

# Modulo 1

## Sesion 1

*whoami* → nombre usuario actual, root si eres admin  
*/etc/passwd/* → informacion cuentas usuarios  
*/etc/shadow/* → Password encriptados  
*/etc/group/* → definicion de grupos y usuarios miembros

Crear cuentas de usuario

*useradd nombre\_usuario -d <home\_dir> -e <fecha\_caducidad> -b <directorio\_base>*  
*adduser*

Gestion de cuentas de usuario

Modificar usuarios:

*usermod nombre\_usuario -d <home> -e <expire\_date> -g <id grupo> -p <contraseña>*

Cambiar contraseña

*passwd <nombre\_usuario>*

Eliminar usuario

*userdel <nombreusuario>*

Crear cuentas de usuario utilizando informacion de un fichero de texto con el formato de  
*/etc/passwd*

*newusers*

Herramienta en modo grafico: *system-config-users*

Archivos de configuracion del shell Bash

*.bash\_profile* >> se ejecuta al hacer login. Variables, alias, configuracion del entorno...

*.bashrc* >> se ejecuta cada vez que iniciamos un shell. Scripts o programas

*.bash\_logout* >> se ejecuta al salir del sistema. Programas, acciones, scripts, ...

Establecer valores usuarios

*chage -d <ultimo\_dia> <usuario>* → Fecha del ultimo cambio de password

*chage -m <min\_dias> <usuario>* → n.º días que han de pasar para poder cambiar la contraseña

*chage -M <max\_dias> <usuario>* → n.º max días que puede estar con la misma pass

*chage -W <warn\_dias> <usuario>* → dias de antelacion para el aviso de cambio de pass

*chage -I <inac\_dias> <usuario>* → n.º dias despues de la expiracion de la pass para que se deshabilite la cuenta de forma automatica si no se cambia la contraseña

*chage -E <exp\_dias> <usuario>* → fecha en la que la cuenta expira y se deshabilita de forma automatica

## Gestion de grupos

*groupadd* <grupo> → crea grupo

*groupmod* <grupo> → modifica grupo existente

*groupdel* <grupo> → borra un grupo

*newgrp* <grupo> → cambia de grupo activo

*gpasswd* <grupo> → asigna una contraseña a un grupo

*gpasswd -a* <user> <grupo> → añade una contraseña a un grupo

*groups -a* <user> <grupo> → añade un usuario a un grupo

*groups* [usuario] → informa de los grupos a los que pertenece a un usuario

*grpck* → comprueba la consistencia del archivo de grupos

# Sesion 2

## Gestion basica de archivos y sistema de archivos

Nombres de directorios y tipo de informacion que contienen

/bin → programas de utilidad fundamental para **cualquier** usuario

/sbin → Programas de utilidad fundamental para el **root**

/boot → archivos fundamentales para el Boot Loader

/dev → Todos los archivos especiales de dispositivo

/etc → Archvos de configuración del sistema

/home → directorios de inicio de todos los usuarios que tienen cuenta en el sistema

/lib → bibliotecas sin las que no pueden funcionar los programas de /bin y /sbin

/media → Puntos de montaje de dispositivos extrabiles (DVD-ROM, Usb...)

/mnt → Puntos de montaje para sistemas de archivos temporales (pen drives)

/opt → programas que no forman parte de la distribucion instalada en el sistema

/proc → sistema de archivos virtual que hace de interfaz con el nucleo y los procesos

/tmp → archivos temporales que se borran al apagar el sistema

/usr → archivos ejecutables, codigos fuente, bibliotecas, doc, y programas y uilidades

/var → Archivos cuyo contenido se espera que cambie durante el funcionamiento del sistema

## Ordenes gestion sistema de archivos

*Pwd* → muestra ruta actual

*ls* → muestra archivos del directorio actual

*cd* → cambia de directorio

*mdkir*

*rmdir*

*cat*

*rm*

*cp*

*mv*

*chmod*

*touch*

*file* y *ls*

En /proc/filesystems se enumera, uno por linea, todos los tipos de sistema de archivos disponibles y en proc/mounts sistemas de archivos montados actualmente, incluyendo los manuales o automaticos

Crear un sistema de archivos

Para formatear logicamente particiones creadas con SA de tipo ext2, ext3 o ext4, utilizamos *mke2fs*. La orden completa es:

*mke2fs -L <etiqueta> -t <tipo\_ext> <ruta particion(normalmente /dev/xxx) >*

Para ver las propiedades de un SA utilizamos *tune2fs*

*tune2fs*

*-l <dispositivo (/dev/xxx)> → Muestra contenido del superbloque del SA*

*-c max-mount-counts <dispositivo> → N.º max de montajes que se pueden realizar sin comprobacion de la consistencia del SA*

