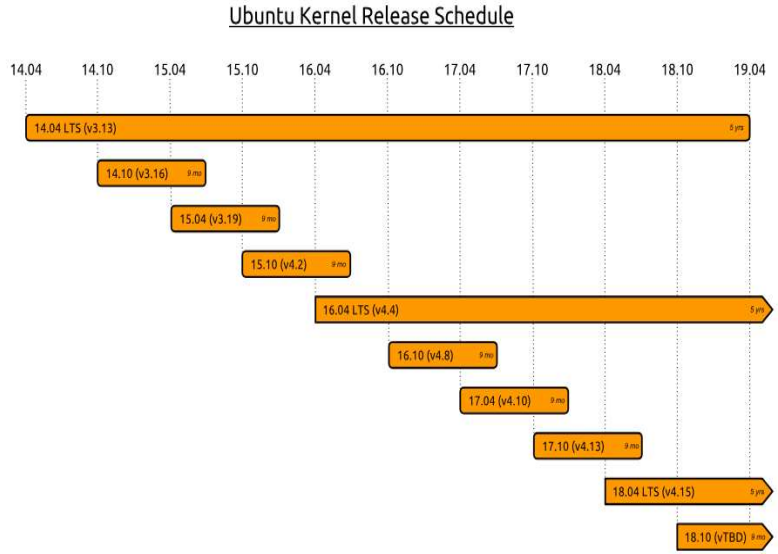
INTRODUCCIÓ A UBUNTU LINUX

Sistemes Informátics 1° DAW

Jose Socuéllamos

VERSIONS D'UBUNTU

- Ubuntu llança nova versió cada sis mesos, numerada d'acord a l'any i mes al qual es va llançar.
  - Per exemple, Ubuntu 20.04 és d'abril de 2020 i Ubuntu 20.10 d'octubre de 2020.
- Pero no totes las versions disposen del mateix suport:
  - Les versions corrents o intermèdies ixen a l'abril i octubre els anys imparells, i a l'octubre els anys parells. Oferixen nou mesos de suport.
    - Exemples: 19.04, 19.10, 20.10
  - Les versions LTS (per *Long Term Support* o suport a llarg termini) ixen a l'abril els anys parells. Oferixen cinc anys de suport i renoven les ISOs d'instal·lació 5 voltes als primers dos anys (se diuen versions de manteniment):
    - Exemples: 18.04.4, 18.04.5, 20.04.1



ACTUALITZACIONS D'UBUNTU (UPGRADES)

- Ubuntu permet actualitzar fàcilment d'una versió a una altra, però amb diferències entre les versions intermèdies i les LTS:
  - Les versions intermèdies oferixen actualitzar a la següent versió immediata al poc de que haja eixit. Per exemple, des de Ubuntu 18.10 podem actualitzar ràpidament a Ubuntu 19.04, Ubuntu 19.10...
  - Les versions LTS oferixen actualitzar <u>només</u> a la següent versió LTS, i només a partir de la segona versió de manteniment. Per exemple, des de Ubuntu 18.04 LTS es pot actualitzar a Ubuntu 20.04.1 LTS.

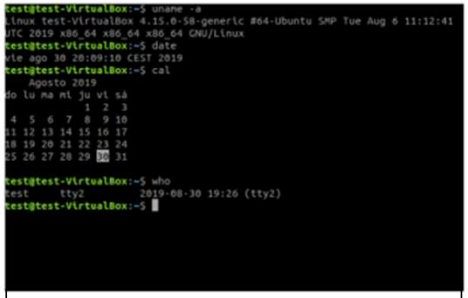
INTERFICIE D'USUARI

- La interfície d'usuari és el mètode d'interacció entre l'usuari i la màquina.
- Es tracta d'un concepte general que aplica a tots els sistemes informàtiques que requerisquen d'interacció humana.
- A Ubuntu podem distingir entre dos tipus d'interfícies d'usuari:

CLI

Command Line Interface

Interacció per mitjà de text. Es basa en l'ús d'un llenguatge codificat



Intèrpret de comandos

GUI

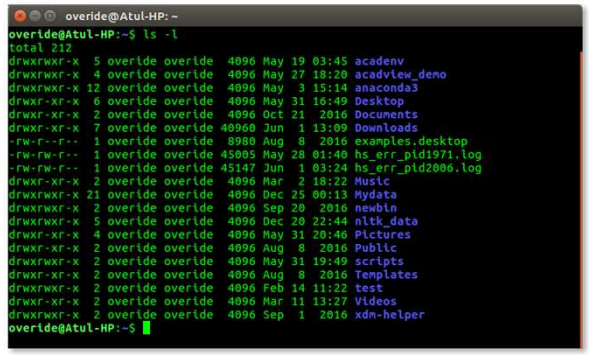
Graphical User Interface

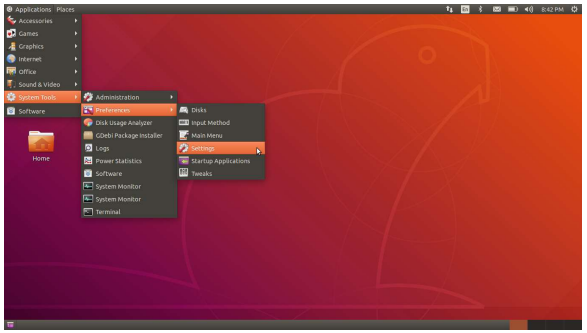
Entorn d'escriptori

Interacció a través d'elements gràfics. Possibilita una interacció amigable i intuïtiva

INTERFICIE D'USUARI – SHELL

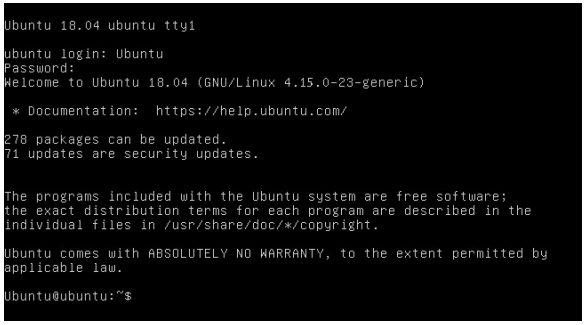
- La ferramenta que possibilita qualsevol d'estes formes d'interacció se la sol conéixer com ***shell***.
- Una shell és una ferramenta que accepta ordenes o instruccions per part de l'usuari i executa operacions.
- A Linux podem interactuar amb multitud de shells diferents, tant a nivell CLI con a nivel GUI:
  - CLI: sh, **bash**, ksh, csh…
  - GUI: GNOME, KDE, Cinnamon, Pantheon…
- Inclús podem tindre més d'un instal·lat simultàniament.





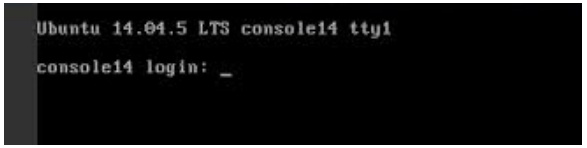
INTERFICIE D'USUARI – CLI vs GUI

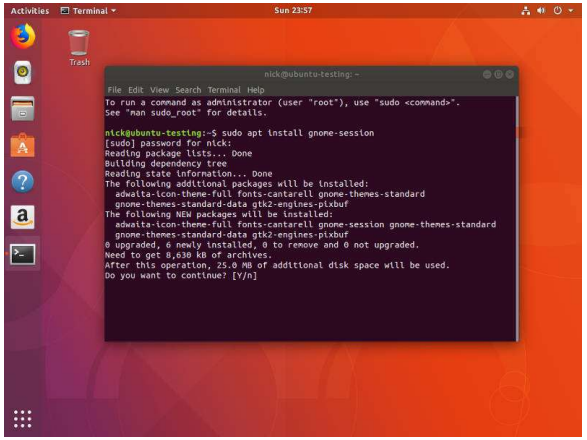
- CLI
  - mètode de comunicació entre usuari i maquina
  - accepta instruccions de l'usuari a través de línies de text
  - ixes instruccions seguixen unes determinades regles de sintaxi.
  - s'utilitzava abans de l'aparició de les interfícies gràfiques o GUI.
- GUI
  - van servir per a acostar la informàtica al gran públic
  - permeten interacció usuari/maquina molt més amigable o intuitiva
  - Son més complexes i consumixen més recursos.
- Encara que hui en dia quasi tots els SO GNU/Linux presenten una GUI plenament integrada amb la resta del sistema, podem accedir a la CLI per mitjà de la **terminal**.



INTERFICIE D'USUARI – TERMINAL

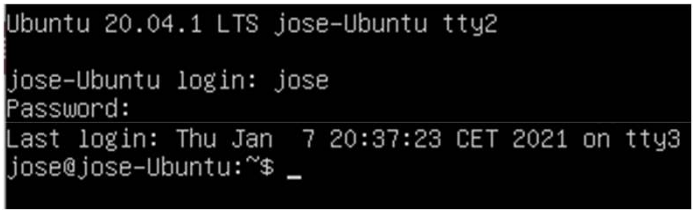
- Terminals TTY
  - Les TTY són les diferents finestres de terminal que permeten interactuar amb el sistema a través de l'interprete de comandos, fora de l'entorn gràfic.
  - A Ubuntu hi ha un total de 7 pantalles TTY: des de la TTY1 fins a la TTY6 són sessions de text, mentres que la TTY7 és la GUI.
  - Per canviar d'una a altra polsem CTRL+ALT+Fx (x=nº de TTY)
- Emulador de terminal
  - La pròpia GUI de Linux també permet utilitzar la terminal a través dels emuladors de terminal, o PTES.
  - És una aplicació que permet virtualitzar una CLI però dins de la pròpia interfície gràfica.
  - Tenen l'avantatge de possibilitar la interacció a través de la línia de comandos, però sense necessitat d'eixir de l'entorn gràfic.





