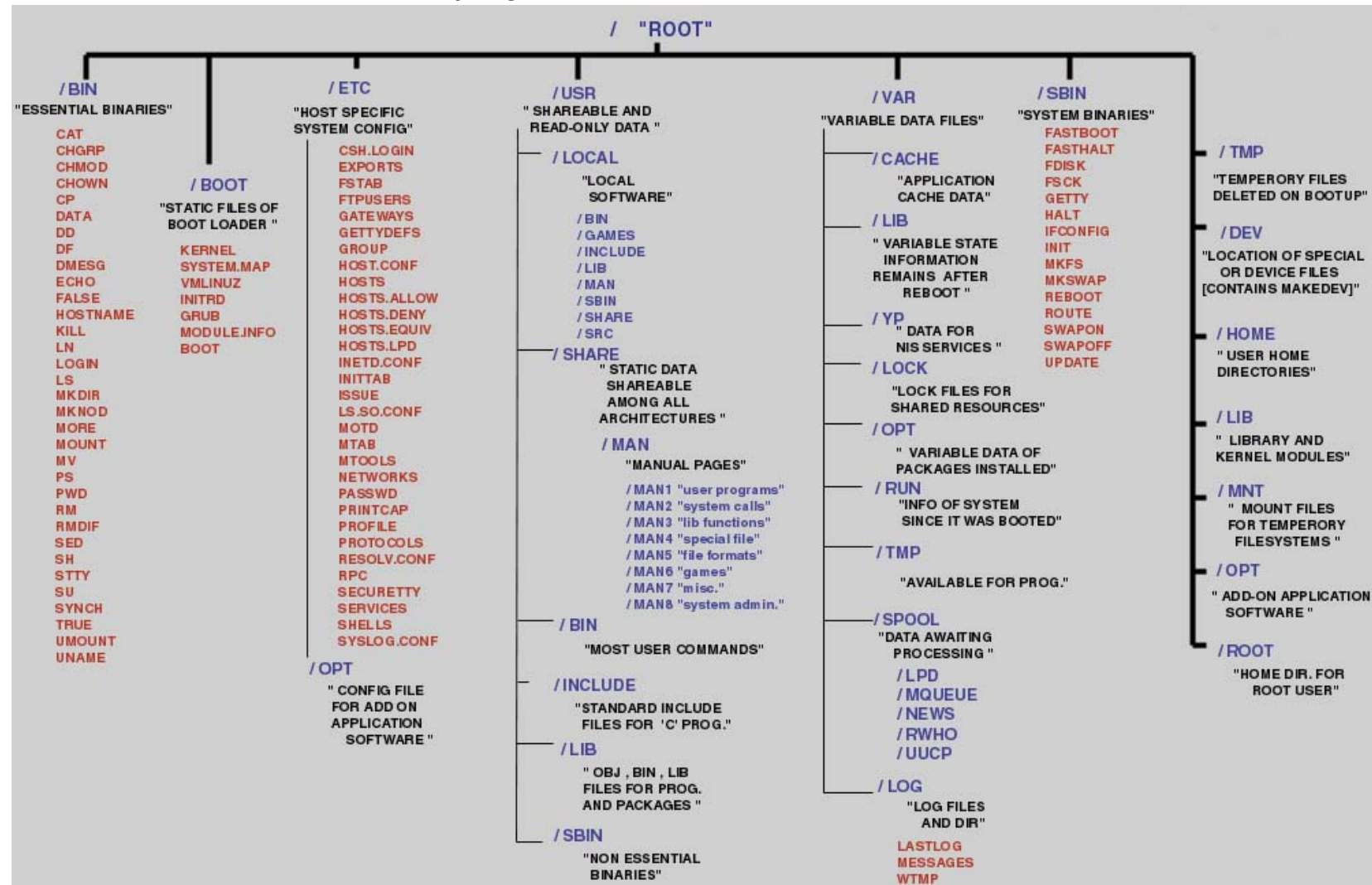


Estructura de directorios de GNU/Linux

La estructura de directorios que sigue **GNU/Linux** es parecida a la de cualquier UNIX. No existe una "unidad" para cada unidad física de disco o partición como en Windows, sino que todos **los discos duros o de red se montan bajo un sistema de directorios en árbol**, y algunos de esos directorios enlazan con estas unidades físicas de disco.