*-L label <dispositivo> → Pone etiqueta al SA*

Para montar un SA y poder utilizarlo utilizamos *mount* pero antes se ha tenido que crear la carpeta donde queramos montarlo

*mkdir <directorio/en/el/que/queramos/montarlo>*

*mount -L <label del dispositivo> <ruta en donde montarlo>*

Parametros:

*-L → para montar sabiendo la etiqueta*

*-r → modo solo-lectura*

*-o <lista\_opciones> → opciones: async, atime, noatime, auto, noauto, context, defaults, dev, nodev, diratime, nodiratime, dirsyc, exec, noexec, group, iversion, noiversion, mand, nomand, ...*

## Gestores de paquetes

Ubuntu → paquetes con extension *.deb*

Fedora → paquetes con extension *.rpm*

En Fedora para instalar paquetes se utiliza el programa **yum**

*yum list* → lista paquetes disponibles en los repositorios para su instalación

*yum list installed* → lista paquetes actualmente instalados

*yum list updates* → muestra los paquetes con actualizaciones disponibles

*yum install <nombre-paquete>* → instalar paquete

*yum update* → actualiza todos los paquetes instalados

*yum remove <nombre-paquete>* → elimina el paquete y sus dependencias

Gestor de paquetes **rpm**

De forma general, rpm utiliza la estructura *rpm <opciones> <nombres-paquetes>*

*rpm -i <nombre-archivo-paquete>* → instalar nuevos paquetes

*rpm -e <nombre-paquete>* → elimina paquetes instalados

*rpm -U <nombre-archivo-paquete>* → actualiza paquetes instalados y elimina la versión previa instalada

*rpm -F <nombre-servidor-HTTP/FTP>* → Actualiza paquetes instalados descargando del servidor

*rpm -qa | grep <parte-nombre-paquete-buscado> | sort* → busca paquete instalados por su nombre o parte

*rpm -qi <nombre-paquete>* → Muestra informacion precisa del paquete instalado

*rpm -V <nombre-paquete>* → Consulta en la bd para verificar si un paquete está instalado

## Administración de cuotas

Las cuotas de disco permiten limitar el número de recursos de un SA que utilizará un usuario. Para trabajar con cuotas hay que instalar **quota**.

Los límites para usuarios y/o grupos y para bloques y/o i-nodos se establecen como:

- Límite hard → el usuario no puede sobrepasarlo. Si se sobrepasa el sistema no le permitirá usar más bloques.
- Límite soft → Se puede sobrepasar el límite durante cierto tiempo y debe ser inferior al límite hard. Después de un tiempo este límite se transforma en hard.

Para asignar cuotas a un usuario: *quota <username>* y para ver las estadísticas de las cuotas de todos los usuarios *repquota <SA>*.

# SESION 3

## Orden uptime

Muestra hora actual, tiempo que lleva en marcha el sistema, numero de users conectados y carga media del sistema de los 1,5 y 15 minutos

## Orden w

Muestra los usuario conectados y lo que estan haciendo.

## Orden time

Mide el tiempo de ejecución de un programa y muestra un resumen del uso de recursos del sistema.

Muestra el tiempo que ha estado ejecutando(real), el tiempo que lo ha ejecutado en modo usuario (user) y el tiempo usando en modo supervisor (sys).

## Orden nice y renice

Planificación por prioridades. Para establecer la prioridad de un proceso a un valor distinto del por defecto se utiliza *nice* en el rango [-20,19]. Valor negativo es el más alto y solo se puede hacer en modo root.

## Orden pstree

Visualiza el arbol de procesos en ejecucion

Las opciones son:

- a → muestra los argumentos de la linea de ordendes
- A → usa caracteres ASCII par dibujar el arbol
- G → usa caracteres VT100
- h → resalta el proceso actual y sus antepasados
- H → igual que -h pero para que el proceso que se esspecifique
- l → formato largo
- n → ordena procesos por PID del antecesor en vez de por nombre
- p → desactiva el mostrar los PIDs entre parentesis despues del nombre del proceso
- u → Si uid difiere del uid del padre, se pone entre parentesis

- V → Visualiza informacion sobre la version
- Z → Muestra el contexteo de seguridad para cada proceso

## Orden ps

Muestra información sobre los procesos en ejecución:

USER: usuario que lanzo el programa

PID: id del proceso

PPID: identificador del proceso padre

%CPU: % entre el tiempo usado realmente y el que lleva en ejecución

%MEM: fracción de memoria consumida

VSZ: tamaño virtual del proceso en KB

RS: memoria real usada en KB

TTY: terminal asociado con el proceso

STAT: estado del proceso( Running, Sleeping, sTopped, Zombie, Durmiendo, N(prioridad baja), <(prioridad alta), s (líder de sesión), l(multithread), + (foreground) y bLoqued)

ps se ejecuta normalmente con -ef : “e” → todo proceso que esté en el sistema y “f” → información completa

## Orden top

Muestra tareas que hacen más uso de la CPU en tiempo real.

