

Tema adicional:

Administración de Archivos



Presentación

En esta cuarta unidad adicional, se enseñará lo que hay que tener en cuenta referido al manejo de archivos y estructura de directorios.



Objetivos

Que los participantes logren...

- Conocer los conceptos generales de configuración apropiada de un sistema operativo Centos.



Bloques temáticos

1. Administración de archivos
2. Principales directorios
3. Rutas Relativas y Absolutas
4. Permisos de archivos y directorios

Administración de Archivos (en una virtual)

¿Qué es un sistema de archivos?

Un sistema de archivos son los métodos y estructuras de datos que un sistema operativo utiliza para seguir la pista de los archivos de un disco o partición; es decir, es la manera en la que se organizan los archivos.

El sistema de archivos de linux es una organización jerárquica que se asemeja a un árbol donde cada hoja es un directorio o un archivo.

Originariamente, en los inicios de Linux, este árbol de directorios no seguía un estándar cien por cien, es decir, podíamos encontrar diferencias en él de una distribución a otra.

Todo esto hizo pensar a cierta gente que, posteriormente, desarrollarían el proyecto FHS (Filesystem Hierarchy Standard), o lo que es lo mismo: Estándar de Jerarquía de Sistema de archivos) en 1993.

FHS no es más que un documento guía, es decir, cualquier fabricante de software independiente o cualquier persona que decida crear una nueva distribución GNU/Linux, podrá aplicarlo o no a la estructura del sistema de archivos, con la ventaja de que si lo integra en el sistema, el entorno de éste será mucho más compatible con la mayoría de las distribuciones.

Es importante saber que el estándar FHS es en cierto modo flexible, es decir, existe cierta

libertad en el momento de aplicar las normas. De ahí que existan en la actualidad leves diferencias entre distribuciones GNU/Linux.

Objetivos principales de FHS

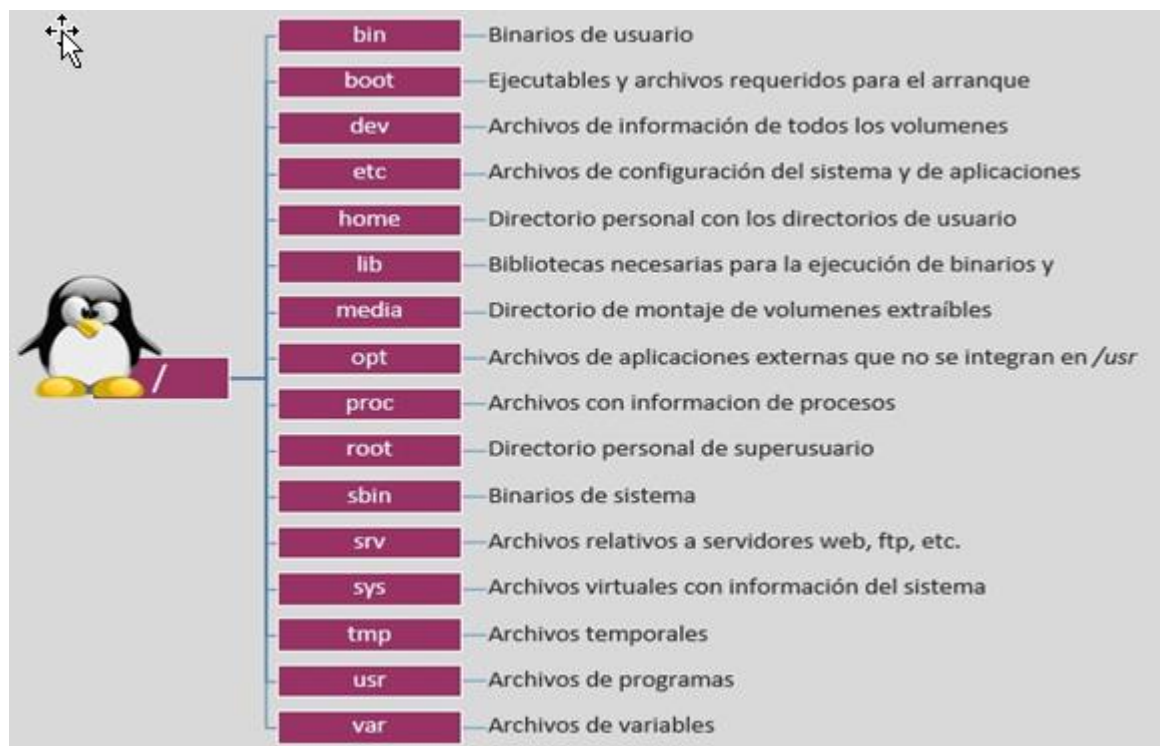
- Presentar un sistema de archivos coherente y estandarizado.
- Facilidad para que el software prediga la localización de archivos y directorios instalados.
- Facilidad para que los usuarios prediga la localización de archivos y directorios instalados.
- Especificar los archivos y directorios mínimos requeridos.

El estándar FHS está enfocado a:

Fabricantes de software independiente y creadores de sistemas operativos, para que establezcan una estructura de archivos lo más compatible posible.

Usuarios comunes, para que entiendan el significado y el contenido de cada uno de los elementos del sistema de archivos.

Principales Directorios y sus Funciones



La barra inclinada “/”

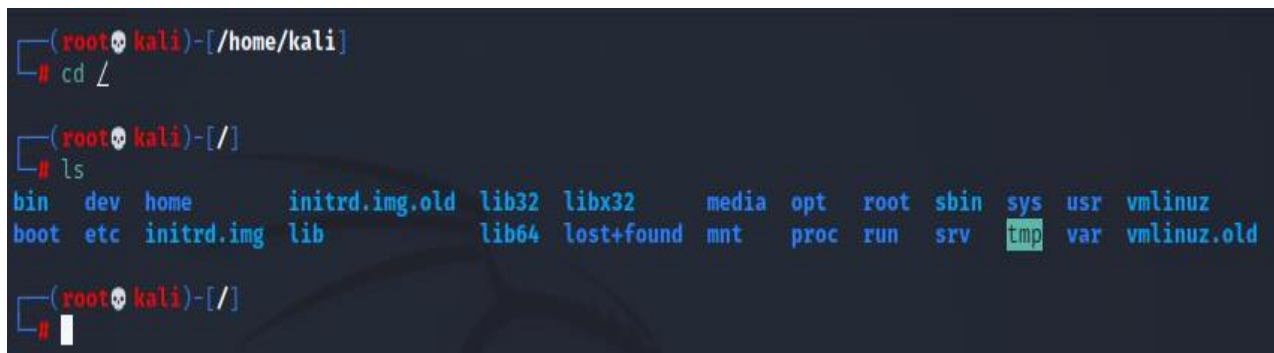
Toda la estructura de directorios en los sistemas basados en UNIX parte de un directorio raíz también llamado directorio **root** y que se simboliza por una barra inclinada o /.

De este directorio, es desde donde nacen todo el resto de directorios, independientemente que estén almacenados físicamente en discos o unidades separadas.

Cualquier dirección de archivo o carpeta en Linux empieza por el directorio raíz o /, seguido de todos los directorios y subdirectorios que lo contienen, separados cada uno de ellos por /.