INICI DE SESIÓ I AUTENTICACIÓ

- El sistema Linux és multiusuari, per la qual cosa cal començar per autenticar-se en l'equip.
- L'autenticació servix per a comprovar que l'usuari situat davant de la consola és qui pretén ser.
- Ha de facilitar la seua identitat (nom d'inici de sessió o login) i una prova d'esta (contrasenya) per a connectar-se.
- Per motius de seguretat, si iniciem sessió per CLI la contrasenya escrita per l'usuari no es mostra en pantalla.
- Quan introduïm el nom d'inici de sessió i la contrasenya s'han de respectar escrupolosament majúscules i minúscules. Cal recordar que Linux es sensible a majúscules.



COMANDOS DE LINUX

- Un comando és una instrucció que li indica al S.O. una tasca a realitzar. Poden ser:
  - Comandos de sistema que permeten crear, modificar o moure arxius i carpetes.
  - Comandos d'execució de programes o processos en el S.O.
- Per regla general, es llançen escrivint el seu nom a la terminal:

```
jose@jose-Ubuntu:~$ comando
```

- La majoria poden rebre arguments, es a dir modificadors de la seua funció:

```
jose@jose-Ubuntu:~$ comando arg1 arg2
```

- Linux incluix el comando **man**, una ferramenta que s'utilitza per a accedir a la documentació de tots els comandos i així aprendre sobre ells i com utilitzar-los.

```
jose@jose-Ubuntu:/etc/gnome$ man comando
```

COMANDOS DE LINUX

- Alguns dels comandos més bàsics son:
  - `who am i`: indica l'usuari que té la sessió activa de l'ordinador.

```
jose@jose-Ubuntu:~$ who am i
jose     tty2         2021-01-09 21:06
```

  - `pwd`: indica el directori a on ens trobem actualment.

```
jose@jose-Ubuntu:/etc/gnome$ pwd
/etc/gnome
```

  - `ls`: mostra una llista dels arxius i carpetes que es troben en el directori actual.

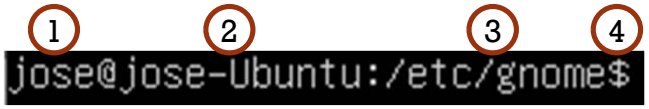
```
jose@jose-Ubuntu:/etc/gnome$ ls
defaults.list  menus.blacklist
```

  - `cd`: permet canviar el directori actual a un nou.

```
jose@jose-Ubuntu:/etc$ cd gnome
jose@jose-Ubuntu:/etc/gnome$
```

SÍMBOL DE SISTEMA DEL SHELL (PROMPT)

- Una vegada connectat a un terminal de text, s'inicia automàticament un programa anomenat Shell que permet escriure els comandos que indicarem més avant.
- El shell indica que està en espera d'una instrucció presentant un símbol del sistema (o prompt) al principi de la línia.
- El prompt pot tindre diversos aspectes. Per defecte, a Ubuntu es el següent:



1. jose: és el nom de l'usuari que ha iniciat sessió.
2. jose-ubuntu: és el nom de l'ordinador (li'l donarem durant la instal·lació)
3. /etc/gnome: directori a on ens trobem actualment

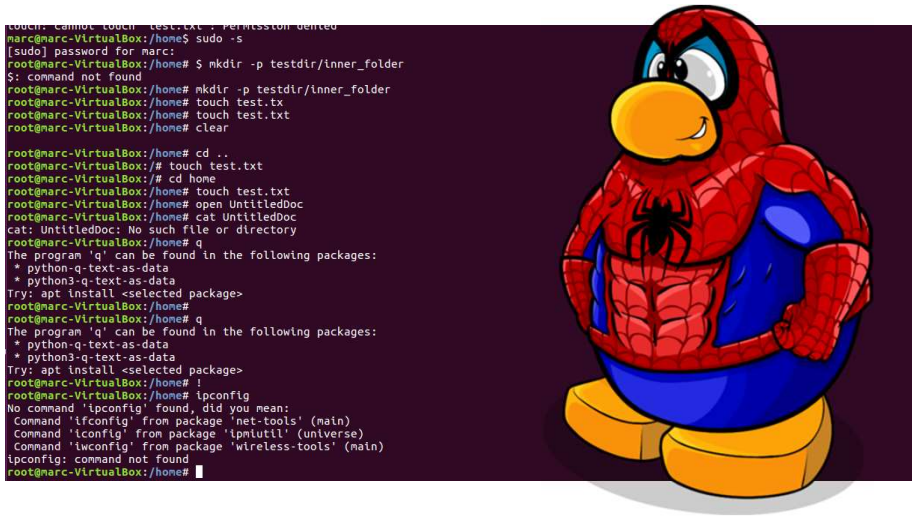
- L'element més important és l'últim carácter (4) que indica el tipus d'usuari conectat:
  - Si és un ($) indica que és un usuari qualsevol sense drets especials
  - Si és un (#) indica que és un usuari administrador que pot configurar i mantindre el sistema

L'USUARI ROOT

- Linux es llibertat, podem fer qualsevol cosa al SO.
- Però no tots els usuaris poden fer qualsevol cosa…
- Molts comandos invoquen accions que poden ser potencialment perilloses per al sistema, com instal·lar o desinstal·lar programes, cambiar la IP de l'equip…
- L'usuari que es va crear durant la instal·lació és un usuari normal que només pot accedir al seu espai de treball i utilitzar els recursos del sistema, no configurar-los.
- Per contra, l'usuari **root** o **superusuari** d'un sistema Unix:
  - té accés administratiu al SO.
  - té accés al directori arrel i en definitiva a tot el sistema.
  - pot executar qualsevol comando sobre qualsevol arxiu o carpeta…

L'USUARI ROOT

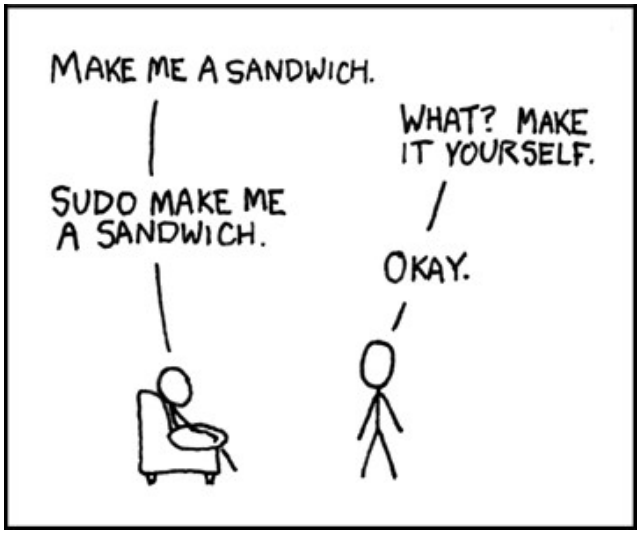
- L'usuari ***root*** pot fer i desfer com vullga, però s'ha de coneixer molt be el sistema per a no provocar danys en ell.
- Per exemple, el comando rm –rf * elimina tot el contingut d'una carpeta sense fer més preguntes. Executar-lo des del directori /…
  - Si som un usuari normal, no passarà res
  - Si som root, borrarà el disc sencer!