OBS: Las barras en Linux al igual que en cualquier UNIX son inclinadas hacia la derecha, como se puede ver más abajo (ese es el motivo de que en Internet sean inclinadas hacia la derecha ya que nació bajo UNIX y en Linux podremos aprovechar todas sus ventajas).

Directorios de GNU/Linux

Directorio	Descripción
/	Es la raíz del sistema de directorios. Aquí se monta la partición principal Linux EXT.
/etc	Contiene los archivos de configuración de la mayoría de los programas.
/home	Contiene los directorios personales de los usuarios.
/bin	Contiene comandos básicos y muchos programas.
/sbin	Contiene binarios y comandos de sistema.
/dev	Contiene archivos simbólicos que representan partes del hardware, tales como discos duros, memoria...
/mnt	Contiene subdirectorios donde se montan (se enlaza con) otras particiones de disco duro, CDROMs, etc.
/tmp	Archivos temporales o de recursos de programas.
/usr	Programas y librerías instalados con la distribución.
/usr/local	Programas y librerías instalados por el administrador.
/sbin	Comandos administrativos.
/lib	Librerías varias y módulos ("trozos") del kernel.
/var	Datos varios como archivos de log (registro de actividad) de programas, bases de datos, contenidos del servidor web, copias de seguridad...
/proc	Información temporal sobre los procesos del sistema.

Usuarios en GNU/Linux

Linux es un sistema operativo **multiusuario**. Cada usuario generalmente tiene su **directorio de usuario** en **/home/usuario**. Por defecto sólo puede **escribir, modificar y borrar** archivos dentro de este directorio. Ningún otro usuario (excepto root) puede acceder a los archivos que hay en este directorio, ni si quiera puede ver cuáles son.

Este usuario -por defecto- **puede leer en el resto de las directorios que hay en el sistema de archivos excepto en la de root y las de otros usuarios**. Todos los programas recuerdan las preferencias de cada usuario, un usuario puede instalar programas de manera restringida sin que los otros usuarios tengan acceso a él (excepto el usuario root). Por este motivo, un usuario “común” no puede cambiar la configuración del sistema ni causar daño alguno.

El usuario root (superusuario)

En cualquier sistema UNIX, **root es "el que todo lo puede"**. Es la excepción que confirma la regla, es el superusuario todopoderoso de estos sistemas. Cuando se hace login como root en una máquina **GNU/Linux**, se tiene poder absoluto sobre el sistema y sus usuarios. **Pero una equivocación... sería fatal**. Por esto, como regla general, **para tareas normales SIEMPRE se loguea al sistema como un usuario normal** por los riesgos que se corren trabajando como root. Incluso algunos programas no permiten ser ejecutados por root por motivos de seguridad.

Por todo lo anterior, la parte más segura para guardar la contraseña de root es en la cabeza ...y asegurarse de que no se le olvida, y por supuesto, se ocuparse de que nadie más la conozca o pueda acceder a ella. Si se cree que la han podido adivinar o están cerca, se cambia. Cuanto más larga, tediosa y sin sentido sea esta contraseña, más seguro estará nuestro sistema.

Una máquina Linux es tan segura como segura sea la contraseña de root.



