

MANUAL DE LINUX

Entrada y salida del sistema:

- login: Petición de usuario y contraseña
- \$: Usuario normal
- #: Superusuario
- exit: salir del Terminal
- logout: terminar sesión
- reboot: reiniciar sistema
- halt: apagar equipo

Información de Linux

- o HOW TO: Documento corto para solucionar un problema Linux.
- o FAQ: respuestas a dudas más frecuentes
- o Libros LPD: libros o manuales de Linux

Ayuda de comandos

- man “comando” => man ls
- info “comando” => info ls
- “comando” -help => ls -help

El prompt del sistema

- \$: usuario normal
- #: superusuario
- [usuario@maquina directorio]
- Se puede cambiar con la variable de entorno \$ PS2

Sintaxis de una orden: \$ orden opciones argumentos

Ej: ls -al

Comandos relacionados con archivos y directorios

- Directorio raíz => /
- Directorio actual: . ls.
- Directorio padre: .. ls..
- ls: muestra el contenido de un directorio
- ls -la: listado largo con ficheros ocultos

Caracteres comodín

- *(asterisco) => cualquier cadena de caracteres
- ? => sustituye a un único carácter en el nombre de un archivo
- [abc] => cualquier carácter de la lista

- rm => borra ficheros o directorios
- pwd => indica directorio de trabajo actual
- mkdir => crea directorios
- rmdir => borra directorios
- cd o chdir => cambia directorio
- gedit => editor de textos
- file “archivo” => tipo archivo
- cp origen destino => copia ficheros y directorios
 - cp texto.txt texto1.txt
 - cp texto.txt /home/Juan

Cuando un usuario copia un archivo se convierte en propietario

-mv origen destino => mueve ficheros y cambia el nombre de los ficheros

Mueve => mv proyecto /tmp

Cambia nombre => proyecto proyecto1

Mueve => mv /home/Juan/texto.txt /home/Luis

Cambia nombre al directorio => mv /home/juan/documentos /home/Juan/documentos2

-ln => crea enlaces entre ficheros (link)

- Duros: Es un nombre más para un fichero. Se elimina el fichero cuando se elimina el último sobrenombre.

Ej: ln /home/luis/foto.jpg fotografia

-Simbólicos: contiene un camino de un fichero. Apuntan a ficheros en el mismo sistema de archivos o en otros.

Ej: ln /home/Isabel/texto

- touch “archivo” => actualiza la fecha de un archivo

- cat “archivo” => visualiza el contenido de un archivo

- grep => busca caracteres en un fichero o ficheros

Ej: grep “hola” prueba.txt

- find => busca el directorio donde se encuentra un archivo

Ej: find / -name proyecto.txt

- user “nombreusuario” => pertenece usuario

- group “nombregroup” => grupo

- mtime “numerodias” => modificado hace nº de días

- head => muestra la parte inicial de un archivo de texto

- tail => muestra la parte final de un archivo de texto

- wc => muestra el nº de bytes, palabras y líneas de un archivo

- less => muestra archivos por pantalla paginando la salida, permite volver atrás

- more => muestra archivos pantalla a pantalla

- cmp => compara archivos

- nl => muestra el archivo numerado en líneas

- csplit => divide un archivo en secciones determinadas por líneas de texto

- cut => imprime secciones de líneas de un archivo

Ej: cut --bytes = 1 suma

- diff => busca diferencias entre los archivos

Ej: diff suma1.txt suma2.txt

- tac => visualiza un fichero de texto con las líneas invertidas

Ej: tac suma.txt

- uniq => borra las líneas duplicadas de un archivo ordenado

Ej: uniq suma.txt

Comandos varios

- cal => muestra calendario

- clear => borra pantalla

- date => proporciona o ajusta la fecha y hora del sistema

- dmesg => permite ver los mensajes de inicio del sistema
- env => muestra el entorno actual de trabajo con todas sus variables
- time => tiempo que tarda en ejecutarse un comando
- echo => muestra texto/contenido de una variable

GESTION DE GRUPOS Y USUARIOS

- groupadd -g gid grupo => crea un grupo
 - GID: Identificador de grupo
 - >500: grupos normales
 - <500: grupos del sistema
- Visualizar los grupos: cat /etc/group
- groupmod -g gid -n groupname grupo => modifica grupo
- groupdel grupo => elimina grupo
- useradd usuario => crea un usuario
 - d => home
 - s => shell
 - g => grupo principal
 - p => password
 - u => uid
 - G => otros grupos

Ficheros	/etc/passwd => información de usuarios
	/etc/shadow => información de usuarios segura
	/etc/default/useradd => información por defecto
	/etc/skel => ficheros pro defecto en \$HOME (esqueleto)
	/etc/login.defs => definiciones por defecto del login

- usermod usuario => modifica las opciones de un usuario
Utiliza las mismas opciones que useradd
- userdel usuario => elimina usuario
-r => borra directorio
- users => usuarios que han iniciado sesión
- passwd [usuario] => cambia contraseña del usuario
- groups [usuario] => grupos del usuario
- id [usuario] => información del usuario
- finger [usuario] => información adicional
- chfn [usuario] => cambia información finger
- chsh => cambia el intérprete de comandos
/bin/bash /bin/csh /bin/zsh
/bin/bashz /bin/bsh
- whoami => usuario actual
- su => cambia a otro usuario (por defecto root)

PERMISOS DE ARCHIVOS Y CARPETAS

- Cada fichero posee un propietario (UID) y cada usuario pertenece al menos a un grupo (GID)
- El usuario que crea un objeto es su propietario
- El sistema asigna permisos a los objetos a 3 niveles:
 - Propietario: u
 - Grupo al que pertenece el propietario: g
 - Todos los demás: o

rwX	rwX	rwX
u.prop	g.prop	resto

- A cada nivel asigna tres tipos de permisos
 - r => lectura.....visualizar
 - w => escritura.....modificar contenido
 - x => ejecución.....navegar en el directorio
- Se visualizan con la opción `ls -l`
- `chmod` => asignación de permisos a un fichero (root, propietario)

Octal	Binario	Permisos
0	000	ninguno
1	001	- - x
2	010	- w -
3	011	- w x
4	100	r - -
5	101	r - x
6	110	r w -
7	111	r w x

- `chmod 751 texto.txt`

Usuario => 111 => r w x
 Grupo => 101 => r - x
 Otros => 001 => - - x

- Clases de usuarios:
 - u => usuario propietario
 - g => grupo
 - o => otros

Cambiar permisos

- Modo absoluto
 - `chmod 652 notas => r w - r - x - w -`
- Modo simbólico
 - `chmod claveusuario (+/-) permisos fichero/s`

Ej: `chmod uo+x datos`
`chmod o+rwX datos`
`chmod o-rwX`

- La primera letra de los permisos
 - d => directorio
 - l => link
 - (guión) => fichero

- Otros permisos

- STICKY BIT (bit pegajoso)
 - Fichero => se almacena en swap al ejecutarse frecuentemente.
 - Directorio => solo el propietario del fichero o directorio pueden borrar el fichero del directorio. Empleado en archivos temporales de acceso de escritura general.
 - Se activa con: `chmod 1000 objeto`

Se inserta una “t” en el permiso de ejecución de otros.

