

APUNTES PAS



*PROGRAMACIÓN Y ADMINISTRACIÓN DE
SISTEMAS*

Lola Romero Espejo
Lucia Cabezuelo Pérez

CURSO 2021/22

TEMA 1

INTRODUCCIÓN A LA ADM DE SISTEMAS



UN BREVE RECORRIDO POR LOS SISTEMAS OPERATIVOS

GENERACIONES DE LOS SO

- **Primera generación (1945-1955):**
 - Ordenadores muy voluminosos, tarjetas perforadas
 - No necesitaban SO, el operario introducía la tarjeta con el código correspondiente
- **Segunda generación (1955-1968):**
 - Aparecen los transistores
 - Los ordenadores disminuyen de tamaño
 - Van apareciendo los SO
 - JCL: Lenguajes de control de tareas
 - Lenguajes de alto y bajo nivel (assembler)
 - Superusuarios y usuarios
 - Dispositivos de entrada/salida
- **Tercera generación (1968-1981):**
 - Aparecen los circuitos integrados (LSI: Large Scale Integration)
 - Equipos de propósito general
 - Escalabilidad, multiprogramación y discos duros
- **Cuarta generación (1981-2001):**
 - VLSI: Very Large Scale Integration y microprocesadores
 - Conectividad con dispositivos
 - Aplicaciones cliente/servidor
 - Máquinas virtuales
- **Quinta generación:**
 - Tendencia a que los ordenadores los puedan manejar personas no expertas en informática

SO MÁS USUALES

MAC OS: 1º SO de Apple para ordenadores Macintosh

WINDOWS: SO gráfico para ordenadores personales cuyo propietario es Microsoft

UNIX: SO, multitarea y multiusuario

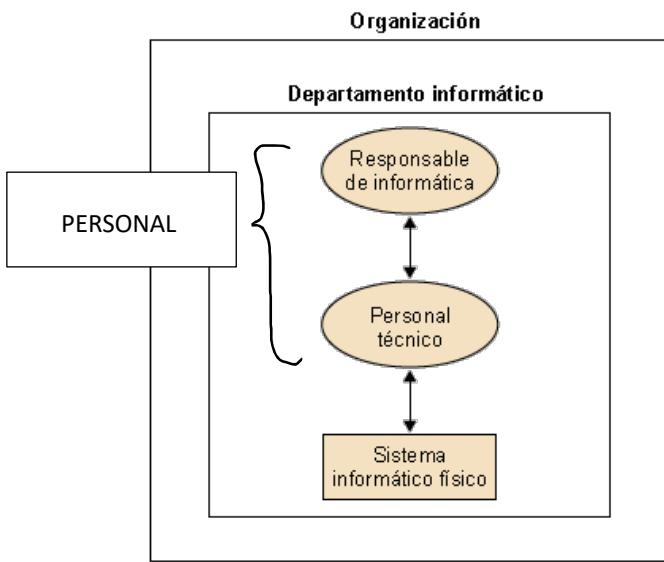
EL SISTEMA INFORMÁTICO Y LA ORGANIZACIÓN

ÓPTICAS POSIBLES DEL SISTEMA INFORMÁTICO PARA LA ORGANIZACIÓN

- **Para la organización:** es un **departamento** como cualquier otro
- **Para los informáticos:** es un conjunto de **redes, servidores y ordenadores personales**
- **Para los usuarios:** es una **herramienta** más que proporciona la organización para mejorar su tarea
- **Para la dirección:** es una gran **base de datos** para hacer consultas que puedan ayudarle en la toma de decisiones

EL DEPARTAMENTO DE INFORMÁTICA

Se encarga de mantener y gestionar el Sistema Informático



SISTEMA INFORMÁTICO FÍSICO:

- **HARDWARE**

- Servidores: múltiples servidores especializados -> + control, - riesgo de fallos
- Ordenadores personales (PCs)
- Cableado y electrónica de red: cables y electrónica de control de la red)
- Centro de datos: sala con condiciones físicas y de seguridad para instalar los servidores

- **SOFTWARE:**

- Sistemas operativos
- Software empresarial de base
- Aplicaciones específicas

FUNCIÓNES DEL DEPARTAMENTO DE INFORMÁTICA

Administración de servidores: Instalar, mantener y reparar los servidores que prestan los servicios del sistema informática

Administración de usuarios

- Atención de las necesidades de los usuarios
- Mantenimiento de sus equipos de sobremesa

Administración de la red:

- Responsabilidad sobre la parte física de la red
- Asegurar que está en buen funcionamiento y que llega a todos los puntos

Administración de los datos

- Mantener la integridad de la inf de la organización
 - i. Esta información debe estar en los servidores (*aunque a veces se encuentra distribuida por todo el sistema*)

Administración de la web: Mantenimiento del servidor web y de su contenido

Administración de la seguridad: La seguridad informática es compleja: desde la seguridad de la inf existente hasta la protección física de equipamiento contra robos ,incendios...

Desarrollo: Una organización suele necesitar un software específico, a veces en lugar de comprarlo, se desarrolla

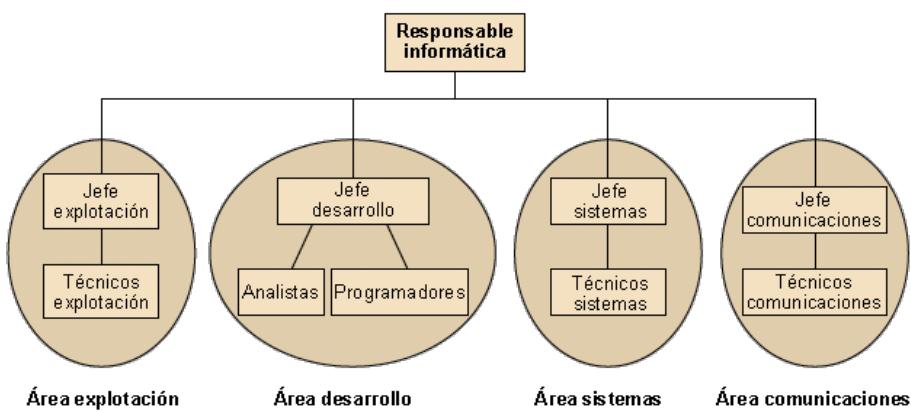
Responsable de informática: es el enlace entre las necesidades de la empresa y el trabajo que se lleva a cabo en el departamento

- Decide que software comprar, servidores necesarios...
- **UCO**
 - i. Área de sistemas: servidores, correo, comunicaciones...
 - ii. Área de gestión: usuario, licencias de software...

Interdependencia: cada función tiene unas tareas predefinidas

- No son independientes, sino que tienen que trabajar coordinadas

ORGANIGRAMA DEL DEPARTAMENTO DE INFORMÁTICA



LA FIGURA DEL ADMINISTRADOR DE SISTEMAS

ROL

Un administrador de sistemas es la persona con el poder y la responsabilidad de establecer:

- Acciones
- Procedimientos
- Normas

Para lograr que el sistema informático sea:

- Eficiente
- Seguro
- Fiable
- Y amigable

CUALIDADES: Autoridad+responsabilidad+servicio+cooperación

DEDICACIÓN

Idealmente sería una persona encargada solo de la administración pero generalmente comparte esa labor con otro tipo de trabajo.

LABORES

Idealmente sería uno de los miembros del departamento de informática pero generalmente puede hacer casi todo el trabajo del departamento de informática, como reparar hardware.

¿QUÉ SE ESPERA DEL ADMINISTRADOR?

- Amplios conocimientos de todo el sistema
- Capacidad para tomar decisiones
- Ambición y espíritu de superación
- Eficacia y moral
- Responsabilidad

TAREAS DETALLADAS

1. Planificar y administrar el entorno físico
2. Planificar los cortes de suministro para realizar actualizaciones o para administrar los dispositivos
3. Localizar, reparar y reemplazar componentes defectuosos (a nivel hardware)
4. Configurar y mantener la conectividad entre los hosts (redes)
5. Instalar y mantener dispositivos del sistemas, hardware y drivers, especificar dispositivos soportados
6. Mantenimiento software
7. Documentación
8. Soporte a usuarios
9. Servicios
10. Seguridad, copias de seguridad

ESTRATEGIAS

Estrategia del administrador de sistemas al realizar una tarea:

1. Planearlo antes de hacer los cambios
2. Hacer los cambios reversibles
3. Realizar los cambios de forma incrementada
4. Probarlo antes de hacerlo público
5. Conocer realmente cómo trabajan las cosas

Cuando se realice cualquier modificación:

- Precaución *antes de...*
- Probarlo *después de...*

Es buena idea disponer de un cuaderno de bitácora:

- En el se registran todos los cambios realizados sobre la configuración del sistema
- Sirve para uno mismo y para los demás

La mayoría de las veces tendremos que editar múltiples ficheros de configuración, para lo que necesitaremos un editor de texto

- Vi (o su versión mejorada vim) : editor estándar
- Pico es mas simple de utilizar
- No podremos utilizar gráficos como gedit o code

DEVOPS:Conjunto de prácticas que agrupan el desarrollo de software (Dev) y las operaciones de TI (Ops)

SOFTWARE LIBRE

HISTORIA

AÑOS 60-70

- Pocas computadoras: grandes computadores (o mainframes) , muy pocos y muy caros
- Se desarrolla software artesanal:
 - El negocio estaba en el hardware ya que había muy poca variedad de software muy específico

AÑOS 80

Emerge Richard Stallman

- 1984: Comenzó a trabajar en el proyecto GNU
- 1985: Funda Free Software Foundation (FSF)
 - Se introducen los conceptos de:
 - Free Software
 - Copyleft

LAS 4 LIBERTADES

LIBERTAD 0: Usar el programa con cualquier propósito

LIBERTAD 1: Estudiar cómo funciona el programa y adaptarlo a tus necesidades

LIBERTAD 2: Distribuir copias, con lo que puedes ayudar a tu vecino

LIBERTAD 3: Mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie

Para que un programa sea considerado software libre debe cumplir estas 4 libertades

COPYLEFT: Restricción que se añade al software libre que impide que alguien distribuya copias o modificaciones restringiendo las 4 libertades

El software que no es libre se llama “software privativo” o “software propietario”

¿QUÉ NO ES SOFTWARE LIBRE?

- **Software de dominio público:** aquel que no tiene copyright (derecho de autor)
 - Si el código fuente también fuera de dominio público sería un tipo de software libre, pero la mayoría de software libre si tiene derechos de autor
- **Software semilibre:** aquel que proporciona las mismas libertades que el software libre, siempre y cuando sea usado sin ánimo de lucro
- **Freeware o software gratuito:** aquellos programas que se pueden descargar gratis de internet son gratis pero no libres, no dan acceso al “código fuente” por lo que no pueden modificarse.
- **Shareware:** programas que el autor permite utilizar pero condicionando su uso, ej: que exige que se pague por su uso cuando pasa un determinado tiempo
- **Software con fuentes:** con algunos programas se entrega el código fuente para demostrar que no hace nada malicioso, pero eso no los hace libres, si su licencia no permite su modificación

VENTAJAS DEL SOFTWARE LIBRE EN ADMINISTRACIÓN DE SISTEMAS

1. Libertad de uso, estudio, redistribución y modificación
2. Sigue un formato estándar, facilitan la interoperabilidad
3. Interdependencia tecnológica, no nos atamos a ningún proveedor en particular
4. Seguridad jurídica: no hay secretismo tecnológico
5. Fiabilidad y rendimiento
6. Métodos simples y unificados de gestión de software
7. Inmensa variedad de soluciones muy maduras
8. Demanda de técnicos FLOSS en expansión
9. Aspectos económicos

DESVENTAJAS

1. Necesidad de una formación especializada
2. Algunas herramientas incorporan interfaces visuales con más funcionalidad privativas
3. No siempre hay soporte para todo tipo de hardware
4. En otros sectores hay mayor mercado laboral con otros sistemas privativos

USO SE SOFTWARE LIBRE EN ADMINISTRACIÓN DE SISTEMAS

- El mercado suele medirse por unidades vendidas o beneficios
- Difícil de evaluar para el caso del FLOSS (los sistemas libres son a menudo obtenidos sin coste)
- Muchas veces se instalan en máquinas que no fueron compradas con software libre precargado
- El método que se usa suele ser mediante acceso a máquinas públicamente accesibles
 - Este método no completa las máquinas no accesibles públicamente

INTERFACES GRÁFICAS DE USUARIO

Muchas distribuciones traen GUIs o herramientas visuales propias que:

- Son útiles y facilitan las tareas
- Suelen ser propietarias
- Nos hacen dependientes de una distribución en concreto
- A veces poseen oscuros detalles en la forma de gestionar los recursos

Estudiamos siempre la tecnología y métodos subyacentes que suelen ser comunes a todas las distribuciones

La configuración manual es siempre mejor

¿CÓMO PROMOVER EL SOFTWARE LIBRE?

- La gratuitad no es el punto fuerte del software libre
 - Insistir en esta gratuitad supone minusvalorar el resto de ventajas (esto es injusto para la gente que lo crea y lo mantiene?)
 - No se debe comenzar hablando de dinero
- No hablar del FLOSS en abstracto (Linuzix es mejor)
- No hay que ser impactante, dejar que el software libre crezca con los clientes introduciendo mejoras de forma progresiva

GNU/LINUX

Linux es un SO libre que surgió como reimplementación de UNIX (siendo la alternativa a UNIX más popular)

- Sigue el estándar POSIX
- Tiene código libre GPL -> se puede conocer, modificar y extender
- Gratuito

CRONOLOGIA

- Agosto 1991: el estudiante finlandés Linus Torvalds presenta en internet la versión 0.01 del kernel de un nuevo SO inspirado en MINIX (*aunque sin código de MINIX*)
 - Esta versión tenía más de 10.000 líneas de código
- 1992: Linux se libera bajo la licencia GPL

- 1994: Linux alcanza la versión 1.0
- 2003: versión 2.6, con casi 6 millones de líneas de código
- 2008: Nace Android basado en el kernel de Linux
- 2019: versión 5.0

DISTRIBUCIONES

Colección de software que forma un SO basado en el kernel Linux, incluye:

- El kernel Linux
- Las aplicaciones GNU
- Software de terceros, libre o propietarios

DEBIAN

- Distribución libre, sin fines comerciales
- Tiene un gran nº de aplicaciones
- Tiene un potente formato de empaquetado (paquetes deb y herramienta apt)
- Pionera en instalación y cambio de versiones a través de red
- Tiene tres ramas:
 - *Stable*: destinada a entornos de producción
 - *Testing*: es un software más nuevo que está en fase de prueba
 - *Unstable*: en fase de desarrollo

UBUNTU

- Enfocada a ordenadores de escritorio
- Concentra su objetivo en la usabilidad, lanzamientos regulares y facilidad en instalación (basada en debian)
- Financiada por el empresario Mark Shuttleworth

RED HAT

- Introduce el formato de empaquetado *rpm*
- Orientado en exclusiva al mercado corporativo, solamente a suscriptores de pago (aunque la última distribución personal ha sido cedida a la comunidad)
- Última versión: Red Hat Enterprise Linux (mayo de 2019)

CENTOS

- OBJETIVO: Ofrecer al usuario un software de “clase empresarial” gratuito
- Robusto, estable y fácil de instalar y utilizar
- Recomendado para servidores

ARCH

- No tiene herramientas de configuración automática pero una vez instalada se puede mantener y administrar el sistema de forma sencilla
- Como herramienta para administrarlo se apoya en su gestor de paquetes, llamado pacman

SUPERUSUARIO DENTRO DEL SISTEMA

DE USUARIO A SUPERUSUARIO

ADMINISTRADOR O SUPERUSUARIO: Usuario que tiene todos los privilegios sobre cualquier fichero, instrucción u orden del sistema

- En GNU/Linux es usuario es **root**
 - Desde el directorio HOME: /root
 - Si estamos usando otro usuario y queremos convertirnos en administrador:
 - Salir de la sesión y entrar usando root como nombre de usuario
 - Utilizar el comando **su ->** que nos pedirá la contraseña de root y abrirá la shell donde tenemos guardados los privilegios de administración

SUDO: Permite a otros usuarios ejecutar órdenes como si fuesen el administrados

- **/etc/sudoers:** fichero de configuración
- **Visudo:** orden para modificar el fichero de configuración
- **Sudo orden:** pide contraseña del usuario

COMUNICACIÓN CON EL RESTO DE USUARIOS

- **write:** enviar un mensaje a un usuario
- **talk:** conversar con un usuario
- **mesg [y/n]:** habilitar/deshabilitar la llegada de mensajes al terminal.
- **wall:** mandar un mensaje a todos los usuarios del sistema.
- **Fichero /etc/motd:** contiene el mensaje del día que se imprime justo después de entrar al sistema (en modo texto).
 - Fichero \$HOME/.hushlogin ⇒ permite evitar el mensaje del día.
- **Fichero /etc/issue:** contiene el mensaje que se muestra antes del login, normalmente muestra la versión de Linux (en modo texto)

TEMA 2

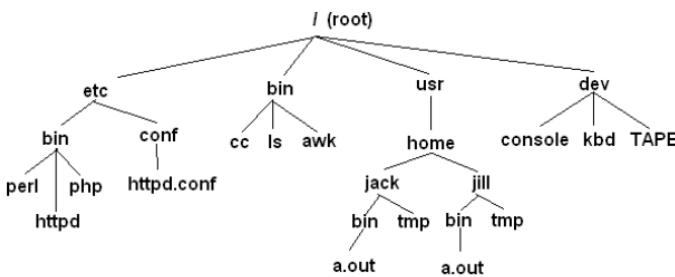
ORGANIZACIÓN DE UN S.O. TIPO GNU/LINUX

2º Ingeniería Informática UCO
PROGRAMACIÓN Y ADMINISTRACIÓN DE SISTEMAS

FICHEROS

SISTEMAS DE FICHEROS

- **En GNU/Linux, todo son ficheros:**
 - Los programas u órdenes: /bin/ls, /usr/bin/find...
 - Los dispositivos I/O son ficheros: /dev/sda, /dev/fd0, /dev/tty0...
 - Comunicación entre procesos: sockets o tuberías (pipes)
 - Directorios, ficheros de datos...
 - El núcleo del SO (kernel)
 -
- **GNU/Linux posee una estructura jerárquica de directorios, conocida como sistema de archivos**
- Sistema de ficheros: Guarda los ficheros del sistema, organizándose de manera jerárquica en directorios (no hay unidades)



NODOS-I: Metadatos sobre los ficheros que nos proporcionan información sobre aspectos como su tamaño, sus permisos, la posición de sus sectores, etc.

