

## 1. Introducción a la consola de Linux

Linux es case sensitive, diferencia entre minúsculas y mayúsculas, etc.

**usuario@debian** significa usuario del ordenador con nombre debian

**root@debian:#** significa que eres root

**usuario@debian:\$** significa que eres usuario

~ directorio home

/ directorio raíz

. directorio actual

.. directorio anterior o directorio padre

? cualquier carácter

\* cualquier cantidad de caracteres | | todo lo que hay dentro

# líneas comentadas

## Opciones básicas de la consola:

Entrar del entorno gráfico al modo comandos:

Ctrl+Alt+F1

Entrar del modo comandos al entorno gráfico:

Ctrl+Alt+F7

Mostrar ayuda de un comando:

man [nombre del comando]

Salir de las paginas man:

presionar tecla Q (de quit o quitar)

Mostrar que hace un comando:

whatis [nombre de comando]

Apagar el sistema:

poweroff

halt

Reiniciar el sistema:

reboot

Cerrar sesion:

exit

logout

Limpiar pantalla del terminal:

clear

reset

Mostrar fecha y hora:

date

cal

cal -N -y

Ver historial de comandos introducidos:

history

- Opciones:
  - [-c elimina el historial]

Cambiar colores de la terminal:

nano ~/.bashrc [descomentar las opciones que queramos colorear]

Detener la ejecucion de un comando:

ctrl+c

ctrl+z

## 2. Gestión de ficheros y directorios en Linux

Veamos los comandos principales para la gestión de ficheros y directorios en Linux.

### Mostrar contenido de directorio:

```
ls [-opciones] [ruta opcional]
```

- Opciones:
  - [-l lista detallada]
  - [-lh lista detallada con el tamaño de los archivo]
  - [-la lista detallada incluyendo archivos ocultos]
  - [-a archivos ocultos]
  - [-d lista el directorio en si, no su contenido]
  - [-R lista el contenido del directorio y subdirectorios]

- Ejemplos:

```
#Lista detallada del directorio actual
```

```
ls -lh .
```

```
#Lista detallada mostrando los ficheros ocultos del directorio /home
```

```
ls -lha /home
```

### Cambiar de directorio:

```
cd [ruta]
```

```
cd .. ir al directorio anterior
```

`cd` [sin ruta va al directorio home del que lo ejecute]

- Ejemplos:

- # Entrar en el directorio home del usuario actual

- `cd ~`

- # Entrar al directorio `/etc` desde el `/home`

- `cd ../etc`

## Mostrar en que directorio estoy actualmente:

`pwd`

## Crear directorio:

`mkdir`

- Opciones:

- `[-p]` crea todos los directorios necesarios para que construir la ruta

- Ejemplos:

- # Crear un directorio llamado "web" y dentro de este otro llamado "php" dentro de `/home` estando en el directorio home del usuario actual

- `mkdir -p ../web/php`

## Eliminar directorio:

`rmdir`

`rm -r`

- Ejemplos:

- # Crear 3 directorios y eliminar los que tengan una letra en la segunda posición

- `mkdir pico paco pista && rmdir p?co`

## Eliminar archivos:

```
rm [-opciones] [ruta opcional]
```

- Opciones:
  - [-r elimina lo que se le ponga por delante, recursivamente]
  - [-f forzar]
  - [-i pide confirmación]

- Ejemplos:

```
# Eliminar todo el contenido del directorio actual
```

```
rm -rf *
```

```
#Eliminar los ficheros .pdf del directorio actual y sus hijos
```

```
rm -rf *.pdf
```

## Crear ficheros:

```
touch nombre de fichero
```

```
echo "contenido" > nombre del fichero
```

```
echo "contenido" >> añade texto al fichero
```

- Ejemplos:

```
# Crear un fichero que contenga el siguiente texto "Freedom is linux" y ver su contenido.
```

```
echo "Freedom is linux" > fichero detexto.txt
```

```
&& cat fichero detexto.txt
```

## Mover/renombrar ficheros y directorios:

```
mv fichero1.txt fichero2.txt (renombra el fichero)
```

```
mv fichero1.txt /ruta absoluta o relativa (mueve el fichero)
```