- T => solo tiene el permiso sticky bit
- t => permiso de ejecución y sticky bit

- SUID => El usuario que ejecute el fichero tome la personalidad, durante la ejecución, del usuario propietario del archivo

- Se activa con: `chmod 4000 objeto`
- Aparece con una “s” en la posición del permiso de ejecución del usuario propietario
- S => solo el permiso SUID
- s => ejecución y SIUD

- SGID => al ejecutar el fichero nuestro GID toma el valor del identificador del grupo propietario del fichero

- Se activa con: `chmod 2000 objeto`
- Aparece una “s” en la posición del permiso de ejecución del grupo
- S => solo SGID
- s => ejecución y SGID

- `chown nuevo-propietario fichero(s)` => cambia el propietario

Ej: `chown luis *.bat`

Debemos ser superusuario (root)

usuario:grupo fichero

Ej: `chown juan:alumnos texto.txt`

- `chgrp nuevo-grupo fichero` => cambia el grupo de archivo(s)

- debemos ser superusuario(root)
- Ej: `chgrp alumnos libro.txt`

- `newgrp` => inicia sesión a un nuevo grupo

- se sale con `exit`
- debemos ser superusuario
- se consulta con `groups`
- Ej: `newgrp topos`

- `umask` => máscara de creación de ficheros

- `umask` => información de la máscara
- `umask 0002` => cambia la máscara

- Se establece:

- Para ficheros: $0666 \& \sim 0002 = 0664$ (rw- rw- r--)
- Para directorios: $0777 \& \sim 0002 = 0775$ (rwx rwx r-x)
- Definido en el script `/etc/bashrc`

GESTION DE PROCESOS

Linux => S.O. multitarea

- Estado de un proceso

- D => un interruptible sleep (procesos Input Output)
- S => dormido (en cola)
- R => ejecutándose (en cola de ejecución)
- T => parado (stopped)
- X => muerto (dead)
- Z => zombie => defunct process (en proceso de defunción)

- PID => número de proceso
- PPID => número de proceso padre
 - Los procesos se organizan jerárquicamente (Árbol de procesos)
- PGID => identificador de un grupo de procesos.
 - Si PGID es igual que PID de una Terminal => procesos en primer plano
 - Si PGID es igual en varios procesos => pertenece a la misma tubería de procesos
 - Los generados en primer plano son sensibles a señales del teclado.
- pstree => árbol de procesos
- init => proceso padre de todos los procesos
- xinetd => proceso padre de ciertos servicios de Internet
- ps => visualiza procesos
 - A => todos los procesos
 - l => formato largo
 - a => procesos de otros usuarios
 - u => nombre usuario, hora comienzo y uso de procesos de este usuario
 - H => árbol de procesos
- kill -9 PID => mata a un proceso
- pstree => ver árbol de procesos
- top => visualiza tareas Linux en tiempo real
- free => visualiza memoria física y swap libre y buffers creados por el kernel
- uptime => tiempo que lleva el sistema ejecutándose
- w => usuarios en la máquina y sus procesos
- pgrep => muestra procesos según una selección
 - Ej: pgrep -l -u root,sshd => procesos de root y sshd
- who => muestra quién ha iniciado sesión
- uname => información del sistema (uname -a => toda)
- kill -"señal" PID => envía una señal a un proceso
 - kill -l => lista de señales
 - kill -9 PID => mata un proceso
 - kill -15 PID => termina un proceso
 - kill -19 PID => para un proceso
 - kill -18 PID => continua un proceso
- at => ejecuta un trabajo en un determinado momento
 - Ej: at -f ficheroscript 10:15
 - at -f ficheroscrip 10:15 30.11.05
- atq => ver tareas programadas
- atrm => elimina una tarea

Procesos en background

- comando & => ejecuta un proceso en background

Ej: find / -name network > b.txt &

- jobs => muestra trabajos activos
- bg %n° => envía un trabajo parado a background
- fg %n° => sigue con el proceso

Ej: find / -name squid > d.txt &

jobs

fg %1

- ldd => muestra librerías compartidas necesarias para ejecutar un programa

Ej: ldd /bin/date

- nice => ejecuta un programa con otra prioridad

Desde -20 hasta 19

- 20 => mayor prioridad

19 => menor prioridad

Ej: nice -n 5 /bin/ls

Se recomienda dar siempre una prioridad baja

- nohup comando => ejecuta el comando que ignora las señales de hanup. Continúa en segundo plano después de cerrar la sesión. Si el comando tiene salida por pantalla crea el archivo \$HOME/nohup.out.

Se debe poner al final "&" para enviarlo a background.

Ej: nohup ls-ls &

Si no es así hay que hacer:

- ctrl.+z => para el proceso
- jobs
- bg %n°

SISTEMAS DE ARCHIVOS

- df => información de la utilización del espacio de disco en sistemas de archivos
 - df -a => muestra todos los sistemas de archivos
 - df -h => formato humano (más comprensible)
- du => espacio ocupado por los archivos y directorios que cuelgan
 - du -sh directorio => en formato humano solo el directorio y no subdirectorios

Sistema de archivos => estructura situada en un área del disco denominada partición.

Un sistema de archivos puede estar montado sobre varias unidades de disco: sistema de archivos multivolumen.

