

U2 Introducción al Sistema Operativo Linux. **Parte II**

Implantación de Sistemas Operativos

Índice

- Procesos
- Instalación de aplicaciones
- Gestión de dispositivos
- Redes

Procesos en Linux

- El kernel o proceso del sistema se ejecuta desde que arrancamos el sistema y tiene un identificador de proceso o **pid=1**. Los restantes procesos que arrancan son hijos de ese proceso.
 - En los primeros Linux basados en Unix era **System V**
 - Posteriormente se adoptó una versión asíncrona: **Upstart** y actualmente es **Systemd** (d de daemon)
- Una vez cargado el núcleo se inician los **servicios del sistema operativo** en un orden determinado para llevarlo a diferentes estados.
 - En systemd estos estados se conocen como **targets**
 - Los targets sustituyen a los anteriores niveles de ejecución (**runlevel**)
- Los **servicios o demonios** son procesos que se ejecutan de forma continua sin intervención del usuario.



Targets vs runlevels

SysVinit	SystemD	Sirve para...
0	runlevel0.target, poweroff.target	Detener el Sistema
1	runlevel1.target, rescue.target	Modo mono-usuario, especial para administración
2	runlevel2.target, multi-user.target	Modo multi-usuario, definido por el usuario y por defecto igual al 3
3	runlevel3.target, multi-user.target	Modo multi-usuario, se usa en modo texto por completo
4	runlevel4.target, multi-user.target	Modo multi-usuario, definido por el usuario, por omisión es igual al 3
5	runlevel5.target, graphical.target	Modo multi-usuario con interfaz gráfica. Usualmente todos los servicios del nivel de ejecución 3 sumando el login gráfico.
6	runlevel6.target, reboot.target	Reboot
Emergency	emergency.target	Shell de emergencia

Procesos en Linux

- Consulta el target por defecto en el que estás con `systemctl get-default`
- Existe el comando `runlevel` por compatibilidad con el sistema anterior, que nos dice la equivalencia con el nivel de ejecución en el que estamos
- Podemos consultar los target existentes con
`ls -al /lib/systemd/system/runlevel?.target`
- El comando `ps tree` muestra el árbol de dependencia de procesos. En siguiente ejemplo, vemos que systemd tiene 2 procesos hijos idénticos (abrt-watch-log) y que accounts-daemon genera 4 hilos o threads

```
systemd---+-NetworkManager--chclient
           |-2*[abrt-watch-log]
           |-abrttd
           |-accounts-daemon--4*[{accounts-daemon}]
```

- Los ficheros de configuración de inicio del sistema se encuentran en **`/etc/systemd/system`** (antes `/etc/inittab`)

Procesos en Linux



- **Uptime** muestra cuánto tiempo lleva en marcha el sistema, el número de terminales abiertas y la carga media del sistema (nº medio de procesos en 1, 5 y 15min)
- **Top**: muestra información de forma dinámica de los procesos activos ordenados en función de su uso de memoria RAM (por eso cambian).
- **Ps**: muestra los procesos activos y admite distintos parámetros:
 - -e muestra todos los procesos existentes en memoria (por defecto lanzados desde nuestro terminal)
 - -s muestra los procesos que han sido ejecutados por el usuario activo en ese terminal
 - -u muestra los procesos con información sobre uso de CPU, de memoria, etc.
 - -a muestra todos los procesos de todos los usuarios
 - -x lista todos los procesos de todas las terminales y usuarios
 - -l muestra información sobre el uid y el valor de nice
 - -f formato completo

Interpretar salida PS

- Algunos de los datos más importantes que nos muestra por cada proceso son:
 - USER (usuario que lanzó el programa),
 - PID (identificador del proceso),
 - PPID (id del proceso padre),
 - %CPU (porcentaje entre el tiempo usado realmente y el que lleva en ejecución),
 - %MEM (fracción de memoria consumida),
 - VSZ: tamaño virtual del proceso (código+datos+pila), en KB.
 - RSS: memoria real usada, en KB.
 - TTY: terminal asociado con el proceso.
 - STAT: estado del proceso:
 - R: en ejecución o listo.
 - N: prioridad baja (valor mayor que 0).
 - S: durmiendo.
 - <: prioridad alta (valor menor que 0).
 - T: parado.
 - Z: proceso zombie.
 - D: durmiendo ininterrumpiblemente (normalmente por E/S de corta duración).
 - l: tiene multi-threads.
 - +: proceso ejecutándose en primer plano.

