

Taula de continguts

Introducció al llibre The Linux Command Line de Willlliam E. Shotts, Jr.....	2
1 Introducció a Linux.....	4
Exercici 1-1 (individual o grup fins a 3 persones).....	4
1.1 Instal·lació d'Ubuntu.....	5
1.1.1 Creació d'un disc d'instal·lació per a usuaris de Windows.....	6
1.1.2 Sessió en viu o LIVE SESSION.....	8
Exercici 1.1.2-1.....	12
Exercici 1.1.2-2.....	12
1.1.3 Instal·lació d'Ubuntu en un disc dur extern.....	13
Exercici 1.1.3-1.....	33
1.2 Organització del sistema d'arxiu de Linux.....	34
1.3 La línia de comando (Command Line Interface, CLI).....	38
1.3.1 Primers passos (pwd, cd, ls).....	39
1.3.2 Nom de ruta (pathname).....	45
Exercici 1.3.2-1.....	49
Exercici 1.3.2-2.....	50
1.3.3 Explorant el sistema d'arxiu (ls, file, less).....	52
Exercici 1.3.3-1.....	56
Exercici 1.3.3-1.....	57
Exercici 1.3.3-2.....	64
Exercici 1.3.3-3 (activitat 5 exercici 1).....	65
Exercici 1.3.3-4 (activitat 5 exercici 2).....	65
1.3.4 Manipulant arxius i directoris (cp, mv, mkdir, rm, ln, find).....	66
Exercici 1.3.4-1.....	68
Exercici 1.3.4-2.....	70
Exercici 1.3.4-3.....	74
Exercici 1.3.4-4.....	93

Introducció al llibre The Linux Command Line de Willliam E. Shotts, Jr.

I want to tell you a story. No, not the story of how, in 1991, Linus Torvalds wrote the first version of the Linux kernel. You can read that story in lots of Linux books. Nor am I going to tell you the story of how, some years earlier, Richard Stallman began the GNU Project to create a free Unix-like operating system. That's an important story too, but most other Linux books have that one, as well. No, I want to tell you the story of how you can take back control of your computer.

When I began working with computers as a college student in the late 1970s, there was a revolution going on. The invention of the microprocessor had made it possible for ordinary people like you and me to actually own a computer. It's hard for many people today to imagine what the world was like when only big business and big government ran all the computers. Let's just say you couldn't get much done.

Today, the world is very different. Computers are everywhere, from tiny wristwatches to giant data centers to everything in between. In addition to

ubiquitous computers, we also have a ubiquitous network connecting them together. This has created a wondrous new age of personal empowerment and creative freedom, but over the last couple of decades something else has been happening. A single giant corporation has been imposing its control over most of the world's computers and deciding what you can and cannot do with them. Fortunately, people from all over the world are doing something about it. They are fighting to maintain control of their computers by writing their own software. They are building Linux.

Many people speak of "freedom" with regard to Linux, but I don't think most people know what this freedom really means. Freedom is the power to decide what your computer does, and the only way to have this freedom is to know what your computer is doing. Freedom is a computer that is without secrets, one where everything can be known if you care enough to find out.

Why use the command line?

Have you ever noticed in the movies when the “super hacker”—you know, the guy who can break into the ultra-secure military computer in under 30 seconds—sits down at the computer, he never touches a mouse? It’s because movie makers realize that we, as human beings, instinctively know the only way to really get anything done on a computer is by typing on a keyboard.

Most computer users today are familiar with only the *graphical user interface (GUI)* and have been taught by vendors and pundits that the *command line interface (CLI)* is a terrifying thing of the past. This is unfortunate, because a good command line interface is a marvelously expressive way of communicating with a computer in much the same way the written word is for human beings. It’s been said that “graphical user interfaces make easy tasks easy, while command line interfaces make difficult tasks possible,” and this is still very true today.

Since Linux is modeled after the Unix family of operating systems, it shares the same rich heritage of command line tools as Unix. Unix came into prominence during the early 1980s (although it was first developed a decade earlier), before the widespread adoption of the graphical user interface and, as a result, developed an extensive command line interface instead. In fact, one of the strongest reasons early adopters of Linux chose it over, say, Windows NT was the powerful command line interface, which made the “difficult tasks possible.”

Aquestes apunts estan basats en [The Linux Command Line](#) de Willliam E. Shotts, Jr.

1 Introducció a Linux

Linux és un sistema operatiu gratuït i de codi obert. Això significa que es troba a la xarxa a disposició de qualsevol que vulgui descarregar o distribuir-lo. El codi obert permet modificar Linux, per això a la xarxa es troben infinitat de variants d'aquest sistema operatiu. Nosaltres utilitzarem una variant de Linux, anomenada Ubuntu.

Considero aquest apartat del curs TIC, en el que aprendrem a instal·lar i utilitzar Linux, el més important de tots, perquè conèixer el sistema operatiu i ser capaços de controlar-lo, és la base necessària, per a qualsevol altra cosa que vulguem fer amb un ordinador.

A primera vista, la utilització de la interfície de línia de comandos (Comand Line Interface, CLI) pot semblar complicada, realment no ho és. El que passa és que no estem acostumats a crear, copiar o moure arxius amb comandos, sinó a utilitzar icones per fer aquestes tasques, en el que s'anomena una interfície gràfica (Gràfic User Interface, GUI).

Per arribar a utilitzar amb comoditat la CLI, és necessari pràctica. Per adquirir-la, a més de fer els exercicis que proposaré, es convenient que intenteu utilitzar la CLI regularment per fer treballs rutinaris en el vostre sistema operatiu.

Exercici 1-1 (individual o grup fins a 3 persones)

Llegeix els articles sobre sistemes operatius del blog ([entrada del 30/12/15](#)) i fes una redacció, explicant quines funcions té el sistema operatiu i argumentant la importància que té per l'usuari poder controlar-ho.

Pots utilitzar informació addicional d'altres fonts. Entre 1000 i 1500 paraules.

Data entrega 25/01/16.

1.1 Instal·lació d'Ubuntu

A continuació s'explicarà com instal·lar Ubuntu en un disc dur extern. Respecte a instal·lar-ho en el disc dur intern, té l'avantatge, que no hi ha cap perill de danyar dades del disc dur intern, on està instal·lat el nostre sistema operatiu habitual, les nostres aplicacions i els arxius que normalment utilitzem.

A més, el disc dur extern el podem portar amb nosaltres, connectar-ho a qualsevol altre ordinador i disposar així d'un sistema operatiu mòbil.

Per a la instal·lació d'Ubuntu, procedirem en dues passes. Primer crearem un disc d'instal·lació i després utilitzarem aquest disc per instal·lar Ubuntu en el disc dur extern.

1.1.1 Creació d'un disc d'instal·lació per a usuaris de Windows

És necessari descarregar un arxiu **.iso** amb una imatge d'Ubuntu. Es tracta d'un arxiu que una vegada descarregat es pot gravar en un DVD que anomenarem DVD d'instal·lació.

La imatge d'Ubuntu es pot descarregar des de <http://releases.ubuntu.com/precise/> (Download Precise Pangolin 12.04 LTS)

Select an image

Ubuntu is distributed on three types of images described below.

Desktop CD

The desktop cd allows you to try Ubuntu without changing your computer at all, and at your option to install it permanently later. This type of cd is what most people will want to use. You will need at least 384MiB of RAM to install from this cd.

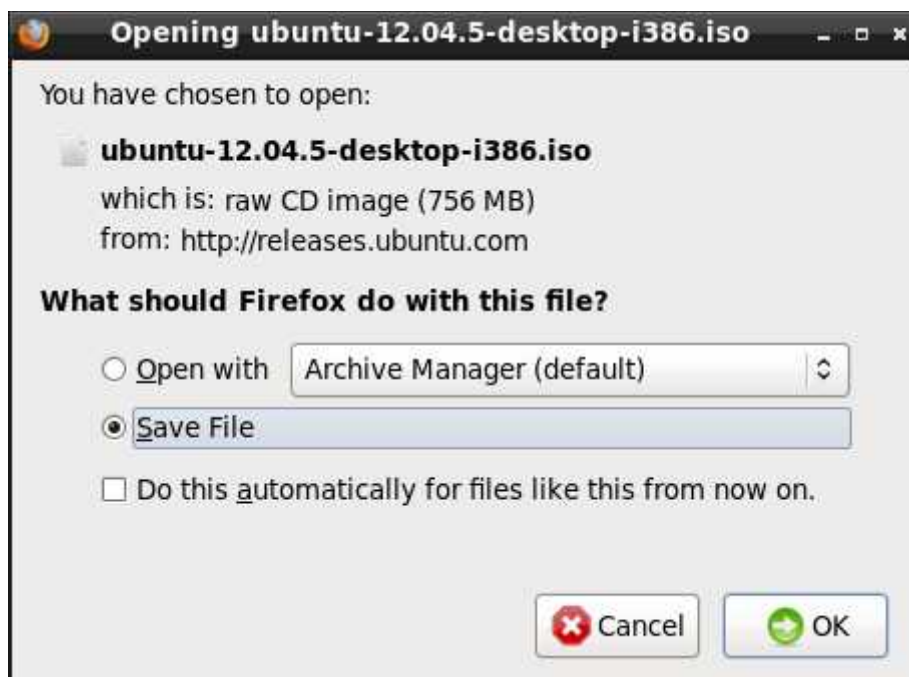
There are two images available, each for a different type of computer:

64-bit PC (AMD64) desktop CD

Choose this to take full advantage of computers based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). If you have a non-64-bit processor made by AMD, or if you need full support for 32-bit code, use the i386 images instead.

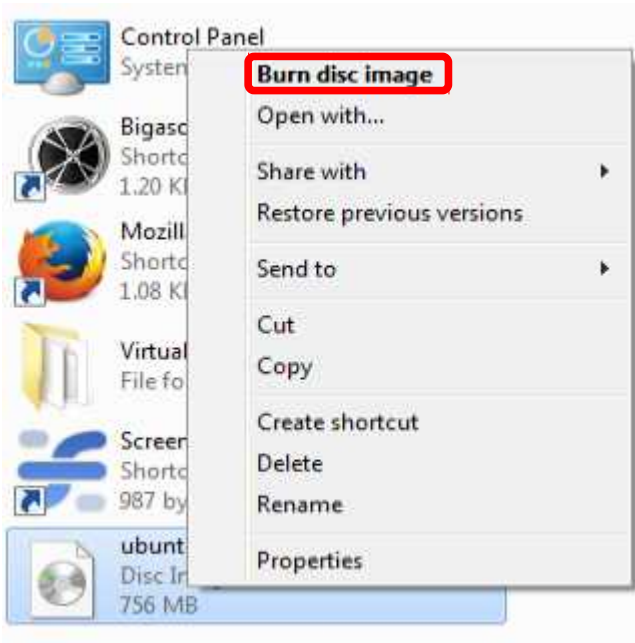
32-bit PC (i386) desktop CD

For almost all PCs. This includes most machines with Intel/AMD/etc type processors and almost all computers that run Microsoft Windows, as well as newer Apple Macintosh systems based on Intel processors. Choose this if you are at all unsure.



Una vegada descarregat el fitxer **ubuntu-12.04.5-desktop-i386.iso**, (aquest fitxer s'anomena imatge) desar-ho en l'escriptori.

Clicant amb el botó dret del ratolí damunt l'arxiu, s'obre la següent finestra.



Seleccionar **burn disc image** per crear un DVD d'instal·lació.

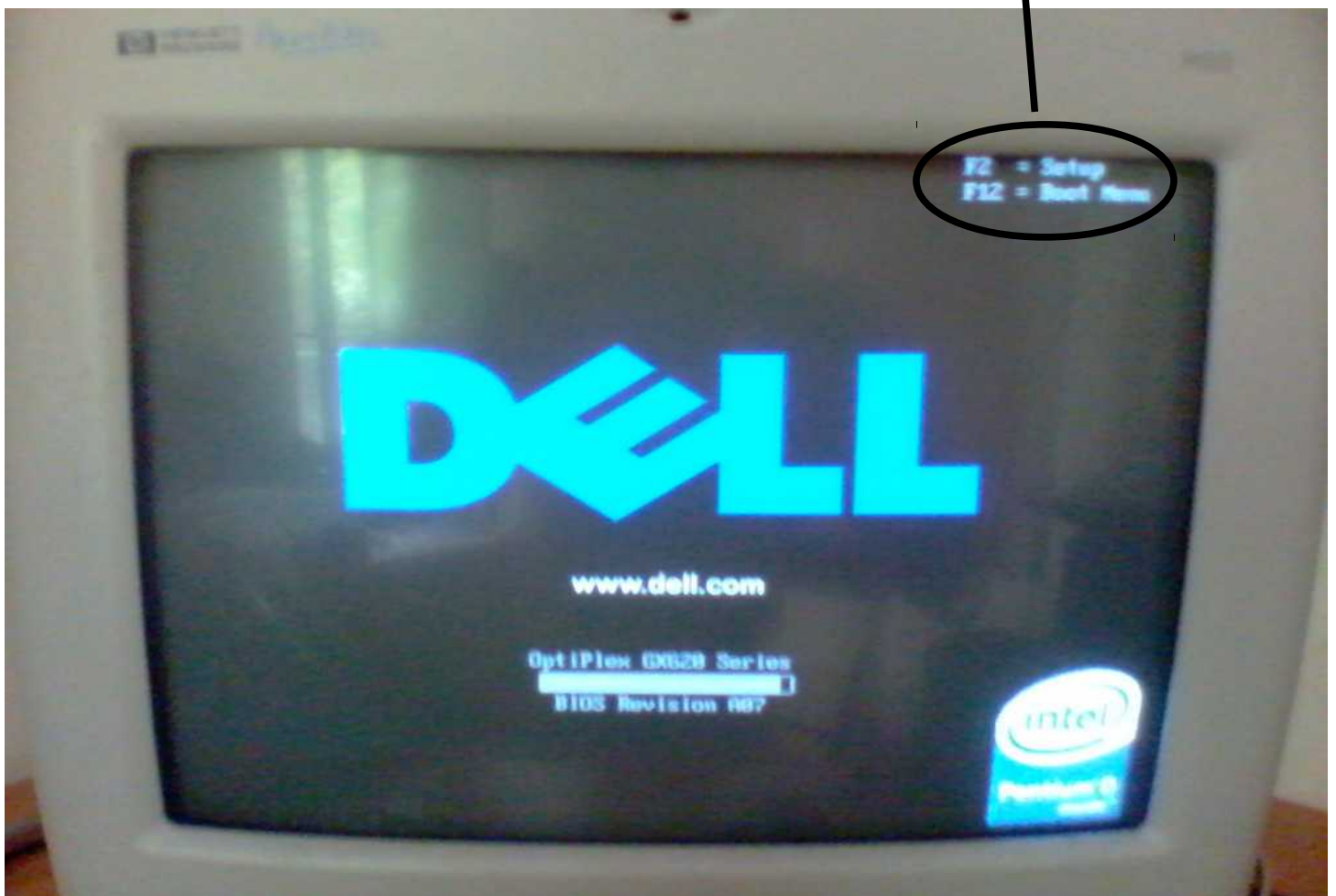
Si el teu sistema operatiu és OS X, trobaràs informació de com crear un disc d'inici en l'entrada del [11/01/16](#) del blog.

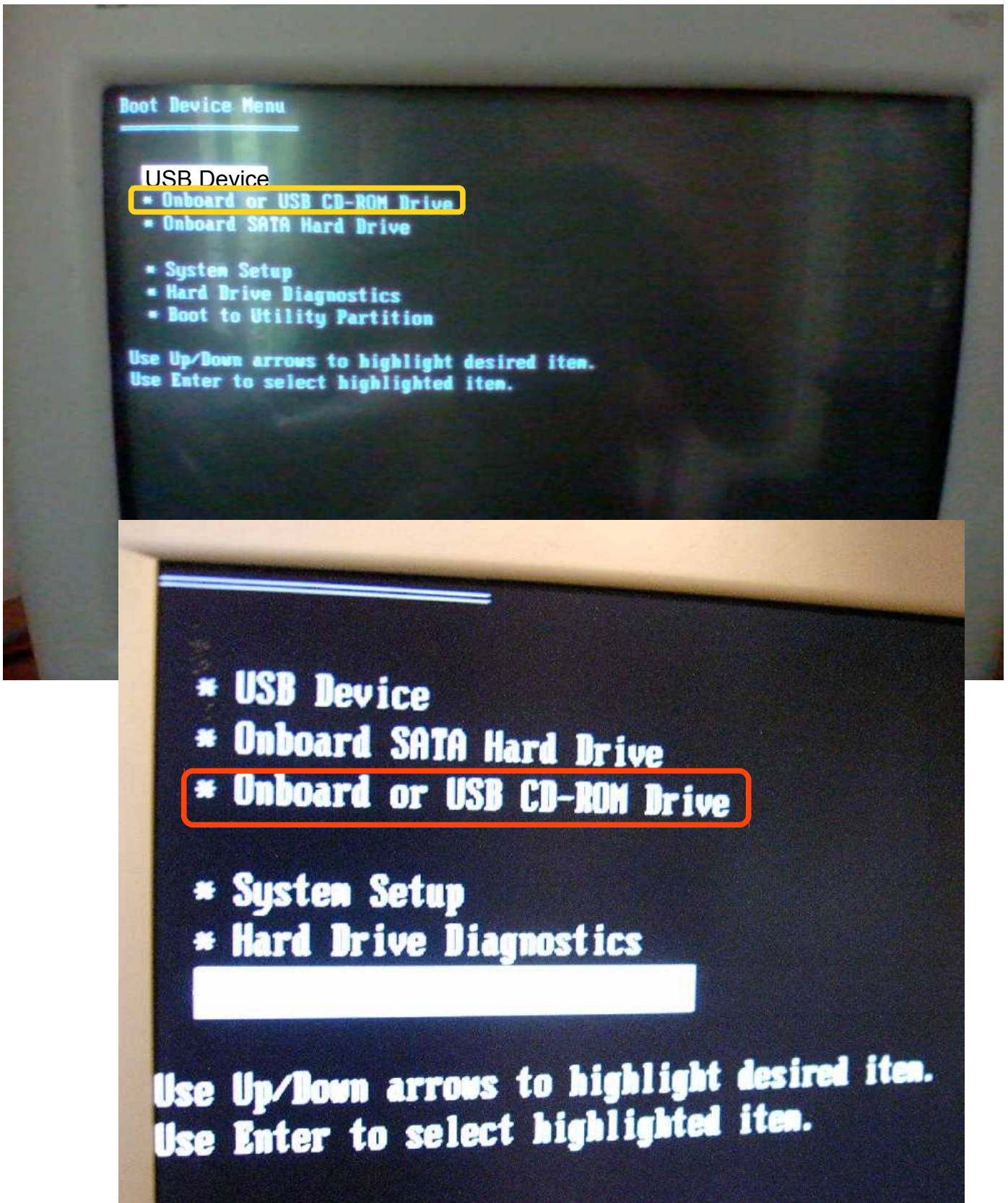
1.1.2 Sessió en viu o LIVE SESSION

Inserir el DVD d'instal·lació en el lector òptic de l'ordinador i reiniciar l'ordinador, accedint a les opcions d'arrencada (boot options), per seleccionar l'arrencada des del DVD d'instal·lació.

Quan un ordinador inicia, normalment s'accedeix a les opcions d'arrencada (Boot Device Menu) pitjant F12.

F2 Setup
F12 Boot Menu





Seleccionem l'arrencada des de el DVD (lector òptic).

La pantalla mostra que Ubuntu s'està carregant.

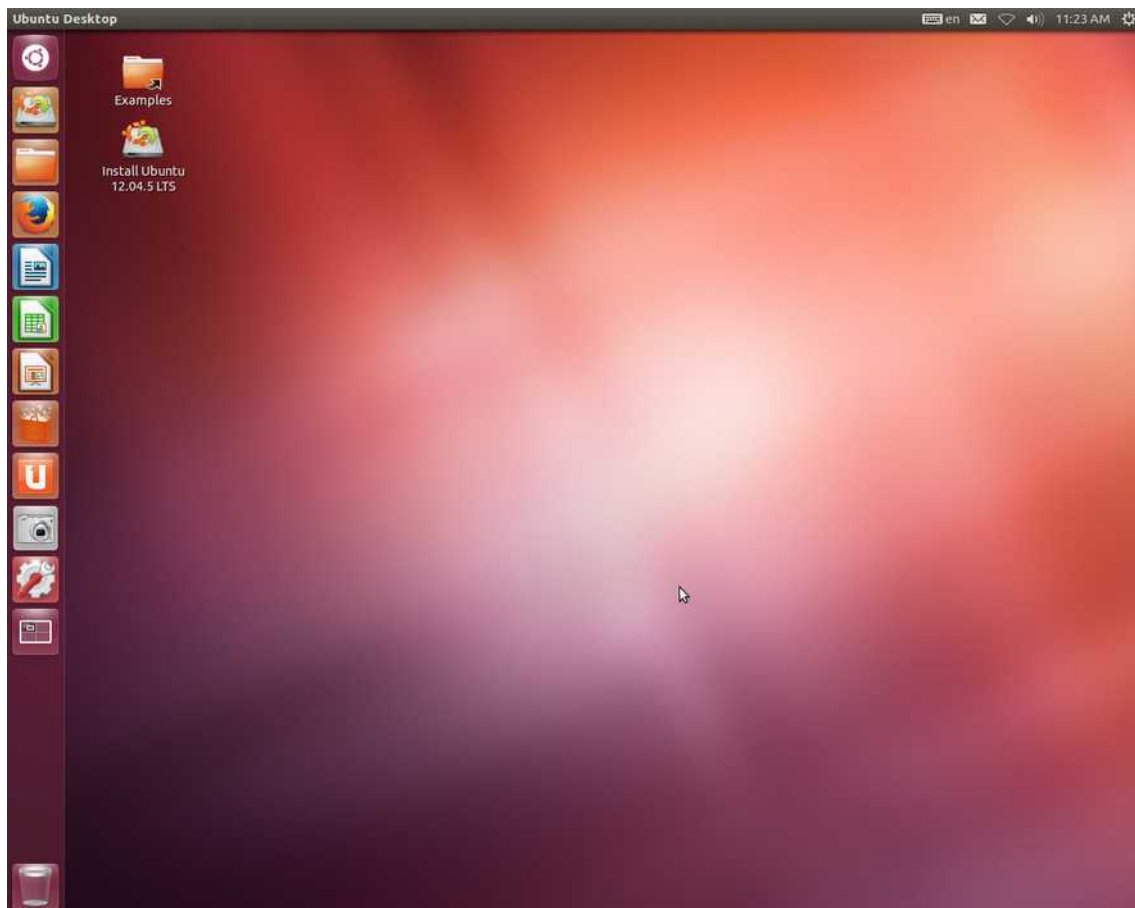


La finestra **Welcome** ens permet iniciar una LIVE SESSION o sessió en viu, que significa fer funcionar Ubuntu des del disc d'instal·lació. En aquest cas, per iniciar una sessió en viu, hem de seleccionar **Try Ubuntu**.

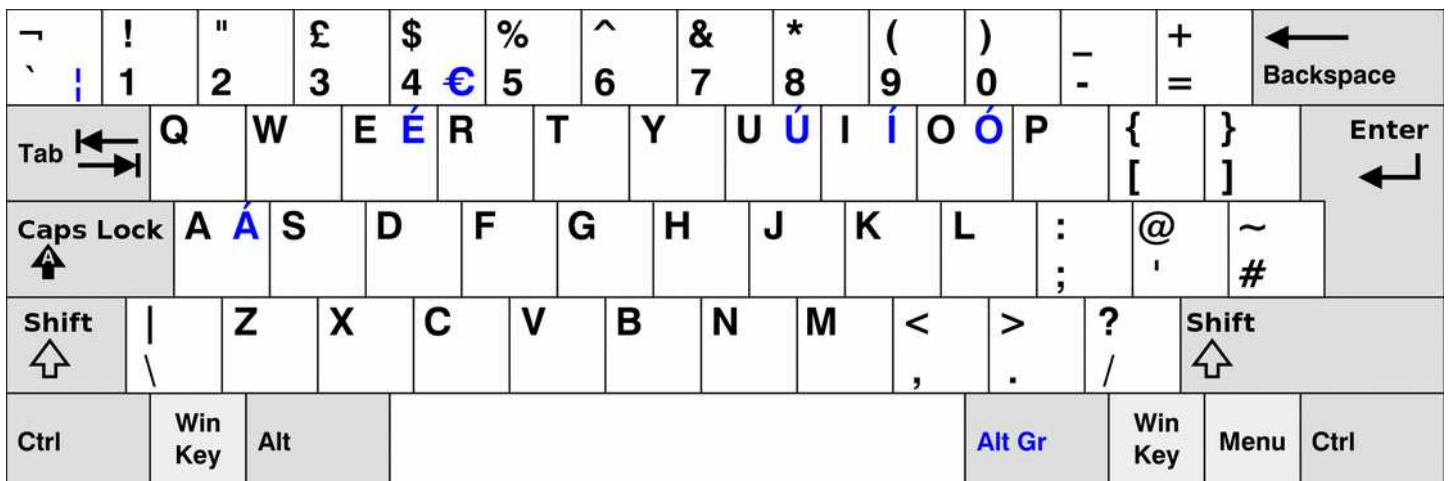


Una sessió en viu té l'avantatge que no es modifica res al disc dur del nostre ordinador, ni a cap altre mitja d'emmagatzematge que estigui connectat a l'ordinador. Però també té desavantatges, no podem escriure damunt el disc d'instal·lació i, per això, no podem modificar el programari d'Ubuntu. A més, la reacció del sistema operatiu i de les aplicacions és molt lenta, ja que accedir al lector òptic és molt més lent que accedir al disc dur intern.

La finestra principal de la sessió en viu del la versió d'Ubuntu 12.04 LTS, és la següent.



Si utilitzes Ubuntu des d'una sessió en viu, es possible que la configuració del teu teclat no coincideixi amb la predeterminada d'Ubuntu, que és la d'un teclat anglès. La següent imatge mostra un teclat anglès. Et facilitarà trobar caràcters en el teu teclat.



Una altra opció per escriure caràcters que no trobes és obrir un document de Writer i inserir els caràcters que no trobes com a caràcters especials. El text del document Writer el pots copiar i enganxar a l'aplicació que vulguis.

Per cert, per enganxar a la línia de comando del Terminal, hauràs de pitjar les tecles Ctrl, Shift, V simultàniament.

Exercici 1.1.2-1

Crear disc d'inici Ubuntu.

Data d'entrega dimecres 13/01/15.

Exercici 1.1.2-2

Enviar captura de pantalla amb versió del sistema operatiu a

pposada@iessonpacs.cat

Informació de com s'accedeix a la [versió del sistema operatiu al blog](#).

Data d'entrega dimecres 13/01/15.

