

④ GNU/Linux en general:

- Se distribuye con licencia GPL.

- Es un SO multitarea (Distribuye el tiempo de procesado entre los procesos que lo necesitan).

- Tres componentes fundamentales:

⊗ Solo se puede tener un único kernel instalado en una máquina.

- Kernel: El núcleo del SO. Administra memoria, CPU y demás recursos de hardware. Es monolítico. Podemos ver su código fuente y modificarlo. Podemos interactuar con él mediante el shell.

- Shell: Interpretador de comandos. Es el medio de comunicación entre el usuario y el SO. Mediante el shell el usuario y el kernel. Cada usuario puede tener una shell, pero se pueden personalizar y programar.

- File System: Sistema de Archivos. Estructura la información guardada en un medio de almacenamiento, que luego será representada textual y gráficamente por un gestor de archivos. La mayoría de SO tienen su propio File System. GNU/Linux utiliza una gran cantidad de File Systems (EXT2, NTFS, etc.).

Directorio más importante del GNU/Linux:

- • / : raíz del sistema de directorios
- • /etc : Archivo de configuración de la mayoría de programas.
- • /home : Archivos personales de los usuarios.
- • /bin : Contiene comandos básicos y muchos programas.
- • /dev : Archivos simbólicos que representan partes del hardware.
 - /tmp : Ficheros temporales.
- • /usr : Programas y librerías instalados con la distribución. (Universal System Resources)
 - /lib : Librerías, scripts y módulos del kernel.
- • /var : Datos variables como archivos de logs, bases de datos, copia de seguridad, etc.
 - /proc : Información temporal sobre los procesos del sistema.

② Discos y particiones:

Debido a limitaciones técnicas del MBR para la tabla de particiones.

• 3 Tipos de particiones:

- Primaria: División primaria de disco. Solo puede haber 4 de ellas y la primera 1 que ocupa todo el espacio del disco.
- Extendida: Idéntica para romper con la limitación de 4 particiones primarias en un solo disco físico. Solo sirve para contener cualquier número de particiones lógicas. Solo puede existir una por disco.
- Lógica: Ocupa un trozo de partición extendida o la totalidad de la misma. Está formateada con un tipo específico de File System.

- En una máquina con varios discos, las particiones se identifican así:

- Primer Disco (Todo el Disco): `/dev/hda`

Particiones Primarias

└ `/dev/hda1`

└ `/dev/hda2`

└ `/dev/hda3`

└ `/dev/hda4`

Particiones Lógicas

└ `/dev/hda5`

└ `/dev/hda6`

...

└ `/dev/hda#` (# = Cantidad de particiones lógicas)

NOTA: Para un segundo disco será igual, pero el nombre `/dev/hdb` (para un tercero será `/dev/hdc`, etc).

- Para instalar GNU/Linux se necesita como mínimo una partición para el "/" (raíz). Se recomienda como al menos 2, para usar la segunda como partición SWAP (memoria virtual o almacenamiento temporal en el disco duro cuando no hay espacio suficiente en la RAM).
- Un disco rígido puede particionarse y en cada partición instalar SO distintos.

③ • Arranque con MBR

- BIOS es el responsable de iniciar la carga del SO a través del MBR. UEFI reemplaza la interfaz del BIOS antiguo. UEFI usa GPT como mecanismo de particionado y es compatible con el MBR tradicional.
- MBR existe en todos los discos. Se encuentra en el cilindro 0, cabeza 0, sector 1, siendo la zona reservada para él. Tiene un tamaño de 512 bytes (tamaño estándar de sector). Los primeros bytes corresponden al MBR (punto al que quisiéramos arrancar el SO), luego a partir del byte 446 está la tabla de particiones (64 bytes) y al final están 2 bytes libres o para firmar el MBR.
- La última acción del BIOS es leer el MBR. Lo lleva a memoria y ejecuta.
- Bootloader: Gestor de Arranque. Su finalidad es la de cargar una imagen de SO de alguna partición para su ejecución. Se ejecuta luego del código del BIOS. Hay varios (GRUB, LILO, etc).