A continuación se conocerá con más en detalle a todos los directorios principales que parten del directorio raíz, junto con sus subdirectorios más importantes y los archivos que suelen contener.

Con el comando “ls” dentro de la consola, se puede listar directorios, pero para llegar al directorio raíz, hay que utilizar el parámetro “cd /”



```
(root@kali)~/home/kali
# cd /

(root@kali)~/[ ]
# ls
bin  dev  home  initrd.img.old  lib32  libx32  media  opt  root  sbin  sys  usr  vmlinuz
boot  etc  initrd.img  lib  lib64  lost+found  mnt  proc  run  srv  tmp  var  vmlinuz.old

(root@kali)~/[ ]
#
```

/bin

El directorio /bin es un directorio estático y es donde se almacenan todos los binarios necesarios para garantizar las funciones básicas a nivel de usuario.

Solo almacena los ejecutables de usuario, ya que los binarios necesarios para tareas administrativas gestionadas por el usuario root o superusuario del sistema se encuentran en el directorio /sbin.

```
(root@kali)-[/bin]
# ls
['2to3-2.7', '7z', '7za', '7zr', 'aa-enabled', 'aa-exec', 'ab', 'acyclic', 'addpart', 'addr2line', 'agentxtrap', 'aircrack-ng', 'airdecap-ng', 'innochecksum', 'innotop', 'install', 'instmodsh', 'inv', 'invoke', 'ionice', 'iostat', 'ip', 'ipcmk', 'ipcrm', 'ipcs', 'iptables-xml', 'ipwhois_cli.py', 'python3', 'python3.9', 'python3.9-config', 'python3-config', 'python3-futureize', 'python3-pasteurize', 'python3-qr', 'python3-tor-prompt', 'python3-wsdump', 'python-faraday', 'pyuic5', 'pyversions', 'pywerview', 'qcollectiongenerator']
```

Imagen de contenido de /bin

```
(root@kali)-[/]
# cd sbin

(root@kali)-[/sbin]
# ls
a2disconf, a2dismod, a2dissite, a2enconf, a2enmod, a2ensite, a2query, aa-remove-unknown, aa-status, aa-teardown, accessdb, addgnupghome, addgroup, add-shell, adduser, agetty, fsck, fsck.cramfs, fsck.exfat, fsck.ext2, fsck.ext3, fsck.ext4, fsck.fat, fsck.minix, fsck.msos, fsck.vfat, fsfreeze, fstab-decode, fstrim, ftl_check, ftl_format, gdisk, mkfs.bfs, mkfs.cramfs, mkfs.exfat, mkfs.ext2, mkfs.ext3, mkfs.ext4, mkfs.fat, mkfs.jffs2, mkfs.minix, mkfs.msos, mkfs.ntfs, mkfs.ubifs, mkfs.vfat, mkhomedir_helper, mkinitramfs, mklost+found, samba_upgradedns, sampasswd, samusrgrp, saned, scrounge-ntfs, select-default-ispell, select-default-wordlist, serve_image, service, setcap, setvesablank, sfdisk, sgdisk, shadowconfig, showmount, shutdown]
```

Imagen de contenido de /sbin

Incluye también los binarios que permiten la ejecución de varias utilidades estándar de la terminal de Linux, concretamente **cat**, **cd**, **cp**, **echo**, **grep**, **gzip**, **kill**, **ls**, **mv**, **rm**, **ping**, **su**, **ps**, **tar** y **vi**.

/boot

Es un directorio estático e incluye todos los ejecutables y archivos que son necesarios en el proceso de arranque del sistema, y que deberán ser utilizados antes que el kernel empiece a dar las órdenes de ejecución de los diferentes módulos del sistema.

Es también donde se encuentra el gestor de arranque GRUB.

En algunas distribuciones, es común que ese directorio se almacene en su propia partición

separada del resto. Esto suele darse sobre todo en el caso de de que utilicen **LVM (Logical Volume Manager)** por defecto, ya que tradicionalmente el gestor de arranque GRUB (en versiones anteriores a la 2) no podía arrancar desde LVM, por lo que se requería que estuviera en una partición separada.

De hecho, si en una instalación normal de Ubuntu o Debian optas por utilizar LVM, verás que el instalador ya te genera un esquema de particiones con el directorio boot en una partición aparte.

En estos casos, en el momento de instalar el sistema es importante prever bien el espacio que le vayas a dar a la partición, ya que a la larga, con la acumulación de diferentes actualizaciones del Kernel, es común que se quede sin espacio.

Si esto sucede, puedes tener problemas a la hora de instalar futuras actualizaciones del núcleo, y debes hacer limpieza de versiones antiguas del kernel.

```
(root@kali)-[/sbin]
# cd /

(root@kali)-[/]
# cd boot

(root@kali)-[/boot]
# ls
config-5.10.0-kali7-amd64  initrd.img-5.10.0-kali7-amd64  System.map-5.10.0-kali8-amd64
config-5.10.0-kali8-amd64  initrd.img-5.10.0-kali8-amd64  vmlinuz-5.10.0-kali7-amd64
grub                      System.map-5.10.0-kali7-amd64  vmlinuz-5.10.0-kali8-amd64
```

/dev

Este directorio incluye todos los dispositivos de almacenamiento, en forma de archivos, conectados al sistema, es decir, cualquier disco duro conectado, partición, memoria USB, o CDROM conectado al sistema y que el sistema pueda entender como un volumen lógico de almacenamiento.

Siendo esto así, verán que la ruta en la que se encuentra cualquier volumen (partición o dispositivo externo) conectado al sistema siempre empieza por /dev.

Este es el directorio que contiene, por decirlo de algún modo, la información de cada uno de los volúmenes, a diferencia del directorio /media, que verás más adelante, que lo que contiene son solo los puntos de montaje, pero no la información real de estos volúmenes.

Para ver esto en la práctica, si se abre una ventana de terminal y se ejecuta el comando “fdisk -l” (como super usuario), se visualizará la estructura de particiones de tu sistema.

Eso en cuanto a particiones. Si se trata de un dispositivo externo, el volumen estará igualmente dentro de /dev, pero en este caso varía el nombre que el sistema le asigna a dicho volumen.

```
(root@kali)-[/boot]
# cd /

(root@kali)-[/]
# cd dev

(root@kali)-[/dev]
# ls
```

autofs	fb0	mqueue	sda1	tty1	tty22	tty35	tty48	tty60	vboxguest	vcsa3	vcsu8
block	fd	net	sda2	tty10	tty23	tty36	tty49	tty61	vboxuser	vcsa4	vfi
bsg	full	null	sda5	tty11	tty24	tty37	tty5	tty62	vcs	vcsa5	vga_arb
btrfs-control	fuse	nvr	sg0	tty12	tty25	tty38	tty50	tty63	vcs1	vcsa6	vhci
bus	hidraw0	port	sg1	tty13	tty26	tty39	tty51	tty7	vcs2	vcsa7	vhost-v
cdrom	hpet	ppp	shm	tty14	tty27	tty4	tty52	tty8	vcs3	vcsa8	vhost-v
char	hugepages	psaux	snapshot	tty15	tty28	tty40	tty53	tty9	vcs4	vcsu	zero
console	initctl	ptmx	snd	tty16	tty29	tty41	tty54	ttyS0	vcs5	vcsu1	
core	input	pts	sr0	tty17	tty3	tty42	tty55	ttyS1	vcs6	vcsu2	
cpu_dma_latency	kmsg	random	stderr	tty18	tty30	tty43	tty56	ttyS2	vcs7	vcsu3	
cuse	log	rfskill	stdin	tty19	tty31	tty44	tty57	ttyS3	vcs8	vcsu4	
disk	loop-control	rtc	stdout	tty2	tty32	tty45	tty58	uhid	vcsa	vcsu5	
dri	mapper	rtc0	tty	tty20	tty33	tty46	tty59	uinput	vcsa1	vcsu6	
dvd	mem	sda	tty0	tty21	tty34	tty47	tty6	urandom	vcsa2	vcsu7	

/etc

Es el encargado de almacenar los archivos de configuración tanto a nivel de componentes del sistema operativo en sí, como de los programas y aplicaciones instaladas a posteriori.

