

Un **Administrador de Sistemas** es la persona que tiene la responsabilidad de implementar, configurar, mantener, monitorizar, documentar y asegurar el correcto funcionamiento de un sistema informático. Tiene por objeto garantizar el tiempo de actividad (uptime), rendimiento, uso de recursos y seguridad de servidores.

Sus cualidades son: Autoridad + Responsabilidad + Servicio + Cooperación.

Tareas detalladas del administrador:

·Hardware:

- Planificar y administrar el entorno físico: Sistema de refrigeración, conexiones de energía, control del entorno.

- Planificar los cortes de suministro para realizar actualizaciones o administrar los dispositivos.

- Localizar, reparar y reemplazar componentes defectuosos.

- Configurar y mantener la conectividad entre los hosts (redes): Monitorización, resolución de problemas, etc.

- Instalar y mantener dispositivos del sistema, hardware y drivers

·Software:

- Mantenimiento software: Instalación y configuración de sistemas operativos, detección de problemas en el software y reparación, configurar y mantener aplicaciones de negocio, etc.

- Documentación: Documentar todo el sistema.

·Soporte a usuarios:

- Formar a los usuarios en el manejo del software y en seguridad, proporcionar software.

- Dar acceso a los usuarios a la documentación.

·Servicios:

- Instalar y mantener las cuentas de usuario.

- Determinar los requisitos SW.

- Instalar, configurar y administrar servidores web.

·Seguridad:

- Determinar las cuotas de disco y las políticas de manejo del espacio.

- Configurar y manejar la seguridad del sistema.

·Copias de Seguridad:

-Determinar la estrategia de copias de seguridad y automatizarlas. Realizar restauraciones y determinar planes de supervivencia a catástrofes.

Estrategia del Administrador de Sistemas al realizar una tarea:

1. Planear la tarea antes de realizar cambios.
2. Hacer los cambios reversibles con una copia de seguridad.
3. Realizar los cambios de forma incremental (si se prueban, mejor).
4. Antes de hacerlo público, probarlo.
5. Conocer como trabajan las cosas realmente.

Linux/GNU: es un SO libre creado y mantenido por voluntarios, por lo tanto es de código abierto y gratuito.

El **Software Libre** es aquél que concede cuatro libertades a sus usuarios:

- Libertad 0: Libertad de usar el programa con cualquier propósito.
- Libertad 1: La libertad de estudiar cómo funciona el programa y adaptarlo a tus necesidades.
- Libertad 2: La libertad de distribuir copias del SW.
- Libertad 3: La libertad de mejorar el programa y hacer públicas las mejoras a los demás.

Para que un programa se designe como SL debe cumplir estas libertades, si no se le designará como **Software Privativo**.

Software de dominio público: Se denomina software de dominio público a aquél el que no tiene copyright (derechos de autor), porque el autor ha renunciado a ellos.

Software semilibre: El software semilibre es aquel que proporciona las mismas libertades que el software libre, siempre y cuando sea usado sin ánimo de lucro.

Freeware: Programas gratuitos

Shareware: programas que el autor permite utilizar, pero condicionando su uso. Por ejemplo, puede exigir que se pague por él si se emplea más allá de un determinado periodo de evaluación.

Software con fuentes: Con algunos programas se entrega el código fuente, para que el usuario pueda comprobar que no hace nada malicioso, pero eso no los hace libres, si su licencia no permite su modificación.

La **independencia tecnológica** quiere decir que no nos atamos a ningún proveedor en particular.

Desventajas del SL

- Necesidad de una formación sobre el uso de SO libres.
- Ausencia de interfaces visuales (Esta apariencia suele ser privativa).
- No siempre es compatible con todo tipo de HW.

SuperUsuario o administrador

Es el usuario que tiene siempre todos los privilegios sobre cualquier fichero, instrucción u orden del sistema

Comandos:

whoami: Devuelve el nombre de usuario de la sesión

su: Super Usuario, nos da permisos de superusuario o admin.

sudo <orden>: Nos pide la contraseña para activar el modo super usuario y poder ejecutar ordenes.

visudo: Orden para modificar el fichero de configuración /etc/sudoers

write: Envía un mensaje a un usuario

talk: Conversa con un usuario aunque esté en otra maquina.

wall: Manda un mensaje a todos los usuarios del sistema.

Ficheros especiales:

/etc/sudoers: Es un fichero de configuración de solo lectura incluso para root en el que se establece quien puede ejecutar qué y cómo desde sudo.

/etc/motd: Contiene el mensaje del día que se imprime justo después de entrar al sistema en modo texto. Esto evita el mensaje del día --> \$HOME/.hushlogin.

/etc/issue: Contiene el mensaje que se muestra antes del login, normalmente muestra la versión de Linux en modo texto.

En GNU/Linux todo lo que no es un fichero es un proceso.

Sistema de ficheros:

Guarda los ficheros del sistema y los organiza de manera jerárquica en directorios.

A nivel lógico parece un árbol pero realmente están desorganizados por el disco duro.

Los **nodos-i** son **metadatos** sobre los ficheros que proporcionan información sobre aspectos como su tamaño, **permisos**, etc.

Cada fichero tiene un nodo-i.

El acceso a los ficheros se gestiona con los propietarios.

Usando **chown** <propietario> <fichero> podemos cambiar el usuario propietario (Se necesitan privilegios root).

Permisos:

Cada fichero tiene una cadena de permisos compuesta por 9 espacios separados de 3 en 3. El primer grupo son los privilegios de usuario sobre el fichero, el segundo privilegios de grupo y el tercer grupo el resto de usuarios.

r: Lectura

w: Escritura

x: Ejecución

Esta cadena también se puede representar como números de 0 a 7 en 3 espacios. Por ejemplo el 741 sería: rwx r-- --x.

Para agregar permisos se usará el comando **chmod**.

Ejemplos:

chmod a+x: Añade permisos de ejecución a los 3 grupos

chmod go-x: Elimina permisos de ejecución al grupo de grupo y otros.

chmod u-x, go-r: Elimina permisos de ejecución al usuario propietario y elimina la lectura al grupo de grupo y otros.

Chmod 700: Le da permisos de rwx al usuario.

Permisos especiales:

-sticky bit(t): chmod o+t fichero.

Is nos permite ver el bit situándose este a la derecha de todos los bits de los permisos del fichero.

Este permiso sirve para directorios y permite que solo root o el propietario de un fichero puedan borrar o renombrar el fichero aunque tengan permisos de escritura en la carpeta.

-suid (s): chmod u+s fichero. PARA USUARIOS

El comando ls lo representa como una s en el tercer bit. Si la s es minúscula es que tiene permiso, si es mayúscula no.

En ejecutables esto sirve para el cambio de dominio a nivel de usuario. Durante la ejecución el usuario efectivo es el propietario fichero y no el que lo ejecuta.

-sgid (s): chmod g+s fichero. PARA GRUPOS

El comando ls lo representa como una s en el sexto bit. Si la s es minúscula es que tiene permiso, si es mayúscula no.

En ejecutables esto sirve para el cambio de dominio a nivel de grupo. Durante la ejecución el grupo efectivo es el propietario fichero y no el del usuario que lo ejecutó.

