

Treballar amb particions, sistemes de fitxers i punts de muntatge

lsblk Mostra la llista de dispositius de bloc reconeguts al sistema. En concret, per defecte les columnes que mostra són: "name", "maj/min", "removable" (o no), "size", "type" (disk/part) i punt de muntatge

-f : Mostra columnes diferents a les per defecte. En concret: "name", "sist. fitxers", "label", "uuid", espai disponible, %espai utilitzat i punt de muntatge

-o col1,col2... +colaafegir,... :Especifica les columnes desitjades (*lsblk -help* mostra llista completa)

-x col1 : Ordena per la columna indicada

-p : Mostra la ruta absoluta dels dispositius (normalment "/dev/xxx")

-P : La sortida la realitza en format "udev", en comptes de per columnes

NOTA: Existeix una comanda similar a *lsblk* anomenada *blkid* però a la seva pròpia pàgina del manual ja s'indica que és preferible utilitzar sempre la primera (o, si calgués, *findmnt*) degut a diversos avantatges com ara que no cal ser "root" per executar-la i que la seva sortida és més completa i ordenada

findmnt Mostra la llista dels punts de muntatge al sistema. En concret, per defecte mostra:

- *La ruta de cada punt de muntatge reconegut al sistema
- *La ruta del dispositiu corresponent -que pot estar sota "/dev" però no té per què-
- *El sistema de fitxers en què està formatat aquest dispositiu
- *Les opcions de muntatge utilitzades.

-T /ruta/PuntMuntatge : Mostra només la informació relacionada amb el punt de muntatge indicat

-S /dev/disp1 : Mostra només la informació relacionada amb la partició indicada

NOTA: En lloc de "/dev/disp1", el dispositiu es pot indicar d'altres maneres alternatives (que estudiarem més endavant), com: LABEL=etiqueta, UUID=uuid, PARTLABEL=etiqueta i PARTUUID=uuid

-t tipusfitxer : Mostra només els punts de muntatge que tinguin el sistema de fitxers indicat

-s : Mostra només els dispositius muntats mitjançant "/etc/fstab". Amb -m és "mstab"

-O opcio,altra,... : Mostra només els punts de muntatge que tenen l'opció especificada

-o col1,col2... +colaafegir: Especifica les columnes desitjades (*findmnt --help* mostra llista completa)

-D : Imita la sortida de *df* Equival a -o SOURCE,FSTYPE,SIZE,USED,AVAIL,USE%,TARGET

-e : Mostra les rutes de dispositiu "/dev/..." en comptes de LABELs i UUIDs

-P : La sortida la realitza en format "udev", en comptes de per columnes

-u : No talla les línies massa llargues

NOTA: La comanda *mount* executada sense paràmetres també mostra una informació similar a *findmnt*, però no d'una forma tan clara. En tot cas, ambdues comandes obtenen la informació que mostren del fitxer **"/proc/mounts"** (el qual es pot consultar directament així, *cat /proc/mounts*), gestionat directament pel kernel

fdisk -l [/dev/disp] Mostra informació (mida i estructura) de tots els dispositius d'emmagatzematge detectats -o de l'indicat- a més de la seva taula de particions (on s'especifica la mida de cadascuna d'elles, el seu tipus, etc).

NOTA: Una altra manera de saber el tipus de la taula de particions d'un determinat dispositiu és *file -s /dev/disp*

fdisk /dev/disp Permet manipular particions del disc especificat de forma interactiva, dins d'un shell propi. Per saber quines opcions es poden fer servir en aquest shell, pots pulsar *m*. En tot cas, les opcions més bàsiques són:

g : crear la taula de particions de tipus **GPT**

n : crear nova partició (i a partir d'aquí assignar els seus nombre, primer sector i darrer -o mida-)

d : eliminar partició que es digui tot seguit

p : visualitza estat actual de la taula de particions (com *fdisk -l*). També està l'opció *i*

w : surt desant els canvis

q : surt sense gravar

NOTA: La comanda *fdisk* ve acompanyada de dues comandes amb la mateixa funcionalitat però formes diferents: *sfdisk* i *cfdisk* La primera està més orientada a scripts i la segona ofereix una interfície basada en Ncurses

`parted -l [/dev/disp]` Alternativa GNU a *fdisk* (que ve amb el paquet "util-linux"). Mostra informació (mida i estructura) de tots els dispositius d'emmagatzematge detectats -o de l'indicat a més de la seva taula de particions (on s'especifica la mida de cadascuna d'elles, el seu tipus, etc).

`parted [/dev/disp]` Alternativa GNU a *fdisk* (que ve amb el paquet "util-linux"). Permet manipular particions del disc especificat de forma interactiva, dins d'un shell propi. Per saber quines subcomandes es poden fer servir en aquest shell, pots escriure *help*. En tot cas, les subcomandes més bàsiques són:

`select /dev/disp` : Selecciona el dispositiu sobre el qual treballar (si no s'ha indicat com a paràmetre)
`mklabel gpt` : crea la taula de particions de tipus GPT
`mkpart nom n°MBiniciFrom0 n°MBfinalFrom0` : crea partició
`rm n°` : Elimina de forma interactiva la partició indicada
`print` : mostra la llista actual de particions del disc
`quit`: surt

`mkfs.{ext4|vfat|xfs|...} /dev/disp1` Formateja amb el sistema de fitxers indicat la partició especificada

`-L etiqueta` : Estableix una etiqueta a una partició ExtX o XFS
`-n etiqueta` : Estableix una etiqueta a una partició FAT32
NOTA: Sobre la utilitat de les etiquetes (i els UUIDs) en parlarem més endavant en aquest document
`-U {valoamà|random|time|clear}` : Estableix el UUID d'una partició ExtX, el qual pot ser indicat a mà (com si fos una etiqueta), o pot ser generat aleatòriament, o pot ser generat aleatòriament tenint en compte la data actual o pot ser esborrat. Si es vol establir a mà, una comanda còmoda per generar UUIDs és *uuidgen*
`-m uuid=valoramà` : Estableix el UUID d'una partició XFS a mà. Si no s'indica res, es genera aleatori
`-b tamany` : Estableix la mida (en bytes) dels blocs per aquesta partició ExtX o XFS
`-S tamany` : Estableix la mida (en bytes) dels blocs per aquesta partició FAT32
`-c` : Fa una comprovació prèvia de blocs a particions ExtX i FAT32
`-v` : Mode verbós (a particions ExtX i FAT32)

NOTA: Per saber el tamany concret d'un bloc d'una determinada partició ja formatada, es pot fer: *blockdev --getbsz /dev/disp1* També es pot veure a *fdisk -l* sota el nom de "physical sector size".

NOTA: En el cas dels sistemes de fitxers ExtX, els valors per defecte, si no es canvien per algun paràmetre, els pren *mkfs* de l'arxiu de configuració `"/etc/mke2fs.conf"`

`fsck.{ext4|vfat|...} /dev/disp1` Fa un xequex del sistema de fitxers de la partició especificada. En concret, comprova que:

- Un bloc no pertanyi a diferents arxius (és a dir, inodes)
- No hi hagin blocs lliures que apareguin com ocupats
- No hi hagin blocs ocupats que apareguin com a lliures
- El comptador d'enllaços durs no sigui erroni
- No hi hagi clústers perduts (es guarden a "/lost+found")
- No hi hagi inodes ilocalitzables
- No hi hagi dades inconsistents a la taula d'inodes

`-p` : repara automàticament, sense fer preguntes. Incompatible amb les opcions `-y` i `-n`
`-y` : respon que sí automàticament a totes les preguntes
`-n` : no fa cap canvi al sistema de fitxers (només informa, no repara)
`-V` : mode verbós
`-N` : Diu quin és el sistema de fitxers de la partició indicada.