- Opciones:
  - [-f fuerza el movimiento]

## Copiar ficheros o directorios:

`cp [origen] [destino]`

- Opciones:
  - [-r copia directorio y todo su contenido, si copias dos ficheros y uno no existe lo crea]
  - [-rf copia el contenido de los directorios y sobrescribe si alguno ya existe]
- Ejemplos:

```
# Copiar todo el contenido de /home a /home_backup

mkdir /home_backup && cp -r /home/* /home_backup
```

## Ver espacio disponible y en uso de las particiones:

`df [-opciones]`

- Opciones:
  - [-h muestra el espacio en mb y gb]

## Ver espacio empleado de carpetas:

`du [-opciones] [ruta opcional]`

- Opciones:
  - [-h muestra el espacio en "humano"]
  - [-s muestra el tamaño total del directorio]
  - [-sh las dos convidadas]

## Crear enlaces:

`ln [origen] [destino]`

- Opciones:
  - [-s crea un enlace débil]
- Ejemplos:

```
# Crear un enlace simbólico en la raíz llamado home_usu que enlace  
a /home/usuario  
  
ln -s /home/usuario /mihome
```

(En Windows podemos crear enlaces con este programa [Link Shell Extension Windows](#))

## Ver contenido de un fichero:

```
cat [-n imprime el numero de linea]
```

## Hacer echo:

```
echo [texto]
```

```
echo [* muestra el contenido de la carpeta actual]
```

## Abrir editor de texto en el interprete:

```
nano [nombre archivo]
```

# 3. Protegido: Gestión de usuarios y grupos en Linux

vamos a ver como gestionar usuarios y grupos en Linux desde la terminal.

## Crear usuarios:

```
useradd [opciones] [nombre usuario]
```

- Opciones:
  - [-g indica el grupo al que pertenece el usuario]
  - [-G indica el grupo secundario al que pertenece el usuario]
  - [-d establece el directorio home de este usuario]
  - [-s indica el shell a usar normalmente /bin/bash]
  - [-m -k crea el home del usuario y copia los archivos necesarios de /etc/skel]

## Poner contraseña a usuario:

`sudo passwd [nombre usuario]` (en ubuntu es necesario ponerle contraseña al usuario root para poder usarlo).

```
passwd [opciones] [nombre usuario]
```

- Opciones:
  - [-S ver el estado de la cuenta(P tiene contraseña, L cuenta bloqueada, NP no tiene contraseña)]
  - [-l bloquea la cuenta]
  - [-u desbloquea la cuenta]
  - [-d elimina la contraseña del usuario]
  - [-g cambia contraseña de grupo]
  - [-x establece a DIAS\_MAX el número máximo de días antes del cambio de la contraseña]
  - [-w establece los días de aviso de expiración a DÍAS\_AVISO]



- [-i establece la inactividad de la contraseña tras su expiración a INACTIVO]

## Modificar usuarios:

```
usermod [opcion] [lugar actual] [lugar nuevo]
```

- Opciones:
  - [-g indica el grupo al que pertenece el usuario]
  - [-d establece el directorio home de este usuario]
  - [-s indica el shell a usar normalmente /bin/bash]
  - [-m -k crea el home del usuario y copia los archivos necesarios de /etc/skel]
  - [-aG añadir grupo secundario sin tocar lo existente]

## Eliminar usuario:

```
userdel [nombre de usuario]
```

- Opciones:
  - [-r elimina el usuario y todos su directorio home]

## Fichero que guarda la información sobre los usuarios:

```
/etc/passwd
```

## Fichero que guarda la información sobre los grupos:

```
/etc/group
```

## Fichero que guarda las contraseñas cifradas:

```
/etc/shadow
```

## Mostrar que usuario soy:

```
whoami
```

## Mostrar información del usuario:

```
id (mas información en el fichero /etc/passwd)
```

## Crear grupo:

```
groupadd
```

## Identificarse con un usuario:

```
su [nombre de usuario]
```

# 4. Permisos en Linux

Veamos como gestionar los permisos en Linux.

