101 - Arquitectura de sistema

1 - Identificar y editar configuraciones de hardware

1. **Activación de dispositivos**
   - El hardware básico se configura desde la BIOS
2. **Inspección de dispositivos**
   - Comandos
     - lpsci: muestra todo lo conectado al bus PCI
       - -s id: Muestra la información de ese dispositivo
       - -v: Muestra todos los detalles
     - lsusb: muestra los dispositivos conectados por USB
       - -d id: muestra la información de ese dispositivo
       - v: muestra detalles
     - lsmod: muestra los módulos (drivers) cargados actualmente
     - modprobe: carga módulos del kernel y todas sus dependencias
     - insmod: sólo carga módulos del kernel, NO carga las dependencias
   - **Los datos del hardware** están almacenados en /proc y /sys, meditante cat podemos mostrarlos.
3. **Tipos de dispositivo:**
   - ColdPlug: Es necesario apagar el pc para conectar el dispositivo, como tarjetas PCI, dispositivos IDE, etc.
   - HotPlug: permite conectar/desconectar en caliente, ej: usb. Cuando se conecta automáticamente se activa un evento Udev que actualiza los dispositivos **/dev**
4. **Dispositivos de almacenamiento**
   - Cualquier dispositivo de almacenamiento se identifica en un archivo dentro de /dev. El nombre del archivo depende del tipo de dispositivo (SATA, IDE..) y de las particiones.

Tipo	Criterio nombre	Ejemplo
IDE	Canal IDE usado, Master / Slave y particiones	/dev/hda1: partición 1 del Master /dev/hda2: 2º partición del primer canal slave
SATA	Por la BIOS y por particiones	/dev/sda1: partición 1 del disco 1 /dev/sdb2: partición 2 del disco 2
SSD	Por la BIOS y por particiones	/dev/sdc: como si fuese sata

Ampliación modprobe

Los kernel de linux tienen un diseño modular. Funcionalmente es extendible a módulos o drivers, el uso de modprobe, añade o quita estos módulos del kernel de Linux. El comando es inteligente y carga las dependencias del módulo automáticamente.

El Kernel usa modprobe para solicitar módulos, modprobe busca a través de los módulos instalados, para encontra el driver necesario.

```
# Modprobe requiere permisos de SU
sudo modprobe <options> <module name>

# Modprobe sin opciones, añadirá un módulo al kernel
sudo modprobe <module name>

# Con -a o -all podemos añadir varios módulos a la vez
sudo modprobe <options> -a <first module name> <second module name>
```

Otras opciones importantes:

Opción	Descripción
--remove -r	Quita el módulo indicado, aplica --remove-dependencies. Usado para quitar módulos rotos
--first-time	Muestra un erro para un un módulo insertado o quitado
--show	No ejecuta el insertar/remover, pero muestra la salida del comando
--verbose -v	Muestra un extra de información

Por defecto, todos los módulos del kernel están listados en **/lib/modules** como .ko (kernel object).

```
# Para encontrar todos los módulos disponibles para el actual kernel
find /lib/modules/$(uname -r) -type f -name '*.ko*' | more
```

2 - Inicio (boot) del sistema:

Es posible pasarle parámetros al kernel en el momento del inicio para especificar datos como memoria, dispositivos, etc.

Cargador de boot (Bootloader):
Los principales son Grub y Lilo, a estos también se les puede añadir parámetros como:

Parámetro	Descripción	Ejemplo
acpi	Conecta/desconecta el soporte ACPI	acpi = off
init	Define otro programa en lugar de /sbin/init	init = /bin/bash
mem	Define cuanta ram estará disponible	mem=512
maxcpus	Nº máximo de núcleos	maxcpus=2
quiet	No muestra la mayoría de los mensajes de inicio	quiet
vga	modo de vídeo	vga=773

Parámetro	Descripción	Ejemplo
root	Define la partición raíz diferente a la de bootloader	root=/dev/sda3
ro o rw	realiza el montaje en sólo lectura	ro

demsg: es un comando de Linux que se usa para examinar o controlar la información mas reciente que se genera sobre el sistema (kernel log). El programa ayuda a los usuarios a imprimir los mensajes de arranque.

journalctl: es el demonio de systemd que recoge los logs del sistema, incluidos los syslog de SysVinit.