NOTA: Una manera alternativa de conèixer el sistema de fitxers d'una partició (a més de també el seu UUID i el seu LABEL) és fent *file -s /dev/disp1*

`-c` : comprova si hi ha blocs erronis executant internament la comanda *badblocks* (només per Ext4)
`-f` : força la comprovació encara que el sistema estigui marcat com a net (només per Ext4)
`-D` : optimitza l'accés al contingut dels directoris, reindexant-lo i reordenant-lo (només per Ext4)

NOTA: La comanda relativa a particions FAT32 es troba dins del paquet "**dosfstools**", instal·lat per defecte. La comanda relativa a particions ExtX es troba dins del paquet "**e2fsprogs**", instal·lat per defecte també. Per XFS hi ha altres comandes diferents (*xfs_check* i *xfs_repair*, i també *xfs_growfs* ; la comanda *fsck.xfs* existeix però no fa res, literalment); que es troben dins del paquet "**xfsprogs**", instal·lat per defecte només a Fedora. Per NTFS hi ha altres comandes diferents també, les quals es troben al paquet "**ntfsprogs**", que no sol venir instal·lat de sèrie.

NOTA: La comanda *fsck* s'executa sempre en iniciar el sistema si aquest es va apagar sense desmuntar el sistema d'arxius.

mount [-t sistfitx] /dev/disp1 /ruta/punt Munta la partició especificada al punt de muntatge especificat, indicant quin és el sistema d'arxiu (el "format") d'aquest dispositiu. Si això últim es desconeix, es pot no escriure el paràmetre *-t* i deixar que Linux ho esbrini per sí sol. Alguns dels sistemes de fitxers que es poden indicar són: "**vfat**", "**ext4**", "**xfs**", "**jfs**", "**btrfs**", "**ntfs**", "**swap**" (per particions swap), "**nfs**" (per punts de muntatges remots), "**cifs**" (per punts de muntatge remots Windows), "**iso9660**" (per CDs), "**tmpfs**" (resideix en memòria), "**procfs**" (conté informació relativa a la configuració del kernel i els processos actuals) o "**sysfs**" (conté informació relativa als dispositius), entre altres. En comptes d'indicar la ruta del dispositiu es pot escriure *-L label* o *-U uuid*

mount -B /ruta/carpeta /ruta/punt "Munta" la carpeta especificada al punt de muntatge especificat. D'aquesta manera, es podrà accedir al mateix contingut des de dos punts d'entrada diferents (la ruta de la carpeta en sí i la del seu punt de muntatge)

-o ... : Serveix per indicar diferents opcions de muntatge (separades per comes), com ara "**ro**" ("**rw**" és per defecte), "**sync**" (per no fer servir memòria "dirty"), "**loop**" (per muntar isos en comptes de dispositius i així accedir al seu interior a través del punt de muntatge especificat; per exemple: *mount -t iso9660 -o loop arxiu.iso /ruta/punt*), etc.

NOTA: La comanda *mount -a* recarrega la informació continguda a l'arxiu */etc/fstab* sense que calgui reiniciar la màquina: molt útil quan fem edicions d'aquest arxiu i volem obtenir immediatament el resultat d'aquesta edició

NOTA: L'arxiu */etc/mstab* llista, amb un format similar a "fstab", els sistemes de fitxers muntats en aquest moment i és automàticament editat cada cop que s'utilitza la comanda *mount*.

systemd-mount [-t sistfitx] /dev/disp1 /ruta/punt Comanda alternativa a *mount* proporcionada per Systemd (i, per tant, integrada de forma molt forta en aquest subsistema). A més de les opcions *-t* i *-o*, idèntiques a les de la comanda *mount*, aquesta comanda aporta moltes més opcions (consultables a la seva pàgina del manual)

umount /ruta/punt Desmunta el punt de muntatge indicat

-l : El desmuntatge es farà quan tots els recursos usats pel dispositiu siguin alliberats
-R : Desmunta tots els possibles subpunts de muntatge que estiguin per sota de l'especificat
-A /dev/disp1 : Desmunta tots els punts de muntatge del dispositiu especificat (útil quan hi ha més d'1)

systemd-umount /ruta/punt Comanda alternativa a *umount* proporcionada per Systemd

df -h [/dev/disp1] Mostra les particions, l'espai total, disponible i ocupat i els seus punts de muntatge. Es pot indicar una carpeta normal com a paràmetre en comptes d'un dispositiu: en aquest cas es mostrarà la informació de la partició on es trobi

-T : Mostra una columna extra amb el format dels sistemes de fitxers
--output nomCol1,nomCol2 ... : Serveix per personalitzar les columnes a mostrar. Els noms de columnes vàlids són: '*source*', '*fstype*', '*size*'/'*itotal*', '*i/used*', '*i/avail*', '*i/pcent*', '*file*' i '*target*'
-i : En comptes d'informar sobre l'espai en disc, informa sobre l'ús dels inodes
-t sistFich : Només mostra la informació de particions formatejades amb el sistema de fitxers indicat
-x sistFich : Mostra la informació de les particions NO " " " " (per ex: *-x tmpfs*)
-l : Només mostra els sistemes de fitxers locals (no Cifs, NFS, etc)
-a : Inclou a la llista els pseudo-sistemes de fitxers

du -h [/ruta/carpeta] Mostra l'espai ocupat per la carpeta especificada i per cadascun dels fitxers que conté. Si no s'indica cap carpeta, ho fa de la carpeta actual. Es poden indicar també comodins per fer subgrups de fitxers.
Un programa gràfic similar (però més versàtil) és el *baobab*

- s : Només mostra la mida total de la carpeta. Si s'empren subgrups amb comodins, millor usar -c
- x : No entra en directoris que pertanyin a un altre sistema de fitxers (és a dir, en punts de muntatge)
- b : Mostra les mides en bytes, sense sufixes
- a : Mostra tots els arxius de tots els subdirectoris (si no es posa, només es mostren aquests)
- d n° : Mostra tots els subdirectoris (o arxius amb -a) fins el nivell de profunditat indicat (0,1,2...)
- time : Mostra a més la data d'última modificació de cada ítem llistat
- t n° : Exclou de la llista el fitxers menors del tamany n°*K,M,G,T,P.
- t -n° : Exclou de la llista el fitxers majors del tamany n°*K,M,G,T,P.