Es un directorio que debería contener únicamente archivos de configuración, y no debería contener binarios.


```
(root@kali)~# ls
adduser.conf      debian_version  gtk-3.0         libaudit.conf   nanorc          ppp             sddm.conf.d     sv
adjtime           debtags         guymager        libblockdev     netconfig       profile         searchsploit_rc sysctl.conf
alsa             default        hdparm.conf     libl-3          netsniff-ng     profile.d       security         sysctl.d
alternatives      deluser.conf   host.conf       libpaper.d      network         protocols       selinux         sysstat
apache2           dhcp           hostname        lightdm         NetworkManager proxychains4.conf sensors3.conf   systemd
apparmor          dictionaries-common hosts           lighttpd        networks        pulse          sensors.d       terminfo
apparmor.d        dns2tcpd.conf hosts.allow     locale.alias    nftables.conf  python2.7      services        theHarvester
apt              dpkg           hosts.deny      locale.gen      nginx           python3        shadow          tightvncserver.conf
avahi            e2scrub.conf  idmapd.conf     localtime       nikto.conf     python3.9      shadow-         timezone
bash.bashrc       emacs          ifplugd         logcheck        nsswitch.conf  rc0.d          shells          timidity
bash_completion   environment    inetsim         login.defs      ntp.conf       rc1.d          skel            tmpfiles.d
bash_completion.d environment.d    init.d          logrotate.conf ODBCDataSources rc2.d          smartd.conf     ucf.conf
bindresvport.blacklist ethertypes     initramfs-tools logrotate.d      odbc.ini       rc3.d          smartmontools  udev
binfmt.d          ethtoolcap     inputrc         macchanger      openal          rc4.d          smi.conf        udisks2
bluetooth         firebird       insserv.conf.d  machine-id      OpenCL          rc5.d          snmp            ufw
ca-certificates   firefox-esr    ipp-usb         magic           openfortivpn   rc6.d          speech-dispatcher updatedb.conf
ca-certificates.conf fonts           iproute2        magic.mime      openni2         rcS.d          sqlmap          update-motd.d
chatscripts       freetds        ipsec.conf      mailcap         openvpn         rearj.cfg      ssh             UPower
cifs-utils        fstab          ipsec.d         mailcap.order   opt            redsocks.conf  ssl            vdpau_wrapper.cfg
cloud            fuse.conf      ipsec.secrets  manpath.config  os-release     request-key.conf sslsplit        vim
console-setup     gai.conf       issue           matplotlibrc    pam.conf       request-key.d  strongswan.conf vpnc
cron.d            geoclue        issue.net       mime.types      pam.d          resolv.conf    strongswan.d   vulkan
cron.daily        ghostscript    java-11-openjdk minicom         papersize      responder      stunnel        wgetrc
cron.hourly       glvnd         john           miredo          passwd         subgid         subuid         wifreshark
cron.monthly      groff          kernel         miredo.conf    mke2fs.conf   subuid         subuid         wpa_supplicant
crontab           group          kernel-img.conf mke2fs.conf     modprobe.d     subuid         subuid         X11
cron.weekly       group-         king-phisher    modules          modules-load.d polkit-1       sane.d          xattr.conf
cryptsetup-ntke-password grub.d         kismet         motd            mtab           postgresql     scalpel        xfce4
crypttab          gshadow       ld.so.cache     motd            mysql          postgresql-common screenrc        sudo.conf      xl2tpd
dbus-1            gshadow-      ld.so.conf      mtabs           libao.conf     powershell-empire sudo_logsvd.conf zsh
dconf             gss           libao.conf      mysqld           debconf.conf   gss            ld.so.conf.d   zsh_command_not_found
```

/home

Es el directorio de los usuarios estándar, y por lo tanto, el destinado a almacenar todos los archivos del usuario, como documentos, fotos, vídeos, música, plantillas, etc. También incluye archivos temporales de aplicaciones ejecutadas en modo usuario, que sirven para guardar las configuraciones de programas, etc.

Dentro /home están los directorios personales de todos los usuarios, nombrados según el nombre de usuario utilizado.

Cada directorio de usuario contiene asimismo diferentes carpetas para ayudarlo a clasificar la información. Estas generalmente son: /Documentos, /Imágenes, /Música, /Plantillas y /Vídeos, así como otros archivos y carpetas ocultas, que son las encargadas de guardar la información de configuraciones de las aplicaciones del usuario.

Por cierto, y muy importante, todos los archivos y carpetas ocultas en Linux empiezan por un punto, seguido del nombre de la carpeta.

En muchas distribuciones es una práctica recomendada el hecho de ubicar el directorio

/home es una partición separada del resto, por tal de facilitar que, en caso de reinstalar el sistema operativo, puedas mantener intacta la partición de la /home, y de este modo mantener todos los archivos personales.

```
(root@kali)-[/home]
# ls
kali

(root@kali)-[/home]
# cd kali

(root@kali)-[/home/kali]
# ls
Desktop Documents Downloads Music Pictures Public Templates Videos
```

/lib

Incluye las bibliotecas esenciales que son necesarios para que se puedan ejecutar correctamente todos los binarios que se encuentran en los directorios /bin y /sbin, así como los módulos del propio kernel.

