdenes para la gestión de cuentas de usuario:

- **usermod <user_name>**: modifica la cuenta de usuario ya existente;
- **userdel <user_name>**: elimina la cuenta de un usuario, excepto su directorio /home. Para eliminar /home, le añadimos la opción -r;
- **newusers <archivo>**: crea cuentas de usuarios utilizando la información introducida en un archivo .txt con el formato de /etc/passwd;
- **system-config-users**: herramienta en modo gráfico.

Actividad 1.3 Creación de usuarios

1. Utiliza el manual en línea para leer la sintaxis completa de la utilidad para creación de cuentas y crea dos o tres usuarios en tu sistema cambiando alguno de los valores por defecto.

\$ vi users

```
user1:x:2506:2506:"User 1":/home/usuario1:/bin/bash
```

```
user2:x:2507:2507:"User 2":/home/usuario2:/bin/bash
```

\$ newusers users

```
LXTerminal
Archivo Editar Pestañas Ayuda
[root@localhost ~]# newusers users
[root@localhost ~]# tail -2 /etc/passwd
user1:x:2506:2506:"User 1":/home/usuario1:/bin/bash
user2:x:2507:2507:"User 2":/home/usuario2:/bin/bash
[root@localhost ~]# tail -2 /etc/group
user1:x:2506:
user2:x:2507:
[root@localhost ~]#
```

Comprobamos en los ficheros /etc/passwd y /etc/group que se nos ha creado ambos usuarios con los atributos que le hemos dado.

2. Elimina alguno de ellos y comprueba que “rastros” ha dejado la cuenta recién eliminada en el sistema.

\$ userdel user1

en /etc/passwd ya no está

en /etc/group tampoco está

```
LXTerminal
Archivo Editar Pestañas Ayuda
[root@localhost ~]# userdel user1
[root@localhost ~]# tail -2 /etc/passwd
smmsp:x:51:51:/var/spool/mqueue:/sbin/nologin
user2:x:2507:2507:"User 2":/home/usuario2:/bin/bash
[root@localhost ~]# tail -2 /etc/group
smmsp:x:51:
user2:x:2507:
[root@localhost ~]#
```

\$ ls /home

vemos que tenemos usuario1 y usuario2, con lo que el home no se nos ha eliminado

```
[root@localhost ~]# ls /home
usuario1 usuario2
[root@localhost ~]#
```

3. Entra (orden su) en el sistema como uno de estos usuarios que has creado y mira qué archivos tiene en su directorio home. La orden sudo permite cambiar el modo de trabajo a modo root específicamente para ejecutar una orden con privilegios de supervisor y tras su ejecución continuar con los privilegios del usuario que abrió la sesión.

```
$ sudo su user2
```

```
$ ls -l /home/usuario2
```

```
total 0
```

El home no tiene nada.

```
[root@localhost ~]# sudo su user2
bash-4.1$ ls -l /home/usuario2/
total 0
bash-4.1$
```

Actividad 1.4 Archivo /etc/passwd

Visualiza el archivo /etc/passwd e indica cual es el formato de cada línea de dicho archivo. Para ello también puedes consultar el man o info de Linux. ¿Quién es el propietario de este archivo y cuáles son sus permisos?

```
$ cat /etc/passwd
```

```
<nombre : contraseña : UID : GID : descripción : home : shell>
```

```
$ ls -l /etc/passwd
```

```
-rw-r--r-- 1 root root 897 Nov 11 04:59 /etc/passwd
```

Vemos que el propietario del archivo es root. User tiene permiso de lectura y escritura. Group y Others tiene solo permiso de lectura.

2.2.2 Cambio de contraseña

Para asignar una contraseña a un usuario, podemos usar la orden:

```
$ passwd <user_name>
```

Si la usamos sin argumento, cambiaría la del usuario actual.

El administrador puede cambiar las contraseñas de todos los usuarios del sistema, mientras que un usuario sólo podría modificar la suya propia.