Permisos en GNU/Linux

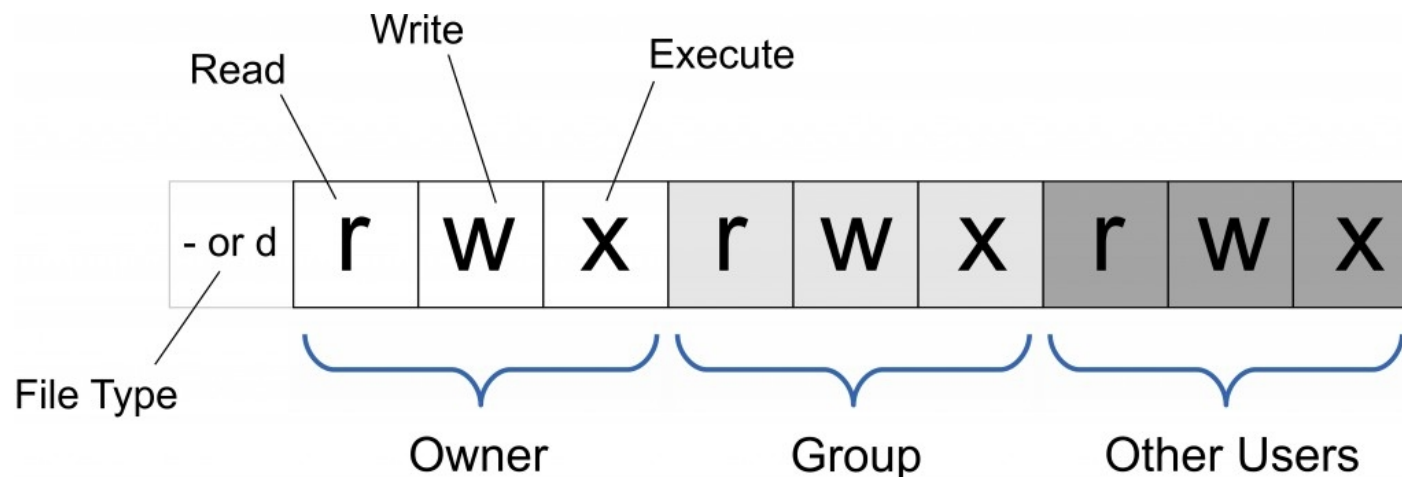
Todos y cada uno de los **archivos** y **directorios** del árbol jerárquico que monta nuestro sistema Linux tienen **permisos**. Estos permisos dicen, para cada usuario del sistema, si puede **ejecutarlo**, si puede ver su contenido (**leer**) o si puede borrarlo o modificarlo (**escribir**).

Del mismo modo, **cada elemento del sistema de archivos tiene un dueño**. Por defecto, el dueño del elemento (tanto directorio como archivo) **corresponde al usuario que lo ha creado**, y tiene acceso total a él y puede realizar todas las acciones posibles permitidas. El resto de usuarios pueden leer y ejecutar este elemento por defecto aunque todo esto se puede cambiar para cada uno de los elementos.

Todos los archivos de las carpetas de sistema y configuración suelen tener a root como propietario. Los de la carpeta personal de cada usuario tienen a ese usuario como propietario, pero el resto de usuarios normales no tienen ningún permiso sobre estos elementos, y lo mismo ocurre con la carpeta de root (que se encuentra en la raíz, en /root).

Los permisos de un archivo/directorio en Linux se denotan como:

r	Lectura
w	Escritura
x	Ejecución



Comandos básicos GNU/Linux para operar con archivos

El comando chmod

Este comando se utiliza para **asignar/modificar los permisos de archivos y directorios**.

El comando asigna los permisos mediante un número de 3 cifras que corresponden a la representación decimal del número binario que se obtiene al generar cada grupo de permisos, esto es como sigue:

Permiso	Permisos del Usuario propietario			Permisos del Grupo propietario			Permisos del Grupo		
	r	w	x	r	w	x	r	w	x
Valor Binario	2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
Valor Decimal	4	2	1	4	2	1	4	2	1

Lectura	Escritura	ejecución	Valor Octal
r	w	x	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Ejemplo:

Se tiene el archivo “texto”, al cual se quiere dar los siguientes permisos:

1. Lectura, escritura y ejecución para su dueño.
2. Lectura y ejecución para el grupo del usuario dueño.
3. Sólo lectura para el resto de los usuarios y grupos.