En los sistemas operativos de 64 bits, además de /lib existe otro directorio denominado /lib64, referida a las bibliotecas para aplicaciones de 64 bits.

```
(root@kali)-[/lib]
# ls
apache2          ettercap          kernel            libmhfdfalt.so.0  ogdi              security
apparmor          file              klibc             libmhfdfalt.so.0.0.0  openssl           sftp-server
apt              firefox-esr       klibc-YUKGbOClhnaZRiUd4cUed0X2XZI.so  libogdi.so.4        openvpn           speech-dispatcher-modules
aspell           firewallld        ld-linux.so.2     libogdi.so.4.1       oracle            ssl
atril            firmware          libarmadillo.so.10  libqscintilla2_qt5.so.15  os-prober        starkiller
bfd-plugins      galera            libarmadillo.so.10.6.2  libqscintilla2_qt5.so.15.0  os-probes        sudo
binfmt.d         gcc              libblosc.so.1       libqscintilla2_qt5.so.15.0.0  os-release       sysctl.d
binfmt-support   git-core          libblosc.so.1.20.1     libsupp.a            p7zip            sysstat
blt2.5           gnome-settings-daemon-3.0  libBLT.2.5.so.8.6     libvfp.so.4          pam.d            systemd
cgi-bin          gnupg            libBLTlite.2.5.so.8.6  libvfp.so.4.1        passing-the-hash  sysusers.d
clang            gnupg2           libdfalt.so.0        linux-boot-probes    php              taskset
cmake            gold-ld          libdfalt.so.0.0.0     llvm-11              pkgconfig        tc
cnf-update-db    groff            libdlrpc.so          llvm-9               pm-utils         tcltk
command-not-found  grub             libbettercap.so       locale               policykit-1      terminfo
compat-ld         grub-legacy      libbettercap.so.0     lsb                  policykit-1-gnome  tmpfiles.d
console-setup     gvfs             libbettercap.so.0.8.3.1  man-db               postgresql        udev
cpp              hashcat          libbettercap-ui.so    mime                 pppd             udisks2
crda             hashcat-utils    libbettercap-ui.so.0   modprobe.d           pulse-14.2        unity-settings-daemon-1.0
cryptsetup       hdparm          libbettercap-ui.so.0.8.3.1  modules              python2.7         valgrind
cryptsetup-ntke-password  ifupdown        libgalera_smm.so      modules-load.d       python3           virtualbox
cups             init             libgdal.so.29         mozilla              python3.10        X11
dbus-1.0         ipsec           libgdal.so.29.0.2     mysql               python3.9         x86_64-linux-gnu
debug           ispell          libguytools.so.2       NetworkManager       qt5              xorg
dplg            java-wrappers    libguytools.so.2.1.0   nginx               rsyslog
emacs-common     john            libhotpatch.a         nodejs              ruby
engrampa         jvm            libhotpatch.so        notification-daemon  runit-helper
environment.d    kali_tweaks     libhotpatchtest.so    ntp                 sasl2
```

/media

Representa el punto de montaje de todos los volúmenes lógicos que se montan temporalmente, ya sean unidades externas USB, otras particiones de disco, etc.

En la mayoría de distribuciones GNU/Linux, desde hace ya algún tiempo, cada vez que se monta una unidad externa, partición, etc., esta se monta dentro del directorio /media y a su vez dentro de un directorio específico dependiendo del usuario del sistema que monta el volumen.

/mnt

Es un directorio vacío que cumple funciones similares a /media, pero que actualmente no se suele utilizar, ya que la mayoría de distribuciones hacen uso de este último para los puntos de montaje temporales.

/opt

En cierto modo vendría a ser como una extensión del directorio /usr, pero en este caso van todos aquellos archivos de solo lectura que son parte de programas auto-contenidos y que, por lo tanto, no siguen los estándares de almacenar los diferentes archivos dentro de los diferentes subdirectorios de /usr (que sería lo recomendable) Haciendo una analogía con Windows, vendría a ser algo como el directorio de “Archivos y Programas”, pero en este caso, como hemos dicho, para determinados programas que ya vienen auto-contenidos.

/proc

Este directorio contiene información de los procesos y aplicaciones que se están ejecutando en un momento determinado en el sistema, pero realmente no guarda nada como tal, ya que lo que almacena son archivos virtuales, por lo que el contenido de este directorio es nulo.

Básicamente son listas de eventos del sistema operativo que se generan en el momento de acceder a ellos, y que no existen dentro del directorio como tales.

```
(root@kali)~# ls /proc
```

1	1078	1145	12	130	17	25	29943	419	540	73	838	892	948	acpi	dma	kallsyms	misc	slabinfo	version
10	1083	1146	121	131	1707	26	29957	471	55	74	847	9	951	asound	driver	kcore	modules	softirqs	vmallocinfo
1005	10857	1147	122	132	178	27	29958	474	555	748	848	903	952	buddyinfo	dynamic_debug	keys	mounts	stat	vmstat
1007	1093	116	12241	13205	179	28	29959	475	556	753	852	905	953	bus	execdomains	key-users	mtrr	swaps	zoneinfo
1010	1098	11833	123	133	18	29	29962	476	57	754	853	908	954	cgroups	fb	kmsg	net	sys	
1017	11	11834	125	134	2	29474	29981	49	58	773	856	913	955	cmdline	filesystems	kpagecgroup	pagetypeinfo	sysrq-trigger	
1018	1106	11837	127	135	20	29775	3	50	59	774	857	928	970	consoles	fs	kpagecount	partitions	sysvipc	
1025	1120	11838	128	136	23	2979	30	51	6	775	868	931	976	cpuinfo	interrupts	kpageflags	pressure	thread-self	
1050	1125	11839	129	14	24	29837	31	52	60	777	878	936	98	crypto	iomem	loadavg	sched_debug	timer_list	
1070	1128	11840	12996	15	242	29884	4	53	690	786	883	940	991	devices	ioports	locks	schedstat	tty	
1074	114	119	13	16	243	29942	416	536	70	836	887	945	998	diskstats	irq	meminfo	self	uptime	

/root

Vendría a ser como el directorio /home del usuario root o superusuario del sistema. A diferencia de los otros usuarios, que se encuentran todos dentro de /home en sus respectivas subcarpetas, el directorio del usuario root está en su propia carpeta colgando directamente de la raíz del sistema.

Como es de esperarse solamente el usuario root puede ver su contenido y modificarlo.

```
(kali@kali)-[~]
$ cd /root
cd: permission denied: /root
```

/sbin

Si hemos dicho que en /bin se almacenaban los binarios relativos a las funciones normales de usuario, /sbin hace lo mismo pero para los binarios relativos tareas propias del sistema operativo, y que solamente pueden ser gestionadas por el usuario root, tales como el arranque, tareas de restauración, reparación, etc.

```
(root@kali)-[/sbin]
# ls
a2disconf          dmsetup           httxt2dbm         mkfs.cramfs       pwck              tkiptun-ng
a2dismod           dmstats           hwclock           mkfs.exfat        pwconv            traceroute
a2dissite          docfdisk          hwloc-dump-hwdata mkfs.ext2          pwunconv          trafgen
a2enconf           doc_loadbios      i2cdetect         mkfs.ext3          racf2john         tune2fs
a2enmod            dosfsck           i2cdump           mkfs.ext4          rar2john           tune.exfat
a2ensite           dosfslabel        i2cget            mkfs.fat           rarp              tzconfig
a2query            dpkg-fsys-usrunmess i2cset            mkfs.jffs2         raw               uaf2john
aa-remove-unknown  dpkg-preconfigure i2c-stub-from-dump mkfs.minix         raw2dyna          ubiattach
aa-status          dpkg-reconfigure  i2ctransfer       mkfs.msdos         readprofile       ubiblock
aa-teardown        dumpe2fs          iconvconfig       mkfs.ntfs          reboot            ubircrc32
accessdb           dump.exfat         ifconfig          mkfs.ubifs         recv_image        ubidetach
addgnupghome       e2freefrag        ifdown           mkfs.vfat          redsocks          ubiformat
addgroup           e2fsck            ifpps            mkhomedir_helper  reged             ubihealthd
add-shell          e2image           ifquery          mkinitramfs        remove-default-ispell ubimkvvol
adduser            e2label           ifup             mklost+found       remove-default-wordlist ubinfo
agetty             e2mmpstatus       iio-sensor-proxy mkntfs             remove-shell      ubinize
airbase-ng         e2scrub           init             mkntfs             request-key       ubirename
aireplay-ng        e2scrub_all       insmod           mkswap             resize2fs         ubirmvol
airmon-ng          e2undo            installkernel    mkvcalcproma       responder         ubirsvol
airodump-ng        e4crypt           integritysetup    ModemManager       responder-DHCP_Auto ubiupdatevol
airodump-ng-oui-update e4defrag          in.tftpd          modinfo            rfdformat         umount.nfs
airserv-ng         easside-ng        invoke-rc.d       modprobe            rfdformat         umount.nfs4
airtun-ng          ebtables          iodine            mount.cifs          rfkill            umount.udisks2
airventriloquist-ng ebtables-nft       iodine-client-start mount.fuse          rmmod             unafs
apache2            ebtables-nft-restore iodined           mount.fuse3         rmt               undrop
apache2ctl         ebtables-nft-save  ip               mount.lowntfs-3g    rmt-tar           unique
apachectl          ebtables-restore   ip6tables        mount.nfs           route             unix_chkpwd
apparmor_parser    ebtables-save      ip6tables-apply  mount.nfs4          rpcbind           unix_update
apparmor_status    ethtool           ip6tables-legacy mount.nfs            rpcdebug          unshadow
applygnupgdefaults eventlogadm        ip6tables-legacy-restore mount.ntfs-3g      rpc.gssd          update-binfmts
arp                exfatlabel        ip6tables-legacy-save mount.smb3          rpc.idmapd        update-ca-certificates
arpd               faillock          ip6tables-nft    mountstats          rpcinfo           update-command-not-found
```

