C/ Literat Azorín, 1 46702 – GANDIA
Tfno. 962829430 – Fax. 962829431
Codi: 46004221 – CIF: Q-9655627-I
46004221@edu.gva.es
http://iesmariaenriquez.com

COMPUTER SYSTEMS
UD5: O.S. ADMINISTRATION
UBUNTU ADMINISTRATION

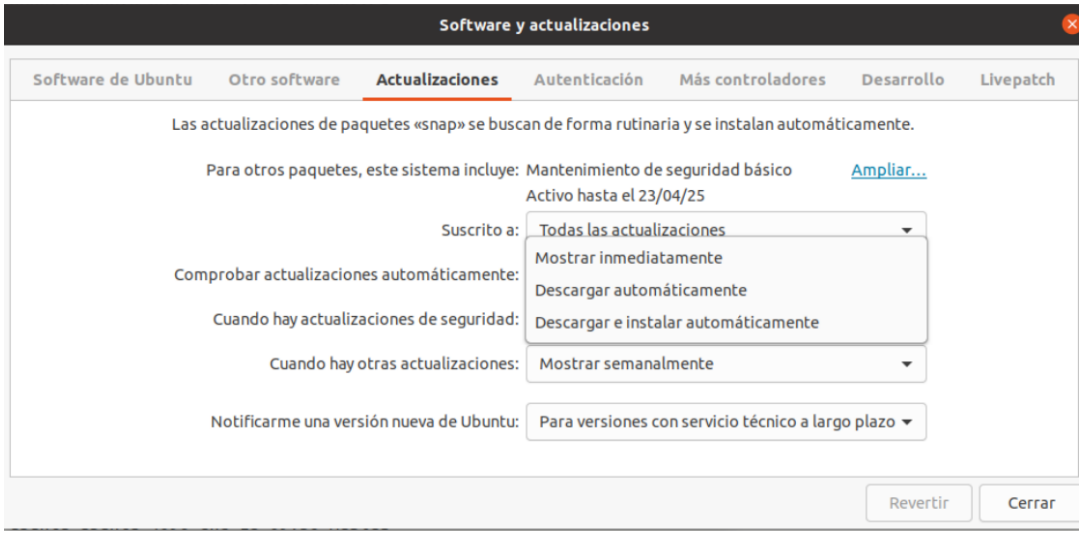
CFGS DAW
DPT INF

ÍNDICE

1. Instalación de aplicaciones y actualizaciones

A fin de preservar la seguridad del sistema operativo es conveniente que se actualice periódicamente. Accedemos al botón de inicio de Ubuntu (**Dash**) y escribimos **Actualización**, entre los elementos encontrados hacemos clic sobre **Actualización de software**.

Desde el botón Configuración indicamos desde donde realizaremos las descargas de software y los momentos en que deben realizarse las actualizaciones, así como qué hacer ante actualizaciones de seguridad.



En Ubuntu existen distintos tipos de aplicaciones, clasificadas en función del formato del fichero que las incluye:

- **Paquetes DEB**
- **Archivos binarios**. Ficheros con extensión **.bin**. Son ejecutables en sí mismos. Para ejecutarlos, debemos agregarle el permiso de ejecución y posteriormente proceder a su ejecución a través del entorno gráfico o en modo texto escribiendo **./nombre.bin.**
- **Archivos run.** Son programas que disponen de un asistente para el proceso de instalación. Su ejecución se lleva a cabo a través del terminal introduciendo la orden: **sh ./nombre.run** o **sudo sh ./nombre.run.**
- **Ficheros tar.gz o tar.bz2.** Este tipo de ficheros comprimidos contienen los códigos fuente de la aplicación que queramos instalar. Normalmente, la instalación de una aplicación se lleva a cabo mediante la secuencia de estos pasos:

    a. Descomprimir el fichero. Si es un fichero tar.gz usaremos **gzip**, en caso de ser un tar.bz2 **bzip2.**
    b. Ejecutamos el fichero **configure** (./configure), este fichero detectará las características del sistema, configurará la compilación y creará el fichero makefile.
    c. Ejecutamos **make** que procede con la compilación del código fuente.
    d. Ejecutamos **make** install para comenzar con la instalación.

https://ubuntinux.blogspot.com/2020/10/instalar-desde--las-fuentes.html

Nota: Cuando adquirimos un software a través de sus ficheros fuentes observar siempre la existencia de un fichero INSTALL, LEEME, README, o algo similar en la raíz de la carpeta descomprimida del mismo, ya que será en este fichero donde el autor incluya las instrucciones precisas para proceder a la instalación del programa.

Debian, sistema en el que está basado Ubuntu, tiene herramientas interactivas como **tasksel**, que permiten escoger unos subconjuntos de paquetes agrupados por tipo de tareas. En el nivel de línea de comandos, dispone de **dpkg**, que es el comando de más bajo nivel, para gestionar directamente los paquetes DEB de software [Deb], típicamente dpkg -i paquete.deb.

El nivel intermedio lo presentan las herramientas **APT** (la mayoría son comandos apt-xxx). APT permite gestionar los paquetes por medio de una lista de paquetes actuales y disponibles a partir de varias fuentes de software, ya sea desde los propios CD de la instalación, sitios ftp o web (HTTP). La configuración del sistema APT se efectúa desde los archivos disponibles en /etc/apt, donde /etc/apt/sources.list es la lista de fuentes disponibles.

La actualización de la lista de paquetes disponibles se realiza mediante el comando: apt-get update. (actualiza el fichero **/etc/apt/sources.list** con la lista de orígenes o repositorios a utilizar para encontrar los paquetes).

Aunque **Synaptic** no es la aplicación por defecto en Ubuntu 20.04 para la instalación de paquetes, todos los usuarios pueden usar esta herramienta procediendo a su instalación desde el centro de software.

2. Drivers en Ubuntu

Los controladores o drivers son programas que se instalan en el sistema operativo, los cuales permiten comunicar a las partes del ordenador con el sistema. Existen multitud de controladores, por ejemplo, controlador de la impresora, de la tarjeta gráfica, de la placa base etc.

La mayoría de ellos vienen instalados en Ubuntu por defecto, aunque son los controladores libres y no los privativos.

El paquete **Hardinfo** (no viene instalado por defecto) es el símil al **Administrador de dispositivos** de Windows. Desde él accedemos a Computer, Devices, Network y Benchmark, desde donde podemos observar toda la información relativa al PC.

Desde la línea de comandos puede sernos de utilidad el comando **dmseg.** (Accede a una ubicación en la memoria que se conoce como el *buffer del anillo Kernel*. Esta parte de memoria se usa para registrar eventos del Kernel, como cambios en el hardware.)
https://vidatecno.net/usando-dmesg-en-linux/

Si con alguna de las anteriores opciones observamos que algún dispositivo no se corresponde con el que tenemos, podemos reinstalarlo descargando la versión del controlador que necesitemos accediendo a la Web del fabricante.

3. Administración de usuarios

GNU/Linux es un sistema operativo multiusuario. Esto significa que permite a varios usuarios utilizar el sistema simultáneamente, a través de la línea de comandos o con conexiones remotas. GNU/Linux controla el acceso al equipo y a sus recursos a través de las cuentas de usuarios y grupos.

En los sistemas GNU/Linux existen tres tipos de usuarios:

- ✔ **Root**. Es el usuario más importante ya que es el administrador y dueño del sistema. Se aconseja utilizar la cuenta de root para las tareas específicas de administración y el resto del tiempo utilizar una cuenta de usuario normal.
- ✔ **Usuarios normales**. Son los usuarios que pueden iniciar sesión en el sistema y tienen una funcionalidad limitada, tanto en los comandos que pueden ejecutar, como a los ficheros a los que tienen acceso.
- ✔ **Usuarios asociados a servicios**. Este tipo de usuarios no pueden iniciar sesión en el sistema. Su utilización es muy útil ya que permiten establecer los privilegios que tiene un determinado servicio. Por ejemplo, el servidor de páginas Web tiene asociado un usuario para poder especificar a qué ficheros tiene acceso; y por lo tanto que ficheros son visibles a través de Internet.

Todos los usuarios del sistema tienen un *identificador de usuario* (**UID**) y un *identificador de grupo* (**GID**). El administrador del sistema root tiene los identificadores de usuario y grupo 0:0 y los demás usuarios tienen un valor mayor que 0.

Existen varias formas de administrar el sistema, que van variando dependiendo de su facilidad o control sobre el sistema. Básicamente, puede administrar el sistema a través de tres formas diferentes:

- **Interfaces gráficas**. Existen diferentes interfaces gráficas que permiten administrar el sistema de una forma fácil y sencilla. Puede utilizar la interfaz de administración de x-Windows. Este método es el más sencillo, pero es el que menos control proporciona sobre el sistema.

· **Terminal del sistema**. Una de las ventajas de los sistemas GNU/Linux es que puedes administrarlo totalmente a través del intérprete de comandos o terminal del sistema. Una de las grandes ventajas de utilizar el terminal del sistema es que permite una gran flexibilidad a la hora de interactuar con el sistema, pudiendo crear pequeños programas (scripts) para simplificar la administración del sistema.