Actividad 1.5 Archivo /etc/shadow

Visualiza el archivo /etc/shadow desde un usuario distinto al root ¿Te da algún problema? ¿Sabes por qué? Intenta averiguarlo.

```
$ cat /etc/shadow
```

```
postdata9@ei143062:~$ cat /etc/shadow
cat: /etc/shadow: Permiso denegado
postdata9@ei143062:~$
```

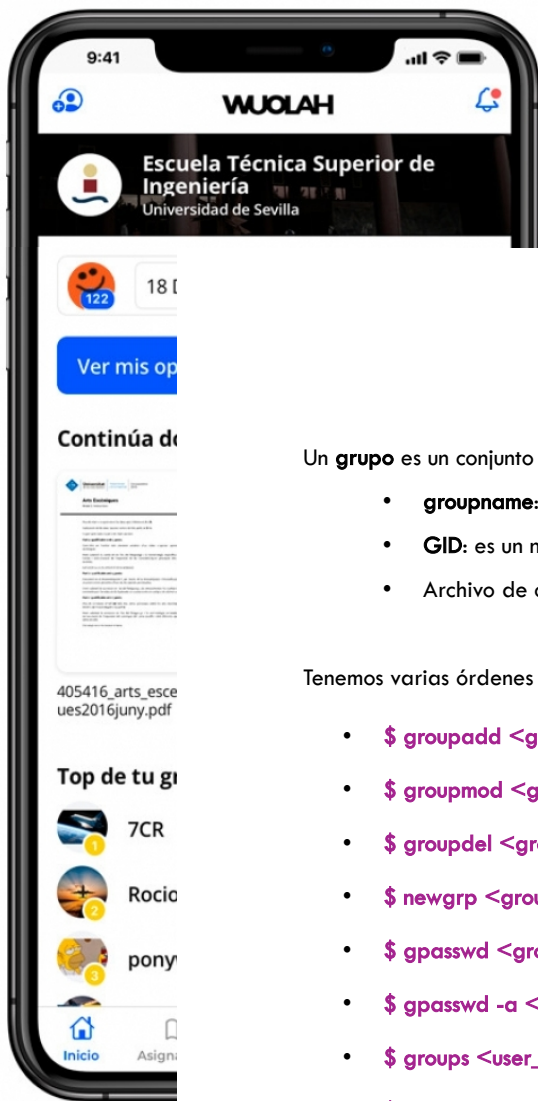
Sale permiso denegado. Se debe a que este archivo es muy delicado, y solo tiene acceso el root por lo mismo.

2.2.3 Parámetros de configuración de una cuenta

Para las cuentas de los usuarios, se pueden establecer restricciones de tiempo (llamadas de envejecimiento) de la validez de la cuenta y de la contraseña. Estos valores se encuentran en /etc/shadow.

Estos valores los establece el administrador con las órdenes **chage** o **passwd**. Algunas opciones y argumentos útiles de chage:

- **-d <dia> <user>**: fecha del último cambio de contraseña;
- **-m <dia> <user>**: número de días que han de pasar para poder cambiar la contraseña;
- **-M <dia> <user>**: número de días máximo que puede estar sin cambiar la contraseña;
- **-W <dia> <user>**: cuántos días antes de que la contraseña expire (maxlife) será avisado de ello, indicándole que tiene que cambiarla;
- **-I <dia> <user>**: número de días después de que la contraseña expire que la cuenta se deshabilitará de forma automática si la contraseña no ha sido cambiada;
- **-E <dia> <user>**: fecha en la que la cuenta expira y se deshabilita de forma automática.



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



2.3 Gestión de grupos

Un **grupo** es un conjunto de usuarios que comparten recursos o archivos del sistema. Se caracteriza por:

- **groupname**: es el nombre del grupo;
- **GID**: es un número entero que identifica al grupo en el sistema;
- Archivo de configuración **/etc/group**.

Tenemos varias órdenes relacionadas con la gestión de grupos:

- **\$ groupadd <group_name>**: crea un nuevo grupo;
- **\$ groupmod <group_name>**: modifica un grupo existente;
- **\$ groupdel <group_name>**: elimina un grupo;
- **\$ newgrp <group_name>**: cambia de grupo activo;
- **\$ gpasswd <group_name>**: asigna una contraseña a un grupo;
- **\$ gpasswd -a <user_name> <group_name>**: añade un usuario a un grupo;
- **\$ groups <user_name>**: muestra los grupos a los que pertenece un usuario;
- **\$ id <user_name>**: muestra el identificador del usuario y los grupos a los que pertenece;
- **\$ grpck**: comprueba la consistencia del archivo de grupos.

Actividad 1.6 Creación de grupos.