/srv

Sirve para almacenar archivos y directorios relativos a servidores que puedas tener instalados dentro de tu sistema, ya sea un servidor web www, un servidor FTP, CVS, etc. Así, por ejemplo, en el caso de tener instalado un servidor web, sería buena idea tener el directorio web público dentro de /srv.

/sys

Al igual que /proc, contiene archivos virtuales que proveen información del kernel relativa a eventos del sistema operativo. Es en cierto modo una evolución de /proc, y a diferencia de este último, los archivos se distribuyen de forma jerárquica.

```
(root@kali)-[/sys]
# ls
block bus class dev devices firmware fs hypervisor kernel module power
```

/tmp

Como ya da a entender su nombre, sirve para almacenar archivos temporales de todo tipo, ya sea de elementos del sistema, o también de diferentes aplicaciones a nivel de usuario.

Es un directorio dispuesto para almacenar contenido de corta duración, de hecho en la gran mayoría de los casos se suele vaciar de forma automática en cada reinicio del sistema. Aun así, no deben borrar su contenido de forma manual, puesto que puede contener archivos necesarios para ciertos programas o procesos que estén ejecutándose.

Las aplicaciones programadas para almacenar archivos en este directorio deben asumir que solo serán recuperables en la sesión actual. En este sentido, hay otro subdirectorio, /var/tmp, dispuesto igualmente para el almacenamiento de archivos temporales, pero cuyo contenido no se borra de forma automática tras el reinicio del sistema.

```
(root@kali)-[/tmp]
# ls
dbus-ZZK823jEyy          systemd-private-82d80abb6b614aefa4920daf9bd0f4c1-ModemManager.service-Vgt52i
ssh-5uPZzedKMsTp         systemd-private-82d80abb6b614aefa4920daf9bd0f4c1-systemd-logind.service-r98duh
systemd-private-82d80abb6b614aefa4920daf9bd0f4c1-colord.service-1P34xg  systemd-private-82d80abb6b614aefa4920daf9bd0f4c1-upower.service-6LnpIh
systemd-private-82d80abb6b614aefa4920daf9bd0f4c1-haveged.service-W12H9g
```

/usr

El directorio /usr viene de “User System Resources” y actualmente sirve para almacenar todos los archivos de solo lectura y relativos a las utilidades de usuario, incluyendo todo el software instalado a través de los gestores de paquetes de cada distribución.

Antiguamente /usr también contenía la carpeta particular de usuario, junto con todos sus documentos, vídeos, fotos, etc., pero más adelante se creó el directorio /home para este propósito, dejando /usr reservado para los archivos relativos a programas.

```
(root@kali)-[/usr]
# ls
bin  games  include  lib  lib32  lib64  libexec  libx32  local  sbin  share  src
```

/var

Contiene varios archivos con información del sistema, como archivos de logs, emails de los usuarios del sistema, bases de datos, información almacenada en la caché, información relativa a los paquetes de aplicaciones almacenados en /opt, etc. En cierto modo se podría decir que actúa a modo de registro del sistema.

```
(root@kali)-[/var]
# ls
backups  cache  lib  local  lock  log  mail  opt  run  spool  tmp  www
```

Rutas relativas vs rutas absolutas

Cuando se trabaja con comandos es habitual tener que pasar como parámetros archivos o directorios. Para indicar un archivo o directorio se utiliza una ruta o path, que puede ser absoluta o relativa.

Antes de nada vamos a recalcar una cosa. Aunque todo el mundo habla de particiones cuando quiere hacer referencia al lugar donde guarda los datos, realmente los datos se guardan en el sistema de archivos que está creado dentro de la partición. Las particiones son una división del espacio de un dispositivo de almacenamiento como un disco duro o un pendrive y por si solas no pueden almacenar nada, por eso es necesario crea un sistema de archivos dentro de ellas. Este proceso de creación es lo que se llama dar formato o formatear la partición. En Windows los sistemas de archivos utilizados son FAT o NTFS. En Linux existen muchísimos sistemas, pero los más comunes son los EXT en sus versiones 3 y 4 (ext3 / ext4).

Como mencionamos anteriormente A diferencia de Windows, Linux no tiene unidades, así que se pueden olvidar de A:\, C:\, D:\ y todas las letras del alfabeto. En Linux las diferentes particiones se montan dentro de la jerarquía del file system o sistema de archivos y se referencian dentro del mismo.

Vamos con el tema de las rutas o paths. Supongamos que dentro de la partición de datos tenemos un archivo llamado pelicula.mpg. Para indicar la ruta a este archivo se puede hacer de varias formas. Por ejemplo, la ruta absoluta a ese archivo es la siguiente:

`/home/pepe/media/datos/pelicula.mpg`

Las rutas absolutas se caracterizan por empezar SIEMPRE desde la raíz o root, es decir la /, y contener todos los directorios que hay desde la raíz hasta el archivo o directorio que

queremos indicar, sólo es posible escribir de una forma una ruta absoluta. Todas las rutas siguientes son rutas absolutas por que empiezan desde la raíz.

**/media/
/media/datos/
/etc/
/home/pepe/
/media/datos
/etc
/home/pepe/archivo.txt**

Cuando tienen la / al final significa que la ruta es de un directorio, aunque no es necesaria la /. Cuando la ruta es de un archivo NUNCA se pone la / al final.

En cambio las rutas relativas NUNCA empiezan por la / y puede haber varias rutas relativas para el mismo archivo o directorio que queremos indicar.