Dispositius "loop":

Un dispositiu "loop" és un tipus de dispositiu (normalment, a Linux, anomenat `/dev/loop0`, `/dev/loop1`, etc) de tipus virtual (és a dir, que no existeix físicament) i que serveix per permetre l'accés, a través d'ell, a un fitxer prèviament associat com si aquest fos un dispositiu de bloc. És a dir, la "connexió" del dispositiu "loop" a un determinat fitxer fa que els programes que utilitzi l'usuari a partir de llavors puguin interaccionar amb aquest fitxer com si fos un dispositiu de bloc. Per tant, si aquest fitxer contingué al seu interior un sistema de fitxers, aquest podria ser muntat com si fos un disc.

losetup /dev/loop0 debian.raw : Associa un determinat dispositiu virtual (en aquest cas, `/dev/loop0`) al fitxer indicat per tal de reconèixer aquest darrer com un dispositiu de bloc.
Aquesta comanda cal executar-la com administrador.

NOTA: Es podria no indicar explícitament el nom del dispositiu "loop" a crear (i a la vegada saber quin nom li ha assignat llavors automàticament el kernel...el qual serà `/dev/loop0` si aquest no està en ús però pot ser un altre `-/dev/loop1`, `/dev/loop2`...el primer que estigui lliure-) si s'executa la comanda (com administrador) *losetup -f --show debian.raw* (aquest és el cas, de fet, quan s'empra l'opció "loop" a seques en la comanda *mount*, com de seguida veurem)

mount /dev/loop0p1 hola : Munta la partició indicada ("p1"), la qual ha d'existir ja en el dispositiu de bloc creat al pas anterior. Aquestes particions en principi són reconegudes automàticament però si no fos així, a la comanda *losetup* anterior caldria afegir llavors el paràmetre *-P* Aquesta comanda cal executar-la com administrador.

Normalment, però, els fitxers que es desitjen muntar no inclouen varies particions sinó només una; és a dir, solen estar formats al seu interior per un únic sistema de fitxers. En aquest cas, les dues comandes anteriors es podrien "col·lapsar" en només una comanda, la següent:

mount -o loop debian.raw hola : Munta el sistema de fitxers inclòs dins el fitxer indicat. De totes formes, moltes vegades el paràmetre *-o loop* (el qual també es podria indicar de forma més explícitament així, *-o loop=/dev/loop0*) no cal escriure'l ja que la comanda *mount* sol ser prou "intel·ligent" per detectar que el dispositiu a muntar en realitat és un fitxer i, per tant, fer els passos pertinents. Aquesta comanda cal executar-la com administrador.

losetup -d /dev/loop0 : Elimina el dispositiu "loop" indicat, prèviament creat (tot i que desmuntant el punt de muntatge associat ja hauria d'eliminar-se automàticament). Aquesta comanda cal executar-la com administrador.

losetup -D : Elimina tots els dispositius "loop" existents. Cal executar aquesta comanda com administrador.

losetup -l : Mostra la llista de dispositius "loop" actualment implementats (i, en tot cas, per veure tots els dispositius de bloc reconeguts, incloent els de tipus "loop", es pot fer *lsblk -f*)

EXERCICIS:

Realitza tots els exercicis següents en una màquina virtual UEFI que tingui el sistema Fedora instal·lat.

1.-Arrenca primer la màquina virtual i, un cop en marxa, atura-la de cop fent servir l'opció adient del quadre d'apagada del VirtualBox. Reinicia aquesta màquina de nou però amb una iso d'un Linux qualsevol que tinguis a mà inserida dins de la seva lectora de DVD virtual. Un cop dins del sistema Live, comprova amb *fsck* l'estat de la partició on estigui instal·lat el sistema al disc dur i corregeix-ne a la vegada els seus possibles errors lògics. Reinicia finalment la màquina ja sense la iso, per tornar a entrar al sistema ja reparat.

2.-a) Utilitza la comanda *findmnt* per conèixer quins dispositius (particions, CDs, etc) hi ha muntats al sistema, quins són els seus punts de muntatge, els sistemes de fitxers que contenen i les seves opcions de muntatge

b) ¿Quin paràmetre de *findmnt* faries servir per obtenir la mateixa informació però només per la partició `"/dev/sda1"`? ¿I només pel punt de muntatge `"/"`? ¿I només per les particions de tipus Ext4? ¿I per conèixer el tamany total, usat i lliure de les particions?

3.-a) Empra la comanda *lsblk* per conèixer quins dispositius de "bloc" (particions, CDs, etc) el teu sistema detecta, quins d'ells estan muntats (i on) i si són de només-lectura (o no). ¿Per a què serveixen els paràmetres *-f* i *-o* d'aquesta comanda (prova'ls)?

b) La comanda *lsblk* et pot servir per conèixer com anomena el kernel al llàpissos USB que tens endollats al sistema. ¿Quin paràmetre de *fdisk* et podria servir pel mateix?

La comanda *dmesg* mostra els missatges generats pel kernel des del darrer arranc. Entre aquests, es troben precisament els corresponents a la detecció de hardware, els quals se solen produir durant l'arranc però, en el cas d'endollar llàpissos USB o similars, es produeixen en el moment que s'endollen. Això es pot aprofitar per esbrinar també, a l'igual que s'ha fet a l'apartat anterior, el nom dels llàpissos USB recentment endollats: només cal ser ràpids i, un cop endollat, executar la comanda *dmesg* immediatament per veure, al final de tot, els missatges informatius corresponents.

c) Un cop llegit el paràgraf blau anterior, digues què fa la comanda *dmesg | grep "sda1"*. ¿I la comanda *dmesg | grep -i "ethernet"*? D'altra banda, ¿per a què serveixen els paràmetres *-C*, *-T*, *-d*, *-L* i *-w* d'aquesta comanda (consulta la seva pàgina de manual)?

4.-a) Atura la màquina virtual i afegeix-li un nou disc dur buit. Arrenca de nou la màquina virtual i crea amb la comanda *fdisk* dos particions en aquest nou disc (que suposarem que és `"/dev/sdb"`). Seguidament formateja-les amb la comanda *mkfs* per a què la primera sigui de tipus FAT32 i la segona de tipus Ext4. Finalment, munta-les "a mà" amb la comanda *mount* (fent servir un punt de muntatge qualsevol) i comprova que pots escriure-hi a dins. Desmunta-les en acabar.

NOTA: Pots realitzar l'apartat anterior fent servir un simple dispositiu "loop" en lloc d'haver d'usar un nou disc dur físic

b) Dedueix (¡no ho facis!) què fa *fdisk /dev/sdb < jaja.txt* ("jaja.txt" és un arxiu de text que conté el següent):

```
g
n
1

+300M
w
```

5.-Dins de la màquina virtual, descarrega't la iso <http://mirror1.slitaz.org/iso/rolling/slitaz-rolling.iso> i munta-la a una carpeta qualsevol. ¿Què hi veus allà dins? ¿Pots crear algun fitxer dins del punt de muntatge anterior? ¿Per què? Amb la iso muntada, empra la comanda *findmnt* per saber el seu sistema de fitxers i el "dispositiu" a la què està associada.

6.-a) ¿Quina diferència hi ha entre executar *df -h* i *df -hT*? ¿I entre executar *du -h -t 10M /home* i *du -hs /home*?

b) Ordena de major espai ocupat a menor espai ocupat, el contingut de la carpeta `"/usr"`. PISTA: fes servir *du -h* entubat amb la comanda *sort -hr* ¿Per a què serveix el paràmetre *-h* de *sort*?