• Bootstrapping: Proceso de inicio de una máquina y carga del sistema operativo.

• El orden de boot es:

- 1- El firmware lee el sector de arranque.
- 2- Se ejecuta el código del BIOS.
- 3- Se carga el gestor de arranque.
- 4- Se carga el kernel.

④ Permisos

• Tres tipos que se lecan en una notación octal:

Permiso	Valor	Octal
Lectura	R	4
Ejecución	X	1
Ejecución	X	1

• Se aplican con el comando `chmod`.

• Se aplican sobre los usuarios:

UGO (U: usuario, G: grupo, O: otros)

Ej: `chmod 755 /tmp/script`

Resultado: $\overline{RWRXR-XR-X}$

En Binario: 111 101 101

⑤ Arranque en System V

- 1) Se empieza a ejecutar el código del BIOS.
- 2) El BIOS ejecuta el POST.
- 3) El BIOS lee el sector de arranque (MBR).
- 4) Se carga el gestor de arranque (MBR).
- 5) El boot loader carga el kernel.
- 6) Se monta el initrd como sistema de archivos raíz y se inician los componentes esenciales.
- 7) El kernel ejecuta el proceso init y se demonta el initrd.
- 8) Se lee el `/etc/passwd`.
- 9) Se ejecutan los scripts de inicio para el runlevel 1.
- 10) El final del runlevel 1 le indica que vaya al runlevel por defecto.
- 11) Se ejecutan los scripts de inicio para el runlevel por defecto.
- 12) El sistema está listo para usarse.

⑥ Proceso INIT

del archivo `/etc/init.d`

- Carga todos los subprocesos necesarios para el correcto funcionamiento del SO.
- Tiene PID 1, y se encuentra en `/sbin/init`. En `System V` se le configura `atime`.
- También se encarga de montar los File System y de hacer disponible los demás dispositivos.
- No tiene padre y es el padre de todos los procesos.

⑦ Runlevels

- Son el modo en que opera Linux. Cada Runlevel es responsable de levantar o bajar una serie de servicios.

Hay 7 niveles:

0: Halt

1: Modo Monousuario

2: Modo Multiusuario sin soporte de red

3: Modo Multiusuario completo por consola

4: No se usa

5: Modo Multiusuario completo con interfaz gráfica

6: Reboot

Scripts RC

- Los scripts que se ejecutan en cada runlevel están en `/etc/init.d`.
- En `etc/rcX.d` (donde $X=0..6$) hay links a los archivos del `/etc/init.d`.
- I mportante se usa para administrar el orden de los valores numéricos del `/etc/rcX.d`.
Utiliza cabeceras en los scripts del `/etc/init.d` que permitan identificar la relación con otros scripts.

③ Proceso de Arranque Uptart

- Sistema de Arranque compatible con System V, pero que ejecuta los tasks de forma asíncrona, permitiendo así una mejora en las prestaciones, a diferencia de System V que es estrictamente sincrónico.
- Estos tasks se llaman "jobs", y su principal objetivo es definir servicios o tasks a ser ejecutados por init o ante determinados eventos (arranque del equipo, transición de USB, etc.).
- Cada job se define en el "/etc/init" (archivos .conf).
- A través de initd podemos administrar los jobs del daemon de Uptart. No más "/etc/initd".

④ System MD

- Sistema que centraliza la administración de daemons y librerías del sistema. Mejora el funcionamiento de los daemons.
- Los runlevels son reemplazados por targets.
- Puede ser controlado por systemctl.
- El daemon systemd reemplaza al proceso init, forzando a tener PID 1.
- No más "/etc/initd".
- Los unidades de trabajo se llaman units: Service, Socket, Target, Snapshot.
- Activación por Socket: Mecanismo de iniciación bajo demanda. Podemos ofrecer una variedad de servicios sin que realmente estén iniciados.
- CGROUPs: Permite organizar un grupo de procesos en forma jerárquica. Agrupa conjuntos de procesos relacionados.
- ESDs: Define qué particiones se montan al arranque.