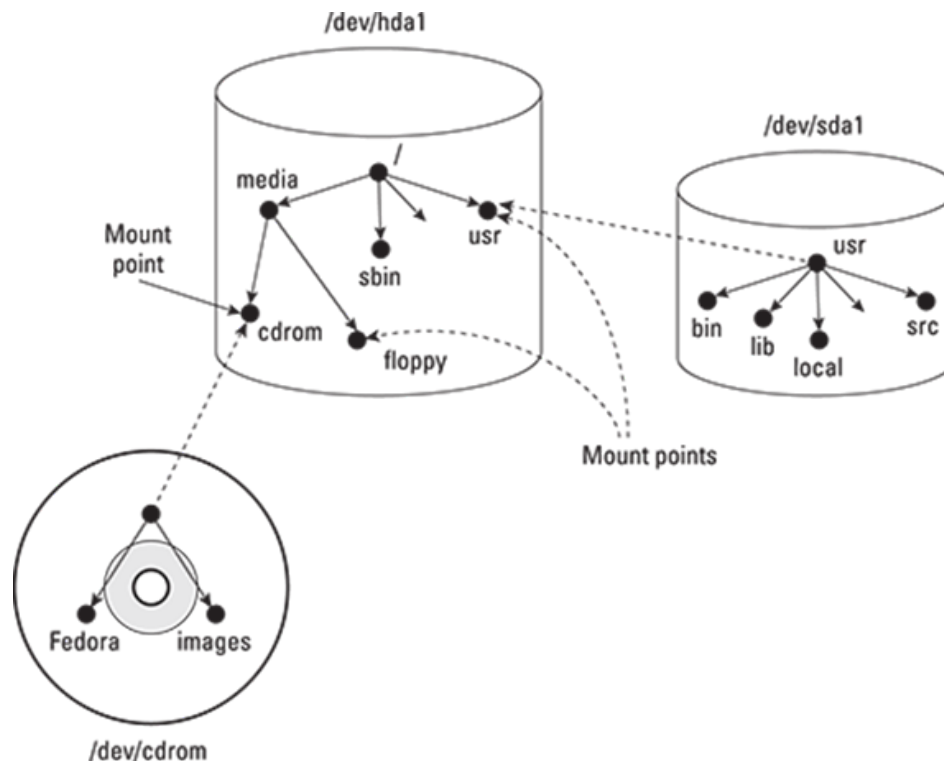


2.3.L Estructura de directorios Linux

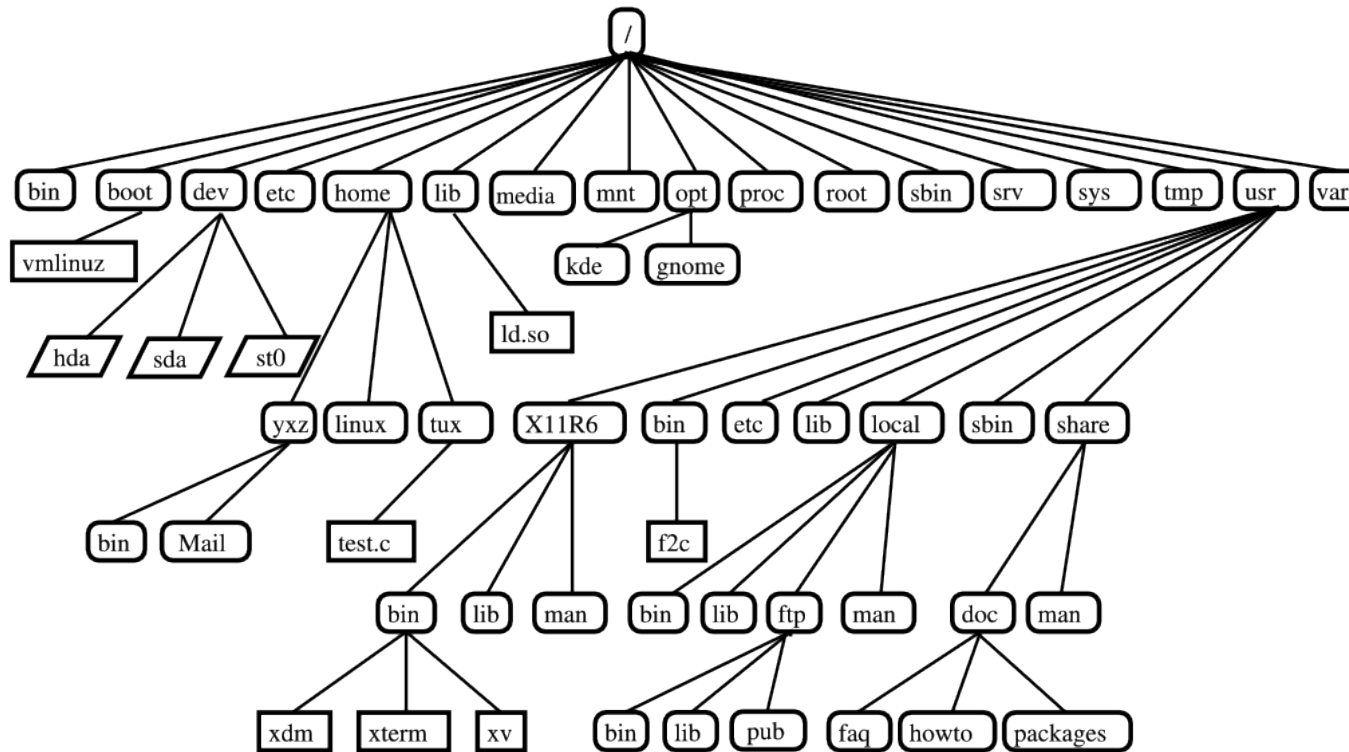
- El árbol de directorios de Linux se basa en el Filesystems Hierarchy Standard
- El estándar facilita la escritura de software para Linux
- El árbol de directorios está diseñado para poder romperse en partes más pequeñas, que se puedan ajustar a diferentes tamaños de disco y de forma que se facilite el backup y las restantes tareas de administración.
- Las partes principales son root (/), /usr, /var y /home. Cada parte tiene un propósito diferente. Algunas partes pueden