Etiquetes i UUIDs de discs i particions

Identificar una partició fent servir la seva ruta de dispositiu (*/dev/sdXn*) és fràgil perquè el kernel podria canviar l'assignació de la lletra (passant de "sda1" a "sdb1", per exemple) en una arrencada qualsevol (això és degut als dispositius extraïbles com els llapisos USB i similars, ja que depenent del moment poden estar presents o no). I això és, òbviament, un problema.

NOTA: Un exemple d'això és, per exemple, el valor del paràmetre "root" del kernel, el qual indica, recordem-ho, la partició que serà muntada a la carpeta "/" per tal d'accedir a la resta del sistema i poder arrencar el procés INIT. Si aquest valor, indicat a la configuració del gestor d'arranc no es correspongués en un arranc determinat amb la ruta de dispositiu realment efectiva (perquè aquesta ruta hagués canviat dinàmicament en aquell arranc per alguna raó), el sistema no podria iniciar-se.

Per evitar aquest problema, existeix dues nomenclatures alternatives que es pot fer servir allà on es poden indicar rutes de dispositius (com per exemple al mencionat paràmetre "root" del kernel o per exemple també dins de l'arxiu "/etc/fstab", del qual en parlarem més endavant). Una nomenclatura alternativa és fent servir "etiquetes" ("labels") i una altra és fent servir UUIDs (essent tot això independent de si l'arranc és BIOS o UEFI: les etiquetes i els UUID valen per tots dos tipus).

Les etiquetes són simplement un nom identificador (únic per aquella màquina i, a poder ser, sense espais) que l'usuari pot assignar a una determinada partició i que sempre romandrà inalterable (com per exemple, "Les_meves_fotos"). Els UUIDs també són noms identificadors únics (de fet, així ho diu el seu nom: "Universally Unique Identifier") però són establerts automàticament pel propi sistema fent servir un algoritme pseudo-aleatori. Per assegurar que no hi hagi col·lisions amb altres possibles UUIDs existents, el format d'aquests consisteix en una cadena de 128 bits representats per 32 dígit hexadecimals.

Cal anar en compte, però, en distingir etiquetes i UUIDs associats a les particions en sí de les etiquetes i UUIDs associats al sistema de fitxers que aquestes particions contenen: aquests darrers es regeneren cada cop que una partició es (re)formateja de nou (les etiquetes a mà, els UUID automàticament) mentre que els primers es regeneren només cada cop que es (re)crea una nova partició (igualment, les etiquetes a mà, els UUID automàticament). A partir d'aquí, podem distingir quatre directives similars però diferents el valor de les quals podem fer servir al paràmetre "root" del kernel, a l'arxiu "/etc/fstab", etc:

- *Valor **LABEL**: Etiqueta associada al sistema de fitxers concret d'una partició
- *Valor **UUID**: UUID associat al sistema de fitxers concret d'una partició
- *Valor **PARTLABEL**: Etiqueta associada a una partició en concret, independentment del sist. fitx.
- *Valor **PARTUUID**: UUID associada a una partició en concret, independentment del sist. fitx.

NOTA: En principi és preferible utilitzar les directives PARTxxx perquè són immutables mentre la partició existeixi: les altres canvien quan la partició es reformateja. Cal tenir en compte, però, que les directives PARTxxx només tenen sentit en discos GPT. En aquest sentit, en definir un disc com a GPT, també es crea una capçalera en els seus primers sectors contenint diverses metadades sobre l'estructura de particions existent. Per més informació sobre l'estructura interna d'un disc GPT, consulteu https://en.wikipedia.org/wiki/GUID_Partition_Table

Cal tenir en compte que, a més de les etiquetes i UUIDs mencionats anteriorment, a les particions GPT existeix una altra metadada (inclosa també la capçalera de cada partició respectiva) anomenada "GUID", la qual formalment és un altre UUID però que, en aquest cas, serveix per definir la "funcionalitat" de la partició. Ja coneixem un GUID, que és el que fa que una partició FAT32 estigui marcada com a ESP (concretament, el GUID C12A7328-F81F-11D2-BA4B-00A0C93EC93B) però a l'estàndard https://uapi-group.org/specifications/specs/discoverable_partitions_specification hi ha uns quants més definits. En concret, si fem servir la comanda *lsblk*, el GUID concret d'una partició es pot saber mostrant la columna **PARTTYPE** (i també **PARTTYPE_NAME**)

NOTA: Un article molt interessant que profunditza sobre els motius i els usos pràctics de l'estàndard "Discoverable Partitions" és: <http://0pointer.net/blog/the-wondrous-world-of-discoverable-gpt-disk-images.html>

Si no es va establir explícitament durant la instal·lació del sistema, la comanda *lsblk* no mostrarà cap valor per les directives LABEL o PARTLABEL. Si els volem assignar ara un valor, hem de tenir el compte el següent:

*Els valors de PARTLABEL i/o PARTUUID només els podrem establir amb una comanda de gestió de particions com ara *fdisk* o *parted* . Igualment passa el mateix amb el valor PARTTYPE. Ho veurem als exercicis.

*Els valors de LABEL i/o UUID els podrem establir en el moment de formatar una partició (mitjançant els paràmetres adients de la comanda *mkfs*, per exemple) o bé, a posteriori, amb una comanda específica per etiquetar que serà diferent segons el sistema de fitxers present (per exemple, a sistemes de fitxers Ext4 s'utilitza la comanda *tune2fs*). Ho veurem als exercicis.

NOTA: Segons el sistema de fitxers que es vulgui etiquetar, la comanda "etiquetadora" es podrà executar amb el sistema de fitxers muntat o caldrà desmuntar-lo prèviament).

NOTA: Per més informació sobre com consultar (i assignar) els diferents tipus d'etiquetes i UUIDs, podeu consultar https://wiki.archlinux.org/title/Persistent_block_device_naming

EXERCICIS:

Arrenca la mateixa màquina virtual UEFI Fedora que has fet servir als exercicis anteriors, amb el segon disc ("/dev/sdb") encara adjuntat. A partir d'aquí:

1.-a) Esbrina quins són els valors de UUID i PARTUUID que ha assignat automàticament el sistema Fedora a la partició "/dev/sda2" i al seu sistema de fitxers respectivament, executant la comanda `lsblk` per exemple així: `lsblk -o NAME,UUID,PARTUUID`

NOTA: Aquesta comanda treu la informació de les carpetes "/dev/disk/by-uuid", "/dev/disk/by-label", "/dev/disk/by-partuuid" i "/dev/disk/by-partlabel", totes elles gestionades pel kernel

b) ¿Quina diferència hi ha entre la comanda anterior i `lsblk -o +UUID,PARTUUID` ?

c) ¿Què mostra en general la comanda `findfs {LABEL|PARTLABEL|UUID|PARTUUID}=valor` ? Prova-la

d) ¿Què mostra la comanda `lsblk -o NAME,PTTYPE`? ¿I la comanda `lsblk -o NAME,PARTTYPE,PARTTYPENAME`?

2.-a) Executa la comanda `fdisk` per treballar sobre el disc "sdb". Pulsa l'opció "l" i observa la llista que t'apareix. ¿Quins informació et mostra (observa com a pista el primer element de la llista)? ¿Quin és el codi numèric corresponent al tipus genèric "Linux filesystem" (vist, per exemple, al darrer apartat de l'exercici anterior)?