-Tipos de ficheros:

- ficheros normales
- directorios y discos físicos
- enlaces
- archivos especiales => ficheros de dispositivos /dev, fifo, tuberías con nombre grupo

-Tipos de sistemas de ficheros:

En Linux existen 2 tipos de sistemas de ficheros nativos: ext3, ext2, ext, minix, xia o swap

Además soporta otros tipos:

- ext3, ext2, ext => sistemas de archivos estandar de Linux.
- msdos => compatibilidad con sistemas de archivos FAT de ms-dos (FAT16).
- VFAT => compatibilidad con sistemas de ficheros Windows (FAT32).
- ISO 9660 => cdrom.
- UMS2 => permite desde Linux que el sistema MS-DOS, usarlo como si fuera de Linux => nombres largos, propiedad, permisos, enlaces y ficheros de dispositivo.
- Hpfs => OS/2.
- nfs => sistema de ficheros de red (compartir sistemas de archivos).
- sysv => UNIX.
- proc => sistema de archivos virtual de Linu. Acceso a ciertas estructuras de datos del Kernel. El Kernel accede a estos como si fueran ficheros. No existe en el disco realmente, el Kernel crea la ilusió. Existe también información de toda la máquina.
- Devps => pseudoterminales.
- ntfs => windows NT.
- Vsbfs

Organización de los directorios:

/bin => comandos básicos del S.O., también /usr/bin
/sbin => comandos esenciales para administración del S.O. También /usr/sbin
/boot => binerios de arranque del sistema
/dev
/etc => ficheros de configuración del sistema
/home => directorio de trabajo del usuario
/root => directorio de trabajo de root
/lib => librerías básicas para trabajar con Linux
/mnt => montaje de sistemas de archivos
/proc => información del sistema
/tmp => información temporal de esa sesión
/usr => programas que no forman parte del sistema. Más básico
/var => directorios que tienden a crecer y cambiar de tamaño

- mkfs => crea un sistema de archivos sobre una partición

Ej: mkfs [-t tipo] partición
mkfs -t ext3 /dev/hda5

- mount => monta un sistema de archivos. El directorio debe de estar creado

- umount => desmonta un sistema de archivos

Ficheros /etc/fstab y /etc/mtab

- /etc/fstab => información descriptiva sobre los distintos sistemas de archivos. Es importante el orden, pues se actúa secuencialmente con mount y umount.

- device => dispositivo
- directorio => directorio de montaje. Debe existir
- tipo => tipo de sistema de archivos. Si es ignore no se monta
- opciones =>
 - auto/noauto => la partición se monta o no en el arranque
 - users/nousers => los usuarios o solo root puede montarla
 - ro/rw => solo lectura o lectura y escritura
 - exec => se pueden ejecutar los binarios de esa partición
 - default => rw, exec, auto, nousers, async.
 - async => el sistema sigue trabajando tras una petición de escritura del dispositivo.
- frecuencia => frecuencia de copias de seguridad por el comando dump : 1 ó 0
- secuencia => orden en el que se realizan chequeos de los sistemas de ficheros en tiempo de arranque. El raíz debe tener un 1 y los demás un 2, si tiene un 0 no es necesario el cheque.

- /etc/mtab => muestra sistemas de archivos montados

- fdformat "device" => formatea un disco

Ej: fdformat /dev/fd0H1440

- fdisk => para manipular particiones, se puede utilizar también "parted"

Ej: fdisk /dev/hda => m => ayuda

p => tabla de particiones

q => salir

- dd => copiar discos

Ej: dd if= /dev/fd0 of= /image

dd if= /image of= /dev/fd0

Mete todos los archivos de un disco en un fichero, haciendo con ellos una imagen que luego traspasa a un disco.

- e2label => cambia la etiqueta el disco en sistemas ext2, ext3

Ej: e2label /dev/hda3 => muestra

e2label /dev/hda3 discolinux => cambia

Sistemas Ext3, swap, RAID y LVM

- **Características Ext3** => no es necesario chequear (e2fsk) tras una caída inesperada del sistema. Características de "journal" (diario) utilizado para consistencia. Mayor velocidad puesto que los journals optimizan el movimiento de cabezales. Fácil transición ext3 => ext2 y viceversa.

- **Swap** => se recomienda el de doble de RAM. Puede ser una partición o un fichero en ext3. Se recomienda la partición.

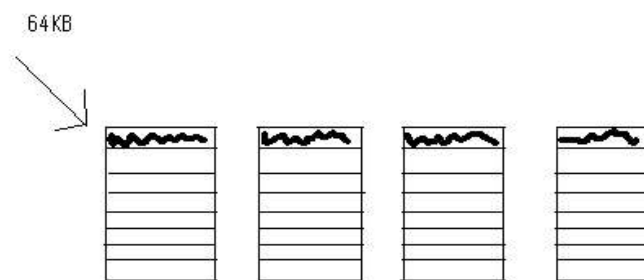
- **RAID** => conjunto de discos agrupados para mejorar rendimiento y seguridad de acceso a datos. Utilizan grupos de 32 KB y 64 KB (bandas)

- **Dos tipos:**

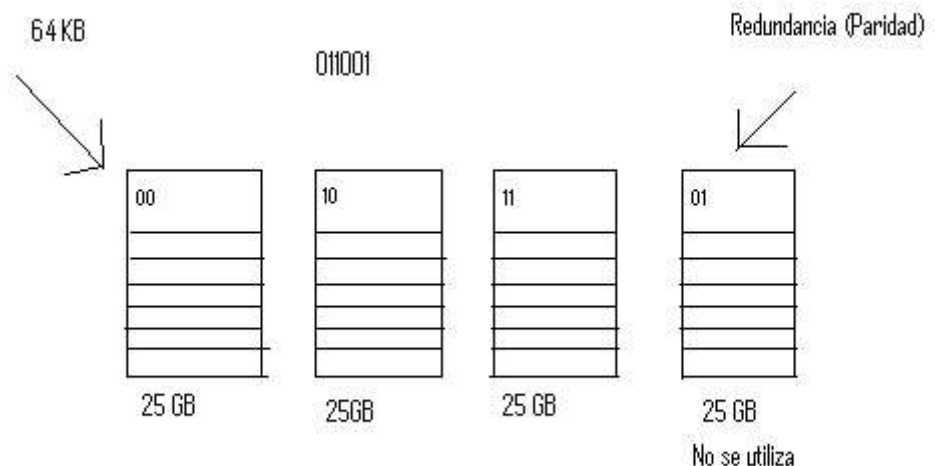
- Hardware => controladores RAID al que se conectan discos duros y es transparente al S.O o chasis hot-swap
- Software => soporte del kernel. Creado en la instalación con disk druid