1.1.3 Instal·lació d'Ubuntu en un disc dur extern

El disc d'instal·lació es pot utilitzar per instal·lar Ubuntu al costat del nostre sistema operatiu habitual, que sol ser Windows. En aquest cas, però, l'utilitzarem per instal·lar Ubuntu en un disc dur extern.

Hem d'assegurar-nos, que el disc dur extern no contengui informació valuosa, ja que serà esborrada durant la instal·lació d'Ubuntu.

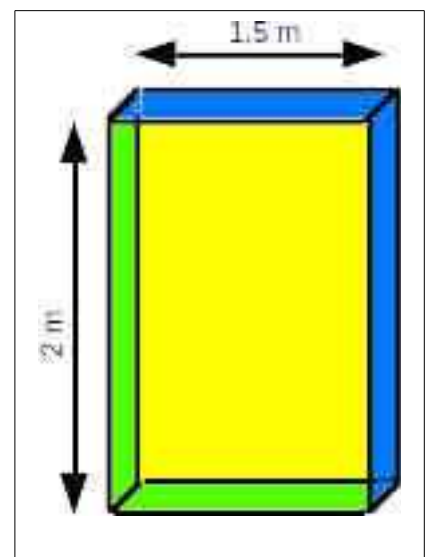
Començarem formatant el disc dur extern amb el programa *Gparted*.

Donar format a un disc, significa organitzar la seva estructura d'emmagatzematge, perquè el sistema operatiu pugui escriure i llegir informació del disc.

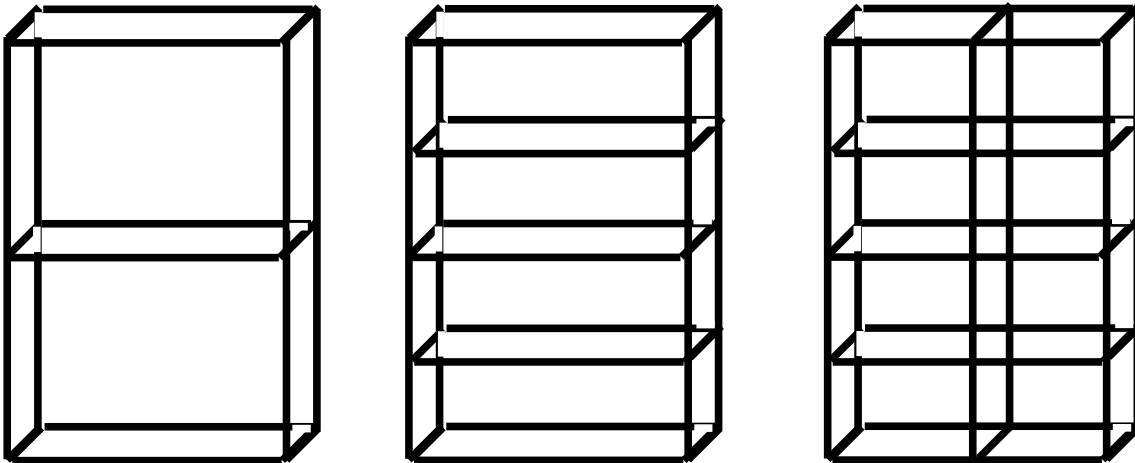
Podem comparar un mitja emmagatzematge amb una prestatgeria, ja que una prestatgeria també s'utilitza per emmagatzemar coses.

Suposem que tenim una prestatgeria sense prestatges, de 2 m d'alçada, 1,5 m d'amplada i 0,3 m de fons. La capacitat total d'emmagatzematge la prestatgeria és de $2\text{m} \times 1,5\text{m} \times 0,3\text{m} = 0,9\text{m}^3$

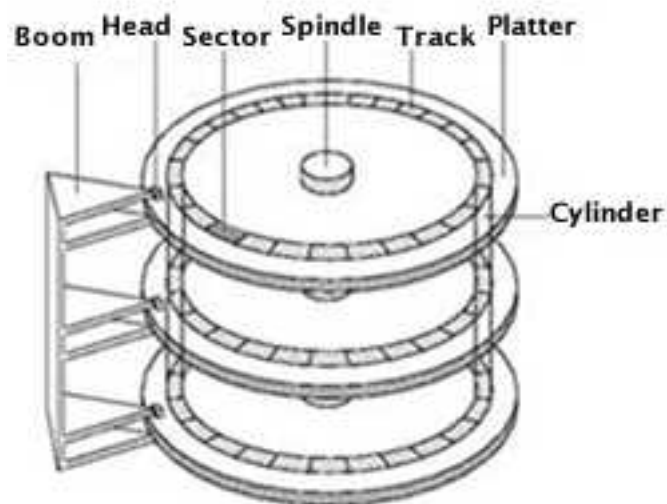
Això equivaldria a un disc dur sense formatar. La capacitat d'un disc dur no es mesura en metres cúbics, sinó en GB.

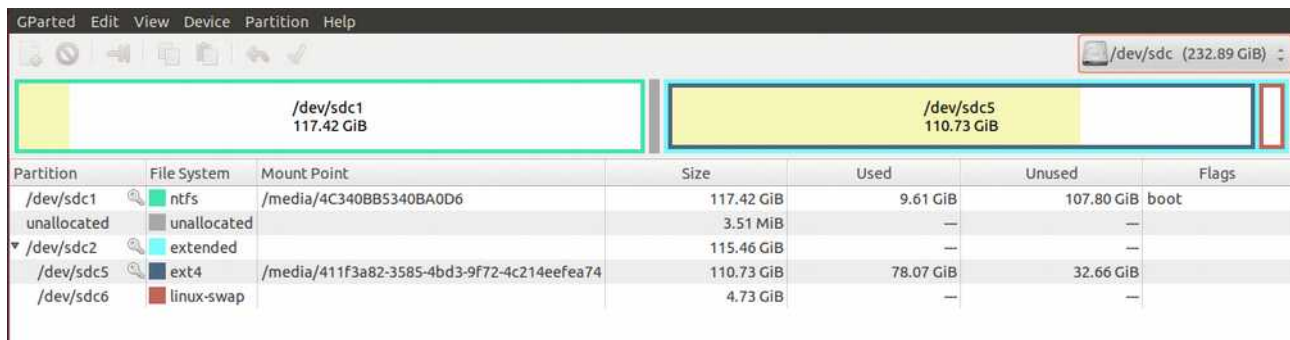


Com la prestatgeria no disposa de prestatges, encara no hi podem guardar res. Per poder guardar-hi coses, hem de decidir quants prestatges hi muntem i si feim compartiments en cada prestatge. Això equivaldria a formatar el disc, és a dir, organitzar la seva estructura d'emmagatzematge.



Com es veu, hi ha infinitat de possibilitats per organitzar l'estructura d'emmagatzematge. Un disc dur, es formata en sectors i cilindres.





Partition	File System	Mount Point	Size	Used	Unused	Flags
/dev/sdc1	ntfs	/media/4C340BB5340BA0D6	117.42 GiB	9.61 GiB	107.80 GiB	boot
unallocated	unallocated		3.51 MiB	—	—	
▼ /dev/sdc2	extended		115.46 GiB	—	—	
/dev/sdc5	ext4	/media/411f3a82-3585-4bd3-9f72-4c214eefea74	110.73 GiB	78.07 GiB	32.66 GiB	
/dev/sdc6	linux-swap		4.73 GiB	—	—	

En aquest exemple, **GParted** identifica el disc a formatar com a **/dev/sdb** i mostra la mida (233 GB).

El disc disposa de 3 particions, que no són altra cosa que 3 subdivisions o apartats.

La subdivisió primera **/dev/sdb1** és del tipus **ntfs**, amb una mida de 117,42 GiB. El sistema d'arxiu **ntfs**, és el que típicament utilitza el sistema operatiu Windows.

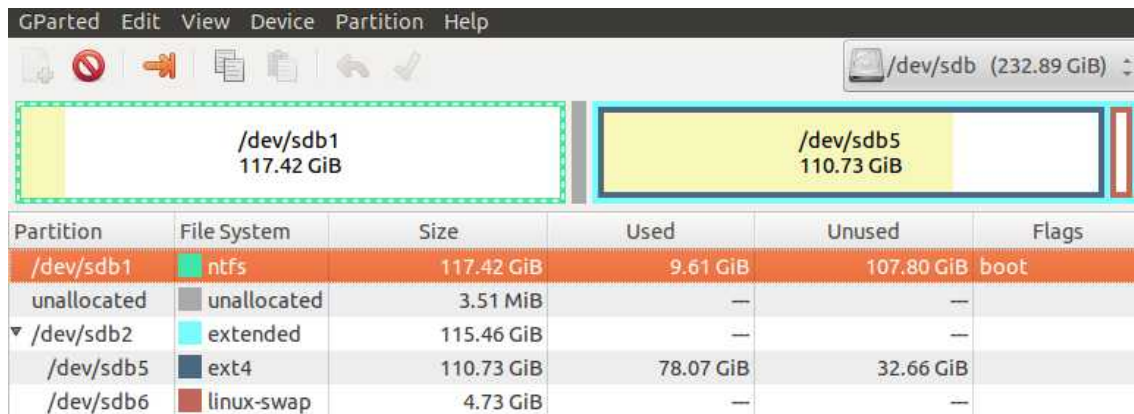
La segona subdivisió **unallocated**, està sense configuració de sistema d'arxiu. La seva mida és de 3,51 MiB.

La tercera subdivisió, **/dev/sdb2**, és del tipus **extended** (ext4), que és el sistema d'arxiu que normalment s'utilitza amb Linux. Aquesta subdivisió consta de 2 apartats, **/dev/sdb5** (110,73 GiB) i **/dev/sdb6** (4,73 GiB).

L'apartat **/dev/sdb6**, anomenat SWAP o memòria d'intercanvi, serveix com a suport a la memòria RAM, situada en la placa base de l'ordinador.

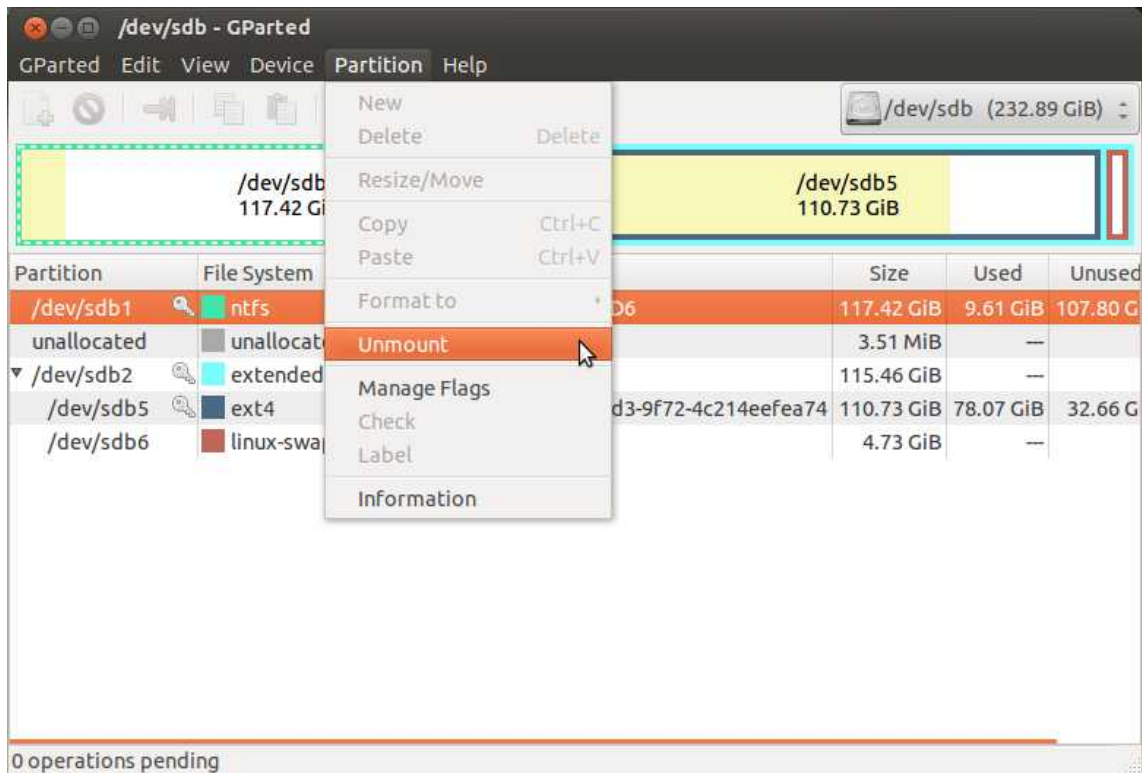
La memòria RAM és necessària perquè les aplicacions que utilitzem funcionin de manera ràpida, sense alentir-se perceptiblement. La memòria d'intercanvi, situada al disc dur, és utilitzada quan tenim moltes aplicacions obertes. L'accés a la SWAP és molt més lent que l'accés a la RAM, però serveix per mantenir funcionant aplicacions que no estan treballant i alliberar així a la RAM, evitant la seva sobrecàrrega i que l'ordinador es pengi.

Per esborrar la primera partició **/dev/sdb1**, es selecciona.

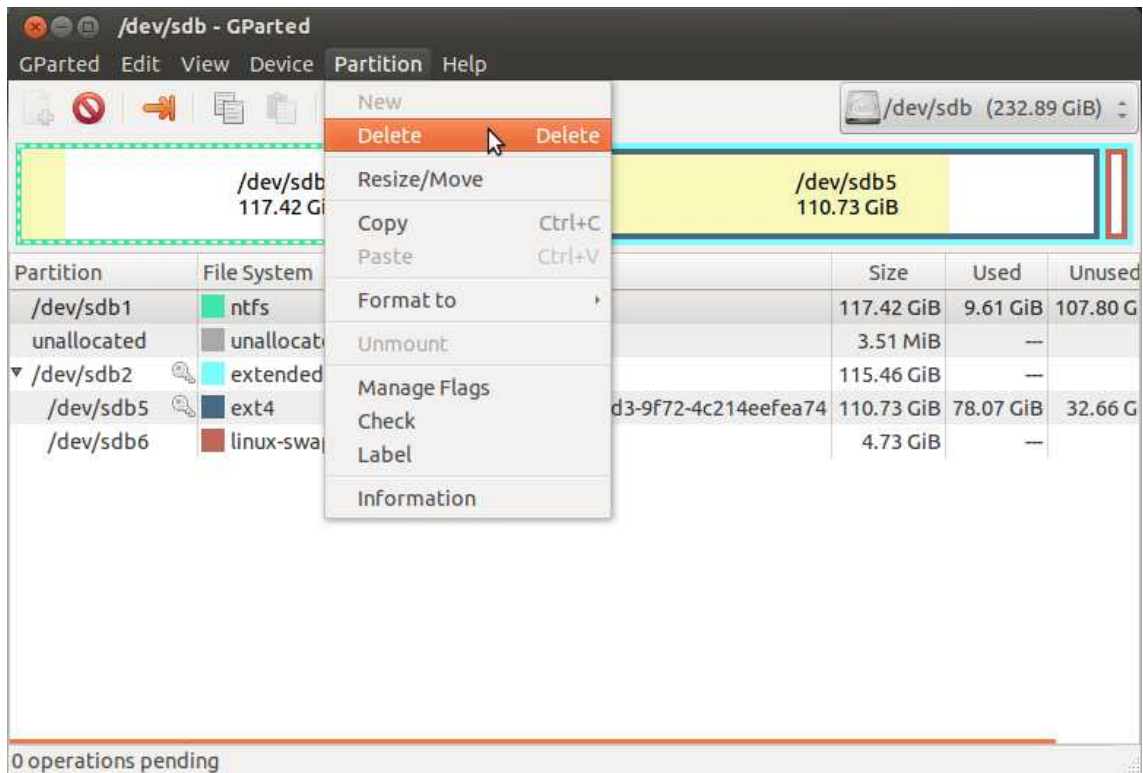


Una partició d'un dispositiu d'emmagatzematge ha d'estar muntada per que el sistema operatiu hi pugui llegir o escriure. Només es pot modificar (esborrar, ampliar, reduir) una partició, estant desmuntada.

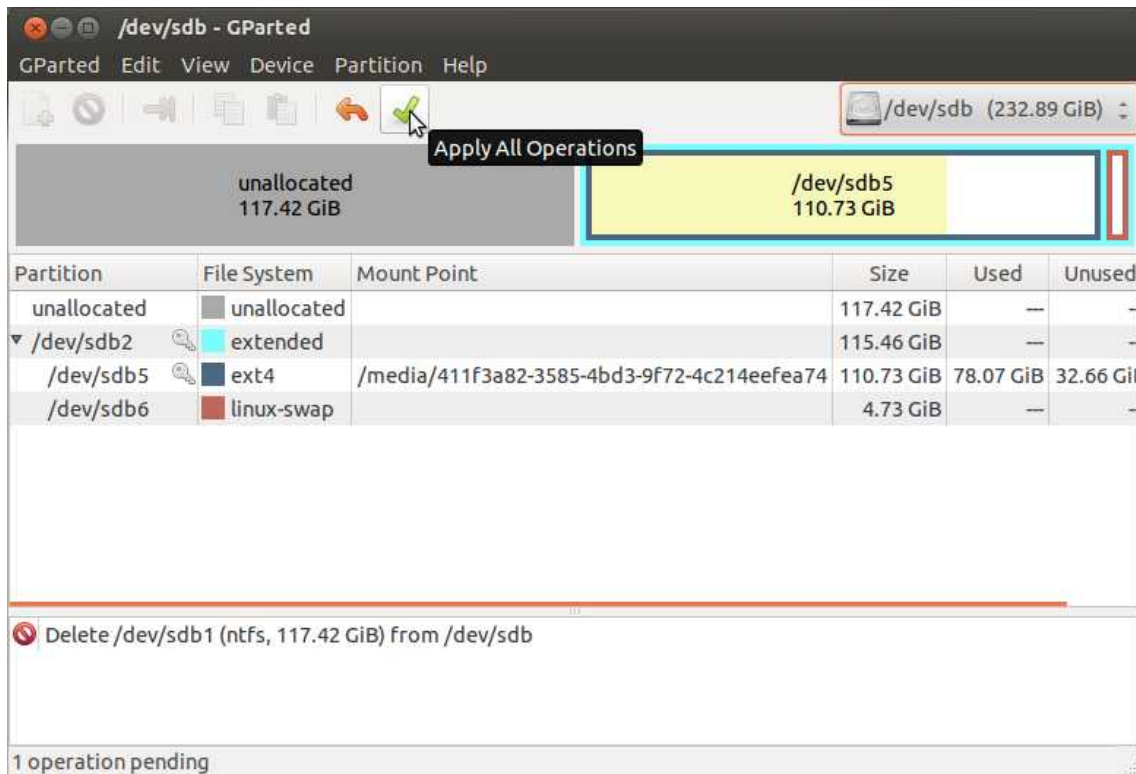
Ara es desmunta */dev/sdb1*, per poder esborrar-la.



I es selecciona esborrar.



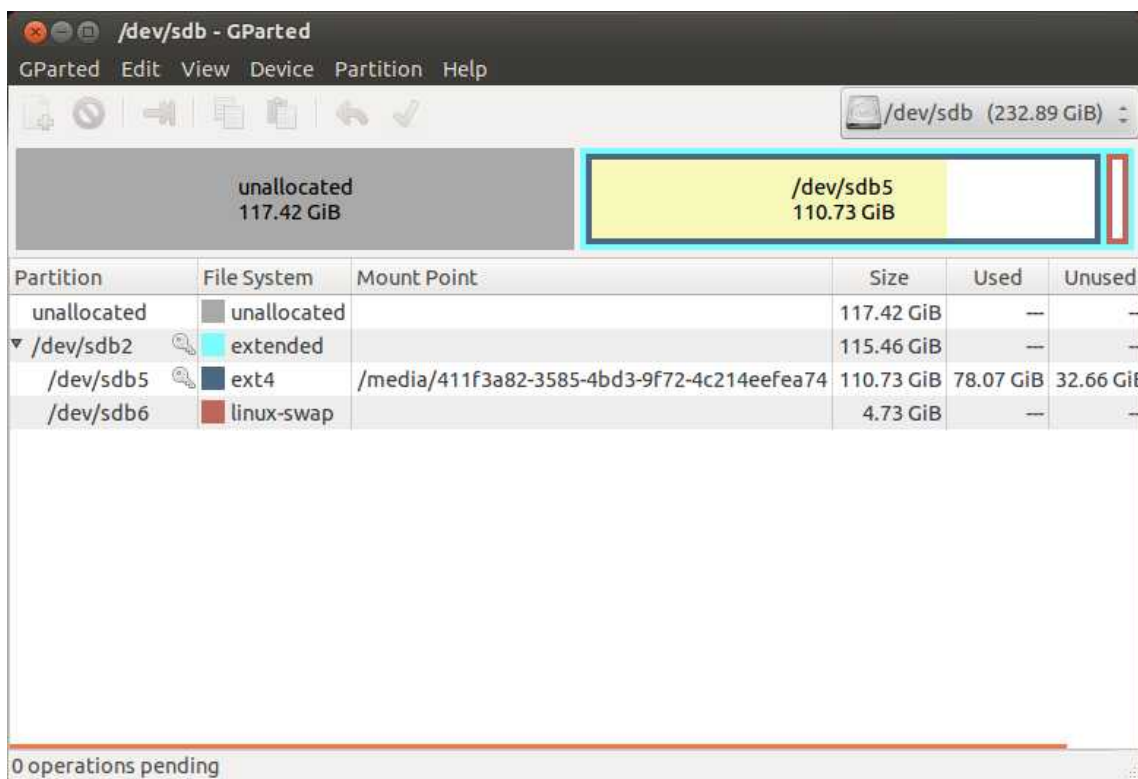
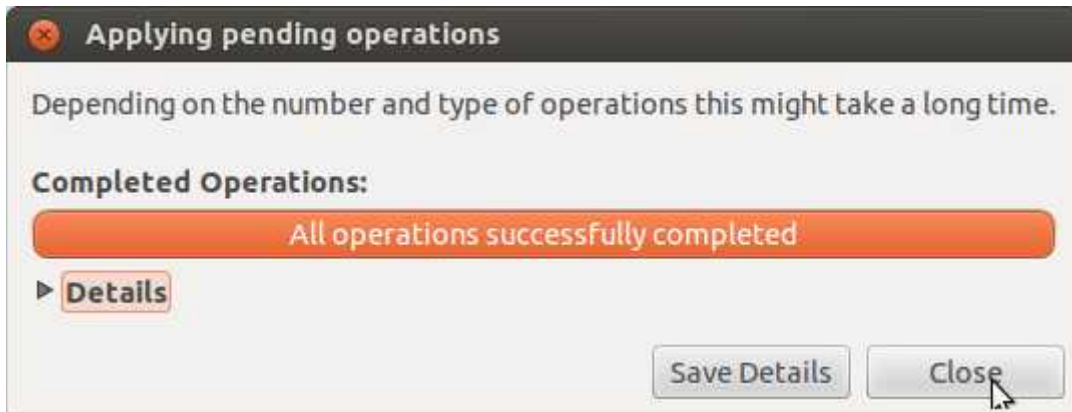
Al peu de la finestra s'informa que hi ha una operació pendent d'aplicar. Es pitja la icona aplicar per esborrar la partició.



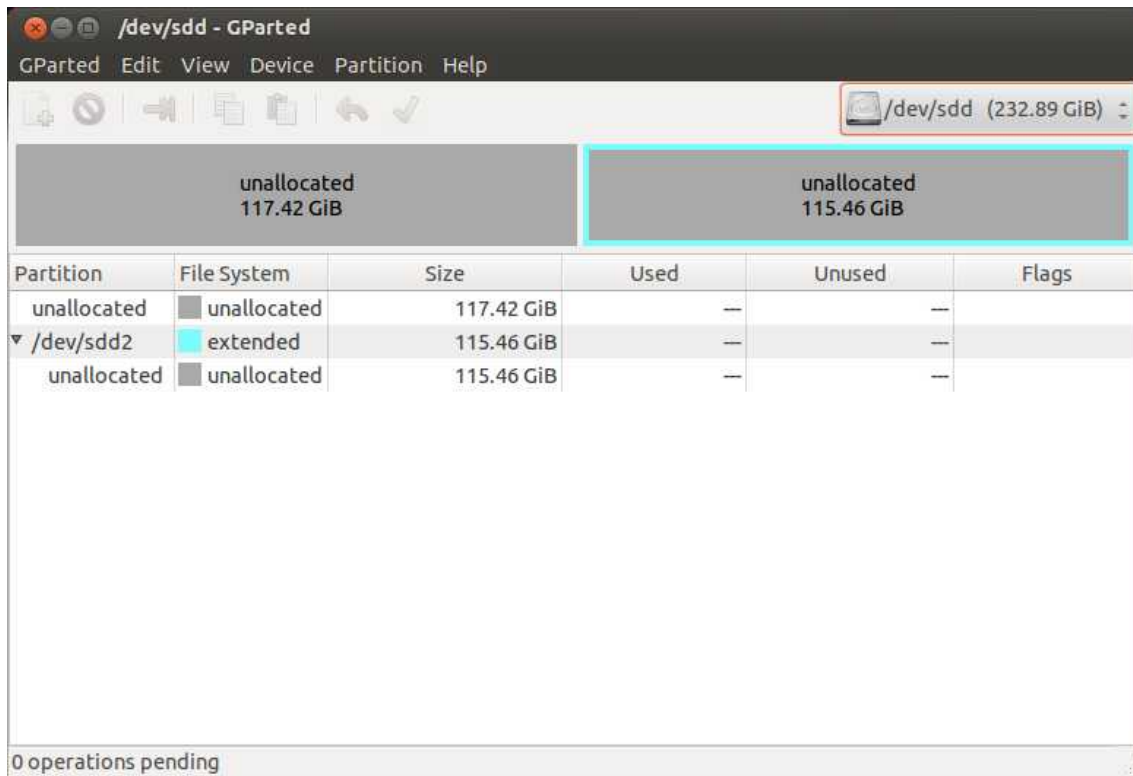
S'obre una finestra, advertint que es faran canvis al disc seleccionat i demanant si es volen aplicar aquests canvis.