NOTA: Per canviar el GUID d'una determinada partició, un cop sabem el seu codi numèric, caldria pulsar l'opció "t".

3.-a) Assigna una etiqueta al sistema de fitxers de la partició "/dev/sdb2". Com que aquesta està formatada en Ext4, la comanda adient (executada com a "root") serà: `tune2fs -L "La_meva_etiqueta" /dev/sdb2` (o també `e2label /dev/sda2 "La_meva_etiqueta"`). Executa, doncs, aquesta comanda i comprova amb `lsblk` que, efectivament, ha sigut assignada l'etiqueta LABEL indicada.

NOTA: En el cas d'altres sistemes de fitxers, la comanda per (re)assignar una etiqueta és diferent. En concret:

*Per una partició XFS ja formatada: `sudo xfs_admin -L "La_meva_etiqueta" /dev/sda2`

*Per una partició FAT32 ja formatada: `sudo fatlabel /dev/sda2 "La_meva_etiqueta"` (ha d'estar desmuntada)

A https://wiki.archlinux.org/index.php/Persistent_block_device_naming#by-label es poden veure més exemples

NOTA: Tal com hem dit, també es poden assignar les etiquetes en el mateix moment de formatar una partició, recordem:

*Per una partició Ext4 o XFS: `sudo mkfs.{ext4|xfs} -L "La_meva_etiqueta" /dev/sda2`

*Per una partició FAT32: `sudo mkfs.vfat -n "La_meva_etiqueta" /dev/sda2`

NOTA: Encara que no és el normal, els UUIDs també es poden (re)assignar explícitament a una partició ja formatada:

*Per una partició Ext4 ja formatada: `sudo tune2fs -U {valoramà|random|time|clear} /dev/sda2`, on:

-U *valoramà* : s'indica el valor explícitament com si fos una etiqueta.

Per generar possibles valors vàlids es pot fer servir la comanda `uuidgen`

-U *random* : canvia el UUID aleatòriament

-U *time* : canvia el UUID basant-se en l'hora actual

-U *clear* : treu el UUID actual

*Per una partició XFS ja formatejada: `sudo xfs_admin -U {valoramà|generate} /dev/sda2`

NOTA: Tal com hem dit, també es poden assignar UUIDs en el mateix moment de formatar una partició, així:

*Per una partició Ext4: `sudo mkfs.ext4 -U {valoramà|random|time|clear} /dev/sda2`

*Per una partició XFS: `sudo mkfs.xfs -m uuid=valoramà /dev/sda2`

aII) Llegeix en les notes anteriors com podries (re)assignar un UUID aleatori a la partició "/dev/sdb2" i fes-ho. Comprova que ho hagis fet correctament. Reinicia la màquina.. ¿Què passa? ¿Per què?

Tal com hem dit, amb la comanda `fdisk` també és possible canviar les metadades de les particions GPT pròpiament dites (no dels seus sistemes de fitxers). Concretament, si pulsem l'opció "x" per anar a un menú d'opcions avançades, allà podrem triar diverses opcions (que es poden mostrar amb un nou menú pulsant "m"), com per exemple:

n : Per canviar el PARTLABEL

u : Per canviar el PARTUUID

S : Per fer la partició de només lectura (entre altres "flags" que es poden activar)

i : Per canviar el GUID del disc sencer (no se sol fer servir gaire)

r : Per tornar al menú principal (i llavors finalment guardar els canvis amb *w*).

4.- Llegeix els paràgrafs blaus anteriors i canvia el PARTLABEL i el PARTUUID de la partició "sdb2". Comprova-ho.

Arxiu "/etc/fstab"

¿Com sap el sistema -o més específicament, el "kernel"- quines són les particions que ha muntar automàticament durant el seu arranc? Com a mínim la partició arrel (és a dir, on es troba tot l'arbre de carpetes d'una distribució) ha de ser obligatòriament muntada a "/" per poder fer accessible tot el sistema. I, a més, podríem tenir les carpetes dels usuaris guardades en una altra partició que caldria muntar en "/home", etc

Doncs bé, el "muntatge automàtic" és possible gràcies a què el "kernel", en arrencar, llegeix un arxiu anomenat **"/etc/fstab"**, el qual conté la informació de quines particions ha de muntar automàticament, en quin punt de muntatge concret i amb quines opcions.

NOTA: Les particions indicades en aquest fitxer es poden correspondre, fins i tot, no només a particions locals sinó també a dispositius remots tipus Samba o NFS i configuracions més complexes

Cada línia d'aquest arxiu consta de sis camps separats per un (o més) espais en blanc o per un tabulador, tant se val (per més informació, consulteu *man fstab*):

*El **1r camp** començant per l'esquerra indica la partició/carpeta compartida concreta que s'haurà de muntar durant l'arranc del sistema operatiu (per exemple, la partició local `"/dev/sda1"`).

NOTA: En el cas concret de voler muntar carpetes compartides per NFS, aquest camp haurà de valer la IP o nom del servidor NFS seguit de ":" i la ruta de la carpeta compartida a la què es vol accedir.

*El **2n camp** indica la ruta de la carpeta que funcionarà com el seu punt de muntatge local

*El **3r camp** indica el sistema de fitxers en què està formatada aquesta partició/carpeta compartida (valors possibles són: "ext4", "xfs", "jfs", "vfat" -per "FAT32"-, "ntfs", "swap", o "auto", entre altres.

NOTA: El valor "auto" serveix per no especificar cap sistema en concret i què el procés INIT el determini ell sol en el moment del muntatge...aquesta opció és útil per lectors d'unitats extraïbles com CD/DVDs que poden contenir diferents sistemes de fitxers dependent de la unitat introduïda.

*El **4r camp** indica les opcions de muntatge s'hi apliquen (separades per comes, veure taula següent)

*El **5è camp** actualment no s'utilitza i quasi sempre val 0

*El **6è camp** indica si (i quan) la comanda *fsck* s'executarà en reiniciar el sistema quan es munti aquesta partició/carpeta compartida.

NOTA: Un valor 0 vol dir que no es comprova el sistema de fitxers, un valor 1 vol dir que sí i un valor 2 vol dir que sí, però després d'haver comprovat les particions marcades amb valor 1; normalment la partició arrel del sistema ha de tenir un 1 mentre la resta de particions pot tenir 0 o 2).

Així doncs, una línia típica que podríem trobar en aquest fitxer seria la següent (la qual munta la partició local `"/dev/sda2"`, de tipus "ext4", sota la carpeta `"/home"` amb les opcions per defecte):

```
/dev/sda2 /home ext4 defaults 0 0
```

NOTA: Aviat veurem que, en muntar particions locals, per identificar-les podem especificar al primer camp del fitxer `"/etc/fstab"` altres valors diferents del típic `"/dev/sdxX"`, com ara una etiqueta (`LABEL=EtiquetaQueVolem`) o bé un UUID (`UUID=uuiddelaparticio`).