En directorios al crear un fichero en su interior, el grupo propietario del nuevo fichero es el grupo del directorio y no del usuario que ejecuta la orden.

Máscara de permisos (umask):

Cada vez que se crea un fichero se le asignan los permisos bases asignados por umask. De base los directorios tendrán 777 permisos y los ficheros 666, esto quiere decir que cuando se crea un fichero, nadie podrá ejecutarlo hasta que no se le den permisos de ejecución con chmod.

¿Se pueden crear ficheros con permisos de ejecución usando umask?

Tipos de ficheros (ls -l):

- Normal.
- Directorio (d): Son ficheros que contienen enlaces a otros ficheros.
- Especial de bloque (b): Fichero especial para interactuar con un dispositivo basado en bloques.
- Especial de carácter (c): Fichero especial para interactuar con un dispositivo basado en caracteres.
- Named Pipes (p): Tubería FIFO con nombre (Comunicación entre procesos de diferentes usuarios).
- Socket (s): Como los pipes pero con comunicación dúplex (Dos sentidos).
- Enlace físico.
- Enlace simbólico (l).

Enlaces:

-Son archivos especiales que permiten que varios nombres (**enlaces**) se asocien a un único e idéntico archivo.

-Son varias instancias de un mismo archivo en diversos lugares de la estructura jerárquica sin necesidad de copiarlos.

•**Físicos:**

-Representa un nombre alternativo para un archivo (dos nombres de fichero apuntando a un nodo-i).

-Al eliminar uno de los enlaces al archivo no se elimina **mientras quede aún un enlace físico**, el archivo no se elimina.

-Solo es posible la unión entre ficheros que ese encuentren en la misma partición y no es válido a directorios.

•**Simbólicos:**

-Es un puntero virtual al archivo real.

-Si se elimina el enlace, no se elimina el fichero

Procesos:

•**Atributos:**

-*PID*: Identificador del proceso

-*PPID*: Identificador del proceso padre.

-*Nice number*: Prioridad asignada al ejecutarlo.

-*TTY*: Terminal en la que se está ejecutando.

-*RUID*: Identificador del usuario real (el que lo ejecutó).

-*EUID*: Identificador del usuario efectivo, si hay cambio de dominio se refleja en **suid**.

-*RGID*: Identificador del grupo real (el que lo ejecutó)

-*EGID*: Identificador del grupo efectivo, si hay cambios de dominio se refleja en **sgid**.

•**Tipos:**

-Interactivos: Hay alguien conectado al sistema que los inicia.

-Encolados: Procesos que se mandan a un buffer para ser ejecutados.

-Demonios: Programas ejecutados en segundo plano durante el arranque, que esperan de forma continua un determinado evento.

Dispositivos

- Ficheros especiales de caracteres: Representan a dispositivos de caracteres.
- Ficheros especiales de bloques: Representan a dispositivos de bloques.

·**Estos ficheros** se almacenan en el directorio /dev:

- /dev/fd0: Disquete de la primera disquetera.
- /dev/sda: Primer disco duro.
- /dev/sda1: Primer partición del primer disco.
- /dev/sdb: Segundo disco duro.
- /dev/sdc: Disco USB.
- /dev/tty1: Primera terminal de consola.
- /dev/lp0: Primer puerto paralelo.

Estructura genérica del sistema de ficheros

Jerarquía Estándar del Sistema de Ficheros (**FHS**).

·**Tipo de contenido de un directorio:**

- Estáticos: Pueden estar en dispositivos de solo lectura (Read-only).
- Dinámicos: Deben de encontrarse en dispositivos de lectura-escritura (read-write).
- Compatibles
- No compatibles

/bin: Ficheros ejecutables básicos compartidos (mv, cp).

/dev: Ficheros especiales de dispositivos.

/etc: Ficheros de configuración local del sistema.

/sbin: Ficheros ejecutables que solo el administrador puede ejecutar normalmente.

/home: El directorio de trabajo del usuario

/lost+found: Contiene “referencias” a los ficheros marcados como erróneos.

/lib: Contiene las librerías necesarias para ejecutar los archivos.

/proc y /sys: Contiene sistemas de ficheros virtuales, contienen información sobre procesos, núcleos...

/tmp: ficheros temporales. Tiene el permiso *t* activo.

/var: ficheros variables: colas de datos (spool) de impresión, e-mail..., ficheros del cron, atd, ficheros de log...

/boot: núcleo y ficheros necesarios para cargar el núcleo y ficheros de configuración del gestor de arranque.

/mnt, /mount ó /media ⇒ montaje de otros sistemas de ficheros: disquetes, cdroms...

/mnt/floppy ó /media/floppy

/mnt/cdrom ó /media/cdrom

/opt: paquetes de aplicaciones estáticas (no actualizables).

Comandos:

chown: Cambia el usuario propietario de un fichero (se necesitan privilegios).

chgrp: Cambia el grupo propietario de un fichero (esto puede realizarlo el propietario del fichero, el que pertenezca al grupo o root).

chmod: Permite el cambio de permisos de un archivo.

umask: Nos permite modificar o consultar con que permisos se van a crear los ficheros y directorios

ls -l: Nos permite visualizar los diferentes ficheros y directorios con sus detalles.

ls -li: para visualizar los enlaces.

ln archivo-real enlace-fisico: Para crear enlaces físicos.

ln -s archivo-real enlace.simb: Para crear enlaces simbólicos

man hier: Manual del FHS.

Arranque y parada del Sistema

- Arranque: El sistema se prepara para ser usado por los usuarios
- Parada: El sistema se deja consistente (vaciar la caché por ejemplo).

Los procesos sencillos se basan en ficheros de configuración y guiones Shell que determinan y controlan procesos.

Proceso de arranque

Este consta de dos fases: HW y SO.

1ª Iniciador ROM

Al encender el ordenador se emite una señal eléctrica que resetea todos los registros a default.

Se carga la dirección de inicio del **iniciador ROM**.

Contiene el SW de configuración del HW del sistema (**BIOS**).

Lee y almacena en memoria el programa que carga el SO y le pasa el control al cargador del SO.

2ª Programa cargador (**Master Boot Program**)

Se sitúa en los primeros sectores del disco (**Master Boot Record**) y tiene un tamaño prefijado.

Carga el Kernel del SO.

3ª Núcleo del SO

Realiza una comprobación del HW del sistema.

Inicializa sus tablas internas, crea las ED que necesite, etc.

Crea el proceso *init* y le pasa el control.

El Kernel permanece en el mismo sitio donde se sitió durante todo el funcionamiento del sistema y controlará el resto de SW.

4ª Initrd

El arranque puede implicar que el medio desde el que se carga el kernel provenga de un sistema de ficheros concreto o la red. Por ello se necesitan módulos específicos alojados en initrd.

Initrd evolucionó a **initramfs** permitiendo un tamaño de disco variable.

5ª Init

Termina el proceso de arranque, dejando el sistema en modo multiusuario para que los usuarios trabajen en él.

Chequea los sistemas de ficheros, monta los sistemas de ficheros permanentes, activa el swapping, activa los demonios, etc.

Gestor de arranque GRUB (*Grand Unified Bootloader*):

Se instala en el MBR y hace las funciones del MBP.

Pregunta por el SO a iniciar: Si este es Linux, carga el kernel solicitado y le pasa el control para que el arranque continúe. Si es Windows, le pasa el control al mismo para que realice el arranque.