Las opciones mas interesantes de journalctl serían:

Opción	Descripción	Ejemplo
-r	Logs mas recientes	journalctl -r
-n x	Mostrará las últimas x lineas	journaltcl -n 25
-f	Muestra las nuevas entradas en tiempo real	journalctl -f
-k	Sólo los logs del kernel	journalctl -k
-xe	Muestra los errores	journalctl -xe
-u servicio	Busca el servicio especificado	journalctl -u ssh
--since --until	Buscar por fechas	journalctl --since "2020-07-10 15:10:00" --until "2020-07-12"
_PID=xxx	Puede filtrar por UID (User ID), GID (Group ID) o PID (process ID)	journalctl _PID=1234
--disk-usage	Muestra lo que están ocupando los logs	journalctl --disk-usage

```
journalctl -u httpd --since '1 minute ago' -n 30
```

3 - Cambiar runlevels, apagar y reiniciar el sistema

El runlevel es el nivel de ejecución del sistema, el programa **/sbin/init** que es invocado tras el bootloader identifica el nive de ejecución metido como parámetro en la carga del kernel o mirándolo en **/etc/inittab**.

Luego carga los programas, scripts y servicios indicados en ese archivo. La mayor parte de los scripts ejecutados por init se guardan en **/etc/init.d** pero en otras distros es en **/etc/rc.d**

Los niveles de ejecución
Se enumeran del 0 al 6 y tienen el formato:

- **id**: nombre de hasta 4 caracteres para identificar la entrada de inittab

- **runlevels**: es la lista de runlevels que se ejecutarán
- **acción**: el tipo de acción:
  - sysinit: inicio del sistema
  - wait: se ejecuta y espera su finalización
  - ctrlaltdel: se ejecuta cuando se pulsa control+alt+del o se reciba la señal SIGINIT
- **proceso**: comando que se activará

Numeración de los runlevels:

- 0: apagar el sistema
- 1: usuario único, sin red o servicios
- 2: multiusuario, estándar en la mayoría de distros
- 3: multiusuario, estándar en otras muchas distros
- 4: No se usa
- 5: No se usa
- 6: reinicio del sistema

Para cambiar entre runlevels se usa telinit nº_runlevel y para ver el runlevel actual usamos el comando runlevel

SysVinit

Los sistemas basados en SysVinit, init es el primer proceso que lee el kernel de linux. El programa por defecto que usa es /sbin/init

/etc/inittab es el fichero de configuración básico, contiene las direcciones para init, los scripts a correr cuando se inicia un determinado runlevel.

Los servicios se lanzan con fichero *.init

SystemD

Se trata de una suite de software, que provee un conjunto de componentes de sistema para Linux. Su principal objetivo es unificar la configuración de servicios a través de todas las distribuciones linux.

Su componente principal es el "system and service manager" (un sistema Init), pero también proporciona reemplazos para varios demonios y utilidades, incluida la administración de dispositivos, la administración de inicio de sesión, la administración de conexiones de red y el registro de eventos.

Los daemon se lanzarán como *.target, almacenados en el sistema por defecto en /usr/lib/systemd.

Las unidades del administrador del sistema estarán ubicadas en /etc/systemd

Hay unos target "especiales", que son lo que arrancarán el sistema, con el comando systemctl get-default, podremos ver en que target iniciamos por defecto.

Con el comando systemctl isolate *.target, podemos cambiar de un target a otro, la comparación con SysVinit sería:

Runlevel	Target
0	poweroff.target
1	rescue.target
2, 3, 4	multi-user.target
5	graphical.target
6	reboot.target

Para cambiar el target por defecto con el que arrancará el sistema, tenemos el siguiente ejemplo que cambia de modo gráfico a modo consola.