UN GRAN PODER COMPORTA UNA GRAN RESPONSABILITAT

COM SER ROOT

- sudo
  - Aquest comando ens dona "poders" temporals de superusuari.
  - El que fa és executar el comando que ho seguix utilitzant, temporalment, els privilegis de root.
  - Sintaxi: `sudo <comando>`
    - `sudo nano /etc/passwd` → **edita l'arxiu d'usuaris**



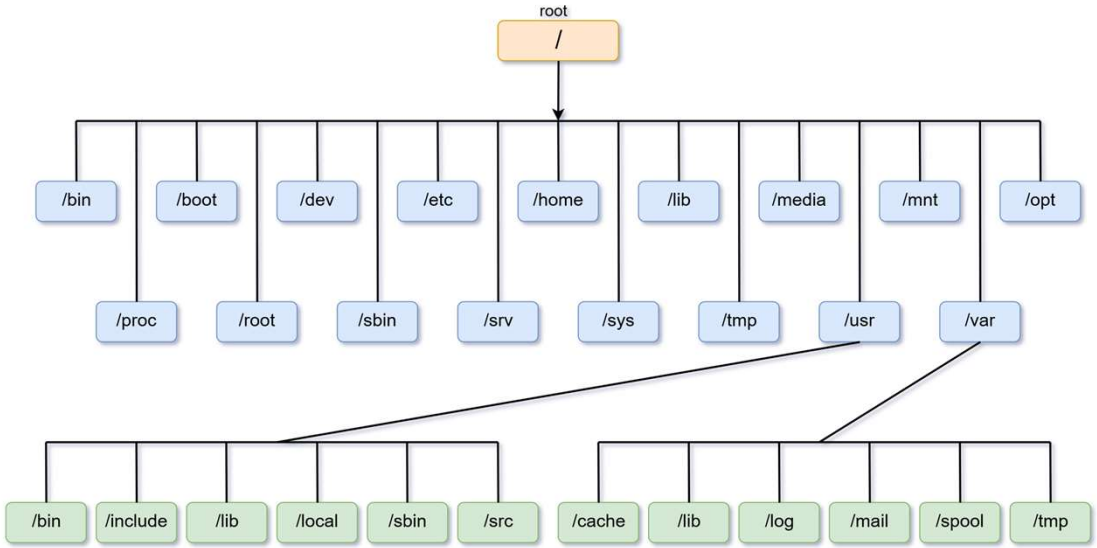
COM SER ROOT

COM SER ROOT

- su
  - Com el seu nom indica (switch user) permet un canvi de sessió d'usuari, sense necessitat de tancar la sessió de l'usuari actual.
  - S'utilitza seguit del nom d'usuari amb el qual es vol iniciar sessió.
  - És molt comú utilitzar-ho amb el compte de superusuari perquè qualsevol usuari que conega la contrasenya puga fer un salt de sessió i convertir-se en root.
  - Sintaxi: `su <usuari>`
    - `sudo su pablo` → cambia l'usuari passant a ser pablo.
    - `sudo su` → cambia l'usuari passant a ser root.

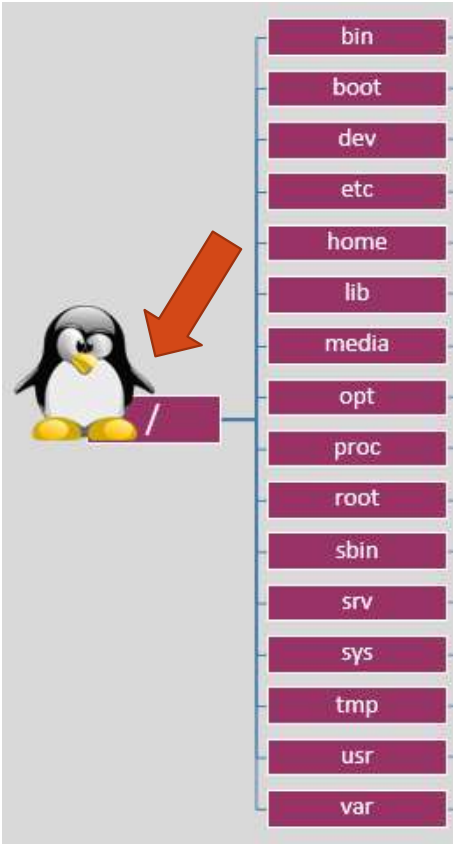
L'ARBRE DE DIRECTORIS DE LINUX

- L'estructura dels directoris de Linux, així com el seu contingut i funcions, ve definida en el denominat Filesystem Hierarchy Standard o FHS.
- Tot l'arbre de directoris partix d'una arrel comuna denominada **root** i que es simbolitza per una barra inclinada (/).



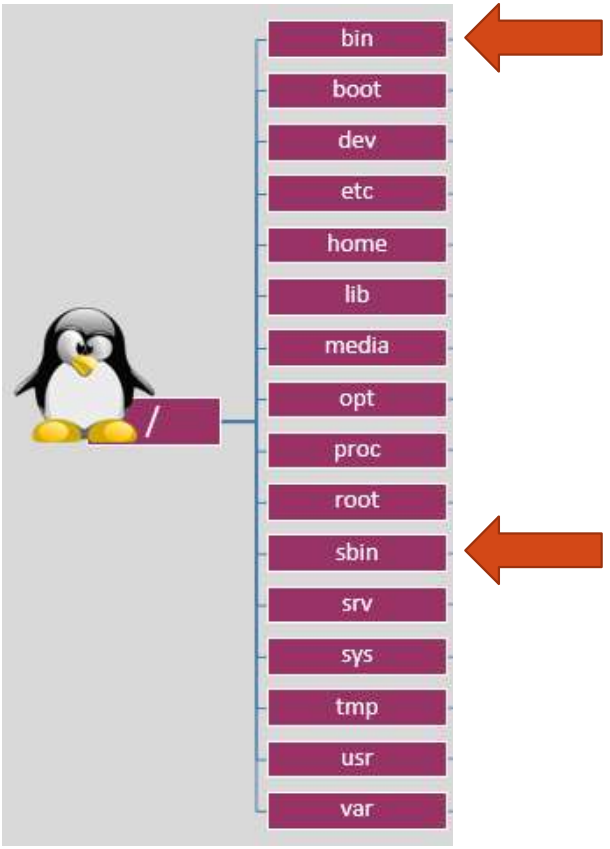
DIRECTORI ARREL, ROOT O /

- Tota l'estructura de directoris part d'un directori arrel també cridat directori root i que se simbolitza per una barra inclinada o /.
- Des d'este directori naixen tota la resta de directoris, independentment que estiguen emmagatzemats físicament en discos o unitats separades.
- Qualsevol direcció d'arxiu o carpeta en Linux comença pel directori arrel o /, seguit de tots els directoris i subdirectoris que ho contenen, separats cada un d'ells per /.
- Això es diu ruta d'un arxiu o directori.



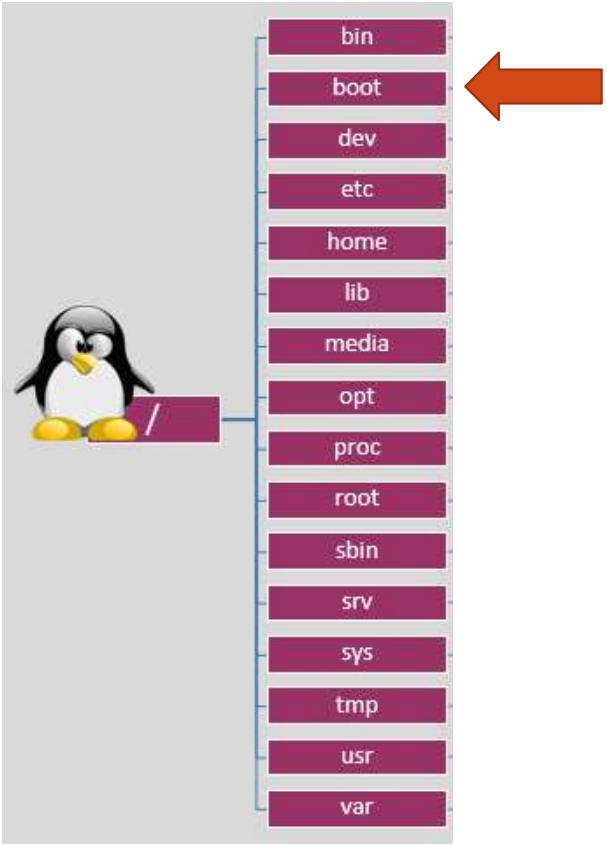
DIRECTORIS /BIN I /SBIN

- /bin
  - És un directori on s'emmagatzemen tots els executables d'usuari per a garantir les funcions bàsiques d'estos.
  - Inclou també els executables de diverses utilitats estàndard de la terminal de Linux.
- /sbin
  - És un directori on s'emmagatzemen tots els executables necessaris per a tasques administratives gestionades per l'usuari root o superusuari del sistema.
  - Inclou tasques pròpies del SO com ara l'arrancada, tasques de restauració, reparació, etc.



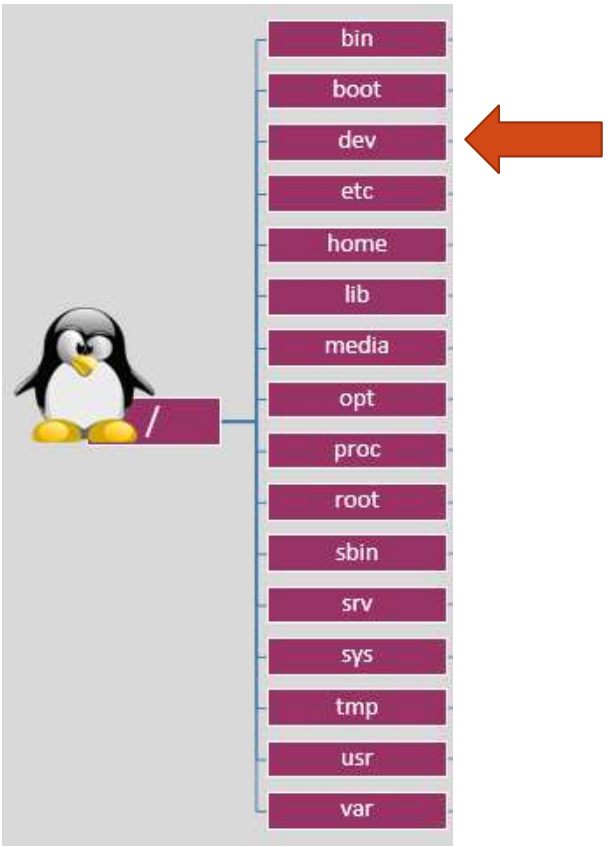
DIRECTORI /BOOT

- /boot
  - Inclou tots els executables i arxius que són necessaris en el procés d'arrancada del sistema
  - És també on es troba el gestor d'arrancada GRUB.
  - Es habitual que estiga en la seua pròpia partició separada de la resta.
  - En estos casos, durant la instal·lació del SO és important preveure bé l'espai que li anem a donar a la partició, ja que les diferents actualitzacions del Kernel poden deixar-la sense espai.
  - Si açò succeïx, tindrem problemes a l'hora d'instal·lar futures actualitzacions del nucli.