- A nivel lógico, el sistema de ficheros parece un árbol, pero, en realidad, se almacenan desorganizados en el disco duro
- Un fichero puede tener sectores a lo largo del disco duro
- Cada fichero tiene un nodo-i

PROPIETARIOS Y PERMISOS

GESTIÓN DEL ACCESO: propietarios y ficheros

PROPIETARIOS

- Cada fichero tiene 2 propietarios:

- Usuario
- Grupo

- **chown:** Cambia de usuario propietario (necesarios permisos root)

```
1 chown pagutierrez fichero
2 chown pagutierrez.profesores fichero
3 chown -R pagutierrez directorio
```

- **chgrp:** Cambia el grupo propietario (necesarios permisos root o ser el propietario del fichero)

```
1 chgrp profesores fichero
2 chgrp -R profesores directorio
```

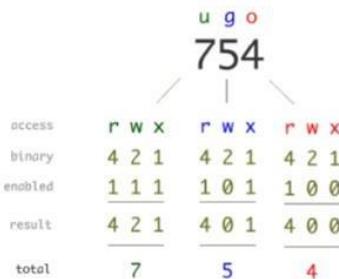
ACCESO A LOS FICHEROS: se gestiona de la siguiente forma:

Acceso	Fichero	Directorio
r	Ver el contenido	Listar el contenido
w	Modificar el contenido	Crear/eliminar ficheros
x	Ejecutar el fichero	Entrar en el directorio

```

1 pedroa@pagutierrezlaptop:~/tmp$ ls -la
2 total 36
3 drwxrwxr-x 4 pedroa pedroa 4096 feb 17 17:52 .
4 drwxr-xr-x 84 pedroa pedroa 20480 feb 17 18:14 ..
5 -rw-r--r-- 1 pedroa pedroa 0 feb 17 14:22 fichero1
6 -rw-r--r-- 1 pedroa pedroa 0 feb 17 14:21 fichero2

```



Se establecen independientemente para el usuario propietario(u), usuarios de grupo propietario (g) y el resto de los usuarios (o)

· PERMISOS ESPECIALES

· t → sticky bit (chmod o+t fichero)

- ls lo representa como una t en el 9º bit (t->o+x // T->o-x)
- En ejecutables, mantiene la imagen del fichero en la memoria de intercambio tras finalizar su ejecución
- Para directorios, solo root puede borrar o renombrar el fichero

· s para usuarios (suid, chmod u+s fichero)

- ls lo representa como una s en el 3º bit (s->u+x // S->u-x)
- Para ejecutables, el usuario efectivo del proceso es el propietario del fichero, y no el usuario que lo ejecutó

· s para grupos (sgid, chmod g+s fichero)

- ls lo representa como una s en el 6º bit (s->g+x // S->g-x)
- Para ejecutables, el grupo efectivo del proceso es el grupo propietario del fichero, y no del usuario que lo ejecutó

Actual	chmod	Resultado	Descripción
rwx-----	a+x	rwx--x---	Agregar a todos (all) permiso de escritura.
rwx--x--x	go-x	rwx-----	Se elimina permiso de ejecución para grupo y otros.
rwxr-xr-x	u-x, go-r	rwx---x--	Al usuario se le quita ejecución, al grupo y a otros se les quita lectura.
rwxrwxrwx	u-x, go-rwx	rw-----	Al usuario se le elimina ejecución, al grupo y a otros se les eliminan todos los permisos.
r-----	a+r, u+w	rw-r--r--	A todos se les agrega lectura, al usuario se le agrega escritura.
rw-r----	u-rw, g+w, o+x	---rw---	Al usuario se le eliminan lectura y escritura, al grupo se le agrega escritura y a otros se les agrega ejecución.

MÁSCARA DE PERMISOS (UMASK)

- Cuando un fichero se crea, se le asignan permisos
- Estos permisos se deciden aplicando una máscara de permisos base
- La máscara de bits indica con un 1 aquellos bits que deben ser 0 en la cadena de permisos, es decir, indica qué permisos están restringidos
- Permisos base para directorios: 777
- Permisos base para ficheros: 606

TIPOS DE FICHEROS

Según comando ls-l

- *Normal*
- *Directorio(d)*: 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
- *Socket (s)*: Similar a los pipes, pero con comunicación en ambos sentidos (dúplex)
- *Enlace físico*
- *Enlace simbólico*

· ENLACES

- Archivos especiales que permiten que varios nombres se asocien a un único archivo e idéntico.
- Ayuda a asegurar la coherencia y ahorrar espacio en el disco
 - Permite que un grupo de personas trabajen sobre un mismo fichero

Enlaces físicos:

- (In archivo-real enlace-físico): Representan un nombre alternativo para un archivo
- Si eliminamos un enlace físico, no se elimina el archivo original
 - Mientras quede al menos 1 enlace físico, el archivo no se elimina
 - Solo es posible entre ficheros que estén en la misma partición

Enlaces simbólicos:

- (In -s archivo-real enlace-simb): Puntero virtual al archivo real
 - Es un fichero de texto que contiene la ruta del archivo al que apunta
 - Si se elimina el enlace simbólico, no se elimina el fichero original

PROCESOS

Programas en ejecución.

ATRIBUTOS DE UN PROCESO (ps-FI)

- **PID**: Identificador del proceso
- **PPID**: Identificador del proceso padre
- **Nice number**: Nº de prioridad al ejecutarlo
- **TTY**: Terminal donde se está ejecutando
- **RUID**: Identificador del usuario que lo ejecutó
- **EUID**: Identificador del usuario efectivo, el que lo usa
- **RGID**: Identificador del grupo que lo ejecutó
- **EGID**: Identificador del grupo efectivo, el que lo usa

TIPOS DE PROCESOS:

- **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 se encuentran a la espera de un evento

DISPOSITIVOS

TIPOS

- **Ficheros especiales de caracteres:** Representan a dispositivos de caracteres
 - Cinta magnética, puerto paralelo...
- **Ficheros especiales de bloques:** Representan a dispositivos de bloques
 - Disquete, partición de un disco duro, pendrive...

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

- **/dev/fd0:** Disquete de la primera disquetera
- **/dev/sda:** 1º disco duro
- **/dev/sda1:** Primera partición del 1º disco duro
- **/dev/sdb:** 2º disco duro
- **/dev/sdc:** Disco USB
- **/dev/tty1:** 1º terminal de consola
- **/dev/lp0:** 1º puerto paralelo

ESTRUCTURA GENÉRICA DEL SISTEMA DE FICHEROS

En Linux, disponible como página de manual: ‘man hier’

TIPOS DE DISTINCIIONES

- **Estáticos:** Ficheros que no cambian sin intervención del administrador, como binarios o bibliotecas. Pueden estar en dispositivos de solo lectura, y no necesitan copias de seguridad
- **Dinámicos:** Contiene ficheros que pueden cambiar sin la intervención del administrador. Deben encontrarse en dispositivos de lectura-escritura, y deben hacerse copias de seguridad a menudo
- **Compartibles:** Contiene ficheros que se pueden encontrar en un ordenador y utilizarse en otro
- **No compartibles:** Contiene ficheros que no podemos utilizar en distintas máquinas

• LUGARES DEL SISTEMA DE FICHEROS

- **/bin:** Ficheros ejecutables básicos compartidos
- **/dev:** Ficheros especiales de dispositivos
- **/etc:** Casi todos los ficheros de configuración locales del sistema
- **/root:** Directorio HOME del administrador
- **/sbin:** Ficheros ejecutables que sólo el administrador puede ejecutar
- **/home:** Directorios de trabajo de los usuarios
- **/lost+found:** Contiene “referencias” a los ficheros marcados como erróneos

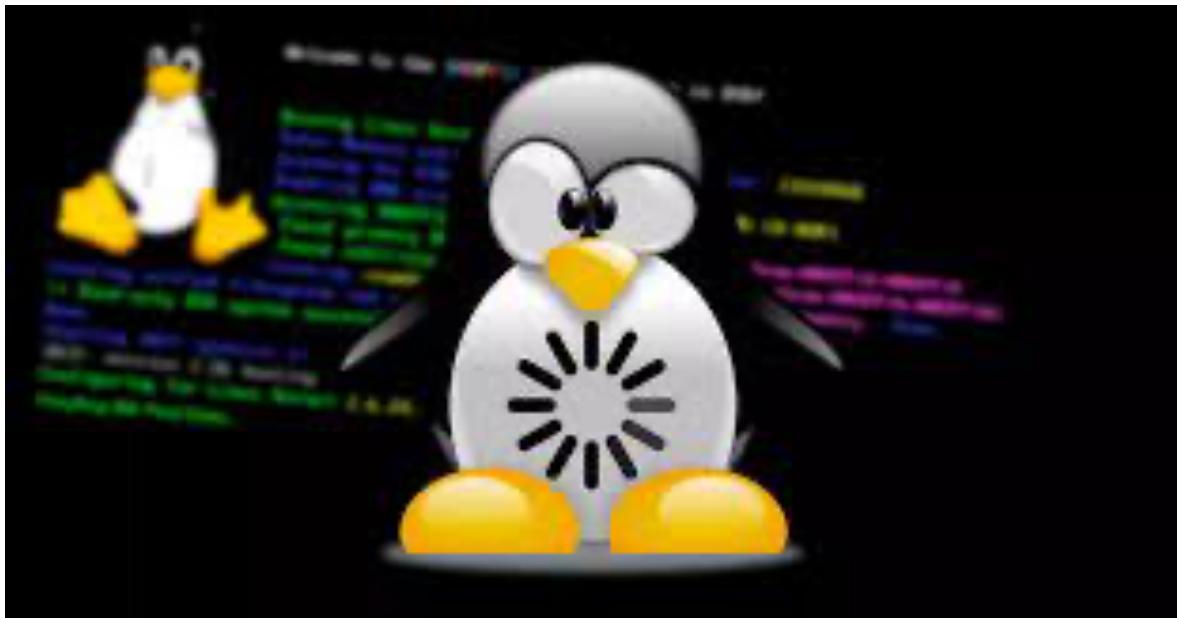
- */lib*: Librerías para ejecutar los archivos
- */proc* y */sys*: Sistemas de ficheros virtuales que contiene información sobre procesos, núcleos...
- */tmp*: Ficheros temporales
- */var*: Ficheros variables, como colas de datos, ficheros de log...
- */boot*: Núcleo y ficheros necesarios para cargar dicho núcleo, además de ficheros de configuración del gestor de arranque
- */mnt*, */mount*, */media*: Montaje de otros sistemas de ficheros
- */opt*: Paquetes de aplicaciones estáticas
- */usr*: Contiene subdirectorios de sólo lectura, que no deben ser específicos de la máquina que los usa
 - */usr/bin*: Ficheros ejecutables por todos los usuarios
 - */usr/sbin*: Ficheros ejecutables de administración
 - */usr/include*: Ficheros de cabecera estándar para compilar
 - */usr/lib*: Librería binarias
 - */usr/share*: Datos compartidos
 - */usr/src*: Código fuente

TIPOS DE DISTINCIones

- **Estáticos**: */bin*, */sbin*, */opt*, */boot*...
- **Dinámicos**: */var*, */home*...
- **Compartibles**: */usr*, */opt*...
- **No compatibles**: */etc*, */boot*, */var*

TEMA 3

ARRANQUE Y PARADA DEL SISTEMA



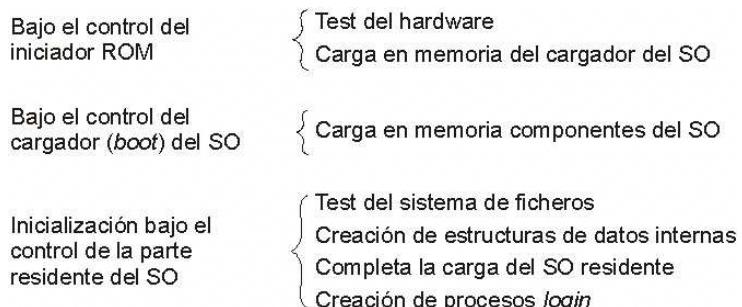
INTRODUCCIÓN

- Cuando se lleva a cabo el **arranque** el sistema se prepara para ser usado por los usuarios
- Con la **parada** el sistema se deja consistente
- El administrador debe saber que ficheros controlan estos procesos y cómo lo hacen
- Los procesos sencillos se basan en un conjunto de ficheros de configuración y de guiones Shell que determinan y controlan los procesos

PROCESO DE ARRANQUE DEL SISTEMA

Tiene dos fases:

1. Arranque del hardware
2. Arranque del SO:

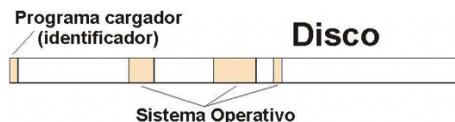


INICIADOR ROM

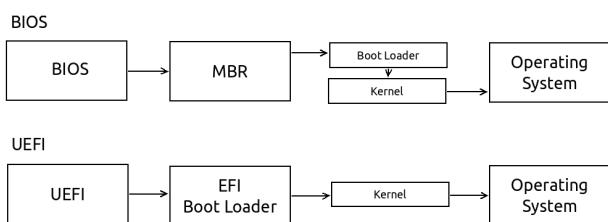
- Al arrancar el ordenador se activa una señal llamada **RESET** que inicializa todos los registros a valores por defecto
- Se carga la dirección de inicio del ROM
- La memoria ROM también contiene la **BIOS** (*software de configuración del hardware*)
- Es un programa independiente del SO, tiene 3 funciones:
 - Comprueba el sistema (*detecta sus características y comprueba su funcionamiento*)
 - Lee y almacena en memoria el programa cargador del SO
 - Pasa el control al cargador del SO, saltando a la dirección de memoria donde lo ha almacenado

PROGRAMA CARGADOR (MASTER BOOT PROGRAM O BOOT PROGRAM)

- Está en los 1º sectores del disco (llamados MBR -> Master Boot Record o Volume Boot Record)
- Tiene un tamaño prefijado
- Encargado de cargar el núcleo del SO y pasarle el control
- ROM y el SO tienen un acuerdo sobre el programa cargador relacionado con la ubicación, dirección de arranque y tamaño



PROCESO DE ARRANQUE: BIOS VS UEFI



NÚCLEO DEL SO

- Este continúa con el proceso de arranque:
 - Realiza una comprobación hardware del sistema
 - Se prepara para ejecutar el sistema
 - Crea el proceso init y le pasa el control
- Es cargado inicialmente en memoria y permanece durante el funcionamiento del sistema
- Parte del código se encuentra en **módulos del núcleo**:
 - Minimizar la cantidad de código que se carga en memoria
 - Maximizar la modularidad

para ver los módulos del núcleo activos en Linux: lsmod

INITRD

- Se necesitan módulos específicos alojados en initrd por si el arranque implica que el medio desde el que se carga el núcleo provenga de un sistema de ficheros en concreto.
- El programa cargador le dice al núcleo la posición de initrd
- FUNCIONAMIENTO:**
 - El núcleo carga primero el initrd
 - Utilizando el initrd se cargan los módulos necesarios
 - El núcleo continua el proceso de arranque
- El proceso init termina el proceso de arranque dejando el sistema en modo multiusuario (preparado para ser usado)
- Utiliza scripts que le indican las acciones a realizar, entre las que tenemos:
 - Chequear los sistemas de ficheros
 - Montar los sistemas de ficheros permanentes
 - Activar las áreas de swapping o intercambio
 - Activar los demonios y la red
 - Limpiar los sistemas de ficheros borrando los directorios temporales
 - Habilitar el login a los usuarios del sistema