## Permisos de ficheros y directorios.

Cuando listamos el contenido de un directorio nos aparecen delante de cada elemento una serie de letras del tipo «drwxr-xr-x». En la siguiente tabla especifico que significan:

drwxr-xr-x	
[Permisos]	
[tipo de elemento]	[propietario][grupo][resto de usuarios]
[-(fichero), d(directorio), l(enlace)]	[r(lectura), w(escritura), x(ejecución de un fichero, poder entrar en directorio), -(no tiene el permiso)]

## Cambiar permisos

```
chmod [permisos] [fichero]
```

### ¿Como dar permisos?

Lectura = 4

Escritura=2

Ejecución=1

Nada=0

Sumar para dar mas o menos permisos, por ejemplo 4+2+1=7 permisos de lectura, escritura y de ejecución.

Si quiero dar todos los permisos a todos los grupos

7(propietario)7(grupo)7(todos)

```
chmod 777 fichero.txt
```

`chmod [u, g, o][+, -, =][r, w, x] [nombre de fichero]` se pueden separar con comas los grupos de permisos

- Opciones:
  - [-R da los permisos todo el contenido de los directorios, subdirectorios y todo su contenido]

## **Cambiar el propietario:**

```
chown [usuario de destino]:[grupo de destino][nombre del elemento]
```

- Opciones:
  - [-R cambia de propietario todo el contenido de los directorios, subdirectorios y todo su contenido]

## **Ver a los grupos que pertenece un usuario:**

```
groups [nombre usuario]
```

# 5. Gestión de la red en Linux

Veamos como configurar la red en Linux.

## Cambiar dns:

```
/etc/resolv.conf
```

```
#Configuración básica del fichero:
```

```
nameserver [ip dns]
```

```
nameserver 8.8.8.8
```

## Cambiar dirección IP , Mascara y puerta de enlace:

```
/etc/network/interfaces
```

```
# Interface Loopback, necesaria para localhost
```

```
auto lo
```

```
iface lo inet loopback
```

```
# Configuración ip dinámica DHCP:
```

```
allow-hotplug eth0
```

```
iface eth0 inet dhcp
```

```
# Configuración ip estática:
```

```
auto eth0
```

```
iface eth0 inet static
```

```
    address 192.168.0.70
```

```
gateway 192.168.0.1  
  
netmask 255.255.255.0  
  
network 192.168.0.0
```

## Ver configuración IP etc:

```
ifconfig
```

## Desactivar una interface:

```
ifdown [interfaz por ejemplo eth0]
```

## Activar una interface:

```
ifup eth0
```

## Reiniciar el servicio de red:

```
/etc/init.d/networking restart
```

o

```
service networking restart
```

## Hacer ping

```
ping -c[numero de peticiones] URL
```

## Ver servicios activos y actividad de la red

<https://www.linuxparty.es/index.php/29-internet/8969-20-comandos-netstat-para-administradores-de-redes-linux>

## 6. Gestión de paquetes en Linux

Veamos como se gestionan los paquetes en Linux, en mi caso es una distribución basada en Debian, por lo que utilizaremos apt o aptitude.

### Cambiar/añadir/quitar repositorios:

```
/etc/apt/sources.list
```

Ejemplos de repositorios dentro de sources.list

```
deb http://ftp.debian.org squeeze main universe [puede haber main  
universe multiverse contrib non-free]
```

```
deb-src http://security.debian.org squeeze main restricted
```

### Actualizar grub:

```
update-grub2
```

### Gestor de paquetes:

```
apt-get
```

```
aptitude
```

### Actualizar lista de paquetes:

```
apt-get update
```

```
aptitude update
```

### Actualizar la distribución:

```
apt-get upgrade
```

```
apt-get dist-upgrade
```

```
aptitude safe-upgrade
```

```
aptitude full-upgrade
```

## Buscar un paquete:

```
apt-cache search [nombre paquete]
```

```
aptitude search [nombre o parte del nombre del paquete]
```