- **Niveles:**

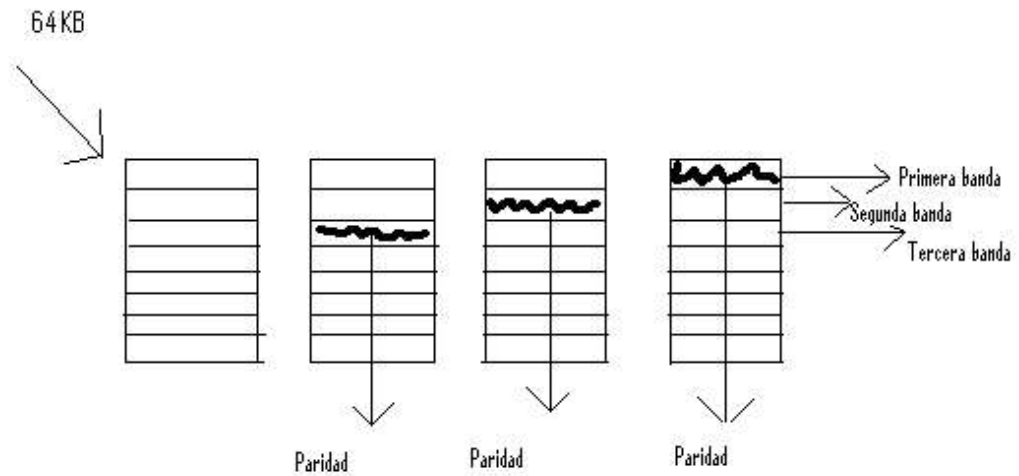
- Nivel 0 => la información se divide en grupos y se distribuye por los discos. Mejora rendimiento pero no proporciona redundancia.
Ej: Si tenemos 4 discos duros y queremos guardar un archivo de 256 KB, ese archivo se dividirá en trozos de 64 KB y se guardaría un trozo en cada disco duro.



- Nivel 1 => redundancia, escribe datos idénticos en cada uno de los discos miembros. Discos espejo
- Nivel 4 => utiliza paridad concentrada en una unidad de disco. No soportado por RED HAT.



- Nivel 5 => distribuye la paridad entre los discos. (Es más utilizado)



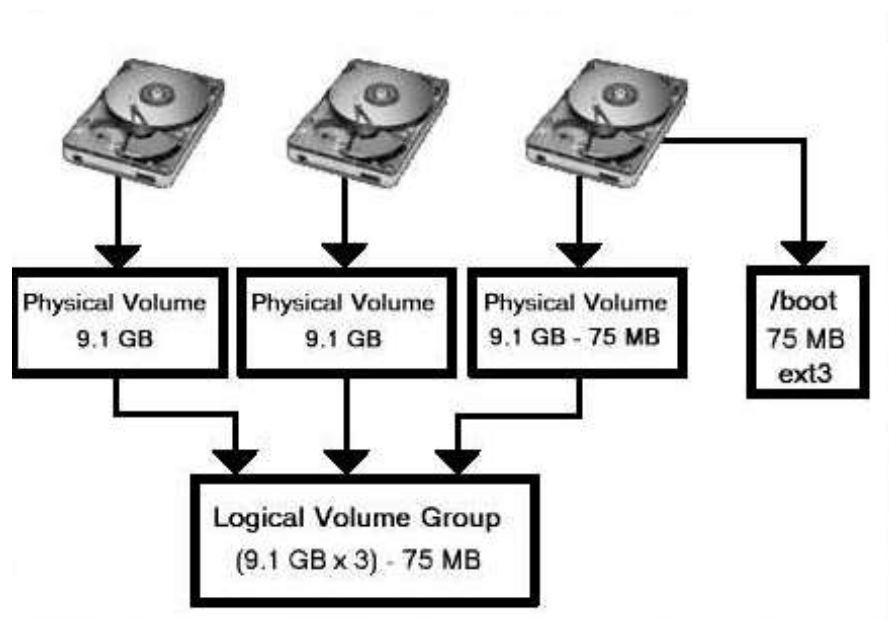
- **Línea RAID** => va rellendo los discos 1 a 1, no tiene ni paridad, ni distribución de datos entre los discos.

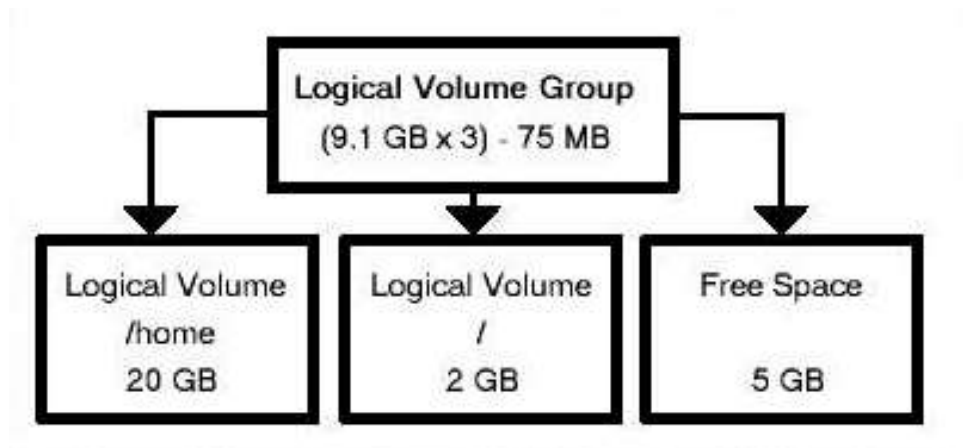
- **LVM** => grupo de volúmenes lógicos formado por varios volúmenes físicos que pueden ser redimensionados.

- La partición /boot no puede estar en un grupo de volumen lógico, porque el gestor de arranque no puede leerlo

Pasos:

- 1) Crear un volumen físico desde las unidades de disco duro.
- 2) Crear un grupo de volúmenes desde los volúmenes físicos.
- 3) Crear volúmenes lógicos desde el grupo de volúmenes y asignar los puntos de montaje de volúmenes lógicos.