PROGRAMA CARGADOR. GESTOR DE ARRANQUE GRUB (*GRAND UNIFIER BOOTLOADER*)

- Se instala en MBR y hace las funciones de MBP
- Pregunta qué SO arrancar
 - Si la respuesta es **Linux** -> carga el núcleo solicitado y le pasa el control para que el arranque continúe
 - Si la respuesta es **Windows** -> pasa el control a Windows que realiza su arranque
- Archivo fundamental de configuración : **/boot/grub/grub.cfg** (se genera a partir del comando *sudo update-grub2*)

- **CARPETA etc/grub.d/**
 - **etc/grub.d/00_header:** cabeceras
 - **etc/grub.d/05_debian_theme:** aspect visual del menú, colores, temas...
 - **etc/grub.d/10_Linux:** contiene scripts que se encargan del kernel de Linux
 - **etc/grub.d/20_***: aplicaciones third party
 - **etc/grub.d/30_os-prober:** contiene scripts que se encargan de otros SO
 - **etc/grub.d/30_uefi-firmware:** contiene comandos que automatizan la extracción de configuraciones incluidas en la partición EFI
- **FICHERO /etc/default/grub:** (es editable)
 - **GRUB_DEFAULT:** Entrada por defecto para el arranque (si ponemos saved será seleccionada por el administrador)
 - **Grub-set-default:** permanente
 - **Grub-reboot:** un solo arranque
 - **00_header:** leer el contenido del fichero
 - **GRUB_SAVEDEFAULT:** La entrada por defecto es siempre la última seleccionada
 - **GRUB_HIDDEN_TIMEOUT=0:** Muestra una pantalla en negro o con una imagen durante un nº de seg indicados
 - No se usa con múltiples sistemas
 - Es 0 cuando solo hay Linux
 - **GRUB_HIDDEN_TIMEOUT_QUIET=true:** sin cuenta atrás
 - **GRUB_TIMEOUT=10:** Nº de seg hasta seleccionar entrada por defecto
 - **GRUB_CMDLINE_LINUX="opciones":** pasar opciones de arranque al kernel Linux (modo normal o recuperación)
 - **GRUB_CMDLINE_LINUX_DEFAULT="quiet splash":** pasar opciones de arranque al kernel Linux (modo normal)
 - **GRUB_TERMINAL:** Desactivar modo gráfico
 - **GRUB_BADRAM="0x7DDF0000,0xfffffc000":** deshabilitar el uso de algunas direcciones de memoria (1º argumento: dirección a la que se aplica, 2º: máscara con un 1 en los bits que voy a considerar)
 - **GRUB DISTRIBUTOR=`lsb release -i -s 2> /dev/null || echo Debian`:** obtener el nombre de la distribución
 - **GRUB_DISABLE_LINUX_UUID="true":** no utilizar el UUID del dispositivo raíz (utilizar nomenclatura tradicional /dev/sda).
 - **GRUBGFXMODE=640x480:** seleccionar manualmente la resolución para el menú
 - **GRUB_INIT_TUNE="480 440 1":** hacer beep antes del menú de inicio.
 - **GRUB_BACKGROUND:** imagen de fondo.

FUNCIONES DEL GRUB

1. Editar las entradas
 - a. Si pulsamos la tecla e podemos modificar las entradas de arranque (*los cambios no son permanentes, solo son para probar*)
2. Ejecutar comandos para arreglar el arranque (pulsar tecla c)
3. Numerar los dispositivos según los reconozca la BIOS
 - a. **Nombres de dispositivos:** (<t><n>,<np>) (hd0,0) ⇒ /dev/sda1
 - b. **Nombres de ficheros** (hd0,0)/boot/grub/grub.conf

MODO MONOUSUARIO/ MULTIUUSUARIO

MODO MONOUSUARIO

- Estado del sistema para realizar tareas administrativas y de mantenimiento y que requieren un control completo y no compartido
- Solo realiza el **montaje del sistema de ficheros raíz** (SF RAIZ)
- Se puede acceder a todo el sistema pero muy pocos demonios están en ejecución y muchas utilidades no están activas
- Para entrar en modo monousuario el proceso **init** crea el Shell por defecto (*/bin/sh*) como usuario **root**
 - Pero antes ejecuta la orden */sbin/sulogin* que pide la contraseña del root para dejar entrar al sistema
- Se puede entrar a este modo de dos maneras:
 - **Manualmente**: indicando al cargador una opción o parámetro (opción single a la entrada del núcleo)
 - **Automáticamente**: si hay problemas de arranque que el sistema no puede solucionar

¡PROBLEMA!: Si cambiamos las opciones de GRUB y ponemos `init=/bin/sh`, no se llama `sulogin`, para esto no hay una solución única pero si hay un conjunto de medidas de mitigación para proteger al sistema de esto (ej: solicitar contraseña para la entrada de administración)

MODO MULTIUUSUARIO

PASOS DEL PROCESO DE ARRANQUE

1. Chequea el sistema de ficheros raíz con **fsck**
 - a. Si al apagar el sistema, el sistema de ficheros se desmontó correctamente, no se chequea
 - b. Aunque hay algunos SO que fuerzan el chequeo siempre o cada cierto tiempo
 - c. Si `fsck` encuentra problemas que no puede solucionar solo lleva el sistema a modo monousuario para que el administrador realice el chequeo manual
2. Monta el sistema de ficheros raíz en modo lectura-escritura
3. Chequeo el resto de SFs con `fsck`
4. Monta el resto de SFs
5. Activa las particiones de intercambio (*swapping*): **swapon -a**
6. Activa las cuotas de disco: **quotacheck -a** y **quotaon -a**
7. Lanza los procesos servidores o demonios: **cond,atd,syslog...**
8. Activa la red
9. Lanza los demonios de red: **xinetd,sshd...**
10. Limpia los sistemas de ficheros
11. Permite que los usuarios entren:
 - a. Crea las terminales, lanzando **getty**
 - b. Borra en caso de que exista el fichero **/etc/nologin** (si este fichero existe los usuarios, excepto el root, no pueden entrar al sistema)

NIVELES DE EJECUCIÓN EN GNU/LINUX

- **NIVEL 0**: Apagado
- **NIVEL 1,s o S**: Modo monousuario, rescue o troubleshooting

- **NIVEL 2:** Modo multiusuario sin funciones de red
- **NIVEL 3:** Modo multiusuario con funciones de red y terminales de texto
- **MODO 4:** Sin usar, a redefinir por el administrador
- **MODO 5:** Modo multiusuario con funciones de red con funciones de red e inicio de sesión gráfico
- **MODO 6:** Sistema reiniciándose

/sbin/runlevel : saber en que nivel está el sistema

/sbin/telinit: cambiar de nivel de ejecución (*ej: telinit 1 -> a modo monousuario*)

El nivel por defecto establecido al arrancar se encuentra en el fichero **/etc/inittab** o con el comando **systemctl set-default multi-user.target**

Al arrancar mediante GRUB, al núcleo se le puede pasar un nº indicando el nivel en el que queremos arrancar

EQUIVALENCIAS INIT Y SYSTEMD

Mapping between init run levels and systemd targets

Run level	Target	Description
0	poweroff.target	System halt
emergency	emergency.target	Bare-bones shell for system recovery
1, s, single	rescue.target	Single-user mode
2	multi-user.target ^a	Multiuser mode (command line)
3	multi-user.target ^a	Multiuser mode with networking
4	multi-user.target ^a	Not normally used by init
5	graphical.target	Multiuser mode with networking and GUI
6	reboot.target	System reboot

a. By default, **multi-user.target** maps to **runlevel3.target**, multiuser mode with networking.

FICHEROS DE INICIALIZACIÓN

Para personalizar niveles de ejecución: carpetas **/etc/rc.?**

- ? Es el nivel de ejecución
- Se ejecutan al arrancar o cambiar de nivel
 - El nombre del script empieza por S o K + 2 dígitos+ nombre descriptivo
 - Los ejecuta en orden alfabético, primero los K, después los S
 - **FICHEROS K:** Detener demonios o matar procesos
 - **FICHEROS S:** Lanzar demonios o ejecutar funciones de inicio
- Carpetas **/etc/rc?.d/**:
 - Todos los ficheros son enlaces al fichero con el mismo nombre localizado en **/etc/init.d**
 - Los scripts reciben varios parámetros
 - Esto permite lanzar demonios sin reiniciar el sistema
 - **Rc:** ejecuta los ficheros K con el parámetro stop y los S con start

SYSTEMD

Es un reemplazo del proceso init (pero con más funciones, esto lo hace más complejo causando algunas dependencias innecesarias)

FILOSOFÍA DE SYSTEMD:

- Mejorar la forma de expresar dependencias
- Permitir que se realicen más tareas en paralelo
- Reducir la carga extra que supone el intérprete

- Se gestiona mediante **unidades** (servicios) y **targets**
- Es compatible con los scripts SysV pero incorpora su propio mecanismo de gestión de servicios
- En la carpeta /etc/systemd cada servicio es un fichero .conf
- La forma de describir dependencias es más flexible
 - AFTER: especifica qué necesita el servicio para poder ejecutarse
 - WANTEDBY: especifica cuando se lanzará el servicio
- Se puede especificar el usuario con **user** y el ejecutable con **ExecStart**
- **Restart=always** hace que el servicio se re-ejecute por si se para.
- **RestartSec=1**: se reinicia tras 1 seg
- **StartLimitIntervalSec=0[service]**. Se intenta el reinicio para siempre

PARADA DEL SISTEMA

En ocasiones es necesario apagar o reiniciar el sistema

Acciones durante proceso de entrada

1. Se notifica a los usuarios
2. Se envía una señal de terminación (**TERM**) a los procesos en ejecución
3. Se paran los demonios
4. A los usuarios que queden conectados se les echa del sistema
5. Se envía una señal de fin (**KILL**) a los procesos que queden en ejecución
6. Actualizaciones del disco pendientes con **sync**

shutdown [opciones] tiempo [mensaje]:

- Sin opciones: modo monousuario (telinit 1).
- -r: reiniciar (telinit 6)
- -h: parar (telinit 0).
- -c: cancelar.
- -k:haceruna simulacióndeapagado.
- tiempo: +minutos, now, horas:minutos.

Al salir del modo monousuario vuelve al nivel por defecto

CAÍDAS DEL SISTEMA Y PROBLEMAS DE ARRANQUE

Posibles caídas del sistema:

- Fallos hardware
- Fallos de luz
- Problemas ambientales
- Problemas de entrada/salida
- Problemas de algún sistema de ficheros
- Errores en la configuración del sistema
- No se puede leer el sistema de ficheros
- Áreas en el disco dañadas

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

Orden **dmesg**: mensajes producidos durante el arranque

Durante el arranque al núcleo se le pueden pasar otros parámetros:

- **Root=partición**: indicar que monte como partición una raíz distinta
- **Init=ejecutable**: que en vez del proceso init lance otro procesos
- **Single**: arrancar en modo monosuaurio
- Un nº indicando el nivel de arranque

TEMA 4

GESTIÓN DE USUARIOS

GERENCIA DE PERSONAS

2º Ingeniería Informática UCO
PROGRAMACIÓN Y ADMINISTRACIÓN DE SISTEMAS

USUARIOS

DEFINICIÓN DE USUARIO:

Persona que trabaja en el sistema: editando ficheros, ejecutando programas...

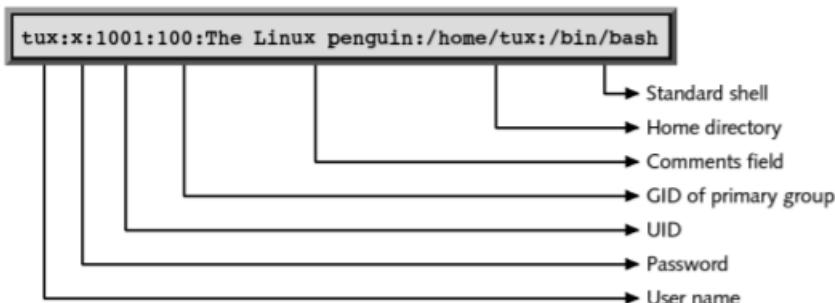
- **Pseudo-usuario:** entidad que sin ser una persona puede ejecutar programas o poseer ficheros

CARACTERÍSTICAS DE USUARIO:

- **nombre** → Nombre del usuario, logname o username.
- **password** → contraseña cifrada o:
 - “~” o “*” o “!!” → la cuenta esta desactivada o bloqueada.
 - “x” → las shadow están activas, la contraseña cifrada se guarda en ~ /etc/shadow.
- **uid** → identificador del usuario.
- **gid** → identificador del grupo primario al que pertenece.
- **gecos** → campo de información referente al usuario (nombre, teléfono, ...)
- **home** → Path del directorio \$HOME del usuario.
- **shell** → Interprete de órdenes.

- El propietario del fichero es **root** y el grupo **root**.
- Los permisos del fichero son **rw-r--r--**.
- El programa **/usr/sbin/vipw** permite editar el fichero manualmente.
- El programa **pwck** verifica la integridad de **/etc/passwd** y **/etc/shadow**.
- Se permite el acceso al fichero **/etc/passwd** en modo lectura para poder leer información del usuario, pero no se debería permitir acceso a las **passwords** (aunque estén cifradas).

Fichero /etc/passwd



CONTRASEÑAS

Passwd <nombre_usuario> ⇒ asignar contraseña a un usuario (o cambiarla). ~

Para la elección de una contraseña adecuada:

- Introducir 2 o más caracteres extras, símbolos especiales...
- Escribir mal las palabras.
- Utilizar mayúsculas y minúsculas, pero no de forma evidente.
- Concatenar, embeber o mezclar 2 o más palabras.
- Usar caracteres poco comunes: \$, &, #...

La contraseña se debe cambiar cuando se sospecha que alguien la ha podido averiguar, un usuario/administrador se marcha del trabajo, etc.

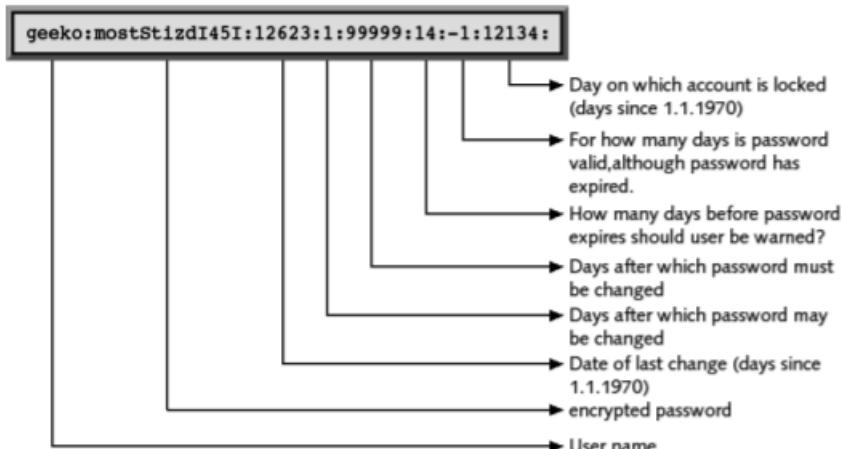
Periódicamente se debe forzar a que los usuarios cambien sus contraseñas, incluido el administrador

Shadow passwords

Las contraseñas cifradas se guardan en /etc/shadow, que tienen permisos rw-----, 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.

Es solo para aquellos usuarios que tengan una "x" en /etc/passwd. Por defecto, están activas y se actualizan automáticamente.



nom:pass:changed:minlife:maxlife:warn:inactive:expired:unused

- **nombre** → Nombre del usuario, logname o username.
- **pass** → contraseña cifrada o:
 - mkpasswd --method=sha-512 contraseña salt
- Comandos de actualización:
 - pwconv ⇒ crear y actualizar el fichero /etc/shadow. •
 - pwunconv ⇒ desactivar los shadow passwords.

Para cifrar una contraseña, se utilizan algoritmos criptográficos de generación (función hash, H(.))