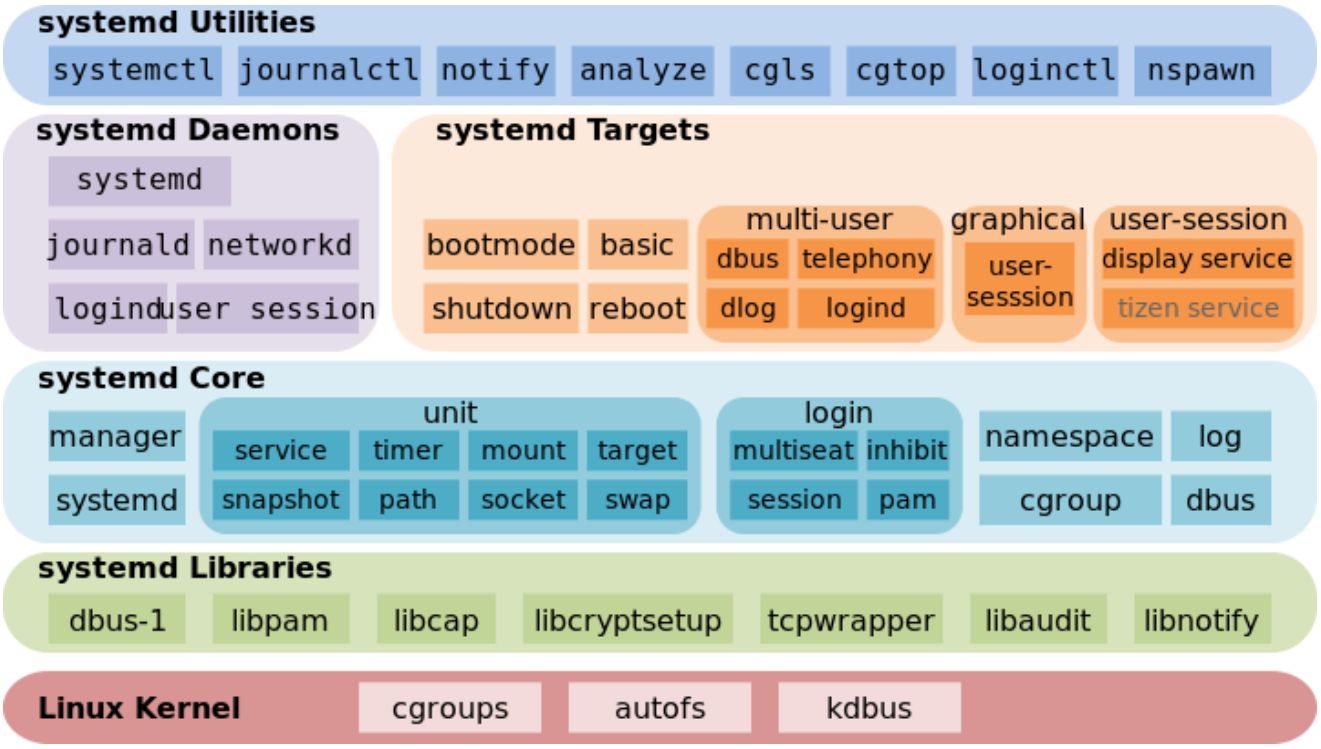
```
systemctl disable graphical.target
systemctl enable multi-user.target
systemctl set-default multi-user.target
```