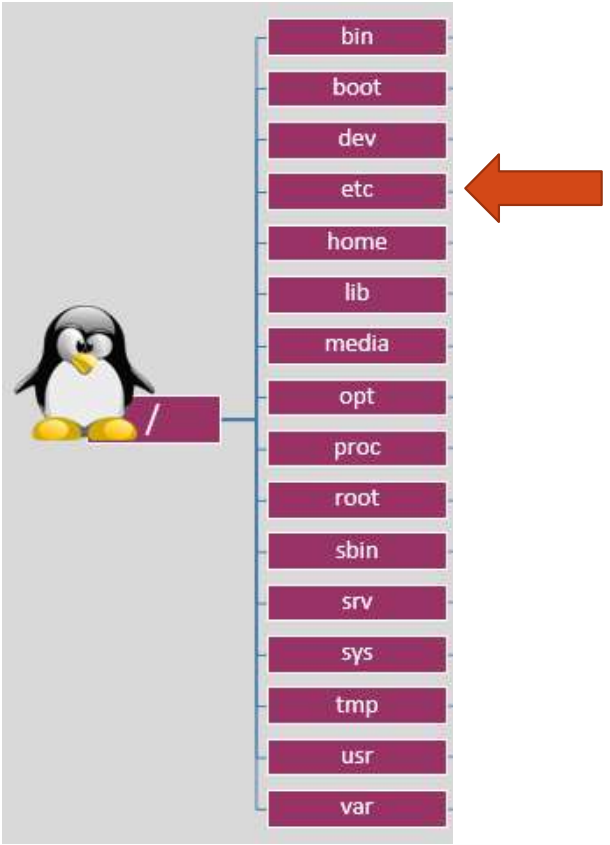
DIRECTORI /DEV

- /dev
  - Inclou tots els dispositius d'emmagatzemament, en forma d'arxius, connectats al sistema.
  - Qualsevol disc dur, partició o CDROM connectat al sistema i que este puga entendre com un volum lògic es "muntarà" com si fora una carpeta.
  - Per tant, la ruta de qualsevol unitat sempre començarà per /dev.
  - Aquest directori conté els punts de muntatge, però no la informació real d'estos discos.
  - Per exemple, executant `sudo fdisk -l`...
    - /dev/sda1 - Partició d'arrancada
    - /dev/sda5 - Partició *Swap*
    - /dev/sda6 - Partició Arrel
    - /dev/sda6 - Partició Home



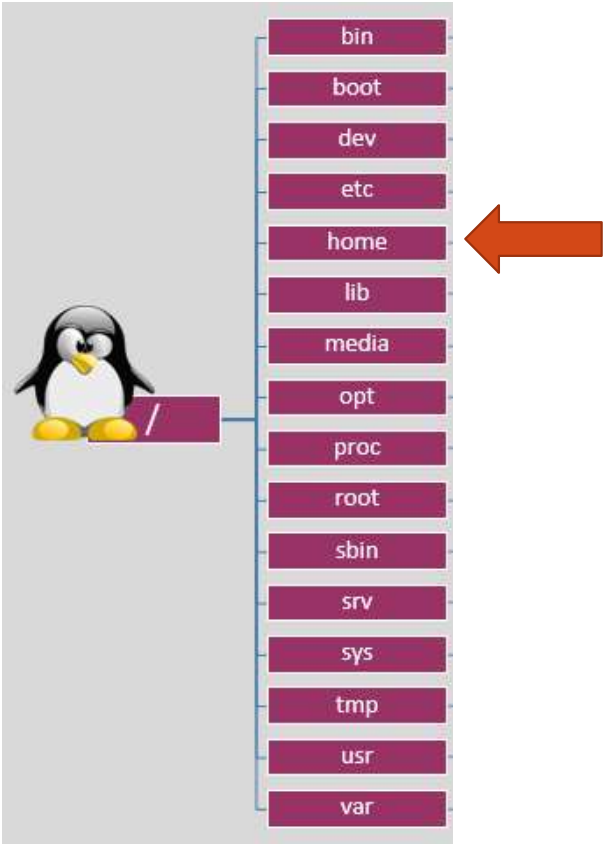
DIRECTORI /ETC

- /etc
  - És l'encarregat d'emmagatzemar els arxius de configuració tant a nivell de components del propi SO, com dels programes i aplicacions instal·lades a posteriori.
  - És un directori que hauria de contindre únicament fitxers de configuració, però no executables.



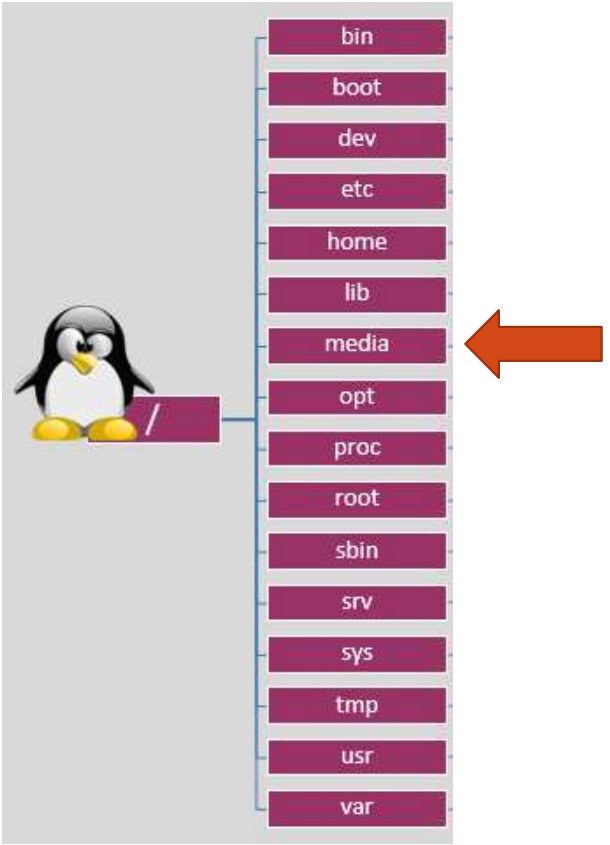
DIRECTORI /HOME

- /home
  - És el directori dels usuaris estàndard, i per tant, el destinat a emmagatzemar tots els arxius de l'usuari.
  - També inclou arxius de configuració de programes, etc.
  - Dins de /home estàn els directoris personals de tots els usuaris, anomenats segons el nom d'usuari.
  - Per exemple, un sistema amb dos usuaris Ana i Jaume tindria la següent estructura:
    - /home/Ana
    - /home/Jaume
  - Així mateix, cada directori d'usuari conté diferents carpetes de classificació: Documents, Imatges, Música, Plantilles, Vídeos…
  - Pot situar-se en una partició propia separada de la resta.



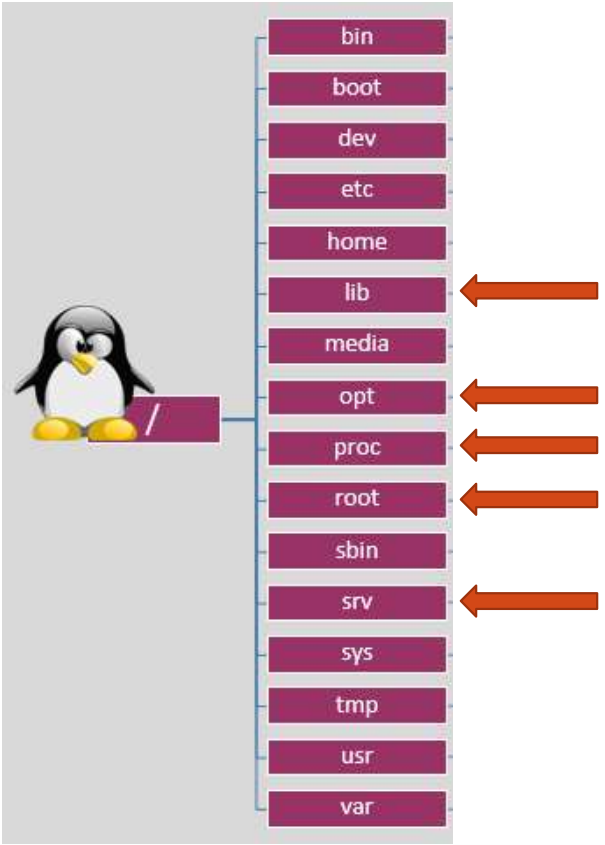
DIRECTORI /MEDIA

- /media
  - La seua funció és molt pareguda a la del directori /dev.
  - És el punt de muntatge de tots els volums lògics que es munten temporalment, com unitats USB, discos externs, etc.
  - Si en un sistema hi ha diversos usuaris, posem Ana i Jaume, els punts de muntatge dels volums que munten cada un d'ells es mostrarien en directoris separats:
    - /media/Ana
    - /media/Jaume