1. Crea un par de grupos y asígnaselos a algunos de los usuarios de tu sistema.

```
$ gpasswd -a user2 ejercicio
```

```
$ tail -1 /etc/group
```

```
[root@localhost ~]# gpasswd -a user2 ejercicio
Adding user user2 to group ejercicio
[root@localhost ~]# tail -1 /etc/group
ejercicio:x:2508:user2
[root@localhost ~]#
```

2. ¿Qué información devuelve la orden id si estás conectado como root?

```
$ id
```

```
[root@localhost ~]# id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
[root@localhost ~]#
```

2.4 Usuarios y grupos especiales

Un **usuario especial** (user) es aquel que no está asociado a una persona física. Podemos tener usuarios y grupos especiales.

Algunos usuarios especiales en Linux pueden ser:

- **root**: administrador del sistema;
- **bin, daemon, lp, sync, shutdown**: para poseer archivos o ejecutar servicios;
- **mail, news, ftp**: herramientas o utilidades;
- **postgres, mysql, xfs**: creados por herramientas instaladas en el sistema para administrar y ejecutar los servicios;
- **nobody/nfsnobody**: usada por NFS y otras utilidades.

Algunos de los grupos especiales del sistema:

- **root, sys, bin, daemon, adm, lp, ftp, nobody**: grupos preconfigurados por UNIX, los GID menores a 500 están reservados para estos grupos;
- **tty, dialout, disk, audio, video, cdrom**: específicos para dispositivos;
- **kernel**: grupo propietario de los programas para leer la memoria del kernel;
- **users**: grupo por defecto para todos los usuarios normales del sistema.

3. Organización del sistema de archivos y gestión básica de archivos

La organización de los archivos se presenta en una estructura jerárquica en forma de árbol, donde los nodos interiores son los directorios, mientras que los nodos finales son los archivos.

Un archivo puede ser referenciado de dos formas distintas:

1. **absoluta:** el nombre empieza por “/”;
2. **relativa:** el nombre no empieza por “/”.

En el sistema de archivos se almacena:

- **programa que contiene el kernel:** donde se carga el arranque del sistema. En Linux es “vmlinuz*” ó “vmlinuz*” y es un archivo ejecutable que contiene el kernel de Linux;
- **archivos especiales de dispositivo:** que se encuentra en el directorio /dev; algunos son: /dev/sda, /dev/sda1 y /dev/tty;
- **archivos para la transferencia de información entre procesos:** que pueden ser de tipo Socket o de tipo FIFO (cauces);
- **archivos de tipo directorio:** que soportan la estructura jerárquica del SA;
- **archivos de tipo enlace:** que son etiquetas asociadas a un archivo, es una forma de identificar el mismo contenido con diferentes nombres. En UNIX tenemos dos tipos de enlace: **enlaces duro (hard link)** y **enlaces simbólicos (soft link)**.

Un enlace simbólico es un acceso directo a otro. Es un archivo que apunta a otro, al registro del SA donde se encuentra, no almacena metadatos y permite enlazar directorios y archivos fuera del equipo.

Un enlace duro es un archivo que almacena los metadatos del fichero al que apunta. Todos los ficheros tienen un enlace duro.

Actividad 1.7 Archivo del kernel de Linux

Utilizando la orden (find) que ya conoces para la búsqueda de archivos en el sistema de archivos, anota el nombre absoluto del archivo del kernel de Linux que se ha cargado en el sistema operativo que estás usando en el laboratorio de prácticas para acceso modo root.

```
$ find / -name Fedora
```

3.1 Organización común en sistemas de archivos tipo Linux

FHS es un estándar que propone una forma sistemática de organizar toda la información que almacena un SO tipo Linux, siendo la Fundación Linux la encargada de mantener este estándar.