GRUB 2.0

Está situado en `/boot/grub/grub.cfg`, bajo ningún concepto se debe editar a mano y se genera bajo el comando `sudo-update-grub2`

El fichero `/etc/default/grub`, si es un fichero editable, por defecto tendrá las siguientes entradas a 0 (**GRUB_DEFAULT** y **GRUB_HIDDEN_TIMEOUT**) la primera es para el estado del grub y puede tomar el valor `saved` que será seleccionada por el administrador y el segundo es solo 0 cuando el SO operativo es solo Linux.

Las siguientes entradas tendrán `true` como valor por defecto (**GRUB_SAVEDEFAULT** y **GRUB_HIDDEN_TIMEOUT_QUIET**) siendo la segunda la cuenta atrás.

Y la opción **GRUB_TERMINAL**: Si se le aplica el valor `console`, este desactivará la interfaz gráfica.

Hay más opciones pero no son reseñables.

Grub permite:

- Editar las entradas, por ejemplo pulsando la tecla `e` y modifica las entradas de arranque para solucionar errores pero ojo, **esto no es permanente**.

- Consola interactiva GRUB pulsando la tecla `c`. Permite arreglar el arranque seleccionando otro `initrd`, módulos, etc.

Modo monousuario

Estado del sistema definido para realizar tareas administrativas y de mantenimiento, **requieren un control completo y no compartido**.

Solo realiza el montaje del sistema de ficheros raíz `/`.

Se puede acceder a todo el sistema pero muy pocos demonios estarán en ejecución, solo los necesarios.

Para entrar en este modo, se le debe indicar manualmente al MBR con una opción o parámetro en la interfaz de edición de GRUB.

No debemos cambiar la opción init ya que si no, no se llama a sulogin que permite tener acceso a todo el sistema.

No hay una solución consistente a este cambio, a no ser que se usen ficheros cifrados.

Y una solución paliativa sería que para poder cambiar los ficheros de configuración se pida una contraseña de acceso.

Aunque hace 2 años esto despertó inquietud, ya que descubrió que había un método de acceso dejando la tecla intro pulsada 70 segundos. Ya hay que estar muy aburrido como para probar eso.

Modo multiusuario

·Uso del **File System Check** (fsck) para chequear el sistema, por ejemplo si al apagar el ordenador, el sistema de ficheros se desmontó correctamente, no se chequea, aunque ciertos SOs con determinados **System File** fuerzan siempre o cada cierto tiempo el chequeo.

- Monta el SF root en modo rw.

- Chequea el resto de SF con fsck.

- Monta el resto de SFs

- Activa el swapping con *swapon -a*

- Activa las cuotas de disco: *quotacheck -a* y *quotaon -a*

- Lanza los procesos demonio (**cupsd, syslogd...**)

- Activa la red

- Lanza los demonios de red (**ssh, ntpd, nfsd...**)

- Limpia los sistemas de ficheros: /tmp, etc.

- Permite que los usuarios entren:

- Crea las terminales usando *getty* en modo texto y el terminal gráfico si fuese precioso.

- Borra en caso de que exista el fichero /etc/nologin, si este existe los usuarios excepto **root** no podrán acceder al sistema.

Niveles de ejecución de GNU/Linux

No solo existe el modo monousuario y multiusuario, también existen los siguientes:

- **0: Sistema apagado.**
- **1, s o S: monousuario, rescue o troubleshooting.**
- **2: Multiusuario sin funciones de red**
- **3: Modo multiusuario con funciones de red y terminales de texto**
- **4: Sin uso predefinido**
- **5: Multiusuario con la funcionalidad normal**
- **6: Reinicio**

Para saber en qué nivel está el sistema consultar **/sbin/runlevel**

Para cambiar el nivel de ejecución **/sbin/telinit**

Para consultar el nivel por defecto

/etc/inittab ---> **id:2:initdefault:**

/etc/init/rc-sysinit.conf ---> **env DEFAULT_RUNLEVEL=2**

Este nivel por defecto es obviado si se le pasa al GRUB al arrancar un parámetro numérico indicando el nivel que queremos arrancar.

Ficheros de inicialización

Para personalizar los niveles de ejecución **/etc/rc?.d/** donde la **?** es el nivel de ejecución, los cuales son ejecutados por init en el arranque mediante scripts que empiezan por K o por S.

K: Detención de demonios/matar procesos

S: Lanzar demonios o ejecutar funciones de inicio.

Esas carpetas donde se personalizan los niveles de ejecución contienen ficheros que todos ellos son enlaces simbólicos al fichero con el mismo nombre localizado en **/etc/init.d**

Manejar servicios

Mirar comandos después del primer *ls* hasta *update-rc.d apache2 remove*

Upstart

Es un el proceso de arranque/parada del sistema basado en eventos (reemplazo de init, aunque los ficheros siguen llamándose init) y realiza de forma asíncrona la dirección del inicio de las tareas y demonios, el control de los mismos mientras el sistema está encendido y la detención de los demonios durante el proceso de apagado.

/etc/init/rc-sysinit.conf es el archivo de configuración del sistema init y se le puede especificar desde que nivel empezar como default, como empezar con **start on** o que hacer al parar **stop on**, o incluso ejecutar scripts si los programamos dentro de este fichero.

Systemd

Reemplaza el proceso **init** en 2015, este permite gestionar por ejemplo un sistema de logs, aunque el sistema en si es un tanto complejo. Este se gestiona mediante servicios y targets (como niveles de ejecución).

Parada del sistema

En ocasiones es necesario apagar o reiniciar el sistema por el simple mantenimiento del mismo.

- Se notifica a los usuarios
- Se le envía la señal **TERM** a los procesos para su detención.
- Se paran los demonios
- Si queda algún usuario conectado se le echa del sistema.
- Si algún proceso se queda rezagado se fuerzan a finalizar con la señal **KILL**.
- Actuaciones de disco pendientes (Integridad del FS) con sync

Parada del sistema: shutdown

En los comandos

Caídas del sistema y problemas de arranque

·Caídas:

- Fallos HW, errores de HW irrecuperables, fallos de luz.
- Problemas ambientales, de E/S o de FS.

·Arranque:

- Fallos HW, no se puede leer el FS de los discos de trabajo.
- Hay áreas dañadas en el disco que no pertenecen al sistema de ficheros
- HW incompatible o errores en la configuración del sistema.

Al rearrancar mirar los mensajes que hay en el fichero **/var/log/messages**.

Comandos:

sudo-update-grub2: Genera el archivo grub.cfg utilizando los scripts de /etc/grub.d/

grub-set-default: Lo configura ajusta permanentemente.

grub-reboot: Lo configura un solo arranque.

sudo grub-install /dev/sda: Reinstalar el GRUB.

grub -mkpasswd -pbkdf2: Para crear contraseñas de acceso a las configuraciones del GRUB.

Swapon -a: Se activa el swapping (Particiones de intercambio).

quotacheck -a: Comprueba la cuota de disco.

quotaon -a: Activa la cuota de disco.

getty: Crea las terminales en el proceso de arranque.

ls /etc/rc2.d/ -la: Lista y contea el número de ficheros del directorio con los enlaces simbólicos.