CUOTAS DE DISCO

- Restringir el almacenamiento en disco de usuarios y grupos. Permitirá que cada usuario tenga una pequeña porción de disco de uso individual y tener más espacio para proyectos en grupo.
- Se puede configurar por número de bloques y por inodos (nº de archivos en disco).

Pasos:

a) Activar cuotas en el sistema de archivos a través de /etc/fstab.

Ej: gedit /etc/fstab

```
LABEL= / / ext3 defaults,usrquota,grpquota
```

b) Volver a montar el sistema de archivos.

- Si es raíz => reiniciamos
- Si es otro => umount /mnt/datos
mount /dev/hda6 /mnt/datos

c) Crear los archivos cuota y generar la tabla de uso de espacio en disco.

Paso 1: quotacheck -acugm => crea los archivos aquota.user y aquota.group

- a => todos los sistemas de archivos de /etc/mtab
- c => crear aquellos que tienen las cuotas activadas
- vg => usuario y grupos
- m => forzar en caso de estar utilizado

Paso 2: quotacheck -avugm => se debe hacer regularmente

- v => verificar

Genera la tabla del uso actual del disco duro por el sistema de archivos con cuotas activadas.

Paso 3: reiniciar sistema ó quota on – quota off

d) Asignar cuotas

edquota username => editar cuota de usuario con vi

- quota username => comprobar cuota de usuario
- edquota -g nombregrupo => cuotas para grupo. El grupo principal del usuario será el que limitaremos
- edquota -t => configurar periodo de gracia para sistemas de archivos.
 - T usuario => usuario particular
- repquota => informe de cuotas
 - a => de todos los sistemas de archivos. Si aparece on ++ se ha excedido el límite
- quotaoff -vaug => desactivar cuotas
- quotaon -vuga => activa cuotas

TEMA: Empaquetado y compresión. Paquetes

tar => empaquetar
gzip => comprimir

- tar => permite empaquetar o desempaquetar ficheros.
Tar “opciones” archivo.tar [archivo origen a empaquetar]

Opciones:

- c => crear archivos empaquetados
- x => expandir archivos empaquetados. Desempaquetar
- v => almacenamos o visualizamos. Información de forma detallada
- f => indicar que es un fichero el archivo.tar
- z => filtrar el archivo a través de gzip(comprimir/descomprimir)
- M => para crear o desempaquetar en varios discos
- t => mostrar un listado del fichero.tar

Ejemplos:

```
tar -cvf paquete.tar texto.txt texto1.txt texto2.txt => empaqueta
tar -xvf paquete.tar => desempaquetar
tar -tvf paquete.tar => visualiza listado
tar -czvf paquete.tar.gz texto.txt texto1.txt => empaqueta y comprime
tar -xzvf paquete.tar.gz => descomprime y desempaqueta
```

- gzip => comprime y descomprime ficheros.

Gzip [opciones] archivo

- l => listado
- d => descomprimir archivos
- t => comprobar que la compresión se ha realizado con éxito
- 1,9 => nivel de compresión
 - 1 => menos ratio, más rapidez
 - 9 => más ratio, menos rapidez
- Defecto => 6

Ejemplos:

```
gzip texto.txt => crea texto.txt.gz
gzip -9 paquete.tar => crea paquete.tar.gz con máxima compresión
gzip -d texto.txt.gz => descomprime
gunzip => descomprime
Ej: gunzip texto.txt.gz
    gzip -d texto.txt.gz      Son equivalentes
```

tar -cvfM /dev/fd0 texto*.txt => copia segura en varios diskettes
tar -xvfM /dev/fd0 => desempaqueta

- compress, uncompress => comprime ficheros y genera ficheros .z . No se utiliza mucho, se puede utilizar gunzip para descomprimirlo.

Ej: compress texto.txt => crea texto.txt.Z
uncompress texto.txt.Z => descomprime

- bzip2, bunzip2

Ej: bzip2 fichero => genera fichero.bz2
bunzip2 fichero.bz2 => descomprime

- File Roller => herramienta gráfica

RPM (Red Hat Package Manager => sistema de paquetes de Red Hat)

- **Paquete** => archivo que contiene todos los archivos de un componente instalable, y además almacena información de control y scripts que se ejecuta al instalar o borrar el paquete.
- Instalará programas y sus librerías.
- **Características:**
 - Mantiene una base de datos de información de todos los paquetes instalados y ficheros que contiene
 - Control de dependencias: necesita otros paquetes para funcionar correctamente. Nos avisará al instalar si necesita otros paquetes y al desinstalar si es utilizado por otros paquetes.
 - Control de incompatibilidades => si al instalarlo deja de funcionar otro.
 - No es necesario reiniciar el equipo
- **Nomenclatura** => nombre, versión, release, plataforma y extensión.
Ej: Xfree-VGA16-3.3.5-3.i386.rpm

Comando rpm

- **Instalación de paquetes** => debemos ser root
rpm -i [opciones] paquete.rpm

Ej: rpm -ivh paquete.rpm => se instala con una monitorización más agradable

- Si existe un fallo de dependencias
 - Buscar la librería en mi disco duro.
 - Si existe pero no es la versión, debemos buscar en internet
rpm -Uvh paqueteversión2.rpm

- **Actualización de paquetes**

rpm -U [opciones] paquete.rpm

Ej: rpm -Uvh paquete.rpm

- Si no es compatible con versiones anteriores renombrará el antiguo como .rpmsave
- Es mejor actualizar que desinstalar y luego instalar

- Desinstalar paquetes

rpm -e [opciones] paquete.rpm

- No es necesario indicar versión, ni extensión
rpm -e --test paquete.rpm => saber que pasa al desinstalar
rpm -e --nodeps paquete.rpm => desinstalarlo pese a problemas de dependencias

rpm -qa |grep paquete => busca un paquete concreto

rpm -qa => lista de paquetes

rpm -qi paquete => información del paquete

rpm -qc paquete => muestra ficheros de configuración

rpm -qd paquete => muestra ficheros de documentación

rpm -qR paquete => muestra todas las dependencias del paquete

rpm -ql paquete => muestra donde está instalado

- Verificación de paquetes

rpm -V [opciones] paquete.rpm

rpm -Va => verifica todo el sistema de paquetes

- Instalación de ficheros fuente

- Es necesario tener el compilador C++
- Descomprimos el paquete
tar -xzh paquete_src.tgz
- Situar en el directorio donde se ha extraído el programa
- Configurar la precompilación => ./configure
- make [all] => makefile(Compilación)
- Instalación => make install
- Ventaja => se puede instalar en cualquier plataforma