ponerse en red y ser compartidas por varias máquinas. Otras podrían estar en un dispositivo de sólo lectura (como un DVD).

2.3.L Arbol de directorios Linux



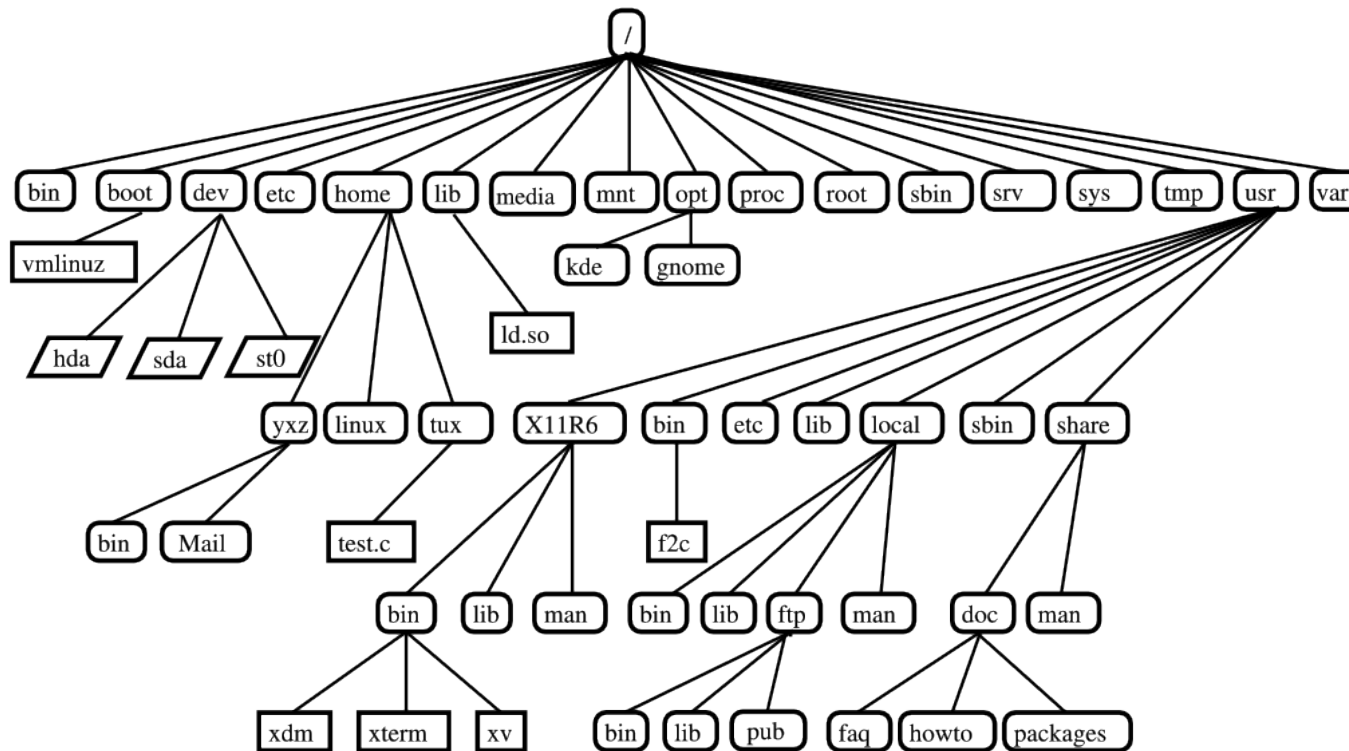
- La partición root (`/`) contiene los ficheros necesarios para botar el sistema. Su contenido es suficiente para botar en modo single user
- El filesystem `/usr` contiene los comandos, librerías, páginas de manual y todos los ficheros que no cambian durante un uso normal de la máquina. En `/usr` no hay ficheros específicos de la máquina, lo cual permite que estos archivos puedan compartirse en red.