/etc/init.d/myservice start: Arrancar un servicio de manera tradicional

service myservice start: Arrancar un servicio de manera **upstart**

/etc/init.d/myservice stop: Parar un servicio de manera tradicional

service myservice stop: Parar un servicio de manera upstart

ls /etc/init.d: Listar servicios de manera tradicional

service -status-all: Listar servicios de manera upstart

update-rc.d apache2 defaults: Añadir un servicio a todos los niveles

rm /etc/rc*/*myscript: Elimina un servicio a todos los niveles

update-rc.d apache2 remove: Elimina un servicio a todos los niveles.

initctl start <event>: Inicio de un evento

initctl stop <event>: Parada del evento

initctl status <event>: Estado del evento

exec <orden> <argumentos>: ejecuta la orden con los argumentos indicados.

shutdown [opciones] tiempo [mensaje]:

- **Sin opciones:** Monousuario.

- **-r:** Reiniciar.

- **-h:** parar (halt).

- **-c:** Cancelar.

- **-k:** Simular un apagado.

- **-tiempo:** +minutos, now, horas:minutos.

dmesg: Muestra los mensajes producidos durante el arranque del sistema.

Gestión de usuarios

Usuario: Persona que trabaja en el sistema.

Pseudo-usuario: Entidad, que sin ser una persona puede ejecutar programas o poseer ficheros, (se reservan identificadores de 0 a 499)

Ficheros de configuración

·/etc/passwd: Información de las cuentas de usuarios, este fichero tiene el siguiente formato:

-Nombre:password:uid:gid:gecos:home:Shell (Intérprete de órdenes).

-El propietario del fichero es root y el grupo root. Y los permisos del fichero son 644, se permite el acceso al fichero en modo lectura para poder leer la información del usuario y aunque las passwords estén cifradas, no se debería permitir el acceso a las passwords.

·/etc/shadow: Passwords cifradas (hash de las contraseñas).

-Nom:Pass:Changed:Minlife:Maxlife:Warn:Inactive:Expired:Unused

Changed: Fecha del último cambio de contraseña.

Minlife: No de días que han de pasar para poder cambiar la contraseña.

Maxlife: No de días máximo que puede estar con la misma contraseña sin cambiarla.

Warn: Cuántos días antes de que la contraseña expire (maxlife) el usuario será informado sobre ello, indicándole que tiene que cambiarla.

Inactive: No de días después de que la contraseña expire en que la cuenta se deshabilitará si no ha sido cambiada.

Expired: Fecha en la que la cuenta expira y se deshabilita de forma automática.

-Permiten que las contraseñas cifradas estén más restringidas. Tiene los permisos 600 y el usuario y grupo propietario es root.

-Este fichero guarda para cada usuario del sistema, la contraseña cifrada junto con su información de envejecimiento.

-Para cifrar una contraseña, se utilizan algoritmos criptográficos de generación.

MD5 (**M**essage-**D**igest algorithm **5**): Aplica funciones no lineales a los 17 segmentos de 32 bits de un bloque de 512 bits y se obtiene un resumen de 128 bits.

SHA (**S**ecure **H**ash **A**lgorithm): Parecido a MD4, pero genera resúmenes más grandes, que lo hacen más seguro contra ataques de fuerza bruta. Se pueden considerar 160, 224, 256, 384 o 512 bits para el resumen.

/etc/skel/: Contiene los ficheros que se copian automáticamente a cada \$HOME

/etc/shells: Se indican los shells permitidos. Si un usuario no tiene ningún Shell asignado, se le asigna automáticamente por defecto **/bin/sh**. Si se desea que un usuario no pueda entrar al sistema, se le puede asignar **/bin/false** o **/sbin/nologin**

Las **cuentas restrictivas** permiten limitar las acciones de los usuarios en el sistema.

El usuario que puede apagar el sistema es aquel que se le ha incluido un script con el comando shutdown y sus opciones.

/bin/rbash es el Shell restrictivo. rbash es un enlace simbólico a **/bin/bash**, se comporta como un intérprete normal pero tiene una restricción a ciertas acciones.

Añadir un nuevo usuario al sistema

- 1º Decidir el nombre de usuario, el UID y a los grupos a los que va a pertenecer.
- 2º Introducir los datos en los ficheros **/etc/passwd** y **/etc/group** **la contraseña es ***
- 3º Asignar un password a la nueva cuenta con *passwd*.
- 4º Si las shadow están activas, escribir la contraseña.
- 5º Establecer los parámetros envejecimiento de la cuenta.
- 6º Crear el \$HOME del nuevo usuario y establecer el propietario y grupo correspondiente con los permisos adecuados.
- 7º Copiar los ficheros necesarios por defecto del directorio **/etc/skel/**
- 8º Establecer las cuotas, mail, permisos, etc
- 9º Ejecutar cualquier tarea de inicialización propia del sistema
- 10º Probar la cuenta.

Se pueden usar herramientas de creación/modificación de cuentas de usuario como:

-adduser, usermod, deluser, newusers, users-admin.

·/etc/group: Información de los grupos y usuarios miembros.

Son colecciones de usuarios que comparten recursos o ficheros del sistema.

Nombre:x(passwd):GID:listaDeUsuarios

Los grupos pueden tener contraseña que son alojadas en **/etc/gshadow**.

Tipos de grupos:

- Primarios: Grupo especificado en **/etc/passwd**
- Secundarios: Otros grupos indicados en **/etc/group**

Cuando creamos un fichero se establece como grupo propietario el grupo activo del usuario.

Algunos Usuarios estándar:

- root**: Cuenta del administrador.
- bin, daemon, lp, sync**: Usados para poseer ficheros o ejecutar servicios.
- nobody**: Usuario sin privilegios.

Algunos Grupos estándar:

- root, sys**.
- bin, daemon, adm, lp, disk**.
- kmem**: Grupo propietario de los programas para leer la memoria del kernel
- user**: Usuario normal

Comandos:

passwd <nombre_usuario>: Asignar una contraseña al usuario.

mkpasswd --method=sha-512 contraseña salt: Crea una contraseña cifrada

pwconv: Crear y actualizar el fichero /etc/shadow.

pwunconv: Desactivar los shadow passwords.

md5sum <NomFich.ext> > <nomFich.md5>: Se obtiene la suma MD5 y la guarda en el fichero.

md5sum -c <nomFich.md5>: Chequea la suma de MD5.

Shasum [-a numBits] <NomFich.ext> > <NomFich.sha>: Se obtiene la suma SHA y la guarda en el fichero.

shasum -c <nomFich.sha>: Chequea la suma de SHA.

Chage <comando> (Admin):

-**d <ult_dia> <usuario>**: Último cambio de password.

-**m <min_dias> <usuario>**: Nº de días que han de pasar para poder cambiar la contraseña.

-**M <max_dias> <usuario>**: Nº de días máximo que han de pasar para poder cambiar la contraseña.

-**W <warn_dias> <usuario>**: Establece un aviso de que la contraseña expira un número de días antes de que expire, indicándole que tiene que cambiarla.

-**I <inac_dias> <usuario>**: No de días después de que la contraseña expire que la cuenta se deshabilitará de forma automática si la contraseña no ha sido cambiada.