1. El mensaje es la contraseña (C)
2. **Salt (S)** es una palabra aleatoria concatenada a los bytes de contraseña → dificulta ataques con diccionarios y tablas de hash precomputadas
3. El sistema concatena C con S {C,S} calcula el resumen F=H({C,S}) y almacena S y F.
4. Cuando el usuario introduce una contraseña C', se repite todo el proceso;
 $F'=H(\{C',S\})$
5. Si $F=F'$, entonces el usuario puede entrar al sistema

ALGORITMOS:

- **MD5 (message-digest algorithm 5):** Aplica funciones no lineales a los 17 segmentos de 32 bits de un bloque de 512 bits. Se obtiene un resumen de 128 bits

Obtener suma MD5:

```
md5sum Fichero.ext > Fichero.md5
```

Chequear suma MD5:

```
md5sum -c Fichero.md5
```

- **SHA (Secure hash algorithm):** genera resúmenes más grandes, que lo hacen más seguro ante diferentes tipos de ataques. Se pueden considerar 160, 224, 256 o 512 bits para el resumen (el de 512 es el +usado)

FICHERO/ETC/SHADOW

Busca introducir restricciones de tiempo o envejecimiento para la validez de la cuenta/contraseña

- **Changed:** fecha del último cambio de contraseña
- **Minlife:** nº de días que tienen que pasar para poder cambiar la contraseña
- **Maxlife:** nº días máximo que puede estar con la misma contraseña cambiarla
- **Warm:** cuántos días antes de que la contraseña expire
- **Inactive:** nºdías después de la contraseña expire, período donde se desactivará la cuenta
- **Expired:** fecha en la que la cuenta expira y se deshabilita

chage → para la realización de cambios

Opción -M	Número de días que puede estar con la misma contraseña
Opción -W	Establece un aviso de que la contraseña expira un número de días antes de que ocurra
Opción -I	Nº de días para cambiar la contraseña una vez haya expirado la anterior, la cuenta se deshabilitará si la contraseña no ha sido cambiada
Opción -E	Fecha en la que la cuenta expira y se deshabilita de forma automática

Ficheros de inicialización

Son scripts Shell que realizan tareas como dar valor a variables, nombrar, alias...

Se ejecutan al hacer un login en el sistema por SSH o por terminal real	.bash_profile en bash .profile en bash y sh .login en csh
Cada vez que se ejecuta un shell, aunque no conlleve login	.bashrc en bash .cshrc en csh
Al salir del sistema el usuario	.bash_logout en bash .logout en C csh

SECCIÓN DE INTÉPRETE DE ÓRDENES

- En el último campo del fichero **/etc/passwd**, se establece el intérprete de órdenes que se ejecuta al entrar al sistema
- En **/etc/shells** se indican los shells permitidos
- Para el cambio de Shell se utiliza el comando de chsh:
- Si un usuario no tiene asignado ningún intérprete de órdenes se usará el Shell por defecto (/bin/sh.)
- Para privar a un usuario a la entrada al sistema se la asigna /bin/false o /bin/nologin/.

CUENTAS RESTRICTIVAS

Permiten limitar las acciones de los usuarios del sistema.

- ❖ Asignar como Shell un fichero ejecutable que realice una tarea determinada, y al terminar sale al sistema. Las tareas que pueden realizar quedan determinadas por nivel de identificador del usuario
- ❖ Usando el Shell restrictivo **/bin/rbash/**, para crear dichas limitaciones, hay que copiar los ficheros que puedan ejecutar en un directorio y que su PATH sea solo ese directorio.

AÑADIR UN USUARIO AL SISTEMA

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

HERRAMIENTAS PARA CREAR/MODIFICAR LAS CUENTAS DE USUARIO:

- **adduser o useradd** ⇒ crear cuentas de usuario, o modificar cuentas ya existentes.
- **usermod** ⇒ modificar cuentas.
- **deluser o userdel** ⇒ eliminar cuentas.
- **newusers** ⇒ crea cuentas de usuarios utilizando la información introducida en un fichero de texto (en batch), que ha de tener el formato del fichero **/etc/passwd** (no copia los ficheros de inicialización).
- **users-admin** ⇒ herramienta en modo grafico

GRUPOS

GRUPOS: colecciones de usuarios que comparten recursos o ficheros del sistema

Al crear un fichero se establece como grupo propietario el grupo activo del usuario en ese momento, el grupo activo sería el primario. Al determinar los permisos sobre el fichero, se usan todos los grupos de usuarios:

- PRIMARIOS: grupo especificado en /etc/passwd.
- SECUNDARIOS: otros grupos, indicados en /etc/group.

`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 no pertenece al grupo, pero este tiene contraseña, se le solicita y pasa a ser 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.

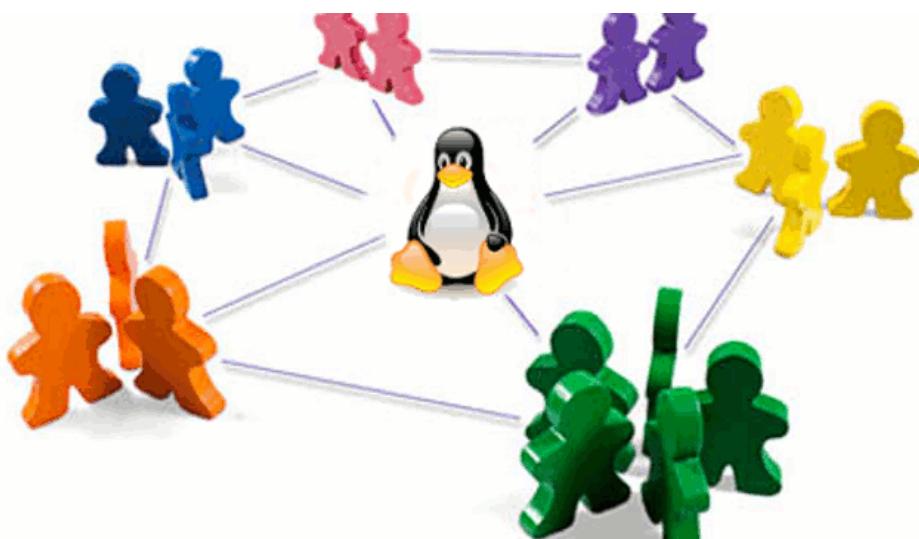
`grpck` ⇒ chequea la consistencia del fichero de grupos

USUARIOS ESTÁNDAR

- ➔ **Root:** cuenta de administrador.
- ➔ **bin:** (utilidades comunes de usuarios), **daemon**(ejecución de demonios)... son los tradicionalmente usados para poseer ficheros y ejecutar servicios
- ➔ **mail, news, ftp:** asociados con herramientas o facilidades
- ➔ **postgres, mysql, xfs:** creados por herramientas instaladas en el sistema para administrar y ejecutar sus servicios
- ➔ **nobody o nfsbody:** usado por NFS y otras utilidades, usuario sin privilegios

TEMA 5

GESTIÓN DE LOS RECURSOS DEL SISTEMA



INTRODUCCIÓN

Una correcta administración del sistema implica obtener información sobre sus recursos y rendimiento:

- Procesos en ejecución
- Cantidad de memoria disponible
- Espacio en disco ...

ACTIVIDADES DE LA CPU

PROCESOS EN GNU/LINUX

Un **proceso** representa un programa en ejecución, es una abstracción a través de la cual la memoria, tiempo de procesador y recursos E/S pueden gestionarse.

- Un sistema de tiempo compartido (GNU/Linux) permite a múltiples usuarios que ejecuten múltiples procesos (*aunque la CPU solo puede ejecutar uno a la vez, cambiando de uno a otro muy rápido*)
- El SO es el encargado de decidir en qué orden se ejecutan los procesos

MODOS DE EJECUCIÓN

- **MODO NÚCLEO:** Se ejecuta funciones del núcleo (*kernel ejecutándose en nombre del proceso*)
 - Llamadas al sistema: los procesos de usuario solicitan servicios a través de la interfaz de llamadas al sistema
 - Excepciones: situaciones excepcionales causan excepciones hardware que requieren intervención del kernel
 - Interrupciones: los dispositivos periféricos interrumpen para notificar al kernel de diversos sucesos
- **MODOS USUARIOS:** Se ejecuta código normal del programa

TIPOS DE PROCESOS

PROCESOS DE USUARIO: Creados por un usuario real y ejecutados en modo usuario

PROCESOS DEMONIO:

- No asociados a un usuario, o asociados a uno ficticio
- Se ejecutan en modo usuario
- Realizan tareas de administración del sistema

PROCESOS NÚCLEO:

- No asociados a un usuario, corresponden al código del kernel
- Se ejecutan en modo núcleo
- Tareas de administración más delicadas

MONOTORIZAR CON PS

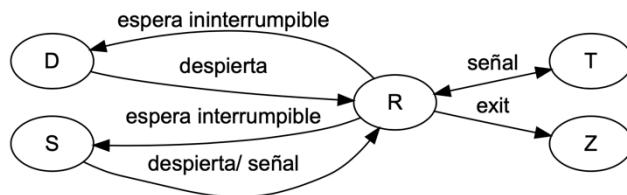
ps información sobre los procesos en ejecución

- USER -> usuario que lanzó el programa
- PID -> identificador del proceso
- PPID -> identificador del proceso padre
- %CPU -> Porcentaje de la CPU consumido por el proceso
- %MEM -> Fracción de memoria consumida
- VSZ -> Tamaño virtual (código+datos+pila) en KB

- RSS -> memoria real usada en KB
- TTY-> terminal asociado con el proceso
- STAT -> estado del proceso

R: en ejecución	N: prioridad baja (> 0)	L: tiene páginas bloqueadas en memoria
S: durmiendo	<:prioridad alta (< 0)	s: líder de sesión
T: parado (señal o trace)		l: tiene multithread
Z: proceso zombie		+: proceso foreground
D: durmiendo ininterrumpible (E/S)		

ESTADOS DE LOS PROCESOS



- **R: En ejecución**, está listo para ejecutarse en cuanto la CPU esté libre
- **S: Durmiendo**: esperando a que ocurra un evento específico
 - Bash y los demonios del sistema pasan casi todo su tiempo durmiendo
 - Estos procesos no recibirán tiempo de CPU hasta que el evento ocurra o se reciba una señal específica
- **D: Durmiendo, espera ininterrumpible**:
 - El proceso no maneja señales, solo despertará cuando pase el evento
 - Normalmente es transitorio y no llegaríamos a verlo en el ps
 - Solo podemos reiniciar o arreglar el problema
- **Z: Zombie** : el proceso termina correctamente pero el padre no recoge su código de error (*consultar el PPID para ver el origen del problema*)
- **T: Detenido temporalmente** mediante señales (Ctrl+Z) o porque está siendo examinado (trace)
 - Solo volverán a ejecutarse tras otra señal
- **I: Ilde**: significa que estamos ante un proceso ocioso de un hilo del núcleo
 - A diferencia del estado D, solo se aplica a procesos del núcleo y no contribuye a la carga de la CPU
- **s: Líder de sesión**: los procesos se pueden agrupar, de todos estos el líder es el que interactúa con la terminal
- **I: hilos creados**: con CLONE_THREAD
- **L**: El proceso ha pedido al kernel bloquear determinadas páginas de memoria, para evitar que no se modifiquen mientras se hacen determinadas operaciones
- **+**: **foreground**: proceso de 1º plano, indicando sin &

PRIORIDAD Y SEÑALES

- **Número nice y prioridad de procesos**:
 - Planificación de procesos por prioridades dinámicas
 - Al lanzar el proceso se le asigna un nº nice o prioridad estática (*se hereda por defecto del proceso padre*)
 - La prioridad depende del nº nice:

- Valores bajos (-): más prioridad
 - Valores altos (+): menos prioridad
- Rango de prioridad estática: [-20,19]
- Asignación de prioridades mayores o menores que la actual:
 - *nice -5 nautilus*: incrementa el nº nice en 5
 - *nice --10 nautilus*: lanzar nautilus con nº nice decrementado 10
 - *renice 14 890*: prioridad 14 al proceso 890
 - *renice 5 -u pedroa*: prioridad 5 para todos los procesos del usuario pedroa
 - Envío de señales a los procesos:
 - *kill -señal pid*: donde señal es un nº
 - *kill pid*: mandar señal por defecto del proceso pid
 - *SIGKILL(9)*: fuerza la salida al proceso
 - *SIGSTOP(19)*: parar un proceso
 - *SIGCONT(18)*: Reiniciarlo
 - *Killall comando*: permite mandar una señal a todos los procesos con un determinado nombre de comando
 - *Pkill o skill*: enviar un señal usando el nombre u otros atributos
 - Los procesos en estado D o Z no se detienen aunque reciban la señal KILL

TSP se puede bloquear y capturar y STOP no (pero en ambos los procesos detenidos se pueden reanudar con la señal CONT o el comando fg)

#	Nombre	Descripción	Por defecto	¿Se puede capturar?	¿Se puede bloquear?	¿core dump?
1	HUP	Hang up (terminal)	Terminar	Si	Si	No
2	INT	Interrumpir (Ctrl+C)	Terminar	Si	Si	No
3	QUIT	Similar a TERM	Terminar	Si	Si	Si
9	KILL	Matar proceso	Terminar	No	No	No
*	BUS	Error manejo bus	Terminar	Si	Si	Si
11	SEGV	Violación de segmento	Terminar	Si	Si	Si
15	TERM	Parar software	Terminar	Si	Si	No
*	STOP	Parada	Parar	No	No	No
*	TSTP	Parada (Ctrl+Z)	Parar	Si	Si	No
*	CONT	Continuar (tras STOP)	Continuar	Si	No	No
*	WINCH	Cambio tamaño	Continuar	Si	Si	No
*	USR1	A definir	Terminar	Si	Si	No
*	USR2	A definir	Terminar	Si	Si	No

*: depende del Sistema Operativo.

- KILL(9): No se puede bloquear ni capturar
- INT(2): La que se envía al pulsar Ctrl+C (se puede bloquear)
- TSTP: La que se envía al pulsar Ctrl+Z
- TERM (15): La que se manda al cerrar el proceso padre o al reiniciar (se puede bloquear y capturar)
 - QUIT(3): similar a TERM pero hace un core dump
- HUP (1):
 - Si se trata de **demonios**: provoca que se reinic peace
 - Si se trata de **procesos** iniciados en una terminal: se manda al cerrar la terminal

MONITORIZAR USO CPU

- **uptime**: hora actual, cuanto tº lleva el sistema en marcha, nº de usuarios conectados y carga media del sistema
 - valores altos -> el sistema se está usando mucho (aunque valores bajos no significan que el tiempo de respuesta vaya a ser bajo)
- **pstree**: visualiza un árbol de los procesos en ejecución
- **top**: proporciona una visión de la actividad de procesador en tiempo real (mostrando aquellas tareas que hacen más uso de la CPU)
 - Las 5 primeras líneas muestran información general:
 - Estadísticas uptime
 - Resumen de procesos en el sistema
 - Porcentaje de tiempo de CPU gastado

- Estado actual de la memoria física
 - Espacio swap
- Los datos de la parte inferior son similares a los de ps excepto:
 - SHR: memoria compartida disponible para ser utilizada
- Procesos ordenados decrecientemente por uso de CPU
- Lista actualizada normalmente cada 5 segundos
- Tareas sobre los procesos:
 - “r”: cambiar la prioridad
 - “K”: matar o enviar una señal
 - “M”: Ordenar según memoria
 - “n”: cambiar el nº de procesos que se muestran
 - “q”: para salir
 - “u”: mostrar un usuario
 - “R”: Cambiar ordenación
 - “l”: información independiente por cada procesador
- **htop**: similar pero con colores
- **vmstat**: información sobre memoria virtual
 - r -> nº de procesos esperando su tiempo de ejecución
 - b -> nº de procesos en espera ininterrumpible
 - us -> tº de CPU en modo usuario
 - sy -> tº de CPU en modo sistema
 - id -> tº de CPU en inactividad
 - wa -> tº de CPU usado en espera de E/S
 - st -> tº de CPU usando virtualización
- **ps y top**: Leen información que necesitan de /proc
- Cada proceso tiene una carpeta (nombre-> pid) 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**: variables de entorno en el momento de la 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 la memoria

PROGRAMAR EJECUCIÓN DE PROCESOS

- **at**: ejecutar tareas a una determinada hora
 - puede recibir un fichero de texto con las órdenes a ejecutar
 - **atd**: demonio que ejecuta las órdenes
 - **atq**: consulta la lista de órdenes
 - **atrm**: eliminar órdenes
- **cron**: ejecutar tareas periódicamente
 - **crond**: demonio encargado de ejecutar órdenes
 - **crontab**: establecer las tareas a ejecutar (formato: minuto hora día_mes mes día_semana [user] comando)
 - -e: añadir/modificar
 - -l: listar

- -r: eliminar
 - */etc/crontab*: fichero de configuración del administrador
 - */etc/cron.d*: directorio en el que el administrador puede copiar ficheros con formato del crontab que ejecutará cron
- Si la máquina no está encendida cuando se quiere lanzar el proceso cron no lo lanza
- **anacron**: no asume que la máquina está siempre encendida
 - permite especificar tareas diarias e introducir aplicaciones o enlaces a las mismas en:
 - */etc/cron.daily/*
 - */etc/cron.hourly/*
 - */etc/cron.monthly/*
 - */etc/cron.weekly/*

RASTREO DE PROCESOS (RASTREO DE SEÑALES Y LLAMADAS AL SISTEMA)