yum => actualización automática de programas a través de internet

/var/cache/yum => guarda cabeceras y paquetes sin instalar

/etc/yum.conf => configuración de los servidores

Sintaxis:

yum [opciones] [comando] [paquete]

yum install paquete => instalar

yum update [paquete] => actualiza paquetes asegurando dependencia

yum check-update => revisa paquetes a actualizar

yum remove [paquete] => borra paquetes

yum lista => lista los paquetes que pueden ser instalados

--download-only => solo descarga

system-config-packages => agregar y quitar paquetes el cd de instalación

up2date => herramienta gráfica para actualizar paquetes

alien => convierte distintos sistemas de paquetes

alien -r paquete.deb => obtenemos paquete.rpm

alien -d paquete.rpm => obtenemos paquete.deb

Comandos de DEBIAN

dpkg => comando para instalar paquetes en Debian

dpkg -i paquete.deb => instalarlo

dpkg -s paquete.deb => información

dpkg -r paquete.deb => desinstala

gsynaptic => instalador de paquetes en Debian. Actualiza a través de Internet

apt => actualiza paquetes debian

servidores => /etc/apt/sources.list

Descargar => apt-get install paquete.deb

apt-get remove paquete.deb

TEMA: La Shell Bash

- Shell (caparazón)

- Procesador de órdenes que sirve para ejecutar comandos
- También sirve como lenguaje de programación
- También para formar comandos más complejos combinando comandos simples (alias)

- Bash => Bourne Again Shell (La shell de Bourne viene de nuevo)

- La que tiene Linux por defecto
- Ejecuta órdenes leídas desde entrada estándar o desde fichero
- Devuelve 0 si la ejecución de una orden se realiza sin errores y un valor distinto de cero si existen errores.

- Otras Shell => /bin/csh

/bin/ksh

/bin/zsh

/bin/bash2

/bin/bsh

/bin/false

- Variables de entorno para la bash

- Definición de una variable:
nombrevariable=valor
apellido=espejo
- Exportar una variable para que sea visible en esta shell y en todos los procesos hijos
export "nombrevariable"
- Ver el valor de una variables
echo \$nombrevariable
- Se puede poner como valor de una variable un comando
micomando=ls
\$micomando

- set => visualiza el nombre y valor de las variables de entorno
 - HOME => directorio de comienzo de usuario
 - PATH => lista de directorios separados por “;” en los que la shell busca para encontrar los comandos. Escoge el primero que encuentre.
 - PS1 => el prompt o indicador de inicio que presenta bash al usuario
 - PWD => directorio de trabajo actual
 - HISTSIZE, HISTFILE => tamaño del histórico de comandos

- Ficheros de inicio de la bash

- /etc/profile => inicializa variables: USER, PATH, LOGNAME, etc
Crea máscara => umask
Ejecuta /etc/profile.d/*.sh
- Después ejecuta
~/.bash_profile
~/.bash_login
~/.profile
- Al terminar se ejecuta ~/.bash_logout
- Si se una shell interactiva y no es el comienzo se ejecuta ~/.bashrc

- Personalizando el prompt

- Variable PS1
- Valor predeterminado => PS1=”[\u@\h\w]\\$”
- echo \$PS1 Ej: PS1=inca
 - \d => fecha
 - \h => nombre del ordenador
 - \H => nombre del ordenador con dominio
 - \s => nombre de la shell
 - \t => hora
 - \u => nombre de usuario
 - \w => directorio de trabajo
 - \\$ => # superusuario \$ usuario normal
 - \T => hora en formato 12h
 - \W => nombre del directorio actual
 - \@ => hora en 12

- Alias

- Nombrar con una sencilla palabra un comando complejo
- alias => ver o asignar alias
- unalias => quitar alias
Ej: alias ver=”ls -la”

- Historia de órdenes

- Ordenes tecleadas con anterioridad
- set -o hystory => da acceso a la shell a la historia de órdenes
- HISTSIZE => número de órdenes que se guardad (Defecto 1000)
- HISTFILE => fichero utilizado (Defecto => ~/.bash_history)
- HISTFILESIZE => número de líneas que guarda. Se trunca el fichero para contener no más de HISTFILESIZE

- Los Builtins (órdenes internas)

- Comandos que ya vienen implementados en la bash
- No hay que buscar comando externo
- A diferencia los externos se buscan en el directorio path

- Redirección

- Antes de ejecutar una orden la E/S puede ser redirigida
- Descriptor fichero 0 => entrada estandar (teclado)
- Descriptor fichero 1 => salida estandar (pantalla)
- Descriptor fichero 2 => salida error estandar (salida estandar)
- Redirección de entrada "<" => [n] < entrada
- Redirección de salida ">" => [n] > salida
- Añadir a la salida redirigida ">>" => [n] >> salida

Ej:

```
-ls > lista.txt
-ps -ax > p.txt
uname >> p.txt
-ls > lista.txt 2>&1 => salida estandar la redirige a lista.txt y la de error se redirige a
la salida que es el fichero lista.txt
-ls 2>&1 > lista.txt
```

- Tuberías

- Secuencia de una o más órdenes separadas por el carácter "|"
- Orden1 | orden2.....
- La salida estándar de orden1 se conecta a la entrada estándar de orden2
- Cada orden de la tubería se ejecuta como proceso separado

Ej:

```
rpm -qa | grep httpd
cat lista.txt | wc -l => cuenta las líneas de lista.txt
cat /etc/passwd | grep alumno => busca alumno en /etc/passwd
ls -la | more => salida página a página
```

- Listas de comandos

- Secuencia de comandos simples o tuberías separadas por operadores: ";", "&", "&&", "||" y terminada en ";", "&" o retorno de carro
- comando1;comando2 => ejecuta secuencialmente los 2 comandos
Ej: ls; ps
- comando1 && comando2 => comando2 se ejecuta si comando1 se ejecuta satisfactoriamente (es decir devuelve código 0)
Ej: ls && ps
- comando1 || comando2 => comando2 se ejecutará si y solo comando1 falla (devuelve distinto de 0)