-**E <exp_dias> <usuario>**: Fecha en la que la cuenta expira y se deshabilita de forma automática.

chsh: Permite al usuario cambiar su Shell (**Ojo, si se prohíbe un Shell no se podrá elegir con chsh, pero los usuarios que ya lo tenían asignado podrán seguir usándolo**).

newgrp: Si un usuario sabe la contraseña del grupo puede usarlo sin pertenecer a el con este comando

addgroup <grupo>: Crear un nuevo grupo.

groupmod <grupo>: Modificar un grupo existente.

Delgroup <grupo>: Eliminar un grupo.

newgrp <grupo>: Cambiar de grupo activo (lanza un shell)

gpasswd <grupo>: Asignar una contraseña a un grupo: Si el usuario no pertenece al grupo, pero el grupo tiene contraseña, se le solicita y pasa a ser su grupo activo.

gpasswd -a <user> <grupo>: Añadir un usuario a un grupo.

groups <usuario>: Grupos a los que pertenece un usuario.

id <usuario>: Lista el identificador del usuario y los grupos a los que pertenece.

grpck: Chequea la consistencia del fichero de grupos

Procesos

Un proceso es un programa en ejecución. Un sistema de tiempo compartido como Linux permite **multiusuarios** y **multiprocesos**, pero solo puede ejecutar **1 proceso a la vez en la CPU.**

La CPU conmuta entre procesos según el **cuanto** de tiempo de cada proceso.

Procesos: modo de ejecución

- Modo **usuario**: se ejecuta el código normal del programa.
- Modo **núcleo**: se ejecuta el kernel en nombre del proceso:
 - Llamadas al sistema (creación de hilos, etc).
 - Excepciones: Situaciones excepcionales como **división por cero, errores de direccionamiento, etc.**
 - Interrupciones: Los dispositivos periféricos interrumpen para notificar al kernel (**terminación de E/S, cambio de estado, etc**).

Procesos: tipos de procesos

- **Procesos de usuario**:
 - Creados por un usuario real.
 - Se ejecutan en modo usuario, excepto en los casos anteriores.
- **Procesos demonio**:
 - No asociados a un usuario
 - Se ejecutan en modo usuario, excepto en los casos anteriores.
 - Realizan tareas periódicas relacionadas con la administración del sistema.
- **Procesos núcleo**:
 - No asociados a un usuario.
 - Corresponden al código del kernel.
 - Se ejecutan siempre en modo núcleo.
 - Tareas de administración más delicadas

Procesos: comando ps

- **USER**: Usuario que lanzó el programa.
- **PID**: Identificador del proceso.
- **PPID**: Identificador del proceso padre.
- **%CPU**: Porcentaje de la CPU consumido por este proceso (en ese momento).
- **%MEM**: Fracción de memoria consumida (es una estimación).

- VSZ**: Tamaño virtual (código + datos + pila) en KB.
- RSS**: Memoria real usada en KB (VSZ incluye a RSS).
- TTY**: Terminal asociado con el proceso.
- STAT**: Estado del proceso.
 - R**: En ejecución +/ò **N**: (Prioridad baja >0) +/ó **L**: Tiene páginas bloqueadas en mem.
 - S**: Durmiendo +/ó **<**: (Prioridad alta <0) +/ó **s**: Líder de sesión.
 - T**: Parado (señal o trace) +/ó **I**: Es multihilo.
 - Z**: Proceso Zombie +/ó **+**: Proceso foreground.
 - D**: Durmiendo ininterrumpible (E/S)
- START**: Hora en la que empezó.
- TIME**: Tiempo que lleva en ejecución
- COMMAND**: Comando que lo ejecutó

Control/gestión de la actividad de la CPU

- Numero nice** y prioridad de procesos:

Al lanzar un proceso se le asigna un número nice o prioridad estática (desde -20 a 19).

•**Envío de señales a los procesos**: Echar un vistazo al apartado comandos después del último renice. Los procesos en estado D o Z no se detienen aunque reciban la señal KILL.

•**UPTIME**: Si los valores son altos es porque el sistema se está usando mucho, pero que un valor sea alto depende del número de núcleos. Y valores bajos no significa tiempo de respuesta bajo.

- vmstat**:

- r**: Nº de procesos esperando su tiempo de ejecución.
- b**: Nº de procesos en espera ininterrumpible.
- swpd**: Cantidad de MV ocupada.
- free**: Cantidad de MV sin usar.
- buff**: Cantidad de memoria empleada como buffers para E/S.
- cache**: La cantidad de memoria empleada como caché de disco.
- si**: Cantidad de memoria traída del espacio de MV desde disco en KB/s.
- so**: Cantidad de MV al disco en KB/s.
- bi**: Bloques recibidos desde un dispositivo.
- bo**: Bloques enviados a un dispositivo.

- in: Nº de interrupciones por segundo
- cs: Nº de cambios de contexto por segundo.
- us: Tiempo de usuario como porcentaje de tiempo total (modo usuario).
- sy: Tiempo de sistema como porcentaje de tiempo total (modo núcleo).
- id: Tiempo de inactividad como porcentaje de tiempo total.
- wa: Tiempo usado en espera de E/S.

·**Espacio de paginación:**

-El tamaño adecuado de paginación depende de la memoria requerida por los procesos, la cantidad de procesos simultáneos, la demanda del sistema, etc.

-Se puede tener una partición de intercambio o un fichero de intercambio, ¿qué opción es la mejor?

Depende de si se tiene o no suficiente espacio en disco. Un fichero lo puedes crear/borrar cuando quieras, una partición swap no.

·**at:**

- atd: Demonio que ejecuta las ordenes
- atq: Consulta la lista de órdenes
- atrm: Eliminar órdenes.

·**cron:**

-crond: Demonio encargado de ejecutar ordenes.

-crontab: Establecer las tareas a ejecutar:

- e: Añadir o modificar tareas
- l: Listar tareas
- r: Eliminar tareas.
- <minuto> <hora> <día_mes> <mes> <día_semana> <user> <comando>

Si la máquina no está encendida cuando se ha requerido cron, este no se lanza.

-anacron: No asume que la máquina está siempre encendida

·**strace:** El comando nos permite observar qué es lo que está haciendo un proceso.

Carpeta /proc

ps y top leen la información que necesitan de /proc, cada proceso tiene una carpeta y en esa carpeta hay información sobre el mismo:

cmdline: Línea de comandos con que fue iniciado.

cwd: Enlace simbólico al directorio actual del proceso.

environ: Las variables de entorno en el momento de invocación.

exe: Enlace simbólico al fichero ejecutado.

fd: Carpeta con cualquier descriptor de fichero abierto.

maps: Información de mapeo de memoria.

root: Enlace simbólico a la raíz del sistema (/).

stat: Estado del proceso.

statm: Uso de memoria.

Rastreo de señales y llamadas al sistema

Cuidado con los bucles iterativos de creación de carpetas y ficheros ya que genera un árbol que ni `rm -R` es capaz de manejarlo.

Comandos:

ps: Nos muestra la información de un proceso. (Ver **Procesos: comando ps** más arriba).

nice -5 nautilus: Lanza nautilus con nº nice incrementado en 5.

nice -10 nautilus: Lanza nautilus con nº nice decrementado en 10 (solo root).

renice 14 890: Proceso 890 con prioridad 14.

renice 5 -u carlos: Prioridad 5 para todos los procesos del usuario carlos.

kill -señal <pid>: Señal es un número y acaba con el proceso.