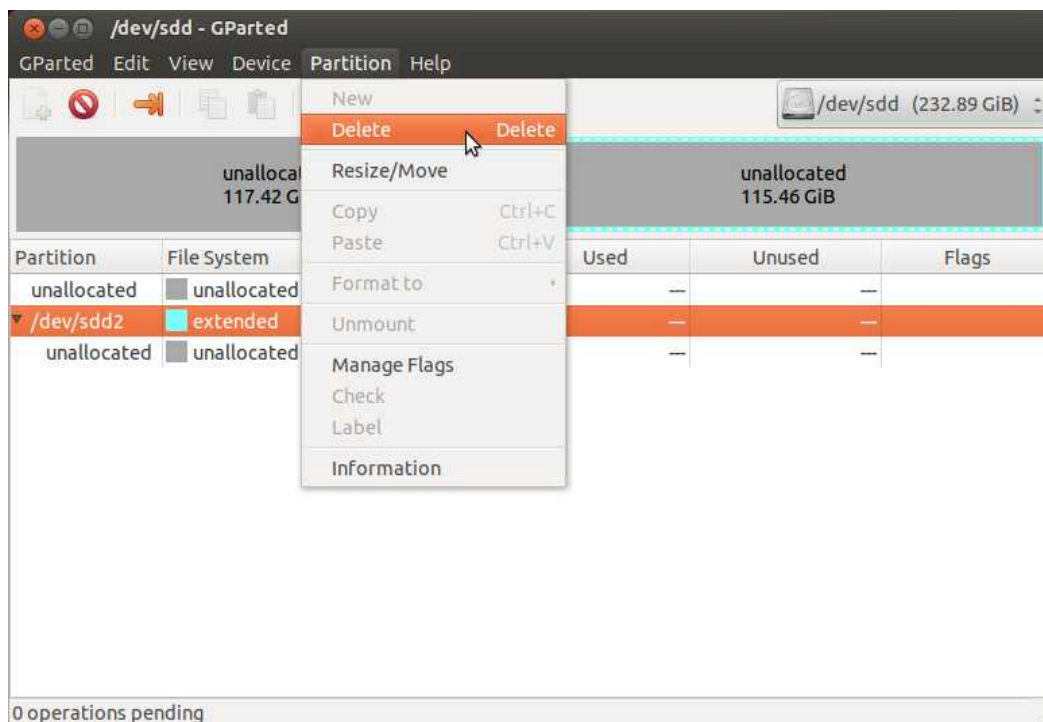
Es pitja aplicar



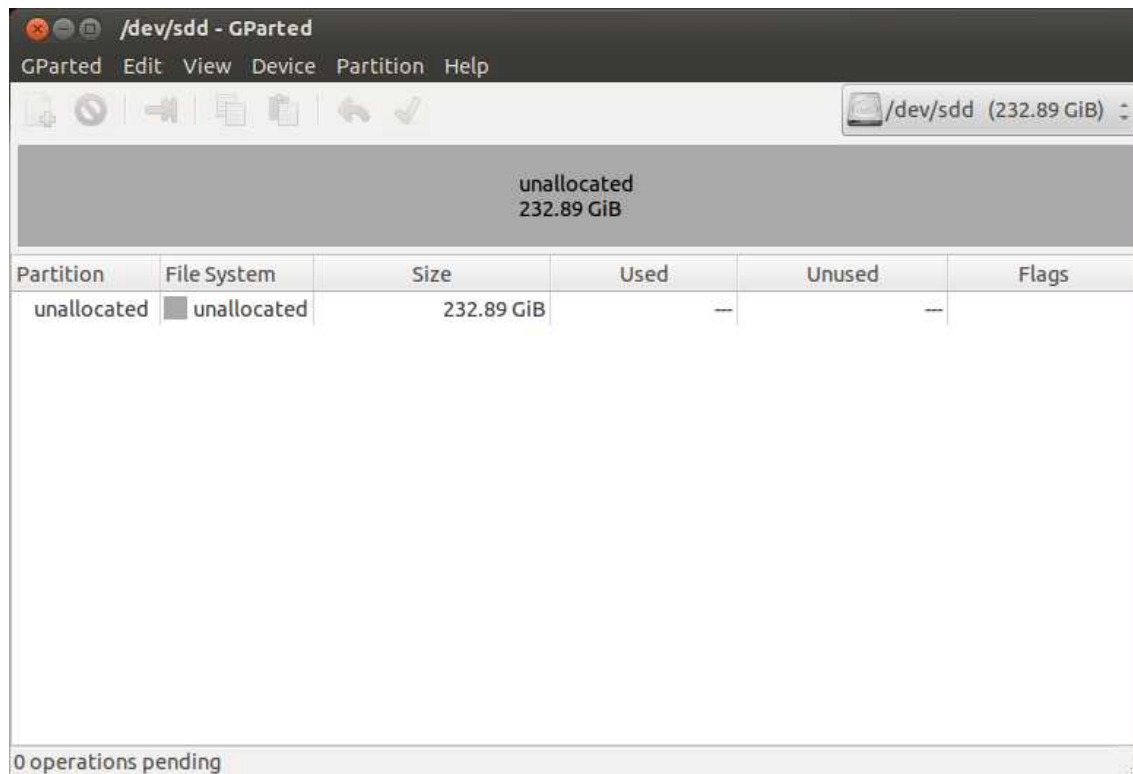
Per esborrar les particions */dev/sdb5* i */dev/sdb6*, es procedeix de la mateixa manera que acabem de veure.



Ara esborrem */dev/sdb2*.

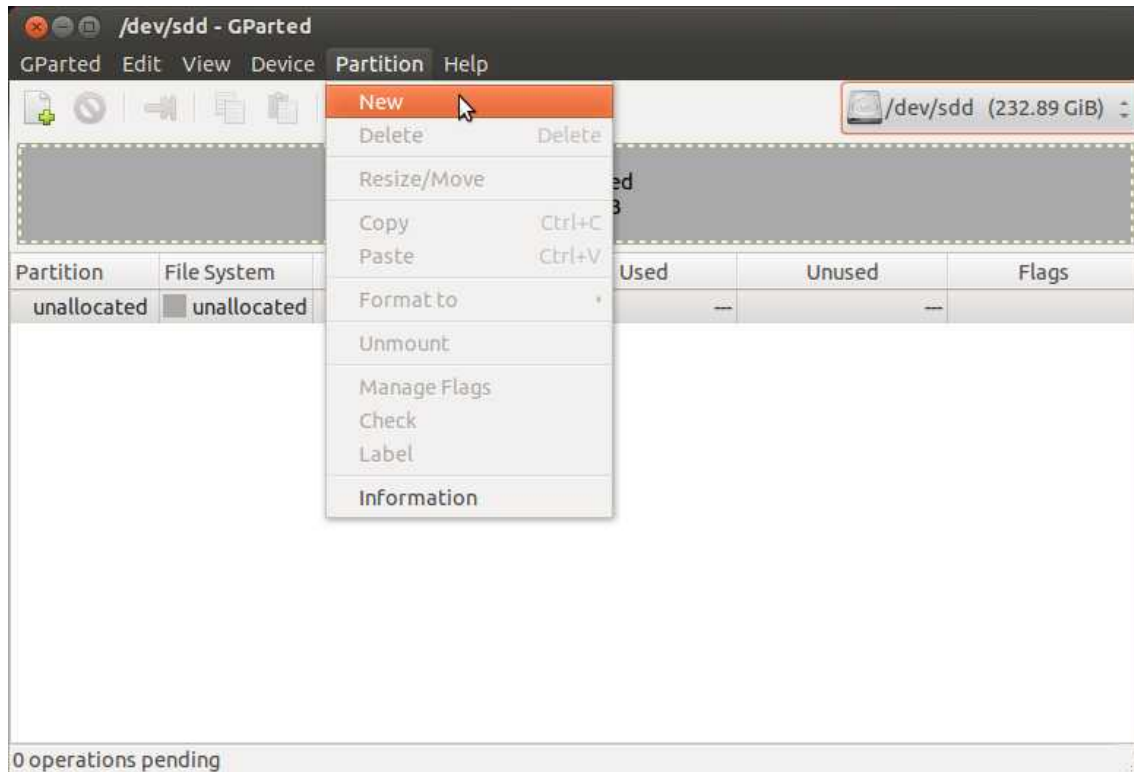


La imatge mostra un disc sense sistemes d'arxiu. Tal com està el disc en aquests moments, el sistema operatiu no hi pot accedir. Només es pot accedir al disc amb programes preparats per formatar discs, com Gparted.

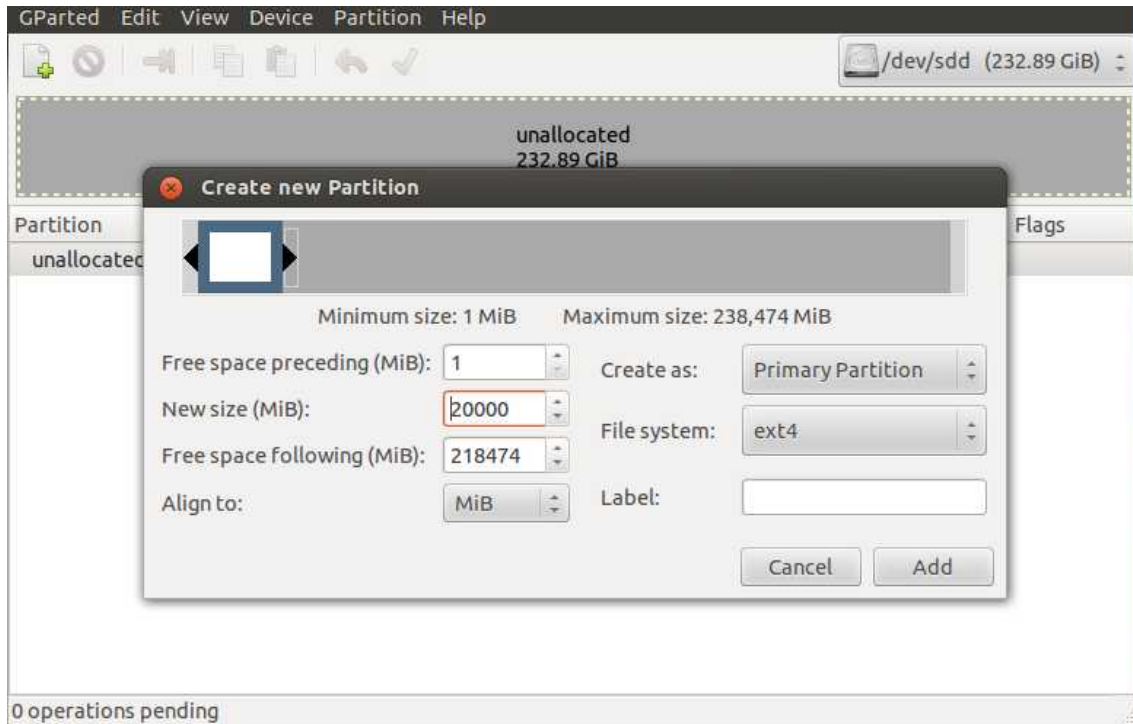


Tornant al símil amb la prestatgeria, aquesta seria la prestatgeria sense prestatges. Durant les següents passes, muntarem els prestatges per poder desar coses a la prestatgeria.

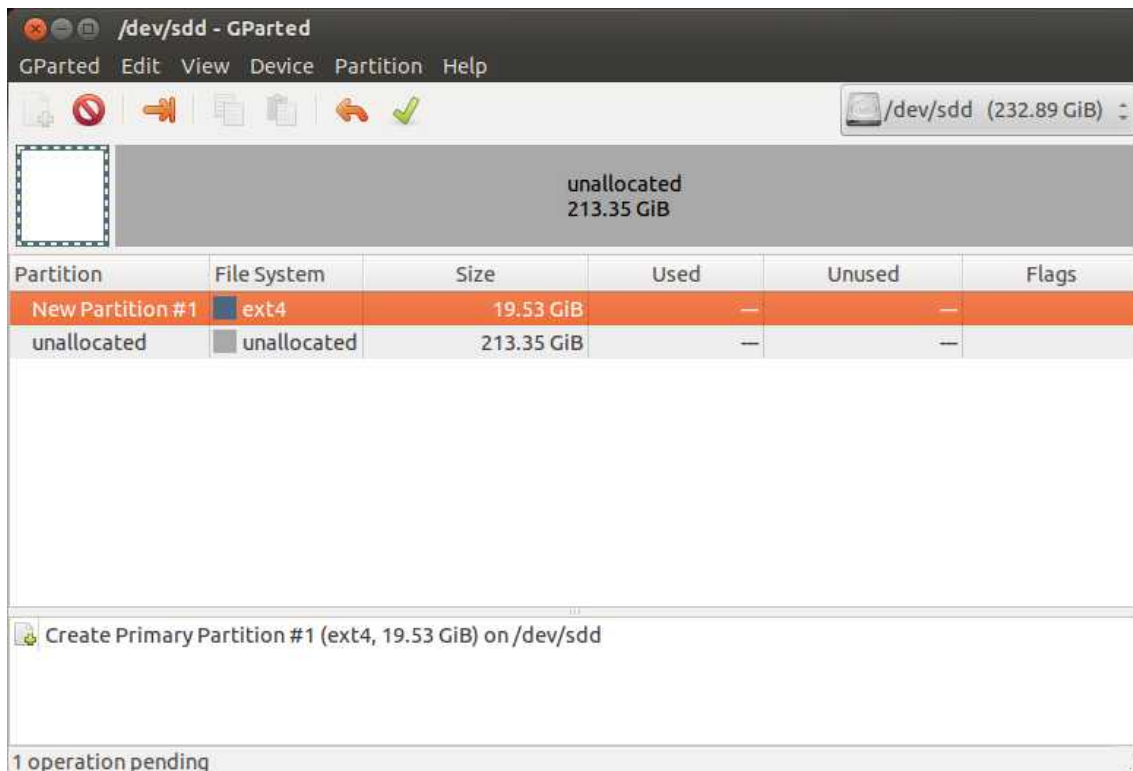
Per instal·lar Ubuntu, necessitem una partició, que triem de 20 GiB. Amb aquesta mida hi ha espai més que suficient per albergar el sistema operatiu i les aplicacions que, amb el temps, s'aniran afegint.



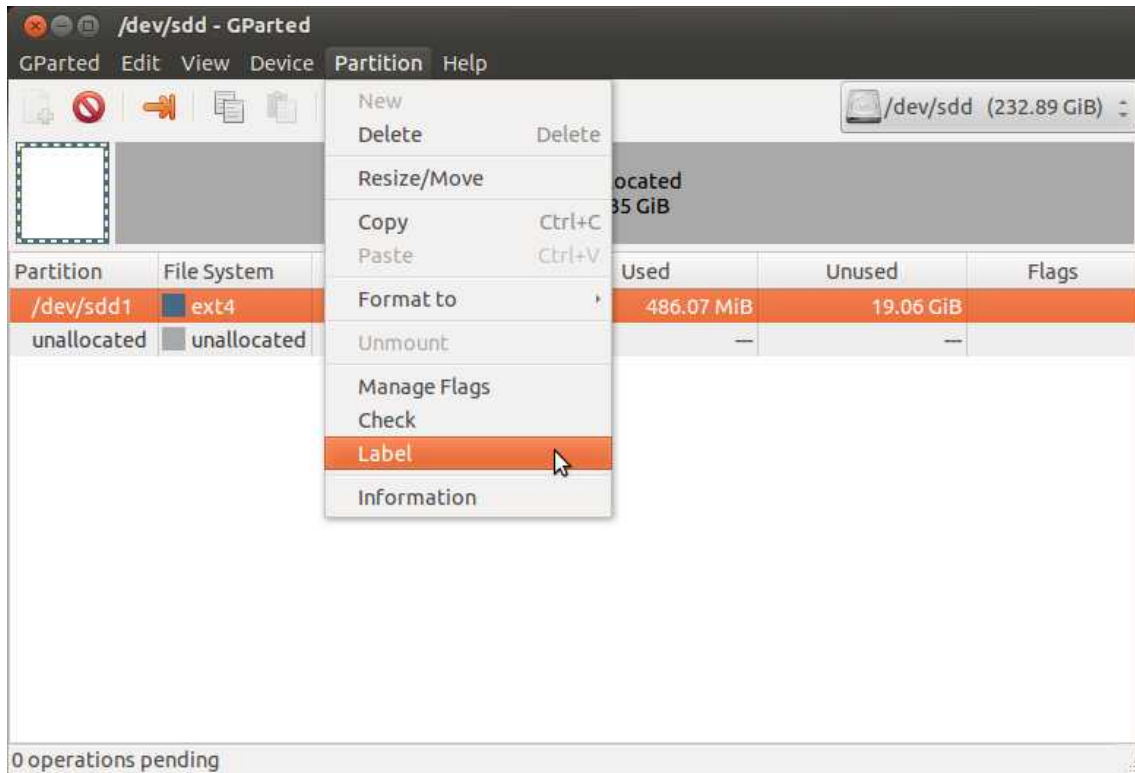
Com es mostra, es crearà la nova partició amb una mida de 2000 MiB. El tipus de sistema d'arxiu serà **ext4**, que és adequat per a Linux.



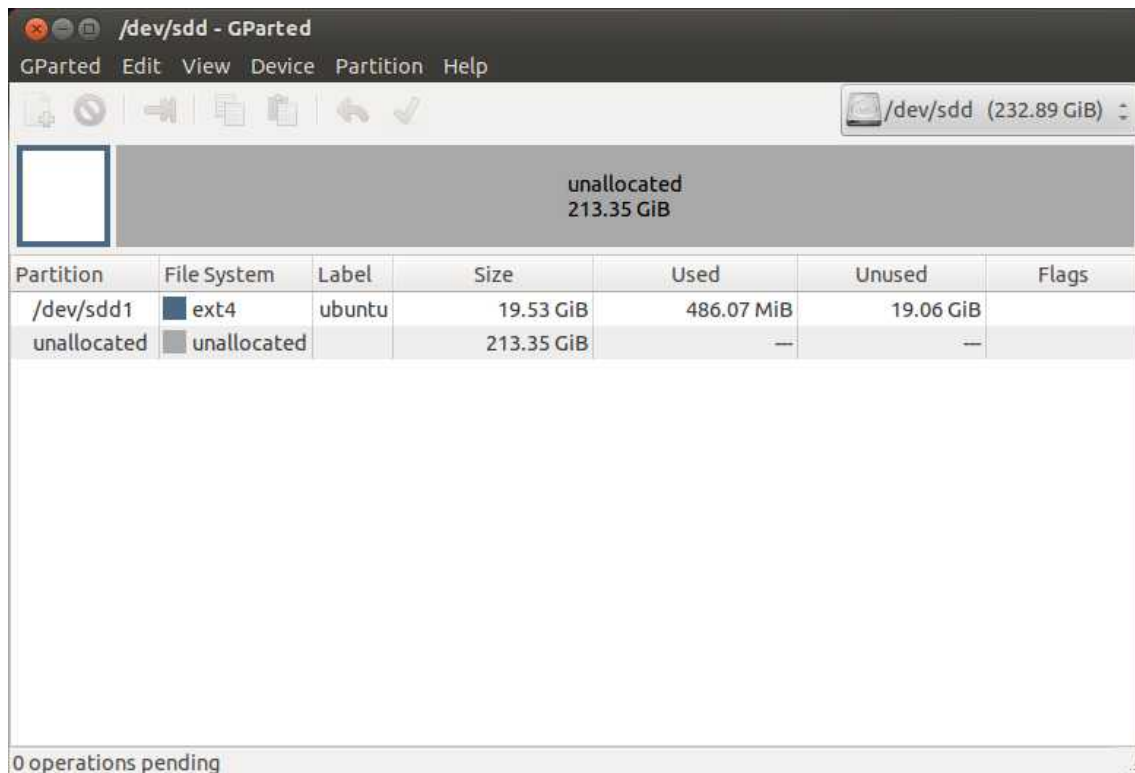
Ara **GParted** està preparat per crear la nova partició. Falta pitjar la icona aplicar i confirmar a les finestres que apareixeran, indicant que volem aplicar els canvis.



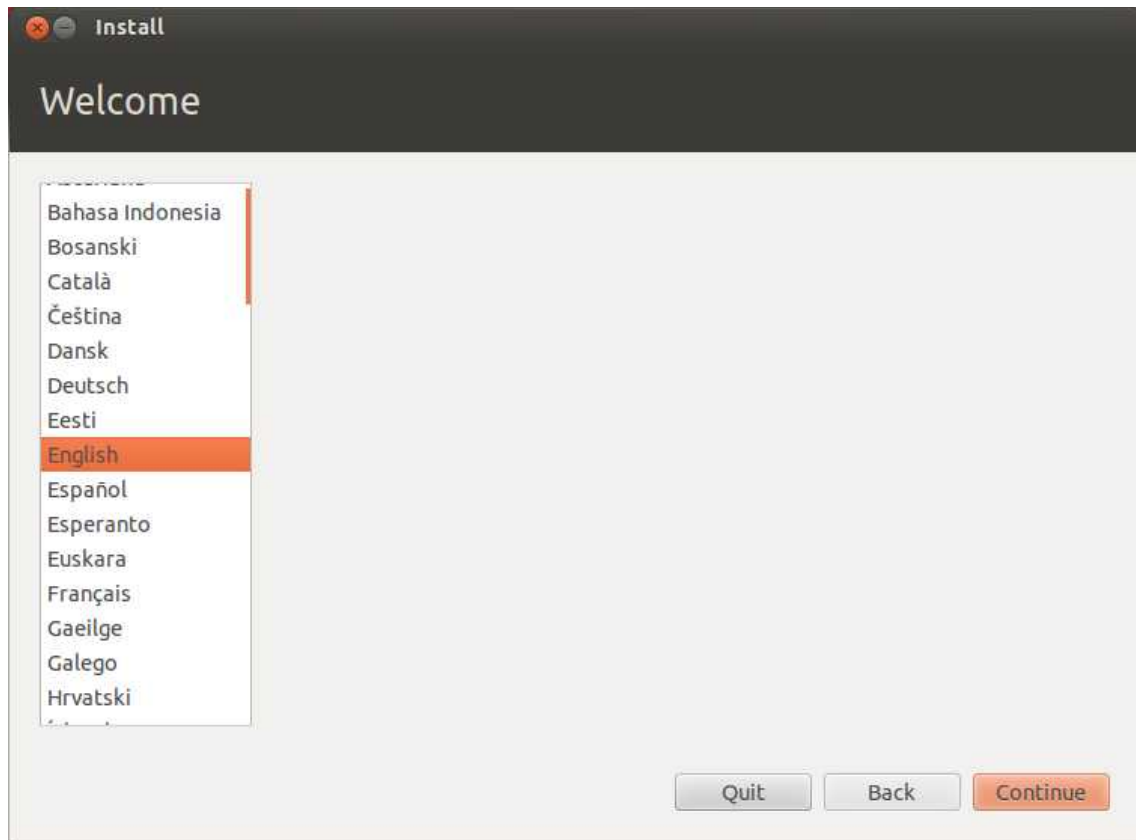
Identificarem la nova partició amb una etiqueta (lable).



Així, el disc està preparat per a la instal·lació d'Ubuntu.



Pitgem la icona *Instal Ubuntu 12.04.5 LTS* de l'escriptori.



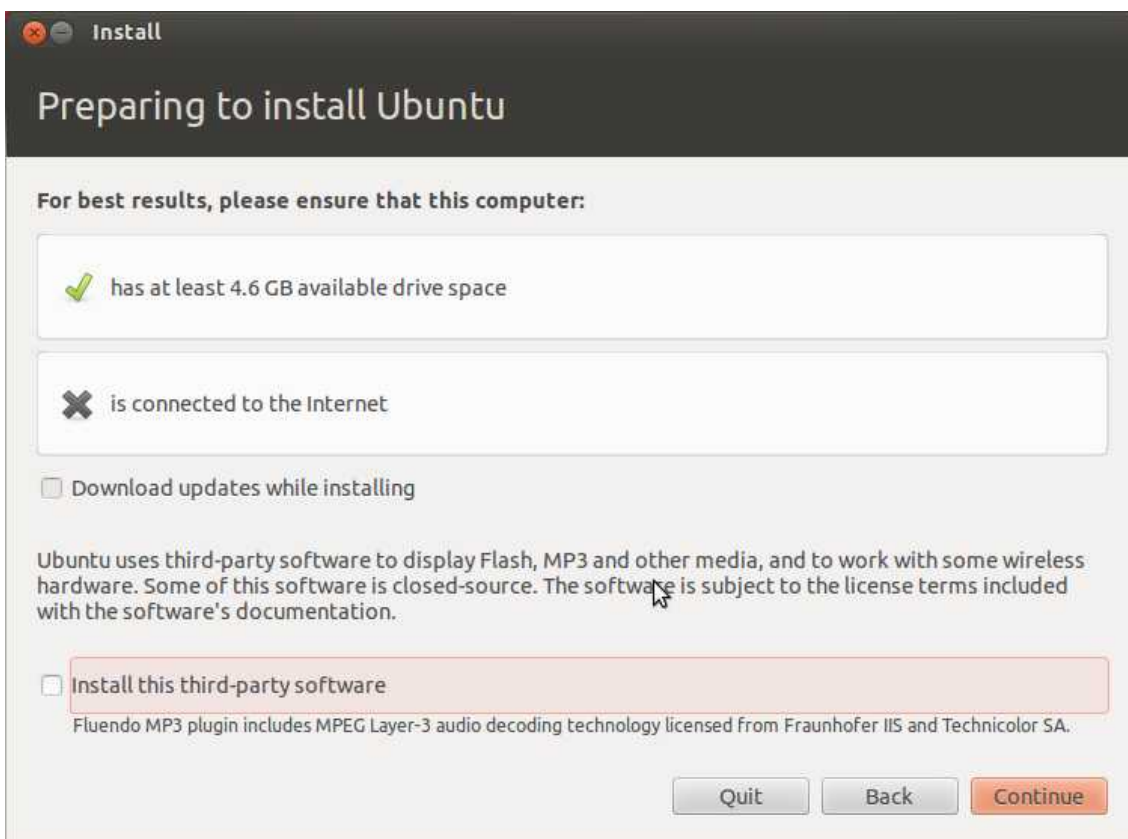
Comença el procés d'instal·lació en l'idioma que hàgim seleccionat a la finestra **Welcome**.

A continuació, apareix la finestra **Install**.

Si l'equip està connectat a internet, és convenient activar les opcions

- Download updates while installing
- Install this third party software

Fetes les seleccions, pitgem **Continue**.