Procesos en Linux

- La ejecución de terminales de comandos o shell da lugar a procesos a los que se asigna un pid.
- Cada programa que se ejecuta a partir de un terminal pasa a ser hijo de ese terminal (por lo que si muere el shell muere el proceso que se lanzó desde una instrucción en ese shell).
- Es posible determinar que en caso de muerte del padre, el proceso hijo pase a depender de Systemd.
- Cuando se mata o muere un proceso en Linux que tiene hijos, por defecto, mueren sus procesos hijos.
- **Nohup** se emplea como prefijo para ejecutar una instrucción e indica que si muere el proceso padre del proceso que se ejecuta, pasa a ser hijo del sistema
 - Ej: nohup xeyes
 - Cuando ejecutamos nohup, si no ponemos nada la salida del proceso se escribe en un fichero llamado nohup.out. Si queremos, podemos especificar el nombre del fichero en el que queremos que se escriba con una redirección:
 - Ej: nohup xeyes > resultado.txt

Procesos en Linux

- **who** muestra los usuarios que están conectados y desde cuando operan. Con **-r** también muestra el runlevel al que operan
- **w** muestra quién está conectado al terminal y qué está haciendo
 - JCPU es el tiempo usado por todos los procesos asociados al terminal
 - PCPU tiempo de CPU usado por el proceso actual
- **Kill**: envía una señal a un proceso (consulta la ayuda de signal). Por defecto envía la 15 para terminar el proceso que permite avisarle para ejecutar una rutina de terminación si la hubiera.
 - Ej kill 1234
- **killall** permite matar varios procesos a la vez con otros criterios
 - Ej killall nautilus, cierra varias ventanas de nautilus sin especificar sus pid's individuales.