2.3.L Arbol de directorios Linux



- El filesystem `/var` contiene todos los ficheros que cambian, tales como los spool de impresión, de mail, etc., las páginas de manual formateadas, los archivos temporales y los ficheros con los logs del sistema. La mayor parte de los ficheros en `/var` estaban bajo `/usr` en las versiones iniciales de Linux.
- Los directorios `/home` contienen los datos de los directorios de los usuarios. Es común que el directorio `/home` abarque varios discos.

2.3.L Arbol de directorios Linux



- No es necesario que /usr, /var, /home estén en filesystems diferentes, siempre y cuando los nombres estándar funcionen. Por ejemplo, /var y /usr pueden compartir filesystem si hay un link simbólico de /var a /usr/var, de modo que el archivo /var/log/messages (por ejemplo) sea accesible con su nombre habitual.

2.3.L El filesystem raíz

- El filesystem raíz debe ser pequeño. Es más difícil que se corrompa un sistema de archivos pequeño y que se modifica con poca frecuencia que lo contrario.
- El directorio raíz no suele contener archivos. Antiguamente contenía la imagen del kernel. Las distribuciones actuales almacenan el kernel en /boot
- Los subdirectorios del filesystem raíz son:
 - /bin: comandos usados al botar que podrían necesitarse por algún usuario
 - /sbin: similar a /bin pero en el path de root
 - /etc: ficheros de configuración
 - /root: Home del usuario /root
 - /lib: Librerías compartidas
 - /lib/modules: módulos del kernel
 - /dev: Ficheros de dispositivo
 - /tmp: Ficheros temporales
 - /boot: Ficheros usados por GRUB
 - /mnt: Puntos de montaje temporales
 - /proc, /usr, /var, /home: Puntos de montaje para otros sistemas de archivos

2.3.L El directorio /etc

- El directorio /etc contiene todos los ficheros de configuración de la máquina
- /etc/rc.d: Scripts que se ejecutan al arrancar (o cambiar de runlevel en máquinas SysV)
- /etc/passwd: Base de datos de usuarios, con campos que contienen el nombre, directorio raíz, etc. de cada usuario
- /etc/shadow: Fichero encriptado con las claves
- /etc/fstab: Lista de ficheros que se montan automáticamente al arrancar con el comando mount -a, e información acerca de las particiones de swap
- /etc/group: Base de datos de grupos de usuarios

2.3.L El directorio /etc

- /etc/inittab: configuración del proceso init (en máquinas SysV)
- /etc/issue: Mensaje de bienvenida que abre getty (antes de login)
- /etc/magic: Configuración de la herramienta “file”, mediante la que se detecta el tipo de contenido de un fichero
- /etc/motd: Mensaje del día (se muestra después de login)
- /etc/mtab: Lista de los filesystems montados en el sistema
- /etc/login.defs: Configuración del comando login