-La **señal 9** fuerza la salida del proceso y no se puede capturar ni bloquear **SIGKILL**.

-La **señal 19** para un proceso **SIGSTOP**.

-La **señal 18** reinicia un proceso **SIGCONT**.

-La **señal 1** cuelga/detiene el terminal **SIGHUP**.

-La **señal 2** interrumpe un proceso **SIGINT**.

-La **señal 3** para el proceso **SIGQUIT**.

-La **señal 11** violación del segmento **SIGSEGV**.

-La **señal 15** parar el software **SIGTERM**.

Hay más señales, pero no son tan reseñables.

kill <pid>: Manda una señal por defecto al proceso pid (**SIGTERM**, número 15).

killall <comando>: Permite mandar una señal a todos los procesos con un determinado comando.

pkill o skill: Envía una señal usando el nombre u otros atributos (uid, gid, etc).

uptime: hora actual, cuánto tiempo lleva en marcha el sistema, número de usuarios conectados, y carga media del sistema (el número medio de procesos del sistema que durante los últimos 1, 5 y 15 minutos han estado en los estados R o D).

ps tree: Visualiza un árbol de los procesos en ejecución.

top o htop: Visualiza los todos los procesos paginados ordenados decrecientemente por uso de CPU, también muestra **SHR** siendo este la memoria compartida disponible

vmstat: Muestra la información sobre la memoria virtual desde el último reinicio.

at: Ejecuta tareas a una determinada hora.

cron: Ejecuta tareas periódicamente.

anacron: Obvia si el ordenador está apagado.

strace: Nos permite observar qué es lo que está haciendo un proceso.

-p <pid>: Rastrea un proceso ya iniciado.

<comando>: Iniciar un proceso y rastrearlo.

-o salida.txt <comando>: Utilizar un fichero para guardar la salida.

swapon -s: Nos da un listado de particiones o ficheros activos.

swapon /dev/sdd1: Activar una determinada partición.

swapoff /dev/sdd1: Desactivar una determinada partición.

mkswap /.fichero_swap: Crear un fichero de paginación, aunque se deben tener permisos de root.

Sync: Se deben tener permisos de root.

swapon /.fichero_swap: Se activa la paginación del fichero, aunque se deben tener permisos de root.

free: Se obtiene información sobre el uso de memoria (lo mismos campos que *top*).

df: Muestra la capacidad, el espacio libre y el punto de montaje de cada FS del equipo.

-i: Nos permite mostrar información sobre los nodos-i.

du: Muestra el espacio usado por cada subdir del dir actual.

Deberíamos poner **--max-depth=1**, si no, nos mostrará todas las carpetas.

du cuenta bloques del sistema de ficheros, estén o no completamente ocupados.

iotop

iotstat: Muestra estadísticas de la CPU, dispositivos y particiones de E/S.

Discos rígidos

Los HDD están compuestos por:

-Plato: Cada uno de los discos que se encuentran apilados. La escritura cambia el estado del material

-Cabezal: Es un brazo que se mueve sobre el plato

-Pista: Son cada una de las líneas esféricas que se pueden formar en un plato.

-Cilindro: Conjunto de varias pistas apiladas.

-Sector: Cada una de las divisiones que se hace de la circunferencia que se forma en el disco.

Indicando cilindro, cabeza y sector podemos acceder cualquier dato del disco.

Archivos

Unidad de almacenamiento lógico no volátil que agrupa un conjunto de información relacionada entre si bajo un mismo nombre.

Sistema de Archivos

Partición: Porción de un disco a la que se le dota de identidad propia y se manipula como entidad lógica independiente. Las particiones deben formatearse para que se creen las estructuras necesarias (*Journal*).

Bloque: Agrupación lógico de sectores físicos del disco, que supone la unidad de transferencia mínima que usa el SA.

-Tamaño de bloque **pequeño:** Mayor número de operaciones de E/S y menor fragmentación.

-Tamaño de bloque **grande:** Menor número de operaciones de E/S y mayor fragmentación.

Agrupación: Conjunto de bloques gestionado como una unidad lógica de almacenamiento.

Estructura del SA

-Volume **Boot Record** o **bloque de carga** contiene código de arranque del ordenador por el iniciador ROM de esa partición.

MBR apunta al VBR de la partición activa.

Se añade un **número mágico** para la comprobación del bloque de carga.

-**Metainformación:** Nodos-i, FAT, mapas de bits, etc.

-**Descriptores físicos de archivos:** Nodos-i, registros de Windows-NT, etc.

Tienen una estructura y tamaño muy dependiente del SO.

El número de descriptores debe ser proporcional al tamaño total del disco.

Para gestionar el espacio libre se pueden usar mapas de bloques o mapas de descriptores de archivos.

Los bloques de datos son el lugar donde se almacena la información.

El **Servidor de archivos** es el componente del SO que se encargará de gestionar el acceso a archivos. Esta organizado por capas:

1º Sistemas de archivos virtual (+ abstracto).

2º Módulo de organización de archivos.

3º Servidor de bloques.

4º Manejador de dispositivos (- abstracto)

Directorios

Es un fichero con un formato determinado, el contenido de un directorio es una serie de registros, una por cada fichero contenido en él.

Cada registro tiene, al menos, el nombre del fichero y el puntero al descriptor físico correspondiente.

Una ruta se interpreta de forma recursiva hasta:

- No se tienen permisos.**
- Se ha encontrado el i-nodo del archivo.**
- No se encuentra el siguiente elemento de la ruta.**

La llamada open() termina con la lectura del i-nodo. La verificación de permisos se hace con los datos del i-nodo. Un directorio no es un i-nodo.

Asignación de bloques

•Contiguos:

-Todos los bloques se encuentran contiguos en el disco. Aunque no es usado porque es necesario conocer el tamaño del archivo creado, la fragmentación del disco, etc.

•No contiguos:

- Los bloques del archivo se encuentran en cualquier posición del disco.
- Se produce menos fragmentación.
- Es la opción usada en la mayoría de SO.

Para saber que bloques no contiguos pertenecen a cada archivo se usan listas enlazadas o índices

FAT de 12 bits: 4K agrupaciones.

FAT de 16 bits: 64K agrupaciones.

FAT de 32 bits: 2^{28} agrupaciones (solo usa 28 bits).

Tamaño de fichero en directorios \rightarrow 32 bits. Tamaño máximo $\rightarrow 2^{32} - 1 = 4\text{GB} - 1$.

·Esquema híbrido:

Por cada nodo-i incluir:

- Punteros directos a los diez primeros bloques.
- Puntero a un bloque índice de primer nivel
- Puntero a un bloque de índice de segundo nivel.

Los **extents** son bloques índice especiales que marcan una zona contigua del disco.

La gestión del espacio libre se necesita para asignar espacio a los nuevos archivos o a los que se les vaya a rellenar de datos.

Los **mapas de recursos** son **mapas de bits** y estos incluyen un bit por recurso, que será 1 si el recurso está libre y 0 si no lo está.

También existe una lista de punteros hacia recursos libres.

Incremento de prestaciones

Acceso a memoria de orden en **nanosegundos**.