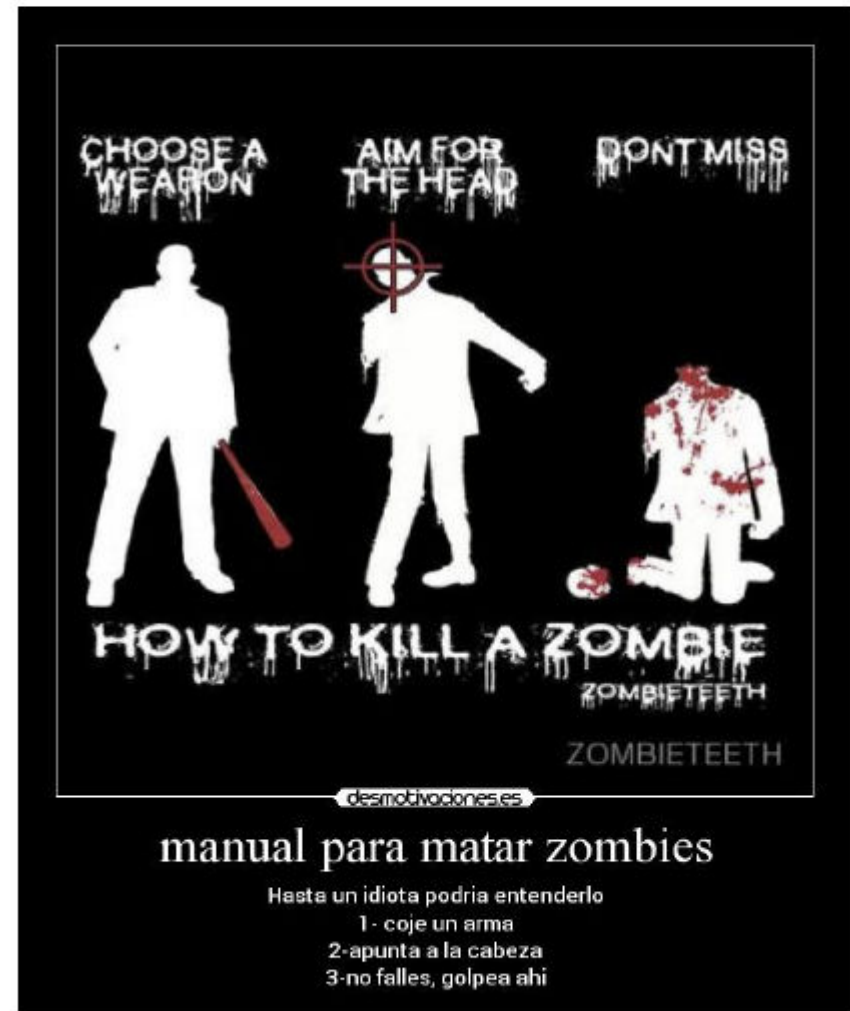
Matar procesos zombie

- Un zombie es un proceso que finaliza su ejecución pero cuyo padre no es consciente (no recibe el código de retorno).
- No consume recursos pero el núcleo del sistema operativo mantiene información acerca del mismo a la espera de que el proceso padre la reclame.

- Consulta de zombies:

```
ps axo stat,ppid,pid,comm | grep  
-e defunct
```

- Después hacemos: `kill -9 <ppid>`



Procesos en Linux

- **Ejecución en primer plano:** El usuario escribe el comando en el prompt de la interfaz de comandos y pulsa INTRO. Sólo después de que el comando se halla ejecutado, la shell muestra el prompt indicando que esta preparada para el siguiente comando.
- **Ejecución en segundo plano:** Una vez que el usuario indica a la shell el comando que quiere ejecutar y pulsa INTRO, esta le devuelve el control (es decir, muestra el prompt) de inmediato. El comando introducido se esta ejecutando en segundo plano, es decir, que el usuario puede seguir introduciendo comandos (que a su vez también pueden estar en segundo plano) mientras el primero se ejecuta.

Procesos en Linux

- Podemos indicar que un proceso se ejecute en segundo plano poniendo `&`:
 - Ej: `xeyes &`, devuelve esto: `[1] 2289`
 - La salida es el número que se asigna al trabajo en ejecución entre corchetes `[1]` y el pid del proceso en este ejemplo (2289).
 - Una vez ejecutado en segundo plano, podemos ejecutar:
 - `fg %número_trabajo`, para pasar a primer plano.
 - Ej: `fg %1`
 - Y si está en primer plano, podemos ejecutar:
 - `bg %número_trabajo`, para pasar a segundo plano.
 - Ej: `bg %1`
- `Jobs`: muestra los procesos en segundo plano y su estado.
- Una vez que un proceso se está ejecutando en primer plano:
 - Podemos pararlo (no es lo mismo que matarlo) con: `Ctrl+z`.
 - Podemos reanudarlo en segundo plano con `bg %número_trabajo`

Número nice y prioridad de procesos

- Linux realiza una planificación por prioridades dinámicas. Inicialmente, al lanzar un proceso se le asigna un valor de prioridad base (el número **nice**), cuyo valor marca la prioridad del proceso:
 - Valores bajos (negativos): más prioridad. El máximo es -20
 - Valores altos (positivos): menos prioridad. El mínimo es 20
- Por defecto, cada proceso hereda la prioridad base de su proceso padre.
- La prioridad dinámica del proceso se calcula en función de la prioridad base, teniendo en cuenta factores como el consumo de CPU realizado, ejecución de código dentro del núcleo, etc.
- Los comandos **nice** y **renice** permiten modificar una prioridad:
 - `nice -5 konqueror` aumenta la prioridad del proceso konqueror en 5
 - `nice --5 konqueror` disminuye la prioridad anterior en 5
 - `renice 14 2507` fija el valor de nice a 14 para el proceso con pid 2507