· **Ficheros de configuración**. Por último, la modificación directa de los ficheros de configuración es el método que permite tener un mayor control del sistema. Como desventaja hay que destacar que para administrar el sistema de esta forma hay que conocer muy bien el sistema.

3.1. Intérprete de comandos.

La gestión de usuarios y grupos se puede realizar directamente a través del intérprete de comandos. En la siguiente tabla se muestran los comandos más importantes para la gestión de usuarios y grupos.

Comandos de administración y control de usuarios	
adduser	Ver useradd
chage	Permite cambiar o establecer parámetros de las fechas de control de la contraseña.
chpasswd	Actualiza o establece contraseñas en modo batch, múltiples usuarios a la vez. (se usa junto con newusers)
id	Muestra la identidad del usuario (UID) y los grupos a los que pertence.
gpasswd	Administra las contraseñas de grupos (/etc/group y /etc/gshadow).
groupadd	Añade grupos al sistema (/etc/group).
groupdel	Elimina grupos del sistema.
groupmod	Modifica grupos del sistema.
groups	Muestra los grupos a los que pertence el usuario.
newusers	Actualiza o crea usuarios en modo batch, múltiples usuarios a la vez. (se usa junto chpasswd)
pwconv	Establece la protección shadow (/etc/shadow) al archivo /etc/passwd.
pwunconv	Elimina la protección shadow (/etc/shadow) al archivo /etc/passwd.
useradd	Añade usuarios al sistema (/etc/passwd).
userdel	Elimina usuarios del sistema.
usermod	Modifica usuarios.

A continuación mostramos las instrucciones para la creación de un usuario llamado mortadelo, con la **shell** por defecto que en **Debian** es /bin/sh, le asignará como directorio personal /home/mortadelo, pero no creará el directorio ni se copian los directorios y archivos contenidos en el **skeleton directory**, crea el grupo **mortadelo** y lo asigna al mismo y **no**

tiene contraseña:

$ sudo adduser mortadelo

Si ejecutamos el siguiente comando, veremos como ha quedado el archivo **/etc/passwd**:

$ cat /etc/passwd | grep mortadelo

mortadelo:x:1001:1001::/home/mortadelo:/bin/sh

Veámos cómo ha quedado el archivo **/etc/group**:

$ cat /etc/group | grep mortadelo

mortadelo:x:1001:

Para asignar una contraseña a mortadelo ejecutaríamos el siguiente comando:

$ sudo passwd mortadelo

Para copiar el esqueleto de directorios ejecutamos el comando siguiente:

$ sudo cp -r /etc/skel/* /home/mortadelo

Para crear el directorio personal con permisos de lectura, escritura y ejecución para el usuario, escritura y ejecución para el grupo y sin permisos para el resto de usuarios, ejecutaremos el siguiente comando:

$ sudo mkdir -m 750 /home/mortadelo

Para asignar el usuario y grupo mortadelo al directorio, ejecutaremos el siguiente comando:

$ sudo chown mortadelo:mortadelo /home/mortadelo/

3.2. Ficheros utilizados.

Para conocer el funcionamiento interno del sistema operativo debemos conocer dos tipos de ficheros: aquellos que se utilizan para guardar la información de los usuarios y grupos, y los ficheros con los valores predeterminados que utiliza el sistema.

Al arrancar el sistema o cada vez que el usuario inicia una sesión comienza a ejecutarse un shell que leerá los archivos **/etc/environment**, **/etc/profile**, **/etc/bash.bashrc**.

A continuación se ejecutará el archivo **~/.bash_profile**, **~/.bash_login** o **~/.profile**, que están en el directorio personal del usuario que inicia la sesión y que sirven para que el usuario pueda configurar ciertos parámetros de su cuenta.

.bashrc: Cuando el usuario accede al sistema especificando su password y este es validado, el sistema operativo localiza en su carpeta personal el fichero .bashrc. Es el fichero del perfil del usuario.

.profile: En caso de que el usuario acceda a un terminal, o a través de una de sus consolas en modo texto, se leerá el fichero .profile para inicializar variables de entorno a

los valores correspondientes.

.bash_logout: Cuando el usuario cierra sesión será este el fichero que lea. Su contenido se muestra a continuación. (Este fichero puede no existir)

```
ubuntu@ubuntu-VB:~$ cat .bash_logout
# ~/.bash_logout: executed by bash(1) when login shell exits.

# when leaving the console clear the screen to increase privacy

if [ "$SHLVL" = 1 ]; then
    [ -x /usr/bin/clear_console ] && /usr/bin/clear_console -q
fi
```

Por motivos de seguridad, las contraseñas de los usuarios se almacenan en el fichero **/etc/shadow** y no en el fichero **/etc/passwd**. En el fichero **/etc/passwd** en vez de almacenar la contraseña se guarda el carácter "**x**" y en el fichero **/etc/shadow** se almacena la contraseña cifrada.

Para cada grupo el sistema almacena el nombre del grupo, el identificador de grupo (GID) y los usuarios que pertenecen al grupo.

Archivos de administración y control de usuarios	
.bash_logout	Se ejecuta cuando el usuario abandona la sesión.
.bash_profile	Se ejecuta cuando el usuario inicia la sesión.
.bashrc	Se ejecuta cuando el usuario inicia la sesión.
/etc/group	Usuarios y sus grupos.
/etc/gshadow	Contraseñas encriptadas de los grupos.
/etc/login.defs	Variables que controlan los aspectos de la creación de usuarios.
/etc/passwd	Usuarios del sistema.
/etc/shadow	Contraseñas encriptadas y control de fechas de usuarios del sistema.

Al dar de alta un usuario si no especifica ningún parámetro el sistema utiliza los valores por defecto. El sistema guarda los valores por defecto en los siguientes ficheros:

/etc/default/useradd Permite establecer el shell que se va utilizar por defecto, el

directorio home que van a tener los usuarios, etcétera.

/etc/login.defs Entre las opciones más importantes permite establecer los datos de expiración de las contraseñas, longitud mínima de las contraseñas, UID y GID mínimos y máximos, etcétera.

3.3. Configuración con asistentes.

La administración de los usuarios del sistema se puede realizar gráficamente con la herramienta **Usuarios y grupos** en x-Windows. Si no viene instalada por defecto, ejecutaremos:

$ sudo apt install gnome-system-tools

Inicia la aplicación **Usuarios y grupos** que se encuentra en el submenú Administración dentro del sistema. Aparece la ventana Gestor de usuarios donde puedes realizar la administración de los usuarios del sistema de una forma fácil y sencilla.

Para añadir un nuevo usuario pulsa el botón Añadir, introduce el nombre de usuario, pulsa Aceptar y posteriormente introduce la contraseña del usuario.

En Linux se describen dos tipos de **grupos, primarios y secundarios**. Un usuario tiene asignado un grupo primario, este grupo es el que se define en **/etc/passwd** en el cuarto campo de cada línea (los campos van separados por ":" en cada línea del fichero). Cada vez que un usuario crea un fichero, en él aparecen los datos del propietario del mismo, es decir, quién lo creó y tiene asignado un grupo propietario que es el grupo primario de dicho usuario. Al mismo tiempo, un usuario puede pertenecer o no a uno o varios grupos secundarios, esto se indica en el fichero **/etc/group**.

4. Sistema de ficheros

4.1. Particionamiento

El particionamiento es uno de los procesos más importantes que hay que tener en cuenta, ya que define cómo se van a utilizar los diferentes discos duros del equipo. En el proceso de particionamiento hay que prestar un especial cuidado para no perder datos del sistema.

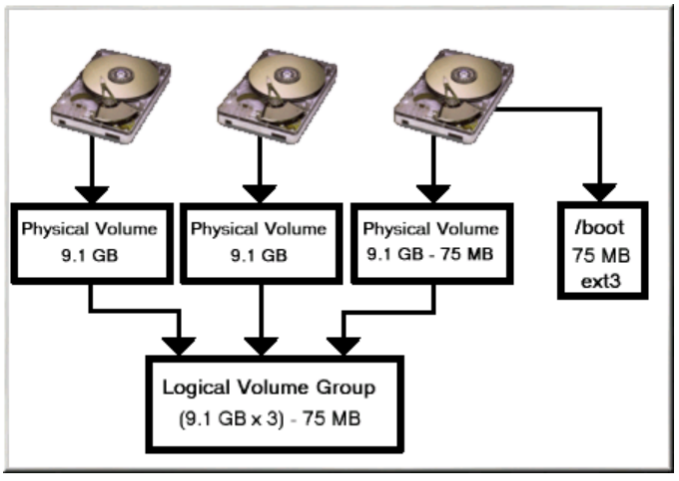
La administración de las particiones de los sistemas de ficheros se puede realizar con herramientas gráficas como la **Utilidad de discos**, el **Administrador de volúmenes lógicos** o, manualmente, con el comando **fdisk**.

Nota: En los servidores es recomendable utilizar un sistema RAID por hardware para permitir que, en caso de rotura de un disco duro, no se pierda la información del sistema.

4.1.1. Herramientas gráficas.

Ubuntu Desktop por defecto instala la herramienta **Utilidad de discos** para administrar el sistema de ficheros. Utilizando la herramienta **Utilidad de discos** puede crear, modificar o eliminar las particiones de los discos duros del sistema.