STRACE: Nos permite observar lo que está haciendo un proceso

- Muestra cada llamada al sistema que hace y que señal que recibe
 - *strace -p pid*: rastrear un proceso ya iniciado
 - *strace comando*: iniciar un proceso y rastrearlo
 - *strace -o salida.txt comando*: utilizar un fichero para guardar la salida

Procesos acaparadores:

- Procesos que acaparan mucha CPU
- Antes de matarlos debemos saber que están haciendo
 - Si el proceso parece legítimo (verdadero) deberíamos:
 - 1. Suspenderlo con STOP
 - 2. Aplicarle renice
 - 3. Reanudarlo con CONT

MEMORIA

CONTROL/GESTIÓN DE LA ACTIVIDAD DE LA MEMORIA

Debemos gestionar la RAM y la zona de intercambio

- **Vmstat (en KBs):**
 - **swpd**: cantidad de memoria virtual ocupada
 - **free**: cantidad de memoria virtual sin usar
 - **buff**: cantidad de memoria empleada como buffers para E/S
 - **cache**: Cantidad de memoria empleada como caché de disco
 - **si**: cantidad de memoria traída del espacio de intercambio desde disco
 - **so**: cantidad de memoria intercambiada al disco
 - **bi**: bloques recibidos desde un dispositivo de bloques (en bloques/s)
 - **bo**: bloques enviados a un dispositivo en bloques (en bloques/s)
 - **in**: nº de interrupciones por segundo
 - **cs**: nº de cambios de contexto por segundo
- Espacio para paginación:
 - El espacio recomendado para la paginación depende de la memoria requerida por los procesos la demanda del sistema...
 - Se puede controlar con nº de prioridad en */etc/fstab*
 - *Swapon -s*: listado de particiones o ficheros activos

- ***Swapon /dev/sdd1***: activar una determinada partición
- ***Swapoff /dev/sdd1***: desactivar una determinada partición
- ***free***: obtener información sobre el uso de memoria

DISPOSITIVOS ENTRADA/SALIDA

- Espacio en disco:
 - ***df***: muestra la capacidad, el espacio libre y el punto de montaje
 - “***-i***”: nos permite mostrar información sobre los nodos ***-i***
 - ***du***: muestra el espacio usado por cada subdirectorio del directorio actual
 - cuenta bloques del sistema estén o no completamente ocupados
 - Para un fichero de 1B cuenta 4KB
 - si no ponemos ***--max-depth=1*** nos muestra todas las carpetas

CONTROL DE DISPOSITIVOS DE ENTRADA/SALIDA

- ***iostat intervalo numero***: presenta estadísticas sobre la CPU y los dispositivos y particiones de E/S
 - ***tps*** ⇒ no de transferencias por segundo.
 - ***kB read/s*** ⇒ no de kBs leídos por segundo.
 - ***kB wrtn/s*** ⇒ no de kBs escritos por segundo.
 - ***kB read*** ⇒ no total de kBs leídos.
 - ***kB wrtn*** ⇒ no total de kBs escritos.

TEMA 6

ORGANIZACIÓN DE SISTEMAS DE FICHEROS Y DISCOS

**2º Ingeniería Informática UCO
PROGRAMACIÓN Y ADMINISTRACIÓN DE SISTEMAS**

INTRODUCCIÓN

La función principal de un disco duro es almacenar la información del PC cuando no se encuentra conectado a la corriente eléctrica. también puede servir de extensión para la memoria RAM, gracias al mecanismo de memoria virtual (intercambio)



Discos rígidos vs SSD

Funcionan de forma parecida a un tocadiscos, mientras que los discos SSD utilizan una memoria formada por semiconductores para almacenar la información.

- PLATO: Cada uno de los discos que se encuentran apilados en su interior, cubiertos de un material magnetizable (de aluminio o cristal). La escritura cambia el estado de este material.
- CABEZAL: es un brazo que se mueve sobre el plato. Como los discos giran, permite acceder a cualquier punto de los mismos
- PISTA: Se trata de cada una de las líneas esféricas que se pueden formar sobre cada plato
- CILINDRO: Conjunto de varias pistas que se encuentran una encima de otra
- SECTOR: Cada una de las divisiones que se hace de la circunferencia que se forma en el disco

ARCHIVOS

Unidad de almacenamiento lógico no volátil que agrupa un conjunto de información relacionada entre si bajo un mismo nombre. Para acceder a un archivo puede ser secuencial (para acceder a una posición hay que conocer la anterior) o directo/aleatorio (se puede acceder a cualquier posición)

SISTEMAS DE ARCHIVOS

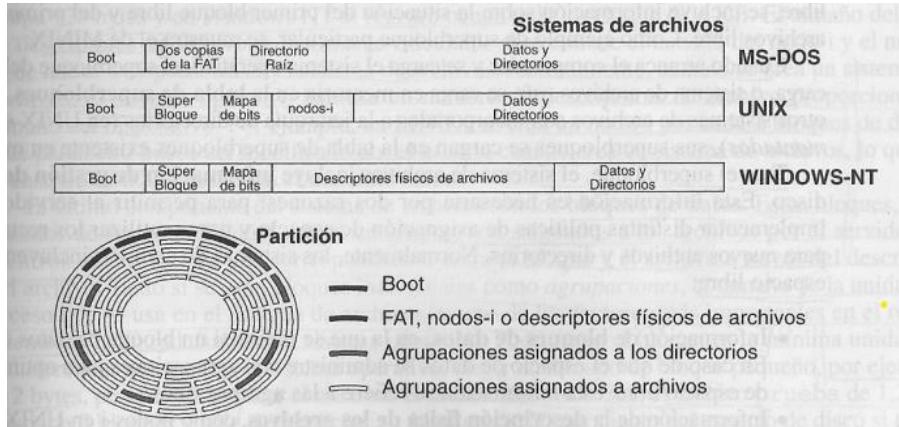
Partición: porción de un disco a la que se le dota de una identidad propia y que se manipula como un entidad lógica independiente

Bloque: agrupación lógica de sectores físicos del disco, la cual supone la unidad de transferencia mínima que usa el SA.

El tamaño de bloque es un parámetro decisivo que afecta a la eficiencia del acceso a disco y a la fragmentación del mismo. (*Cuanto más pequeño, mayor nº de operaciones de E/S pero menor fragmentación; en el otro caso al revés*)

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

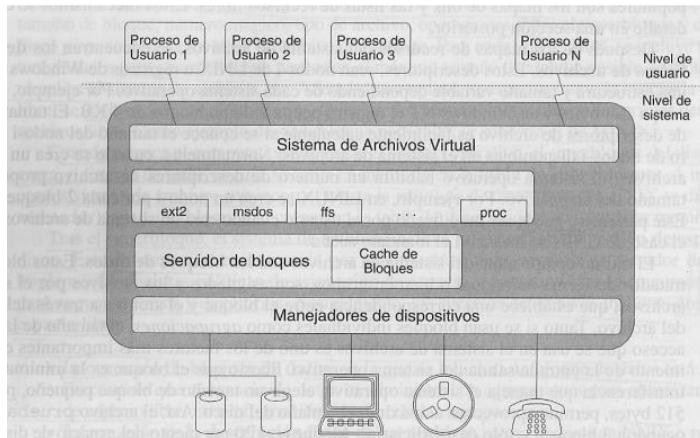
ESTRUCTURA DEL SISTEMA DE ARCHIVOS



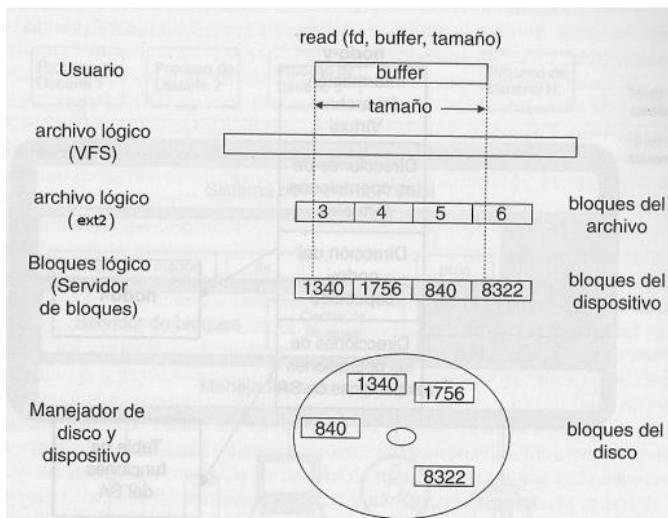
- **EL BLOQUE DE CARGA** (*boot* o *Volume Boot Record*) contiene código ejecutado al arrancar el ordenador por el iniciador ROM utilizando esa partición. Se suele incluir en todas las particiones (aunque no contengan el SO) para así mantener una estructura uniforme. Se añade un número mágico, el cuál será comprobado por el iniciador ROM para demostrar que el bloque de carga es válido.
- **METAINFORMACIÓN:** Describe el SA y la distribución de sus componentes. Es necesaria para poder acceder a los datos.
- **SUPERBLOQUE:** Se mantiene una serie de información común para todos los SAs y una entrada característica para cada tipo de SA. Al arrancar la máquina, los superbloques de todos los SAs que son cargados se mantienen en memoria.
- **DESCRIPTORES FÍSICOS DE ARCHIVOS:** Describen cada uno de los archivos almacenados, tienen una estructura y tamaño muy dependiente del SO. El número de descriptores debe ser proporcional al tamaño no total del disco
- **GESTIÓN DEL ESPACIO LIBRE:** distintos mecanismos permiten gestionar el espacio libre. Se gestionan dos tipos de recursos:
 - **Mapas de bloques:** indican qué bloques o agrupaciones están libres.
 - **Mapas de descriptores de archivos:** indican qué descriptores de archivos están libres.
- **BLOQUES DE DATOS:** es donde se almacena realmente la información

SERVIDOR DE ARCHIVOS

Servidor de archivos: es el componente del SO que se encargará de gestionar el acceso a archivos. Se sigue una organización por capas, proporcionan servicios a los niveles superiores los niveles inferiores, y en cada nivel se aumenta la abstracción de las operaciones.



- ❖ **SISTEMAS DE ARCHIVOS VIRTUAL:** Proporciona la interfaz para las llamadas de E/S que deseen realizar los procesos de usuario, interactuando con el módulo de organización de archivos. Cumple las funciones de manejo de directorios, gestión de nombres,... para ello o, es necesario utilizar una estructura adicional que incluye las características comunes a todos los sistemas de archivos y un enlace al descriptor de archivo particular.
 - Los nodos virtuales contienen la siguiente información:
 - Atributos del archivo.
 - Puntero al nodo-i real.
 - Punteros a funciones que realizan las operaciones genéricas de cualquier SA.
 - Punteros a funciones que realizan las operaciones propias del SA concreto.
- ❖ **MÓDULO DE ORGANIZACIÓN DE ARCHIVOS** Se implementa por separado para cada tipo de SA. Relaciona la imagen lógica de un archivo con su imagen física, traduciendo direcciones lógicas del archivo a las direcciones físicas del dispositivo. Se basa en la información de los nodos-i y utiliza los servicios del servidor de bloques para realizar las operaciones correspondientes.
- ❖ **SERVIDOR DE BLOQUES:** Este nivel emite los mandatos genéricos para leer y escribir bloques en los manejadores de dispositivo. Se traducirán en llamadas al manejador específico del SA. En este nivel se realiza la cache de bloques, es decir, se almacenan datos para que las solicitudes futuras de esos datos se puedan atender con mayor rapidez.
- ❖ **MANEJADOR DE DISPOSITIVOS:** traducen ordenes de E/S de alto nivel a un formato que pueda entender el dispositivo (dependiente del hardware).



DIRECTORIOS

Es un fichero con un formato determinado. El contenido de un directorio es una serie de entradas (registros), una por cada fichero contenido en él. Cada registro tiene, al menos, el nombre del fichero y el puntero al descriptor físico correspondiente.

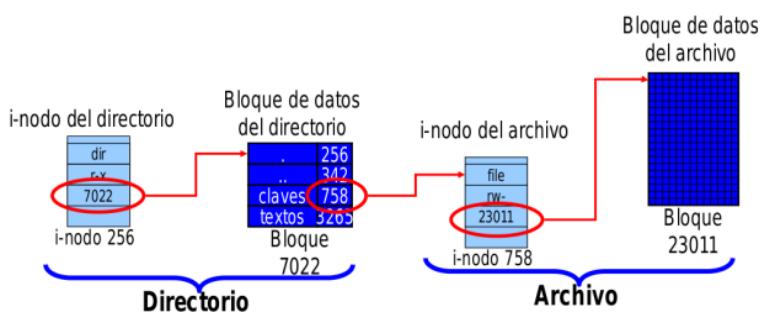
Ejemplo de uso, con la ruta /users/luis/claves se interpreta de forma recursiva:

1. Traer a memoria bloque del i-nodo 2 (i-nodo raíz, conocido).
2. Se busca dentro users y se obtiene el i-nodo 342.
3. Traer a memoria bloque del i-nodo 342.
4. Se busca dentro luis y se obtiene el i-nodo 256.
5. Traer a memoria bloque del i-nodo 256.
6. Se busca dentro claves y se obtiene el i-nodo 758.
7. Al leer el i-nodo 758, se detecta que es un fichero y ya se tienen donde están los datos del archivo
8. Leer los bloques del fichero.

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 Y MECANISMOS DE ASIGNACIÓN

Asignación: cómo se hace la correspondencia entre los bloques físicos del disco y los bloques lógicos del archivo. Hay dos tipos de mecanismos de asignación:

- ✓ **Asignación de bloques contiguos:** actualmente está en desuso. Todos los bloques de archivos se encuentran contiguos en el disco.

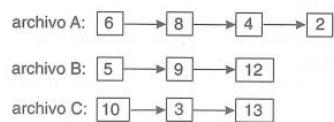
- ✓ **Asignación de bloques no contiguos:** Los bloques del archivo se encuentran en cualquier posición del disco. Para tener constancia de que bloques no contiguos pertenecen a cada archivo, se utilizan listas enlazadas o índices.

LISTA ENLAZADA

Cada bloque tiene un apuntador al siguiente bloque que seguiría en el archivo. El descriptor del archivo solo debe incluir la referencia al primer bloque

- ✓ **Tabla de asignación de archivos:** Es una variación del método lista enlazada. Los apuntadores se almacenan en una tabla independiente de los Bloques. La tabla se aloja en cache para mejorar las prestaciones y se mantiene una copia doble en el disco para mayor fiabilidad.

FAT															
x	x	EOF	13	2	9	8	FREE	4	12	3	FREE	EOF	EOF	FREE	BAD
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15



ÍNDICES

Los punteros a los bloques están juntos y contiguos en una localización concreta, *bloques índice*. Cada archivo tiene un bloque índice.

- ✓ **Índice multinivel:** consiste en introducir n niveles de apuntadores, de manera que los apuntadores del descriptor apuntan a otros.

SOLUCIÓN UNIX → ESQUEMA HÍBRIDO

Bloques de datos o bloques índice. En ext4 y en NTFS existen los extents (bloques índice especiales que marcan una zona contigua del disco “numeroBloqueInicial, numeroBloques”). Por cada nodo-i hay que incluir:

- Punteros directos a los 10 primeros bloques
- Puntero a un bloque índice de primer nivel
- Puntero a bloque índice de 2º nivel

GESTIÓN DEL ESPACIO LIBRE

Se necesita para asignar espacio a los archivos nuevos o a los que se les desea añadir datos. Se mantienen mapas de recursos, implementados como **mapas de bits** o listas de recursos libres, con esto se incluye un bit por recurso que será 1 si el recurso esta libre y 0 en caso contrario

- ❖ **Lista de recursos libres:** se mantiene una lista de apuntadores a los recursos libres, acto seguido si se ocupa el recurso es borrado de la lista. Como solución a los discos que tienen que tener mucho espacio libre, y tienen que cargar la lista, se incluye un número de bloques consecutivos en la lista.