Procesos en Linux

- **time** muestra el tiempo que tarda en ejecutarse una orden cualquiera. El texto que devuelve el comando tiene tres partes:
 - real indica el tiempo efectivo de ejecución de dicho comando.
 - user define el tiempo de CPU utilizado.
 - sys indica el tiempo empleado en realizar llamadas al sistema.
- **lsuf** muestra información sobre los ficheros abiertos por procesos
 - -c permite especificar un proceso. Ej `lsuf -c nautilus`
- **sleep** hace una pausa el tiempo indicado. Se suele emplear para retener la ejecución de un comando. Ej `ls -l; sleep 2; pwd; sleep 3; clear`
- **watch** ejecuta un comando de forma periódica mientras esté abierto el shell.
 - -n para indicar el número de segundos entre intervalos. Ej: `watch -n 5 free -m`
 - -d=cumulative muestra las diferencias en sucesivas actualizaciones. Ej monitorización cambios en un directorio:
 - `watch -d ls directorio`
 - `watch -d=cumulative ls -l directorio`

Instalación de aplicaciones

- **Apt-get** permite instalar paquetes a través de Internet, para lo cual consultará en los repositorios del fichero `/etc/apt/sources.list`.
- La sintaxis es `apt-get [opciones] [nombre_paquete]` siendo las opciones básicas las siguientes:

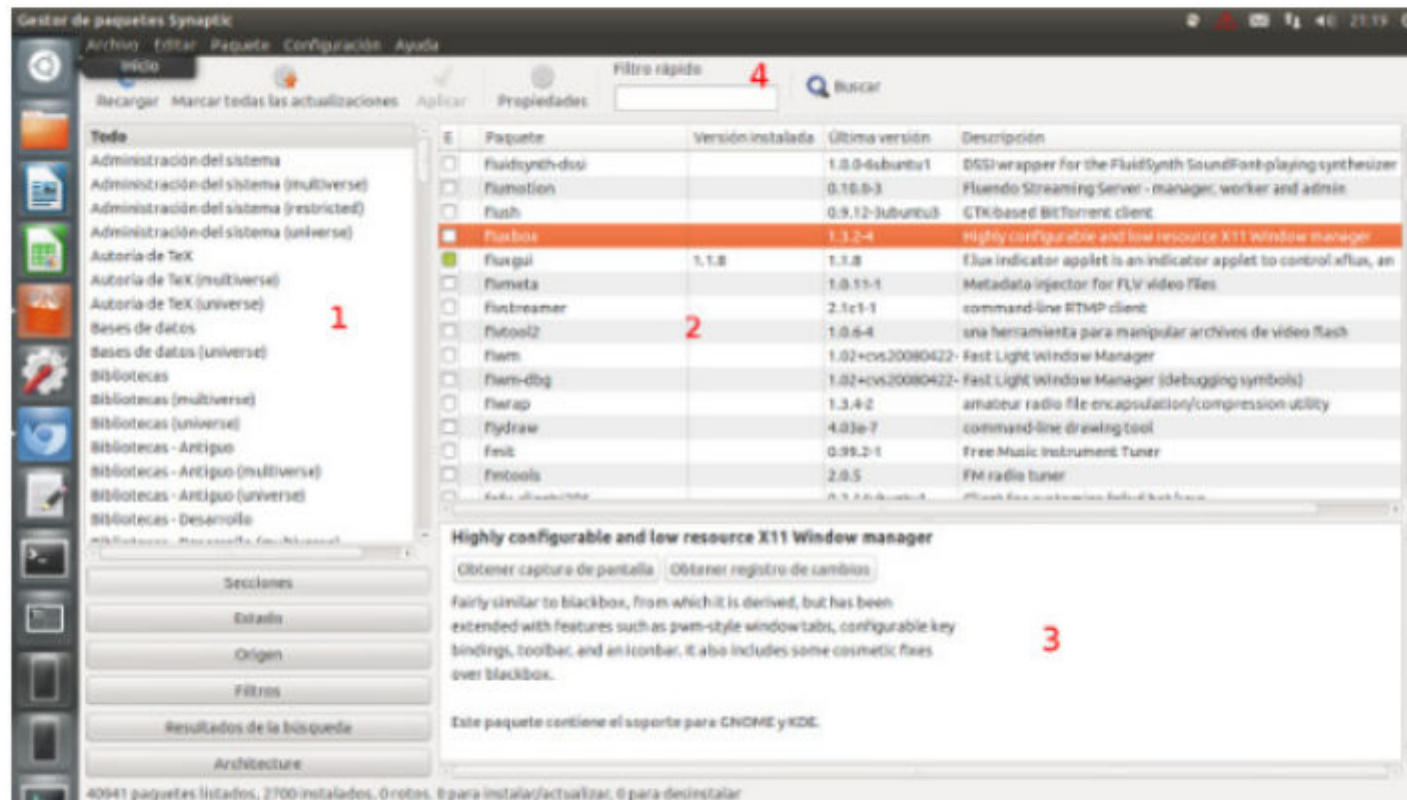
install	Descarga un paquete de internet y lo instala. Ej.- <code>apt-get install quota</code>
remove	Desinstala un paquete pero no lo elimina del disco duro.
purge	Desinstala un paquete y lo borra del disco duro
autoremove	Se eliminan automáticamente los paquetes desfasados. Ej.- <code>apt-get autoremove</code>
update	Se usa para actualizar la lista de paquetes que tenemos instalados. Los índices de paquetes disponibles se obtienen de los lugares especificados en <code>/etc/apt/sources.list</code> . Siempre se debe realizar un update antes de un upgrade.
upgrade	Se usa para instalar la versión más nueva de todos los paquetes instalados en el sistema provenientes de alguna de las fuentes listadas en <code>/etc/apt/sources.list</code> . Los paquetes instalados con una nueva versión disponible son descargados y actualizados.