## Mostrar información de un paquete:

```
apt-cache show [nombre paquete]
```

```
aptitude show [nombre paquete]
```

## Instalar un paquete:

```
apt-get install [nombre del paquete]
```

```
aptitude install [nombre del paquete]
```

## Reinstalar un paquete:

```
apt-get reinstall [nombre paquete]
```

```
aptitude reinstall [nombre paquete]
```

## Borrar paquetes descargados:

```
apt-get clean
```

```
aptitude clean
```

```
aptitude autoclean
```



Mas opciones de aptitude:

```
install      - Instala/actualiza paquetes.
remove       - Elimina paquetes.
purge        - Elimina paquetes junto con sus ficheros de configuración.
hold         - Bloquea paquetes.
unhold       - Desbloquea un paquete.
markauto     - Marca paquetes como instalados manualmente.
unmarkauto   - Desmarca paquetes como instalados manualmente.
forbid-version - Prohíbe a aptitude actualizar a una versión específica de un p
aquete.
update       - Descarga las listas de paquetes nuevos/actualizables.
safe-upgrade - Realiza una actualización segura.
full-upgrade - Realiza una actualización, posiblemente instalando
y eliminando paquetes.
build-dep    - Instala las dependencias de los paquetes.
forget-new   - Olvida qué paquetes son "nuevos".
search       - Busca un paquete por nombre y/o expresión.
show         - Muestra información detallada de un paquete.
versions     - Muestra las versiones de los paquetes especificados.
clean        - Elimina los ficheros de paquetes descargados.
autoclean    - Elimina los ficheros de paquetes descargados obsoletos.
changelog    - Muestra el registro de cambios del paquete.
download     - Descarga los ficheros .deb para un paquete.
reinstall    - Descarga y (posiblemente) reinstala un paquete ya instalado.
why          - Muestra los paquetes manualmente instalados que necesitan un paqu
ete
              o por qué uno de los paquetes necesita un paquete concreto.
why-not      - Mostrar los paquetes manualmente instalados que llevan a un
```

## Instalar paquete .deb:

```
dpkg -i [nombre paquete]
```

## Desinstalar paquete:

```
dpkg -r [nombre paquete]
```

## Listar los archivos de un paquete:

```
dpkg -L [nombre paquete]
```

## Configurar un paquete:

```
dpkg-reconfigure [nombre paquete]
```

## Ver historial de instalaciones,etc:

```
ls -lh /var/cache/apt/archives
```

# 7. Procesos en Linux

Veamos los principales comandos para administrar los procesos en Linux.

## Ver los procesos:

`ps [-opciones]`

- Opciones:
  - [-e muestra todos los procesos]
  - [-f muestra opciones completas]
  - [-F muestra las opciones completas extra]
  - [-H muestra árbol de procesos]
  - [-c no muestra la ruta solo el nombre real]
  - [-l muestra el formato largo de varias columnas]
  - [-L lista todos los códigos de formatos]
  - [-aux muestra muchas opciones]

- Ejemplos:

`ps -el`

`ps -ef`

`ps -aux`

## Ver árbol de procesos:

`pstree [-Agu]`

## Matar procesos:

`kill [opciones] [pid o numero de identificación del proceso]`

- Opciones:
  - [-l muestra todas las posibles señales para enviar a un proceso]
  - [-numero -NOMBRE hace la acción que le mandemos]

- [-9 mata un proceso completamente]

## **Matar proceso por su nombre o matar muchos:**

```
killall [opción] (nombre del proceso)
```

- Opciones:
  - [-l muestra todas las posibles señales para enviar a un proceso]
  - [-NOMBRE hace la acción que le mandemos]
  - [-i pide confirmación]

## **Dar prioridad a un proceso que se inicia(de -20 a 19):**

```
nice[-n indicar un numero de -20 a 19 para establecer su prioridad]  
(nombre del proceso)
```

## **Cambiar la prioridad a un proceso activo:**

```
renice [-n indicar un numero de -20 a 19 para establecer su prioridad]  
(nombre del proceso)
```

## **Ejecutar procesos en segundo plano:**

```
(nombre del proceso) & nohup (nombre del proceso)
```

## **Ver procesos que se están ejecutando en segundo plano:**

```
jobs
```

## **Monitoreo a tiempo real los procesos:**

```
top
```

# 8. Dispositivos de almacenamiento en Linux

Vamos a ver como gestionar particiones y dispositivos de almacenamiento en Linux.