Para ejecutar la herramienta debes ir al menú **Sistema** > **Administración** y seleccionar la herramienta **Utilidad de discos**.



Grupo de volumen lógico

Por otra parte, es posible utilizar el Administrador de volúmenes lógicos. A diferencia de la herramienta Utilidad de discos, con el Administrador de volúmenes lógicos es posible crear volúmenes o unidades RAID. Recuerda que *un volumen permite agrupar uno o más discos duros para tener un sistema de ficheros de mayor tamaño*. Además, puede crear volúmenes en los que se mejore la seguridad de los datos. Por

ejemplo, en *un volumen reflejado* (o RAID 1) los datos se guardan de forma simultánea en dos discos duros.

La duplicación del disco se puede simular en un entorno de software. La utilidad **mdadm** se puede utilizar para crear, administrar y monitorear arreglos MD (discos múltiples) para RAID de software o E/S de múltiples rutas. La instrucción para instalar dicho paquete es:

 # apt-get install mdadm

Los pasos para la configuración del RAID 1 los puedes encontrar en :
https://conpilar.es/como-crear-linux-raid-1-espejo-usando-mdadm/

4.1.2. fdisk.

La utilidad **fdisk**, a pesar de que es un poco incómoda de utilizar porque no trabaja bajo una interfaz gráfica, es muy útil y potente. Los pasos que hay realizar para utilizar un disco son:

- ✔ Crear la partición.
- ✔ Fomatear la partición.
- ✔ Montar la partición.

Crear la partición

El primer paso a realizar es conocer los discos duros y particiones que tiene el sistema. Para ello ejecutamos:

 # fdisk –l

Por ejemplo, supongamos que el equipo tiene dos discos duros (/dev/sda y /dev/sdb). El primer disco duro (/dev/sda) tiene dos particiones donde está el sistema operativo (/dev/sda1) y la partición swap (/dev/sda2). Y el segundo disco duro no contiene ninguna tabla de particiones válida.

Si queremos utilizar fdisk en el segundo disco duro entonces hay que ejecutar:

 # fdisk /dev/sdb

Una vez dentro del disco duro el sistema informa que el disco duro no contiene ninguna tabla de particiones válida. Para conocer los comandos disponibles pulsamos **m**.

Para crear una partición pulsamos **n** y realizamos los siguientes pasos:

- ✔ Selecciona el tipo de partición que quieres crear: (p) primaria y (e) extendida. Pulsa **p**.
- ✔ Indica el número de la partición primaria. Como es la primera pulsa 1.
  - ✔ Ahora hay que indicar el tamaño de la partición.
  - ✔ A continuación, indica el último cilindro. Para especificar el tamaño de la partición puedes indicar el número del último cilindro o indicar el tamaño en

MegaBytes que quieres asignarle a la partición de la forma +tamañoM (p.e.: 1000M). Por ejemplo, pulsa Enter para utilizar todo el disco duro.

Una vez creada la partición pulsa **p** para ver la tabla de particiones.

Una vez realizados todos los cambios hay que guardar la configuración y salir de la aplicación, utilizando **w**.

Formateo

Una vez creada la partición, el siguiente paso es formatearla con el comando mkfs. Para formatear la partición ejecuta:

 # mkfs /dev/sdb1

Montar la unidad

Una vez lista la partición /dev/sdb1 para poder utilizarla hay que montarla en un directorio existente.

 # mkdir /datos

Existen dos formas diferentes de montar una partición:

1. **Manualmente** con el comando **mount**. Esta opción es la más sencilla y permite montar un sistema de ficheros de forma puntual ya que si se reinicia el ordenador se pierde el punto de montaje.

2. **Automáticamente** editando el fichero /etc/fstab. Esta opción permite montar de forma permanente un sistema de ficheros. Es la mejor opción en el caso de querer utilizar siempre el sistema de ficheros, o si quieres realizar en él acciones especiales como, por ejemplo, utilizar las *cuotas de usuarios*.

Para montar manualmente nuestra partición la sintaxis del comando sería:
mount -t sistema_de_archivos dispositivo punto_de_montaje

En nuestro caso, el comando a usar para montar el disco sería el siguiente:

Ej: # mount -t ext4 /dev/sdb1 /datos

Si deseamos montar de forma definitiva el sistema de ficheros entonces hay que editar el fichero **/etc/fstab** y añadir al final la siguiente línea de configuración. Hay que tener mucho cuidado al modificar el fichero /etc/fstab ya que puede dañar el sistema.

/dev/sdb1 /datos ext4 defaults 0 0

Una vez modificado el fichero de configuración, la partición se monta automáticamente al reiniciar el equipo o puedes montarla ahora ejecutando **mount /datos**. Esta opción resulta interesante para montar unidades de red, de forma que al

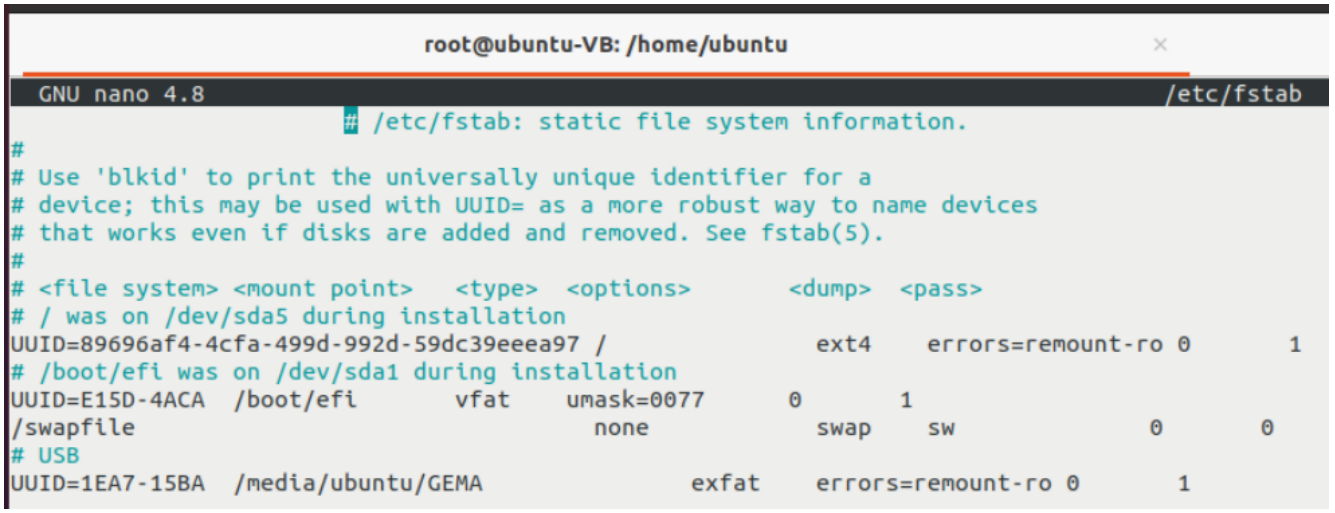
arrancar el sistema tengamos un directorio asignado para visualizar el contenido de un directorio de red.

Para finalizar, si queremos ver que la partición está correctamente montada ejecutamos el comando **mount** o **df**. (También el comando **lsblk** puede resultarnos útil)

Cuando queramos dejar de usar un sistema de archivos usaremos el comando umount. La sintaxis es: **umount dispositivo** o **umount directorio**.

Para montar un dispositivo USB seguiríamos las siguiente guía: https://geekland.eu/montar-la-memoria-usb-en-la-terminal/

Podemos añadir el dispositivo al fichero **/etc/fstab** indicando el **UUID** (el comando **blkid** devuelve el **UUID** de los distintos dispositivos de bloque).



4.2. Monitorización

Existen muchas herramientas que permiten monitorizar el sistema de ficheros entre las que destacan:

df. Muestra un resumen sobre el espacio libre que queda en los discos duros del sistema.

du. Muestra la cantidad de espacio que están utilizando los directorios o archivos específicos. Por ejemplo, si queremos ver el espacio que ocupa el directorio /datos en Megabytes ejecutamos: **$ du –ms /datos**

fsck, ej: fsck /dev/sda3. Permite comprobar el estado y reparar un sistema de ficheros (el dispositivo no puede estar montado para este proceso).

5. Permisos

Es muy importante establecer correctamente los permisos en el sistema de ficheros porque así evitarás usos indebidos o pérdidas de datos en el sistema.

Si ejecutas en un directorio el comando **ls –la** puedes ver los permisos del sistema de ficheros. Para cada fichero o directorio se muestran los siguientes datos:

- ✔ **Permisos**. Indica los permisos que tiene el fichero o directorio.
- ✔ **Usuario propietario**.
- ✔ **Grupo propietario**.