Añadir y consultar repositorios

- Los repositorios de instalación son las ubicaciones en las que se buscan las aplicaciones a instalar.
- Estas direcciones vienen listadas en el fichero `/etc/apt/sources.list`.
- El comando `add-apt-repository ppa:[repositorio]` permite actualizar ese listado. Ejemplo: `add-apt-repository ppa:do-core/ppa`
- Apt-cache permite consultar en los repositorios información sobre paquetes que se pueden instalar. Sus opciones son:

search	Busca los paquetes asociados con un determinado concepto, usando para ello una cadena de búsqueda. Ej.- <code>apt-cache search "atari"</code>
depends	Muestra la información de dependencias de un determinado paquete
showpkg	Muestra una descripción de un paquete. Ej.- <code>apt-cache showpkg NOMBRE_PAQUETE</code>
pkgnames	Muestra toda la lista de paquetes que podemos instalar. Ej.- <code>apt-cache pkgnames</code>

Gestor de paquetes: Synaptic



1. Índice de paquetes y programas
2. Paquetes resultado de la búsqueda
3. Descripción del paquete y dependencias que se instalarán por defecto
4. Buscador

Sistemas de ficheros

- El sistema de nombrado de dispositivos es el siguiente:

- /dev/sda disco 1
 - /dev/sda1 disco 1, partición 1
 - /dev/sda2 disco 1, partición 2 (hasta 4 particiones físicas)
 - /dev/sda5 disco 1, partición 5 (desde 5 particiones lógicas)
- /dev/sdb disco 2
- /dev/sdc disco 3 (hasta la letra p)

lsblk Nos da información de los discos que hay con sus particiones y el punto de Montaje. Reconoce todos los discos aunque no estén montados.

df muestra info de los discos montados, con el espacio libre y ocupado.