Para conocer el UUID de nuestras particiones podemos usar el comando `blkid` o con `ls -l /dev/disk/by-uuid/`

## Directorio donde se incorporan los sistemas de ficheros:

```
/media
```

## Mostrar particiones:

```
fdisk -l
```

## Tipos de sistemas de ficheros:

```
ext4
```

```
ext3
```

```
ext2
```

```
fat == vfat
```

```
fat16 == vfat
```

```
fat32 == vfat
```

```
ntfs == ntfs-3g (necesario instalar el programa ntfs-3g)
```

## Montar un dispositivo o partición:

Paso 1: crear directorio en `/media`

Paso 2: mostrar las particiones y seleccionar la que queremos `dev/sda?`

Paso 3: `mount [-t tipo de sistema de ficheros] (/dev/sda? partición)`  
(directorio creado en `/media/directorio`)

Ejemplo:

```
mount -t ntfs-3g /dev/sda5 /media/win7
```

## Desmontar dispositivos o particiones:

```
umount (directorio en el que esta montada)
```

## Automontar dispositivos:

Paso 1: entrar en fichero donde están las configuraciones de los distintos dispositivos(`/etc/fstab`).

Paso 2: añadir una linea que contenga lo siguiente `(partición /dev/sda*)`  
`(directorio) (tipo de sistema de ficheros)[auto,rw 0 0]`

Ejemplo:

```
nano /etc/fstab
```

```
/dev/sda5 /media/win7 ntfs-3g auto,rw 0 0
```

## 9. Comandos para filtrar en Linux

Uno de los fuertes de la terminal son los comandos para filtrar, que nos permiten filtrar resultados y buscar líneas y ficheros. Veamos los principales.

### Redireccionar salida:

```
[comando][opciones] > (fichero en el que queremos que salga)
```

Ejemplo:

```
ls -lh > lista.txt
```

### Sobrescribir contenido:

```
[comando][opciones] > (fichero en el que queremos que salga)
```

Ejemplo:

```
echo "hola" > lista.txt
```

### Añadir contenido:

```
[comando][opciones] >> (fichero en el que queremos que salga)
```

Ejemplo:

```
echo "hola" >> lista.txt
```

### Tuberías:

Las tuberías sirven para que la salida de un comando sea la entrada de otro por ejemplo: `ls -lh | wc -l`

### Crear ficheros:

```
touch nombre de fichero
```

```
echo "contenido" > nombre del fichero
```

```
echo "contenido" >> añade texto al fichero
```

Especificar la salida es opcional, por defecto es la salida estándar (monitor).

## Ordenar las lineas de un fichero:

```
sort [opciones] (fichero) (salida)
```

- Opciones:
  - [-r ordena al revés]
  - [-f trata las mayúsculas y minúsculas por igual]
  - [-g ordena de forma numérica]

## Lineas repetidas de ficheros ordenados:

```
uniq [opciones] (fichero) (salida)
```

- Opciones:
  - [-c utiliza como prefijo en cada linea su numero]
  - [-d solo imprime las lineas duplicadas]

## Ver contenido de un fichero:

```
cat [-n imprime el numero de linea]
```

## Ver las ultimas 10 lineas de un documento de texto:

```
tail [opciones] (fichero)
```

- Opciones:
  - [-f se sigue ejecutando el comando hasta detenerlo]
  - [-p no coloca los encabezamientos con el nombre de los ficheros cuando se indican varios (quiet)]
  - [-n (numero de lineas a mostrar) muestra el numero de lineas que queramos]



## Ver las 10 primeras líneas de un documento de texto:

`head [opciones] (fichero)`

- Opciones:
  - [-f se sigue ejecutando el comando hasta detenerlo]
  - [-p no coloca los encabezamiento con el nombre de los ficheros cuando se indican varios (quiet)]
  - [-n(numero de líneas a mostrar) muestra el numero de líneas que queramos]

## Imprimir por pantalla el numero de líneas, bytes, o palabras que contiene un fichero:

`wc [opciones] [ficheros]`

- Opciones:
  - [-l solo cuenta líneas]
  - [-c solo cuenta bytes]
  - [-w solo cuenta palabras]