Acceso a disco de orden en **milisegundos**.

·Almacenamiento intermedio de los datos

- Mantener una caché de datos en MP.
- Caché de nombres: Lista con {nombre, nodo-i}.
- Caché de bloques: Colección de bloques leídos o escritos recientemente.

Se usa Least Recently Used para comprobar esta caché.

-Los **bloques sucios** son aquellos que han sido cambiados en caché pero no en Memoria Principal.

-**Escritura inmediata (write-through)**: Siempre actualizado.

-**Escritura diferida (write-back)**: Actualizamos cuando el bloque salga de la caché.

-**Escritura periódica (delayed-write)**: Establecer un tiempo periódico para las actualizaciones. Compromiso entre rendimiento y fiabilidad.

No se debe quitar un disco del sistema sin antes volcar los datos de la cache (**comando sync**).

Montar y desmontar SFs

• Hay un único SF lógico en la que se organizan todos los dispositivos de almacenamiento disponibles.

Cada partición tiene su propio sistema de ficheros con su directorio raíz y su jerarquía.

En el arranque se monta primero la partición correspondiente a dicha raíz y luego cualquier partición auxiliar.

• **/etc/fstab** es un fichero con información sobre todos los SFs a montar o ya montados y las zonas de swap a activar. La disposición del fichero es la siguiente:

-fi_especial: Fichero especial de bloques (/dev/...)

-pto: Directorio que sirve de punto de montaje. Hay que tener especial cuidado, ya que los permisos normalmente son 755, **pero si el sistema está montado en solo lectura, aunque tenga los permisos 755 no se le permitirá escribir ni ejecutar.**

-tipo: Tipo de SF (nfs, ntfs, ext2, etc).

-Opciones para el proceso de montaje separadas por , y sin espacios.

-rw: Lectura/Escritura (por defecto).

-ro: Sólo lectura.

-suid/nosuid: Permitido (o no) que los bits suid o sgid tengan efecto.

-auto/noauto: Montar automáticamente (o no).

-exec/noexec: Permitir (o no) la ejecución de ficheros.

-usrquota, grpquota: Activar cuotas.

-uid=num, gid=num (en formato numérico): Propietario y grupo del propietario de los SFs.

-umask=numero: Permisos de los ficheros (640 por defecto).

-dev: Interpretar ficheros especiales en el SA.

-sync: Forzar a que todas las operaciones sean síncronas.

-user: Permite que los usuarios puedan montar el sistema de ficheros.

-users: Cualquiera podrá desmontarlo.

-nouser: Solo root puede montar el SF.

-owner: Permite que el usuario propietario pueda montar el SF.

-defaults: rw, suid, dev, exec, auto, nouser, async.

-dump_freq: Frecuencia del dump para hacer una copia de seguridad de ese SF mediante el comando *dump*, **aunque ya no se usa este comando**

-pass_num: En tiempo de arranque, en qué orden hay que chequear los SFs.
Usar el comando *fsck* para comprobar el estado.

Comprobación del SF

Para poder hacer uso de los comandos de chequeo los SF deben estar desmontados o montados en modo de sólo lectura.

El SF raíz debe estar montado en modo solo lectura, ya que al no poderse desmontar, no se le va a permitir la escritura ni la ejecución a los usuarios normales.

Journaling: Para evitar la verificación completa (*fsck*) de SF de gran tamaño, se implementa un modelo de control basado en logging.

Las suboperaciones que modifiquen los metadatos y datos de un mismo archivo se agrupa en la misma transacción. Si el sistema falla, las acciones parcialmente realizadas se deshacen o completan recorriendo el log.

Creación del SF

- 1º Realizar la conexión física.
- 2º Crear un fichero especial de dispositivo, si este fuera necesario.
- 3º Crear las particiones con *fdisk*.
- 4º Crear el SF con *mke2fs*
- 5º Etiquetar la partición con *e2label*.
- 6º Crear el directorio que hará de punto de montaje.
- 7º Montar el nuevo SF
- 8º Actualizar */etc/fstab* con las opciones necesarias.

Diferencias ext2, ext3 y ext4:

ext3 tiene el mismo formato que ext2 pero además se le añade el **journal** que permite recuperar la consistencia tras una caída del sistema.

ext4 tiene el mismo formato que ext3 pero incluye una extensión que describe un conjunto de bloques lógicos y se retrasa la reserva de memoria.

ext2: Muy rápido en general, pero no tiene journaling. Se puede usar en un SF en el que se guardarán ficheros temporales.

ext3: Buen rendimiento en general y journaling.

ext4: Menor uso del CPU y mayor rapidez en los procesos de lectura y escritura que ext3. Estándar de facto en Linux.

Cuotas de disco

-**Límite hard:** El usuario no puede sobrepasarlo, si lo hace, ya no podrá usar más bloques o crear más ficheros.

-**Límite soft:** Es inferior al límite hard y se puede sobrepasar durante un cierto tiempo, siempre que no se alcance el límite hard.

-**Período de gracia:** Tiempo durante el que se puede sobrepasar el límite soft.

Establecer cuotas de disco

1ª Instalar el paquete quota: *apt-get install quota*

2ª Indicarlo en **fstab**, en ext3 y ext4 es diferente la salida por terminal.

3ª Remontar la partición para que se activen las opciones: *mount -o remount <NombreDelRootAREmontar>* .

4ª Añadir el contenido de los ficheros de control de cuotas: *quotacheck -avugm*

5ª Activar las cuotas: *quotanon -avug*

6ª Desactivarlas: *quotaoff -avug*

7ª Editar la cuota del usuario carlos: *edquota carlos*

8ª Establecer el periodo de gracia: *edquota -t*

9ª Copiar cuotas: *edquota -up carlos miguel*

10ª Estadísticas de las cuotas: *repquota /dev/sdb1*

Administración de volúmenes dinámicos

·**RAID:** Array redundante de discos independientes. Varias unidades de disco convertidas en una sola unidad lógica y se puede implementar mediante SW o HW.

-**Nivel 0:** Expande la información en diversos discos y que se ven en un SF.

Aumenta el espacio según el número de discos usado.

Se consigue E/S paralela en lecturas y escrituras, siempre que los bloques a tratar no sean del mismo disco.

No hay redundancia de datos.

-**Nivel 1:** Se utilizan dos o más discos duros, que forman un único SF.

Son discos espejo (guardan la misma información).

Si hay redundancia de datos.

Las lecturas pueden ser en paralelo, la escrituras no.

Cuando un disco falla el sistema sigue trabajando con el otro.

-Nivel 4/5: División de los datos a nivel de bloques

En RAID 4: Se usan mínimo 3 discos duros de los cuales 1 almacenará la paridad de los otros discos que son usados para datos. Esto trae un problema, ese disco duro es un cuello de botella. **RAID 5:** Reparte la paridad entre todos los discos, no solo 1.

Se consigue un dispositivo de almacenamiento más grande

Si hay redundancia de datos.

Lecturas y escrituras en paralelo

RAID 5 ofrece la mejor relación rendimiento-coste en un entorno con varias unidades.

Cat /proc/mdadm muestra por pantalla todos los sistemas RAID. Tienen un flag las cuales son usadas para que en el proceso de arranque sean detectadas y activadas.