Esto es así porque la ruta relativa depende del directorio en el que se esté trabajando (donde estoy parado). Por ejemplo si estamos en el mismo directorio que el archivo pelicula.mpg, la ruta relativa es simplemente el nombre del archivo:

pelicula.mpg

En cambio, si estamos dentro del directorio media, pero fuera del directorio datos, la ruta relativa se escribiría:

datos/pelicula.mpg

Si estamos en la raíz del sistema de archivos: media/datos/pelicula.mpg

Y ¿Qué pasaría si estuviésemos dentro del directorio floppy?

Aquí es necesario indicar que para llegar al archivo antes tenemos que ir al directorio padre de floppy y después entrar en media y luego en datos. Para indicar el directorio padre se usan dos puntos y la /:

../media/datos/pelicula.mpg

Si tenemos que salir de más de un directorio se pueden poner más veces los dos puntos y la /. Por ejemplo:

../../otroarchivo.pdf

También existe el punto y la / para indicar el directorio actual de trabajo. Por eso, si estamos en un directorio donde existe un archivo llamado documento.pdf, podemos indicar la ruta relativa con sólo el nombre del archivo:

documento.pdf

O con el punto y la /:

./documento.pdf

Luego también pueden aparecer cosas curiosas como por ejemplo:

./documento.pdf

Que es lo mismo que las dos anteriores, pero evidentemente, nadie en su sano juicio hace esto.

Todo en Linux es un archivo

Como mencionamos anteriormente todo en un Linux es un archivo, tanto el Software como el Hardware, bueno el hardware es hardware, pero también es un archivo o por lo menos se puede acceder desde uno.

Desde el mouse, pasando por la impresora, el reproductor de DVD, el monitor, un directorio, un subdirectorio y un archivo de texto.

De ahí vienen los conceptos de montar y desmontar por ejemplo un CDROM.

El CDROM se monta como un subdirectorio en el sistema de archivos. En ese subdirectorio se ubicará el contenido del disco compacto cuando esté montado y, nada cuando esté desmontado.

Para ver que tenemos montado en nuestra distribución GNU/Linux, podemos ejecutar el comando “mount”.

Este concepto es muy importante para conocer cómo funciona Linux.

Como vimos anteriormente Linux el directorio `/dev` es donde podemos ver los dispositivos que tenemos en nuestro sistema operativo.

Algo interesante para hacer es ver información de nuestros dispositivos mediante los archivos que podemos encontrar en `/proc` , para ello utilizaremos el utilitario llamado `"cat"`

```
(root@kali)~# ls -l /proc
```

1	1083	11834	127	136	24	29999	30123	416	540	74	848	905	954		cmdline	fs	kpageflags	sched_debug	tty
10	10857	11837	128	14	242	3	30127	419	55	748	852	908	955		consoles	interrupts	loadavg	schedstat	uptime
1005	1093	11838	129	15	243	30	30140	471	555	753	853	913	970		cpuinfo	iomem	locks	self	version
1007	1098	11839	12996	16	25	30000	30145	474	556	754	856	928	976		crypto	ioports	meminfo	slabinfo	vmallocinfo
1010	11	11840	13	17	26	30001	30147	475	57	773	857	931	98		devices	irq	misc	softirqs	vmstat
1017	1106	119	130	1707	27	30035	30152	476	58	774	868	936	991		diskstats	kallsyms	modules	stat	zoneinfo
1018	1120	12	131	178	28	30036	30177	49	59	775	878	940	998		dma	kcore	mounts	swaps	
1025	1125	121	132	179	29	30082	30182	50	6	777	883	945	acpi	driver	keys	mtrr	sys		
1050	1128	122	13205	18	29775	30084	30183	51	60	786	887	948	asound	dynamic_debug	key-users	net	sysrq-trigger		
1070	114	12241	133	2	2979	30102	30193	52	690	836	892	951	buddyinfo	execdomains	kmsg	pagetypeinfo	sysvipc		
1074	116	123	134	20	29957	30103	31	53	70	838	9	952	bus	fb	kpagecgroupp	partitions	thread-self		
1078	11833	125	135	23	29998	30122	4	536	73	847	903	953	cgroups	filesystems	kpagecount	pressure	timer_list		

```
(root@kali)-[/proc]
└─# cat cpufreq
processor      : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 126
model name    : Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz
stepping      : 5
cpu MHz       : 1190.400
cache size    : 6144 KB
physical id   : 0
siblings      : 2
core id       : 0
cpu cores     : 2
apicid        : 0
initial apicid : 0
fpu           : yes
fpu_exception : yes
cpuid level   : 22
wp            : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx rdtscp lm constant_tsc rep_good nopl x
topology nonstop_tsc cpuid tsc_known_freq pni pclmulqdq ssse3 cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx rdrand hypervisor lahf_lm abm 3dnowprefetch invp
cid_single fsgsbase avx2 invpcid rdseed clflushopt md_clear flush_l1d arch_capabilities
bugs          : spectre_v1 spectre_v2 spec_store_bypass swapgs itlb_multihit
bogomips      : 2380.80
clflush size  : 64
cache_alignm  : 64
address sizes : 39 bits physical, 48 bits virtual
power management:
```

Es muy importante mencionar que Linux es un sistema operativo “Case Sensitive”.

Esto quiere decir que al contrario de Windows, en Linux las letras mayúsculas y minúsculas son interpretadas de forma diferente.

Esto implica que un directorio o un archivo llamado “Pepe” es diferente a uno llamado “pepe”, “pePe”, “pEpe”, etc...

Permisos de Archivos en Linux

Una de las cosas más importantes en GNU/Linux son los permisos de los archivos.

Comprender esto debería considerarse como algo esencial. Se suele creer que es algo complicado, pero veremos que no es así, y trataremos de verlo de manera sencilla y clara.

El sistema de permisos que utiliza GNU/Linux está directamente tomado de los permisos de archivos que utiliza UNIX. Un sistema potente que apenas ha sufrido variaciones durante todos estos años y que sigue siendo totalmente vigente. Por tanto esto hace pensar que es algo importante, y bien hecho.

Entender el sistema de archivos de GNU/Linux no es algo que sea para gurús, ni algo que

te vaya a llevar varios días de estudio. Es un sistema sencillo que trataremos de aprender a usarlo.

Lo primero que deben comprender es que todos los archivos en Linux pertenecen a un grupo de usuarios y a un usuario en particular.

También que a su vez los usuarios en UNIX/Linux pertenecen a grupos y que los permisos se asignan tanto a grupos como a usuarios.

Los usuarios pueden pertenecer a uno o más grupos y por ende tendrás diferentes permisos sobre los archivos/directorios.

Existen tres tipos de permisos que se pueden aplicar, estos son:

Lectura: otorga al grupo el permiso de poder leer el archivo. Se indica con la letra r (inicial de **Read**, que significa leer en inglés)

Escritura: otorga al grupo el permiso de poder editar el archivo pudiendo escribir en él. Se indica con la letra w (inicial de **Write**, que significa escribir en inglés)

Ejecución: otorga al grupo el permiso de poder ejecutar el archivo. Se indica con la letra x (que viene de la palabra **Execute**, que significa ejecutar en inglés)

Para entender mejor cómo esto es aplicado a un grupo, podrían, por ejemplo, darle a un grupo de usuarios el permiso de leer y escribir en un archivo, pero no la capacidad de poder ejecutarlo.