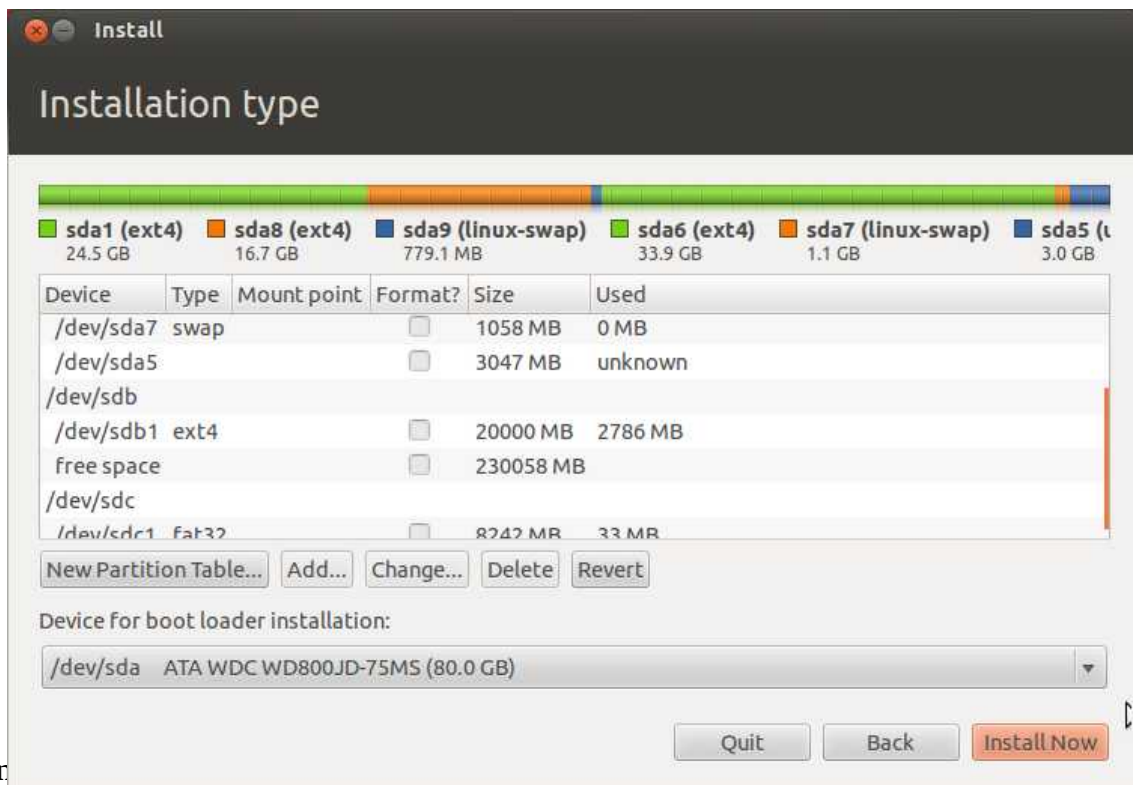
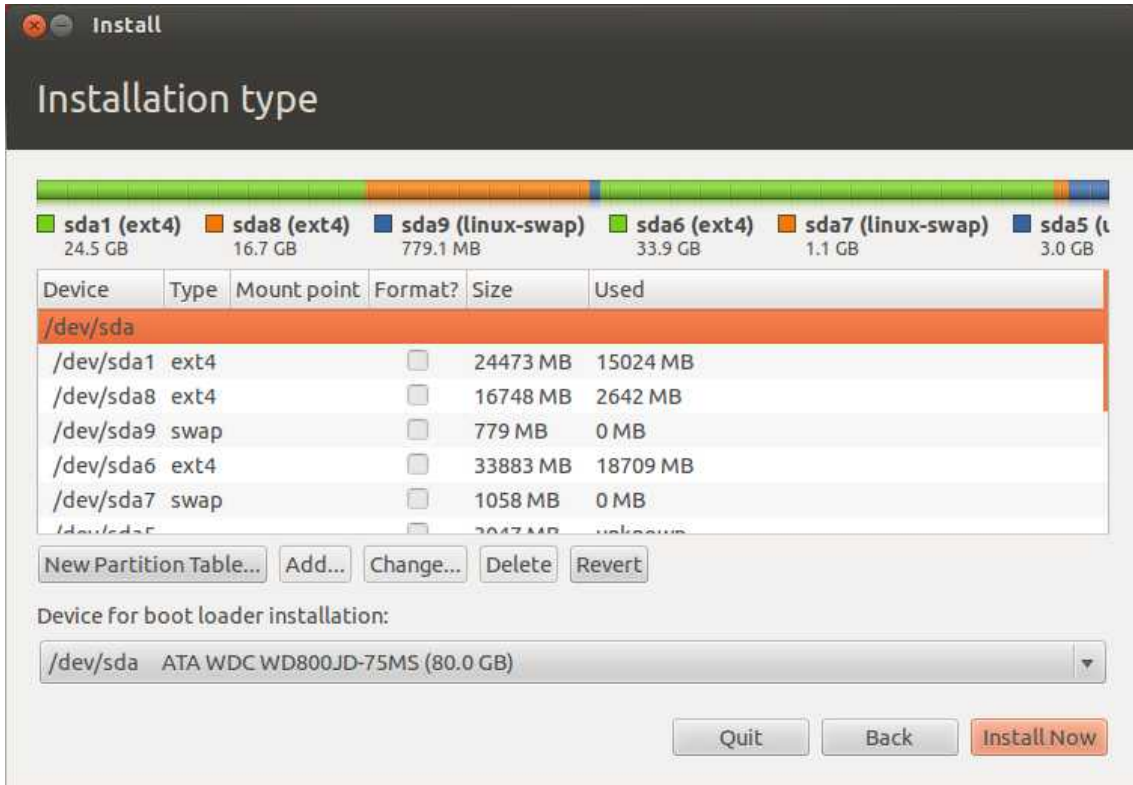


La finestra **Install**, permet triar entre la instal·lació d'Ubuntu en el disc dur intern, al costat del nostre sistema operatiu habitual (Install Ubuntu alongside them), esborrar el disc dur intern i instal·lar només Ubuntu (Erase disk and install Ubuntu) o fer una altra cosa (Something else).

Serà la darrera opció (Something else), la que triarem per instal·lar Ubuntu en un disc dur extern.



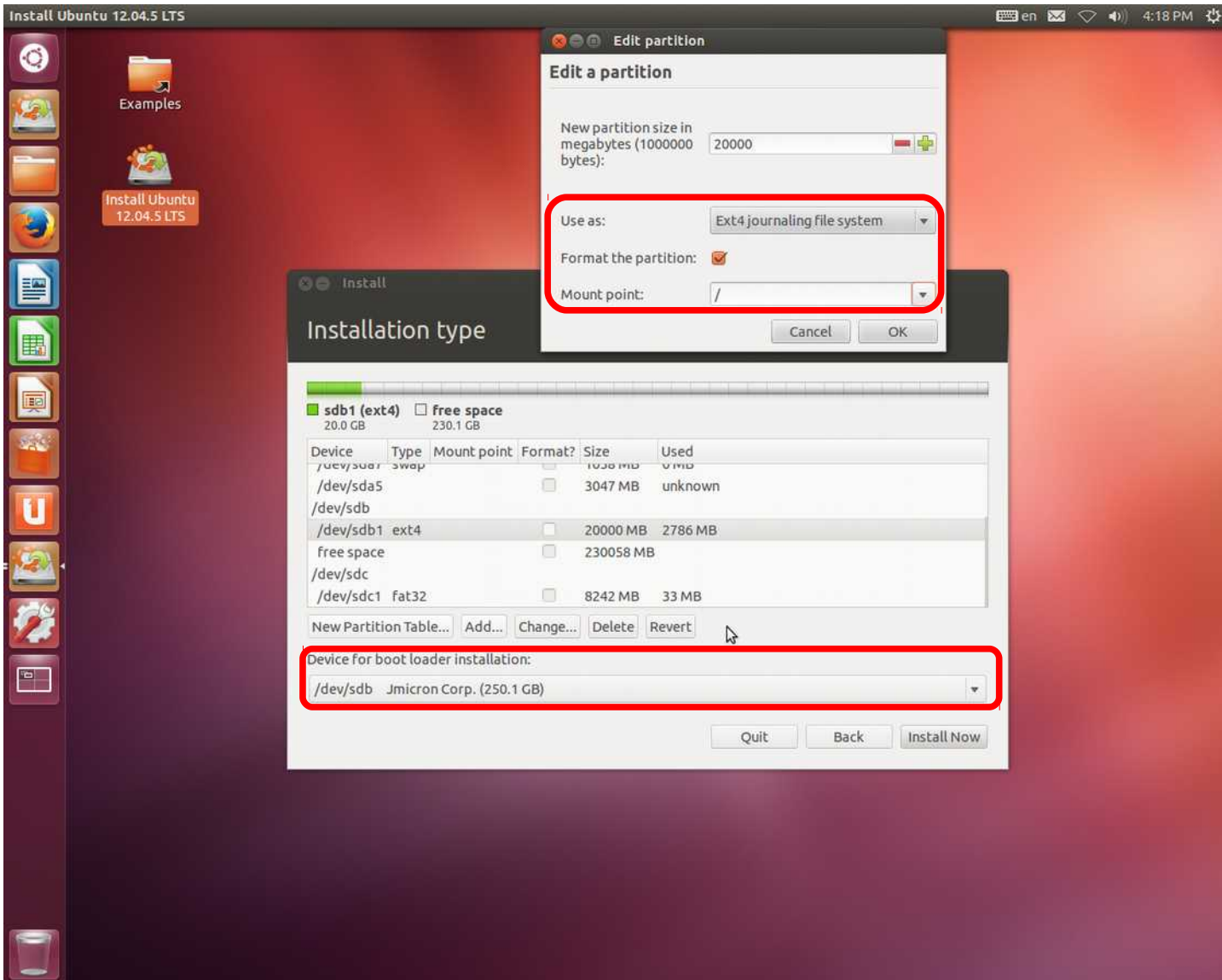
La finestra **Installation type** mostra els disc durs detectats per Ubuntu.



En les imatges anteriors, es poden distingir 3 dispositius d'emmagatzematge, **sda**, **sdb** i **sdc**.

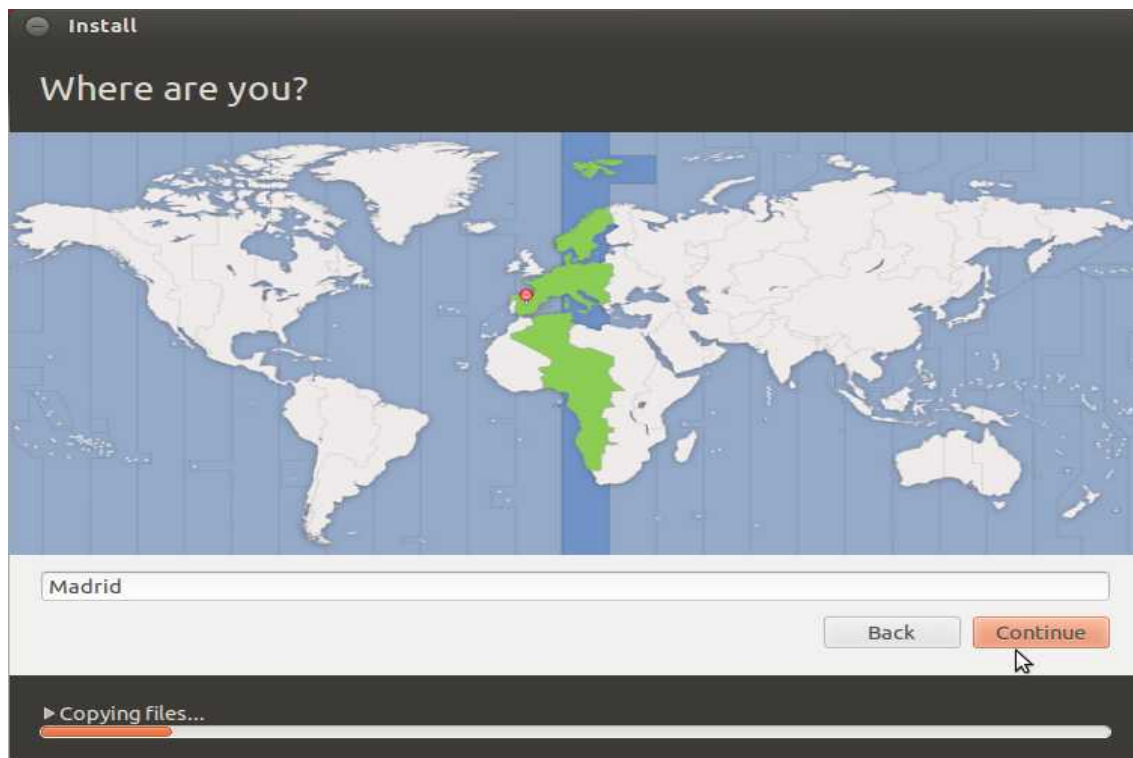
- **sda** és el disc dur intern, que està subdividit en les particions sda1 a sda9.
- **sdb** és el disc dur extern, que només disposa d'una única partició, **sdb1**, de 20 GB.
- **sdc** és una memòria USB de 8 GB.

Ara hem de configurar la partició sdb1 del disc dur extern, perquè es pugui instal·lar Ubuntu. Fent doble clic damunt sdb1, apareix la finestra Edit partition, on triarem el tipus de sistema d'arxiu (ext4), activarem la formatació i seleccionarem el punt de muntatge com a root (/). A continuació confirmarem la selecció amb OK.

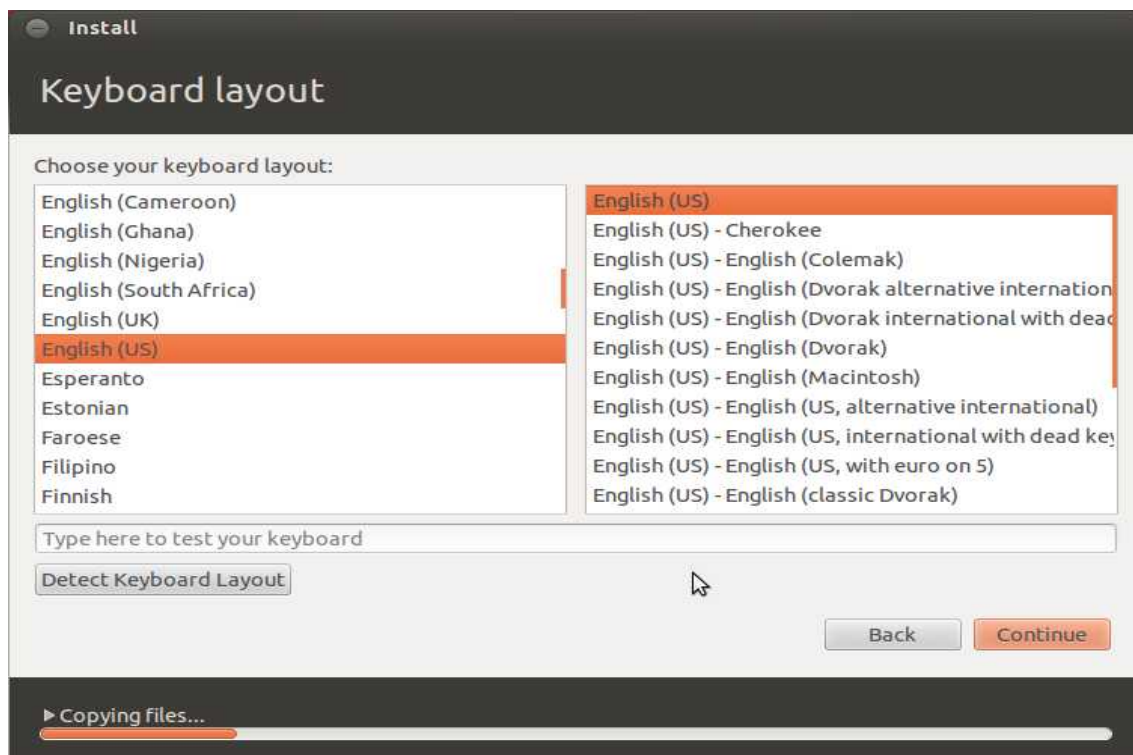


És molt important que seleccionem el dispositiu d'arrencada com a */dev/sdb* en ***Device for boot loader installation.***

A la finestra Where are you?, seleccionem la nostra zona horària.



I a Keyboard layout, la versió del nostre teclat.



En la finestra Who are you, triarem un nom d'usuari i una contrasenya per obrir una sessió d'Ubuntu.

Install

Who are you?

Your name: ✓

Your computer's name: ✓
The name it uses when it talks to other computers.

Pick a username: ✓

Choose a password:

Confirm your password:

Log in automatically

Require my password to log in

Encrypt my home folder

Back Continue

▶ Copying files...

A continuació comença la instal·lació d'Ubuntu en el disc dur extern.

Finalment, hauria d'aparèixer aquest missatge, indicant que Ubuntu s'ha instal·lat correctament. Per iniciar Ubuntu, s'ha de reiniciar l'equip.



Si vols iniciar l'ordinador des del disc extern, has d'accedir al menú d'arrencada i triar USB Device.

Captura de pantalla amb Screenshot

Per resoldre alguns dels exercicis d'aquests apunts, necessitaràs fer captures de pantalla. La forma més senzilla de fer ho és pitjant el botó Imprimir Pantalla (Print Screen), que activarà l'aplicació Screenshot.

Si no funcionés el botó Impr Pant, trobaràs l'aplicació Screenshot en l'apartat d'accessoris.

El directori (carpeta) predeterminada on es desaran les captures de pantalla és Pictures (Imatges).

Exercici 1.1.3-1

Instal·lar Ubuntu en un disc dur extern.

Data d'entrega dilluns 18/01/15.

1.2 Organització del sistema d'arxiu de Linux

S'anomena sistema d'arxiu (filesystem), la forma en què el sistema operatiu, en aquest cas Linux, organitza els directoris en els quals es poden guardar arxius.

Un arxiu és un document informàtic, per exemple un document creat amb el processador de textos **Writer**, del paquet ofimàtic LibreOffice (OpenOffice), una imatge, un arxiu de so o de vídeo, etc. Els arxius es reconeixen perquè normalment duen una extensió. Per exemple, un document creat amb Writer podria anomenar-se 'deures.odt', on 'odt' és l'extensió que està separada de la primera part del nom per un punt i identifica el document com a arxiu creat amb l'aplicació **Writer**. Tots els arxius creats amb l'aplicació **Writer** duen l'extensió 'odt'. Igualment podem identificar arxius de música per l'extensió 'mp3' i imatges per 'jpeg'.

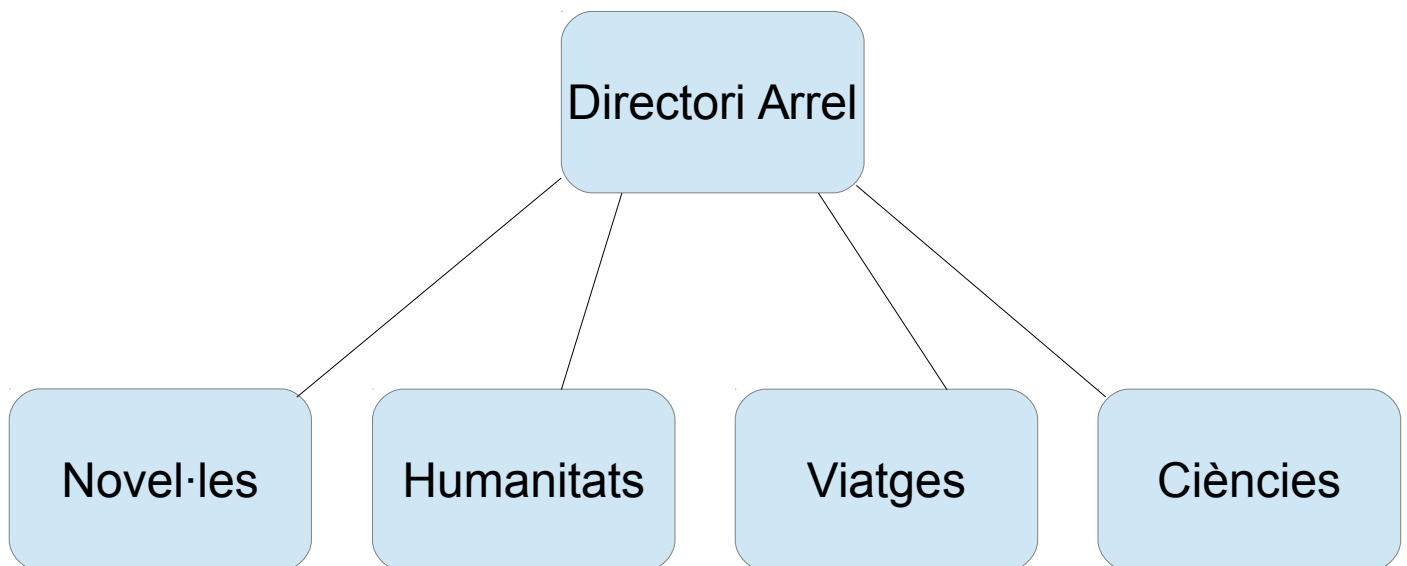
Quan treballem amb l'ordinador hem de guardar molts, sovint milers, d'arxius. Per poder guardar-los de forma organitzada, és necessari el sistema d'arxiu.

El sistema d'arxiu es compon de directoris (també anomenats carpetes), on guardem els arxius.

El sistema d'arxiu de Linux té una estructura que podríem comparar amb la d'una biblioteca, a la qual s'accedeix per una porta, que dona a una habitació. Aquesta primera habitació s'anomena **Directori Arrel** (root directory). En l'habitació **Directori Arrel**, podem guardar llibres. Els llibres de la nostra biblioteca representen els arxius o documents que guardem a l'ordinador.

En principi podríem guardar tots els nostres llibres en aquesta primera habitació, anomenada **Directori Arrel**, però això no és una bona idea, perquè tenim milers de llibres i, encara que tots tindrien lloc al **Directori Arrel**, seria complicat trobar un llibre en particular.

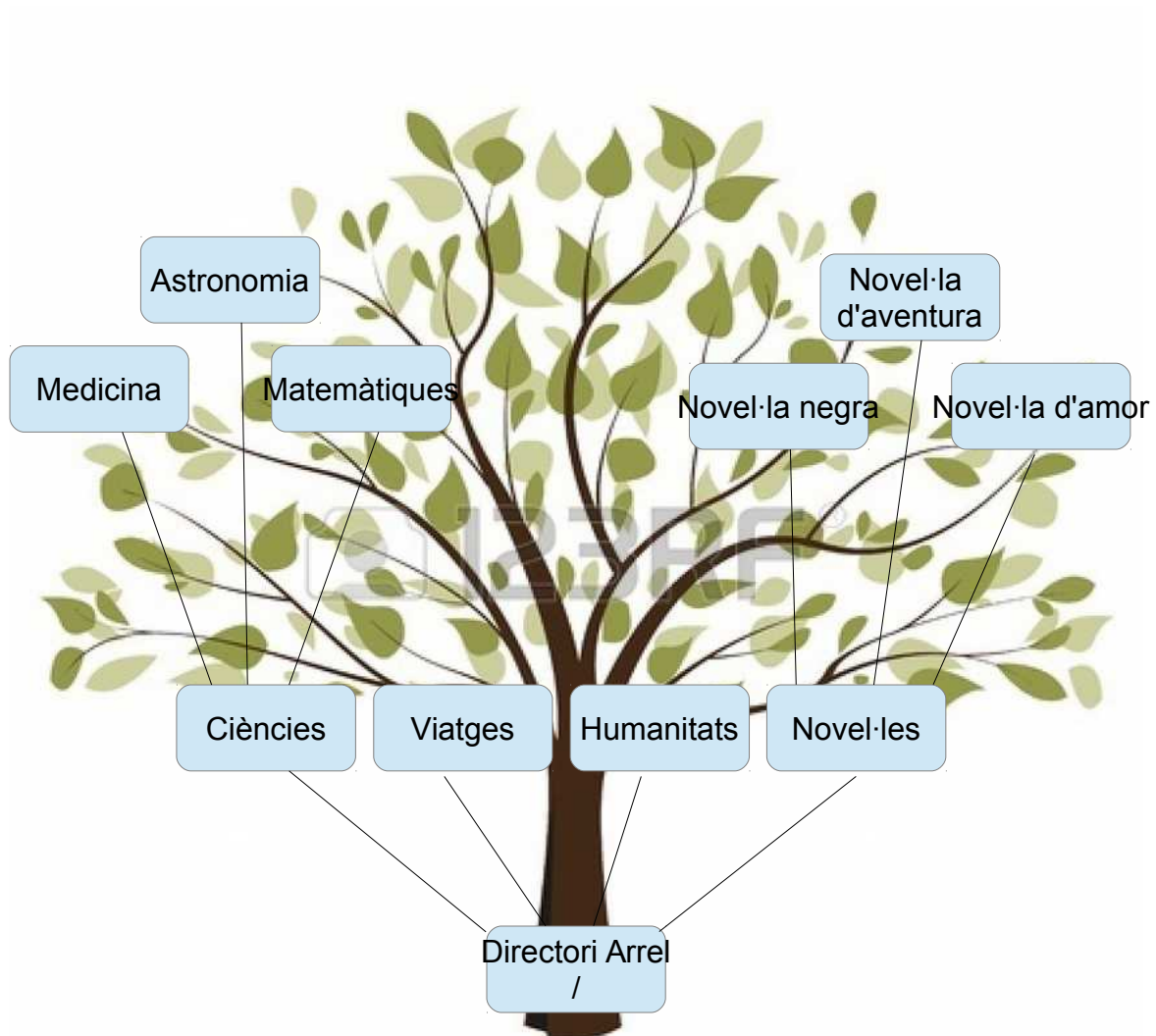
Seria més fàcil trobar un llibre, si el **Directori Arrel** disposés de portes a altres habitacions, per exemple anomenades Novel·les, Ciències, Humanitats, Viatges, etc. Així podríem organitzar els nostres llibres per continguts i seria molt més fàcil de trobar un llibre en particular.



Igual que el *Directori Arrel* té portes per passar a altres habitacions, aquestes habitacions poden tenir més d'una porta, és a dir, portes que duen a **noves** habitacions. La paraula noves de la frase anterior és molt important, perquè, per exemple, l'habitació Novel·les, té una porta que la comunica amb l'habitació *Directori Arrel*, però no pot tenir cap porta cap a les altres habitacions que comuniquin amb *Directori Arrel*. Per tant, per anar de l'habitació Novel·les a l'habitació Viatges, serà obligatori passar per l'habitació *Directori Arrel*. És impossible crear una porta directa entre les habitacions que comuniquen amb *Directori Arrel*.

Les habitacions Novel·les, Humanitats, Ciències i Viatges, que representen els **directoris** (carpetes) **del sistema d'arxiu**, s'anomenen **subdirectoris** del *Directori Arrel*. El *Directori Arrel* s'anomena **directori pare o original dels seus subdirectoris**.

Si volem afegir les habitacions Novel·la Negra, Novel·la d'Amor, Novel·la d'Aventura, Matemàtiques, Medicina, Astronomia, ho podríem fer de la següent manera.



L'organització d'aquest sistema d'arxiu es pot comparar amb la d'un arbre. El *Directori Arrel* representa l'arrel i el tronc de l'arbre, des dels quals creixen les branques.