INCREMENTO DE LAS PRESTACIONES

El acceso a memoria se realiza en el orden de nanosegundos, y el acceso en disco en el orden de milisegundos. Como almacenamiento intermedio de los datos se usa la caché cargada en Memoria Principal, se almacena toda la caché en bloques:

ALMACENAMIENTO INTERMEDIO DE LOS DATOS

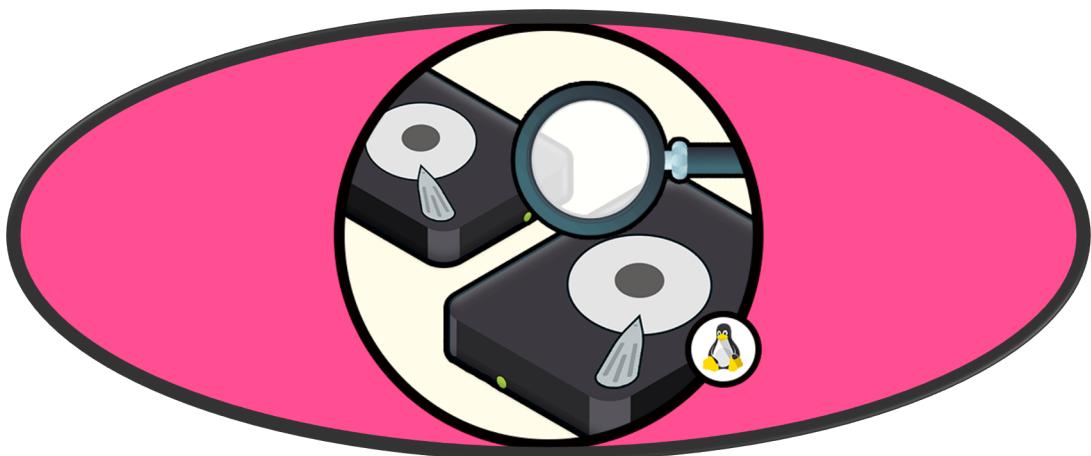
- Mantener una caché de datos en MP
- Aprovecha la proximidad espacial y temporal en las referencias a los datos accedidos
- Cache de nombres: lista con {nombre, nodo-i}. Si se vuelve a acceder al archivo, no hay que hacer toda la búsqueda del nodo-i
- Caché de bloques: colección de bloques leídos o escritos recientemente. Si se vuelve a acceder a ese bloque, no hay que cargarlo de nuevo.

CACHÉ DE BLOQUES

- Si el bloque está en MP, se escribirá o leerá en MP.
- Posteriormente, se moverán los bloques de MP al dispositivo.
- Si la cache está llena, hay que eliminar algún bloque, suelen eliminarse los bloques que llevan mucho tiempo sin usarse:
 - Políticas de reemplazo: First In First Out (FIFO), Most Recently Used (MRU), Least Recently Used (LRU)...
 - Lo más común es LRU: aprovecha que los bloques no utilizados durante mucho tiempo, posiblemente no volverán a ser utilizados.
- Bloques sucios (cambiados en cache pero no en el disco). Distintas políticas a la hora de mantener la **coherencia**:
 - ESCRITURA INMEDIATA: siempre actualizado
 - ESCRITURA DIFERIDA: actualizamos cuando el bloque salga de la caché
 - ESCRITURA PERIÓDICA: establecer un tiempo periódico para las actualizaciones, tiene un compromiso entre rendimiento y fiabilidad. Conlleva una reducción de los posibles daños por caídas.
- Se puede distinguir entre bloques **especiales** (directorios, nodos-i o bloques índice) y bloques de **datos**. Tienen una escritura inmediata.
- No se debe quitar un disco del sistema sin antes volcar los datos de la cache (comando **sync**).

TEMA 7

ADMINISTRACIÓN DE SISTEMAS DE FICHEROS Y DISCOS



INTRODUCCIÓN

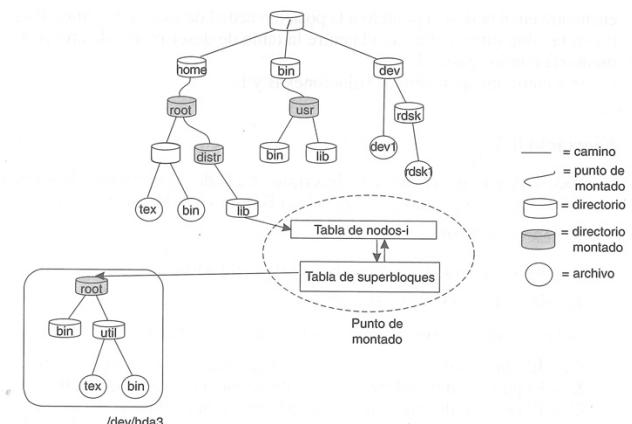
TAREAS ASOCIADAS AL SISTEMA DE FICHEROS EN GNU/LINUX

- Permitir el acceso a ficheros locales y remotos
- Controlar los recursos que proporcionan cuotas de disco, estadísticas de uso...
- Proteger los posibles fallos o errores
- Controlar y proporcionar seguridad de los datos
- Durante el arranque: chequear (y corregir) el sistema de ficheros
- Instalar y configurar nuevos dispositivos de almacenamiento

MONTAJE Y DESMONTAJE DEL SISTEMA DE FICHEROS

CONCEPTO

- En GNU/Linux solo hay un **sistema de ficheros lógico** (o una jerarquía de directorios)
 - en ella se organizan todos los dispositivos de almacenamiento disponible
- Cada partición tiene su propio sistema de ficheros (formado por un **directorio raíz** y su **jerarquía**)
 - **MONTAR UN SIST DE FICHEROS:** Añadirlo al sistema de ficheros lógico -> sus datos pasan a estar disponibles a partir de un punto de montaje
 - **DESMONTAR UN SIST DE FICHEROS:** este deja de estar disponible, dejándolo consistente
- Los ficheros principales del SO están disponibles desde la raíz del sistema de ficheros lógico (/)
- En el arranque se monta primero la partición correspondiente a la raíz (*root*) y luego cualquier partición auxiliar



HERRAMIENTA MOUNT

- **mount [opc]** <FicheroEspecialBloque> <PtoMontaje>
 - **-t tipo -sf** -> tipo de sistema de ficheros
 - **-r** -> montaje en modo sólo lectura
 - **-o opcionesMontaje** -> opciones del proceso de montaje
- **umount <PtoMontaje> (o <FicheroEspecialBloque>)** -> desmontar un sistema de ficheros
 - Si está utilizando **busy** no se podrá desmontar
- **Fuser** -> saber que ficheros se están utilizando y qué procesos los usan

- **f**: fichero abierto para lectura
- **F**: fichero abierto para escritura
- **c**: directorio de trabajo
- **e**: ejecutando el fichero
- **m**: memoria compartida
- **lsof** -> obtener un listado de todos los ficheros abiertos

FICHERO /ETC/FSTAB

Es un fichero con información sobre todos los sistemas de ficheros a montar o ya montados y las zonas de intercambio a activar

fi especial pto tipo opciones dump freq pass num

- *fi_especial* -> fichero especial de bloques
- *pto* -> directorio que sirve de punto de montaje
- *tipo* -> tipo de SF
- *dump_freq* -> frecuencia del dump para hacer una copia de seguridad de ese SF mediante el comando dump
- *pass_num* -> en tiempo de arranque, en qué orden hay que chequear los SFs

OPCIONES DEL FICHERO /ETC/FSTAB

- **rw** -> lectura-escritura
- **ro** -> solo 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=500, gid=100** -> propietario y grupo propietario de los ficheros del SF (si el SF no incorpora esta información o si se quiere cambiar)
- **umask=137** -> permisos de los ficheros (en este caso, 640)
- **dev** -> interpretar ficheros especiales en el sistema de archivos
- **sysnc** -> forzar a que todas las operaciones sean sincronas
- **user** -> permite que los usuarios puedan montar el sistema de ficheros (*solo el mismo usuario podrá desmontarlo*)
- **users** -> igual que user pero cualquiera podrá desmontarlo
- **nouser** -> solo root puede montar el SF
- **owner** -> permite que un usuario pueda montar el sistema de ficheros siempre que sea dueño del fichero de dispositivo
- **defaults** -> rw,suid...

Al montar **mount** como *root*:

- **mount /media/dvd**: coge las opciones que faltan del fichero
- **mount -t iso9660 -r /dev/dvd /media/dvd**: no las coge

Si se asigna permisos de montaje a los usuarios solo pueden ejecutar **mount /media/dvd** (sin opciones)

Mount -a: montar todas las unidades que sean **auto**

Udev y dbus: automontado de unidades

COMPROBACIÓN DEL SISTEMA DE FICHEROS

CONCEPTO Y HERRAMIENTAS DE CHEQUEO

- Durante el arranque, fsck o e2fsck chequean el estado del sistema de ficheros, detectando e intentando repararlos
- Se actúa sobre la estructura (no sobre el contenido):
 - Bloques que pertenezcan a varios ficheros
 - Bloques marcados como libres pero que se encuentran en uso
 - O la situación contraria, que estén marcados en uso y estén libres
 - Inconsistencias en cuanto al nº de enlaces hacia el nodo-i
 - Nodos-i marcados como libres, pero que están en uso
 - La situación contraria
- Para chequear un SF siempre debe estar desmontado o montado solo en modo lectura (al igual que el SF raíz)
- El SF raíz no se puede desmontar ya que si al arrancar el proceso de chequeo encuentra problemas que no puede solucionar, obliga al administrados a que realice el chequeo a mano ejecutando fsck o e2fsck (modo monousuario)

SISTEMA DE JOURNALING

JOURNALING: Para evitar la verificación completa de SF de gran tamaño (es muy costosa)

- Se implementa un modelo de control transaccional basado en **logging**
 - Las suboperaciones que modifiquen los metadatos y datos de un mismo archivo se agrupan en la transacción
 - Si el sistema falla las acciones realizadas se deshacen o completan (recorriendo el log)
 - No es seguro que el sistema esté actualizado al finalizar la recuperación

SISTEMAS con esta filosofía: JFS (IBM), NTFS, ext3

- Por cada sub-operación que altera las estructuras de disco se escribe un registro en el **log**, que incluye las modificaciones en los buffers de i-nodos y de bloques.
- Cuando se ha copiado a disco (log) el registro de *commit*, se empiezan a procesar realmente los buffers.
- Despues de una caída:
 - Se completan las transacciones committed
 - Se descartan el resto de transacciones

CREACIÓN DEL SISTEMA DE FICHEROS

PASOS NECESARIO PARA AÑADIR UN NUEVO DISCO O SF

1. Realizar la conexión física
2. Crear un fichero especial de dispositivo (*si es necesario*)
3. Crear las particiones: **fdisk** (o parted)
4. Crear sistema de ficheros: **mke2fs -t ext2 /dev/sdb3**
5. Etiquetar la partición usando **e2label** -> asigna una etiqueta al SF que se puede usar en el fichero */etc/fstab*, en el campo *fi_especial*, mediante *LABEL=etiqueta*
6. Crear el directorio que hará de punto montaje
7. Montar el nuevo sistema de ficheros
8. Actualizar **/etc/fstab** con las opciones necesarias

DIFERENCIAS ENTRE EXT2,EXT3 Y EXT4

- **Ext3** tiene el mismo formato que ext2 pero además es transaccional (*añade un registro o journal que permite recuperar la consistencia tras una caída del sistema*)
- **Ext4** tiene un formato parecido a extc3 pero incluye:
 - Una extensión describe un conjunto de bloques lógicos contiguos de un fichero que también se encuentran contiguos en disco
 - Se retrasa la reserva de bloques de disco hasta que se va a escribir en él
 - Implementa una herramienta de desfragmentación online, *e4defrag*
 - Manejo de sistemas de ficheros y ficheros de mayor tamaño

¿QUÉ SISTEMA ELEGIR?

- Ext2: muy rápido pero no tiene journaling
 - Se puede usar en un SF en el que se guardaran ficheros temporales
- Ext3: buen rendimiento y journaling
- Extc4: menor uso del CPU
 - Mayor rapidez en la lectura y escritura
 - Estándar de facto en Linux
- Tune2fs -> conocer y ajustar parámetro de un SF ext4/ext3/ext2
 - **-l dispositivo:** listar el contenido del superbloque del SF
 - **-c -max -mount -counts dispositivo:** establecer el nº max de montajes 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 es un 5%

ASPECTOS AVANZADOS

CUOTAS DE DISCO

- Permiten limitar el nº de bloques y/o ficheros que un usuario puede usar en una partición
- Hay dos tipos de límites
 - Límite hard: el usuario no puede sobrepasarlo
 - Si lo hace ya no se podrá usar más bloques o crear más ficheros
 - Límite soft: es inferior al límite hard y se puede sobrepasar durante cierto tiempo (siempre que no se alcance el límite hard)
- Periodo de gracia: tiempo durante el que se puede sobrepasar el límite soft (los límites y espacios se establecen de forma independiente para bloques y nodos-i)

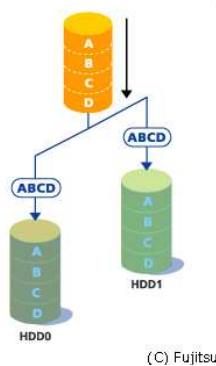
ESTABLECER CUOTAS DE DISCO

1. Instalar el paquete quota: apt-get install quota
2. Indicarlo en fstab (ext4 con cuota integrada en el journal y en ext3)
3. Remontar la partición para que se activen las opciones: mount -o remount/home
4. **Quotacheck -avugm**: añade el contenido de los ficheros de control de cuotas
 - a: todos los dispositivos con cuotas

- b. **v**: verbose
 - c. **u**: cuotas para usuarios
 - d. **g**: cuotas para grupos
 - e. **m**: no remontar los archivos en modo solo lectura
5. activar las cuotas **quotaon -avug**
 6. desactivarlas **quotaoff -avug**
 7. editar la cuota del usuario: **edquota usuario**
 8. establecer el periodo de gracia: **edquota -t**
 9. Copiar cuotas: **-up usuario1 usuario 2**
 10. Estadísticas de las cuotas: **repquota /dev/sdb1**

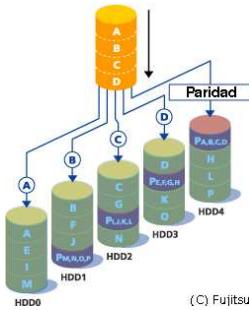
ADMINISTRACIÓN DE VOLÚMENES DINÁMICOS

- RAID: Array redundante de discos independientes
 - Varias uds de disco se ven como una sola ud lógica
 - Se pueden implementar por *software* o por *hardware*
- LVM: Logical Volume Management: agrupar las particiones en volúmenes
- RAID NIVEL 0:
 - Expande la información en diversos discos (se ven como un único SF)
 - Aumenta el espacio según el nº de discos usado
 - Se consigue E/S pasarela en lecturas y escrituras, siempre que los bloques a tratar no sean del mismo disco
 - No hay redundancia de datos
- RAID NIVEL 1:
 - Se utilizan 2 o + discos duros, que forman un único SF
 - Son discos espejos (todos guardan la misma información)
 - Hay redundancia de datos
 - Las lecturas pueden ser en paralelos, las escrituras no
 - Cuando uno de los discos falla, el sistema sigue trabajando con el otro sin problemas
 - La recuperación de un disco es transparente al usuario



(C) Fujitsu

- RAID NIVEL 4/5:
 - División de los datos a nivel de bloques
 - RAID 4: Mín 3 discos duros, de ellos 1 almacena la paridad de los otros discos, que son usados para datos
 - Problema: el disco con paridad es un cuello de botella
 - RAID 5: Repartir paridad entre todos los discos, ofrece la mejor relación rendimiento-coste
 - Se consigue un dispositivo de almacenamiento más grande
 - Hay redundancia de datos
 - Lectura y escritura en paralelo



PARIDAD

Cada vez que se escriben datos, se calcula el XOR bit a bit

- 1: nº de unos impar
- 0: nº de unos par

Disco 1:	00101010	(Datos)
Disco 2:	10001110	(Datos)
Disco 3:	11110111	(Datos)
Disco 4:	10110101	(Datos)
Disco 5:	11100110	(Paridad)

Si alguno de los discos falla (por ej el 4) el contenido se suele restaurar a partir de la paridad:

Disco 1:	00101010	(Datos)
Disco 2:	10001110	(Datos)
Disco 3:	11110111	(Datos)
Disco 4:	11100110	(Paridad)
Disco 5:	10110101	(Datos)

CONTROL DE DISPOSITIVOS DE ENTRADA/SALIDA

- La herramienta mdadm permite crear o administrar un dispositivo RAID
- Tiene distintos modos de funcionamiento:
 - create: configurar y activar sistemas RAID
- */proc/mdstat* lista todos los sistemas RAID activos con información sobre su estado
- Las particiones que formen el RAID tienen un flag RAID (de esta manera serán detectadas y activadas en el proceso de arranque)

EJEMPLO DE CREACIÓN DE UN RAID1

- Instalar el paquete mdadm
- Activar el flag RAID en la partición
- Crear el RAID (es necesario instalar el paquete mdadm)
- Crear un SF sobre el sistema RAID: mke2fs -t ext4 /dev/md1
- Añadirlo al fichero /etc/fstab para montarlo en tiempo de arranque

AÑADIR NUEVO DISCO AL RAID COMO DISCO DE REPUESTO

```
mdadm /dev/md2 -a /dev/sdc3
```

Activar el nuevo disco: mdadm --grow /dev/md2 -n 2.

A continuación, introducirlo en /etc/fstab.

- Información sobre el estado: mdadm --detail --scan /dev/md1
- Todo esto se puede configurar utilizando el fichero /etc/mdadm.conf.