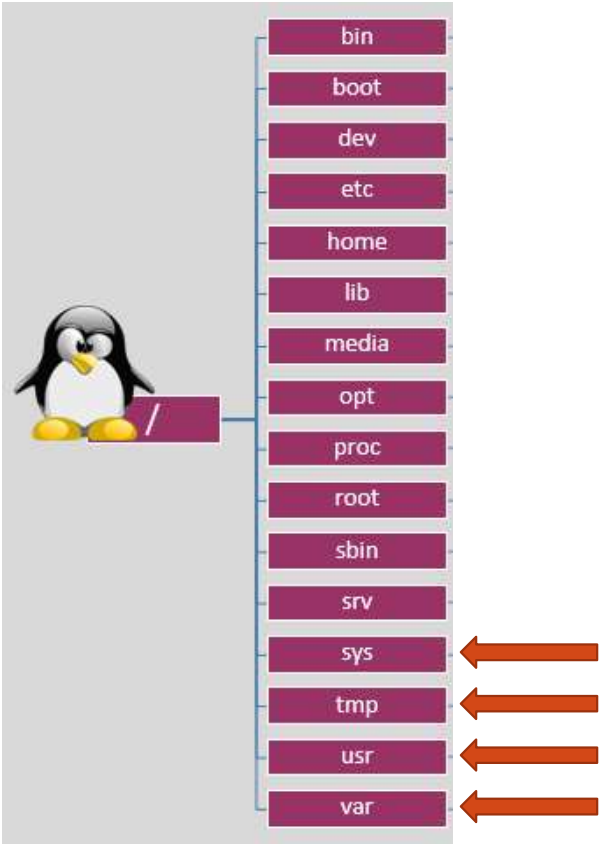
RESTA DE DIRECTORIS

- /lib:
  - Biblioteques essencials i mòduls del propi kernel.
- /opt:
  - Utilitats i software auto-contingut.
- /proc:
  - Informació de les aplicacions en execució.
- /root:
  - Directori /home de l'usuari root o superusuari del sistema.
- /srv:
  - Arxius i directoris de servidors instal·lats (www, ftp...).



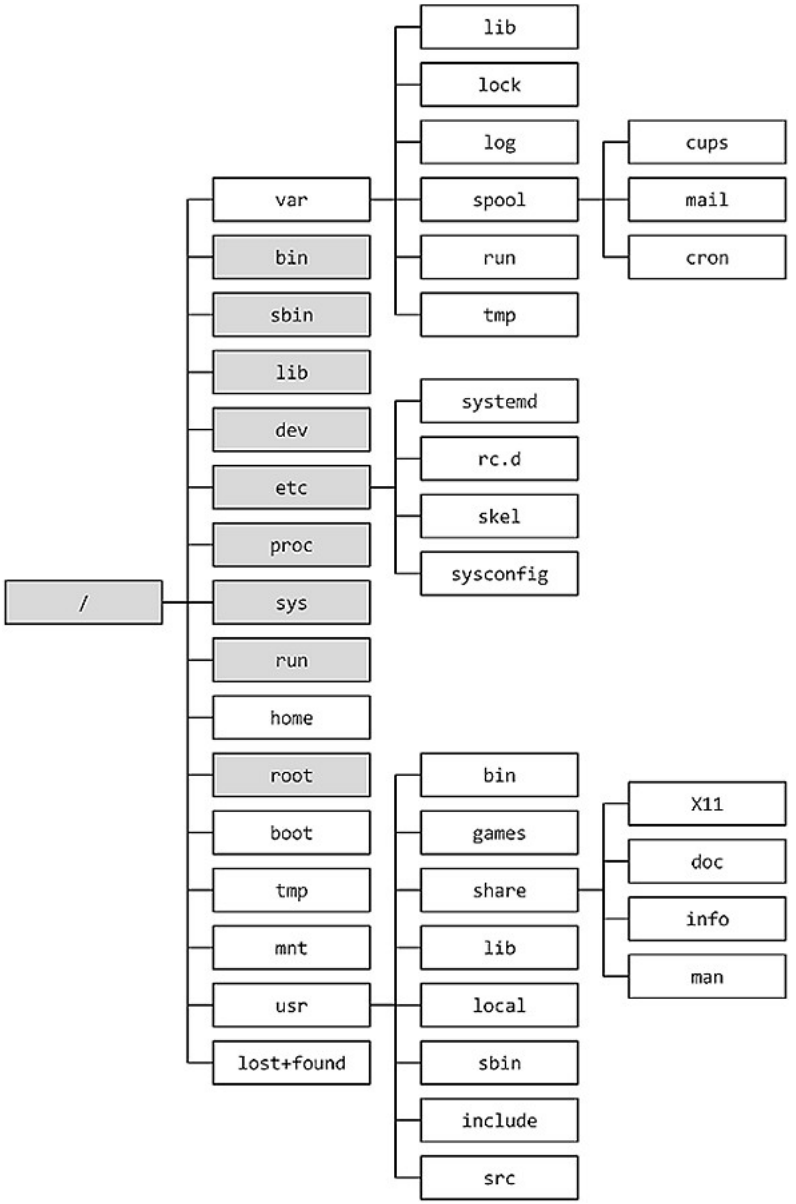
RESTA DE DIRECTORIS

- /sys:
  - Arxius virtuals amb informació del kernel.
- /tmp:
  - Arxius temporals de qualsevol tipus. Es buida al reiniciar.
- /usr:
  - Ve de "User System Resources". Inclou utilitats d'usuari i software instal·lat per mitjà de gestors de paquets.
- /var:
  - Informació del sistema, com logs, bases de dades, caché...



L'ARBRE DE LINUX

- En instal·lar una distribució de Linux, és possible crear, a més de la partició principal que conté /, altres particions dedicades a certs directoris de l'arbre.
- No obstant això, els directoris indispensables per a iniciar el sistema han d'estar en la mateixa partició que / i no poden, per tant, instal·lar-se en una partició separada.
- Eixos directoris essencials son els que apareixen en gris al següent esquema.



NOMS D'ARXIUS I DE DIRECTORIS

- Un arxiu es l'estructura que conté dades de l'usuari.
- Els arxius estàndard estan constituïts per una sèrie de bytes i tenen un format que no ve imposat pel sistema, sinó per les aplicacions.
- Els directoris són arxius especials que poden contindre molts altres arxius (directoris o no).
- Açò permet organitzar els arxius de forma jeràrquica en la ja coneguda estructura d'arbre.
- Els termes "directori" i "arxiu" de Linux són equivalents a "carpeta" i "document"empleats habitualment en Windows i Mac OS.

NOMS D'ARXIUS I DE DIRECTORIS (1)

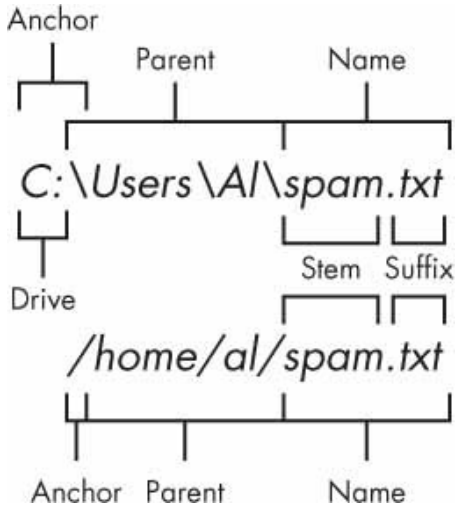
- Els noms d'arxius i de directoris en Linux responen a certes regles d'escriptura:
  - La distinció de majúscules i minúscules és determinant en la sintaxi dels noms d'arxius.
    - els arxius val, Val i VAL són distints.
  - Només han d'utilitzar-se els caràcters alfanumèrics (a-z, A-Z i 0-9) junt amb alguns més (_ . \ @ - +) en els noms d'arxius.
  - Alguns caràcters (com * i /) tenen un significat especial en línia de comandos. Han d'evitar-se.
  - Els signes - i + han d'evitar-se al principi d'un nom d'arxiu perquè pot haver ambigüitat amb algunes opcions de la línia de comandos.

NOMS D'ARXIUS I DE DIRECTORIS (2)

- Els noms d'arxius i de directoris en Linux responen a certes regles d'escriptura (cont):
  - A Windows, els últims caràcters darrere del punt són la extensió i determinen el tipus d'arxiu. A Linux no signifiquen res. Un programa.exe pot no ser executable.
  - Els noms poden contindre espais ( ) però s'han d'usar d'una forma especial:
    - Tenint un directori anomenat *dades empresa*…
      - `cd dades empresa` → no entrem en el directori.
      - `cd 'dades empresa'` ó `cd dades\ empresa` → les dos sintaxis són correctes.
  - Per aconseguir ocultar un arxiu només hem de començar el seu nom per punt (.)
    - `.ocult` → estarà ocult, i només es mostrarà afegint al comando `ls` l'opció `-a` (`ls -a`).