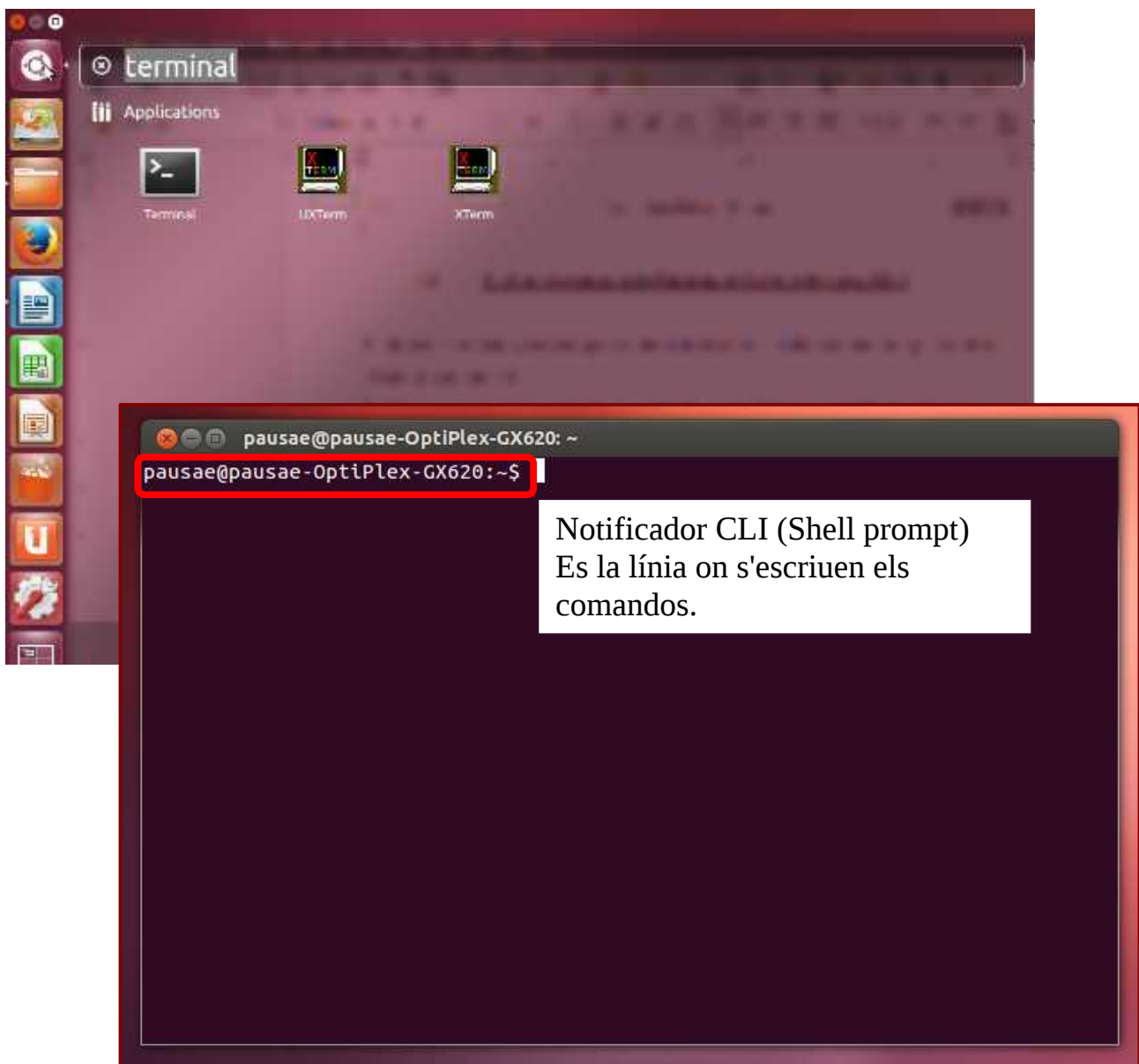
El *Directori Arrel* de Linux, des del qual 'creix' el sistema d'arxiu, es representa amb el símbol /

1.3 La línia de comando (Command Line Interface, CLI)

La línia de comando, serveix per comunicar-se amb el sistema operatiu i per controlar el seu funcionament.

En Linux, el programa que permet la comunicació entre usuari i sistema operatiu s'anomena **Shell**. El **Shell** rep comandos des del teclat i els transmet al sistema operatiu, que els executa.

Per accedir a la línia de comandos utilitzem l'aplicació **Terminal**.



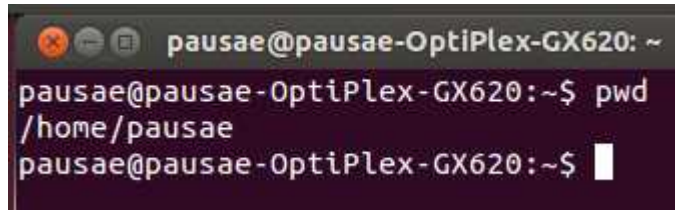
1.3.1 Primers passos (pwd, cd, ls)

En aquest moment tenim els coneixements necessaris per explorar el sistema d'arxiu del nostre sistema operatiu.

Per començar l'exploració, només ens falten algunes eines per poder-nos moure a través del sistema. Aquestes eines s'anomenen comandos. Un comando bàsic és el **pwd**.

pwd, del anglès **print working directory**, significa mostra el directori de treball, és a dir, mostra el directori on em trobo actualment (on estic treballant actualment).

Obrint l'aplicació **Terminal**, accedim al **Shell**. Si introduïm el comando **pwd**, el resultat podria ser el següent.



```
pausae@pausae-OptiPlex-GX620: ~  
pausae@pausae-OptiPlex-GX620:~$ pwd  
/home/pausae  
pausae@pausae-OptiPlex-GX620:~$
```

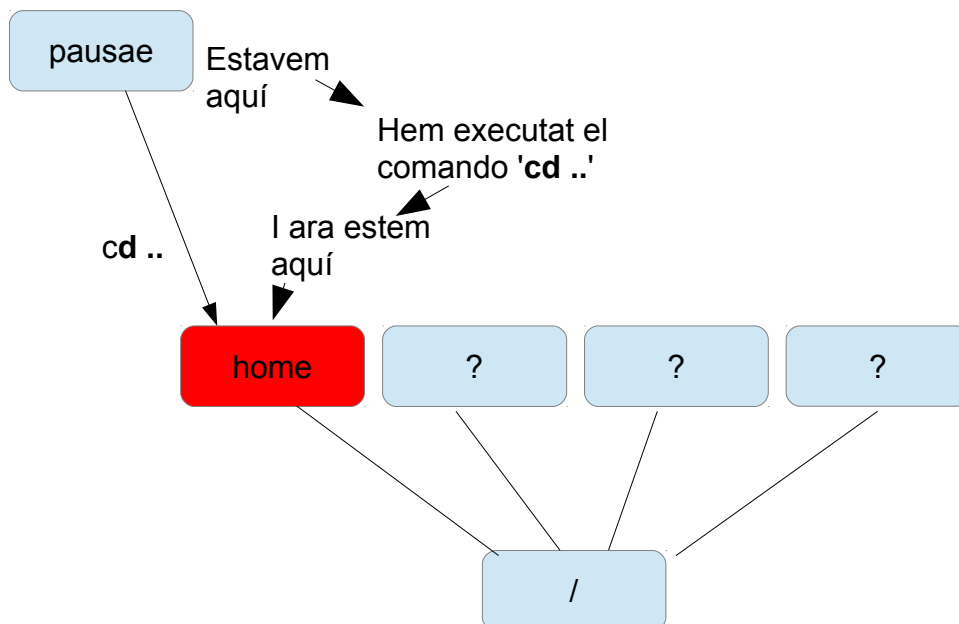
El resultat és `/home/pausae`, que vol dir, que estem en el directori anomenat *pausae*. Però no només diu això, també ens diu que el directori original (pare) del directori *pausae* és el directori *home* i que el directori original del directori *home* és `/`, el Directori Arrel.

Molt bé, sabem on estem, però encara no sabem com moure'ns del directori *pausae*, el nostre directori actual, a altres directoris.

Un aclariment, *home* vol dir llar en anglès i *pausae*, a més de ser el nom d'un directori, és el nom del usuari que està utilitzant l'ordinador. El directori llar (*home*) és aquell en el que l'usuari *pausae* guarda, de forma predeterminada, els seus arxius, i és raonable que dintre del directori llar hi hagi un subdirectorí anomenat igual que l'usuari, és a dir, *pausae*.

cd .. és el comando que necessitem per pujar al directori original, des del directori en el qual ens trobem. Provem-ho.

```
pausae@pausae-OptiPlex-GX620: /home
pausae@pausae-OptiPlex-GX620:~$ pwd
/home/pausae
pausae@pausae-OptiPlex-GX620:~$ cd ..
pausae@pausae-OptiPlex-GX620:/home$ pwd
/home
pausae@pausae-OptiPlex-GX620:/home$
```



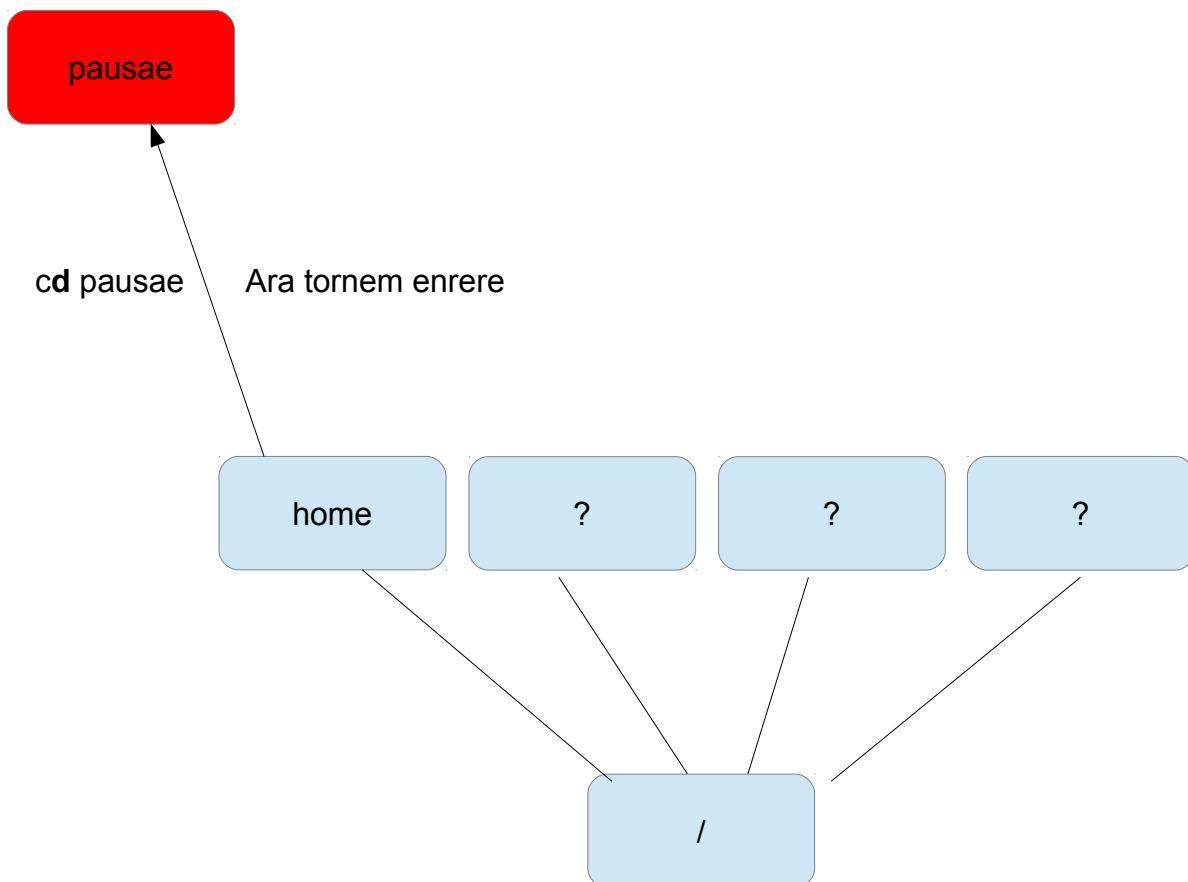
Per cert, **cd** ve de **change directory**, canvia directori.

Ara, algú hauria de protestar, dient: “ A mi no m'agrada aquest directori *home*, jo vull tornar al directori *pausae*.”

cd ./pausae

fa exactament això.

```
pausae@pausae-OptiPlex-GX620: ~  
pausae@pausae-OptiPlex-GX620:~$ pwd  
/home/pausae  
pausae@pausae-OptiPlex-GX620:~$ cd ..  
pausae@pausae-OptiPlex-GX620:/home$ pwd  
/home  
pausae@pausae-OptiPlex-GX620:/home$ cd pausae  
pausae@pausae-OptiPlex-GX620:~$ pwd  
/home/pausae  
pausae@pausae-OptiPlex-GX620:~$
```



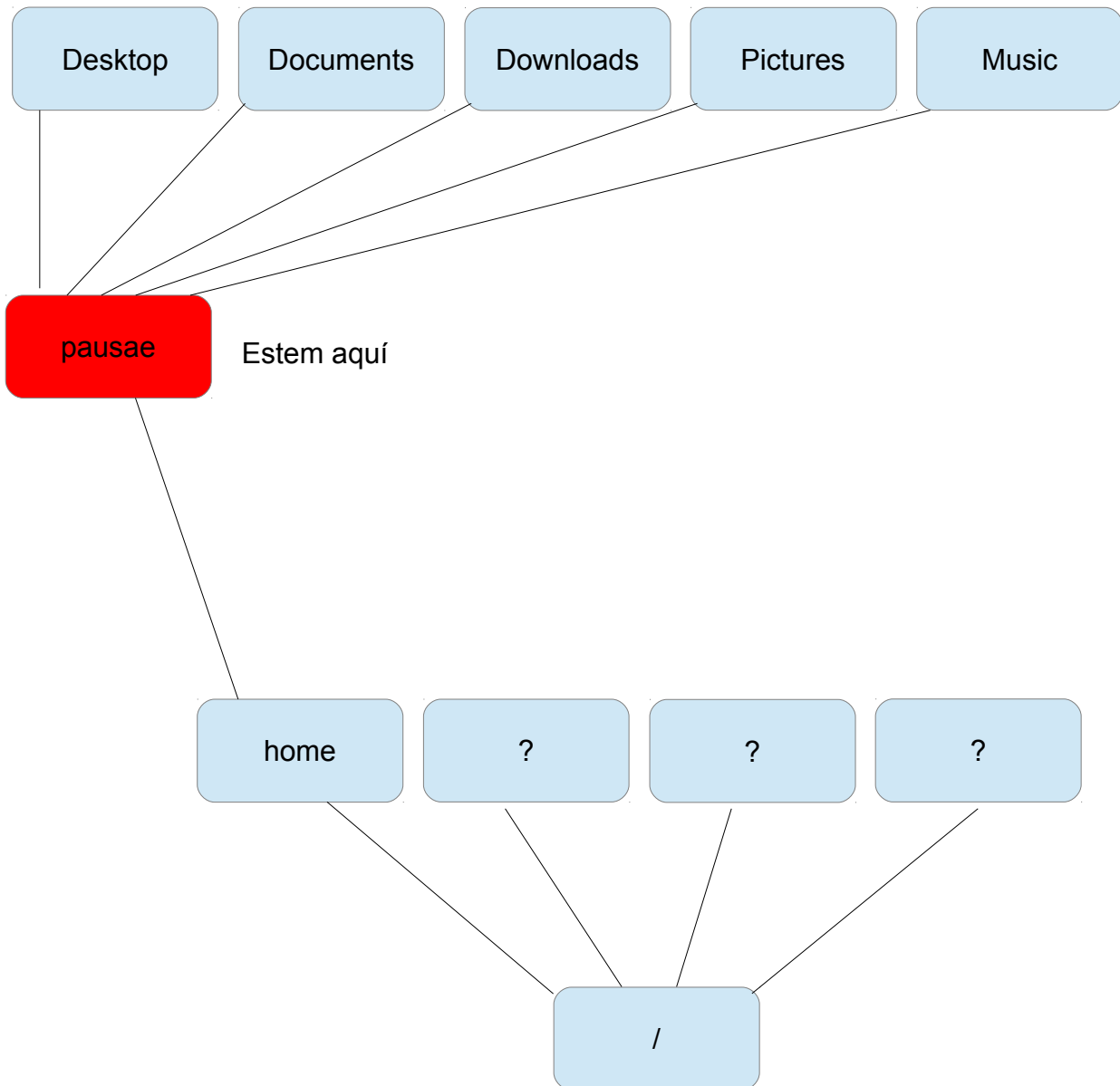
Sabem que el directori original de *pausae* és *home*, però *pausae*, tindrà subdirectoris?

ls donarà la resposta. **ls** és l'abreviació de *list*, és a dir, llistar.

```
pausae@pausae-OptiPlex-GX620: ~
pausae@pausae-OptiPlex-GX620:~$ pwd
/home/pausae
pausae@pausae-OptiPlex-GX620:~$ cd ..
pausae@pausae-OptiPlex-GX620:/home$ pwd
/home
pausae@pausae-OptiPlex-GX620:/home$ cd pausae
pausae@pausae-OptiPlex-GX620:~$ pwd
/home/pausae
pausae@pausae-OptiPlex-GX620:~$ ls
Desktop      Documents  examples.desktop  hs_err_pid3058.log  Public
desktopp    Downloads  fp.cfg            man                 Templates
desktoppcd  dwhelper  fp.dsk            Music               Ubuntu One
deures      EA1_05.jpg fp.ini            My Digital Editions Videos
doc_data.txt exam_1_2bc hs_err_pid2774.log Pictures            work
pausae@pausae-OptiPlex-GX620:~$
```

Aquí veiem tots els subdirectoris i arxius de *pausae*

Només s'han representat algunes dels subdirectoris de pausae



1.3.2 Nom de ruta (pathname)

Una ruta (path), es un camí a través del sistema d'arxiu.

```
ubuntu@ubuntu:~$ pwd
/home/ubuntu
ubuntu@ubuntu:~$ █
```

La imatge, mostra l'execució el comando `pwd`, i el resultat, que és la ruta des del *Directorí Arrel (/)* al directori `home`, i d'`home` al directori actual, que és *ubuntu*.

`/` → `home` → `ubuntu`

En iniciar una sessió en Ubuntu, el directori en el qual ens trobem per defecte és `/home/usuari`. En una sessió en viu, l'usuari es `ubuntu`.

Com he mostrat a l'apartat anterior, el comando `cd` permet canviar de directori. Amb `cd` podem moure'ns a través del sistema d'arxius utilitzant **noms de ruta absoluts** o **relatius**.

Nom de ruta absolut

El **nom de ruta absolut** comença pel *Directorí Arrel (/)* i segueix la branca que duu al directori o arxiu de destí.

Per exemple, el directori on el sistema operatiu desa la majoria de les aplicacions instal·lades, es `/usr/bin`. Com mostra el nom de ruta, el Directorí Arrel conté el directori `usr` i `usr` conté `bin`.

```
ubuntu@ubuntu: /usr/bin
ubuntu@ubuntu:~$ cd /usr/bin
ubuntu@ubuntu: /usr/bin$ █
```

Amb el comando `ls`, podem veure que el directori `bin` està ple d'arxius i directoris.

Us heu fixat com ha canviat el notificador CLI?

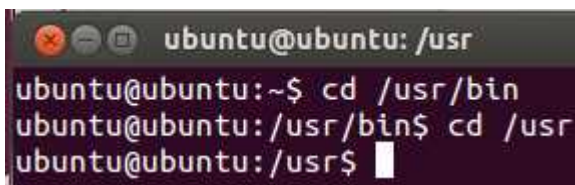
El comando `pwd` sempre mostra un nom de ruta absolut.

Nom de ruta relatiu

Mentre que els noms de ruta absoluts s'inicien en el Directori Arrel i finalitzen en el seu destí, els **noms de ruta relatius** comencen en el directori de treball, en el qual estem treballant. Per identificar un nom de ruta relatiu s'utilitza . (punt) o .. (punt punt). El símbol . representa el directori de treball i el símbol .. representa el directori original del directori de treball, és a dir, el directori del qual el directori de treball és subdirector.

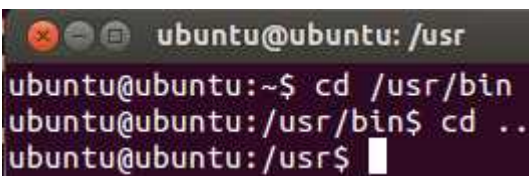
Per provar el funcionament dels noms de ruta, anem al directori /usr/bin.

Ara volem canviar al directori original de bin, que es usr. Podem fer ho de dues maneres diferents, primera utilitzant un **nom de ruta absolut**,



```
ubuntu@ubuntu: /usr
ubuntu@ubuntu:~$ cd /usr/bin
ubuntu@ubuntu:/usr/bin$ cd /usr
ubuntu@ubuntu:/usr$
```

segona utilitzant un **nom de ruta relatiu**.



```
ubuntu@ubuntu: /usr
ubuntu@ubuntu:~$ cd /usr/bin
ubuntu@ubuntu:/usr/bin$ cd ..
ubuntu@ubuntu:/usr$
```

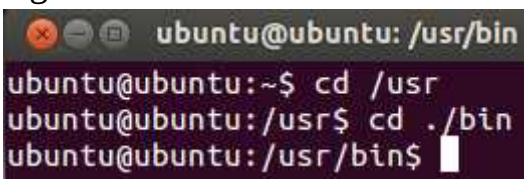
Igualment, per tronar al directori bin, tenim dues opcions,

primera amb **nom de ruta absolut**



```
ubuntu@ubuntu: /usr/bin
ubuntu@ubuntu:~$ cd /usr
ubuntu@ubuntu:/usr$ cd /usr/bin
ubuntu@ubuntu:/usr/bin$
```

segona amb **nom de ruta relatiu**



```
ubuntu@ubuntu: /usr/bin
ubuntu@ubuntu:~$ cd /usr
ubuntu@ubuntu:/usr$ cd ./bin
ubuntu@ubuntu:/usr/bin$
```

Característiques importants dels noms d'arxiu

- Els noms d'arxiu que comencen amb un punt estan amagats. Això significa que el comando ls únicament els mostrarà, si s'utilitza amb l'opció -a. Algunes aplicacions creen arxius de configuració amagats, perquè no es modifiquin de forma involuntària.
- Linux diferencia entre majúscules i minúscules en els noms d'arxiu i directori. Per tant, els arxius File1 i file1 són arxius diferents.
- Linux no utilitza el concepte d'extensió en els noms d'arxiu. Pots anomenar un arxiu com vulguis. No obstant això, és convenient seguir les convencions i utilitzar extensions per identificar arxius, per compartir-los amb altres sistemes operatius que sí en fan ús.
- Encara que Linux suporta noms d'arxiu llargs que poden contenir espais en blanc i caràcters de puntuació, es convenient utilitzar únicament lletres i punt (.), guió (-) i guió baix (_).

Com accedir a una memòria USB

Les explicacions, exemples i exercicis d'aquests apunts, estan fetes suposant que has iniciat el teu ordinador amb un disc d'inici i et trobes en una sessió en viu (live session).

Com sabem, en una sessió en viu, tot el que creem i desem en el directori home i els seus subdirectoris es perdrà en sortir de la sessió, ja que no es pot desar en el disc òptic amb la imatge d'Ubuntu. Per tant, si volem desar la nostra feina, hem d'accedir a un dispositiu d'emmagatzematge, que serà una memòria USB.

Sabem, que iniciant la sessió en viu, el directori de treball és /home/ubuntu.

Podem accedir a la memòria USB utilitzant el comando **cd** i especificant la ruta

```
cd /media/nom_de_la_memoria_USB
```

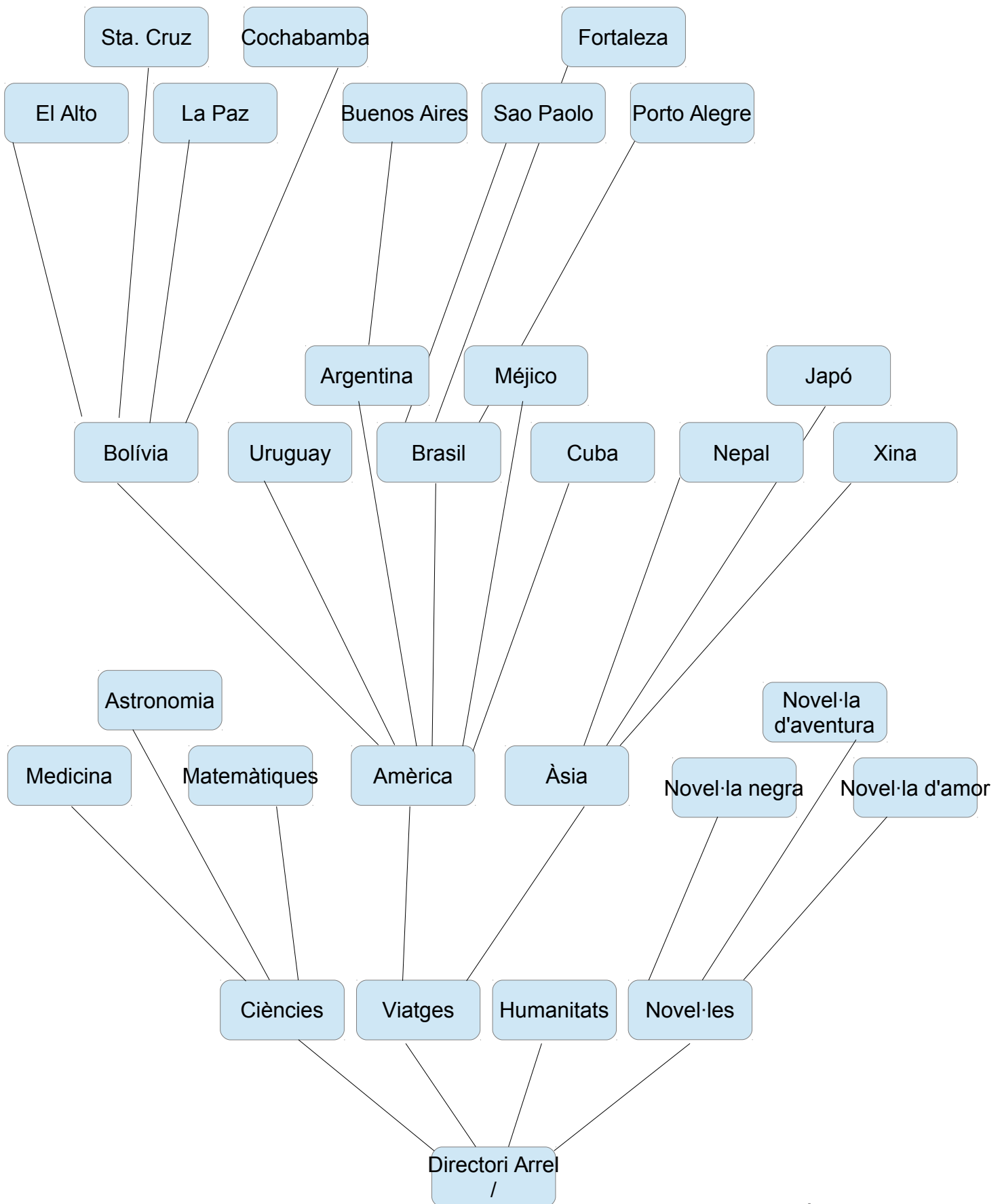
La meua memòria USB, per exemple, s'anomena PPOSADA, llavors, jo hi accedeixo així:

```
cd /media/PPOSADA
```

Exercici 1.3.2-1

- a) Quin és el directori original del directori novel·les (pàg. 37 [#arbre sistema arxiu](#))?
- b) Quins són els subdirectoris del directori novel·les?
- c) Estàs al directori Novel·les, indica la ruta absoluta del Directori Arrel a Astronomia.
- d) Estàs al directori Ciències, indica la ruta relativa a / i a Matemàtiques.
- e) Explica la diferència entre un nom de ruta absolut i un relatiu. Dóna un exemple de cadascun.
- f) Obre Terminal i des del Directori Arrel recorreix 3 branques diferents del sistema d'arxiu, amb una profunditat mínima de 3 directoris en cada branca. Escribeu el nom de ruta absolut de cada una de les branques. Exemple `/proc/sys/net/unix`.
- g) Et trobes en el directori `treballs_TIC` de la ruta:
`/home/ubuntu/son_pacs/TIC/treballs_TIC/segon_trim/activitat_5/exercicis`
Indica els comandos que necessaris per anar al directori ubuntu.
Indica els comandos necessaris per anar al directori exercicis.