NOTA: Ja que els espais són usats per delimitar camps en l'arxiu `"/etc/fstab"`, si algun camp (com per exemple `LABEL` o la ruta d'algun punt de muntatge) contingués espais, aquests haurien de ser reemplaçats pel caràcter d'escapament `\040`. Un exemple seria així escriure un punt de muntatge tal com aquest: `/media/100\040GB\040(Storage)`

Les opcions de muntatge poden ser, a més de les més específiques per cada sistema de fitxers (per exemple, les de NFS4 estan llistades a *man nfs*, les d'Ext4 a *man ext4*, etc), qualsevol de les generals que aporta la comanda *mount* de forma genèrica (és a dir, les que estan llistades a *man mount*), com ara, per exemple:

| | |
|-------------------|---|
| ro | Munta en mode de només lectura. Per defecte les muntatges es fan en mode escriptura (rw). |
| noatime | No canvia la data d'últim accés als fitxers. Aquesta metadata es guarda juntament amb més metadades (com la data d'última modificació, etc) en una "base de metadades" de fitxers anomenades "inodes". Als kernels antics, per defecte l'opció atime equivalia a strictatime , que implica que cada accés a un fitxer es canviï la data d'últim accés: amb <i>noatime</i> això no es fa, accelerant així l'accés al disc (i evitant escriptures de metadades a disc quan realment només es vol llegir informació). En tot cas, com que <i>strictatime</i> és una opció massa exigent pel disc, actualment els kernels fan que <i>atime</i> equivalgui a relatime , que implica que només es canvia la data d'últim accés si aquesta és anterior a la d'última modificació o bé si ha passat més d'un dia del darrer accés. |
| nodiratime | No canvia la data d'últim accés als directoris. L'opció contrària (i per defecte) és diratime . Si s'indiqués <i>noatime</i> ja implica <i>nodiratime</i> |

De fet, hi ha força d'aquestes opcions que tenen més sentit dins de " /etc/fstab" que no pas en una comanda *mount* normal. A continuació mostrem les més habituals:

| | |
|----------------------------|--|
| auto | Indica que la partició en qüestió es muntarà automàticament durant l'arranc. En altres paraules: és la manera "d'activar" aquesta línia. L'opció contrària és noauto , que serveix per "desactivar" la línia sense haver d'esborrar-la, fent que només es pugui muntar la partició de forma manual amb la comanda <i>mount</i> o bé si es combina amb l'opció "x-systemd.automount" (veure més avall) |
| exec | Permet que es puguin executar els binaris existents a la partició en qüestió. L'opció contrària és noexec |
| suid | Permet reconèixer el permís especial "suid" en els binaris existents a la partició en qüestió. Aquest permís especial s'utilitza per permetre a usuaris que no són administradors executar binaris que en teoria només podria executar l'usuari administrador (com ara el binari <i>passwd</i> , per exemple). L'opció contrària és nosuid |
| dev | Permet reconèixer els arxius de dispositius que puguin existir a la partició en qüestió. L'opció contrària és nodev . |
| user | Permet que qualsevol usuari (sense ser administrador) pugui muntar amb la comanda <i>mount</i> la partició en qüestió (però només l'usuari que l'hagi muntat la podrà desmuntar). Aquesta opció implica "noexec", "nosuid" i "nodev", a no ser que s'indiqui el contrari. L'opció contrària (és a dir, que només l'usuari root pugui muntar la partició en qüestió) és nouser |
| users | Permet que qualsevol usuari (sense ser administrador) pugui muntar amb la comanda <i>mount</i> la partició en qüestió, i també que qualsevol altre usuari (o el mateix) la pugui desmuntar. Aquesta opció implica "noexec", "nosuid" i "nodev", a no ser que s'indiqui el contrari. |
| sync | Fa que les escriptures a la partició en qüestió es realitzin de forma síncrona (és a dir, que s'apliquin immediatament). L'opció contrària és async , la qual implica que les escriptures es realitzin de forma asíncrona (és a dir, que totes les escriptures s'apliquin de cop passat un temps determinat). Aquestes opcions en NFS no s'utilitzen perquè no s'estableixen al client sinó en el servidor a l'hora de compartir la carpeta dins de "/etc/exports" (són opcions homònimes, de fet). |
| defaults | Equival a "rw", "suid", "dev", "exec", "auto", "nouser" i "async" |
| x-systemd.automount | Aquesta opció s'ha d'escriure juntament amb l'opció "noauto". Serveix per no muntar la partició en qüestió durant l'arranc del sistema sinó quan l'usuari vulgui accedir-hi per primera vegada (el retràs és imperceptible). Això permet no tenir muntades unitats que potser l'usuari no utilitzarà en aquella sessió, i a més, en el cas de sistemes de fitxers remots com NFS o CIFS, permet no obtenir errors en arrencar el client si en aquell moment el servidor no està disponible. Es pot utilitzar a més l'opció x-systemd.device-timeout=n° , on n° és el número de segons que el client intentarà muntar el dispositiu abans de deixar-ho còrrer (perquè aquest no estigui disponible, per exemple). No obstant, només funciona per sistemes amb Systemd. |

EXERCICIS:

1.-¿Quines línies hauries d'introduir/eliminar de l'arxiu "/etc/fstab" actual per tal de muntar de forma automàtica a la carpeta "/opt" la partició "/dev/sdb2" amb les següents opcions?:

- *En mode de lectura-escriptura i síncron. ¿Què vol dir això de "síncron"?
- *Impedint l'execució de qualsevol programa que estigui emmagatzemat allà
- *No modificant la data d'accés quan hi hagi un accés als fitxers del seu interior
- *Sense cap possibilitat de ser (des)muntat si no s'és "root"
- *Invocant a *fsck* en cada muntatge.

Empaquetament de carpetes i fitxers amb Tar i compressió amb Gzip/Bzip2/Xz

A continuació es llisten les opcions més importants dels compressors/descompressors més comuns. El fet de què hi hagin tantes eines de compressió/decompressió és degut al fet de què segons l'algoritme que emprin, unes seran més ràpides per comprimir i/o descomprimir però ho faran a una ràtio de compressió menor i/o a un major consum de CPU (o viceversa). No existeix un algoritme que ho tingui tot: que sigui ràpid, comprimeixi molt i exigeixi pocs recursos al sistema.

Cal tenir en compte que tots ells només són capaços de comprimir fitxers individuals, no pas carpetes. De fet si s'indiquen varis fitxers a comprimir/descomprimir, es realitzarà el procés adient fitxer per fitxer de forma independent. És per això que aquestes eines les haurem de combinar amb una altra que faci la tasca d'"empaquetar"/"desempaquetar" (és a dir, reunir el conjunt desitjat de fitxers i carpetes dins un sol fitxer -l'anomenat "paquet"-, el tamany del qual serà la suma dels tamanyes dels seus membres; aquest paquet llavors sí que podrà ser comprimit individualment si es desitja). L'exemple més evident és la comanda *tar* (tot i que n'hi ha més, com *cpio* i altres).

