**LA BIBLIA DE LINUX**

Todas las distribuciones de Linux (Debian, Ubuntu, Kali Linux…) Provienen de un kernel llamado GNU/Linux.

Distribuciones cronológicas: <https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux_Distribution_Timeline.svg>

Distribuciones principales de GNU/Linux para Ciberseguridad:

* Ciberseguridad para principiantes:
  + Kali Linux: Basado en Debian, con muchas (demasiadas) herramientas. Está bien para principiantes. Se usa para seguridad ofensiva o defensiva. Sobre todo para pentesting.
  + Kali Purple: Basado en Debian, sobre todo para seguridad defensiva (Blue team).
  + Parrot: Debian. La diferencia con Kali es el entorno del escritorio, pero se parecen bastante.
  + BlackBuntu: Basado en Ubuntu (Ubuntu se basa en Debian).
* Ciberseguridad para anonimato y privacidad:
  + Tails: Para anonimato y privacidad.
* Ciberseguridad para informática forense:
  + Sift WorkStation.
  + Caine.

## **Estructura de Ficheros de Linux:**

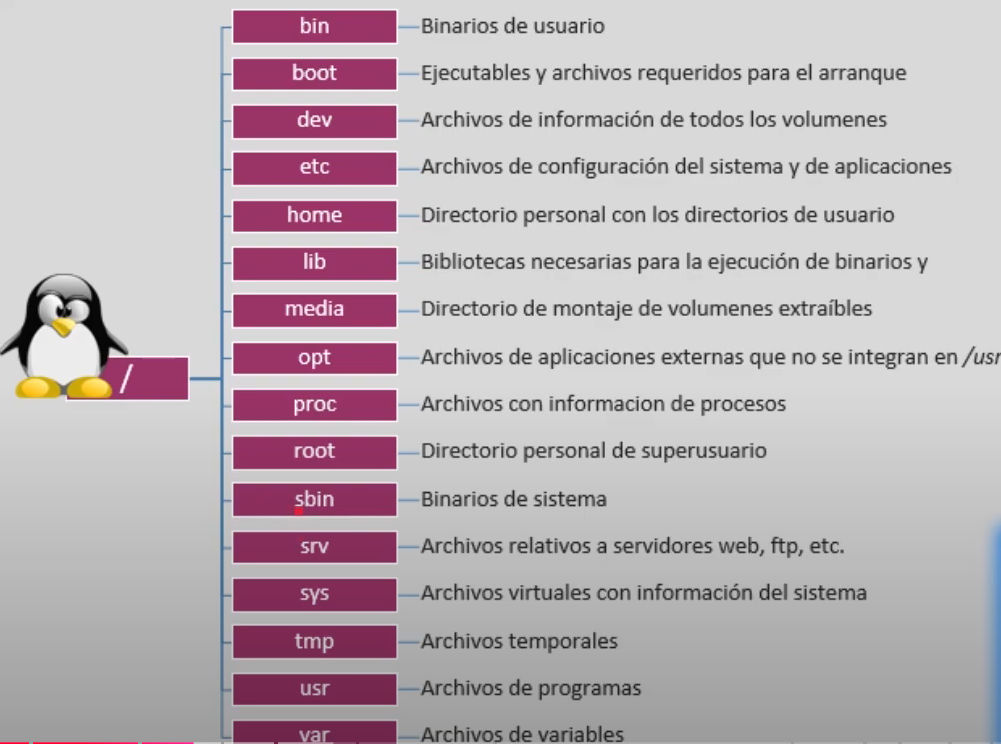
*INODOS, BLOQUES Y SUPER BLOQUES*

* **INODOS:** los Inodos contienen metadatos de todos los ficheros y directorios. Todos los ficheros y directorios van a tener asociados su Inodo y es de esta forma que el SO va a referenciarlos, mediante su Inodo, no mediante su nombre.
* **Bloques:** Es una unidad de almacenamiento donde se guardan los datos de los ficheros. Aquí relacionamos Ficheros con Inodos.
* **Super Bloques:** Contiene información crucial sobre el sistema de ficheros.