- ✔ **Tamaño del fichero o directorio**.
- ✔ **Fecha de creación o de la última modificación**.
- ✔ **Nombre**.

```
drwxr-xr-x   2 ubuntu gruposin   4096 ene 18 09:36 Documentos
drwxrwxr-x   2 ubuntu gruposin   4096 ene 18 09:36 Escritorio
drwxr-xr-x  14 ubuntu gruposin   4096 feb  8 14:10 geany-1.38
drwxrwxr-x   2 ubuntu gruposin   4096 feb  8 15:01 Imágenes
drwxrwxr-x   2 ubuntu gruposin   4096 ene 18 09:36 Música
drwxrwxr-x   2 ubuntu gruposin   4096 ene 18 09:36 Plantillas
drwxrwxr-x   2 ubuntu gruposin   4096 feb  1 14:17 practica_sin_archivos
drwxrwxr-x   2 ubuntu gruposin   4096 ene 18 09:36 Público
drwx------   3 ubuntu gruposin   4096 ene 18 11:54 snap
drwxrwxr-x+  4 ubuntu gruposin   4096 feb  1 13:51 test
drwxrwxr-x   3 ubuntu gruposin   4096 feb  1 13:39 test.1
```

El formato para establecer los permisos es (rwx) donde r indica **lectura**, w **escritura** y x indica **ejecución**. Si existe el permiso entonces se muestra su correspondiente letra y en el caso de que no exista ese permiso entonces aparece el carácter (-).

Por ejemplo, los permisos para el directorio **Documentos** son drwxr-xr-x. El carácter **d** indica que es un directorio. Luego se muestran tres grupos de caracteres (rwx) (r-x) (r-x) que permiten indicar los permisos del usuario propietario, del grupo propietario y de los demás usuarios.

El directorio **Imágenes** tiene todos los permisos para el usuario propietario **ubuntu** (rwx) e igualmente el grupo propietario **gruposin**; el resto de los usuarios tienen permisos de lectura y ejecución (r-x).

En un fichero el permiso de ejecución permite ejecutar un programa y en el caso de los directorios el permiso permite indicar que es posible entrar en ese directorio.

5.1. Establecer los permisos

Para definir los permisos de un fichero o directorio se emplea el comando chmod. Su sintaxis es:

 # chmod <modo> fichero

donde <modo> indica los permisos que le quiere asignar al fichero.

Por ejemplo, si queremos establecer los permisos rw- para el propietario y r-- para el resto, el comando que se debe utilizar es:

 # chmod 644 fichero

Con chmod se puede establecer los permisos con tres valores numéricos (por ejemplo, 664): el primer valor corresponde al **usuario propietario**, el segundo al **grupo propietario** y el tercer valor corresponde a **todos los demás usuarios** del sistema.

Cada permiso tiene una equivalencia numérica donde r vale **4**, w vale **2** y x vale **1**. De esta forma si tiene el valor **7** corresponde a (rwx), el valor **6** corresponde a (rw-), etcétera. El **propietario** de un fichero es aquel usuario que creó dicho fichero. GNU/Linux permite cambiar al propietario de cualquier fichero o directorio. Opcionalmente se puede cambiar también al grupo al que pertenece dicho fichero o directorio. Para ello se utiliza la orden **chown** que tiene la siguiente sintaxis:

chown <NombreUsuario> [.<NombreGrupo>] <fichero>...

donde <NombreUsuario> identifica el nuevo propietario de fichero o directorio. <NombreGrupo> el nuevo grupo y <fichero> identifica el fichero o directorio sobre el que se va a actuar.

El comando chmod permite otros modos de trabajo:
chmod [opciones] modo[,modo] fichero

Por ejemplo:

chmod o=rwx * Asigna permisos de lectura, escritura y ejecución para los usuarios "otros" a todos los archivos de la carpeta

chmod a=rwx fichero.txt Asigna todos los permisos a todos los usuarios para el archivo fichero.txt

También se pueden añadir o quitar permisos con los operadores + y -. Para ello se indica el tipo de usuario y el permiso que se resta o añade. Algo como esto:

chmod a+r,gu+w * Este comando asigna permisos de lectura a todos los usuarios y permisos de escritura al dueño del archivo y el grupo del dueño.

Por otro lado, para cambiar el **grupo** al que pertenece un directorio se utiliza **chgrp**. Su sintaxis es:

 # chgrp <NombreGrupo> <fichero>...

donde <NombreGrupo> identifica el nuevo nombre de grupo que se le va a asignar al fichero o directorio <fichero>. Se puede actuar sobre varios ficheros a la vez.

En los comandos **chmod**, **chown** y **chgrp** la opción **–R** significa que se establecen los permisos al directorio y a todos los datos que contiene.

Por ejemplo, el comando:

 # chmod 777 /datos -R

establece todos los permisos a la carpeta datos y a todo su contenido.

5.2. Permisos especiales

Existen 3 permisos especiales: SUID, SGID y Sticky. Los 2 primeros se aplican a ficheros ejecutables mientras que el último se aplica a directorios.

SUID (s), ej: chmod u+s touch: Aplicado sobre un archivo permitimos que cualquiera pueda usar el fichero como si fuera el usuario propietario, con los mismos privilegios. Por ejemplo, el comando **touch** es propiedad del usuario root, pero puede ser usado por el resto de usuarios con limitaciones. (No pueden modificar la fecha de modificación de los ficheros que estén más allá de la carpeta personal del usuario).

Si aplicamos el permiso SUID a touch, podremos crear ficheros vacíos en lugares donde antes no podíamos, como si fuéramos el root, de hecho, si realizamos un listado largo del fichero veremos que el usuario propietario es el root.

```
root@ubuntu-VB:/usr/bin# ls -l touch
-rwxr-xr-x 1 root root 100728 sep  5  2019 touch
root@ubuntu-VB:/usr/bin# chmod ug+s touch
root@ubuntu-VB:/usr/bin# ls -l touch
-rwsr-sr-x 1 root root 100728 sep  5  2019 touch
```

Si ejecutamos **touch -d "2022-02-06 19:15" p** …

```
ubuntu@ubuntu-VB:/$ ls -l p
-rw-rw-r-- 1 root root 0 feb  6 19:15 p
```

SGID (s): Similar a GUID, aunque aplicado al grupo propietario, es decir, si agregamos este permiso el fichero se ejecutará con los privilegios del grupo propietario del mismo.

Sticky (t): Permiso aplicado sobre carpetas. Con él aseguramos que un directorio sea accesible por todos los usuarios, pero cada uno de ellos solo puede borrar o cambiar los ficheros o subdirectorios creados por ellos mismos, no por los demás.

Una forma de activarlo para el directorio **test** sería con el comando: **chmod +t /test,** o bien **chmod 1776 test.** Se suele utilizar con el directorio **tmp**

```
ubuntu@ubuntu-VB:~$ ls -l / | grep tmp
drwxrwxrwt  20 root root      4096 feb  8 15:57 tmp
```

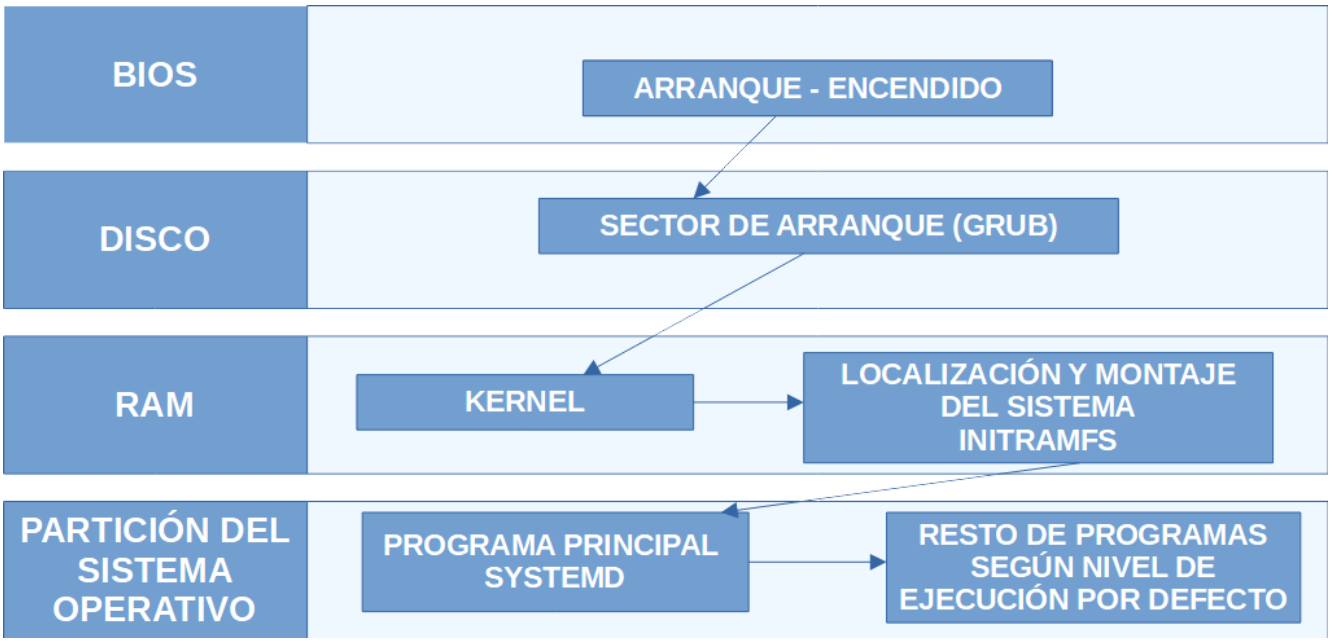
6. Arranque y parada

Una de las funciones de un administrador de sistemas es poder contestar en todo momento las siguientes preguntas: ¿qué sistema operativo se ejecuta en nuestro sistema? ¿qué servicios o programas se ejecutan en el sistema? ¿cuándo se ejecutan? Lógicamente, estos factores afectan muy estrechamente a la seguridad y al rendimiento del sistema.