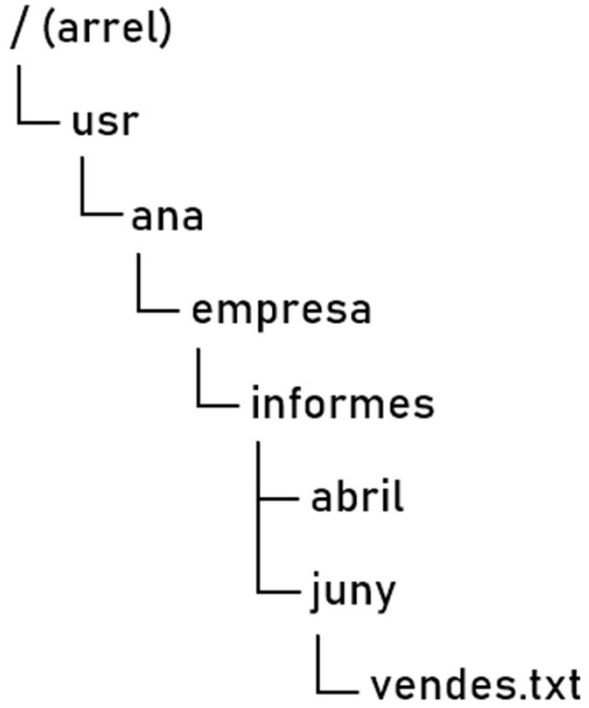
RUTES EN DISC

- Conéixer el nom d'un arxiu (o d'un directori) no és prou per a accedir a ell; un mateix nom pot estar associat a distints arxius en directoris diferents.
- La referència plenament qualificada d'un arxiu es denomina "ruta" i indica el directori en què es troba l'arxiu.
- A diferencia de Windows, on el separador és \ (barra inversa), a Linux els noms de directoris i d'arxius van separats per una / (barra obliqua) en les rutes.
- Hi ha tres tipus de rutes que poden utilitzar-se indiferentment en anomenar arxius per terminal:
  - absolutes
  - relatives
  - personals



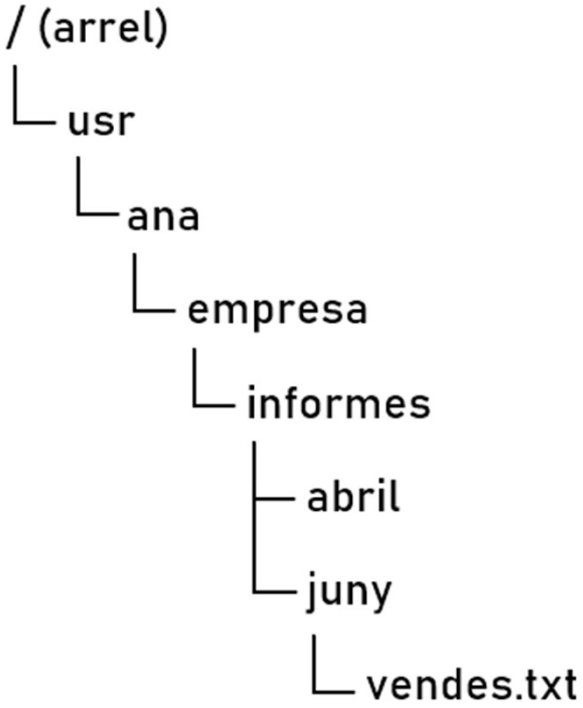
RUTES RELATIVES

- Considerem la línia següent:
  - /usr/ana/empresa/informes/juny/vendes.txt
- Hi ha diverses formes d'accedir al dit arxiu, depenent de quin siga el "directori actual":
  - Des de /
    - usr/ana/empresa/informes/juny/vendes.txt
  - Des de /usr/ana
    - empresa/informes/juny/vendes.txt
  - Des de /usr/ana/empresa/informes/junio
    - vendes.txt
  - Des de /usr/ana/empresa/informes/abril
    - ../juny/vendes.txt



RUTES ABSOLUTES

- Si no sabem o no ens interesa quin és el "directori actual", sempre haurem d'usar la ruta sencera per arribar a qualsevol arxiu. Això es coneix com a *ruta absoluta*.
  - /usr/ana/empresa/informes/juny/vendes.txt
- Les rutes absolutes es poden utilitzar des de qualsevol directori de l'arbre de directoris i sempre serán vàlides.
- La ruta a un arxiu que s'aconseguix en funció al "directori actual" és una *ruta relativa*.
- Si una ruta comença amb "/" serà una *ruta absoluta*. Qualsevol altra serà una *ruta relativa*.
- Tots els programes de Linux poden manejar tant rutes absolutes com relatives.



COMANDOS D'EXPLORACIÓ DE L'ARBRE

- pwd (*print working directory*)
  - Mostra la ruta absoluta del directori actual on es troba l'usuari
- cd (*change directory*) → `cd directori`
  - Permet canviar de directori actual. S'afegix com a argument el directori al que es vol anar.
  - Admet rutes absolutes i relatives com a arguments.
  - Si es crida sense arguments ens torna al directori personal de l'usuari.
  - Hi ha alguns directoris especials:
    - **.** : es el directori actual
    - **..** : puja al directori pare, el que conté al actual. S'ha d'afegir un espai entre el comando i els dos punts
    - **~** : és el directori home del usuari conectat
- ls (*list*) → `ls [-opc]`
  - Permet llistar el contingut del directori actual i admet moltes opcions.
  - Són molt útils –alF que llista tots els arxius amb detalls, inclús els ocults.
  - També –R que llista recursivament el contingut del directori i dels seus subdirectoris.

COMANDOS DE DIRECTORIS

- mkdir (*make directory*) → `mkdir [-opc] dir1 dir2…`
  - Permet crear directoris.
  - Admet com a argument el nom o noms del directoris que volem crear.
  - L'opció –p (pares) crea el directori que volem i tots els directoris en el seu camí si no existixen.
- rmdir (*remove directory*) → `rmdir [-opc] dir`
  - Elimina el directori que se li pasa com a argument si està buit.
  - Admet algunes opcions com –p que elimina el directori i tots els seus avantpassats.

COMANDOS D'ARXIUS

- touch → `touch [-opc] arxiu1 arxiu2...`
  - Si l'arxiu passat com a argument no existix, el crea, si no existix, canvia les dates de darrer accés i darrera modificación.
- cp (copy) → `cp [-opc] <origen ...> <destí>`
  - Copia l'arxiu o arxius origen a un nou directori o amb un nou nom, o les dos coses al mateix temps.
  - Admet moltes opcions, entre elles –R que copia recursivament tots els arxius de l'origen.
- mv (move) → `mv [-opc] <origen ...> <destí>`
  - Mou l'arxiu o directori origen a una nova ubicación.
  - També s'utilitza per a cambiar el nom a un arxiu o directori.
- rm (remove) → `rm [-opc] arxiu1 arxiu2…`
  - Elimina l'arxiu o arxius passats com a argument.
  - Per defecte no elimina directoris.
  - Els arxius borrats amb `rm` son recuperables. Per evitar això podem usar el comando `shred`.

- `sudo` → execució com a superusuari
- `rm` → eliminación d'arxius
- `-r` → borra directoris i el seu contingut recursivament
- `-f` → no demana confirmació a l'usuari
- `/*` → tots els arxius des de el directori / (arrel del disc)

es.gizmodo.com/asi-fue-como-pixar-rescato-toy-story-2-tras-eliminarla-1755147725

www.youtube.com/watch?v=8dhp_20j0Ys

COMANDOS DE CONTINGUT D'ARXIUS

- cat → `cat arxiu1 arxiu2`
  - Mostra per pantalla el contingut dels arxius de text passats com a argument.
- less → `less arxiu`
  - Permet mostrar el contingut de arxius de text, peró pàgina per página.
  - Les tecles útils son:
    - [Intro] Desplaça el text línia a línia.
    - [Espai] Desplaça el text pàgina per pàgina.
    - [q] Ix del programa.

COMANDOS DE EDICIÓ D'ARXIUS DE TEXT

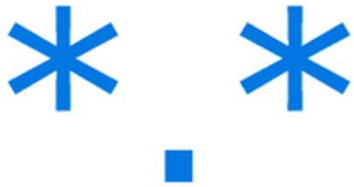
- vi → `vi arxiu`
  - L'editor ***vi*** és l'editor clàssic de Linux i es troba tant en versions antigues com modernes.
  - És un editor que per als principiants és complicat i prou confús. Dona vertader por enfrontar-se per primera vegada a ell!
  - La part positiva es que pot usar-se en qualsevol terminal i sempre el tindrem disponible.
  - Existix una versió millorada de nom ***vim***.
- nano → `nano arxiu`
  - ***GNU nano*** és un editor de text minimalista i amigable.
  - És especialment útil per a modificar arxius de configuración, crear llançadors…
  - Suporta sintaxi per colors, raó per la qual també pot ser utilitzat per a escriure codi.
- gedit → `gedit arxiu`
  - ***gedit*** és un editor gràfic molt útil, pero inservible des de terminal.