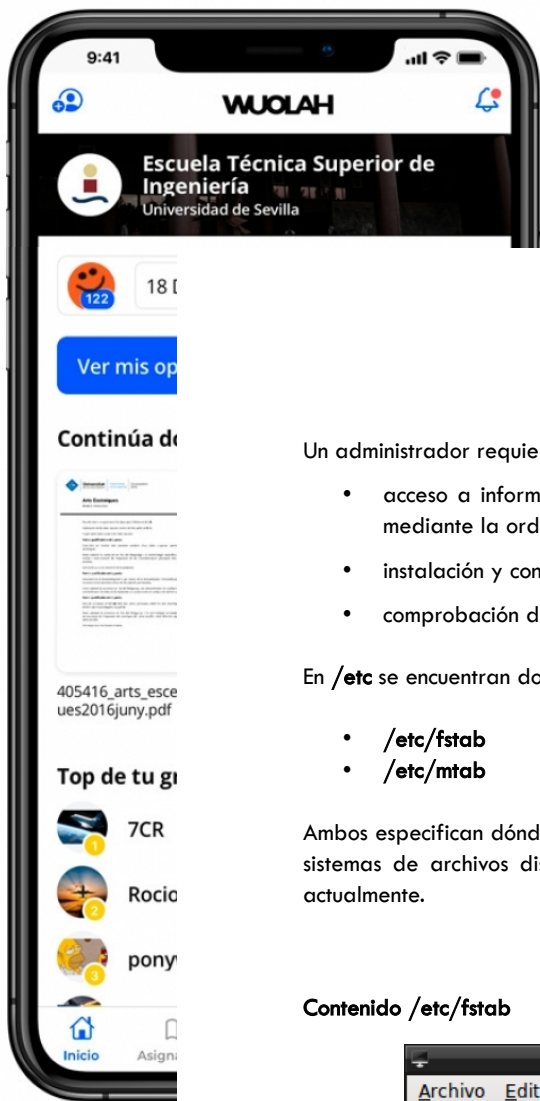
Alguno de los directorios que son interesantes conocer y que recoge este estándar son los siguientes:

- **/bin:** almacena programas de utilidad para cualquier usuario del sistema;
- **/sbin:** contiene programas de utilidad para el usuario root;
- **/boot:** archivos fundamentales para el programa **Boot Loader** (gestor de arranque);
- **/dev:** archivos especiales de dispositivo;
- **/etc:** archivos de configuración del sistema;
- **/home:** directorio de inicio de todos los usuarios, excepto root;
- **/lib:** bibliotecas necesarias para los programas de **/bin** y **/sbin**;
- **/media:** punto de montaje para los dispositivos extraíbles;
- **/mnt:** punto de montaje para los sistemas de archivos montados temporalmente;
- **/opt:** programas que no forman parte de la distribución instalada en el sistema;
- **/proc:** SA virtual que hace de interfaz con el núcleo y los procesos;
- **/tmp:** archivos temporales que no se mantienen una vez se apaga el sistema;
- **/usr:** archivos ejecutables, archivos de código fuente, bibliotecas, documentación y, en general, todos los programas y utilidades;
- **/var:** archivos cuyo contenido se espera que cambie durante el funcionamiento normal del sistema.

Actividad 1.8. Organización del SA

Un programa que se ejecuta en modo root, ¿dónde podría guardar la información temporal de forma que ésta se mantuviese entre arranques del sistema?

En el home de root, en **/var/tmp**.



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



3.2 Órdenes básicas para gestión del sistema de archivos

Un administrador requiere de una serie de funcionalidades, como las siguientes:

- acceso a información del SO relativa a SA y ampliación de la estructura jerárquica de directorios mediante la orden mount;
- instalación y configuración de nuevos dispositivos de almacenamiento y creación de SA sobre estos;
- comprobación del estado del SA y cuotas de disco.

En **/etc** se encuentran dos archivos fundamentales para los SA:

- **/etc/fstab**
- **/etc/mtab**

Ambos especifican dónde y cómo montar los dispositivos; se diferencian en que fstab muestra una lista de los sistemas de archivos disponibles actuales mientras que mtab tiene una lista de los dispositivos montados actualmente.

Contenido **/etc/fstab**

```
LXTerminal
Archivo Editar Pestañas Ayuda
[root@localhost ~]# cat /etc/fstab
#
# /etc/fstab
#
LABEL=ROOT / auto noatime 1
1
tmpfs /dev/shm tmpfs defaults 0
0
tmp /tmp tmpfs rw,mode=17
77,fscontext=system_u:object_r:tmp_t:s0 0 0
devpts /dev/pts devpts gid=5,mode
=620 0 0
sysfs /sys sysfs defaults 0
0
proc /proc proc defaults 0
0
[root@localhost ~]#
```

Contenido **/etc/mtab**

```
LXTerminal
Archivo Editar Pestañas Ayuda
[root@localhost ~]# cat /etc/mtab
LABEL=ROOT / auto rw,noatime 0 0
proc /proc proc rw 0 0
sysfs /sys sysfs rw 0 0
devpts /dev/pts devpts rw,gid=5,mode=620 0 0
tmpfs /dev/shm tmpfs rw 0 0
/tmp /tmp tmpfs rw,mode=1777 0 0
none /proc/sys/fs/binfmt_misc binfmt_misc rw 0 0
[root@localhost ~]#
```

Este archivo es una copia de **/proc/mounts** y es usado por los usuarios, mientras que **/proc/mounts** es usado por el kernel.