O podrían darle los permisos de poder leer y ejecutar un archivo, pero no de poder modificarlo. Incluso pueden darle a un grupo todos los permisos de lectura, escritura y ejecución de un archivo, o ningún permiso quitándoselos todos.

Visto el tema de los tres permisos, veamos ahora el tema de los grupos, del que has estado leyendo.

Los grupos del archivo a los que nos podremos referir son:

Usuario: el propietario actual del archivo (nos referimos a este grupo con la letra u)

Grupo: un grupo de usuarios de un archivo (nos referimos a este grupo con la letra g)

Todos: todos los usuarios (nos referimos a este grupo con la letra a de all, que significa todos en inglés)

En términos generales, sólo tendremos que trabajar con los tres primeros grupos.

El grupo de todos es sólo utilizado como un atajo (después veremos qué quiero decir con esto).

Si abren una terminal, y en la línea de comandos, escriben el comando “**ls -l**”, verán un listado de todos los archivos y directorios que hay dentro del directorio de trabajo actual en el que nos encontremos.

```
(rootkali)-[/home/kali]
# ls -l
total 36
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Desktop
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Documents
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Downloads
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Music
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Pictures
-rw-r--r-- 1 root root  5 Oct 10 22:21 pruebas.txt
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Public
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Templates
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Videos
```

Los permisos se agrupan en grupos de 3 letras, las primeras 3 posiciones corresponden a los permisos del owner (creador), las siguientes 3 al grupo y los últimos 3 al resto del mundo...

La primera columna es lo que nos brinda la información sobre los permisos, verán que hay muchos directorios (comienzan con la letra d y a continuación los permisos

d rwx r-x r-x

En el caso de **Descargas** por ejemplo, se interpreta de la siguiente manera:

Es un Directorio

Read, Write , Execute para el dueño.

Read , Execute para el grupo

Read , Execute para el resto del mundo.

También verán que hay un archivo llamado **pruebas.txt** (tiene un guión en vez de la d inicial) y los siguientes permisos:

- rw- r--r--

Se interpreta de la siguiente manera:

NO es un Directorio

Read, Write para el dueño.

Read , para el grupo

Read , para el resto del mundo.

Luego verán que hay 2 columnas, que dicen root y root.

Esto es el usuario y grupo, que en este caso se llama igual porque al crear un usuario siempre se crea un grupo con el mismo nombre.

Aquí un ejemplo donde no son lo mismo:

```

crw--w---- 1 root tty      4,  61 Oct  9 18:44 tty61
crw--w---- 1 root tty      4,  62 Oct  9 18:44 tty62
crw--w---- 1 root tty      4,  63 Oct  9 18:44 tty63
crw--w---- 1 root tty      4,   7 Oct 10 21:29 tty7
crw--w---- 1 root tty      4,   8 Oct  9 19:05 tty8
crw--w---- 1 root tty      4,   9 Oct  9 18:44 tty9
crw-rw---- 1 root dialout  4,  64 Oct  9 18:44 ttyS0
crw-rw---- 1 root dialout  4,  65 Oct  9 18:44 ttyS1
crw-rw---- 1 root dialout  4,  66 Oct  9 18:44 ttyS2
crw-rw---- 1 root dialout  4,  67 Oct  9 18:44 ttyS3
crw----- 1 root root    10, 239 Oct  9 18:43 uhid
crw----- 1 root root    10, 223 Oct  9 18:43 uinput
crw-rw-rw- 1 root root      1,   9 Oct  9 18:44 urandom
crw-rw---- 1 root root    10,  61 Oct  9 18:44 vboxguest
crw-rw-rw- 1 root root    10,  60 Oct  9 18:44 vboxuser

```

¿Te animas a explicar que es la letra “c” al principio?

Con lo explicado hasta ahora ya sabemos que en el ejemplo del archivo que se muestra en la captura el usuario tiene otorgados los permisos de lectura y escritura, el grupo al que pertenece el usuario tiene otorgado el permiso de lectura, escritura y otros tienen también permiso lectura y escritura.

Si alguno de los grupos tuviera el permiso de ejecutar ese archivo (en el caso de que se pudiera porque fuera un script) entonces como hemos visto estaría representado con una x. ¿Sencillo, verdad?

Manipulando los permisos

Para añadir, o quitar permisos a un archivo se utiliza el comando “chmod”. Para otorgar o quitar derechos podremos utilizar tanto las letras referidas a permisos y grupos que hemos visto como las equivalencias numéricas, lo que nos sea más fácil o lo que necesitemos en cada momento.

Y ahora veamos cómo poder manipular esos permisos. Lo primero aclarar que para poder manipular los permisos de un archivo debes ser el propietario de ese archivo o debes tener el permiso de poder editar el archivo, o tener acceso de superusuario con poderes totales (recuerden que un gran poder lleva una gran responsabilidad).

Sigamos con nuestro archivo del ejemplo, y supongamos que ahora le llamaremos script.sh y que es un script (secuencia de comandos con una lógica dentro de un archivo) escrito en bash y necesita poder ejecutarse... pero que sólo quieres darte a ti mismo permiso de ejecución.

```

(rootkali)-[/home/kali]
# ls -l
total 40
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Desktop
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Documents
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Downloads
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Music
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Pictures
-rw-r--r-- 1 root root  5 Oct 10 22:21 pruebas.txt
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Public
-rw-r--r-- 1 root root  7 Oct 10 22:26 script.sh
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Templates
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Videos

```

Con todo lo que han leído quizás piensen “necesito añadir una x de ejecución al primer grupo que es el usuario, o sea, yo mismo.”

Correcto, ahora veamos cómo añadir esa x a nosotros mismos mediante la línea de comandos. Para ello tan simple como:

chmod u+x script.sh

```

(rootkali)-[/home/kali]
# chmod u+x script.sh

(rootkali)-[/home/kali]
# ls -l
total 40
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Desktop
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Documents
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Downloads
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Music
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Pictures
-rw-r--r-- 1 root root  5 Oct 10 22:21 pruebas.txt
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Public
-rwxr--r-- 1 root root  7 Oct 10 22:26 script.sh
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Templates
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Videos

```

Así de simple. Indicándole con la u de user que queremos añadir el permiso de ejecución con la x al archivo. Pruébenlo con un archivo suyo, y verán que no es difícil de entender y de ejecutar.

Ahora hagámoslo un poco más interesante.

Supongamos que por alguna razón un archivo tiene el permiso de ejecución para todos los grupos algo parecido a: **-rwx-r-x-r-x**.

Si quieres quitar el permiso de ejecutarlo al grupo de otros tan sencillo como

chmod ugo-x script.sh


```
(root@kali)~/home/kali
# chmod ugo-x script.sh

(root@kali)~/home/kali
# ls -l
total 40
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Desktop
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Documents
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Downloads
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Music
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Pictures
-rw-r--r-- 1 root root  5 Oct 10 22:21 pruebas.txt
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Public
-rw-r--r-- 1 root root  7 Oct 10 22:26 script.sh
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Templates
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Videos
```

Efectivamente ugo=usuario/grupo/otros y con -x eliminas este permiso. Otra manera de hacerlo sería con el siguiente atajo (¿te acuerdas lo que comenté al inicio del artículo?):

chmod a-x script.sh

```
(root@kali)~/home/kali
# chmod a-x script.sh

(root@kali)~/home/kali
# ls -l
total 40
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Desktop
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Documents
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Downloads
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Music
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Pictures
-rw-r--r-- 1 root root  5 Oct 10 22:21 pruebas.txt
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Public
-rw-r--r-- 1 root root  7 Oct 10 22:26 script.sh
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Templates
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Videos
```

Tengan cuidado al quitar o dar permisos a un archivo si utilizan este método.