REDIRECCIÓ D'ENTRADA/EIXIDA

- Com ja hem comentat a Linux tot es un arxiu… inclús la pantalla i el teclat!
- Podem fer que l'eixida d'un comando no siga la pantalla sino un arxiu:
  - `comando > arxiu` → `ls -l > llistat.txt`
    - El resultat d'executar `ls -l` no eixirà per pantalla sino que es guardarà a l'arxiu `llistat.txt`
    - Si `llistat.txt` existix es sobrescriurà, i si no existix es crearà.
    - Per evitar això podem activar l'opció `noclobber` amb `set -o noclobber` (+o la reactiva).
  - `comando >> arxiu` → `ls -a >> llistat.txt`
    - El resultat d'executar `ls -a` s'afegirà al final del contingut de l'arxiu `llistat.txt`
- També podem redirigir l'eixida dels errors d'un comando a un arxiu:
  - `comando 2> arxiu` → `ls -l /root > errors.log`
    - El comando no pot executar-lo un usuari normal. El error es guardarà a l'arxiu `errors.log`.
  - `comando &> arxiu` → `rm arxiu_inexistent &> eix-errors.log`
    - L'arxiu no existix aixina que tant l'eixida normal com l'error es guardarà a l'arxiu `eix-errors.log`.

COMODINS I EXPRESIONS REGULARS

- Els comodins són patrons de busca.
- Un patró és una seqüència de caràcters que ens permet seleccionar paraules, frases o seqüències de caràcters.
- Per exemple, un patró, pot ser d'utilitat per a seleccionar tots aquells arxius d'un directori, que comencen per una determinada lletra o seqüència de lletres.
- No obstant això, els comodins i expressions regulars són molt més i ens obrin un ventall de possibilitats extraordinari.
- Tots els comodins poden utilitzar-se conjuntament.



COMODINS I EXPRESIONS REGULARS

- ? → Representa un únic carácter:
  - `ls /dev/sda?`: llista tots els arxius que siguen igual a /dev/sda més un únic carácter
    - /dev/sda1, /dev/sda2, /dev/sda3…
- * → Aquest comodí pot representar res o qualsevol quantitat de caràcters i dígits.
  - `ls /dev/sd*`: llista tots els arxius que siguen igual a /dev/sd més qualsevol quantitat de caràcters
    - /dev/sda, /dev/sda1, /dev/sda2, /dev/sda3, /dev/sdb, /dev/sdb1,…
- [ ] → Aquest comodí representa un rang, ya siga de caràcters o de números.
  - `ls /dev/sda[1-5]`: llista tots els arxius que comencen per /dev/sda i després tinguen un número entre 1 i 5
    - /dev/sda1, /dev/sda2, /dev/sda3, /dev/sda4, /dev/sda5.
  - Altres exemples: `[A-Z]`, `[a-z]`, `[0-9]`, `[a-zA-z]`, `[a-zA-Z0-9]`.

COMODINS I EXPRESIONS REGULARS

- {} → Representa un conjunt de comodins separats per comes (sense espais alrededor).
  - `ls {sda*,sdb*}`: llista tots els arxius que comencen per sda o sdb.
    - /dev/sda1, /dev/sda2..., /dev/sdb1, /dev/sdb2...
  - `rm {*.doc,*.docx}`: elimina els arxius amb extensió "doc" o "docx".
- [!] → Llista tot el que no estiga dins dels corxets. També admitix rangos.
  - `ls /dev/sda[!1-5]`: llista tots els arxius que comencen per /dev/sda, amb un número després que no estiga entre 1 i 5
    - /dev/sda6, /dev/sda7...

CONCATENACIÓ DE COMANDOS

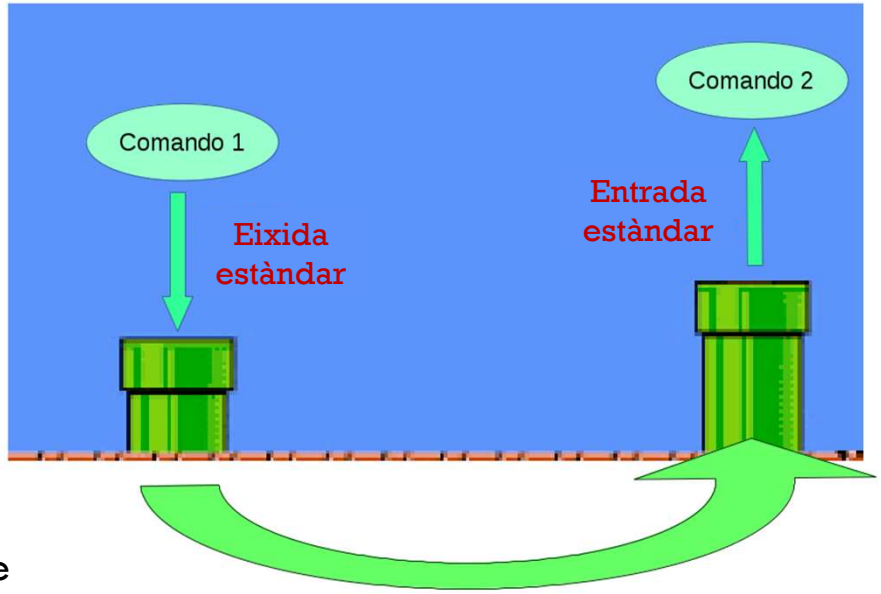
- A més de les tuberíes (|) també podem concatenar comandos d'altres formes:
  - Amb "&" els dos (o més) comandos se executaràn de manera simultània.
    - `$ mkdir hola & mkdir adeu`: sempre es creen els dos directoris
  - Amb "||" el segón comando s'executarà només si el primer termina sense éxit.
    - `$ mkdir hola || mkdir adeu`: si no es pot crear hola, es crearà adeu
  - Amb "&&" segón comando s'executarà només si el primer termina amb éxit.
    - `$ mkdir hola && mkdir adeu`: si es crea hola, també es crearà adeu, si no, no es crearà.
  - Amb ";" el segón comando s'executarà sense tindre en compte el resultat del primer.
    - `$ mkdir hola ; mkdir adeu`: adeu es crearà tant si hola s'ha pogut crear, com si no.

CONCATENACIÓ DE COMANDOS

- Podem usar "&&" i "||" en combinació per a crear una estructura if-then-else:
  - "&&" és equivalent a un AND lògic i "||" a un OR lògic
  - `comando && comando_veritat || comando_fals`
  - Si comando s'executa sense errors, aleshores executem comando_veritat, si no executem comando_fals.
    - `rmdir directori_ple && echo Directori esborrat! || No s'ha pogut esborrar`
    - Usem `echo` per a mostrar un missatge tant si `rmdir` ha pogut borrar el directori com si no.
- Amb "&" al final d'un comando, el shell no espera a que s'acabe d'executar.
  - `$ ls -R / > tots.txt &`: mentre es fa el llistat recuperem el control del prompt, sense esperar a que termine la seua execució. Rebrem un missatge quan es termine d'executar.

TUBERÍES (PIPES) |

- Per mitjà de les tuberíes (pipes) també podem encadenar comandos, fent que l'eixida d'un comando siga l'entrada del següent:
  - comando1 | comando2 → ls -l | less
    - El llistat del directori es passarà al comando `less` que ho mostrarà página a pàgina
- L'eixida del segón comando també pot enviar-se a un tercer per mitjà d'una tubería:
  - comando1 | comando2 | comando3 → ls | rev | less
    - El llistat del directori es passarà al comando `rev` que li dona la volta a les paraules, i el resultat es pasa a `less` que ho mostrarà página a pàgina



$ COMANDO_1 | COMANDO_2

TUBERÍES (PIPES) |

- Algunes ferramentes útils per a usar conjuntamente amb canonades són:
  - `head`: extrau les primeres línies d'un arxiu:
    - `head -1` → trau la primera línia de l'arxiu.
  - `tail`: extrau les darreres línies d'un arxiu:
    - `tail -1` → trau la darrera línia de l'arxiu.
  - `cut`: trau per pantalla part d'una línia d'un arxiu:
    - `cut -d" " -f3` → extrau el camp 3 de la línia, sempre que els camps estiguen separats per espais.
  - `sort`: ordena les línies d'un arxiu de text:
    - `sort -f arxiu.txt` → ordena les línies de l'arxiu sense distingir majúscules i minúscules

- **Exemple:** escrivint el comando `ls -lah` obtenim la següent eixida:

```
total 152K
drwxrwxr-x  2 jose jose 4,0K ene 10 19:16 .
drwxr-xr-x 18 jose jose 4,0K ene 10 18:52 ..
-rw-rw-r--  1 jose jose 141K ene 10 19:16 10.txt
```

  - Si només volem saber el tamany de l'arxiu 10.txt…
    - `ls -lah | tail -1 | cut -d" " -f6`

```
jose@jose-Ubuntu:~/dades$ ls -lah | tail -1 | cut -d" " -f6
141K
```

TUBERÍES (PIPES) |

- Algunes ferramentes útils per a usar conjuntamente amb canonades són:
  - `grep`: obté les línies d'un arxiu que complixen un patró:
    - `grep -i -n miquel alumnes.txt` → trau les línies de l'arxiu que contenen "miquel" sense distingir majúscules i minúscules, i prefixant el número de línia.
  - `find`: busca arxius a un directori:
    - `find ~ -type f -name *.txt` → buscar arxius, no directoris, amb una extensió .txt.
  - `wc`: conta el número de línies, paraules o caràcters dins d'un arxiu:
    - `wc -l arxiu.txt` → conta el número de línies de l'arxiu.
- **Exemple:**
  - `ls -l | grep u | wc -l`
    - Llistem el contingut del directori actual → busquem elements que continguen la lletra "u" → contem el número de llínies
    - En realitat, estem contant la quantitat d'arxius que contenen la lletra "u" al seu nom.