## Cortar una línea de texto, para obtener un subconjunto en lugar de la línea completa:

`cut [opciones] [fichero]`

- Opciones:
  - [-c N-M corta desde el carácter número N hasta el carácter número M.]
  - [-c N- corta desde el carácter número N hasta el final]
  - [-c -N corta desde el principio hasta el carácter número N]
  - [-c N,M corta el carácter número N y el carácter número M]
  - [-d":" -f1 separa la línea en campos divididos por el carácter : y nos muestra sólo el primer campo]
  - [-d"-" -f3 separa la línea en campos divididos por el carácter - y nos muestra sólo el 3 campo]

- Ejemplo:

```
cut -c 3-9 /etc/passwd
```

```
cut -d ":" -f4 /etc/passwd
```

## **Invertir linea(en vez de poner las lineas de izquierda a derecha las pone de derecha a izquierda):**

```
rev
```

Seleccionar el principio de la linea: ^patrón

Seleccionar el final de la linea: patrón\$

# 10. Patrones y búsquedas en Linux

Buscar patrones en ficheros:

```
grep [opciones] (patrón) (ficheros)
```

Opciones:

- [-c devuelve sólo la cantidad de líneas que contienen al patrón.]
- [-i ignora las diferencias entre mayúsculas y minúsculas.]
- [-H imprime además de las líneas, el nombre del fichero donde se encontró el patrón. Es así por defecto cuando se hace la búsqueda en más de un fichero.]
- [-l cuando son múltiples ficheros sólo muestra los nombres de aquellos donde se encontró al patrón y no las líneas correspondientes.]
- [-v devuelve las líneas que no contienen el patrón]
- [-r busca en un directorio de forma recursiva.]
- [-n imprime el número de cada línea que contiene al patrón.]

Ejemplos:

```
grep linux /usr/share/doc
grep root /etc/passwd
grep -n error /var/log/messages
grep -i pepe /etc/passwd
grep -c root /etc/group
grep -l -r -i img /var/www/html/
ls -lia | grep "carta roja"
```

Buscar de forma recursiva en un directorio:

```
find (ruta en la que buscar) [opciones]
```

Opciones:

- -name permite especificar patrones para los nombres de los ficheros a buscar.
- -iname permite especificar patrones para los nombres de los ficheros a buscar sin tener en cuenta mayúsculas y minúsculas.
- -type permite indicar el tipo de fichero a buscar. Este puede ser d para directorios, f para ficheros regulares, l para enlaces simbólicos, b para

dispositivos de bloque, c para dispositivos de carácter, p para tuberías y s para sockets.

- -size +/- permite indicar el tamaño máximo y/o mínimo de los ficheros a buscar. Por defecto el tamaño se expresa en bloques de 512 bytes, pero si se precede este por un carácter c se referirá a bytes, k a kilobytes, w a palabras de dos bytes y b a bloques.
- -perm [+|-] permite referirse a aquellos ficheros cuyos permisos sean exactamente modo, incluya todos los de modo (signo -) o incluya alguno de los de (signo +). El valor de se expresa en forma numérica.
- -user especifica el usuario propietario del fichero
- -group especifica el grupo propietario del fichero
- -exec ; permite definir un comando a ejecutarse para cada resultado de la búsqueda. La cadena {} se sustituye por el nombre de los ficheros encontrados. El carácter ; permite indicar la finalización del comando. (Tanto {} como ; tienen que ir entre comillas o entre contrabarras para evitar que sea sustituido por el shell).

#### Ejemplos:

```
$ find /etc -name '*.conf' # busca en /etc todos los ficheros
con extensión conf

$ find / -size +10240k -size -20480k # busca los ficheros cuyo
tamaño esté entre 10M y 20M

$ find -perm +1000 -type d # busca los directorios que posean el permiso
t

$ find / -name core -exec rm -i "{}" ";"
# busca todos los ficheros que se nombren core y los borra
interactivamente. Los signos "" se utilizan para proteger de la
interpretación del shell
```

# 11. 20 ejercicios de filtrado en Linux

Vamos a ver 20 ejercicios de filtrado en Linux que vienen muy bien para practicar los comandos aprendidos.