Encendemos nuestro PC y la BIOS se encarga, mediante el POST, de observar irregularidades en el HW.

A partir de ahí, carga el gestor de arranque (que en Linux se llama GRUB) y en el caso de iniciar un sistema GNU/Linux accede al directorio /boot donde carga el kernel o núcleo del sistema operativo (carga de la **imagen del kernel** Linux) y ejecuta el proceso **systemd,** si se ha especificado como gestor de arranque en el archivo de configuración GRUB. A continuación, **Systemd** gestiona el proceso de administración de servicios y arranque mediante “targets”. Los ficheros “targets” en Systemd se utilizan para agrupar diferentes unidades de arranque e iniciar procesos de sincronización. El target es el equivalente al concepto de runlevel, es decir un conjunto de servicios que se ejecutan en determinadas circunstancias.

Para el caso de UEFI y dispositivos con sistema de partición GPT, hay una partición denominada ESP donde se almacenan los cargadores para todos los sistemas operativos instalados en el sistema. Estos cargadores pueden ser para Windows: bootmgfw.efi; para Linux: shimx64.efi, bootx64.efi, grubx64.efi;...)



Initramfs (*Initial Ramdisk File System*), sistema de archivos ram inicial (ramdisk)

http://recursostic.educacion.es/observatorio/web/es/software/software-general/534-el-gestor-de-arranque-gnu-grub

6.1. Gestor de arranque

El gestor de arranque es el encargado de iniciar cualquier sistema operativo que haya sido previamente instalado en el sistema (por ejemplo, Windows, GNU/Linux, FreeBSD). De forma tradicional el gestor de arranque utilizado en GNU/Linux era LILO, aunque actualmente el gestor de arranque más utilizado en la actualidad es **GRUB** (grub2).

Por defecto **GRUB** no muestra su menú en el booteo. Para ver **GRUB** durante el arranque hay que pulsar la tecla SHIFT derecho. Valores por defecto, como el tiempo de espera de Grub se pueden configurar desde el fichero **/etc/default/grub.**

GRUB (Grand Unified Bootloader) es un gestor de arranque que permite gestionar el inicio de nuestro equipo entre diferentes sistemas operativos.

Siempre que realices operaciones sobre el gestor de arranque es muy importante estar seguros de las opciones y parámetros introducidos, ya que es posible dañar el arranque del sistema. Aún así, siempre es posible utilizar alguna utilidad de recuperación del arranque, como por ejemplo Super GRUB Disk, de libre distribución. Esta herramienta además permite a usuarios avanzados realizar operaciones potencialmente peligrosas en el MBR (Master Boot Record o Registro de Arranque Principal) de forma segura.

6.2. Proceso de arranque y parada del sistema

El proceso de arranque tradicional del sistema Linux es manejado principalmente por el conocido proceso init. Muchas de las principales distribuciones de Linux han adoptado **Systemd** como el sistema de inicio predeterminado, entre ellas Ubuntu. El principal objetivo del desarrollo de **Systemd** fue el de reducir el tiempo de arranque y la sobrecarga computacional.

Systemd ofrece herramientas para identificar y solucionar problemas relacionados con el arranque o problemas de rendimiento. A continuación se enumeran algunos comandos útiles de **systemd-analyze**.

systemd-analyze time muestra el tiempo empleado en el kernel y el espacio de usuario normal.

```
ubuntu@ubuntu-VB:~$ systemd-analyze time
Startup finished in 3.052s (kernel) + 1min 1.381s (userspace) = 1min 4.434s
graphical.target reached after 1min 1.249s in userspace
```

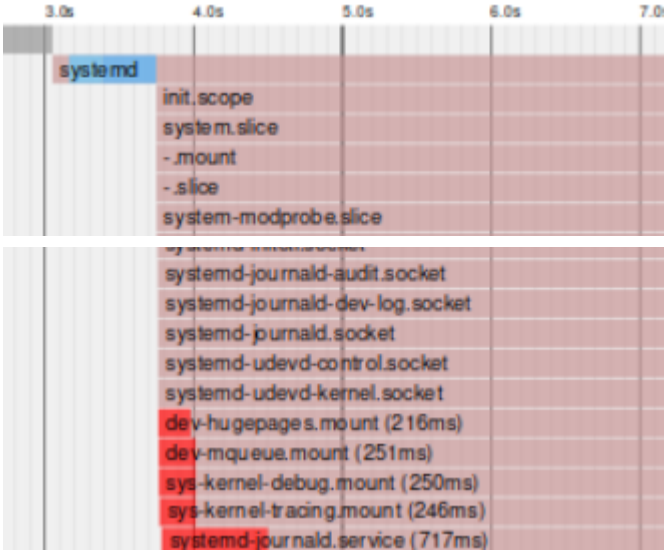
systemd-analyze blame imprime una lista de todas las unidades en ejecución, ordenadas por el tiempo que tarda en inicializarse, de esta manera puede tener una idea de qué servicios están tardando mucho en iniciarse durante el arranque.

```
ubuntu@ubuntu-VB:~$ systemd-analyze blame
44.494s plymouth-quit-wait.service
27.235s vboxadd.service
10.601s snapd.service
10.245s dev-sda5.device
 5.666s fwupd-refresh.service
 5.063s fwupd.service
 4.781s networkd-dispatcher.service
 4.456s NetworkManager-wait-online.service
 4.250s accounts-daemon.service
 4.185s dev-loop9.device
 4.155s dev-loop12.device
 4.129s dev-loop8.device
```

systemd-analyze critical-chain muestra una información más gráfica a modo de árbol con la cadena de unidades que tienen los mayores tiempos de carga.

Si usamos el parámetro **plot** y además redirigimos la salida obtendremos algo similar a la siguiente imagen.

```
ubuntu@ubuntu-VB:~$ systemd-analyze plot > analisisgrafico.svg
ubuntu@ubuntu-VB:~$
```



 # systemctl disable *nombre*.service

6.3. Servicios del sistema

Los **servicios** son aplicaciones que se ejecutan, en segundo plano, independientemente del usuario y ofrecen una determinada funcionalidad. Normalmente se asocia el término "servicio" sólo a servicios de red (por ejemplo, servidor web, servidor ftp) pero existen servicios que ofrecen todo tipo de funcionalidades (gestionan las conexiones de red, monitorizan el sistema, comprueban las actualizaciones y seguridad del sistema, permiten utilizar el hardware del equipo, etcétera).

El **administrador de servicios** permite establecer los servicios que se van a ejecutar al iniciar el sistema, y permite parar, ejecutar o reanudar los servicios que se ejecutan actualmente en el sistema.

Iniciar y detener servicios

Para iniciar un servicio **systemd**, ejecutar instrucciones en el archivo de la unidad del servicio, utilice el comando **start**. Si está ejecutando como usuario non-root, tendrá que usar **sudo**, ya que esto afectará al estado del sistema operativo.

$ sudo systemctl start application.service

systemd sabe buscar los archivos ***.service** para los comandos de administración de servicio, de forma que el comando podría escribirse fácilmente así:

$ sudo systemctl start application

Aunque puede usar el formato anterior para la administración general, para mayor claridad, usaremos el sufijo **.service** para el resto de los comandos, con el objetivo de ser explícitos sobre el destino en el que estamos operando.

Para detener un servicio que se esté ejecutando actualmente, puede usar el comando **stop**:

$ sudo systemctl stop application.service

Para comprobar el estado de un servicio en su sistema, puede usar el comando **status**:

$ sudo systemctl status application.service

6.4. Procesos

En los sistemas GNU/Linux se ejecutan una gran cantidad de servicios que permiten realizar una determinada actividad en el sistema. Cada servicio o demonio

consiste en uno o más procesos que se ejecutan en el equipo. Además de los procesos vinculados a servicios, en el sistema se encuentran los procesos que ejecuta un usuario. Por ejemplo, un editor de textos, un navegador Web, etcétera.

Para ver los procesos que se ejecutan en el equipo hay que ejecutar el comando **ps**. Para cada proceso se muestra su identificador (PID), terminal donde se ejecuta (TTY), tiempo de uso de CPU (TIME) y el comando que ejecuta (CMD).

Si quieres ver todos los procesos que se ejecutan en el sistema con información detallada, utiliza la **opción -aux**: **# ps –aux**

Si deseas eliminar un proceso que se está ejecutando en el sistema puedes utilizar el comando **kill** de la siguiente forma:

 # kill -9 <ID del proceso>

Otra aplicación que permite ver los procesos que se ejecutan en el sistema es **top**. **Top** es una aplicación que, en tiempo real, informa sobre la actividad del sistema. Proporciona información sobre la carga del sistema operativo, grado de utilización de la CPU, memoria y swap, y los procesos que se encuentran en ejecución.

En ocasiones podemos usar **xkill.** Este comando permite que, una vez ejecutado, se haga clic sobre alguna de las aplicaciones abiertas provocando el fin de ejecución del proceso.