Cuando hayamos creado el SF debemos añadirlo en el fichero */etc/fstab* para montarlo en tiempo de arranque.

Spare: Disco de repuesto.

/etc/mdadm.conf: Se puede configurar la creación y todo ahí también.

Paridad: Cada vez que se escriben datos, se calcula el XOR bit a bit (1 unos impar, 0 unos par) de los bloques implicados en cada disco. Si por ejemplo un disco falla, el disco o los discos que contengan paridad pueden trabajar en lugar del dañado.

·Logical Volume Management: Agrupa las particiones en volúmenes.

Comandos:

mount <opciones> <FicheroEspecialBloque> <PtoMontaje>: Si lo ejecutamos como root escogerá las opciones los argumentos que falten del fichero */etc/fstab*

-t <tipo-sf>: Tipo de sistema de ficheros.

-r: Montaje en modo sólo lectura.

-w: Montaje en modo lectura/escritura.

-o <opcionesMontaje>: Opciones del proceso de montaje.

-nosuid

-exec

-remount

-etc

Mount -a: Se montan todos los SF siempre que la opción del fichero */etc/fstab* de auto esté activa para cada punto de montaje.

umount <PtoMontaje> ó <FicheroEspecialBloque>: Desmontar un sistema de ficheros, pero si este está **ocupado** no se podrá desmontar.

fuser: Nos permite conocer que ficheros están siendo usados y qué procesos lo usan (f: fichero abierto, c: directorio de trabajo, e: ejecutando un fichero, etc.).

-m: Ficheros montados

-v: Verbose

lsdf: Obtener un listado de todos los ficheros abiertos.

fsck: Comprobar y reparar el estado de los SFs.

e2fsck: Comprobar y reparar el estado de los sectores de los SFs (igual que fsck).

fdisk o parted: Crear las particiones.

mke2fs -t <tipo-sf> <ruta-de-montaje>: Crear el sistema de ficheros.

e2label: Asigna una etiqueta al SF.

e4defrag: Desfragmentación online de ext4.

tune2fs: Conocer y ajustar parámetros de un SF ext4/ext3/ext2.

-l <dispositivo>: Listar el contenido del superbloque del SF.

-c <max-mount-counts> <dispositivo>: Establecer el no de montajes máximo sin realizar un fsck.

-i <numero[d|m|w]> <dispositivo>: Indicar el tiempo máximo entre dos chequeos.

-L <etiqueta> <dispositivo>: Poner una etiqueta al sistema de ficheros.

-m <porcentaje> <dispositivo>: Fijar el porcentaje de bloques reservados para procesos especiales (de root). Por defecto, 5 %.

apt-get install quota: Instalación del paquete de cuotas.

quotacheck: Comprobador de cuotas.

-a: Todos los dispositivos con cuotas.

-v: verbose.

-u: Cuotas para usuarios.

-g: Cuotas para grupos

-m: No remontar los archivos en modo solo lectura.

quotaon -avug: Se activan las cuotas.

quotaoff -avug: Se desactivan las cuotas.

edquota carlos: Editar la cuota del usuario carlos.

edquota -t: Establecer el periodo de gracia.

edquota -up carlos miguel: Copiar las cuotas del usuario carlos al usuario miguel.

repquota /dev/sdb1: Imprime por pantalla las estadísticas de las cuotas

apt-get install mdadm: Instalar la herramienta de administración de RAIDS.

mdadm: Permite crear o administrar un dispositivo RAID.

Ejemplo: *mdadm --create/dev/md1 --level=1 --raid-devices=2 /dev/sda1 /dev/sdc1*

--create <directorio (suele ser /dev/mdx (donde x es el siguiente número a crear RAID))> *--level=y* (donde y es el nivel de RAID.) *--raid-devices=z* (donde z es el número de discos para crear la raid) <PrimerDisco> <HastaElDiscoQueVayamosAMeter>

--create --force: Para crear una RAID en la que la partición ya tenía datos

<RutaRaid(x)> -a <RutaDelDisco/Particion>: Crea un disco de repuesto.

mdadm /dev/md2 -a /dev/sdc3

--grow /dev/mdx -n x

mke2fs -t ext4 /dev/md1: Crea un SF sobre el RAID que hemos creado antes

mdadm --detail --scan /dev/md1: Información sobre el estado del RAID

Introducción

Common Unix Printing System: Conseguió hacer mucho más fáciles las tareas de administración de impresoras. Basado en el protocolo HTTP, usa operaciones POST para imprimir y GET para ver el estado.

Los archivos con extensión PPD le sirven a CUPS como archivo de configuración de opciones de la impresora el cual usa un lenguaje nativo que entiende la impresora

No todos los comandos son exclusivos de CUPS.

/etc/cups/classes.conf: Información de las clases.

/etc/cups/cupsd.conf: Configuración del demonio.

/etc/cups/printers.conf: Información impresoras.

/etc/cups/ppd/: Ficheros de filtro para cada impresora.

/var/spool/cups: Directorio de spool.

Al añadir una nueva impresora, o realizar cambios de configuración, hay que reiniciar el demonio.

Browsing: Los equipos clientes localizan y usan la impresora del servidor de impresión, sin necesidad de instalarla previamente.

Internet Printing Protocol

Page Description Languages: Describen como representar una página en papel usando la tinta o tóner.

- PostScript:** Adobe Systems por ejemplo.

- Printer Control Language:** Ficheros binarios por ejemplo.

- Portable Document Format:** Realiza una ccompresión de la información

Los directorios **spool** son usados por las colas de impresión y sirven para guardar los trabajos pendientes de imprimir hasta que se puedan enviar a la impresora. También guardan un fichero con las propiedades del trabajo de impresión. Se encuentran en /var/spool .

¿Cómo podemos convertirnos en servidores de impresión? Dando los permisos oportunos para que la impresora pueda ser usada de forma remota

Comandos:

CUPS:

lpadmin: Se está configurando una impresora

- p <NombreDeLaImpresora>:** Añadir una nueva impresora

- E:** Activa la impresora

- v <parallel:/dev/lpx(siendo x el puerto de la impresora)>:** Agiliza la impresora por el puerto paralelo

serial:

ipp:

lpd:

usb:

-m <PPD>: Se indica el fichero de configuración de la impresora

-x <impresora>: Elimina una impresora

-c <nombreDeLaClase>: La primera vez que se ejecuta se crea el conjunto, y la próxima vez que se use el mismo nombre se añadirá esa impresora a la clase.

cupsenable <impresora>: Deshabilita la impresora.

cupsenable <impresora>: Habilita la impresora.

cupsreject <impresora>: Deshabilita la cola de impresión.

cupsaccept <impresora>: Habilitará la cola de impresión.

lp -d <impresora> <fichero 1> [fichero2]: Imprimir un fichero o varios

lpr -P <impresora> <fichero 1> [fichero2]: Imprimir un fichero o varios

cancel <id_tra1> [id_tra2] [impresora]: Elimina un trabajo o varios de la cola de impresión.

lprm [-P impresora] <id_tra1> [id_tra2]: Elimina un trabajo o varios de la cola de impresión.

lpq -P <impresora>: Consulta la cola de impresión y el estado de los trabajos.

lpinfo -m: Consultar el listado de impresoras soportadas.