## Orden mpstat

Muestra estadísticas del procesador del sistema junto con la media global de todos los datos mostrados.

La información que muestra es:

CPU (n.º procesador)

%user (% de uso de la CPU con tareas nivel usuario)

%nice (% de uso de la CPU con prioridad nice >0)

%sys (% CPU con tareas del sistema)

%iowait (% CPU desocupada)

%irq (% CPU interrupciones hardware)

%soft (% tiempo CPU con interrupciones software)

%idle (% tiempo CPU desocupada sin peticiones de disco pendientes)

intr/s (n.º interrupciones por segundo recibidas por el procesador)

Se ejecuta con *mpstat [intervalo] [numero]* donde intervalo indica cada cuantos segundos debe mostrar datos, y numero cuantos muestreos se solicitan.

## Control y gestión de memoria

### Orden free

Consume menos recursos que *top* que sirve para ver el uso actual de memoria informando del consumo de la memoria real o principal (ram) y de la memoria de espacio de intercambio (swap)

### Orden vmstat

Supervisar el sistema mostrando información de memoria pero también de procesos, E/S y CPU.

Se ejecuta *vmstat [tiempo\_muestreo] [iteraciones]*

La columna *us* muestra programas o peticiones de usuario, la columna *sy* tareas del sistema, esperas E/S, actualización de estadísticas, gestión de prioridades, etc y la columna *id* cuando no está haciendo nada.

La columna *r* muestra cuántos procesos están en cola de ejecución, la columna *wa* indica que no hay process en swap, la columna *so* indica que se está incrementando el uso del espacio de intercambio y la columna *free* la memoria principal libre se está agotando.

## Control y gestión de dispositivos de E/S

### Consulta de información de archivos

Para ver la información de los archivos de un directorio o datos de un archivo utilizamos **ls**.

Las opciones son :

- l para mostrar mas información
- n muestra ID de usuarios y grupos
- la muestra lo mismo que -l pero con archivos ocultos
- li lo mismo que -l pero con inodos
- lh muestra lo mismo que -l pero el tamaño se lee en KB, MB o GB.
- X ordena alfabeticamente por directorios
- t ordena por tiempo de modificacion
- u ordena por tiempo de acceso



-c ordena por ctime (timepo de ultima modificacion)

Para ver los metadatos de cada SA montado utilizamos **df**, donde podemos ver capacidad de almacenamiento total, espacio usado y espacio libre, punto de montaje, etc

Para ver el espacio en disco que gasta un directorio de la jerarquia de directorios utilizamos **du**. La ultima linea muestra la cantidad total de bloques de disco utilizados.

## Enlaces a archivos

Hay dos tipos de enlaces:

Enlaces simbolicos que hacen referencia al nombre de un archivo

Enlaces duros que hacen referencia a los metadatos de un archivo

Para crear enlaces duros utilizamos **ln** **<nombre\_archivo>** **<nuevo\_nombre>**

# Sesion 4

## Procesos demonios

### Atd

Demonio que ejecuta una orden en momento de tiempo especificado. Se utiliza según la finalidad con:

at -f <script> -t <tiempo> -q <cola> → ordenar la ejecución de ordenes a una determinada hora

atq → consultar lista de ordenes

atrm → eliminar ordenes

batch → ordenar la ejecucion de ordenes que se ejecutaran cuando la carga del sistema sea baja

En /etc/at.deny y /etc/at.allow se puede establecer que usuarios puede usar at

### Cron

Sirve para ejecutar ordenes con periodicidad.

El formato de un archivo crontab es:

*minuto          hora   dia-mes          mes   dia-semana   orden*

Los posibles valores son:

\* : cualquier valor posible

numero: activo para ese valor

dos enteros separados por guion: rango de valores

serie de numeros separados por coma: activa cualquier valor de la lista

Ejemplos

1          20          \*          \*          1-5          → a las 20:01 de lunes a viernes

0,30      \*          13          \*          5          → minuto 0 y 30 (cada media hora) el viernes y el dia 13 del mes

Para crear un demonio crontab se crea un fichero con su formato y se ejecuta **crontab <fichero>**

Para consultar la lista de trabajos actual se utiliza **crontab -l**

Para establecer variables de entorno en el crontab se escribe en el fichero crontab <nombre>=<valor> (SHELL , LOGNAME y HOME ya están incluidas)

Para filtrar los usuarios que pueden o no pueden ejecutar la orden crontab se modifican los archivos /etc/cron.deny y /etc/cron.allow