Jobs. Este comando visualiza aquellos procesos que estén ejecutándose en segundo plano.

Demonios (Daemon) son los procesos que están cargados en memoria, ejecutándose en segundo plano y contínuamente a la espera para ofrecer un determinado servicio. Por ejemplo: los servidores de http, de correo electrónico, etc.

6.5. Programación de tareas

La programación de tareas permite programar la ejecución de un determinado programa en un momento determinado. Por ejemplo, se puede programar una copia de seguridad, enviar un fichero, comprobar la seguridad del sistema, enviar un informe, etcétera.

Antes de programar las tareas hay que comprobar que el servicio **cron** se encuentra en ejecución mediante el comando:

 # service cron status

Para modificar el fichero de configuración de cron, ejecuta el comando:

 # crontab –e

y aparece un fichero con el siguiente formato:

```
PATH=/bin
0 0 * * * /root/comprobar_seguridad.sh
0 0 1 * * /root/copia_seguridad.sh
```

```
La sintaxis de las tareas programadas es:
# .---------------- minuto (0 - 59)
# | .------------- hora (0 - 23)
# | | .---------- día del mes (1 - 31)
# | | | .------- mes (1 - 12) o jan,feb,mar,apr ... (los meses en inglés)
# | | | | .---- día de la semana (0 - 6) (Domingo = 0 o 7) OR sun,mon,tue,wed,thu,fri,sat (los días en inglés)
# | | | | |
# | | | | |
 * * * * * Comando a ejecutar
```

En el ejemplo anterior se ejecuta el script *comprobar_seguridad.sh* **todos los días a las 0:00h** y se ejecuta *copia_seguridad.sh* **el primer día de cada mes**.

Otra forma de poder programar tareas es guardar el script que quiere ejecutar en las siguientes carpetas de configuración de cron:

```
/etc/cron.hourly # Ejecuta el script cada hora
/etc/cron.daily # Ejecuta el script diariamente
/etc/cron.weekly # Ejecuta el script semanalmente
/etc/cron.monthly # Ejecuta el script mensualmente
```

Existe un archivo crontab «principal» **/etc/crontab**, que será gestionado por el administrador del equipo, pero además cada usuario del sistema tiene su propio archivo.

Una ventaja muy interesante que permite **crontab** es que cada vez que se ejecuta la tarea manda un correo electrónico con el resultado de la ejecución de dicha tarea.

https://blog.ahierro.es/programar-tareas-en-ubuntu-con-cron/

6.6. Reinicio y parada del sistema

El proceso de parada y reinicio del sistema se puede realizar de forma gráfica o por

terminal. Para hacerlo de forma gráfica tan sólo hay que pulsar en el botón de apagar que se encuentra en la esquina superior derecha y en el menú que aparece seleccionar la operación a realizar.

Además, puedes utilizar comandos específicos para apagar el equipo como **halt** o **shutdown**, o se puede reiniciar el equipo ejecutando **reboot**. (La sintaxis de shutdown permite indicar la hora y minuto del apagado o reinicio: shutdown -h [TIME])

```
ubuntu@ubuntu-VB:~$ shutdown -r +1
Shutdown scheduled for Sun 2022-02-06 18:08:56 CET, use 'shutdown -c' to cancel.
```

7. Monitorización del sistema

Para conocer el comportamiento del sistema es necesario obtener información sobre las prestaciones de los diferentes subsistemas que lo componen. En GNU/Linux se dispone, por una parte, de una serie de comandos que proporcionan datos sobre el rendimiento del hardware y del sistema operativo y, por otra parte, de una aplicación cliente-servidor que registra los eventos que suceden en el equipo (**syslog**).

Los objetivos de las herramientas de monitorización del sistema operativo son:

- Ofrecer el grado de disponibilidad adecuado.
- Facilitar al administrador la revisión y análisis de los datos de rendimiento para que pueda comprobar que se cumplen los niveles de servicio esperados.
- Supervisar el sistema permanentemente y avisar al administrador cuando se produzca un fallo de disponibilidad.
- Minimizar el tiempo de detección de incidentes

7.1. Herramienta gráfica

Aplicación ***Monitor del sistema*** (***gnome-system-monitor***) donde encontramos 3 pestañas: Recursos, Procesos y Sistema de Archivos.
Cuando nuestro ordenador no ofrece los niveles de rendimiento que esperamos de él, podemos observar los datos que nos ofrece el Monitor del sistema. En muchos casos, podremos resolver el problema ampliando la memoria o sustituyendo la tarjeta de red o el disco duro.

La solapa *Recursos* muestra información sobre tres subsistemas distintos:

CPU: Se muestra un gráfico con el grado de ocupación del procesador durante el último minuto y el porcentaje de uso actual. Si disponemos de varios procesadores (o varios núcleos) se mostrará la información diferenciada por colores para cada uno de ellos. Por

ejemplo, en la imagen siguiente comprobamos que el procesador dispone de un solo núcleo y que está ocupado al 34% de su capacidad.

***Memoria*:** Como en el caso de la *CPU* se muestra la ocupación de la memoria tanto en el último minuto como en el momento actual. En el mismo gráfico se representa tanto la *memoria principal* (*RAM*) como la *memoria de intercambio*. Así, en la imagen superior comprobamos que el equipo dispone de 1,9 GiB de *memoria RAM* y tiene ocupados 1,2 GiB, lo que supone un 61,5%. Por su parte, disponemos de 687,5 MiB de *memoria de intercambio* y la tenemos ocupada un 0,2%.

***Red*:** De nuevo se muestra un gráfico con el nivel de ocupación de la conexión de red durante el último minuto. También se muestra la cantidad de información enviada y recibida y el ancho de banda ocupado en el instante actual. En la imagen vemos que en ese momento en particular, la red está enviando 59 bytes/s y ha recibido un total de 3,3 MiB, por otra parte se está enviando información a 59 bytes/s y se han enviado un total de 180,6 KiB.

Además, la solapa *Sistema de archivos*, ofrece información sobre los sistemas de archivos que tengamos montados en estos momentos. Ahí podemos consultar el tipo de sistema de archivos, su capacidad de almacenamiento, su nivel de ocupación, etc.

En cuanto a los procesos, aunque Linux no tiene los mismos problemas que Windows, ya que todos los que no se necesitan están en modo «zombie», siempre es mejor intentar tener los menos procesos posibles cargados en memoria, aunque estén inactivos, para evitar un uso innecesario de RAM, y otros posibles problemas.

Controlar los procesos o servicios en Linux es una tarea de lo más complicada, no apta para usuarios sin experiencia. Esto se debe a que la mayoría de ellos están incluidos, o bien dentro del propio arranque del sistema, o en el núcleo. Por lo tanto, no se recomienda modificarlo si no sabemos muy bien lo que estamos haciendo. Modificar estos elementos en

Linux sin saber son sinónimo de Kernel Panic, error crítico en el sistema y, probablemente, tener que reinstalarlo todo de nuevo.

La mejor forma de optimizar el funcionamiento de Linux, y eliminar todos esos procesos y todas esas dependencias es, directamente, eliminar los programas que no necesitemos

7.2. Herramientas básicas.

Según el tipo de información que presentan, los comandos se pueden clasificar en:

- ✔ Procesos. Muestra información sobre los procesos que se están ejecutando en el sistema.
- ✔ Almacenamiento. Proporcionan información sobre la entrada y salida al subsistema de almacenamiento.
- ✔ Memoria. Proporcionan información sobre el espacio de memoria real y swap.
- ✔ Red. Facilitan estadísticas de uso de las interfaces de red.
- ✔ Polivalentes. Muestran información sobre distintos subsistemas del equipo.

Herramientas básicas de monitorización en GNU/Linux	
Procesos	
ps	Muestra el estado de los procesos que se están ejecutando en el equipo.
Almacenamiento	
df	Muestra el espacio libre del sistema de ficheros.
du	Muestra el espacio ocupado a partir de un determinado directorio.
Memoria	
free	Proporciona información relativa a la cantidad de memoria física, espacio de swap libre y usado por el sistema operativo, estado de los buffers y memoria caché utilizada por el núcleo.
pmap	Proporciona información referente a la utilización de la memoria por parte de un determinado proceso.
Red	
ifstat	Muestra la estadística de tráfico de entrada y salida de las interfaces de red.
iftop	Muestra las conexiones de red de un equipo.
iptraf	Es una completa herramienta que permite mostrar las estadísticas de red en tiempo real.
netstat	Proporciona estadísticas e información de estado sobre tablas de rutas, interfaces de red, conexiones establecidas, etcétera.
ping	Permite comprobar el estado de una conexión.
traceroute	Permite obtener el camino que se sigue un paquete para establecer una comunicación con un destinatario, es decir, los routers que se atraviesan.

En la siguiente tabla se muestra un resumen de las herramientas básicas de monitorización en GNU/Linux.