Estructura de ambos archivos:

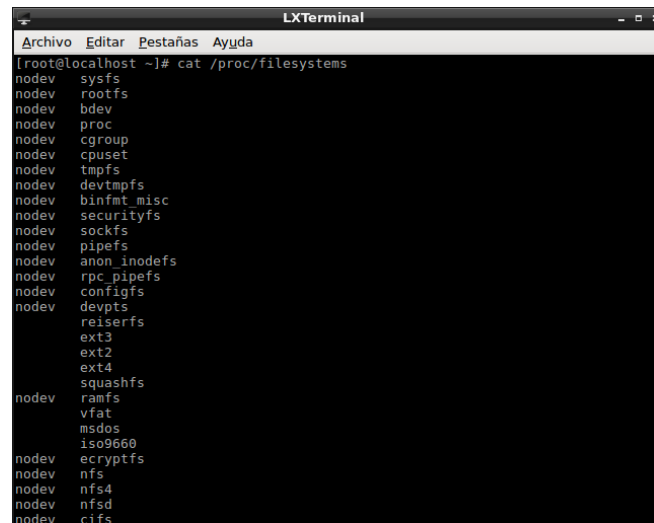
< nombre : directorio : tipo : opciones : copia_seguridad : fsck >

- **dispositivo** es el nombre del dispositivo;
- **directorio** es el punto de montaje donde se debe montar el dispositivo. Este campo no puede estar vacío y debemos crear el directorio antes de usar **mount**;
- **tipo** es el tipo de archivo del sistema. Algunos tipos importantes son:
 - **ext2, ext3, ext4**;
 - **auto**: supone que el tipo es detectado automáticamente;
 - **sysfs**: almacena recursos del sistema compilados como variables de configuración y ajustes;
 - **ntfs**: suele ser de Windows;
 - **fat**: suele ser para unidades flash;
- **opciones** describe las opciones de montaje. Los más importantes:
 - **defaults**: valores por defecto → rw, suid, dev, exec, auto, nouser y async;
 - **auto**: especifica que el dispositivo se debe montar automáticamente en el momento del arranque;
 - **noauto** especifica que el dispositivo se debe montar explícitamente;
 - **rw**: especifica que debe ser montado para lectura/escritura;
 - **ro**: especifica que debe ser montado para sólo lectura (read only);
 - **uid=number**: el propietario de los ficheros del SA es number;
 - **gid=number**: el grupo propietario de los ficheros del SA es number;
 - **suid**;
 - **nosuid**;
 - **exec**: los archivos que se encuentren podrán ejecutarse;
 - **noexec**: no se permite la ejecución de los archivos;
 - **usrquota/grpquota**
 - **user**: el usuario puede montar el dispositivo;
 - **nouser**: sólo root puede montar el dispositivo;
- **copia_seguridad**, si es 0 no se hará una copia de seguridad, mientras que si está a un número distinto de 0, realiza una copia de seguridad del dispositivo;
- **fsck** es una utilidad de los sistemas UNIX que corrige los posibles errores en el sistema. Si está a 0, el dispositivo será excluido de fsck check; si es distinto de 0, se ejecutará en el orden que establece el valor.
 - La partición raíz tiene este valor establecido a 1, para que fsck lo verifique primero.

Otro directorio del FHS importante es `/proc`, que soporta el sistema de archivos virtual `proc`. Contiene archivos de texto que permiten acceder a información de estado del sistema. Dos de sus archivos importantes son:

- **filesystems**: enumera todos los tipos de sistemas de archivos disponibles soportados por el kernel;
- **mounts**: lista los sistemas de archivos montados actualmente (manual o automáticamente tras el arranque del sistema).