- Expresiones

- test expr => evalúa expresiones (si es verdadera devuelve 0 y si es falsa devuelve distinto de 0)
- También se puede usar corchetes [expr]
- Las expresiones:
 - opción fichero (ver fotocopias)
 - Ej: -e fichero => si existe ese fichero
 - d fichero => si es un directorio
- Binarias:
 - fichero1 -nt fichero2 => verdad si fichero1 es más reciente
 - fichero1 -ef fichero2 => verdad si ambos tienen el mismo número de inodo y dispositivo
 - fichero1 -Ot fichero2 => verdad si fichero1 es más antiguo
- Cadenas
 - z cadena => longitud 0
 - n cadena => longitud no es 0
 - cadena = cadena2 => compara cadenas
 - !=
 - <
 - >
- arg1 OP arg2 => -eq => igual
- ne => distinto
- lt => menor
- le => menor o igual
- gt => mayor
- ge => mayor o igual

- Evaluación aritmética

- Precedencia de operador, paréntesis (fotocopias)

- Combinación de expresiones

- (expresión) => cambiar precedencia
- ! expresión = negar
- expresión1 && expresión2 => si ambos es verdadero
- expresión1 || expresión2 => si alguno es verdadero

BUCLES

- **Until**
until condición; do listacomandos; done => ejecutará hasta que la condición sea 0
- **While**
while condición; do listacomandos; done => ejecutará mientras la condición devuelva 0

- **For**

```
for nombre in palabras....;
```

```
do
```

```
listacomandos
```

```
done
```

- Sustituye el nombre por palabras en cada ciclo
- Si palabras no existe se asume la lista de parámetros (\$@)

- Ej: for i in manuel ana luis;

```
do
```

```
echo $i
```

```
done
```

SENTENCIAS CONDICIONALES

- **If**

```
if condición ; then
```

```
comandos
```

```
[elseif condición ; then
```

```
comandos]
```

```
[else comandos alternativos;]
```

```
fi
```

Ej:

```
if [ -e /etc/hosts ];
```

```
then
```

```
cat /etc/hosts
```

```
else
```

```
echo El fichero hosts no existe
```

```
fi
```

- **Case**

```
case palabra in
```

```
patron1 | patron2....) comandos;;
```

```
patron3 | patron4....) comandos;;
```

```
*) comandos
```

```
esac
```

Ej:

```
echo Introduce animal:
```

```
read animal
```

```
case $animal in
```

```
caballo|perro|gato) echo -n "cuatro";;
```

```
hombre|mono|pato) echo -n "dos";;
```

```
*) echo -n "un numero desconocido de";;
```

```
esac
```

```
echo "patas"
```

- **break, continue, exit**

- break => termina el bucle
- continue => continua con la siguiente interacción no terminado el bucle actual
- exit => detiene la ejecución actual del script

Ej:

```
a=1
while true
do
    echo $a
    a=`expr $a + 1`
    if [ $a -gt 10 ];
    then
    fi
echo se termino
```

select => visualiza un menú de opción

```
select i [in lista]
do
    ordenes
done
```

Ej:

```
PS3="Opcion:"
select i in Listado Quien Salir
do case $i in
    Listado) ls -l ;;
    Quien) who ;;
    Salir) exit 0 ;;
    *) echo Opcion incorrecta
esac
done
```

Ficheros de órdenes (shell scripts)

- Ficheros de texto que contienen comandos de la shell
- Al invocarse desde la shell se lee y se ejecuta
- Debemos dar permiso de ejecución => `chmod a+x "script"`
- Si la primera línea empieza por `#` se especifica el intérprete de comandos que aparece a continuación
Ej: `# /usr/bin/perl`
- Si no aparece se utiliza el intérprete por defecto
- **Argumentos** => variable que se transmite cuando se invoca un script. También se llaman argumentos posicionales:
\$0 => nombre del propio scripts
\$1.....\$9 => parámetros

Ej:

```
echo $0
echo $1 $2
```

./param	primero	segundo
\$0	\$1	\$2

- **read variable** => toma una línea de entrada de usuario y la asigna a una variable.
- **Variables especiales**
 - \$# => número de argumentos
 - \$* => cadena completa de argumentos
 - \$? => código de retorno de la última orden ejecutada. Si es 0 error, distinto de 0 no hay error
 - \$@ => cadena de argumentos como lista de cadenas

Ej:

```
for i in $@ ;
do
    echo $i
done
```

- **shift n°** => desplaza n posiciones los argumentos

Ej:

```
echo $1
shift 1
echo $1
shift 1
echo $1
```

- **#** => comentarios
- **expr** => ejecutar operaciones aritméticas

Ej:

```
i = `expr $i + $h`
i=8
h=9
i = `expr $i $h`
echo $i
```

- **Funciones** => agrupación de un conjunto de órdenes usadas frecuentemente

Ej:

```
error()
{ echo Erro de sintaxis
  exit 2
}
if [ $# = 0 ];
then
    error
else
    echo Hay $# argumentos
fi
```

- **Señales y orden trap**

- Ciertos eventos generan señales que se envían a los procesos en ejecución

Ej:

logout => envía señal 1 a los procesos batch
delete => envía señal 2 a los procesos activos
kill PID => envía señal 15 al proceso PID

- La orden trap captura señales antes de matar al proceso
 - trap -l => lista de señales
 - Sintaxis => trap argumento señal

Ej:

```
trap 'echo adios; exit' 2
while true
do
    echo hola
done
```

EJERCICIOS

– Crea:

```
if [ ! -d $1 ];  
then  
    mkdir $1  
    chmod 700 $1  
fi
```

Si no existe la carpeta la crea y le da esos permisos

– Comprobar

```
if [ $# = 0 ];  
then  
    echo Debes introducir argumentos  
    exit 1  
fi  
for i in $@  
do  
    if [ -f "$1" ];  
    then  
        echo -n "$1 es regular"  
    if [ -x $1 ];  
    then  
        echo "ejecutable"  
    else  
        echo "no es ejecutable"  
    fi  
    elif [ -d "$1" ];  
    then  
        echo "$1 es un directorio"  
    else  
        echo "$1 es una cosa rara o no existe"  
    fi  
shift  
done
```

– Fecha

```
semana=`date | cut -c 0-3`  
case $semana in  
lun) echo Hoy es lunes ;;  
mar) echo hoy es martes ;;  
mie) echo hoy es miercoles ;;  
jue) echo hoy es jueves ;;  
vie) echo hoy es viernes ;;  
sab) echo hoy es sabado ;;  
dom) echo hoy es domingo ;;  
esac
```