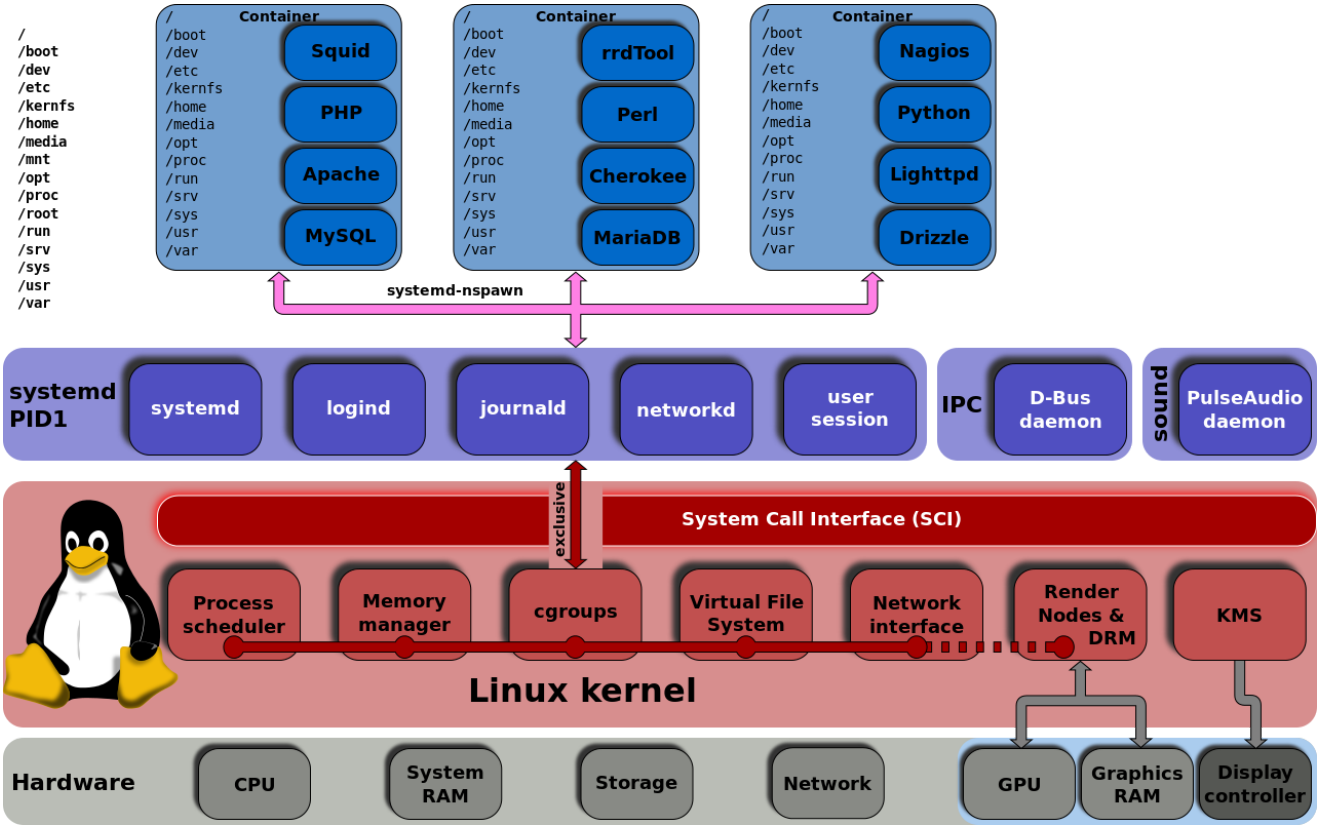
El resto de servicios, los arrancaremos, pararemos con los siguientes comandos:

```
# Todos los comandos seguirán el patrón
systemctl <opción> <servicio sobre el que actuar o target>
>> systemctl start httpd
```

Comando	Acción
start	arranca un servicio hasta que se apague el equipo
stop	para un servicio
status	mira el estado de un servicio
enable	añade un servicio al arranque del sistema
disable	quita un servicio del arranque del sistema.
restart	apaga y enciende un servicio de forma ordenada
reload	relee un servicio sin apagarlo
mask	enmascara un servicio, no lo deja arrancar enviándolo a null
unmask	desenmascara un servicio

Como veremos a continuación, las utilidades de systemd son varias, tal como journalctl visto anteriormente o logintcl (para ver el arranque de usuarios) o systemd-analize entre otras.





Apagar y reiniciar:

Comando shutdown

El horario puede ser:

- hh:mm hora de ejecución

- +m minutos para la ejecución
- now o +0 inmediato

Opciones:

- -a: usar el archivo de permiso /etc/shutdown.allow
- -r: reiniciar la máquina
- -h: apagar el pc
- -t segundos: tiempo de espera antes de que se ejecute la acción

El mensaje será enviado a todos los usuarios.

Para prevenir que cualquiera pueda reiniciar la máquina con ctrl+alt+supr la opción -a en el comando shutdown de la líndea ctrlaltdel de /etc/inittab