TEMA 8

INSTALACIÓN DE IMPRESORAS

2º Ingeniería Informática UCO

PROGRAMACIÓN Y ADMINISTRACIÓN DE SISTEMAS

INTRODUCCIÓN

Las impresoras son mucho más complicadas que otros periféricos:

- Disponen de un SO propio, que recoge los trabajos y los imprime en papel.
- Reconocen formatos específicos y algunas son accesibles desde la red.
- En Linux **CUPS (Common Unix Printing System)**, permite realizar mucho más fáciles las tareas de administración de impresoras.
- Medidas de rendimiento:
 - **Dpi (Dots per inch):** resolución, puntos que imprime por cada pulgada.
 - Páginas por minuto: velocidad

LENGUAJE DE LAS IMPRESORAS

Un trabajo de impresión puede verse como un programa escrito en un lenguaje que la impresora entiende.

- ❖ Un lenguaje de impresión es **Page Description Languages (PDLs)** ⇒ describen como representar una página en el papel utilizando el cartucho de tinta, se usa para ello un formato vectorial.
- ❖ Es más rápido y fácil que transmitir la imagen en crudo
- ❖ Es independiente del dispositivo y de la resolución

Formato no vectorial sería a través de un mapa de bits

Se podría pasar de un fichero PDL a mapa de bits: rasterizar, para ello se usan programas que hacen Raster image processing (RIP), ejemplo *Ghostscript*.

ORGANIZACIÓN

Cada equipo puede gestionar muchas impresoras a la vez, cada impresora se le asigna un nombre y entiende un PDL (o varios); cada impresora tiene su propia cola de impresión en la que guardar y secuenciar los trabajos. Administración lanza órdenes para añadir impresoras, gestionar las tareas de impresión, etc.

DIRECTORIOS DE SPOOL:

Son los usados por las colas de impresión. Guarda un fichero con las propiedades del trabajo de impresión, guarda los trabajos pendientes para imprimir. Están contenidos en `/var/spool`

ORGANIZACIÓN CLIENTE/SERVIDOR:

En el servidor se abre un **proceso** (es un demonio que realiza la impresión). El cliente manda un fichero, que se encola copiándolo al directorio de spool, se informa al demonio.

Y luego el demonio se impresión es el encargado de que se imprima.

Se usa un **filtro de impresión**, el programa modifica el fichero a imprimir, transformándolo al PDL de la impresora.

Para convertirse en **servidor de impresión**, se tiene que dar los permisos oportunos para que la impresora pueda ser usada de forma remota.

ELEMENTOS DE CUPS

Este lenguaje se basa en el protocolo HTTP:

1. Operaciones POST para imprimir y GET para ver el estado
2. Los ficheros de configuración son muy parecidos a los de Apache
3. Las conexiones se realizan en el puerto 631
4. CUPS es una evolución del IPP o internet Printing Protocol

Cups sabe manejar una impresora:

- Gracias a los ficheros PPD (Postscript Printer Description)
 - Opciones soportadas por la impresora
 - Lenguaje que entiende de forma nativa
- Junto con los **filtros**:
 - Cadenas de conversores, basados en los tipos MIME (describen el tipo de medio del contenido)
 - Una línea en el PPD indica cual es el tipo final que necesita la impresora y el programa que utilizará para convertirlo al formato nativo

FOOMATIC: es una base de datos instalable en cualquier sistema que integra controladores de impresoras con los “spoolers” habituales en UNIX: CUPS, LPRng, LPD, etc.

Actúa como un filtro configurable, según los ficheros PPD

ADMINISTRACIÓN DE IMPRESORAS CON CUPS

➔ Imprimir un fichero:

- lp [-d impresora] fichero1 [fichero2] (System V).
- lpr [-P impresora] fichero1 [fichero2] (Berkeley).

➔ Eliminar un trabajo de la cola de impresión:

- cancel id tra1 [id tra2] [impresora] (System V).
- lprm [-P impresora] id tra1 [id tra2] (Berkeley).

➔ Eliminar una impresora (o clase):

- ladmin -x impresora

➔ Consultar la cola de impresión:

- lpad lpq -P impresora ⇒ listado de la cola impresión y del estado de los trabajos
- Introduce la idea de clase de impresoras: conjunto de impresoras que actúa como una sola, de manera que el trabajo se manda a la primera que hay libre.

➔ Crear clases de impresoras:

- ladmin -p HP -Color -c ClasePrueba

➔ Crear una instancia de impresora con opciones concretas:

- loptions -p HP -Color/2up -o number -up=2
- lpr -P HP -Color/2up tmp.ps

Consultar el listado de impresoras soportadas: lpinfo -m

Añadir una impresora: **con la orden ladmin**

```
# -E: habilita impresora; -v URI; -m fichero.ppd
pagutierrez@PEDROLaptop:~$ ladmin -p groucho -E -v parallel:/dev/lp0 -m pwlcolor
.ppd
pagutierrez@PEDROLaptop:~$ ladmin -p fezmo -E -v socket://192.168.0.12 -m
laserjet.ppd
```

HABILITAR/DESHABILITAR IMPRESORAS:

- **cupsdisable impresora** → Deshabilita la impresora (se aceptan trabajos en la cola, pero no los imprime).
- **cupsenable impresora** → Iniciar de nuevo la impresora (imprimirá los trabajos pendientes y los que reciba nuevos).
- **cupsreject impresora** → Deshabilita la cola de impresión (no aceptará nuevos trabajos).
- **cupsaccept impresora** → Habilitara la cola de impresión (que aceptará de nuevo trabajos).

El demonio de impresión es cupsd (necesario para imprimir)

- /etc/init.d/cups ⇒ script para lanzar el demonio.

FICHEROS DE CUPS:

Ficheros de configuración:

- **/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.

COMPARTIR IMPRESORAS CON CUPS:

Desde el punto de vista de CUPS, la impresión por red no es muy distinta de la impresión local. Editamos el fichero (**cumpds.conf**), para que acepte trabajos desde la red

```
1 <Location />
2   Order Deny,Allow
3   Deny From All
4   Allow From 127.0.0.1
5   Allow From IP
6 </Location>
```

Si así lo deseamos, podemos decir a CUPS que publique en broadcast las impresoras disponibles.

Desde la interfaz web de configuración, se pueden compartir fácilmente las impresoras, para que otros equipos las usen.

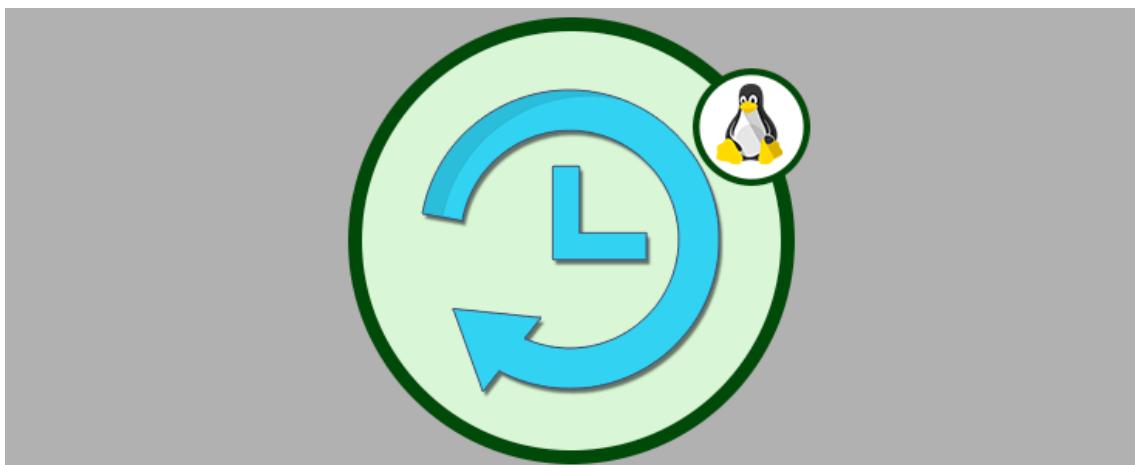
En Windows, bastara con introducir la dirección web correspondiente (` http://192.168.117.1:631/printers/LaserJet-1200), y nos instalara la impresora (puede requerir drivers específicos).

También se puede utilizar ` cups-lpd, que es un interfaz entre el antiguo LPD y CUPS (LPD es soportado directamente por Windows como puerto).

Otra opción es instalar SAMBA y compartir la impresora utilizando el protocolo ` CIFS (protocolo de ficheros compartidos de Microsoft).

TEMA 9

COPIAS DE SEGURIDAD Y RESTAURACIÓN



PLANES DE PREVENCIÓN DE CATÁSTROFES

En algún momento algunos archivos serán ilegibles, las copias de seguridad dependen de la situación y es necesario determinar de que archivos hacer copia, donde, como...

- El **administrador** debe:
 - Planear e implementar un sistema de copias de seguridad
 - Hacer copias periódicamente de los ficheros
 - Guardar las copias de seguridad en un lugar seguro
- La estrategia de copias de seguridad tiene que ser efectiva para conseguir seguridad
- Tener en cuenta:
 - La capacidad de restaurar el sistema en un tiempo aceptable
 - El tiempo que tarda en hacerse una copia de seguridad
 - La facilidad de recuperar algún fichero de forma independiente

ESCENARIOS DE PÉRDIDA DE INFORMACIÓN

- CAUSAS:
 - Errores de usuario
 - Software destructivo y virus
 - Personas malintencionadas
 - Fallos mecánicos
 - Fuerzas mayores (desastres naturales...)
- ERRORES HUMANOS: (*el usuario puede destruir información de forma no intencionada*)
 - Comandos mal escritos
 - Errores durante el redireccionamiento y uso de tuberías
 - Usuarios con acceso de root
 - PREVENCIÓN DE ERRORES HUMANOS:
 - Utilizar alias:

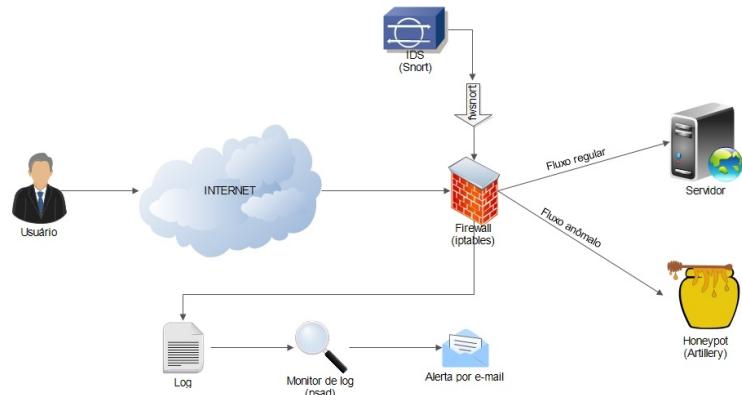
```
alias rm='rm -i' # El -i fuerza confirmacion
```
 - Utilizar sistema de control de versiones (Git): conservan el archivo original y llevan un histórico de los cambios realizados sobre éste
 - Crear copias de seguridad personales
 - Utilizar **sudo** para limitar el acceso de los usuarios con privilegios de root
 - *Solo se limita el acceso a los comandos necesarios para que el usuario pueda llevar a cabo su tarea*

VIRUS Y SOFTWARE DESTRUCTIVO

- **VIRUS:** programa que se adhiere a un ejecutable y se propaga a otros al mismo tiempo que realiza otra acción
 - Caballos de Troya: programas que se hacen pasar por otros (funcionando como estos y además realizando otras operaciones)
 - El grado destructivo depende de quien los ejecuta
 - Gusanos: Programas que se aprovechan de las debilidades de un sistema para propagarse a otros
 - Software destructivo: aplicaciones no mal intencionadas pero con errores de propagación que pueden ser dañinos

- Linux dispone de mecanismos de seguridad que dificultan su propagación
- Medidas de prevención sencillas:
 - Software específico de búsqueda y destrucción de virus
 - Configuración del entorno
 - Hosts y redes víctimas:
 - Se usan equipos y redes para probar software nuevo o descubrir nuevos ataques
 - Se suelen basar en un sistema de detección de intrusos (IDS) que genera reglas para el firewall separando el tráfico normal del anómalo

HONEYPOTS



PERSONAS MALINTENCIONADAS

- **CRACKERS:** (Son distintos a los hackers) : personas que entran en los sistemas de forma ilegal con fines malintencionados
 - Medida preventiva: cortafuegos y seguridad física
- **USUARIOS DESCONTENTOS:** Usuario con acceso al sistema y recelo
 - Medida preventiva: seguimiento controlando sus accesos y privilegios

FALLOS DE HARDWARE

- Fallo en la unidad de disco duro: el kernel suele avisar antes de un fallo completo
- Fallo de la memoria: pérdida de información por la caída del sistema o información corrupta en memorias copiadas a disco
- PREVENCIÓN Y RECUPERACIÓN:
 - Redundancia de la información: utilizar RAID
 - Supervisión de registros del sistema
 - Recuperación desde copias de seguridad
 - Intentar leer bloques para construir una imagen con 'dd'
 - Recuperación en entorno estéril

CONSEJOS GENERALES

1. Etiquetar siempre las copias realizadas
2. Elegir correctamente la frecuencia de copias
3. Usar particiones distintas para el sistema de ficheros
4. Hacer que el backup diario quepa en la unidad
5. Llevarse la copia a otro lugar y protegerse de este

6. Limitar la carga computacional durante el proceso de backup
7. No esperar a que ocurra un problema para verificar las copias
8. Tener en cuenta el tº de vida de los dispositivos

FACTORES A CONSIDERAR EN UNA ESTRATEGIA DE COPIAS DE SEGURIDAD

- ¿Qué ficheros se deben copiar y dónde están esos ficheros?
- Conocer lo más importante del sistema
- ¿Quién hará la copia?
- ¿Dónde, cuándo y bajo qué condiciones se deben hacer? -> *mejor hacerlas cuando no haya usuarios trabajando*
- Frecuencia de cambios en los ficheros
- ...

ESTRATEGIAS DE COPIAS DE SEGURIDAD

	COPIA DE SEGURIDAD COMPLETA	COPIA DE SEGURIDAD PARCIAL	COPIA DE SEGURIDAD INCREMENTAL
¿QUÉ SE GUARDA?	Todos los archivos asociados a un ordenador	Sólo algunos archivos específicos	Aquellos ficheros que hayan cambiado desde la última copia <i>Nivel 0-> se copia el backup completo</i> <i>Nivel 1-> todos los ficheros que cambien desde el backup de nivel 0...</i>
RESTAURACIÓN	Necesita un solo fichero pero tarda mucho tiempo Además es difícil recuperar un archivo suelto	Proceso sencillo ya que hay menos archivos implicados	Debemos asociar nosotros una estrategia de restauración
CUANDO LA HACEMOS	ante grandes cambios		Casi a diario
INCONVENIENTES		nos dejamos archivos sin copiar	

SOPORTES DE SEGURIDAD

CRITERIOS PARA REALIZAR LAS COPIAS

- Guardar las copias de seguridad en el mismo disco no es seguro, hay multitud de dispositivos:
 - Cintas magnéticas
 - Discos extraíbles: disco duro que puedes extraer sin apagar la máquina
 - CD-Roms o DVD's regrabables
 - Disquetes
 - Librería de cintas

CRITERIOS PARA ELEGIR EL SOPORTE

- COSTE: tanto del dispositivo como del soporte físico
- SOPORTE KERNEL
- CAPACIDAD DE ALMACENAMIENTO de datos de los soportes físicos
- MECANISMO DE CARGADOR AUTOMÁTICOS:
 - Cuando se llena una cinta se inserta otra automáticamente
 - Permite las copias no supervisadas de grandes volúmenes

COMPARATIVA DISCOS VS CINTAS

VENTAJAS DE LAS CINTAS:

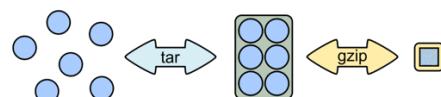
1. Alta capacidad
2. Menor coste
3. Seguridad: se desconectan al terminar las copias e implementan sistemas para evitar que los datos se sobrescriban
4. Fiabilidad: los discos siempre están conectados y en marcha
5. Durabilidad: pueden durar hasta 30 años
6. Velocidad secuencial

DESVENTAJAS: (*no afectan a los discos*)

1. Acceso aleatorio lento
2. Necesitan un mantenimiento especial
3. El tiempo de recuperación es mayor
4. Se duplica mucha información

COPIAS DE SEGURIDAD Y RESTAURACIÓN

TAR



Opciones:

- **c** -> crea un fichero contenedor
 - **x** -> extrae ficheros de un fichero contenedor
 - **v** -> Modo verbose (mayor cantidad de mensajes)
 - **f** -> permite especificar el nombre del fichero contenedor
 - **z**-> comprime o descomprime mediante gzip
 - **j**-> comprime o descomprime mediante bz2
 - **p**-> conserva los permisos de los ficheros
 - **P**-> guarda los ficheros con su ruta absoluta
 - **N** -> Considera solo archivos cuya fecha sea superior al argumento
-
- **tar cpf /dev/nst0 /home** -> copia todos los ficheros del directorio /home en la unidad cinta
 - **tar tzvf practicas.tgz** -> listar el contenido de la copia de seguridad realizada en el fichero
 - **tar xzvf practicas.tgz** -> descomprimir
 - **tar xzvf practicas.tgz prac_aso/boletin1.pdf** -> recuperar el fichero boletin1 (hay que indicar la ruta con la que tar lo almacenó)
 - **tar cf practicas.tar -N '3 days ago'** -> copia los ficheros creados/modificados hace menos de 3 días

CPIO

Copias de seguridad de conjuntos de ficheros seleccionados arbitrariamente

- Empaque los datos en una cinta más eficientemente que tar
- Lee de la entrada estándar el nombre de los ficheros a guardar, para usarlo enlazado con otras ordenes con tuberías

Opciones:

- **o** -> copiar fuera (crear la copia)
- **i**-> copiar dentro (descomprimir)
- **m** -> conserva fecha y hora de los ficheros
- **t**-> muestra la tabla de contenidos, el contenido de la copia
- **A**-> Añade ficheros a un contenedor existente
- **d** -> crear directorios al descomprimir
- **v** -> modo verbose
- **F** -> crear la copia en un fichero
- Find /home | cpio -o >/dev/nst0 -> se copia en la unidad de cinta
- Find /home | cpio -o -F h.cpio -> la copia la realiza en un fichero
- Cpio -i <h.cpio -> restaura la copia de seguridad de ese fichero
- Cpio -i -F h.cpio fichero -> restaura sólo el fichero indicado

DUMP

- Hace copias de seguridad de un sistema de ficheros Ext2, Ext3 o Ext4, copiando la partición completa.
- Permite realizar copias de seguridad por niveles: desde el nivel 0, copia completa, al nivel 9 (que es el valor por defecto).
- Actúa solo a nivel de dispositivo
- /etc/dumpdates: información sobre las copias de seguridad de cada SF y de qué nivel son

Opciones:

- **0-9** -> nivel de copia de seguridad, no requiere argumento
- **-u**-> actualiza /etc/dumpdates, no requiere argumento
- **-f** -> indica fichero destino diferente al usado por defecto, si requiere argumento (por defecto se usa la unidad de cinta)

RESTORE

- Restaura copias de seguridad creadas con dump.
- Permite recuperar ficheros, directorios y SF enteros.
- Se ha de recuperar el más reciente de cada nivel empezando por el 0
- Para recuperar SF → crear y montar un SF limpio y vacío, entrar en el punto de montaje y deshacer el *backup*.