Permisos a directorios

También puedes otorgar o quitar permisos a directorios con el comando “chmod”.

Cuando creas un nuevo directorio como usuario, lo normal es crearlo con los siguientes permisos

drwxrwxr-x

Cabe señalar que los archivos que creamos dentro del directorio no tienen por qué heredar estos permisos, pudiendo tener otros distintos.

Como ya hemos visto, la d inicial indica que es un directorio. Como puedes ver en este ejemplo tanto el usuario como el grupo tienen plenos poderes sobre el directorio. Pero supongamos que dentro de un directorio queremos darles a todos los archivos que contiene unos permisos iguales para eso añadiremos al comando chmod el parámetro -R

que indica que efectuará de manera recursiva la asignación de permisos que estipulemos. Imaginemos que tenemos un directorio llamando Pruebas con un montón de scripts dentro. Todos esos archivos y el directorio en sí tiene los siguientes permisos -rwxrwxr-x. Si queremos quitar al grupo el permiso de escritura tendremos que utilizar el siguiente comando:

chmod -R g-w TEST

Es decir de manera recursiva (-R) al grupo (g) le quitaremos el permiso de escribir (-w) en los archivos que contiene el directorio (Pruebas).

Equivalente numérico

Hagámoslo ahora un poco más complejo (no se asusten, verán que no es para tanto). Cada permiso que hemos expresado con letras, también puede ser representado con números, lo que en algún caso nos puede resultar útil. Estas equivalencias son:

lectura = 4

escritura = 2

ejecución = 1

En el ejemplo del archivo que hemos visto anteriormente, la sustitución de las letras de los permisos por números sería la siguiente:

- 42- 4-- 4--

Ahora podemos sumar el número de los distintos grupos entre sí. Los permisos del usuario sería 4+2 lo que da 6 (para el owner). Los permisos de grupo y otros son simplemente 4, por tanto no hay nada que sumar.

El equivalente numérico quedaría tal que así:644

Enlaces físicos

Un enlace físico no es más que una etiqueta o un nuevo nombre asociado a un archivo.

Es una forma de identificar el mismo contenido con diferentes nombres.

Éste enlace no es una copia separada del archivo anterior sino un nombre diferente para exactamente el mismo contenido.

Para crear un enlace físico en Linux del archivo “pruebas.txt” a “prueba_In” ejecutamos:

```
(root@kali)-[/home/kali]
# ln pruebas.txt prueba_ln

(root@kali)-[/home/kali]
# ls
Desktop Documents Downloads Music Pictures prueba_ln pruebas.txt Public script.sh Templates Videos

(root@kali)-[/home/kali]
# ls -l
total 44
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Desktop
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Documents
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Downloads
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Music
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Pictures
-rw-r--r-- 2 root root  5 Oct 10 22:21 prueba_ln
-rw-r--r-- 2 root root  5 Oct 10 22:21 pruebas.txt
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Public
-rw-r--r-- 1 root root  7 Oct 10 22:26 script.sh
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Templates
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Videos
```

El enlace aparecerá como otro archivo más en el directorio y apuntará al mismo contenido de “prueba”. Cualquier cambio que se haga se reflejará de la misma manera tanto para “pruebas.txt” como para “prueba_ln”.

Un enlace se puede borrar usando el comando “rm” de la misma manera en que se borra un archivo. Esto puede tener varias ventajas, pero también puede complicar la tarea de seguimiento de los archivos. Un enlace físico tampoco puede usarse para hacer referencia a directorios o a archivos en otros equipos.

Enlaces simbólicos

Un enlace simbólico también puede definirse como una etiqueta o un nuevo nombre asociado a un archivo pero a diferencia de los enlaces físicos, el enlace simbólico no contiene los datos del archivo, simplemente apunta al registro del sistema de archivos donde se encuentran los datos.

Tiene mucha similitud a un acceso directo en Windows o un alias en OS X.

Para crear un enlace simbólico del archivo “prueba” a “prueba_ln”, ejecutamos:

\$ ln -s pruebas.txt prueba_ln

```
(root@kali)-[/home/kali]
# ln -s pruebas.txt prueba_ln

(root@kali)-[/home/kali]
# ls -l
total 40
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Desktop
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Documents
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Downloads
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Music
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Pictures
lrwxrwxrwx 1 root root  11 Oct 10 22:37 prueba_ln -> pruebas.txt
-rw-r--r-- 1 root root  5 Oct 10 22:21 pruebas.txt
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Public
-rw-r--r-- 1 root root  7 Oct 10 22:26 script.sh
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Templates
drwxr-xr-x 2 kali kali 4096 Sep  8 05:48 Videos
```

Este enlace también aparecerá como otro archivo más en el directorio y apuntará al mismo contenido de “prueba”, reflejando todos los cambios que se hagan tanto para “pruebas.txt” como para “prueba_ln”.

Sobre un enlace simbólico también se pueden usar todos los comandos básicos de archivos (rm, mv, cp, etc). Sin embargo cuando el archivo original es borrado o movido a una ubicación diferente el enlace dejará de funcionar y se dice que el enlace está roto.

Un enlace simbólico permite enlazar directorios y, también permite enlazar archivos fuera del equipo. En un principio puede parecer complicado, pero luego de leer detalladamente seguro que tendrás más claro cuándo usar un enlace simbólico y cuándo usar uno físico.



Ejercicios

Dada la siguiente estructura de directorios y archivos, escribe la ruta absoluta de todos los archivos visibles. Escribe después la ruta relativa a todos los directorios vacíos tomando como directorio de trabajo UTN.

```
.
|-- Datos
|   |-- DB
|   |   |-- pp.db
|   |-- www
|   |   |-- index.html
|-- Documentos
|   |-- Libro
|   |-- Revistas
|   |-- X
|       |-- area61.txt
|       |-- cia.txt
|       |-- secreto.txt
-- UTN

8 directories, 5 files
```

Usando la estructura de directorios anterior crear un link simbólico del archivo area61.txt tomando como directorio de trabajo UTN.

Usando la estructura de directorios anterior crear un link físico del archivo cia.txt tomando como directorio de trabajo Test.

Con motivo de un control de las unidades expuestas, les solicitamos que en caso de alguna incoherencia, error o inconvenientes de lectura, nos haga saber a través del foro de la unidad.



Link complementarios:

En caso de que los links que se exponen no funcionen, por favor avisar al instructor (es normal que un sitio pueda cambiar su URL, dominio o variables, lo cual como la unidad se prepara a principio de año podría suceder que se haya modificado).

<https://www.gnome.org/>

<https://extensions.gnome.org/>

<https://www.gnome-look.org/>

<http://glx-dock.org/>

Consultas y/o dudas a través de la plataforma o por mail al Instructor.
Las mismas serán respondidas en un plazo no mayor de 72 horas.