Exercici 1.3.2-2



1. Indica una ruta absoluta al directori Brasil.

2. Et trobes al directori Bolívia.
 - a) Indica una seqüència de comandos per anar al directori Sta. Cruz.
 - b) Indica una seqüència de comandos per anar al directori Amèrica.
 - c) Indica una seqüència de comandos per anar al directori Viatges.
 - d) Indica una seqüència de comandos per anar al directori Àsia.

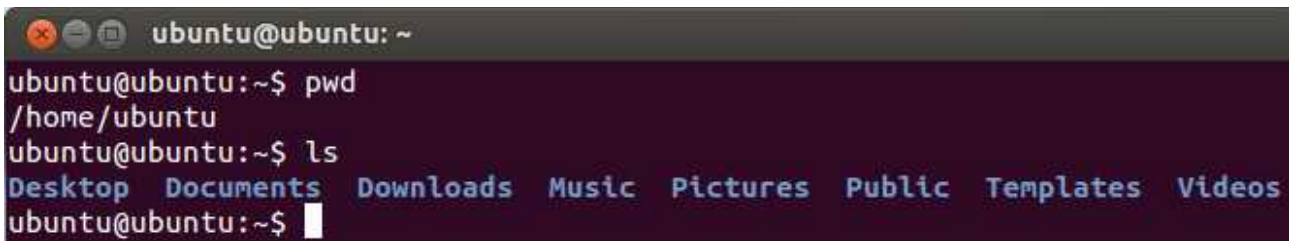
3. Et trobes al directori Viatges.
 - a) Utilitzant només un comando has de canviar al directori Novel·les. Quin és el comando?
 - b) Quina és la manera més ràpida d'arribar al Directori Arrel?

1.3.3 Explorant el sistema d'arxiu (ls, file, less)

Ja sabem com moure'ns pel sistema d'arxiu, ara ampliarem els nostres coneixements del comando *ls*, aprendrem a utilitzar *file*, per determinar el tipus d'arxiu i *less*, per mostrar el contingut d'arxius.

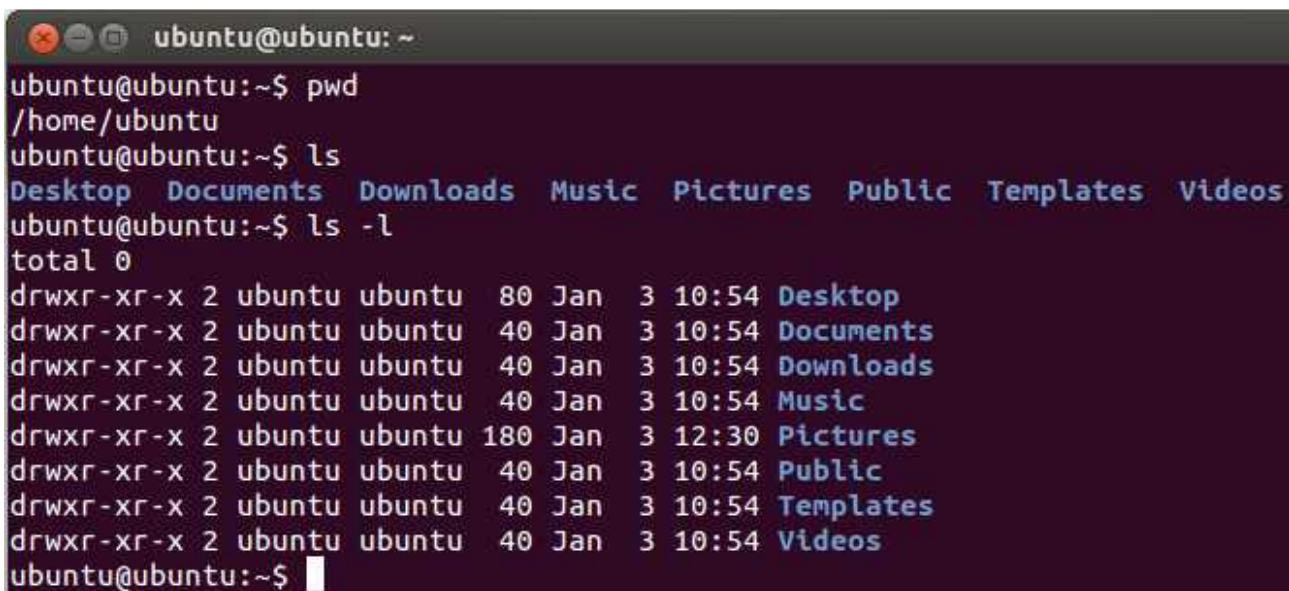
Més diversió amb ls

ls (list, llistar) és probablement el comando més utilitzat. Sabem que *ls* ens mostra el contingut d'un directori i atributs importants dels directoris i arxius que conté. En el directori de treball predeterminat el resultat és el següent.



```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ pwd  
/home/ubuntu  
ubuntu@ubuntu:~$ ls  
Desktop Documents Downloads Music Pictures Public Templates Videos  
ubuntu@ubuntu:~$
```

Per conèixer més detalls dels directoris o arxius mostrats, podem especificar l'opció *-l* (long format).



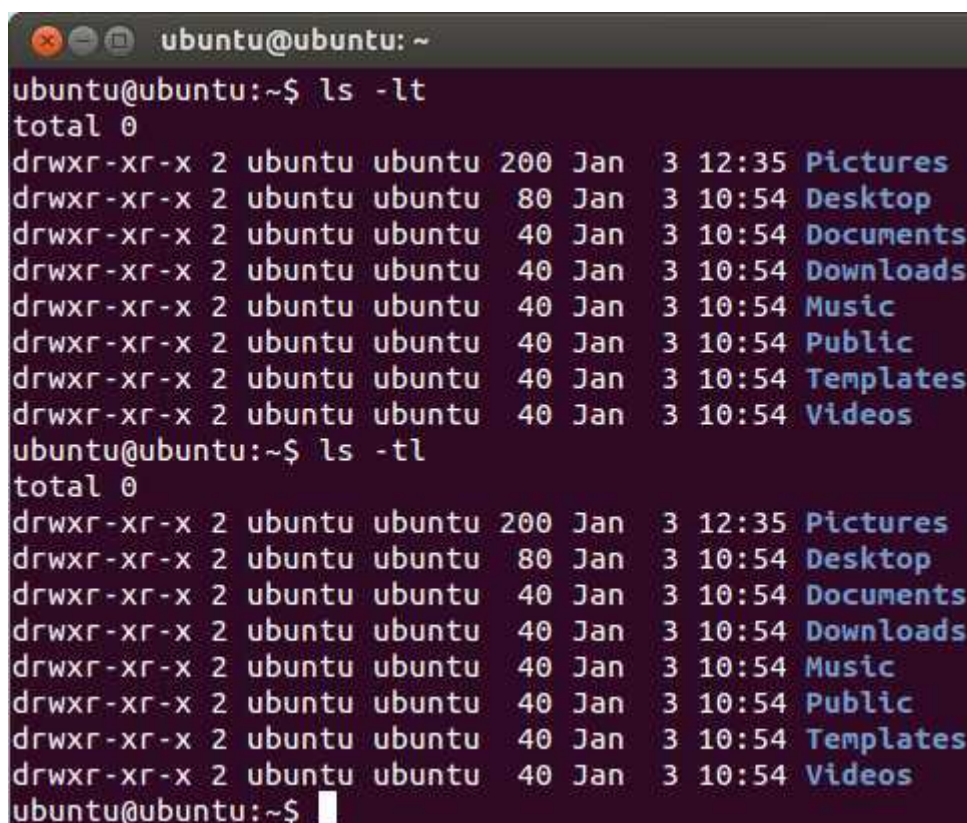
```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ pwd  
/home/ubuntu  
ubuntu@ubuntu:~$ ls  
Desktop Documents Downloads Music Pictures Public Templates Videos  
ubuntu@ubuntu:~$ ls -l  
total 0  
drwxr-xr-x 2 ubuntu ubuntu 80 Jan 3 10:54 Desktop  
drwxr-xr-x 2 ubuntu ubuntu 40 Jan 3 10:54 Documents  
drwxr-xr-x 2 ubuntu ubuntu 40 Jan 3 10:54 Downloads  
drwxr-xr-x 2 ubuntu ubuntu 40 Jan 3 10:54 Music  
drwxr-xr-x 2 ubuntu ubuntu 180 Jan 3 12:30 Pictures  
drwxr-xr-x 2 ubuntu ubuntu 40 Jan 3 10:54 Public  
drwxr-xr-x 2 ubuntu ubuntu 40 Jan 3 10:54 Templates  
drwxr-xr-x 2 ubuntu ubuntu 40 Jan 3 10:54 Videos  
ubuntu@ubuntu:~$
```

Opcions i arguments

En el darrer exemple hem vist un aspecte molt important, que mostra com funcionen la majoria dels comandos. A un comando sovint segueix una o més opcions que modifiquen el comportament del comando. A les opcions del comando, les segueixen els arguments sobre els quals el comando actua. Així, la sintaxis (estructura), de la majoria dels comandos és

comando -opció arguments

La majoria de les opcions d'un comando consisteixen en un únic caràcter, precedit per un guió, com vàrem veure amb `-l`. Però, també es possible combinar diverses opcions, amb el resultat d'un guió seguit per diverses lletres. Així, l'opció `t` del comando `ls`, ordena el resultat per hora de modificació dels arxius o directoris i es pot combinar amb l'opció `l`, com mostra el següent exemple



```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ ls -lt
total 0
drwxr-xr-x 2 ubuntu ubuntu 200 Jan  3 12:35 Pictures
drwxr-xr-x 2 ubuntu ubuntu  80 Jan  3 10:54 Desktop
drwxr-xr-x 2 ubuntu ubuntu  40 Jan  3 10:54 Documents
drwxr-xr-x 2 ubuntu ubuntu  40 Jan  3 10:54 Downloads
drwxr-xr-x 2 ubuntu ubuntu  40 Jan  3 10:54 Music
drwxr-xr-x 2 ubuntu ubuntu  40 Jan  3 10:54 Public
drwxr-xr-x 2 ubuntu ubuntu  40 Jan  3 10:54 Templates
drwxr-xr-x 2 ubuntu ubuntu  40 Jan  3 10:54 Videos
ubuntu@ubuntu:~$ ls -tl
total 0
drwxr-xr-x 2 ubuntu ubuntu 200 Jan  3 12:35 Pictures
drwxr-xr-x 2 ubuntu ubuntu  80 Jan  3 10:54 Desktop
drwxr-xr-x 2 ubuntu ubuntu  40 Jan  3 10:54 Documents
drwxr-xr-x 2 ubuntu ubuntu  40 Jan  3 10:54 Downloads
drwxr-xr-x 2 ubuntu ubuntu  40 Jan  3 10:54 Music
drwxr-xr-x 2 ubuntu ubuntu  40 Jan  3 10:54 Public
drwxr-xr-x 2 ubuntu ubuntu  40 Jan  3 10:54 Templates
drwxr-xr-x 2 ubuntu ubuntu  40 Jan  3 10:54 Videos
ubuntu@ubuntu:~$
```

Es veu que l'ordre de les opcions no influeix en el resultat del comando.

Explicació del llistat en format llarg

Però què significa tot aquest text que surt en utilitzar ls -l?

Analitzem la següent línia.

<code>drwxr-xr-x 2 ubuntu ubuntu 200 Jan 3 12:35 Pictures</code>						
1	2	3	4	5	6	7

	Camp de text	Significat
1	drwxr-xr-x	El primer dels 10 caràcters identifica un directori amb d i un arxiu amb -. A continuació només parlaré d'arxiu, però tot és igualment vàlid per a un directori. Els següents 3 caràcters (caràcters 2 a 4), mostres els drets d'accés a l'arxiu pel propietari. Els caràcters 5 a 7, mostres els drets d'accés a l'arxiu pels membres del grup. Els darrers 3 caràcters, mostres els drets d'accés a l'arxiu per la resta d'usuaris (els que no en són propietaris i els que no pertanyen al grup). La diferència entre el propietari d'un arxiu, el grup i la resta d'usuaris s'explicarà més endavant.
2	2	Nombre d'enllaços físics (hard links). S'explicarà més endavant el que és un enllaç físic (hard link).
3	ubuntu	Nom d'usuari del propietari de l'arxiu.
4	ubuntu	Nom del grup amb drets sobre l'arxiu.
5	200	Mida de l'arxiu en bytes.
6	Jan 3 12:35	Data i hora de la darrera modificació de l'arxiu.
7	Pictures	Nom de l'arxiu, en aquest cas un directori.

Permisos d'un arxiu o directori

Linux és un sistema operatiu multiusuari. Això significa que està dissenyat perquè diverses persones treballin amb el mateix ordinador. A casa teva, pot ser molt útil que cada un dels membres de la família disposi del seu nom d'usuari particular, i quan algú iniciï sessió de treball, només pugui accedir als directoris i arxius que ha creat i dels quals és propietari. D'aquesta manera, el teu germà petit no podrà eliminar els documents que vares crear per entregar demà a la professora de TIC.

En Linux és diferència entre 3 categories en relació amb un arxiu o directori:

- propietari
- grup
- tothom (tothom són tots aquells que no són ni propietari, ni membres del grup).

Però quins són els permisos que es poden tenir damunt aquests elements?

Doncs són 3 tipus de permisos:

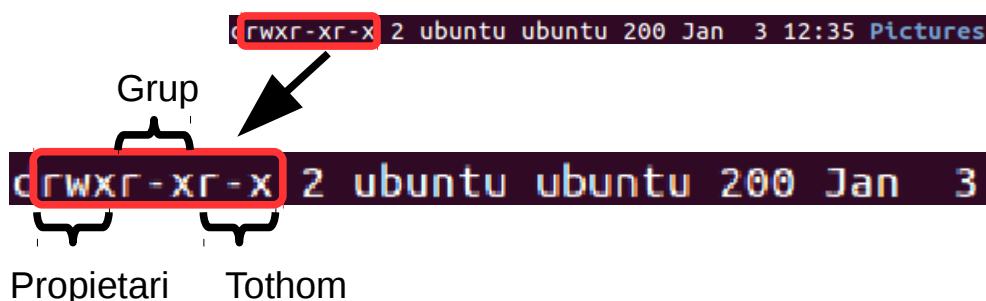
- permís d'escriptura **w** (write)
- permís de lectura **r** (read)
- permís d'execució **x** (execute)

És a dir, per a cada arxiu o directori s'han d'establir **9 permisos**.

Els del propietari, que normalment seran **rwX**, és a dir, podrà llegir i escriure l'arxiu i, si l'arxiu fos executable, executar-ho.

Els del grup, que és un grup d'usuaris que comparteixen arxius, que podrien ser **r-x**. Això significa, que els membres del grup podran llegir i executar els arxius dels altres membres del grup, però no els podran modificar, ja que no tenen permís **w** per escriure.

Finalment falta establir els permisos per a tothom, és a dir, la resta d'usuaris, que no són ni propietaris dels arxius, ni pertanyen al grup. Els permisos podrien ser **--** i significa, qui no sigui propietari ni membre del grup no pot llegir **r**, ni escriure **w**, ni executar **x** l'arxiu.



En la imatge, tothom té els mateixos permisos que el grup, **r-x**. Només el propietari pot escriure.

Exercici 1.3.3-1

Exercici 1.3.3-1

Indica quins permisos tenen el propietari, els membres del grup i tothom en els següents arxius i directoris.

```
p@p-OptiPlex-GX620:~/Desktop/carp/dijous$ ls -l
total 56
-----rwx 1 p p 1015 ene 27 06:16 caminante
-r--r--r-- 1 p p 524 ene 27 06:23 carmelina
----rwx--- 1 p p 646 ene 27 06:27 gatos
drwxr-x--- 3 p p 4096 ene 27 06:02 institut
-rw-r--r-- 1 p p 534 ene 27 06:31 musica.txt
d----- 2 p p 4096 ene 27 06:02 pelis
p@p-OptiPlex-GX620:~/Desktop/carp/dijous$
```

L'ordinador que mostra el llistat de la imatge l'utilitzen 4 usuaris:

- mare (propietaria de caminante, pelis)
- pare (propietari de gatos)
- filla (propietaria de institut)
- fil (propietari de carmelina)

Fil, filla i mare pertanyen al mateix grup.

Quins usuaris poden llegir l'arxiu musica.txt?

Quins usuaris poden escriure a l'arxiu caminante?

Quins usuaris poden executar els arxius del directori institut?

Determinant el tipus d'arxiu amb *file*

El comando *file* s'utilitza per determinar el contingut d'un arxiu. En Linux, un nom d'arxiu no necessàriament indica el seu contingut. Com vaig comentar abans, no és obligatori utilitzar el sistema d'extensions que coneixem d'altres sistemes operatius. Si ens trobem amb un arxiu anomenat imatge.jpg, suposarem, que es tracta d'un arxiu que conté una imatge comprimida JPEG. En Linux, el contingut d'imatge.jpg pot ser qualsevol cosa, per exemple un text.

Podem utilitzar el comando *file* així,

```
file nom_d_arxiu
```

El comando *file* presenta una breu descripció dels continguts de nom_d_arxiu.

```
ubuntu@ubuntu: /media/PPOSADA/sem_11_17/bat/linux
ubuntu@ubuntu:~$ cd /media/PPOSADA/sem_11_17/bat/linux/
ubuntu@ubuntu:/media/PPOSADA/sem_11_17/bat/linux$ file teo_1.html
teo_1.html: HTML document, ASCII text
ubuntu@ubuntu:/media/PPOSADA/sem_11_17/bat/linux$
```

La primera línia mostra com he passat del directori predeterminat al directori *linux* de la memòria USB, on està desat l'arxiu [teo_1.html](#). En la segona línia he executat el comando *file* teo_1.html i el resultat, en la tercera línia, mostra que teo_1.html és un arxiu html.

```
ubuntu@ubuntu: /media/PPOSADA/sem_11_17/bat/linux
ubuntu@ubuntu:~$ cd /media//PPOSADA/sem_11_17/bat/linux/
ubuntu@ubuntu:/media/PPOSADA/sem_11_17/bat/linux$ file teo_2.html
teo_2.html: JPEG image data, JFIF standard 1.01, comment: "GPEncoder"
ubuntu@ubuntu:/media/PPOSADA/sem_11_17/bat/linux$
```

Al segon exemple es veu, que l'arxiu [teo_2.html](#) no es html, sinó d'imatge (per visualitzar imatge, descarregar-la, afegint ?dl=1 a l'adreça http).

Llegint els continguts d'arxius amb *less*

El comando *less* serveix per veure el contingut d'un arxiu de text.

Per què volem llegir el contingut d'arxius de text? Perquè molts dels arxius de configuració de Linux tenen aquest format. Si som capaços de llegir el contingut d'aquests arxius, entendrem com funciona el nostre sistema operatiu. A més, molts dels programes que utilitza el sistema operatiu, anomenats scripts, també tenen aquest format.

Podem utilitzar el comando *less* així,

```
less nom_d_arxiu
```

less mostra el contingut de l'arxiu *nom_d_arxiu*

less permet avançar i retrocedir en l'arxiu de text. Per sortir de *less*, prem la tecla Q (quit -> finalitzar).

less comandos

Comando	Accio
Retrocedir p[ag o b	retrocedeix una p[agina
Avançar p[agina o espai	avança una p[agina
Fletxa arriba	retrocedeix una linia
Fletxa abaix	avança una linia
G	va a final d'arxiu
g	va a inici d'arxiu
/abc	cerca abc
n	continua cercant abc
h	mostra ajuda
q	finalitza <i>less</i>

Per exemple, per veure l'arxiu passwd. on es defineixen tots els comptes d'usuari del sistema, executem

```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ less /etc/passwd
```

amb el següent resultat

```
ubuntu@ubuntu: ~  
root:x:0:0:root:/root:/bin/bash  
daemon:x:1:1:daemon:/usr/sbin:/bin/sh  
bin:x:2:2:bin:/bin:/bin/sh  
sys:x:3:3:sys:/dev:/bin/sh  
sync:x:4:65534:sync:/bin:/bin/sync  
games:x:5:60:games:/usr/games:/bin/sh  
man:x:6:12:man:/var/cache/man:/bin/sh  
lp:x:7:7:lp:/var/spool/lpd:/bin/sh  
mail:x:8:8:mail:/var/mail:/bin/sh  
news:x:9:9:news:/var/spool/news:/bin/sh  
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh  
proxy:x:13:13:proxy:/bin:/bin/sh  
www-data:x:33:33:www-data:/var/www:/bin/sh  
backup:x:34:34:backup:/var/backups:/bin/sh  
list:x:38:38:Mail List Manager:/var/list:/bin/sh  
irc:x:39:39:ircd:/var/run/ircd:/bin/sh  
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh  
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh  
libuuid:x:100:101::/var/lib/libuuid:/bin/sh  
syslog:x:101:103:/home/syslog:/bin/false  
messagebus:x:102:105:/var/run/dbus:/bin/false  
colord:x:103:108:colord colour management daemon,,,:/var/lib/colord:/bin/false  
lightdm:x:104:111:Light Display Manager:/var/lib/lightdm:/bin/false  
whoopsie:x:105:114::/nonexistent:/bin/false  
avahi-autoipd:x:106:117:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false  
avahi:x:107:118:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false  
usbmux:x:108:46:usbmux daemon,,,:/home/usbmux:/bin/false  
kernoops:x:109:65534:Kernel Oops Tracking Daemon,,,:/bin/false  
pulse:x:110:119:PulseAudio daemon,,,:/var/run/pulse:/bin/false  
rtkit:x:111:122:RealtimeKit,,,:/proc:/bin/false  
speech-dispatcher:x:112:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/sh  
hplip:x:113:7:HPLIP system user,,,:/var/run/hplip:/bin/false  
saned:x:114:123:/home/saned:/bin/false  
ubuntu:x:999:999:Live session user,,,:/home/ubuntu:/bin/bash
```

Visita guiada pel sistema d'arxiu

L'estructura del sistema d'arxiu en Linux, s'assembla molt a la d'altres sistemes operatius similars a UNIX.

A continuació farem una petita excursió pel sistema d'arxiu, això ens donarà l'oportunitat de practicar les nostres facultats de moviment en el sistema.

Descobrirem que molts dels arxius són llegibles, de tipus text. Durant la visita practica utilitzar els següents comandos

1. Utilitza **cd** per moure't a través del sistema.
2. Llista directoris amb **ls -l**
3. Si trobes arxius interessants, determina el contingut amb **file**.
4. Si creus que és llegible, prova obrir-ho amb **less**.

Pren temps per ficar-hi una ullada. El sistema està a la teva disposició per ser explorat. Recorda, Linux no amaga la seva forma de funcionar.

La següent taula presenta alguns dels directoris més importants del sistema d'arxiu.

Directoris	Comentari
/	Directory Arrel, l'inici de tot plegat
/bin	Conte binaris (programes) necessaris per fer funcionar el sistema operatiu i per arrencar (boot).
/boot	Conté el kernel de LINUX, la imatge RAM inicial, necessària per arrencar el sistema (bootloader)
/media	En les versions de Linux modernes, el directori /media conté els punts de muntatge pels mitjans emmagatzematge com memòries USB, discs dura externs, càmeres o tablets connectats a l'ordinador.
/etc	Conté tots els arxius de configuració del sistema. Tots els arxius d'aquest directori haurien de ser llegibles amb less.

Directori	Comentari
/home	Cada usuari registrat en el sistema disposa d'un directori home. Els usuaris sense privilegis especials, només poden crear arxius en aquest directori
/lib	Conté les llibreries compartides. Correspon a DLL en Windows,
/usr	Aquest normalment és el directori amb més arxius en un sistema Linux. Conté tots els programes i arxius de suport utilitzats pels usuaris.
/usr/bin	Conté els programes executables (aplicacions) instal·lats en la distribució Linux. Sovint conté milers d'arxius.

Enllaços simbòlics

La primera línia de la imatge mostra el comando `ls -l`. Normalment, aquest comando ens hauria de mostrar en pantalla un llistat del directori `/usr/bin`. Però aquest llistat és molt llarg, ocupa diverses pantalles. Per això, està redirigit cap a un arxiu anomenat `list_usr_bin.txt`.

La redirecció es fa amb `>`, seguit de la ruta cap a l'arxiu `list_usr_bin.txt`.

```
ubuntu@ubuntu:/usr/bin$ ls -l > /home/ubuntu/Desktop/list_usr_bin.txt
ubuntu@ubuntu:/usr/bin$ less /home/ubuntu/Desktop/list_usr_bin.txt
```

Ara podem obrir l'arxiu `list_usr_bin.txt` amb **less**, com es fa a la segona línia i moure'ns còmodament, observant el resultat de `ls -l`.

Veurem que els arxius i directoris estan llistats per ordre alfabètic i trobarem `libreoffice`.

```
lrwxrwxrwx 1 root root 34 May 12 2014 libreoffice -> ../lib/libreoffice/program/soffice
```

Fixeu-vos, que la primera lletra és una `l`, i que sembla haver dos noms d'arxiu, `libreoffice` i `soffice`. Es tracta d'un tipus d'arxiu especial, anomenat enllaç simbòlic (simbolic link, soft link, symlink). En la majoria dels sistemes operatius similars a UNIX, es pot fer referència a un arxiu amb diversos noms.