`/proc/filesystems`



```

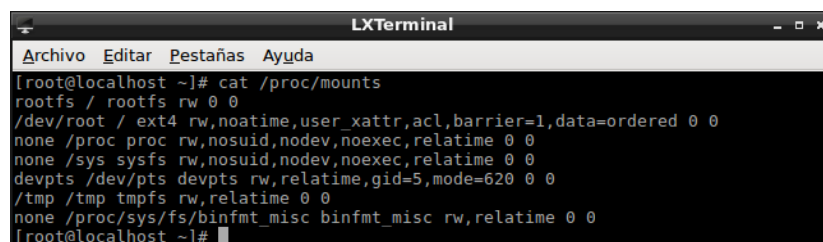
[root@localhost ~]# cat /proc/filesystems
nodev    sysfs
nodev    rootfs
nodev    bdev
nodev    proc
nodev    cgroup
nodev    cpuset
nodev    tmpfs
nodev    devtmpfs
nodev    binfmt_misc
nodev    securityfs
nodev    sockfs
nodev    pipefs
nodev    anon_inodefs
nodev    rpc_pipefs
nodev    configfs
nodev    devpts
nodev    reiserfs
nodev    ext3
nodev    ext2
nodev    ext4
nodev    squashfs
nodev    ramfs
nodev    vfat
nodev    msdos
nodev    iso9660
nodev    ecryptfs
nodev    nfs
nodev    nfs4
nodev    nfsd
nodev    cifs
```

< montado : nombre >

- **montado** es si el sistema de archivo está o no montado en un bloque de dispositivo;
 - **nodev**: no está montado en un dispositivo;
- **nombre** es el nombre de los dispositivos soportados.

`/proc/mounts`

Sigue la estructura de los archivos `/etc/mtab` y proporciona una lista de todos los montajes en uso por el sistema. Su diferencia con `/etc/mtab` es que este archivo está más actualizada y es usada por el kernel.



```

[root@localhost ~]# cat /proc/mounts
rootfs / rootfs rw 0 0
/dev/root / ext4 rw,noatime,user_xattr,acl,barrier=1,data=ordered 0 0
none /proc proc rw,nosuid,nodev,noexec,relatime 0 0
none /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
devpts /dev/pts devpts rw,relatime,gid=5,mode=620 0 0
/tmp /tmp tmpfs rw,relatime 0 0
none /proc/sys/fs/binfmt_misc binfmt_misc rw,relatime 0 0
[root@localhost ~]#
```

Actividad 1.11. Archivos de información para los SAs

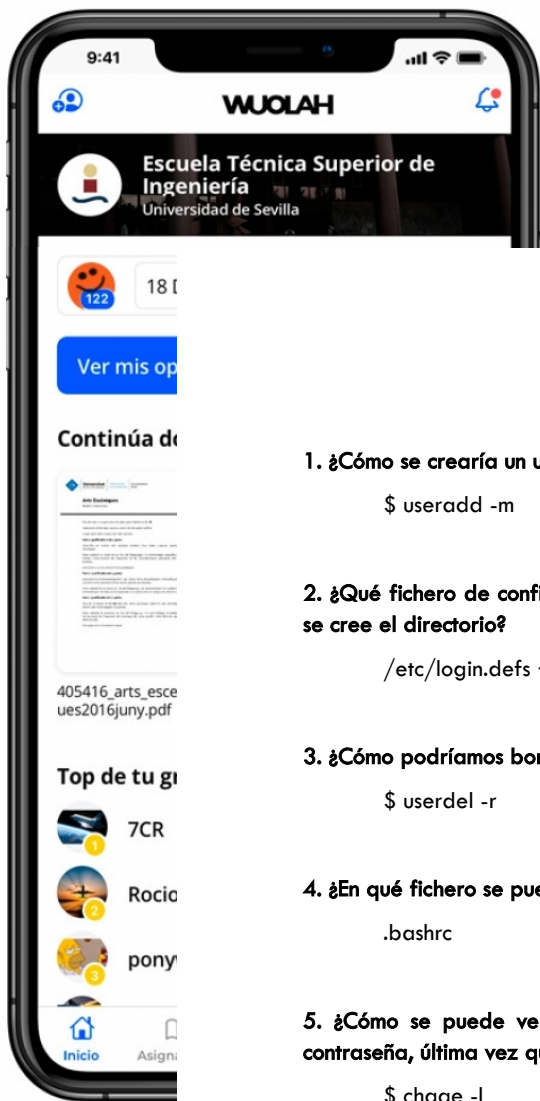
Compara la información que contienen los cuatro archivos de texto que se han presentado en este apartado (`/etc/fstab`, `/etc/mtab`, `/proc/filesystems` y `/proc/mounts`). Describe en un párrafo para qué te sirve la información que registra cada archivo.