Per comprimir es poden fer servir les següents comandes i paràmetres comuns:

gzip unarxiu unaltre ... Comprimeix els arxius indicats (anomenant-los "unarxiu.gz"...) amb l'algoritme GZ
bzip2 unarxiu unaltre ... Comprimeix els arxius indicats (anomenant-los "unarxiu.bz2"...) amb l'algoritme BZ
xz unarxiu unaltre ... Comprimeix els arxius indicats (anomenant-los "unarxiu.xz"...) amb l'algoritme XZ

-1 ... -9 Especifica el nivell de compressió: -1 = menys (però més ràpid) i -9 més (més lent i costós)
-v Mode verbós
-t Comprova la integritat dels fitxers
-l Mostra informació sobre la ràtio de compressió sobre l'arxiu especificat
-k No elimina l'arxiu original (és a dir, s'obté l'arxiu comprimit i a més es deixa l'original)
-c Envia les dades comprimides (o descomprimides) a la sortida estàndard
-r Si s'especifica una carpeta en lloc d'un arxiu, comprimeix el seus arxius interns (només *gzip*)

Per descomprimir es poden fer servir les següents comandes i paràmetres:

gunzip unarxiu.gz ... Descompreix l'arxiu GZ (anomenant-lo "unarxiu"). Equival a *gzip -d unarxiu.gz ...*
bzunzip2 unarxiu.bz2 ... Descomprimeix l'arxiu BZ (" "). Equival a *bzip2 -d unarxiu.bz2 ...*
unxz unarxiu.xz ... Descomprimeix l'arxiu XZ (" "). Equival a *xz -d unarxiu.xz ...*

-v Mode verbós
-t Comprova la integritat dels fitxers

NOTA: Existeixen moltes més eines, com ara *compress/uncompress* (o *compress -d*) o *zip/unzip* (equivalent a *tar+compress*), el qual té l'interessant paràmetre *-e* per afegir una contrasenya al paquet "zip" i així protegir-lo

NOTA: En el cas de tenir arxius comprimits de tipus text, es pot consultar el seu contingut sense haver de descomprimir-los prèviament gràcies a les comanda *zgrep,zcat,zless...* (del paquet "gzip"), *bzgrep,bzcat,bzless...* (del paquet "bzip2") o *xzgrep,xzcat,xzless...* (del paquet "xz")

Per empaquetar una carpeta (i/o un conjunt d'arxius) es poden usar els següents paràmetres de la comanda *tar*:

tar -cf/ruta/arxiu.tar ruta/relativa/carpeta ruta/relativa/fitxer ... Crea a la ruta indicada un paquet "tar" anomenat "unarxiu.tar" amb el contingut sencer de la carpeta i de tots els fitxers individuals indicats. És important tenir en compte que les rutes de les carpetes i fitxers indicades a incloure a l'arxiu TAR han de ser relatives per tal de què quan es faci el desempaquetament, el resultat sigui relatiu a la carpeta on es faci aquest.

-z Comprimeix amb l'algoritme GZ (opcions per defecte). L'extensió del paquet sol ser "tgz"
-j Comprimeix amb l'algoritme BZ (opcions per defecte). L'extensió del paquet sol ser "tbz"
-J Comprimeix amb l'algoritme XZ (opcions per defecte). L'extensió del paquet sol ser "txz"
-a Comprimeix amb l'algoritme que es dedueixi de l'extensió indicada pel paquet

NOTA: Els paràmetres anteriors ens eviten haver de fer el *tar* normal i posteriorment haver d'utilitzar a mà el compressor desitjat (per exemple, així, *gzip arxiu.tar*); és a dir, haver de fer el procés en dues passes separades. Si es fes així, en qualsevol cas, l'extensió del paquet resultant sol ser "tar.gz", "tar.bz2" o "tar.xz". Per descomprimir aquest tipus d'arxius s'haurà d'utilitzar llavors primer el descompressor (per exemple, així, *gunzip arxiu.tar.gz*) per obtenir el fitxer .tar i després s'ha d'utilitzar l'opció corresponent del tar normal (que de seguida veurem) per desempaquetar el seu contingut.

- P Permet indicar rutes absolutes en crear el paquet "tar" (és a dir, no es treu la "/" del principi)
- h Segueix els enllaços simbòlics per tal d'empaquetar els fitxers apuntats pròpiament
- T *llista.txt* En lloc d'indicar la llista de rutes d'arxius o carpetes a empaquetar com a paràmetres de la comanda *tar*, es poden incloure cadascuna com una línia diferent d'un arxiu de text indicat amb aquest paràmetre. Es pot indicar un guió en lloc de l'arxiu de text per indicar "stdin"
- exclude **.txt* Indica noms (poden tenir comodins) de fitxers que no s'inclouran al paquet
- no-recursion Si s'indiquen carpetes a afegir el paquet, s'afegiran sense recursivitat
- N *YYYYMMDDhhmmss* Només empaquetarà els fitxers que siguin més nous de la data indicada
- v Mode verbós

NOTA: Per defecte, en fer un paquet "tar" es guarden els permisos i el propietari dels fitxers empaquetats. Si es volen guardar altres metadades, com ara les ACLS o els atributs extesos, caldrà afegir llavors el paràmetre adients (--acls i --xattrs respectivament)

Per desempaquetar un arxiu "*.tar" es poden usar els següents paràmetres de la comanda *tar*:

tar -xf/ruta/arxiu.tar [uncarpetaoarxiu] [...] Extreu el contingut del paquet "tar" a la carpeta actual. Si s'indiquen carpetes o arxius concrets (que se suposa que estan empaquetats), només s'extreuran aquests (de forma recursiva) i prou.

- z Si el fitxer fos "tgz", s'ha d'afegir aquest paràmetre per descomprimir-ho abans de desempaq
- j Si el fitxer fos "tbz", s'ha d'afegir aquest paràmetre per descomprimir-ho abans de desempaq
- J Si el fitxer fos "txz", s'ha d'afegir aquest paràmetre per descomprimir-ho abans de desempaq
- p Manté els permisos originals en els arxius desempaquetats (si no s'indica, aquests es crearan amb l'umask per defecte del sistema). Aquesta opció és la per defecte si *tar* l'executa "root"
- same-owner Manté el propietaris i grups originals en els arxius desempaquetats (si no s'indica, el propietari i grup seran els corresponents a l'usuari que realitzi el desempaquetament). Aquesta opció és la per defecte si *tar* l'executa "root"
- m Canvia la data de modificació dels fitxers extrets per la data actual
- exclude **.txt* Indica noms (poden tenir comodins) de fitxers que no s'extreuran del paquet
- C */ruta/carpeta/on/sextreuen/els/fitxers* (en lloc de la carpeta actual). Aquest paràmetre és necessari que s'especifiqui el primer de tots, just després del nom del binari
- strip-components=*n*^o Indica els nivells en la ruta dels fitxers empaquetats que es vol saltar a l'hora de desempaquetar-los. Per exemple, si dins d'un fitxer "tar" tenim dos fitxers amb una ruta com "unacarpeta/unaaltra/unfitxer.txt" i "unacarpeta/unaaltra2/unfitxer2.txt", en fer *tar --strip-components=1 -xf arxiu.tar* a la carpeta actual es desempaquetarà "unaltra/unfitxer.txt" i "unaltra2/unfitxer2.txt"
- k Si a la carpeta de destí hi hagués algun arxiu amb el mateix nom que un dels desempaquetats, no serà sobreescrit
- v Mode verbós

NOTA: En lloc d'"arxiu.tar" es pot escriure un guió ("-"), indicant així la sortida estàndard (en el cas d'empaquetar) o l'entrada estàndard (en el cas de desempaquetar). Això pot ser útil per utilitzar canonades o redireccionadors, com mostra aquest exemple *tar cf - . | (cd /una/ruta; tar xvf -)*

Altres possibilitats interessants de la comanda *tar* són:

tar -uf/ruta/arxiu.tar unnouarxiu ... Afegeix al final d'un paquet "tar" ja creat els fitxers indicats (si ja hi ha algun arxiu amb el mateix nom a dins, només s'afegirà si és més actual).