blkid muestra los discos con su id de disco (parecido a un numero de serie)

Sistema de ficheros

- **fdisk** permite crear, ver y modificar particiones en discos.
 - -l permite mostrar todos los discos. Ej: `fdisk -l`
 - -l disco permite mostrar el disco especificado. Ej: `fdisk -l /dev/sda`
 - Ej `fdisk /dev/sda` permite modificar las particiones del disco
- **fsck** permite revisar y reparar un disco con sistema de ficheros ext. Es necesario desmontar el disco previamente
 - -n informa de los errores
 - -p repara los errores
 - -v genera mensajes de diagnóstico
- **badblocks** permite buscar bloques erróneos y corregirlos
- **mkfs** formatea la partición existente en un sistema de ficheros
 - -t permite especificar el sistema de ficheros (ext2, ext3, ext4, ntfs, vfat)
 - Ejemplo: `mkfs t ext4 /dev/sdb1`
- **mkswap** permite crear una partición de tipo swap.
 - Ejemplo: `mkswap c /dev/hda3 50000`

Gestión sistema ficheros con GParted

The screenshot shows the GParted application window titled "/dev/sda - GParted". The menu bar includes "GParted", "Edit", "View", "Device", "Partition", and "Help". The toolbar contains icons for "New", "Delete", "Resize/Move", "Copy", "Paste", "Undo", and "Apply". On the right, a dropdown menu shows the selected device: "/dev/sda (60.00 GiB)".

Below the toolbar, a visual representation of the disk layout shows two partitions highlighted with colored borders: /dev/sda2 (33.11 GiB) with a green border and /dev/sda5 (24.79 GiB) with a blue border. A red border highlights a small unallocated space to the right of /dev/sda5.

The main table displays the following partition information:

Partition	File System	Label	Size	Used	Unused	Flags
/dev/sda1	ntfs	System Reserved	100.00 MiB	24.14 MiB	75.86 MiB	boot
/dev/sda2	ntfs		33.11 GiB	8.89 GiB	24.22 GiB	
▼ /dev/sda3	extended		26.79 GiB	---	---	
/dev/sda5	ext4		24.79 GiB	3.39 GiB	21.41 GiB	
/dev/sda6	linux-swap		2.00 GiB	---	---	
unallocated	unallocated		1.00 MiB	---	---	

At the bottom of the window, a status bar indicates "0 operations pending".

Sistema de ficheros

- El fichero `/etc/fstab` almacena la información de los sistemas de ficheros que deben montarse en el arranque del sistema. El formato de este fichero es el siguiente:

`<disp> <punto_montaje> <sistema_de_ficheros> <opciones> <dump> <pass>`

- Se trata de:
 - Dispositivo a montar
 - Ubicación de montaje del dispositivo
 - Tipo de sistema de ficheros
 - Opciones de montaje (permisos, privilegios, etc)
 - Dump: se activa (1) para hacer un seguimiento por si surge un error. Normalmente está desactivado (0).
 - Pass: con un 1 o 2 revisa el sistema de archivos durante el proceso de arranque por si hubiera errores lógicos con `fsck` (con un 1 le da más prioridad). Si se usa el 0 está desactivado

Gestión de dispositivos

- Linux permite montar dispositivos dentro de un directorio vacío del sistema de ficheros
 - **Mount** permite consultar los sistemas de ficheros montados
 - **-t** permite especificar el tipo de sistema de ficheros
 - Auto detecta automáticamente el sistema de archivos que usa el dispositivo
 - Por ejemplo podemos especificar ext2, ext3, ntfs, vfat, etc.
 - **-o** especifica opciones de montaje
 - Auto permite montaje automático con el comando mount -a
 - Nouser sólo root puede montar este sistema de ficheros
 - Exec permite ejecutar programas o ficheros en ese sistema de ficheros
 - Owner permite que un usuario no root pueda montar o desmontar
 - Remount intenta montar automáticamente si se desmonta
 - Ro se monta como sólo lectura (read only)
 - Rw se monta como lectura y escritura
 - Defaults añade las opciones por defecto (rw, suid, dev, exec, auto, nouser, async)
- Umount permite desmontar los sistemas de ficheros con **umount dispositivo**