El comando chmod sería de la forma:

chmod 754 texto

`chmod {permisos del dueño}{permisos del grupo}{permisos del resto}
nombre_de_archivo`

Comando	Acción	Comentarios
ls	Lista los archivos de un directorio concreto	ls -l = Añade información sobre los atributos
		ls -a = Lista todos los archivos incluyendo los ocultos
		ls -R = Lista el contenido del directorio y todos sus subdirectorios recursivamente
cat [Archivo]	Muestra el contenido de un Archivo	cat -n [Archivo] = Muestra el contenido de un Archivo numerando sus líneas
more [Archivo]	Muestra un Archivo de forma tabulada como una página del man	
less [Archivo]	Similar a more , permite tanto avanzar como retroceder entre las páginas	
tail [archivo]	Muestra las ultimas lineas de un archivo	tail -n [n] [archivo] = Muestra las ultimas n lineas del archivo
head [archivo]	Muestra las primeras lineas de un archivo	head [n] [archivo] = Muestra las <i>n</i> primeras lineas de un archivo
find [patron]	Busca las coincidencias con el patrón dentro del directorio y sus subdirectorios	find [ruta] [patron] = Busca las coincidencias con el patron dentro de la ruta
		find [patron] -print = Busca las coincidencias y muestra la ruta completa de estas.
		find -size [tam] = Busca aquellos archivos menores que el tamaño señalado
		man find = Muestra un listado las múltiples opciones y usos de find
locate [patron]	Busca las coincidencias con el patrón dentro de una base de datos indexada.	updatedb Actualiza la base de datos de archivos y directorios.
whereis [programa]	Busca la ruta donde se encuentra el programa, su ayuda, etc.	
grep [opciones] [expresión regular] nombredearchivo	Toma una expresión regular de la línea de comandos, lee la entrada estándar o una lista de archivos, e imprime las líneas que contengan coincidencias para esa expresión regular. También podemos redireccionar la salida de un determinado comando por medio de una tubería (Alt Gr+ 1) y filtrarla con grep usando la expresión regular como criterio.	grep cadena archivo = Muestra las líneas del archivo que contienen la cadena grep palabra * = Muestra todas las líneas que contienen la cadena “palabra” en una lista de archivos (donde “*” representa todos los archivos en el directorio actual) ps -ef grep nombredeproceso = Lista los procesos que se estén ejecutando y que contengan la cadena <i>nombredeproceso</i>

pwd	Muestra el directorio actual	
cd	Cambia al home o al directorio raíz si se lanza como root	cd [ruta] = Se desplaza al directorio especificado en la ruta
		cd .. = Se mueve al directorio anterior
		cd ../.. = Se mueve dos directorios atrás.
cp [origen] [destino]	Copia el archivo origen al directorio destino	cp -R [origen] [destino] = Copia un directorio recursivamente (directorio y subdirectorios)
		cp -p [origen] [destino] = Copia preservando los permisos y las fechas
		cp [archivo] [archivo nombre cambiado] = Copia el archivo y lo cambia de nombre
mv [origen] [destino]	Mueve al archivo origen al directorio destino	mv [archivo] [archivo_nombre_cambiado] = Cambia de nombre un archivo
		mv se parece a cp , excepto que borra el archivo original después de copiarlo. Se puede entender como la utilización conjunta de cp y rm .
		mv renombra un archivo (que es el primer parámetro) si el segundo parámetro es un archivo. Si el segundo parámetro es un directorio, mv moverá el archivo al nuevo directorio, manteniendo el mismo nombre simple.
mkdir [directorio]	Crea un nuevo directorio dentro del directorio.	
rm [archivo]	Elimina un archivo completamente	rm -r [directorio] = Elimina un directorio recursivamente (directorio y subdirectorios)
man comando	Muestra el manual de un comando, un paquete, etc. Resulta muy útil para aprender a utilizar dicho comando, sus modificadores y argumentos.	man aptitude = Nos despliega la ayuda de aptitude , sus opciones y posibilidades.