*DIRECTORIOS*

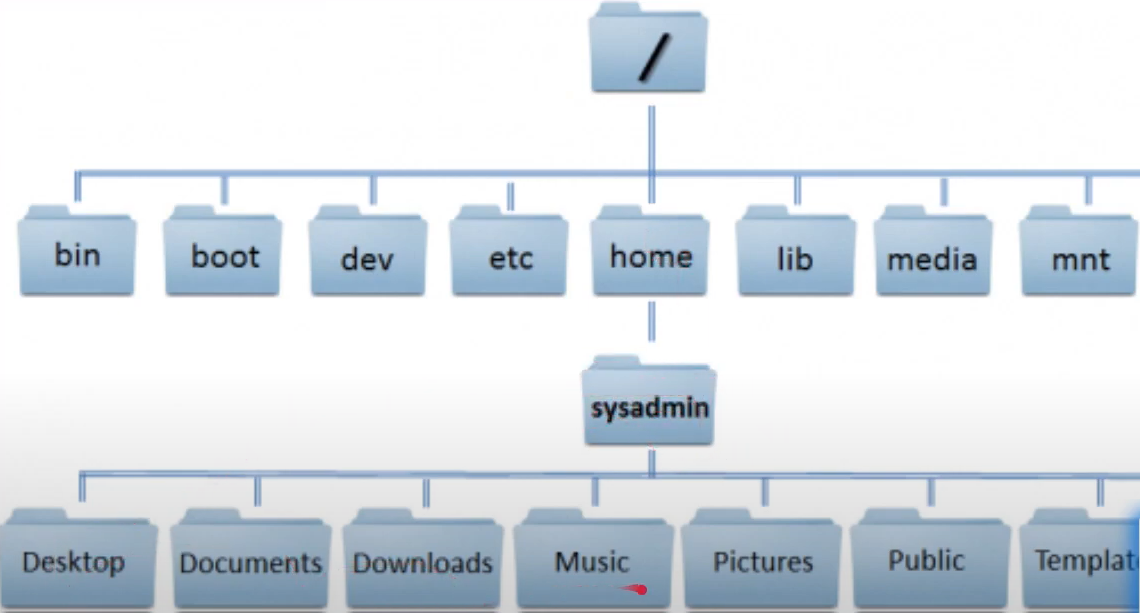
La raíz es la base de todo, esta se representa con /. De esta derivan todos los directorios.

Estos son los directorios más importantes:



* Bin: Contiene los programas que se pueden ejecutar por TODOS los usuarios.
* Boot: Aquí se almacenan los archivos para el arranque del sistema.
* Sbin: Ejecutables esenciales para el arranque del sistema y su recuperación.
* Dev: Archivos de información que hacen el puente entre el hardware y el software.
* Etc: Archivos de configuración del sistema y aplicaciones.
  + /etc/passwd: Aquí se guardan datos de los usuarios del sistema. Nombre de usuario, Id, Id del grupo, información del usuario, Home…
  + /etc/group: Contiene información sobre los grupos del sistema.
  + /etc/shadow: Almacena las contraseñas de los usuarios de manera encriptada.
  + /etc/fstab: aquí se definen como y donde se montan los sistemas de archivos. Principalmente contiene información sobre las particiones.
  + /etc/network/interface: Aquí definimos la configuración de red para las interfaces de red. Funciona en la mayoría de distribuciones, excepto en las muy recientes.
  + /etc/hostname: Contiene el nombre del host del sistema.
  + /etc/resolv.conf: Aquí se definen los servidores DNS.
* Home: Directorio personal donde tenemos el directorio de cada usuario y de este descienden el escritorio, documentos, descargas… Cada usuario va a tener su directorio Home.

Adicionalmente, en este directorio vamos a tener directorios ocultos que empiezan por un punto, y dependiendo de la shell que uses vas a tener uno llamado .bashrc, zshrc…. Son importantes mencionarlos porque vamos a poder modificar o definir los alias, setear variables de entorno y realizar configuraciones permanentes.