2.3.L El directorio /etc

- /etc/profile, /etc/bashrc, /etc/csh.cshrc: Ficheros que se ejecutan al arrancar por los shells Bourne, BASH o C. Permiten al administrador que inicie servicios globales para todos los usuarios al iniciar sesión.
- /etc/securetty: Terminales desde los que root puede entrar
- /etc/shells: Shells en los que se confía (se usan por el comando chsh y por el daemon de ftp)
- /etc/termcap, /etc/printcap: Características de terminales y de impresoras

2.3.L El directorio /dev

- **/dev/hda, /dev/hdb...** : Antiguos discos IDE
- **/dev/sda, /dev/sdb..** : Discos SCSI, SAS, SATA o USB
- **/dev/loop0**: Dispositivos loopback. Se usan para montar filesystems que no están localizados en dispositivos de bloque. Por ejemplo, una imagen de un DVD. Su uso es transparente para el usuario. Se manejan mediante las órdenes mount y **losetup**.
- **/dev/md0**: Metadisco (Array RAID)
- **/dev/null**: Dispositivo al que se puede enviar data y que desaparezca
- **/dev/random, /dev/urandom**: Generador de numeros aleatorios del kernel

2.3.L El directorio /usr

- **/usr/X11R6**: X Windows
- **/usr/bin**: La mayoría de los comandos de usuario (los demás están en /bin y en /usr/local/bin)
- **/usr/sbin**: Comandos de administración del sistema que no son necesarios en el filesystem raíz
- **/usr/share/man, /usr/share/info, /usr/share/doc**: Manuales
- **/usr/include**: Cabeceras del C
- **/usr/lib**: Librerías del sistema
- **/usr/lib/systemd**: unidades de systemd
- **/usr/local**: Instalación de nuevo software, local para esta máquina, que se desea mantener separado de los ficheros de sistema gestionados por rpm/yum

2.3.L Directorio /var

- **/var/cache**: Cache de archivos, como /var/cache/man (páginas formateadas de manual)
- **/var/lib**: Archivos del sistema cuyo contenido cambia (y que por tanto no pueden estar en /usr)
- **/var/lock**: Ficheros de bloqueo de dispositivos (por ejemplo, para que dos usuarios no accedan a la vez al puerto serie) y de archivos
- **/var/log**: Logs del sistema, especialmente de login (/var/log/wtmp) y de syslog (/var/log/messages)
- **/var/spool/mail, /var/mail**: mailboxes de correo
- **/var/run**: Información del sistema válida hasta que se bota la siguiente vez, como /var/log/utmp, que muestra los usuarios actuales de la máquina

2.3.L Filesystem /proc

- El filesystem **/proc** no existe físicamente en un disco. El kernel lo crea en memoria. Se usa para proporcionar información acerca del sistema. Originariamente, acerca de los procesos, y de ahí su nombre.
- **/proc/1**: Directorio con información acerca del proceso 1. Cada proceso tiene un directorio con su número
- **/proc/cpuinfo**: Tipo de procesador, marca, modelo, velocidad, etc.
- **/proc/devices**: Dispositivos configurados en el kernel que está ejecutándose
- **/proc/dma**: Canales de DMA usados

2.3.L Filesystem /proc

- **/proc/filesystems**: Tipos de filesystems soportados
- **/proc/interrupts**: Interrupciones en uso
- **/proc/iports**: Puertos de entrada/salida
- **/proc/kcore**: Imagen de la memoria física del sistema (que no ocupa espacio, se genera cuando los programas acceden)+
- **/proc/kmsg**: Mensajes del kernel (que son enrutados a syslog)
- **/proc/ksyms**: Tabla de símbolos

2.3.L Filesystem /proc

- **/proc/loadavg**: Carga del sistema
- **/proc/meminfo**: Uso de memoria física y swap
- **/proc/modules**: Módulos del kernel cargados
- **/proc/net**: Directorio que contiene información acerca del estado de diferentes protocolos de red
- **/proc/self**: Enlace simbólico al directorio del proceso que está consultando a /proc
- **/proc/stat, /proc/uptime, /proc/version**: Estadísticas, tiempo arriba y version del kernel

2.3.L Orden mount

- En Linux hay una única jerarquía de directorios, de la que cuelgan todos los dispositivos de almacenamiento
- Cada partición tiene su propio sistema de ficheros, con su propio directorio raíz
- Montar un sistema de ficheros consiste en añadir un sistema de ficheros en el sistema de ficheros lógico. Sus datos estarán disponibles a partir de un directorio.
- Desmontar un sistema de ficheros hace que deje de estar disponible
- El sistema operativo se instala en el sistema de ficheros raíz, que siempre está montado en el directorio “/”