- **Contador**

```
a=42
while [ $a -le 53 ]
do
    echo contador = $a
    a=`expr $a +1`
done
```

Administración del sistema de impresión

- Desde la versión 9 de Red Hat Linux se utiliza CUPS como sistema de impresión predeterminado, antes se utilizaba LPRng.
- Menú principal / Configuración del sistema / Impresión
- Desde la línea de comandos => system-config-printer => gráfica y texto
system-config-printer -tui => texto
- Ficheros implicados => /etc/printcap
/etc/cups/* => cupsd.conf => configuración de cups
printers.conf => configuración impresoras
- Tipos de cola de impresión
 - Conectada localmente => Puerto paralelo /dev/lp0
Puerto USB /dev/usb/lp0
 - Conectada CUPS (IPP) => servidor que ejecuta CUPS por tco/ip
 - Conectada UNIX (LPD) => servidor ejecuta LPD
 - Conectada WINDOWS (SMB) => Samba
 - Conectada NOVELL (NCP) => Novell
 - Conectada de JetDirect
- Conectar una impresora de red IPP (Servidor ejecutando IPP)
 - Compartir la impresora en el servidor (Menú contextual)
 - habilitar en el cortafuegos cliente-servidor el puerto UDP 631
 - Agregar la impresora en la máquina cliente
 - Tipo conectada CUPS (IPP) Ej: cola /printers/hp
- system -config -printer -tui --Xexport > fichero
Guardar fichero de configuración de las impresoras
- system -config -printer -tui --Ximport < fichero
Restaurar configuración de impresoras desde el fichero
- system -config -printer -tui --Ximport --merge < fichero
Restaurar agregando pero no sobrescribe
- Service cups => stop
start
statut
restart
- Administración web <http://ip:631>

Comandos relacionados con la impresión

- lp,lpr => enviar un trabajo a la impresora
- lpq => ver cola de impresión
- lprm nº trabajo => elimina un trabajo de la cola de impresión
- lpc => comprobar el estado de la impresora
status, enable[imp/all], disable[imp/all], start[imp/all], stop[imp/all], restart
- lpstat => estado de la cola de impresión. Información impresora cups
- cancel nº => cancela un trabajo
- enable imp => activa impresora => /usr/bin/enable hp
- disable imp => desactiva impresora => /usr/bin/disable hp
- lpadmin -p imp [opciones] => administrar impresoras
- accept imp => acepta peticiones para la cola /usr/sbin/accept hp
- reject imp => denega peticiones para la cola /usr/sbin/reject hp

- `lpmove job dest2` => mueve trabajos a otra cola

Clases => grupos de impresoras. Envía el trabajo a la primera impresora disponible

Cuotas de impresión

- Se almacenan en `/etc/cups/printers.conf`
- `usr/sbin/lpadmin -p hp -o job-quota-period = 604800 (segundos)`
 - o `job-k-limit = 30 (kbytes)`
 - o `job-page-limit = 100`

Restricciones de acceso

- `/usr/sbin/lpadmin -p -u allow:all`
 - u `allow:pedro,juan`
 - u `deny:luis,maria`
 - u `allow:@grupo1`

SAMBA

- Usa el protocolo SMB para compartir archivos e impresoras en la red, compatible con Windows, OS2 y Linux entre otros.
 - Permite compartir y acceder a recursos con Servidores Windows
- **NFS** => compartición de carpetas solo en Linux
 - Fichero de configuración => **`/etc/samba/smb.conf`**
 - Formado por dos programas que se ejecutan en demonios => **`smbd` y `nmbd`**
 - Proporcionan cuatro servicios clave:
 - Servicios sobre archivos e impresoras
 - Autenticación y autorización

Estos dos son en `smbd`

 - Resolución de nombres
 - Anuncio de servicios en la red (browsing)

Estos dos en `nmbd`
 - **Arrancar el servicios**
`service smb { start/stop/restart/reload/status/condrestart }`
 - **El fichero `smb.conf`**
 - Está dividido en secciones que son nombres entre corchetes
 - Cada sección designa un recurso compartido. Existen tres secciones especiales: `[global]`, `[homes]`, `[printers]`
 - `testparm` => comando para ver si el fichero de configuración es correcto y vuelca los recursos compartidos
 - Sección global: se aplica al servidor SAMBA en general.
 - Ej: `netbios name = PC6` (nombre netbios del servidor)
 - `server string = servidor Samba (descripción)`
 - `workgroup = nombre` (nombre del grupo)
 - Sección homes: genera recursos compartidos en tiempo de ejecución. Cuando un usuario solicita un recurso previa autenticación de nombre/contraseña y dicho recurso no existe se conecta al recurso home, siempre que la autenticación sea correcta. El nombre del nuevo recurso es el nombre de usuario. Si no se proporciona el atributo `path`, se tomará el directorio `$HOME`.

– **Atributos usuales**

- path = /home/usuarioSamba/%u
- guest ok = yes => no es necesaria autenticación
- comment = comentario
- read only = yes => solo lectura
- writeable = yes => escritura

– **Crear un recurso compartido**

[recurso]

comment = Archivos públicos
path = /home/anónimo
read only = no
public = yes

[publico]

path = /datos
public = yes
writeable = yes
only guest = yes

– **Cliente Samba => smbclient**

- smbclient -L servidor => ver recursos
- smbclient //servidor/recurso
mismos comando que FTP
- smbclient -U usuario //servidor/recurso => autenticado

– **Autenticación de usuarios en Samba**

/etc/samba/smbusers => usuarios Samba (alias)

/etc/samba/smbpasswd => usuarios/contraseñas

- Las contraseñas van encriptadas (W' 95 service pack 3)
- Usuarios deben existir en el sistema
- cat /etc/passwd | mksmbpasswd.sh > /etc/samba/smbpasswd
- smbpasswd usuario => cambia contraseñas
- smbpasswd -a usuario (debe existir en Linux)
- mount -t smbfs -O username = angel, passwd = angel
//pc1/recurso /mnt/redes

– **Opciones de red**

- host allow => permitir
- host deny => denegar
Ej: host allow 172.26.0. => 172.26.0.0/24
- subredes => 192.18.1.
- nombres de dominio => servidor1.inca.edu
- nombres de subdominio => .inca.edu
- ALL => todos
- ALL EXCEPT => todos menos