1. ¿Cuántos usuarios tienen su directorio personal en /home?

```
cat /etc/passwd | grep /home | wc -l  
cat /etc/passwd | grep /home | cut -d ":" -f6 | wc -l  
grep /home /etc/passwd | wc -l
```

2. ¿Cual es el directorio que mas espacio ocupa dentro de /usr?

```
du -s /usr/* | sort -g | tail -1
```

3. Buscar todos los directorios en los que el usuario tenga permisos de escritura.

```
find /home -type d -perm -200 -user usuario
```

4. Buscar todos los directorios en los que el usuario tenga permisos de escritura y que los cuente.

```
find /home -type d -perm -200 -user usuario | wc -l
```

5. Numero de fotos (.jpg) que hay en el directorio /home

```
find /home -iname "*.jpg" | wc -l
```

6. Directorio Home de los usuarios que contengan en su nombre la cadena "Ana" sin tener en cuenta las mayúsculas

```
grep -i *ana* /etc/passwd | cut -d ":" -f6
```

7. Número de usuarios (cuántos) que están identificados en el sistema (usar comando who)

```
who | cut -d " " -f1 | sort | uniq | wc -l
```

8. Listado de todos los usuarios registrados ordenados por orden alfabético

```
cat /etc/passwd | cut -d ":" -f1 | sort
```

9. Guardar en el fichero musica.txt todos los archivos de música (.mp3) que son propiedad usuario "usuario".

```
find -type f -iname "*.mp3" -user usuario >  
/home/usuario/Escritorio/musica.txt
```

10. Listar todos los procesos que se estén ejecutando actualmente en el sistema y pertenezcan al root.

```
ps -ef | cut -d " " -f1 | grep root  
ps -ef | grep ^root
```

11. Guardar en el fichero nuevos\_directorios.txt el nombre de los directorios que han creado todos los usuarios que pertenecen al grupo users ordenados alfabéticamente de manera descendente.

```
find / -type d -group usuario | sort -f >  
/home/usuario/Escritorio/nuevos_directorios.txt
```

12. Obtener un listado de los 30 archivos de todo el sistema que acaben en .h, y su nombre contenga el texto "sys"

```
find / -type f -iname "*sys*.h" | head -n30
```

13. Listar sólo el GID los dos últimos grupos del sistema que contengan la letra "a"

```
grep a /etc/group | tail -n2 | cut -d ":" -f3
```

14. Listar todos los directorios del sistema que tengan activados todos los permisos.

```
find / -type d -perm -777
```

15. Muestra sólo los nombres de usuarios que tengan la contraseña activa en el sistema

```
cat /etc/shadow | cut -d ":" -f1,2 | grep -v \* | grep -v \! | cut -d ":" -f1
```

16. Buscar en todas las particiones de windows los ficheros que en su nombre contengan readme indistintamente mayúsculas y minúsculas:

```
mount -t ntfs-3g /dev/sda5 /media/windows1  
find /media/windows /media/windows1 -iname "*readme*" -type f
```

17. Listado de todos los grupos del sistema que no se utilizan como grupo secundario de ningún usuario del sistema ordenados de la z a la a sin que aparezcan repetidos.

```
grep ":$" /etc/group | cut -d ":" -f1 | sort -rf
```

18. Listado de todos los directorios que contiene el directorio «home» del usuario que ejecuta la orden y que se guarden en el fichero «lista-directorios»

```
find ~ -type d > lista-directorios.txt
```

19. ¿Qué comando utilizaría para saber cuantos (Sólo un número) repositorios están activos y utilizan la sección «multiverse»?

```
grep -v ^# /etc/apt/sources.list | grep multiverse | wc -l
```

20. Buscar todos los archivos empezando en el directorio home del usuario llamado «usuario» que tengan sólo los permisos de lectura y escritura para el usuario y permisos de lectura para el grupo y los otros.

```
find /home/usuario -type f -perm -644
```

## Servidor SSH en Debian Wheezy

<https://victorroblesweb.es/2013/12/16/servidor-ssh-en-debian-wheezy/>