2.3.L Orden mount

- Durante el proceso de arranque, como se ha visto, primero se monta el sistema de ficheros raíz y después el resto de los sistemas de ficheros
- La orden que se usa para montar un sistema de ficheros es **mount**
- **Formato:** mount [opciones] FicheroConDispositivoBloque PuntoMontaje
 - -t tipo de sistema de ficheros
 - -r en modo lectura
 - -w en modo escritura
 - -o opciones: nosuid, [no]exec, remount, etc.

2.3.L Otras órdenes relacionadas

- **umount** PuntoDeMontaje (o umount FicheroDispositivoBloque) desmonta el sistema de ficheros. Sólo se puede desmontar si no está siendo utilizado
- **fuser**: Indica qué ficheros se están usando y qué procesos los usan (f: fichero abierto, c: directorio de trabajo, e: ejecutando un fichero, etc.)
- **lsuf**: Lista de todos los ficheros abiertos

```
$ mount -t iso9660 -r /dev/cdrom /media/cdrom
```

```
$ cd /media/cdrom
```

```
$ /sbin/fuser -mv /media/cdrom/
```

	USER	PID	ACCESS	COMMAND
/media/cdrom/	pilar	1118	..c..	bash

2.3.L El fichero /etc/fstab

- **/etc/fstab** contiene información sobre todos los ficheros disponibles, y de las particiones swap a activar
- El formato es: fi_especial pto_montaje tipo opciones dump_freq pass_num

```
#
# /etc/fstab
# Created by anaconda on Mon Sep 30 08:11:29 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=eb20781f-a6d5-46c6-b1bf-39b2187c44fd /                xfs      defaults        0 0
UUID=22fe73a3-1232-4433-8f65-b7dee3e1e55c /boot            ext4      defaults        1 2
UUID=0C6E-551F      /boot/efi       vfat      umask=0077,shortname=winnt 0 2
UUID=0d362014-421f-4306-9b25-12d9838bf7e4 /home           xfs      defaults        0 0
UUID=8cdf019d-5dbe-4a40-b1e3-8732150095b2 swap            swap      defaults        0 0
[root@localhost ~]#
```

2.3.L /etc/fstab y /etc/mtab

- En el arranque, **systemd** ejecuta **systemd-fstab-generator** para convertir /etc/fstab en una lista de unidades systemd del tipo mount

```
[root@localhost ~]# systemctl -t mount
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
-.mount                            loaded active mounted Root Mount
boot-efi.mount                     loaded active mounted /boot/efi
boot.mount                         loaded active mounted /boot
dev-hugepages.mount               loaded active mounted Huge Pages File System
dev-mqueue.mount                  loaded active mounted POSIX Message Queue File System
home.mount                        loaded active mounted /home
run-user-0.mount                  loaded active mounted /run/user/0
sys-kernel-config.mount           loaded active mounted Kernel Configuration File System
sys-kernel-debug.mount            loaded active mounted Kernel Debug File System

LOAD    = Reflects whether the unit definition was properly loaded.
ACTIVE  = The high-level unit activation state, i.e. generalization of SUB.
SUB      = The low-level unit activation state, values depend on unit type.

9 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
[root@localhost ~]#
```

2.3.L systemd-fstab-generator

- **systemd-fstab-generator** genera un archivo para cada punto de montaje con los parámetros de montaje **what/where/type/options**, y las dependencias en **before/after**

```
[root@localhost ~]# cat /usr/lib/systemd/system/tmp.mount
# SPDX-License-Identifier: LGPL-2.1+
#
# This file is part of systemd.
#
# systemd is free software; you can redistribute it and/or modify it
# under the terms of the GNU Lesser General Public License as published by
# the Free Software Foundation; either version 2.1 of the License, or
# (at your option) any later version.
#
[Unit]
Description=Temporary Directory (/tmp)
Documentation=man:hier(7)
Documentation=https://www.freedesktop.org/wiki/Software/systemd/APIFileSystems
ConditionPathIsSymbolicLink=!/tmp
DefaultDependencies=no
Conflicts=umount.target
Before=local-fs.target umount.target
After=swap.target
#
[Mount]
What=tmpfs
Where=/tmp
Type=tmpfs
Options=mode=1777,strictatime,nosuid,nodev
[root@localhost ~]#
```