`/etc/fstab` muestra una lista de los sistemas de archivos disponibles actuales y es leído por el demonio `init` para poder montar los dispositivos al arrancar el sistema operativo. También se utiliza su información con la orden `mount`.

`/etc/mtab` lista los sistemas de archivos actualmente montados y es utilizado por las órdenes `mount` y `umount`. Sus dos últimas columnas carecen de sentido ya que se utilizan para montar, y aquí se almacenan las que ya están montadas. Se encuentran por motivos de compatibilidad con `fstab`.

`/proc/mounts` proporciona una lista de todos los montajes en uso por el sistema. Se diferencia de `mtab` en que `mounts` está más actualizada y es usado por el kernel.

`/proc/filesystems` contiene una lista de los tipos del sistema de archivos soportados actualmente por el kernel. También es usado por `mount`.



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



4. Preguntas de repaso

1. ¿Cómo se crearía un usuario con la opción de que se cree su directorio home?

`$ useradd -m`

2. ¿Qué fichero de configuración habría que modificar para, sin añadirle el parámetro a la orden `useradd`, se cree el directorio?

`/etc/login.defs` → `#CREATE_HOME yes`

3. ¿Cómo podríamos borrar un usuario eliminando toda la información, directorio, etc., del sistema?

`$ userdel -r`

4. ¿En qué fichero se puede ver la configuración que tenemos sobre el terminal `bash`?

`.bashrc`

5. ¿Cómo se puede ver la información de “envejecimiento” de una cuenta (tiempo de expiración de la contraseña, última vez que se cambió, etc)?

`$ chage -l`

o en el fichero `/etc/shadow`

6. Responda Verdadero o Falso. Supongamos que una línea del archivo `/etc/passwd` es:

`user1:x:500:300:user1:/home/user1:/usr/bin/top`

1. El grupo inicial de `user1` tiene como identificador 500:

Falso, el GID es 300; el identificador 500 es el de UID.

2. Podría existir otro usuario cuyo directorio inicial fuera `/home/user1`:

Verdadero

3. `User1` no puede pertenecer a más grupos que al indicado en dicha línea:

Falso, ese es el grupo inicial, pero puede pertenecer a más.

4. El valor 300 no puede aparecer como cuarto campo en más líneas de `/etc/passwd`:

Falso

5. Cuando termine la ejecución de `/usr/bin/top` se lanzará una ejecución del intérprete de órdenes:

Falso

6. La contraseña está almacenada en el archivo `/etc/shadow`:

Verdadero

7. Si creamos un nuevo usuario añadiendo toda su información en el archivo `/etc/passwd`, indicando su ruta home:

Tendremos que crearle el directorio home, porque no se crea automáticamente.

8. ¿Aunque puedan aparecer las contraseñas en `/etc/passwd`, por qué no se suelen guardar ahí?

WUOLAH

Porque las contraseñas se pueden consultar por cualquier usuario, suponiendo un riesgo de seguridad.
Porque las contraseñas en ese fichero no se cifran con el algoritmo de encriptación SHA-512.

9. Supongamos que hemos ejecutado con éxito la orden:

```
useradd -U -u 115 -s /usr/bin/top user1
```

Se ha creado el usuario de nombre «user1» con identificador de usuario 115. Y cuando «user1» haga login se ejecutará /usr/bin/top y cuando este proceso termine el usuario saldrá del sistema

10. Supongamos que una línea del archivo /etc/passwd es:

```
usu1:password:701:115:Usuario1:/home/usu1:/usr/bin/top
```

El grupo principal de Usuario1 tiene como identificador numérico 115, y su contraseña encriptada es «password» y no está en /etc/shadow.

11. Imagina que un usuario tiene la siguiente línea en el archivo /etc/passwd:

```
usuario:password:0:0:Usuario:/home/usuario:/bin/sh
```

El campo «password» es la contraseña descriptada del usuario, porque la podemos leer:

Falso

12. El archivo /etc/passwd:

Podría guardar las contraseñas de los usuarios como antiguamente, pero no se suele hacer por motivos de seguridad.