NOTA: És important entendre que la comanda anterior no substitueix la versió antiga de l'arxiu dins del paquet, sinó que n'afegeix el nou al final. Ara bé, com que els arxius es recuperen de forma seqüencial des del principi del paquet, en una extracció normal la nova versió sobreescrirà a l'antiga i, per tant, a la pràctica, haurem recuperat la versió més recent. Si la nostra intenció fos recuperar una versió antiga, podem usar l'opció --occurrence=*n*^o, on el nombre representa la versió de l'arxiu que es vol rescatar (és a dir, la seva posició dins del paquet; si no s'indica cap nombre s'entén que és la primera versió).

tar --delete -f/ruta/arxiu.tar unnouarxiu ... Elimina els arxius indicats del paquet "tar"

tar -tf/ruta/arxiu.tar Mostra el contingut del paquet "tar". Amb -v la sortida és similar a la de *ls -l*

En el cas de tenir fitxers d'un tamany molt gran, es poden dividir en trossos amb la comanda *split*, així:

split -b tamanytros{KMG} /ruta/arxiu Talla l'arxiu indicat en trossos amb el tamany indicat. Els arxius resultants tenen per defecte el nom "xaa", "xab", "xac", "xad",... Per concatenar-los, només cal usar la comanda *cat* així: *cat xaa xab > final*. Opcionalment, després d'indicar la ruta de l'arxiu es pot indicar el prefixe que es vol pels trossos en comptes del "x"

EXERCICIS:

1.-a) Empaqueta sense comprimir el teu directori personal de la màquina real, guardant el paquet a `"/tmp"`

b) Repeteix l'apartat a) però ara comprimint el paquet generat amb *xz*. Fes-ho de dues maneres: invocant el compressor directament a través de la comanda *tar* en una sola comanda i invocant-lo explícitament mitjançant una segona comanda després d'haver creat el paquet "tar". Compara el tamany d'aquests dos paquets comprimits amb el paquet sense comprimir generat a l'apartat anterior.

NOTA: Pots comparar també el temps invertit en generar el paquet respectiu (se suposa que en comprimir-lo trigarà més) si afegeixes la comanda *time* al davant de la invocació a *tar*

c) Repeteix l'apartat b) -tria el mètode que vulguis per invocar el compressor- però ara fent que dins del paquet comprimit es guardi la ruta completa dels fitxers. Observa el seu contingut per confirmar-ho.

d) Desempaqueta (i descomprimeix) el paquet comprimit generat a l'apartat b) dins de la carpeta `"/tmp/personal"`

2.- Utilitza la comanda *split* per dividir un arxiu qualsevol en trossos de 5MB. Un cop fet això, esborra l'arxiu original i utilitza seguidament la comanda *cat* per agrupar el contingut d'aquests trossos en un de sol per tenir un altre cop l'arxiu de l'inici.

Imatges binàries amb Dd

La comanda *dd* l'únic que fa és copiar bits d'un lloc a un altre, literalment: no entén ni de carpetes ni de fitxers ni de cap concepte relacionat amb sistemes de fitxers: només entén de blocs de bytes. És per això que és compatible amb qualsevol tipus de formatatge que hi hagi en un dispositiu d'emmagatzematge perquè en realitat "passa" d'ell ja que treballa a més baix nivell. El funcionament bàsic d'aquesta comanda és el següent:

dd if=/dev/sda of=/dev/sdb Fa una còpia bit a bit del disc sda en sdb. El dispositiu d'origen també podria ser el dispositiu */dev/zero* ó */dev/urandom*, com veurem.
dd if=/dev/sda1 of=/dev/sda2 Fa una còpia bit a bit de la partició 1 a la 2.
dd if=/dev/sda1 of=imatge.dd Genera una imatge bit a bit de la partició 1 . També podria ser a la inversa

Alguns paràmetres interessants d'aquesta comanda són:

bs=midablocacopiar{KMG} Indica la mida dels "paquets" (blocs) que s'esperen rebre de l'origen (*ibs=*) per tal de copiar-los al destí amb aquesta mateixa mida (*obs=*). Augmentant la mida del bloc (per defecte és 512 bytes) es pot accelerar el procés de copia
count=nºblocsacopiar Indica el nombre de blocs a copiar (per defecte, tots fins que s'acabi el flux d'entrada)
status=progress Mostra una barra de progrés mentre es realitza la còpia
skip=nº (o *iseek=nº*) Indica el nombre de blocs que se salten a l'origen abans de fer la còpia
seek=nº (o *oseek=nº*) Indica el nombre de blocs que se salten al destí abans de fer la còpia
conv=noerror Fa que *dd* no es detingui en detectar els errors de l'entrada (no escrivint-ne llavors res a la sortida pel bloc en qüestió)
conv=noerror,sync Fa que, a més de no detenir-se en no poder llegir algun bloc d'entrada, a la sortida aquests blocs no llegits se substitueixin per zeros (per mantenir així l'alineament en la posició dels bytes amb l'entrada)
conv=fsync Fa que *dd* escrigui, abans de finalitzar, físicament al disc tots els blocs de sortida que puguin estar encara ubicats a la catxé. Seria equivalent a executar a posteriori la comanda *sync* Si no s'indica cap d'aquestes opcions, l'escriptura real al disc serà responsabilitat del kernel i no computarà en el temps mesurat per *dd* No confondre amb el paràmetre *oflag=sync*, el qual força a l'escriptura en disc de cada bloc, un per un (també existeix el paràmetre *oflag=direct*, el qual obvia l'existència de la catxé i fa una escriptura directa de l'entrada a disc...en tot cas, ambdues darreres opcions penalitzen força el rendiment)

Alguns exemples pràctics d'ús d'aquesta comanda són:

dd if=/dev/sr0 of=arxiu.iso Genera una iso d'un CD
dd if=/dev/urandom of=/dev/sda Escriu dades aleatòries al disc
dd if=/dev/zero of=/dev/sda Esborra un disc
dd if=/dev/zero of=contenedor.bin bs=1k count=100 Crea un fitxer (ple de zeros) amb una mida de 100KB.

EXERCICIS:

1.-Escriu la comanda *dd* concreta que faries servir per:

- Crear un fitxer iso al teu \$HOME que sigui la imatge d'un llapis USB identificat com a */dev/sdb*
- Posar a zero tots els sectors d'un altre llapis USB identificat com a */dev/sdc*
- "Planxar" la imatge iso del primer punt al llapis USB del segon punt

2.- Digues què fan cadascun de les següents comandes i què acabes tenint al final

```
dd if=/dev/zero of=file.bin bs=1k count=100 status=progress
mkfs.ext4 file.bin
mkdir /mnt/topsecret
mount -t ext4 file.bin /mnt/topsecret
cp contrasenyes.txt /mnt/topsecret
umount /mnt
```