Gestión de dispositivos

- Ejemplos mount:

`mount /dev/hda3`

Busca en `/etc/fstab` donde debe montar ese fichero de dispositivo y lo monta con las opciones que se indiquen

`mount -t ext3 -o exec,users /dev/hda4 /home2`

Monta la partición 4 del disco hda en el directorio `/home2`, especificando que se trata de un sistema de tipo ext3. Además realiza el montaje con las opciones `exec` y `users`.

- Ejemplos umount:

`umount /dev/hda2`

desmonta la partición 2 del disco hda

`umount -a`

desmonta todos los sistemas de ficheros indicados en `/etc/mtab`

El fichero `/etc/mtab`, mantiene los sistemas de ficheros montados en el sistema. Es mantenido automáticamente por el sistema y su modificación manual puede ocasionar inconsistencias.

Gestión de dispositivos

- `df` muestra el tamaño y espacio libre de los sistemas de ficheros del disco. Por defecto los muestra todos.
- `dmesg` muestra información sobre las alertas del kernel del sistema.
- `du` Indica la cantidad de espacio usada por un archivo o directorio

Gestión de red: configuración básica

- Para configurar el interfaz de red de forma permanente editamos el fichero `/etc/netplan/archivo.yaml` añadimos:
 - Para IP estática

```
1 # The primary network interface
2   auto eth0
3   iface eth0 inet static
4   address 192.168.3.90
5   gateway 192.168.3.1
6   netmask 255.255.255.0
7   network 192.168.3.0
8   broadcast 192.168.3.255
```

- A continuación reiniciamos el interfaz mediante:
`ip a flush eth0`
- Reiniciamos el servicio mediante
`systemctl restart network-manager.service`

Gestión de red: configuración básica

- Hostname permite establecer o comprobar el nombre del equipo
 - Ej: `hostname Mipc`
- Podemos cambiar el nombre del equipo de forma permanente si editamos `/etc/hostname` y `/etc/hosts`
- Podemos añadir direcciones IP estáticas en `/etc/hosts` incluyendo IP NICKNAME.
- Por ejemplo si tenemos un servidor para copias de seguridad en la IP 192.168.1.100, si incluimos 192.168.1.100 backup podemos hacer ping backup en lugar de usar la IP.

Servicios en systemd

- Systemd permite la gestión de servicios de la siguiente manera:
 - `systemctl start [name.service]` arrancamos un servicio
 - `systemctl stop [name.service]` paramos un servicio
 - `systemctl restart [name.service]` reiniciamos un servicio
 - `systemctl reload [name.service]` recargamos un servicio (si el servicio lo permite puede cargarse la configuración sin necesidad de pararlo y volver a iniciarlo)
 - `systemctl status [name.service]` consultamos el estatus
 - `systemctl is-active [name.service]` consultamos si está activo
 - `systemctl enable [name.service]` habilitamos un servicio
 - `systemctl disable [name.service]` deshabilitamos un servicio
- Podemos consultar los servicios activos con `systemctl list-units --type service --all`
- Dentro de los servicios que tiene el sistema, network-manager es el encargado de la conexión de red.
 - Ejemplo:
`sudo systemctl restart network-manager.service`
Permite reiniciar el servicio de red

Otros comandos de interés

- Los comandos que permiten apagar, reiniciar, suspender e hibernar el sistema son:
 - `systemctl halt`
 - `systemctl poweroff`
 - `systemctl reboot`
 - `systemctl suspend`
 - `systemctl hibernate`
- El comando `shutdown` permite especificar con parámetros si queremos hacer alguna de las acciones anteriores:
 - `-H halt`
 - `-P poweroff`
 - `-r reboot`