Opciones:

- **-r** -> restaura la copia completa, no requiere argumento
- **-f**-> indica el dispositivo o archivo donde está el backup (requiere argumento)
- **-i** -> modo interactivo (no requiere argumento)
- **-x** -> extrae los archivos y directorios desde el directorio actual
- **-t** -> imprime los nombres de los archivos de la copia (no requiere argumentos)

- `dump 0 -u -f /dev/nst0 /dev/sda1` → Copia de nivel 0 de `/dev/sda1` en la unidad de cinta, actualizando `/etc/dumpdates`.
- `dump 1 -u -f /dev/nst0 /dev/sda1` → Copia de nivel 1 de `/dev/sda1` en la unidad de cinta, actualizando `/etc/dumpdates`.
- `dump 0 -f jj.dump /dev/sda1` → Copia de nivel 0 de `/dev/sda1` en el fichero `jj.dump`.
- `restore -t -f fichero_backup` → listado de la copia.
- `restore -x -f fichero_backup practicas/smallsh.c` → restaura sólo el fichero `practicas/smallsh.c`.
- `restore -r -f /dev/nst0` → restaura una copia completa.
- `restore -i -f /dev/nst0` → permite restaurar ficheros interactivamente (con `ls`, `cd`, `pwd`, `add` y `extract`).

RESTAURACIÓN DE UN SISTEMA COMPLETO

Si se tiene una copia de todo el sistema:

1. Arranca desde un dispositivo distinto
2. Si es necesario crear los ficheros especiales de dispositivos para los discos
3. Prepara el disco duro, crear las particiones
4. Crear el sistema de ficheros en la partición donde se restaurarán los datos y montarlo en un directorio
5. Restaurar la copia de seguridad sobre ese sistema de ficheros
 - a. Restaurar la copia más reciente de nivel 0
 - b. Restaurar la copia más reciente del nivel más bajo después del último restaurado
 - c. Si quedan más copias por restaurar, volver al paso anterior
6. Desmontar el sistema de ficheros restaurado
7. Volver al paso 2, para restaurar otros SF adicionales

De las siguientes copias realizadas, ¿qué copias de seguridad se restaurarían?:

- 0 0 0 0 0 0 .
- 0 5 5 5 5 5 .
- 0 3 2 5 4 5 .
- 0 9 9 5 9 9 3 9 9 5 9 9 .
- 0 3 5 9 3 5 9 .

▪ Solución (restauraciones en negrita):

- 0 0 0 0 0 0 .
- 0 5 5 5 5 5 .
- 0 3 2 5 4 5 .
- 0 9 9 5 9 9 3 9 9 5 9 9 .
- 0 3 5 9 3 5 9 .

TEMA 10

GESTIÓN DE LAS COMUNICACIONES



CONCEPTOS BÁSICOS

TAREAS DE GESTIÓN DE LA RED

Tareas:

- Manejo de la red
- Monitorizar el tráfico
- Añadir nuevos hosts
- Montar discos remotos o exportar los discos locales
- Servicio de información
- Configurar y administrar otros servicios de red
- Prevenir problemas de seguridad
- Enrutado de tráfico

Labor mínima: opciones de configuración de la red más importantes

xinetd: para administrar servicios en Linux

- Maneja a otros demonios a los cuales inicializa cuando hay un trabajo para ellos
- */etc/xinetd.conf* -> fichero de configuración de xinetd
- */etc/xinet.d/* -> ficheros de configuración de los demonios gestionados por xinetd

DEMONIOS MÁS COMUNES

/etc/init.d/networking o */etc/init.d/network-manager* : script que activa la red en tiempo de arranque

- Algunos demonios:
 - **Ntpd:** encargado de sincronizar la hora del sistema
 - **Dhcpd:** encargado del servicio de Dynamic Host Configuration Protocol (servidor que proporciona IPs privadas a las máquinas que se conecten)
 - **Named:** encargado del servicio de Domain Name System (servicios que traduce nombres de demonio)
 - **Sendmail:** permite ssh (conexión remota segura)
 - **Httpd:** servidor web
 - **Smbd:** servicio de compartición de ficheros con Windows

NFS: NETWORK FILE SYSTEM

CONCEPTOS BÁSICOS

- Posibilita que un sistema de ficheros que físicamente reside en un host remoto se pueda usar en otros ordenadores
- En el servidor se indica: Que sistemas de ficheros se exportan (*puede ser completo o 1 solo directorio*)
 - A qué ordenadores se exportan (*a uno concreto o a todos los equipos de una red*)
 - Condiciones para la exportación
- Los equipos cliente montan el sistema de ficheros remoto con **mount**, acceden a los datos como si fueran locales e incorporan en cada operación una cookie secreta que se les manda cuando montan el directorio
- Al exportar un fichero se exporta su nodo-i y sus bloques de datos
- Un equipo puede ser servidor y cliente NFs al mismo tiempo
- VERSIONES:
 - NFS≤2 : Operaciones de escritura bloqueantes
 - NFS=3: esquema que permite escrituras asíncronas (mayor eficiencia)
 - NFS=4: funcionalidades adicionales

ORGANIZACIÓN Y ARQUITECTURA

- Se basa en el protocolo RPC (REMOTE CALL PROCEDURE) -> encapsula llamadas al servidor cuando se piden archivos remotos
- **Stateless (v2/v3)**: el servidor trabaja sin mantener información del estado de cada uno de los clientes
 - Necesidad de bloquear archivos accedidos concurrentemente por varios clientes (demonios independientes)
 - El cliente es responsable de mantener la coherencia
- **Statefull (v4)**: el servidor trabaja manteniendo el estado de las operaciones
- NFS tiene bastantes problemas de seguridad por lo que necesita utilizar herramientas adicionales

LADO SERVIDOR

NFS: CONFIGURACIÓN DEL LADO SERVIDOR

- **/etc/exports**: fichero donde se indica que SFs se exportan, en qué condiciones y a qué ordenadores
- **/usr/sbin/exportfs**: actualiza la información de los SFs exportados y muestra un listado con dicha información:
 - -r -> re-exporta los directorios indicados en /etc/exports
 - -a -> exporta o deja de exportar /etc/exports
 - -v -> muestra los directorios exportados y las opciones
- **/usr/sbin/showmount** : información en un servidor NFS
 - -a -> clientes conectados y directorios utilizados
 - -d -> listado de los directorios montados

DEMONIOS EN EL LADO SERVIDOR

- **rpcbind o portmap**: facilita la conexión entre el cliente y el servidor mediante las llamadas RPC (*tiene que estar lanzado para que NFS funcione*)
- **nfsd**: implementa (en nivel usuario) los servicios NFS
- **rpc.mountd**: maneja las peticiones de montaje de directorios de los clientes
 - para lanzar rpc.mountd y rpc.nfsd -> /etc/init.d/nfs-kernel-server
- **OPCIONES EN EL SERVIDOR**
 - **/etc/exports**: para configurar que directorios se exportan, bajo qué condiciones y a qué equipos

ruta dirección(opción)

- RUTA: Nombre del directorio a exportar vía NFS
- DIRECCIÓN: a quién es exportado
- OPCIÓN: tipo de acceso al directorio
 - Rw o ro -> modo lectura-escritura o solo lectura
 - Root_squash -> mapea uid/gid 0 o uid/gid anónimo
 - No_root_squash -> no hacer lo anterior
 - Anonuid o anongid -> establecer uid o gid del usuario al que realizar el mapeo

LADO CLIENTE

NFS: CONFIGURACIÓN DEL LADO CLIENTE

MOUNT: permite montar el SF remoto

```
$ mount -t nfs -o opciones_nfs 191.168.6.10:/home /datos
```

- -t nfs: tipo de SF
- 191.168.6.10:/home : servidor y directorio remoto a montar
- Si en el fichero /etc/fstab se indica el listado de los sistemas de ficheros remotos a montar, el punto de montaje y las opciones, el montaje se puede realizar en tiempo de arranque

OPCIONES PARA MOUNT

- SOFT: Si el servidor NFS falla durante un tiempo, las operaciones que intentaban acceder a él recibirán un código de error
 - o Va en contra de la filosofía NFS
- HARD: Si un proceso está realizando una operación de E/S con un fichero vía NFS y este no responde el proceso no puede ser parado salvo que usemos la opción initr
 - o POR LO QUE SIEMPRE QUE USEMOS RW USAR HARD (Para no dejar SF inconsistente)
- INITRD: Se permiten señales de interrupción para los procesos bloqueados en una operación de E/S
- BG: Si el montaje del SF remoto falla, que siga intentándolo en background
- RETRY=n: nº de intentos que se deben hacer para montar el SF remoto antes de desistir si la conexión falla
- TIMEO=n: tiempo a esperar entre cada intento de montaje si la conexión falla
- RSIZE=8192 o WSIZE=8192: Tamaño de los buffers de lectura o escritura

VER EJEMPLOS EN LA PÁGINA 8 DEL PDF

NIS: NETWORK INFORMATION SYSTEM

CONCEPTOS BÁSICOS

Ficheros de configuración -> muchos ficheros de configuración son parecidos en una máquina u otra, si tenemos n máquinas tenemos n réplicas de ficheros que gestionar -> esto es muy difícil.

Por lo que con NIS (servicio de red para compartir cierta información):

- Todos los servicios acceden a una misma base de datos de configuraciones
- Permite centralizar la autenticación de servicios

Aunque también tiene INCONVENIENTES:

1. Solo para una subred y no cifra los datos
2. No permite establecer jerarquías de usuarios complejas
3. Un cambio supone reconstruirlo todo y redistribuirlo

Los servicios de las bases de datos están en el equipo servidor, este distribuye información que contiene a los clientes

En el lado servidor:

- Los ficheros se preprocesan para convertirlos a un formato binario con hashing
- Dominios NIS -> Clave para poder localizar al servidor
- Los ficheros de las BDs residen a partir del directorio /vay/yp en un subdirectorio con el nombre del dominio

CONFIGURACIÓN

Podemos configurar varios servidores esclavos (*tendrán una copia en las bases de datos*) y un cliente podrá acudir a varios servidores

- NSS (Name Service Switch): Indicar como se resolverá cierta información de configuración ([/etc/nsswitch.conf](#))
- DEMONIOS:
 - **Rpcbind** o **portmap**: facilita la conexión entre el cliente y el servidor mediante las llamadas RPC
 - **Ypserv**: encargado de gestionar el servicio NIS (tiene que estar en ejecución en el servidor)
 - **Rpc.yppaswdd**: permite la actualización de contraseñas desde los equipos clientes (en ejecución en el servidor)
 - **Ypbind**: encargado de gestionar las peticiones (en el cliente)

INSTALACIÓN DEL SERVIDOR

1. Instalar el paquete **nis**
 - a. Indicar dominio a utilizar (*pas_nis*)
 - b. Esperar el intento fallido de **binding**
2. Cambiar el fichero */etc/default/nis* e indicar **NISERVER=MASTER**
3. Añadir la IP del servidor al fichero */etc/yp.conf*: con **ypserver localhost**
4. Configurar el servidor (crear la base de datos): `sudo /usr/lib/yp/ypinit -m`
 - a. Repetir este paso cada vez que cambiemos la base de datos
5. Reiniciar el servicio: `sudo /etc/init.d/nis restart`
6. Comprobar que todo funciona con **rpcinfo -p**
7. Configurar el NSS (*/etc/nsswitch.conf*)

INSTALACIÓN EN EL CLIENTE

1. Instalar el paquete **nis**
 - a. Indicar dominio a utilizar (*pas_nis*)
 - b. Esperar el intento fallido de **binding**
2. Añadir la IP del servidor al fichero */etc/yp.conf*: con **ypserver localhost**
3. Configurar el NSS
4. Reiniciar el servicio: `sudo /etc/init.d/nis restart`

El dominio por defecto se encuentra en */etc/defaultdomain*

AÑADIR UN USUARIO

1. Añadir el usuario desde la máquina principal con **adduser**
2. Reconstruir y distribuir los mapas NIS: **make .c /var/yp**
3. Ya estaría listo para usarse dentro de la red

SEGURIDAD

- Utilidades como clientes:
 - **Yppasswd**: permite que los usuarios puedan cambiar su contraseña en el servidor NIS
 - **Ypchsh**: permite cambiar el Shell del usuario
 - **Ypchfn**: cambia el campo gecos del usuario
 - **Ypcat**: permite conocer el contenido de un mapa NIS

- **Ypcat passwd:** visualiza el fichero passwords
- **Ypcat ypservers:** muestra los servidores disponibles
- **Ypwhich:** devuelve el nombre del servidor NIS
- En el fichero /etc/ypserv.conf se pueden indicar listas de control de acceso
 - FORMATO: HOST:NISDOMAIN:MAP:SECURITY

SAMBA

INTRODUCCIÓN

- Aparece de los inconvenientes del NFS, problemas de seguridad, no existe una implementación libre para Windows...
 - Existe una implementación libre llamada SAMBA
- ¿QUÉ ES?
 - Sistema de compartición de archivos e impresoras en red
 - Permite la interconexión de sistemas heterogéneos entre sí (por ej Linux y Windows)
- PROTOCOLOS:
 - **SMB (Server Message Block):** compartir los recursos
 - **CIFS (Common Internet File System) :** implementación mejorada de SMB
 - **NetBIOS (Networking Basic Input/Output System)** servicio de nombres
- Es útil cuando:
 - No quieres pagar un servidor Windows NT
 - Homogeneizar la red local ante clientes Windows y Unix
 - Compartir impresora entre clientes Windows y Unix
- Utiliza dos demonios:
 - **Smbd:** permite compartir archivos e impresoras sobre una red SMB
 - Proporciona autenticación y autorización de acceso para clientes SMB
 - **Nmbd:** ANUNCIA SERVICIOS -> Informa a las máquinas en la red de cuales son los servicios disponibles
- Podemos configurar SAMBA mediante el fichero smb.conf o mediante el front-end SWAT (*no se recomienda ya que es poco seguro*)

CONFIGURACIÓN

Lo recomendable es mediante /etc/samba/smb.conf

- Qué recursos del sistema vas a compartir y que restricciones deseas poner en ellos
- Consta de varias secciones distintas que empiezan por [nombre-recurso]
 - [global]: variables de carácter general, aplicables a todos los recursos
 - [homes]: permite a usuarios remotos acceder a su directorio personal desde su máquina local
 - [printers]: para compartir impresoras

INICIO CON SAMBA: /etc/init.d/samba start

PARADA CON SAMBA: /etc/init.d/samba stop

VER EJEMPLOS EN LAS PÁGINAS 14 Y 15 DEL PDF