UTILITATS PER A TUBERÍES – GREP

- El comando `grep` (Globally Regular Expressions Pattern) busca patrons en fitxers.
- Per defecte torna totes les línies que contenen un patró (cadena de text) determinat en un o més fitxers, però utilitzant les opcions que admet es pot variar molt aquest comportament.
- Si no se li passa cap fitxer com a argument, fa la busca en la seua entrada estàndar.
  - Sintaxi: `grep [opcions] <patró> [fitxers]`
- Algunes opcions interesants:
  - `-c`: torna només la quantitat de línies que contenen el patró.
  - `-i`: ignora les diferencies entre majúscules y minúscules.
  - `-H`: a més de les línies, torna el nom del fitxer a on ha trobat el patró.
  - `-l`: en múltiples fitxers nomes mostra els noms dels fitxers a on ha trobat el patró, no les línies.
  - `-v`: torna les línies que no contenen el patró.
  - `-r`: busca en un directori de forma recursiva.
  - `-n`: imprimix el número de cada línia que conté el patró.
  - `-E`: permet buscar més d'un patró al mateix temps.

UTILITATS PER A TUBERÍES – GREP

- Alguns exemples de l'ús de `grep`:
  - `grep -i alumne arxiu.txt`
    - La busca es insensible a majúscules, per tant trobarà alumne, Alumne, ALUMNE, aLumNe, ALumNE...
  - `grep -v alumne arxiu.txt`
    - La busca torna les línies que no contenen la paraula *alumne*.
  - `grep -A1 alumne arxiu.txt`
    - A més de la línia que continga *alumne*, també es mostrarà la següent línia. Admet qualsevol número.
  - `grep -B1 alumne arxiu.txt`
    - A més de la línia que continga *alumne*, també es mostrarà la anterior línia. Admet qualsevol número.
  - `grep -C1 alumne arxiu.txt`
    - A més de la línia que continga *alumne*, també es mostraràn les línies anterior i següent.
  - `grep –l –r bash /etc/*`
    - Busca la paraula *bash* a tots els arxius del directori /etc i els seus subdirectoris, i només mostra el nom del arxius que la continguen

EXPRESIONS REGULARS EN GREP

- Linux permet utilitzar expresions regulars i comodins com a patrons de busca:
  - . → un únic carácter:
    - `grep 20.. /arxiu.txt`
  - [ ] → rango o secuencia de caràcters o números:
    - `grep [A-D] /etc/passwd`
  - ^ → Llista totes les línies que comencen pel text que posem darrere d'ell.
    - `grep ^[p-r] /etc/passwd`: llista totes les línies que comencen per *p, q* i *r*, destacant la inicial
  - $ → Llista totes les línies que terminen pel text que posem davant d'ell.
    - `grep bash$ /etc/passwd`: llista tot el que termine per la lletra *a.*
  - | → Llista totes les línies que continguen algú dels patrons especificats.
    - `grep -E "cp|mv|ls" .bash_history`: llista les línies del històric que continguen cp, mv o ls

EXPRESIONS REGULARS EN GREP

- Entre dos parells de corxets i dos punts també podem utilitzar algunes paraules reservades com a patrons de busca:
  - `[[:alnum:]]` – Caràcters alfanumérics.
  - `[[:alpha:]]` – Caràcters alfabètics.
  - `[[:blank:]]` – Caràcters buits : espai i tab.
  - `[[:digit:]]` – Dígits: '0 1 2 3 4 5 6 7 8 9'.
  - `[[:lower:]]` – Lletres minúscules: 'a b c d e f g h i j k l m n o p q r s t u v w x y z'.
  - `[[:upper:]]` – Lletres majúscules: 'A B C D E F G H I J K L M N O P Q R S T U V W X Y Z'.
    - `grep ^[[:upper:]]..[[:digit:]] arxiu.txt`
    - Línies que comencen per una majúscula, tinguen dos caràcters qualsevol i després un dígit.
    - `grep [[:blank:]][[:lower:]][[:lower:]]$ arxiu.txt`
    - Línies que terminen per un espai seguit de dos minúscules.

UTILITATS PER A TUBERÍES – CUT

- El comando cut s'utilitza per a obtindre una part de cada línia d'un arxiu, depenent d'un delimitador o una quantitat de caràcters.
  - Sintaxi: `cut [opcions] [fitxers]`
- Algunes opcions interesants:
  - `-cN-M`: talla des del caràcter número N fins al caràcter número M.
  - `-cN-`: talla des del caràcter número N fins al final.
  - `-c-N`: talla des del principi fins al caràcter número N.
  - `-cN,M`: talla el caràcter número N y el caràcter número M.
  - `-fn`: indica que talle el camp n.
  - `-d carácter`: indica el carácter que servirà com a separador de camps.
  - `-s`: suprimix les línies que no tinguen un caràcter delimitador de camp quan es gasta la opció -f.

UTILITATS PER A TUBERÍES – CUT

- Alguns exemples de l'ús de `cut`:
  - `cut -c2 arxiu.txt`
    - Mostra el segón caràcter de cada línia de l'arxiu.
  - `cut -c1-3 arxiu.txt`
    - Mostra els 3 primers caràcters de cada línia de l'arxiu.
  - `cut -c3- arxiu.txt`
    - Mostra per cada línia de l'arxiu, des del tercer caràcter fins al final.
  - `cut -c-5 arxiu.txt`
    - Mostra per cada línia de l'arxiu, des del principi fins al cinqué caràcter.
  - `cut -d':' -f1,3 /etc/passwd`
    - Dividix per camps cada línia amb el delimitador ":" i mostra el primer i el tercer d'ells.
  - `grep "/bin/bash" /etc/passwd | cut -d':' --complement -f7`
    - Amb grep busquem els usuaris que gasten /bin/bash com a Shell, dividim les línies per camps amb ":" i mostrem tots els camps excepte el seté.

UTILITATS PER A TUBERÍES – HEAD I TAIL

- Aquests dos comandos están molt conectats entre si perque permitixen extraure els primers y darrers elements d'un arxiu, respectivament.
  - Sintaxi: `head [opcions] [fitxer]`
    `tail [opcions] [fitxer]`
- Alguns exemples d'ús:
  - `head -4 arxiu.txt`: trau les quatre primeres línies de l'arxiu.
  - `tail -5 arxiu.txt`: trau les cinc darreres línies de l'arxiu.
  - `tail -n+10 arxiu.txt`: trau des de la línea 10 dins al final de l'arxiu.
  - `head -10 arxiu.txt | tail -5`: trau desde la línea 6 fins a la 10 (trau les 10 primeres línies i d'elles es queda amb les 5 últimes).
  - `ls –lrS /etc | tail -1`: trau l'arxiu més gran del directori /etc.

UTILITATS PER A TUBERÍES – FIND

- El métode més comú per a trobar i filtrar arxius a Linux es el comando `find`.
- Sintaxi: `find <directori> <opcions> <patró>`
- Algunes opcions interesants:
  - `-name patró`: arxius amb el patró en el seu nom (`-iname` per a insensibilitat a majúscules)
  - `-not`: arxius que no complixquen la següent opció.
  - `-maxdepth N`: només busca als N subdirectoris dins del directori sel·leccionat.
  - `-type [f,d,l]`: nomes troba arxius (f), directoris (d), enllaços (l)…
  - `-atime, -mtime`: troba arxius per temps d'últim accés o de modificació.
  - `-size N[c,k,M,G]`: troba arxius per tamany (c=bytes, k=kbytes, M=megabytes, G=gigabytes).
  - `-user, -group`: troba arxius propietat d'un usuari o d'un grup.
  - `-delete`: esborra els arxius trobats.

UTILITATS PER A TUBERÍES – FIND

- Alguns exemples d'ús de `find`:
  - `find . -name nom_arxiu`: troba l'arxiu amb eixe nom exacte al directori actual.
  - `find . -iname nom_arxiu`: troba l'arxiu amb eixe nom exacte sense distingir majs-mins.
  - `find . -name "*.txt"`: troba els arxius amb extensió .txt.
  - `find . -name "*.txt" -delete`: troba els arxius amb extensió .txt i els esborra.
  - `find . –maxdepth 1 -not –name "a*"`: troba només al directori actual (no recursivament) els arxius que no comencen per "a".
  - `find . –type d –name "?s*"`: troba els directoris que tinguen una "s" com a segona lletra.
  - `find . -atime 1`: troba els arxius als que es va acceder fa un día.
  - `find . -size +5M`: troba els arxius amb un tamany superior a 5 MB.
  - `find . -user joan`: troba els arxius que siguen propietat de l'usuari "joan".