Polivalentes	
dstat	Permite realizar estadísticas de CPU, utilización de disco, red, paginación y estado del sistema.
iostat	Permite ver la carga de CPU y del disco duro.
top	Informa en tiempo real sobre la actividad del sistema. Proporciona información sobre la carga del sistema operativo, grado de utilización de la CPU, memoria y swap, y los procesos que se encuentran en ejecución.
vmstat	Muestra información sobre los procesos que se están ejecutando en el equipo, la memoria, las operaciones de entrada y salida a disco, y la utilización de la CPU. Es una aplicación clásica en los sistemas.
who	Permite ver de forma resumida el tiempo que lleva activo el sistema (uptime), la carga del sistema y la actividad de los usuarios que se encuentran conectados al sistema
xosview	Es una aplicación gráfica que proporciona información sobre el uso de CPU, memoria, cantidad de carga del sistema, red, interrupciones y swap en espacio de usuario.

Ejecuta **top** para ver el estado del sistema mientras ejecutas otra aplicación como firefox, por ejemplo.
Ejecuta iptraf para ver la actividad de la red mientras descargas un fichero de internet.

7.3. Directorio /proc

El núcleo de Linux almacena información relativa a su funcionamiento en archivos situados en el directorio **/proc**, de tal forma que, para analizar el comportamiento de un sistema, también se puede recurrir a la consulta de los archivos de este sistema de ficheros. De hecho, prácticamente todas las herramientas analizadas obtienen sus datos de esta fuente.

Un ejemplo de la información que reside en /proc es:
✔ Estado de la memoria disponible en el fichero /proc/meminfo.
✔ Sistema de comunicaciones en /proc/net.
✔ Datos referentes a un proceso que se encuentran en un subdirectorio del estilo a /proc/pid_del_proceso.
✔ Etcétera.

7.4. Archivos de registro (syslog)

Hasta ahora se ha visto cómo ver el estado actual del sistema. Pero sin duda es muy importante saber lo que ha pasado en el servidor. Existen muchos motivos por los que se pueden generar mensajes. Entre los más frecuentes se encuentran los fallos del servidor (por ejemplo, problema de hardware, fallo en un servicio), de autentificación (por ejemplo, fallo en la autentificación de un usuario) o por la utilización de un servicio (por

ejemplo, petición de un cliente de una página web). Estos mensajes se encuentran en los **archivos de registro o archivos log** ubicados en el directorio **/var/log**. Por ejemplo, muchos mensajes son guardados en los ficheros /var/log/syslog o en el /var/log/messages. Pero si un servicio genera muchos mensajes lo normal es que sean escritos en un fichero o carpeta separada como lo hace apache (/var/log/httpd) o el servidor de correo (/var/log/mail).

El registro de todos los mensajes del sistema lo realiza el servicio **syslogd** (o rsyslogd), el cual no es exclusivo de los servicios del sistema sino que nosotros también podemos registrar nuestros propios mensajes usando **syslog**.

8. Copias de seguridad

Existen muchas herramientas que permiten realizar copias de seguridad del sistema. Estas herramientas se pueden clasificar en tres categorías: herramientas o comandos básicos, herramientas avanzadas de copias de seguridad y herramientas de clonación de sistemas.

La forma más habitual de realizar las copias de seguridad es utilizando los **comandos básicos** que proporciona el sistema (por ejemplo: **dump/restore, tar**). Con los comandos básicos se pueden realizar copias de seguridad de un equipo de forma individual.

Además, existen herramientas avanzadas que permiten centralizar y administrar todas las copias de seguridad de un sistema en un único servidor. Un ejemplo de este tipo de herramientas es **Amanda**, que permite centralizar todas las copias de seguridad de los sistemas Windows y GNU/Linux de una empresa en un único servidor.

Otra forma muy útil de realizar copias de seguridad de sistemas enteros es la clonación de discos duros. La **clonación de discos duros** permite realizar una copia exacta de un disco duro o partición para poder restaurarlo en otro equipo de características similares. Este tipo de herramientas es muy útil en el caso de que quieras realizar una copia exacta de un servidor o restaurar muchos equipos con la misma configuración como, por ejemplo, un aula de informática. En la tabla se muestran las herramientas de clonación de sistemas más importantes, destacando la herramienta **Clonezilla** que se verá más adelante.

Herramientas de clonación de discos	
Clone Maxx.	http://www.pcinspector.de/clonemaxx/info.htm?language=1
Clonezilla.	http://www.clonezilla.org/
Dubaron DiskImage.	http://www.dubaron.com/diskimage/
g4U.	http://www.feyrer.de/g4u/
NFGdump.	http://sourceforge.net/projects/nfgdump/
Norton Ghost.	http://es.norton.com/ghost
Partition Saving.	http://www.partition-saving.com/
Partimage.	http://www.partimage.org/
WinDD.	http://sourceforge.net/projects/windd/

8.1. Comandos básicos

Aunque muchas distribuciones de UNIX/Linux ofrecen sus propias herramientas para realizar copias de seguridad de todo tipo, casi todas estas herramientas suelen presentar un grave problema a la hora de recuperar ficheros cuando se trata de software propietario, por lo que si deseas restaurar total o parcialmente ficheros necesitas el propio programa para hacerlo. En determinadas situaciones, esto no es posible o es muy difícil. Imagina un departamento que dispone de sólo una estación Silicon Graphics y pierde todos los datos del sistema. Si has utilizado herramientas propias del sistema, necesitarás otra estación con el mismo sistema operativo para poder restaurar estas copias, lo que obviamente puede ser problemático. Por este motivo, muchos administradores utilizan herramientas estándar para realizar las copias de seguridad de sus máquinas. Estas herramientas suelen ser tan simples como: dump/restore, tar, dd,gzip, rsync, etcétera. Para mejorar las prestaciones de dichas herramientas se realizan, y programan, scripts para que se realicen las copias de forma automática.

A continuación, se van a ver los comandos más utilizados para realizar copias de seguridad en sistemas GNU/Linux.

8.1.1. El comando tar

La utilidad **tar** (*Tape ARchiver*) es una herramienta de fácil manejo disponible en todas las versiones de UNIX/Linux que permite copiar ficheros individuales o directorios completos en un único fichero. Oficialmente fue diseñada para crear ficheros de cinta (esto es, para transferir ficheros de un disco a una cinta magnética y viceversa), aunque en la actualidad casi todas sus versiones pueden utilizarse para copiar a cualquier

dispositivo o fichero, denominado “contenedor”.

En la siguiente tabla se muestran las opciones de tar más habituales. Algunas de ellas no están disponibles en todas las versiones de tar, por lo que es recomendable consultar la página del manual de esta orden antes de utilizarla.

Opciones de la orden tar	
Opción	**Acción**
c	Crea un contenedor.
x	Extrae ficheros de un contenedor.
t	Testea los ficheros almacenados en un contenedor.
r	Añade ficheros al final de un contenedor.
v	Modo verbose.
f	Especifica el nombre del contenedor.
z	Comprime o descomprime el fichero.

En primer lugar, debe saber cómo crear contenedores con los ficheros deseados. Por ejemplo, para copiar el directorio /home/ en el fichero /root/copia.tgz hay que ejecutar el siguiente comando:

 # tar cvf /root/copia.tgz /home/

La opción “v” no es necesaria, pero es útil para ver el proceso de empaquetamiento del fichero. En muchas situaciones también resulta útil comprimir la información guardada (tar no comprime, sólo empaqueta) por lo que hay que utilizar las opciones “cvfz”.

En lugar de indicar un único directorio con todos sus ficheros y subdirectorios es posible especificar múltiples ficheros (o directorios). Por ejemplo, la siguiente orden crea el fichero /tmp/backup.tar, que contiene /etc/passwd y /etc/hosts*.

 # tar cvf /tmp/backup.tar /etc/passwd /etc/hosts*

Para recuperar los ficheros guardados en un fichero tar se utilizan las opciones “xvf” (o “xvfz” si se ha utilizado compresión con gzip). Puedes indicar el fichero o ficheros a extraer; si no lo haces se extraerán todos los ficheros. A continuación, puedes ver un ejemplo:

 # tar xvf /tmp/backup.tar /etc/passwd

En el ejemplo anterior, la restauración se ha realizado desde el directorio de trabajo, creando en él un subdirectorio /etc con los ficheros correspondientes en su interior.

Un fichero con extensión “.tar” se llama empaquetado ya que el fichero ocupa lo mismo que su contenido. Mientras que un fichero con extensión “.tar.gz” o “.tgz” está

comprimido y ocupa menos espacio que su contenido.

8.1.2. El comando dd

El comando **dd** permite realizar copias exactas (bit a bit) de discos duros, particiones o ficheros. La sintaxis de dd es la siguiente:

 # dd if=fichero_origen of=fichero_destino

Antes de duplicar un disco duro debes saber los discos duros que tiene el sistema por lo que tienes que ejecutar el comando:

 # fdisk –l

Por ejemplo, si desea clonar el disco duro que se encuentra en /dev/sda en el disco duro /dev/sdb ejecuta el comando:

 # dd if=/dev/sda of=/dev/sdb

8.1.3. rsync

rsync es una aplicación para sistemas GNU/Linux que permite sincronizar carpetas de forma incremental y permite trabajar con datos comprimidos y cifrados. Mediante una técnica que se conoce como de ***delta encoding*** (*"Delta compression*" o "differential compression"), permite sincronizar archivos y directorios entre dos máquinas de una red o entre dos ubicaciones en una misma máquina, minimizando el volumen de datos transferidos por la red.