Per entendre la utilitat d'aquesta característica, imaginem la següent situació.

Un programa necessita un arxiu compartit amb altres programes, anomenat foo.

L'arxiu foo s'actualitza amb freqüència amb versions millorades. Això significa que es convenient que el nom d'arxiu foo inclogui un nombre de versió. D'aquesta manera sabríem si estem utilitzant una versió de foo actualitzada o una antiga. Aquí tenim un problema. Si canviem el nom de l'arxiu compartit, hem de cercar cada un dels programes que l'utilitza i modificar en cada un d'aquests programes el nom de foo. Aquest procediment significa molta feina.

És en casos com aquest, que els enllaços simbòlics mostren la seva utilitat. Suposem que instalem la versió 2.6 de foo, amb el nom d'arxiu foo-2.6 i llavors creem un enllaç simbòlic anomenat foo, dirigit cap a foo-2.6 (foo → foo-2.6). Problema solucionat. Tots els programes que utilitzen foo com a recurs compartit, quan obren foo, realment obren foo-2.6. Els programes obren la versió de foo adequada i nosaltres, executant el comando ls -l, sabem quina versió de foo estem utilitzant. En el moment en que volguem actualitzar de foo-2.6 a foo-2.7, afegim el nou arxiu foo-2.7 al nostre sistema, eliminem el enllaç simbòlic foo → foo-2.6 i creem el nou enllaç simbòlic foo → foo-2.7.

En el nostre exemple de la imatge, l'enllaç simbòlic libreoffice està dirigit cap a l'arxiu soffice.

```
lrwxrwxrwx 1 root root 34 May 12 2014 libreoffice -> ../lib/libreoffice/program/soffice
```

Exercici 1.3.3-2

- Quin tipus d'arxiu és libreoffice?
- Quin 'es el nom de ruta absolut al directori on es troba l'arxiu libreoffice.?
- Quin 'es el nom de ruta absolut al directori on es troba l'arxiu soffice.?
- Accedeix a la teva memòria USB, fes una captura de pantalla, anomena-la 1331d i envia-la a pposada@iessonpacs.cat.

Exercici 1.3.3-3 (activitat 5 exercici 1)

Quins programes coneixes per editar arxius de text?

Utilitza *less* per llegir els arxius [momo.odt](#) i [momo.txt](#) que trobaras al [blog](#).

Quina diferència hi ha entre un arxiu de text i un document creat amb un processador de textos?

Com es codifiquen els caràcters d'un arxiu de text?

Data d'entrega dilluns 18/01/15.

Exercici 1.3.3-4 (activitat 5 exercici 2)

a	Obre l'aplicació <i>Terminal</i> . Executa un comando que mostri el directori on et trobes actualment. Fes una captura de pantalla i anomena-la <i>captura_1332a.png</i> .
b	Executa un comando que mostri un llistat dels arxius i subdirectoris del directori on et trobes actualment. Fes una captura de pantalla i anomena-la <i>captura_1332b.png</i> .
c	Canvia del directori actual al directori anomenat escriptori (Desktop). Quan hagi arribat al directori escriptori executa un comando que mostri en què directori et trobes. Fes una captura de pantalla, anomena-la <i>captura_1332c.png</i> .
d	Executa un comando que mostri un llistat dels arxius i subdirectoris del directori on et trobes actualment. Fes una captura de pantalla, anomena-la <i>captura_1332d.png</i> .

Desa les imatges en un document Writer anomenat *exercici_1.3.3-2.odt*.

Envia l'arxiu *exercici_1.3.3-2.odt* per correu electrònic a pposada@iessonpacs.cat.

Data d'entrega 18/01/15.

1.3.4 Manipulant arxius i directoris (cp, mv, mkdir, rm, ln, find)

- cp – Copiar arxius i directoris
- mv – Moure o renombrar arxius i directoris
- mkdir – Crear directoris
- rm – Eliminar arxius i directoris
- ln – Crear enllaços
- find – Cercar arxius

Aquests cinc comandos, s'utilitzen molt a Linux, per manipular tant arxius com directoris.

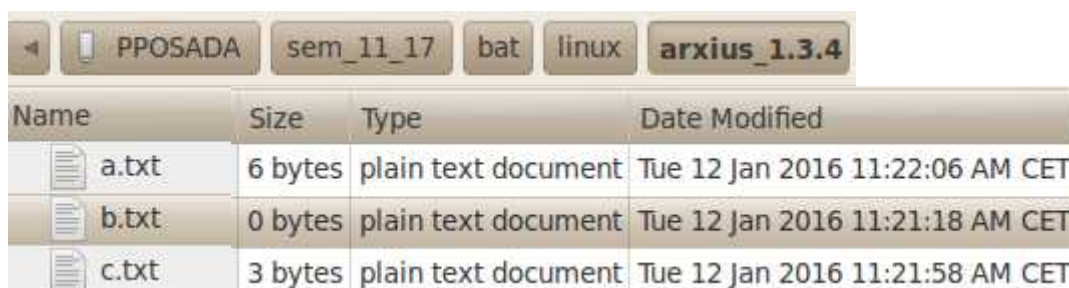
Sincerament, algunes de les tasques a realitzar amb aquests comandos, són més senzilles de realitzar amb la interfície gràfica (GUI) que amb la línia de comando (CLI). Amb un gestor d'arxius, podem copiar i enganxar arxius, o moure'ls de directori, amb molta facilitat. Llavors, per què utilitzar la línia de comando per aquesta tasca?

La raó és que les manipulacions senzilles, o de pocs arxius són fàcils de fer a la GUI, mentre que les manipulacions complicades, o de molts arxius, poden ser més senzilles utilitzant la línia de comando.

Per exemple, de la carpeta [arxius_1.3.4 actualitzada](#) volem copiar tots aquells arxius a la carpeta [arxius_1.3.4](#), que no existeixin en la carpeta de destí o, si hi existeixen, siguin més nous.

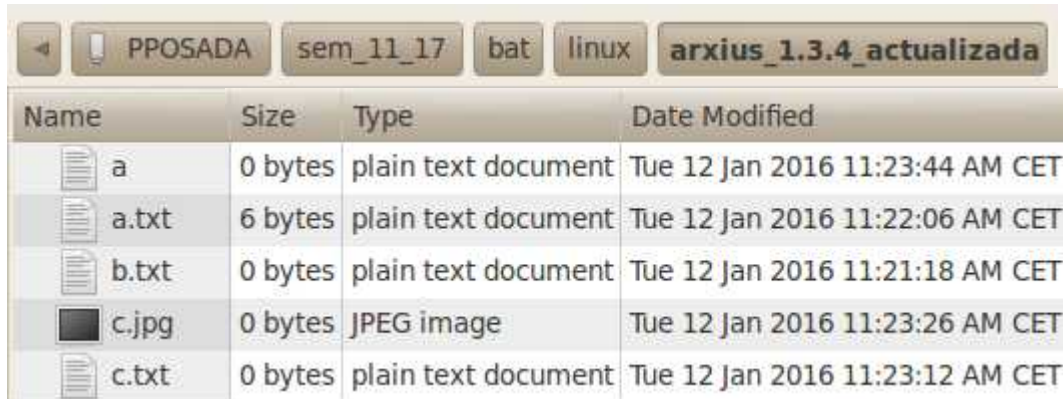
Per fer ho des de la GUI, procedim de la següent manera:

1. Obrim el directori `arxius_1.3.4` i mirem quins arxius conté.



Name	Size	Type	Date Modified
a.txt	6 bytes	plain text document	Tue 12 Jan 2016 11:22:06 AM CET
b.txt	0 bytes	plain text document	Tue 12 Jan 2016 11:21:18 AM CET
c.txt	3 bytes	plain text document	Tue 12 Jan 2016 11:21:58 AM CET

2. Ara obrim el directori `arxius_1.3.4_actualitzat` per comparar el seu contingut amb el d'`arxius_1.3.4`.



The screenshot shows a file manager window with the following breadcrumb path: PPOSADA > sem_11_17 > bat > linux > arxius_1.3.4_actualitzada. Below the path is a table listing the files in the directory.

Name	Size	Type	Date Modified
a	0 bytes	plain text document	Tue 12 Jan 2016 11:23:44 AM CET
a.txt	6 bytes	plain text document	Tue 12 Jan 2016 11:22:06 AM CET
b.txt	0 bytes	plain text document	Tue 12 Jan 2016 11:21:18 AM CET
c.jpg	0 bytes	JPEG image	Tue 12 Jan 2016 11:23:26 AM CET
c.txt	0 bytes	plain text document	Tue 12 Jan 2016 11:23:12 AM CET

3. La primera cosa que veiem és, que per actualitzar `arxius_1.3.4`, hem de copiar-hi els arxius `a` i `c.jpg`. A més, hem de mirar les dates de creació dels arxius i veiem, que no és necessari copiar `a.txt` ni `b.txt`, però sí `c.txt`.

Si tinguéssim molts arxius als directoris, fer-ho amb la GUI seria quasi impossible. Requeriria moltíssim temps i concentració, a més, seria fàcil cometre un error, mentre que des de la línia de comando es resol de forma molt senzilla amb el comando

```
cp -u * ../arxius_1.3.4
```

```
paulino@paulino-desktop:~$ cd /media/PPOSADA/sem_11_17/bat/linux/arxius_1.3.4
paulino@paulino-desktop:/media/PPOSADA/sem_11_17/bat/linux/arxius_1.3.4$ ls -l
total 8
-rwxr-xr-x 1 paulino paulino 6 2016-01-12 11:22 a.txt
-rwxr-xr-x 1 paulino paulino 0 2016-01-12 11:21 b.txt
-rwxr-xr-x 1 paulino paulino 3 2016-01-12 11:21 c.txt
paulino@paulino-desktop:/media/PPOSADA/sem_11_17/bat/linux/arxius_1.3.4$ cd ../arxius_1.3.4_actualizada
paulino@paulino-desktop:/media/PPOSADA/sem_11_17/bat/linux/arxius_1.3.4_actualizada$ ls -l
total 4
-rwxr-xr-x 1 paulino paulino 0 2016-01-12 11:23 a
-rwxr-xr-x 1 paulino paulino 6 2016-01-12 11:22 a.txt
-rwxr-xr-x 1 paulino paulino 0 2016-01-12 11:21 b.txt
-rwxr-xr-x 1 paulino paulino 0 2016-01-12 11:23 c.jpg
-rwxr-xr-x 1 paulino paulino 0 2016-01-12 11:23 c.txt
paulino@paulino-desktop:/media/PPOSADA/sem_11_17/bat/linux/arxius_1.3.4_actualizada$ cp -u * ../arxius_1.3.4
paulino@paulino-desktop:/media/PPOSADA/sem_11_17/bat/linux/arxius_1.3.4_actualizada$ cd ../arxius_1.3.4
paulino@paulino-desktop:/media/PPOSADA/sem_11_17/bat/linux/arxius_1.3.4$ ls -l
total 4
-rwxr-xr-x 1 paulino paulino 0 2016-01-17 08:02 a
-rwxr-xr-x 1 paulino paulino 6 2016-01-12 11:22 a.txt
-rwxr-xr-x 1 paulino paulino 0 2016-01-12 11:21 b.txt
-rwxr-xr-x 1 paulino paulino 0 2016-01-17 08:02 c.jpg
-rwxr-xr-x 1 paulino paulino 0 2016-01-17 08:02 c.txt
paulino@paulino-desktop:/media/PPOSADA/sem_11_17/bat/linux/arxius_1.3.4$
```

Exercici 1.3.4-1

Descarrega les carpetes [carp](#) i [carp_act](#) al directori home, i actualitza la carpeta *carp* seguint l'exemple anterior.

Fes una captura de pantalla que mostri el procés i envia-la a pposada@iestsonpacs.cat.

Caràcters comodí

A diferència de la GUI, on fitxers i carpetes es representen amb icones, a la CLI, s'utilitza text, els noms dels arxius i carpetes. Per poder especificar de forma senzilla grups de noms, s'utilitzen els caràcters comodí.

Comodí	Representa
*	Qualsevol grup de caràcters
?	Qualsevol caràcter individual
[conjunt de caràcters]	Qualsevol caràcter que pertany al conjunt de caràcters
[!conjunt de caràcters]	Qualsevol caràcter que no pertany al conjunt de caràcters
[:class:]	Qualsevol caràcter que pertany a la classe especificada

Classes de caràcters	Representa
[:alnum:]	Qualsevol caràcter alfanumèric
[:alpha:]	Qualsevol caràcter de l'alfabet
[:digit:]	Qualsevol dígit
[:lower:]	Qualsevol caràcter en minúscula
[:upper:]	Qualsevol caràcter en majúscula

Classes de caràcters	Representa
[:alnum:]	Qualsevol caràcter alfanumèric
[:alpha:]	Qualsevol caràcter de l'alfabet
[:digit:]	Qualsevol dígit
[:lower:]	Qualsevol caràcter en minúscula
[:upper:]	Qualsevol caràcter qen majúscula

Utilitzant els comodins, es poden construir patrons molt sofisticats per a la identificació de noms d'arxius o carpetes.

Exemples d'utilització dels comodins

Classes de caràcters	Representa
*	Qualsevol nom d'arxiu
g*	Qualsevol d'arxiu que comenci per g
b*.txt	Qualsevol arxiu que comenci per b i acabi amb .txt
Data???	Qualsevol arxiu que comenci amb Data , seguit de tres caràcters
[abc]*	Qualsevol arxiu que comenci amb un caràcter dels caràcters a, b o c
BACKUP.[0-9][0-9][0-9]	Qualsevol arxiu que comenci amb BACKUP. , seguit de 3 xifres.
[:upper:]*	Qualsevol arxiu que comenci amb un caràcter en majúscula
	Qualsevol arxiu que no comenci amb una xifra
*[:lower:]123]	Qualsevol arxiu que termini amb un caràcter en minúscula o una de les xifres 1, 2 o 3

Els caràcters comodí es poden utilitzar amb qualsevol comando que tingui un nom d'arxiu com a argument.

Exercici 1.3.4-2

El comando `find` *argument* cerca els arxius o directoris que coincideixen amb *argument*. Exemple, `find ?a*` dóna com a resultat tots els arxius amb *a* com a segona lletra.

```
p@p-OptiPlex-GX620:/media/p/PPOSADA/sem_11_17/bat/linux/carp$ find ?a*
gamma.txt
kappa
lambda.ods
lamda.mp3
tau
```

Quin són els comandos necessaris per obtenir els següents resultats amb els arxius del directori [carp](#)?

<p>a</p> <pre>alpha.txt beta.txt gamma.txt IOTa123.txt omicron.txt pis.txt xi.txt zeta.txt</pre>	<p>b</p> <pre>omega omicron omicron.txt</pre>	<p>c</p> <pre>alpha.txt beta.txt delta.jpg epsilon.mp3 eta.mov gamma.txt IOTa123.txt iota.odt lambda.ods lamda.mp3 mu.odt nu.jpg omicron.txt phi123.jpg pis.txt PSI456.mov psi.jpg psi.ods psi.odt theta.jpg xi.txt zeta.txt</pre>	<p>d</p> <pre>GAMMA kappa omega sigma theta</pre>	<p>e</p> <pre>BETA_456 epsilon.mp3 IOTa123.txt lamda.mp3 phi123.jpg PSI456.mov TAU789</pre>	<p>f</p> <pre>epsilon.mp3 lamda.mp3</pre>
<p>g</p> <pre>BETA_456 epsilon.mp3 lamda.mp3 TAU789</pre>					

mkdir – Crear directoris (make directory)

El comando **mkdir** s'utilitza per crear directoris. Funciona així:

```
mkdir directori...
```

Una observació respecte a la notació: Els punts suspensius darrere l'argument del comando, significa que hi pot haver més d'un argument, per exemple

```
mkdir dir1
```

crearia el directori dir1, mentre que

```
mkdir dir1 dir2 dir3
```

crearia els directoris dir1, dir2 i dir3.

cp – Copiar arxius i directoris (copy)

El comando **cp** copia arxius o directoris. Pot ser utilitzat de dues formes diferents:

```
cp element1 element2
```

per copiar un arxiu o directori element1 a l'arxiu o directori element2 i:

```
cp element... directori
```

per copiar múltiples elements a directori.

Les següents taules mostren algunes de les opcions més freqüents utilitzades amb **cp**.

cp - opcions	Descripció
-a, --archive	Copia arxius i directoris amb tots els atributs, inclosos propietari i permisos. Normalment, les còpies reben els atributs per defecte de l'usuari que fa la còpia. S'explicarà més endavant en detall.
-i, --interactive	Abans de sobreesciure un arxiu existent, avisa l'usuari perquè confirmi l'acció. Sense aquesta opció, cp sobreescrirà els arxius sense mostrar-ho.
-r, --recursive	Copia directoris i els seus continguts de forma recursiva. Aquesta opció (o l'opció -a) són necessàries per copiar directoris.
-u, --update	Copiant arxius d'un directori a un altre, copia només els arxius que no existeixen en el directori de destí o que, si existeixen en el directori de destí, tenen una data de creació més nova.
-v, --verbose	Mostra missatges d'informació mentre que s'està realitzant el procés de còpia.

cp - exemples	Descripció
cp arxiu1 arxiu2	Copia l'arxiu1 a l'arxiu2. Si l'arxiu2 existeix, se sobreesciu. Si l'arxiu2 no existeix, es crea.
cp -i arxiu1 arxiu2	Igual que l'anterior, excepte que si l'arxiu2 existeix, l'usuari és avisat perquè confirmi l'acció, abans de sobreesciure l'arxiu2.
cp arxiu1 arxiu2 dir1	Copia els arxius 1 i 2 al directori dir1. El directori dir1 ha d'existir prèviament.
cp dir1/* dir2	Utilitzant un caràcter comodí, tots els arxius de dir1 són copiats a dir2. Dir2 ha d'existir prèviament.
cp -r dir1 dir2	Copia el dir1 i els seu contingut, al dir2. Si dir2 no existeix prèviament, és creat, i el seu contingut serà idèntic al de dir1.

Exercici 1.3.4-3

Descarrega la carpeta [carp](#) al directori home. Obre un Terminal i accedeix al directori home. Llista els continguts de home i comprova que s'hi trobi carp.

- a) Copia l'arxiu beta.txt de carp a /home/usuari, canviant el seu nom a beta_usuari.txt.
- b) Obre beta_usuari.txt per llegir-ho. Fes una captura de pantalla i anomena-la 1.3.4.3-b.
- c) Obre l'arxiu nu.jpg per llegir-ho. Fes una captura de pantalla i anomena-la 1.3.4.3-c.
- d) Copia l'arxiu nu.jpg de carp a l'arxiu beta_usuari.txt del directori usuari. Fes la copia utilitzant una opció que t'avisi que sobreescriràs beta_usuari.txt. Obre beta_usuari.txt per llegir-ho.
- e) Copia l'arxiu beta_usuari.txt al directori carp.
- f) Elimina l'arxiu beta_usuari.txt del directori /home/usuari.
- g) Crea el directori carp_copia a /home/usuari. Fes una captura de pantalla i anomena-la 1.3.4.3-g.
- h) Copia el directori carp al directori carp_copia.
- i) Elimina l'arxiu beta_usuari.txt de carp_copia.
- j) Crea a carp_copia el directoris carp_sub_1 i carp_sub_2.
- k) Copia a carp_sub_1 tots els arxius amb extensió .txt i a carp_sub_2 tots els arxius amb una xifra en el seu nom.
- l) Copia el directori carp_copia a carp_copia_2 de forma recursiva. Fes una captura de pantalla i anomena-la 1.3.4.3-l.

Insereix les imatges 1.3.4.3-b, 1.3.4.3-c, 1.3.4.3-g i 1.3.4.3-l en un document Writer, anomenat exercici_1343.odt, i envia'l a pposada@iessonpacs.cat.

mv – Moure i renombrar arxius (move)

El comando **mv** serveix per moure o canviar el nom d'arxius, depenent de com s'utilitza. En qualsevol cas, el nom d'arxiu original desapareix del directori on es trobava. **mv** s'utilitza d'una forma molt similar a **cp**.

mv element1 element2

Mou o canvia el nom d'un arxiu o directori d'element1 a element2.

mv element... directori

Mou un o diversos elements d'un directori a un altre.

mv comparteix moltes de les opcions amb **cp**.

mv - opcions	Descripció
-i, --interactive	Abans de sobreescrivre un arxiu existent, avisa l'usuari perquè confirmi l'acció. Sense aquesta opció, mv sobreescrirà els arxius sense mostrar-ho.
-u, --update	Mou arxius d'un directori a un altre, però només aquells arxius que no existeixen en el directori de destí o que, si hi existeixen, tenen una data de creació més nova.
-v, --verbose	Mostra missatges d'informació mentre que s'està realitzant el procés de còpia.

mv - exemples	Descripció
mv arxiu1 arxiu2	Mou l'arxiu1 a l'arxiu2. Si l'arxiu2 existeix, se sobreescriv. Si l'arxiu2 no existeix, es crea. En amdos casos, l'arxiu1 s'elimina.
mv -i arxiu1 arxiu2	Igual que l'anterior, excepte que si l'arxiu2 existeix, l'usuari és avisat perquè confirmi l'acció, abans de sobreescrivre l'arxiu2.
mv arxiu1 arxiu2 dir1	Mou els arxius 1 i 2 al directori dir1. El directori dir1 ha d'existir prèviament.
mv dir1 dir2	Mou el dir1 i els seu contingut, al dir2. Si dir2 no existeix prèviament, és creat.El dir1 queda eliminat.

rm – Eliminar arxius i directoris (remove)

El comando **rm** s'utilitza per eliminar arxius i directoris.

rm *element...*

Precaució amb rm!

Els sistemes operatius similars a UNIX no disposen d'un comando desfer (undelete). Una vegada eliminat un element amb rm, no tornarà. Linux assumeix que saps el que fas quan elimines.

Posa especial atenció quan utilitzis caràcters comodí. Un exemple típic. Suposa que vols eliminar únicament els arxius HTML d'un directori. El comando seria

rm *.html

això en principi és correcte, però si accidentalment escrivissis un espai en blanc entre * i .html

rm * .html

el comando eliminaria tots els arxius del directori i donaria un missatge avisant que no existeix l'arxiu .html.

Un truc molt útil és provar el resultat de rm amb ls, abans d'utilitzar rm. Així abans d'eliminar, veuràs en pantalla els elements a eliminar.

El comando

ls * .html

mostraria tots els elements a eliminar i t'adonaries de l'error.

rm - opcions	Descripció
-i, --interactive	Abans d'eliminar un arxiu existent, avisa l'usuari perquè confirmi l'acció. Sense aquesta opció, rm sobreescrirà els arxius sense mostrar-ho.
-r, --recursive	Elimina directoris i els seus continguts de forma recursiva. Aquesta opció (o l'opció -a) són necessàries per eliminar directoris amb diversos nivells de subdirectoris.
-f, --force	Ignora elements no existents i no avisa. Suprimeix missatges d'avís.
-v, --verbose	Mostra missatges d'informació mentre que s'està realitzant el procés d'eliminació.

rm - exemples	Descripció
rm arxiu1	Elimina l'arxiu1 sense mostrar avís.
rm -i arxiu1	Igual que l'anterior, excepte que l'usuari és avisat perquè confirmi l'acció, abans d'eliminar l'arxiu1.
rm -r arxiu1 dir1	Elimina arxiu1 i dir1 i els seus continguts.
rm -rf arxiu1 dir1	Igual que l'anterior, excepte que si arxiu1 o dir1 no existeix, no es mostrarà avís.

ln – Crear enllaços (link)

El comando **ln** s'utilitza per crear tant enllaços físics (enllaç dur, hard link), com simbòlics. S'utilitza d'una de les dues següents maneres:

ln *arxiu link*

per crear un enllaç físic i

ln -s *element link*

per crear un enllaç simbòlic, on element pot ser un arxiu o un directori.

Enllaços físics (hard link)

L'enllaç físic és la forma original d'enllaç utilitzada en UNIX, l'enllaç simbòlic és més modern. De forma predeterminada, cada arxiu disposa d'un enllaç físic que identifica l'arxiu amb un nom. Un enllaç físic té dues limitacions importants:

- L'enllaç físic no pot fer referència (enllaçar) a un arxiu fora del seu sistema d'arxiu. Això significa que l'enllaç no serveix per anomenar un arxiu fora de la partició del disc on es troba l'enllaç.
- L'enllaç físic no serveix per anomenar un directori.

L'enllaç físic és indistingible de l'arxiu. A diferència d'un llistat de noms de directoris, que contenen enllaços simbòlics i on els enllaços simbòlics estan identificats per una marca específica, en un llistat de noms de directori amb enllaços físics, els enllaços físics no estan marcats.

Enllaços simbòlics (symbolic link)

Els enllaços simbòlics foren creats per superar les limitacions dels enllaços físics. L'enllaç simbòlic és un tipus especial d'arxiu que indica una ruta cap a l'arxiu o el directori amb el qual enllaça.

Un arxiu enllaçat amb un enllaç simbòlic i l'enllaç simbòlic mateix, aparenten ser el mateix element. Així, si s'escriu a l'enllaç simbòlic, la informació es desa a l'arxiu amb el qual enllaça. No obstant, quan s'elimina un enllaç simbòlic, només s'elimina l'enllaç, l'arxiu al qual l'enllaç feia referència continua existint. Si l'arxiu s'elimina abans que l'enllaç simbòlic, l'enllaç continua existint, però no senyalarà a cap element. En aquest cas es diu que l'enllaç està romput (broken link). Moltes versions de Linux ressalten els enllaços romputs en vermell.

Un lloc on practicar - playground

Per practicar de forma segura amb els comandos que hem après fins ara, crearem un lloc segur on poder manipular arxius i directoris. En el directori `/home/usuari` crearem el subdirectori anomenat *playground*.

Creant directoris

El comando `mkdir` s'utilitza per crear directoris. Abans de crear el directori *playground*, ens comprovarem en quin directori estem.

```
p@p-OptiPlex-GX620:~$ pwd
/home/p
p@p-OptiPlex-GX620:~$ mkdir playground
p@p-OptiPlex-GX620:~$ ls
2016-01-10-180249_1280x1024_screenshot.png  dwl_...
carp                                          examples.desktop
carp_act                                    Music
carp_act.zip                                Pictures
carp.zip                                    playground
deja-dup                                    PlayOnLinux's virtual drives
dell                                         Public
Desktop                                     swap~
Documents                                   Templates
Downloads                                   Videos
p@p-OptiPlex-GX620:~$
```

Comprovat on estem
Creat directori i comprovat contingut
Enllaços romputs

Dintre de *playground* creem els directoris *dir1* i *dir2*. Per això canviem del nostre directori actual al directori *playground* i creem els directoris.

```
p@p-OptiPlex-GX620:~$ cd ./playground/
p@p-OptiPlex-GX620:~/playground$ mkdir dir1 dir2
p@p-OptiPlex-GX620:~/playground$ ls
dir1 dir2
p@p-OptiPlex-GX620:~/playground$
```

Observa que `mkdir` accepta múltiples arguments, de forma que és possible crear *dir1* i *dir2* en un sol pas.