* Media: Sirve para gestionar cuando enchufamos USB’s o cualquier hardware externo.
* Root: Es el Home del Administrador.
* Tmp: Archivos temporales que se eliminan cada vez que se apaga el equipo.
* Var: en este directorio es importante destacar que podemos ver los logs del sistema.
* Usr: Contiene los ficheros y directorios compartidos con todos los usuarios.
  + /Usr/bin/: Aquí encontramos los ejecutables.
  + /Usr/lib/: Aquí encontraremos las librerías.

Caben destacar dos directorios especiales que no se almacenan en los discos duros. Son sistemas de archivos virtuales mantenidos por el kernel del SO. Estos directorios sirven de interfaz entre el kernel y el espacio de usuario. En ellos podemos realizar configuraciones del sistema en tiempo real. Interactúas directamente con el Kernel en lugar de leer los datos del disco duro.

* /proc/: Nos muestra información sobre los procesos en ejecución, sobre el hardware del sistema e información básica del sistema.
* /sys/: Estructura de directorios que representa el Hardware del sistema. Este es bastante importante.

*ENLACES DUROS Y BLANDOS*

* **Enlace duro:**  Si por ejemplo tenemos un archivo llamado pepe.txt en dos directorios distintos. Un enlace duro es el que relaciona este archivo por su Inodo. Esto quiere decir que son el mismo archivo, puesto que tienen el mismo Inodo. Si se modifica uno, el otro también es modificado.
* **Enlace blando:** Este sería lo que en Windows es hacer una copia. Tenemos dos archivos llamados pepe.txt en dos directorios distintos y en esta ocasión si se modifica uno el otro no se modifica.

*PERMISOS*

* **Lectura (W):** Permisos de escritura.
* **Escritura (R):** Permisos de lectura.
* **Otro (o):** Engloba el resto de usuarios que no son el propietario.

*SHELL Y TIPOS*

El Shell es un intérprete de comandos, que interpreta un comando a lenguaje máquina.

* Sh: Es la Shell más básica y viene por defecto en la mayoría de distribuciones.
* Bash: Surgió de Sh, es como Sh con proteínas. Esta la vas a encontrar en la mayoría de distribuciones de forma predeterminada. También puede usarse como lenguaje de programación.
* ZsH: Recoge características de otras Shells. Es una mejora de Bash.
* Hay muchas shells, pero estas son unas de las más importantes.

En la Shell, si tenemos el símbolo $ estamos como usuario normal, si el símbolo es # estamos como administrador.

Nota: Los ejemplos los vamos a realizar con Kali Linux y en bash.

*VARIABLES DEL ENTORNO*

Lo primero, para saber que variables de entorno tenemos en este momento utilizamos el comando:

Si queremos ver el valor que tiene una variable, por ejemplo, si tenemos una variable llamada HOME y queremos saber su valor, utilizaríamos el comando:

Si queremos definir una variable, por ejemplo, queremos crear una variable llamada pentágono con un valor de 5:

De la misma manera nos sirve para modificar el valor de una variable ya existente.

Nota: Estas variables son temporales, se borran al apagar el equipo.

Si queremos mantener el valor de la variable, pero añadiendo algo adicional:

Para eliminar una variable utilizamos:

**¿QUE VARIABLES DE ENTORNO VIENEN POR DEFECTO EN EL SISTEMA?** Vamos a repasar algunas de las más importantes:

* PATH: especifica las rutas en las que el sistema busca los ejecutables.
* HOME: Especifica la ruta home del usuario que la solicite.
* USER: Especifica el nombre del usuario actual.
* SHELL: Especifica la Shell que estamos usando.
* TERM: Especifica el terminal que estamos usando.
* EDITOR: Especifica el editor de texto predeterminado.
* LANG: Especifica el idioma y localización.

**Crear una variable persistente**

Para hacerlo debemos añadirla de forma manual en el fichero de la Shell que utilicemos.

* .bashrc: Si estamos usando el bash.
* .zshrc: Si estamos usando la Shell zsh.

Dentro del documento creamos la variable de la forma en que creamos una variable temporal.