2.3.L /etc/mtab

- **/etc/mtab**: información acerca de los sistemas ya montados (igual que mount sin argumentos)

```
sysfs /sys sysfs rw,seclabel,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
devtmpfs /dev devtmpfs rw,seclabel,nosuid,size=663412k,nr_inodes=165853,mode=755 0 0
securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev/shm tmpfs rw,seclabel,nosuid,nodev 0 0
devpts /dev/pts devpts rw,seclabel,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0
tmpfs /run tmpfs rw,seclabel,nosuid,nodev,mode=755 0 0
tmpfs /sys/fs/cgroup tmpfs ro,seclabel,nosuid,nodev,noexec,mode=755 0 0
cgroup /sys/fs/cgroup/systemd cgroup rw,seclabel,nosuid,nodev,noexec,relatime,xattr,release_agent=/usr/lib/systemd/systemd-cgroup
ps-agent,name=systemd 0 0
pstore /sys/fs/pstore pstore rw,seclabel,nosuid,nodev,noexec,relatime 0 0
efivarfs /sys/firmware/efi/efivars efivarfs rw,nosuid,nodev,noexec,relatime 0 0
bpf /sys/fs/bpf bpf rw,nosuid,nodev,noexec,relatime,mode=700 0 0
cgroup /sys/fs/cgroup/blkio cgroup rw,seclabel,nosuid,nodev,noexec,relatime,blkio 0 0
cgroup /sys/fs/cgroup/cpuset cgroup rw,seclabel,nosuid,nodev,noexec,relatime,cpuset 0 0
cgroup /sys/fs/cgroup/net_cls,net_prio cgroup rw,seclabel,nosuid,nodev,noexec,relatime,net_cls,net_prio 0 0
cgroup /sys/fs/cgroup/pids cgroup rw,seclabel,nosuid,nodev,noexec,relatime,pids 0 0
cgroup /sys/fs/cgroup/freezer cgroup rw,seclabel,nosuid,nodev,noexec,relatime,freezer 0 0
cgroup /sys/fs/cgroup/hugetlb cgroup rw,seclabel,nosuid,nodev,noexec,relatime,hugetlb 0 0
cgroup /sys/fs/cgroup/rdma cgroup rw,seclabel,nosuid,nodev,noexec,relatime,rdma 0 0
cgroup /sys/fs/cgroup/perf_event cgroup rw,seclabel,nosuid,nodev,noexec,relatime,perf_event 0 0
cgroup /sys/fs/cgroup/cpu,cpuacct cgroup rw,seclabel,nosuid,nodev,noexec,relatime,cpu,cpuacct 0 0
cgroup /sys/fs/cgroup/devices cgroup rw,seclabel,nosuid,nodev,noexec,relatime,devices 0 0
cgroup /sys/fs/cgroup/memory cgroup rw,seclabel,nosuid,nodev,noexec,relatime,memory 0 0
configfs /sys/kernel/config configfs rw,relatime 0 0
/dev/md127 / xfs rw,seclabel,relatime,attr2,inode64,sunit=1024,swidth=2048,noquota 0 0
selinuxfs /sys/fs/selinux selinuxfs rw,relatime 0 0
systemd-1 /proc/sys/fs/binfmt_misc autofs rw,relatime,fd=27,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=18489 0 0
mqueue /dev/mqueue mqueue rw,seclabel,relatime 0 0
debugfs /sys/kernel/debug debugfs rw,seclabel,relatime 0 0
hugetlbfs /dev/hugepages hugetlbfs rw,seclabel,relatime,pagesize=2M 0 0
/dev/sda5 /home xfs rw,seclabel,relatime,attr2,inode64,noquota 0 0
/dev/md126 /boot ext4 rw,seclabel,relatime 0 0
```

2.3.L fsck

- La orden **fsck** chequea la consistencia del sistema de ficheros:
 - Bloques que pertenecen a varios ficheros
 - Bloques marcados como libres pero que están en uso
 - Bloques marcados como usados pero que están libres
 - Inconsistencia del número de enlaces
 - Nodos-i marcados como ocupados que están libres
- Para chequear un sistema de archivos siempre debe estar desmontado. El Sistema de Archivos raíz no se puede desmontar, por lo que debe estar montado en modo lectura para hacer fsck.