Copiant arxius

A continuació inserirem dades en el directori *playground*. Ho farem copiant un arxiu. Utilitzant el comando *cp*, copiarem l'arxiu *passwd* que trobarem al directori */etc* al directori *playground*.

```
p@p-OptiPlex-GX620:~/playground$ cp /etc/passwd .
p@p-OptiPlex-GX620:~/playground$ ls
dir1 dir2 passwd
p@p-OptiPlex-GX620:~/playground$
```

El primer argument del comando *cp* mostra un nom de ruta absolut al fitxer que volem copiar. Fixat com es va utilitzar el *.* com a nom de ruta relatiu, per indicar el directori al qual s'ha de copiar l'arxiu *passwd*. L'arxiu es copia al directori en el qual que ens trobem actualment (*playground*).

A *playground*, *ls* mostra els directoris *dir1*, *dir2* i l'arxiu *passwd* que acabem de copiar.

Podem utilitzar l'opció *-l* amb *ls* per ampliar la informació.

```
p@p-OptiPlex-GX620:~/playground$ ls -l
total 20
drwxrwxr-x 2 p p 4096 ene 22 12:30 dir1
drwxrwxr-x 2 p p 4096 ene 22 12:30 dir2
-rw-r--r-- 1 p p 2755 ene 22 12:40 passwd
p@p-OptiPlex-GX620:~/playground$
```

Ara tornem a copiar l'arxiu *passwd*, utilitzant l'opció *-v*.

```
p@p-OptiPlex-GX620:~/playground$ cp -v /etc/passwd .
'/etc/passwd' -> './passwd'
p@p-OptiPlex-GX620:~/playground$
```

El comando *cp* ha tornat a copiar l'arxiu, però aquesta vegada ha mostrat un missatge indicant el què fa. Observa, que va sobre escriure l'arxiu *passwd* existent, sense mostrar cap missatge d'avís.

Perquè *cp* ens avisi abans de sobre escriure *passwd*, utilitzarem l'opció *-i* (interactive).

```
p@p-OptiPlex-GX620:~/playground$ cp -iv /etc/passwd .
cp: overwrite './passwd'? y
'/etc/passwd' -> './passwd'
p@p-OptiPlex-GX620:~/playground$ cp -iv /etc/passwd .
cp: overwrite './passwd'? n
p@p-OptiPlex-GX620:~/playground$ █
```

Com es veu a la imatge, ara, el comando *cp*, abans de sobre escriure *passwd* demana permís. Contestant *y* (yes), sobre escriu *passwd*. Contestant *n* (no), no el sobre escriu, deixant-ho sense fer canvis.

Movent i reanomenant arxius

El nom *passwd* no ens agrada massa, així que el canviarem per *divertit*.

```
p@p-OptiPlex-GX620:~/playground$ mv passwd divertit
p@p-OptiPlex-GX620:~/playground$ ls
dir1 dir2 divertit
p@p-OptiPlex-GX620:~/playground$ █
```

Ara mourem *divertit* als directoris *dir1* i *dir2*, per, finalment, tornar a deixar-ho en *playground*.

Mou *divertit* a *dir1*.

```
p@p-OptiPlex-GX620:~/playground$ mv divertit ./dir1
p@p-OptiPlex-GX620:~/playground$ ls ./dir1
divertit
p@p-OptiPlex-GX620:~/playground$ █
```

Ara mou *divertit* de *dir1* a *dir2*.

```
p@p-OptiPlex-GX620:~/playground$ mv ./dir1/divertit ./dir2
p@p-OptiPlex-GX620:~/playground$ ls ./dir2
divertit
p@p-OptiPlex-GX620:~/playground$ █
```

Finalment tornem *divertit* a *playground* amb:

```
p@p-OptiPlex-GX620:~/playground$ mv ./dir2/divertit .
p@p-OptiPlex-GX620:~/playground$ ls
dir1 dir2 divertit
p@p-OptiPlex-GX620:~/playground$ █
```

A continuació provem moure directoris. Començarem movent l'arxiu *divertit* al directori *dir1*.

```
p@p-OptiPlex-GX620:~/playground$ mv divertit ./dir1
p@p-OptiPlex-GX620:~/playground$ ls ./dir1
divertit
p@p-OptiPlex-GX620:~/playground$
```

Ara movem *dir1* a *dir2* i observem el resultat amb *ls*.

```
p@p-OptiPlex-GX620:~/playground$ mv dir1 dir2
p@p-OptiPlex-GX620:~/playground$ ls -l ./dir2
total 4
drwxrwxr-x 2 p p 4096 ene 22 13:36 dir1
p@p-OptiPlex-GX620:~/playground$ ls -l ./dir2/dir1
total 12
-rw-r--r-- 1 p p 2755 ene 22 13:06 divertit
p@p-OptiPlex-GX620:~/playground$
```

Observa, que com *dir2* ja existia, *mv* ha mogut *dir1* a *dir2*, és a dir, *dir1* ja no existeix com a subdirectori de *playground*, sinó que ha passat a ser subdirectori de *dir2*. Si *dir2* no hagués existit, *mv* hauria canviat el nom de *dir1* per *dir2* i *dir 2* seria subdirectori de *playground*. En aquest darrer cas, no existiria cap directori amb el nom *dir1*.

Finalment, deixem tot com estava.

```
p@p-OptiPlex-GX620:~/playground$ mv ./dir2/dir1 .
p@p-OptiPlex-GX620:~/playground$ ls ./dir1
divertit
p@p-OptiPlex-GX620:~/playground$ mv ./dir1/divertit .
p@p-OptiPlex-GX620:~/playground$ ls
dir1 dir2 divertit
p@p-OptiPlex-GX620:~/playground$
```

Creant enllaços fixos

Ara provarem crear alguns enllaços fixos.

```
p@p-OptiPlex-GX620:~/playground$ ln divertit mola
p@p-OptiPlex-GX620:~/playground$ ln divertit ./dir1/mola_mucho
p@p-OptiPlex-GX620:~/playground$ ln divertit ./dir2/mola_poco
p@p-OptiPlex-GX620:~/playground$
```

Hem creat per l'arxiu *divertit* tres noms d'arxiu addicionals, *mola*, *mola_mucho* i *mola_poco*.

Fem un llistat del directori playground.

```
p@p-OptiPlex-GX620:~/playground$ ls -l
total 32
drwxrwxr-x 2 p p 4096 ene 22 18:29 dir1
drwxrwxr-x 2 p p 4096 ene 22 18:29 dir2
-rw-r--r-- 4 p p 2755 ene 22 16:50 divertit
-rw-r--r-- 4 p p 2755 ene 22 16:50 mola
p@p-OptiPlex-GX620:~/playground$
```

Si us fixeu, el nombre que indica els enllaços físics de *divertit* i *mola* és 4.

Recorda que un arxiu ha de tenir com a mínim un enllaç, el nom d'arxiu que es crea amb l'arxiu.

Com podem saber que *divertit* i *mola* es refereixen al mateix arxiu?

El resultat de `ls -l` ens mostra que tenen el mateix nombre de bytes, 2755, però això només és un indicatiu, que no assegura que realment siguin el mateix arxiu.

Per contestar a la pregunta de la identitat dels arxius *divertit* i *mola*, haurem d'aprofundir una mica més en l'assignació de noms a un arxiu.

Imagineu que un arxiu es compon de dues parts, una part que conté les dades de l'arxiu, per exemple text o codi binari, i una altra que conté el nom de l'arxiu. Quan creem un enllaç físic, el que estem fent és ampliar la part de l'arxiu on s'especifiquen els noms, que corresponen tots a la mateixa part de dades. L'inode és una cadena de blocs del dispositiu d'emmagatzematge que té la funció d'adreça. Dit amb altres paraules, l'inode és un nombre que ubica la part de dades d'un arxiu al dispositiu

d'emmagatzematge. Cada enllaç físic (nom) que assignem a un arxiu està enllaçat al mateix inode que ubica l'arxiu.

Si utilitzem ls amb l'opció -l, veurem l'inode de cada arxiu.

```
p@p-OptiPlex-GX620:~/playground$ ls -li
total 32
1365397 drwxrwxr-x 2 p p 4096 ene 22 18:29 dir1
1365396 drwxrwxr-x 2 p p 4096 ene 22 18:29 dir2
1350677 -rw-r--r-- 4 p p 2755 ene 22 16:50 divertit
1350677 -rw-r--r-- 4 p p 2755 ene 22 16:50 mola
p@p-OptiPlex-GX620:~/playground$
```

Com veieu podem comprovar que *divertit* i *mola* són noms diferents del mateix arxiu, ja que la ubicació al dispositiu d'emmagatzematge és la mateixa. L'inode de tots dos és 1350677.

Podem comprovar si els arxius *mola_mucho* i *mola_poco* als directoris *dir1* i *dir2* respectivament també són el mateix arxiu.

```
p@p-OptiPlex-GX620:~/playground$ ls -li ./dir1
total 12
1350677 -rw-r--r-- 4 p p 2755 ene 22 16:50 mola_mucho
p@p-OptiPlex-GX620:~/playground$ ls -li ./dir2
total 12
1350677 -rw-r--r-- 4 p p 2755 ene 22 16:50 mola_poco
p@p-OptiPlex-GX620:~/playground$
```


L'inode o aquell misteriós desconegut (tan atractiu)

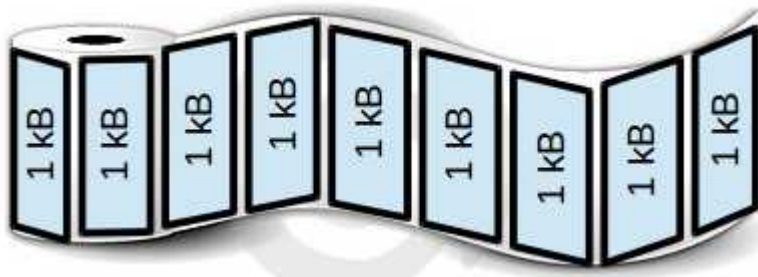
Siguem sincers, això de l'inode, encara que sigui anglès sona a xino. Llavors, caldrà aprendre xinès, només serà una mica.

Imaginem una memòria USB (pen drive). Imaginem-la com un full de paper immensament llarg, una mena de rotllo de paper.

Imaginem, aquesta tira de paper desenrotllada. La utilitzarem per desar-hi tres arxius de text, plain text, és a dir, text pur sense formatar, codificat amb [ASCII](#). L'ASCII és un estàndard de codificació que assigna a cada caràcter un nombre d'un byte. Per exemple, a la lletra **a** li correspon en nombre binari 01000001.

Suposem que la memòria USB d'aquest exemple sigui de
1 GB = 1000 MB = 1000 000 kB = 1000 000 000 byte

Sabem, que per poder desar informació en un dispositiu d'emmagatzematge l'hem de formatar, donar-li una estructura. Llavors el primer que farem és dividir la memòria en blocs de 1000 bytes cadascun. Això ens dóna 1000 000 de blocs de 1 kB, en el nostre rotllo de paper.



Podríem retallar tots els blocs d'1 kB i fer-ne un llibre, de 1000 000 de pàgines. Cada pàgina podria contenir fins a 1 kB d'informació, és a dir fins a 1000 caràcters ASCII, ja que cada caràcter ASCII es codifica amb 1 byte.

Bé, ja que està organitzat el nostre llibre (formatada la nostra memòria USB), podem començar a escriure capítols (desar-hi arxius).

Comencem pel **capítol 1 «Les meves vacances 2015»**, sense entrar en detalls escabrosos, diguem simplement que hem necessitat 5500 caràcters per contar les nostres vacances, això serien 5 pàgines i mitja en el nostre llibre, o 5500 bytes que hem de desar a l'arxiu **vacances_2015.txt** i que ocuparia 5 blocs i mig.

Atenció que ara ve la part important.

En el nostre llibre feim **dues anotacions**, una a l'**índex on posem capítol 1 «Les meves vacances 2015», pàg. 1** i una altra a les **pàgines 1, 2, 3, 4, 5 i 6** on escrivim el text.

El capítol 2 «Perquè no tornaré a fer un creuer» té una extensió de 10 200 caràcters que són 10 pàgines i una quinta part de pàgina. El capítol 2 començarà en la pàgina 7 i terminarà en la pàgina 17. A l'índex anotarem capítol 2 «Perquè no tornaré a fer un creuer», pàg. 7 i a les pàgines 7 a 17 escriurem el text.

El capítol 3 «Iniciació al càmping» té una extensió de 3000 caràcters que són 3 pàgines. El capítol 3 començarà en la pàgina 18 i terminarà en la pàgina 20. A l'índex anotarem capítol 3 «Iniciació al càmping», pàg. 18 i a les pàgines 18 a 20 escriurem el text.

En la memòria USB els capítols 1, 2 i 3 corresponen als arxius vacances_2015.txt, creuer.txt i camping.txt.

Les entrades a l'índex del llibre equivalen als enllaços físics (noms d'arxiu) i el nombre de pàgina a l'inode en la memòria USB.

Vacances_2015.txt s'ubica en l'inode 0000001, creuer.txt en 0000007 i camping.txt en 0000018. Tenim la possibilitat de donar més noms al mateix text, creant enllaços fixos addicionals. Per exemple, a les dades del capítol 1, les podríem anomenar hotel_flotant.txt. A l'índex, el text del capítol 1 tindria dues entrades «Les meves vacances 2015» i «L'hotel flotant» que fan referència a la pàgina 1 del llibre.

En la memòria USB tindríem els noms d'arxiu vacances_2015.txt i hotel_flotant.txt que tenen el mateix inode 0000001 ja que fan referència a les mateixes dades dels blocs 0000001 a 0000006.

Índex – capítols llibre		Índex – arxius USB	
	pág.		inode
Les meves vacances 2015		vacances_2015.txt	
L'hotel flotant	1	hotel_flotant.txt	0000001
Perquè no tornaré a fer un creuer	7	creuer.txt	0000007
Iniciació al càmping	18	camping.txt	0000018

Creant enllaços simbòlics

Els enllaços simbòlics són necessaris per compensar dos desavantatges dels enllaços físics: Els enllaços físics no poden indexar arxius fora de la partició on es troba l'enllaç físic, i només poden indexar arxius, no directoris. L'enllaç simbòlic és un tipus d'arxiu especial, que conté un marcador de text cap a una arxiu o un directori. Tornant a l'exemple anterior de l'índex d'un llibre, els enllaços físics són les entrades de l'índex del llibre i, òbviament, només poden fer referència als capítols del llibre en el qual es troben. Els enllaços simbòlics podrien ser una combinació de les [signatures](#) dels llibres d'una biblioteca amb cada un dels capítols dels llibres de la biblioteca. Els enllaços simbòlics es creen de forma similar als físics, afegint l'opció `-s`.

```
p@p-OptiPlex-GX620:~/playground$ ln -s divertit divertit_sim
p@p-OptiPlex-GX620:~/playground$ ln -s ../divertit ./dir1/mola_mucho_sim
p@p-OptiPlex-GX620:~/playground$ ln -s ../divertit ./dir2/mola_poco_sim
p@p-OptiPlex-GX620:~/playground$ ls -l
total 36
drwxrwxr-x 2 p p 4096 ene 23 11:31 dir1
drwxrwxr-x 2 p p 4096 ene 23 11:31 dir2
-rw-r--r-- 4 p p 2755 ene 22 16:50 divertit
lrwxrwxrwx 1 p p 8 ene 23 11:29 divertit_sim -> divertit
-rw-r--r-- 4 p p 2755 ene 22 16:50 mola
p@p-OptiPlex-GX620:~/playground$
```

En l'exemple hem creat l'enllaç simbòlic *divertit_sim* dirigit a l'enllaç físic *divertit*. Tots dos enllaços es troben al directori *playground*.

A continuació hem creat l'enllaç simbòlic *mola_mucho_sim*, ubicat a *dir1*, dirigit a l'enllaç físic *divertit* a *playground*.

En la tercera línia hem creat l'enllaç simbòlic *mola_poco_sim*, ubicat a *dir2*, dirigit a l'enllaç físic *divertit* a *playground*.

Com es veu, quan fem un enllaç simbòlic, hem d'indicar la ruta relativa al directori on es troba l'enllaç simbòlic, cap al directori on es troba l'element amb el qual volem enllaçar.

ls -l mostra que al directori *playground* hi ha l'enllaç simbòlic *divertit_sim* que varem crear en la primera línia. Els enllaços simbòlics *mola_mucho_sim* i *mola_poco_sim*, a *divertit*, es troben en els directoris *dir1* i *dir2* respectivament.

```
p@p-OptiPlex-GX620:~/playground$ ls -l ./dir1
total 16
-rw-r--r-- 4 p p 2755 ene 22 16:50 mola_mucho
lrwxrwxrwx 1 p p  11 ene 23 11:31 mola_mucho_sim -> ../divertit
p@p-OptiPlex-GX620:~/playground$ ls -l ./dir2
total 16
-rw-r--r-- 4 p p 2755 ene 22 16:50 mola_poco
lrwxrwxrwx 1 p p  11 ene 23 11:31 mola_poco_sim -> ../divertit
p@p-OptiPlex-GX620:~/playground$
```

També es veu que els enllaços simbòlics estan identificats amb una **l** com a primera lletra. Tant *mola_mucho_sim* com *mola_poco_sim* estan dirigits cap a *divertit*, com indica la ruta relativa. Fixat, que la mida dels enllaços simbòlics és de 11 byte, el nombre de caràcters del text «../divertit», metre que la mida de l'arxiu *divertit* és de 2755 byte.

Per crear un enllaç simbòlic, també es pot utilitzar un nom de ruta absolut.

```
p@p-OptiPlex-GX620:~/playground$ ln -s /home/p/playground/divertit ./dir1/mola_algo_sim
p@p-OptiPlex-GX620:~/playground$ ls -l ./dir1
total 20
lrwxrwxrwx 1 p p  27 ene 23 11:59 mola_algo_sim -> /home/p/playground/divertit
-rw-r--r-- 4 p p 2755 ene 22 16:50 mola_mucho
lrwxrwxrwx 1 p p  11 ene 23 11:31 mola_mucho_sim -> ../divertit
p@p-OptiPlex-GX620:~/playground$
```

Creant enllaços simbòlics, és convenient utilitzar noms de ruta relatius, perquè permet reanomenar o moure un directori amb enllaços simbòlics, sense rompre'ls. Reanomenem el directori *playground* a *playground_nou* i mirem que passa amb els enllaços.

```
p@p-OptiPlex-GX620:~/playground$ cd ..
p@p-OptiPlex-GX620:~$ mv ./playground/ ./playground_nou
p@p-OptiPlex-GX620:~$ cd ./playground_nou/
p@p-OptiPlex-GX620:~/playground_nou$ ls -l ./dir1
total 20
lrwxrwxrwx 1 p p  27 ene 23 11:59 mola_algo_sim -> /home/p/playground/divertit
-rw-r--r-- 4 p p 2755 ene 22 16:50 mola_mucho
lrwxrwxrwx 1 p p  11 ene 23 11:31 mola_mucho_sim -> ../divertit
p@p-OptiPlex-GX620:~/playground_nou$
```

mola_algo_sim s'ha romput, perquè en reanomenar *playground* a *playground_nou*, la ruta */home/p/playground/divertit*, que dirigeix l'enllaç simbòlic *mola_algo_sim* cap a *divertit*, ja no existeix. Ara el nom de ruta correcte seria */home/p/playground_nou/divertit*.

Un enllaç simbòlic pot estar dirigit cap a un directori.

```
p@p-OptiPlex-GX620:~/playground_nou$ ln -s ./dir1 dir1_sim
p@p-OptiPlex-GX620:~/playground_nou$ ls -l
total 40
drwxrwxr-x 2 p p 4096 ene 23 12:26 dir1
lrwxrwxrwx 1 p p   6 ene 23 12:31 dir1_sim -> ./dir1
drwxrwxr-x 2 p p 4096 ene 23 11:31 dir2
-rw-r--r-- 4 p p 2755 ene 22 16:50 divertit
lrwxrwxrwx 1 p p   8 ene 23 11:29 divertit_sim -> divertit
-rw-r--r-- 4 p p 2755 ene 22 16:50 mola
p@p-OptiPlex-GX620:~/playground_nou$
```

En aquest cas hem dirigit *dir1_sim* cap a *dir1*, tots dos subdirectoris de *playground*.

Eliminant arxius i directoris

Es va explicar anteriorment, que el comando *rm* s'utilitza per eliminar elements.

Ara l'utilitzarem per fer dissabte al directori *playground_nou*. Començarem eliminant un dels enllaços fixos.

```
p@p-OptiPlex-GX620:~/playground_nou$ ls
dir1 dir1_sim dir2 divertit divertit_sim mola
p@p-OptiPlex-GX620:~/playground_nou$ rm mola
p@p-OptiPlex-GX620:~/playground_nou$ ls -l
total 28
drwxrwxr-x 2 p p 4096 ene 23 12:26 dir1
lrwxrwxrwx 1 p p 6 ene 23 12:31 dir1_sim -> ./dir1
drwxrwxr-x 2 p p 4096 ene 23 11:31 dir2
-rw-r--r-- 3 p p 2755 ene 22 16:50 divertit
lrwxrwxrwx 1 p p 8 ene 23 11:29 divertit_sim -> divertit
p@p-OptiPlex-GX620:~/playground_nou$
```

Ha funcionat, l'enllaç físic *mola* ha estat eliminat i el nombre d'enllaços físics a l'arxiu ha baixat de 4 a 3, com es mostra a la línia de *divertit*.

Ara eliminarem l'enllaç físic *divertit*, afegint l'opció *-i*.

```
p@p-OptiPlex-GX620:~/playground_nou$ rm -i divertit
rm: remove regular file 'divertit'? y
p@p-OptiPlex-GX620:~/playground_nou$ ls
dir1 dir1_sim dir2 divertit_sim
p@p-OptiPlex-GX620:~/playground_nou$
```

Veieu que *divertit_sim* s'ha tornat vermell? És clar, *divertit_sim* és un enllaç simbòlic dirigit cap a *divertit*, però com *divertit* ha estat eliminat, l'enllaç s'ha romput. Un enllaç simbòlic romput més que problemàtic és molest. Provem llegir el contingut de *divertit_sim* amb *less*.

```
p@p-OptiPlex-GX620:~/playground_nou$ less divertit_sim
divertit_sim: No such file or directory
p@p-OptiPlex-GX620:~/playground_nou$
```

El resultat és l'avís de que l'arxiu o directori no existeix.

Netegem, eliminant tots els enllaços simbòlics.

```
p@p-OptiPlex-GX620:~/playground_nou$ rm ./dir1_sim ./divertit_sim
p@p-OptiPlex-GX620:~/playground_nou$ ls
dir1 dir2
```

Una cosa que hem de recordar respecte als enllaços simbòlics és, que la majoria dels comandos executats amb un enllaç simbòlic com a argument, actuen sobre l'element al qual senyala l'enllaç, no sobre l'enllaç simbòlic. *rm* és una excepció d'aquesta regla. Quan s'elimina un enllaç simbòlic, l'element al qual senyala no desapareix. Finalment, eliminarem *playground_nou*. Utilitzarem l'opció **-r** per eliminar tot el contingut, inclòs el dels subdirectoris.

```
p@p-OptiPlex-GX620:~/playground_nou$ rm -r ../playground_nou/
p@p-OptiPlex-GX620:~/playground_nou$ ls
p@p-OptiPlex-GX620:~/playground_nou$ cd ..
p@p-OptiPlex-GX620:~$ ls
2016-01-10-180249_1280x1024_scrot.png  dwhelper
carp                                     examples.desktop
carp_act                                Music
carp_act.zip                            Pictures
carp.zip                                 PlayOnLinux's virtual drives
deja-dup                                 Public
dell                                     swap~
Desktop                                  Templates
Documents                                Videos
Downloads
p@p-OptiPlex-GX620:~$ █
```

Cercant arxius amb *find*

Sovint necessitarem trobar un arxiu i això pot ser complicat perquè el sistema d'arxius és molt gran i seria massa feina fer un llistat de cada subdirectori cercant l'arxiu.

El comando ***find*** ens permet cercar un arxiu de forma còmoda.

```
find -name nom_darxiu
```

Utilitzant l'opció `-name`, `find` cercarà per `nom_darxiu.`, fent diferència entre lletres majúscules i minúscules. Si volem que la recerca sigui independent de majúscules i minúscules, utilitzarem l'opció `-iname`.

Com a exemple, cerquem a la carpeta [*carp*](#) l'arxiu *eVa.txt*.

```
p@p-OptiPlex-GX620:/media/p/PPOSADA/sem_18_24/bat/linux/carp$ find -name eva*
p@p-OptiPlex-GX620:/media/p/PPOSADA/sem_18_24/bat/linux/carp$ find -iname eva*
./dijous/institut/primer_hora/eVa.txt
p@p-OptiPlex-GX620:/media/p/PPOSADA/sem_18_24/bat/linux/carp$
```

Com es veu, la recerca de la primera línia no va donar resultat, perquè la `v` de `eVa.txt` està en minúscula. Utilitzant l'opció `-iname`, `find` sí troba `eVa.txt`.

Exercici 1.3.4-4

Descarrega la carpeta [*carp*](#) al directori home. Obre un Terminal i accedeix al directori home. Llista els continguts d'*home* i comprova que s'hi trobi *carp*.

- a) Al directori *carp*, crea el subdirectori *el_teu_nom*.
- b) Desplaçat cap al directori amb l'arxiu *balada*. Executa un comando que mostri la ruta absoluta cap a l'arxiu *balada*. Fes una captura de pantalla i anomena-la 1344b.
- c) Fes una captura de pantalla del contingut de l'arxiu *balada* i anomena-la 1344c.
- d) Copia l'arxiu *balada* a *carp/misteris/aleluya*, reanomenat l'arxiu com a *rafael_alberti.txt*. Mostra els arxius del directori *aleluya* i fes-ne una captura de pantalla anomenada 1344d.
- e) Elimina el directori *carp/musica/guitarra*. Fes una captura de pantalla i anomena-la 1344e.
- f) Copia el directori *carp/familia* a *carp/el_teu_nom*. Mostra el contingut d'*el_teu_nom* i fes una captura de pantalla anomenada 1344f.
- g) Ves al directori *el_teu_nom* i canvia de nom el directori *familia* a *familiars*.

Autocomplete

Search history (p.74)

What exactly are commands (p. 40)

Identifying commands (p. 40)

type, which

Creating your own command with alias (p. 46)

9 Permissions (p.78)

Creant diferents user accounts

Com instal·lar aplicacions amb Software Center

Activitat 2

Donar nom a USB