*RUTAS*

* Rutas absolutas: Las que ponemos desde la raíz, la ruta completa.
* Rutas relativas: Las que utilizamos desde una ubicación en la que ya estamos, omitiendo los directorios que hay hasta llegar a nuestra ubicación.
* .. : Si utilizamos .. en la Shell, significa que nos dirigimos al directorio padre de donde nos encontramos.

## **COMANDOS**

Dentro de los comandos hay que diferenciar entre parámetros y argumentos. Los parámetros son aquellas instrucciones que modifican un comando, por lo general vienen con un guion delante. Por ejemplo -l delante de un comando.

Por otro lado, los argumentos son datos necesarios para que un comando pueda funcionar, pueden ser obligatorios u opcionales. Por ejemplo un /home con el comando ls, es opcional y nos listaría lo que hay en ese directorio.

Que utilicemos uno no significa que no podamos utilizar el otro, por ejemplo.

*COMANDOS DE AYUDA*

* MAN: Viene de manual y te dice que hace un determinado comando. Por ejemplo, si pongo man ls, nos indica que hace el comando ls.
* INFO: Similar a man, pero algunas veces puede dar más información.
* WHATIS: Nos provee una descripción breve del comando que consultemos. Por ejemplo,
* APROPOS: Ayuda a encontrar comandos en base de una palabra clave. Por ejemplo, si queremos encontrar comandos de directorio, pondríamos , Lo cual nos listaría los comandos relacionados a esta palabra. Es lo mismo que usar el comando
* WHEREIS: Localiza el archivo binario en la página de manual de un comando. Por ejemplo, si uso , nos da el directorio del binario de ese comando.
* WHICH: Te muestra la ruta del archivo ejecutable asociado a un comando que le pases. . En este archivo están las ordenes que se van a ejecutar cuando se usa ese comando.

La mayoría de comandos en Linux tienen este parámetro:

* -h o –help: ls -h. Nos dá información sobre el mismo.
* HELP: El comando help no es lo mismo que el parámetro help, con este comando podemos ver info de comandos que son únicos de la Shell BASH. No sirve en otra Shell que no sea BASH.

*COMANDOS DE NAVEGACIÓN Y GESTIÓN DE FICHEROS Y DIRECTORIOS*

* CD: Change directory. Cambiamos de directorio, admite tanto rutas relativas como absolutas.
* PWD: Imprime por pantalla el directorio donde te encuentras.
* LS: List. Lista los ficheros y directorios que hay dentro de un directorio. Cuenta con varios parámetros, los más usados son:
  + -a -> Muestra ficheros/directorios ocultos.
  + -l -> lista en formato largo, viendo así más detalles.
* TOUCH: Crea ficheros vacíos.

Tiene algunos parámetros como:

* + -a -> Modifica la hora de acceso al fichero, se usa en caso de que quieras cambiar la hora en la que accediste a un fichero.
  + -m -> Cambia la fecha de modificación.
* CP: Copy. Copia un fichero o directorio. Admite rutas absolutas o relativas.
  + -i -> Fuerza a pedir confirmación en caso de que haya un archivo con el mismo nombre, para no machacarlo.
  + -r -> Sirve para copiar directorios enteros, con todo lo que el contenga.
* MV: Move. Mueve ficheros de un sitio a otro. También sirve para cambiar el nombre de un fichero.
  + -i -> Pide confirmación en caso de que haya un archivo con el mismo nombre.
  + -u -> Sobrescribe el archivo si el archivo a sobrescribir es mas antiguo.
* RM: Remove. Elimina ficheros.
  + -i -> Pide confirmación antes de eliminar.
  + -r -> Se usa principalmente en directorios, cuando este tiene muchos otros directorios. Para borrarlo todo usamos este parámetro.
* MKDIR: Make Directory. Creas un Directorio.
  + -p -> Permite crear sucesiones de directorios, uno dentro de otro.
* RMDIR: Remove Directory. Elimina directorios.
  + –ignote-fail-on-non-empty -> Ignora los fallos cuando eliminamos un directorio que no está vacío.
* CAT: se usa para abrir un fichero. Pero realmente este comando es Concatenate y su origen fue para concatenar archivos. Ejemplo de abrir fichero:

Ejemplo concatenar:

+1+2)