Al sincronizar las carpetas de dos equipos los datos se envían a través de SSH por lo que es posible configurar el servidor SSH para que no solicite la contraseña a la hora de sincronizar las carpetas.

Si deseas sincronizar dos carpetas locales ejecuta:

$ rsync –avz /carpeta_origen /carpeta_destino

donde se sincroniza el contenido de la /carpeta_origen en la /carpeta_destino. De forma análoga si quieres sincronizar las carpetas de dos equipos ejecuta:

$ rsync –avz /carpeta_origen dirección_IP:/carpeta_destino

Lógicamente, tanto el origen como el destino puede ser un equipo remoto siguiendo la sintaxis anterior.

8.1.4. Backups sobre CD-ROM

Cada vez es más común realizar copias de seguridad sobre discos compactos. Para poder grabar datos en un CD o DVD primero es necesario crear la imagen ISO (el "molde" del futuro CD ROM). Una vez creada la imagen se graba en el disco utilizando un software de grabación.

Por ejemplo, si quieres realizar una copia del directorio /home/, en primer lugar, ejecutarás **mkisofs** para crear una imagen con todos los ficheros y subdirectorios de los usuarios:

 # mkisofs -o /root/imagen.iso /home/

Con esta orden se ha creado una imagen ISO denominada /root/imagen.iso y que contiene toda la estructura de directorios de /home/.

Una vez creada la imagen hay que grabarla en un CD-ROM, por ejemplo, mediante **cdrecord**:

 # cdrecord /root/imagen.iso

Con esta orden el sistema detecta la grabadora de CD/DVD disponible en el sistema y realiza la grabación de la imagen ISO. La mejor forma de automatizar una copia de seguridad es crear un script con todos los pasos de la copia de seguridad y programar su ejecución con crontab.

8.2. Herramientas gráficas

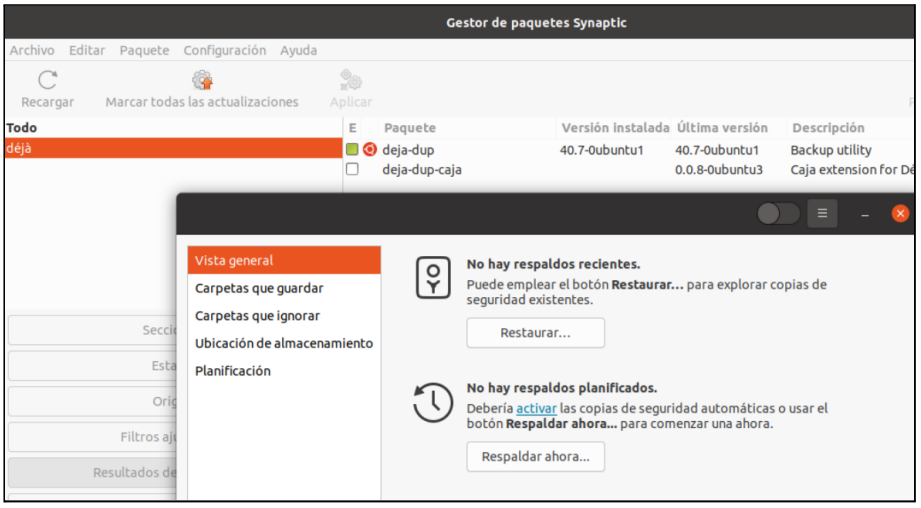
Además de realizar las copias de seguridad por comandos puede realizar la copia de seguridad del sistema mediante herramientas gráficas. Las herramientas más utilizadas son:

Déjà-Dup es una aplicación para realizar copias de seguridad de forma sencilla e intuitiva. Entre sus características más importantes destaca la posibilidad de cifrar los datos para asegurar la privacidad, programación de las copias, permite almacenar las copias en diferentes destinos (por ejemplo, el servidor externo, local...).

La instalación de Déjà-Dup se puede realizar a través de la herramienta Agregar/quitar software o ejecutando en el terminal el siguiente comando:

 # apt-get install deja-dup

Para iniciar la aplicación de copias de seguridad puedes ejecutar el comando deja-dup en un terminal, o ir al menú Aplicaciones>Herramientas del sistema y ejecutar Herramienta de respaldo Déjà-Dup. Una vez iniciada la aplicación puede realizar dos acciones principales: Respaldar (*realizar*) o restaurar copias de seguridad.



http://somebooks.es/copias-de-seguridad-integradas-en-ubuntu-20-04-lts-parte-i/

Clonezilla es la distribución LiveCD más potente y utilizada en la actualidad que permite realizar la clonación y restauración de sistemas. Clonezilla está licenciada bajo GPL y entre sus características más importantes destacan:

- Permite la clonación y restauración de particiones o de discos duros completos.
- Utiliza diferentes sistemas de ficheros como FAT32, NTFS y ext3 por lo que permite trabajar con cualquier instalación GNU/Linux o Windows.
- Permite realizar y restaurar las copias de seguridad utilizando diferentes medios como, por ejemplo, discos duros locales, servidores Samba, servidores SSH, llaveros USB, etcétera.
- Es fácil y sencilla de utilizar.

Para ejecutar CloneZilla podemos descargar la imagen ISO y grabarla en un CD o en una memoria USB y configurar el arranque del equipo desde la unidad de CD o USB dependiendo lo que hayamos elegido. Podremos clonar el disco o partición que queramos, sin importar el sistema instalado en la misma.

Inicias el LiveCD en el equipo que deseas clonar y en el menú de arranque seleccione la opción cuya resolución se adapte mejor a nuestras necesidades. A

continuación se inicia el asistente que te guía para poder clonar o restaurar una copia del equipo.

En la página oficial de Clonezilla puedes descargar la imagen ISO. http://clonezilla.org/

9. Otras Herramientas

• **Los múltiples comandos de utilidad UNIX básicos**: grep, awk, sed, find, diff, gzip, bzip2, cut, sort, df, du, cat, more, file, which...

• **Los editores**, imprescindibles para cualquier tarea de edición, cuentan con editores como **Vi**, muy utilizado en tareas de administración por la rapidez de efectuar pequeños cambios en los ficheros. **Vim** es el editor compatible Vi, que suele incorporar GNU/Linux. Permite sintaxis coloreada en varios lenguajes. **Emacs**, editor muy completo, adaptado a diferentes lenguajes de programación (sintaxis y modos de edición), dispone de un entorno muy completo y de una versión X denominada xemacs. Joe es un editor compatible con Wordstar. Y muchos otros...

• **Lenguajes de tipo script**, útiles para administración, como **Perl**, muy adecuado para tratamiento de expresiones regulares y análisis de ficheros (filtrado, ordenación, etc.); **PHP**, lenguaje muy utilizado en entornos web; **Python**, otro lenguaje que permite hacer prototipos rápidos de aplicaciones...

• **Herramientas de compilación y depuración de lenguajes de alto nivel**:
GNU **gcc** (compilador de C y C++ entre otros), **gdb** (depurador), **xxgdb** (interfaz X para gdb), **ddd** (depurador para varios lenguajes).

10. Directivas Ubuntu

10.1. Seguridad de cuentas de usuario

Hay un comando que permite modificar el comportamiento de la contraseña de un usuario:

chage [opciones] login

Algunos de los parámetros de este comando son:

- d: días. Cuenta los días desde el 1-1-1970 transcurridos desde que se cambió la contraseña por última vez. También se puede especificar con el formato AAAA-MM-DD.

- E fecha: Modifica la fecha en que la cuenta del usuario expirará y será bloqueada.
- I días: Modifica cuántos días pueden permanecer una cuenta con contraseña expirada antes de ser bloqueda.
- -M días: Modifica el número máximo de días durante los cuales es válida la contraseña de usuario.
- -m días: Modifica el número mínimo de días entre cambio de contraseña
- -l: Muestra la información de la cuenta de usuario en relación al comportamiento de sus contraseñas.

10.2. Seguridad de contraseñas en Linux

Será necesario instalar una librería que nos permita controlar la complejidad de las contraseñas. La librería es **libpam-cracklib y** se encarga de comprobar que la contraseña que vas a introducir no forma parte de un diccionario, es decir, que no sea una palabra sencilla. También comprobará que la contraseña no sea igual a otra que hayas utilizado anteriormente, es decir, no nos dejará reutilizar las contraseñas que ya hayamos utilizado anteriormente…

sudo apt install libpam-cracklib

Una vez instalada la librería, la configuración de la complejidad de la contraseña la realizaremos sobre el archivo de texto **/etc/pam.d/common-password**.

Un ejemplo de política de claves podría ser la siguiente:

```
password requisite pam_cracklib.so retry=3 minlen=12 difok=3 ucredit=-3
lcredit=-3 dcredit=-3 ocredit=-3
```

Con la política anterior tendremos:

- Longitud mínima de la contraseña 12 caracteres.
- Si cambiamos la clave, como mínimo debe haber tres diferencias.
- 3 caracteres en mayúscula, 3 en minúscula, 3 dígitos y 3 símbolos como mínimo

Ahora podemos probar los cambios ejecutando forzando a los demás usuarios al cambio de contraseña en el siguiente inicio de sesión ejecutando:

passwd -e USUARIO

https://www.redeszone.net/tutoriales/seguridad/configurar-politica-contrasenas-debian/