* + -n -> Cuenta el número de líneas.
* HEAD: Muestra las primeras 10 filas del documento.
  + -nx -> muestra las líneas que le indiquemos de un fichero.
* TAIL: Lo contrario a HEAD. Lee las últimas 10 líneas. De la misma forma tiene el parámetro -n.
  + -f -> Con este parámetro seguimos el contenido de un fichero en tiempo real. Esto es útil para ir viendo los logs, que se van almacenando en las últimas líneas.
* LESS: Sirve para ver documentos de forma paginada, es decir, verlos por páginas. Muy útil para ficheros grandes.
  + -f -> Vemos el fichero en tiempo real.
  + -n -> Visualiza el número de líneas.
  + +g -> Muestra el final del fichero.

*Concepto stdin, stdout y stderr*

STD vienen de estándar y en Linux en comandos tenemos 3 estándar.

* STDIN: Standar IN: Se relaciona al número 0 y son todos los datos que son recibidos por un programa o comando de un usuario. Por ejemplo, lo que escribimos en el Shell para introducir un comando.
* SDTOUT: Standar OUT: Se relaciona al número 1 y es aquello que devuelve un programa o comando cuando se ejecuta correctamente al usuario o a otro programa. Por ejemplo, lo que nos devuelve el comando ls al ejecutarse correctamente.
* STDERR: Standar ERROR: Se relaciona al número 2 y es todo aquello que nos devuelve un programa o comando cuando no es ejecutado correctamente.

Los números que se relacionan con los estándares se utilizan para las redirecciones, es decir, para guardar lo que nos devolvería en un archivo. Se hace con STDOUT y STDERR Por ejemplo:

Con este comando guardaremos la lista que nos devolvería ls en archivo.txt, redirigiendo la información al archivo.

Ejemplo 2:

Esto redirigiría los mensajes de error al archivo.txt.

*Comandos de gestión de usuarios*

* USERADD y USERDEL: Añade y elimina usuarios, pero no están optimizados.
* ADDUSER: Crea usuarios en el directorio /etc/passwd. Y meteremos datos para rellenar la ficha del usuario. Se realiza con el comando sudo.
* DELUSER: Elimina un usuario del SO.

Con el parámetro --remove-home eliminamos también el directorio home de ese usuario.

* USSERMOD: Nos permite modificar las propiedades de un usuario. Tiene muchos parámetros y es un comando muy potente. Dos de los más importantes.
  + -l -> Permite cambiar el nombre de login del usuario.
  + -d -> Permite cambiar el directorio HOME del usuario.
* PASSWD: Cambia la contraseña de un usuario. Los parámetros más relevantes son:
  + -l -> Bloquea la contraseña del usuario, es como desactivar la cuenta de ese usuario.
  + -u -> Desbloquea la contraseña del usuario, es decir, reactiva la cuenta.
* ID: Nos identifica la información del usuario actual o del que le indiquemos.
  + -u -> Nos muestra el UserId (UID) del usuario.
  + -g -> Nos muestra el GroupId (GID) del usuario
* WHO: Nos permite saber quien está conectado al sistema.
  + -u -> Nos muestra una lista de los usuarios conectados y la hora de su conexión.
* SU: Switch User. Sirve para cambiar de usuario. Si no se especifica nombre de usuario se cambiará al usuario Root.
  + -l o --login -> Nos muestra un entorno similar al que tendría el usuario si se conectara.
  + -c -> Permite ejecutar un comando con los permisos del usuario que le indiquemos, sin tener que loguearnos con ese usuario.
* SUDO: Nos permite ejecutar comandos con permisos de otro usuario, es similar al comando SU con el parámetro -c. Si no ponemos usuario por defecto coge el usuario Root.
  + -u -> Especificamos el usuario con el que ejecutaremos el comando.
  + -i -> Proporciona entorno similar al que tendría el usuario si se conectara.

Este comando es más potente que SU.

* WHOAMI: Nos indica con que usuario estamos logueados.
* FINGER: Proporciona información sobre un usuario en particular, como cuando inicio sesión, etc…

*Comandos de gestión de usuarios*

* ADDGROUP: