

Comandes Linux (I)

***Treballar amb processos i memòria:**

ps Mostra (una "foto" de l'estat d)els processos executants per mi només al shell actual.
Columna PID: Identificador del procés (nº únic assignat pel kernel en posar-se en marxa)
Columna TTY: Terminal on s'està executant el procés (pts/0, tty1... o "?" si és un dimoni)
Columna TIME: Temps "d'ús efectiu" de la CPU per part del procés (hh:mm:ss)
Columna CMD: Ruta absoluta de l'executable corresponent al procés en qüestió

El temps "d'ús efectiu" de la CPU és la suma del temps anomenat "user" més el temps anomenat "system". El primer és el temps que està "dins de la CPU" el codi propi del procés en sí; el segon és el temps que està "dins de la CPU" el codi de les funcions del kernel cridades pel procés (és a dir, tot aquest codi que no forma part del procés en sí -ni de les seves llibreries- sinó que és ofert de forma estàndar pel kernel Linux a tots els processos sota el nom de "system calls" per tal de què siguin invocades per aquests quan sigui necessari).

NOTA: Les "system calls" s'encarreguen d'executar tasques de baix nivell per a què no calgui que ho facin els processos per sí mateixos (per exemple, cada cop que un procés vol obrir un fitxer, interpreten l'estructura del disc dur subyacent i el sistema de fitxer que hi hagi a sobre per tal de què el procés no hagi d'implementar aquests detalls de baix nivell).

Els [] que apareixen envoltant el nom d'alguns processos denoten que aquests processos no tenen associat un executable individual accessible des de terminal (generalment, això passa quan són threads del kernel i en alguns dimonis). El /0, /1 ... indica el nucli de la CPU on s'executa aquest fil/dimoni.

-a Mostra els processos executats per tots els usuaris en qualsevol shell (pts/0, tty1, ...)
-e Mostra el mateix que -a però a més a més els dimonis (normalment executats com root)
-p nºpid,nºpid,... Mostra només processos el PID dels quals coincideixi amb l'escrit
--ppid nºppid Mostra només processos fills del procés indicat
-C nom,nom,... Mostra només processos el nom de l'executable dels quals coincideixi amb l'escrit
-t tty1,pts/0,? Mostra només processos associats als terminals indicat
-u usuari,... Mostra només processos d'aquest usuari
-N Mostra els processos que NO tinguin les característiques indicades

-H Mostra l'arbre de processos. També està l'opció -forest. Incompatible amb -L
-f Mostra més columnes:
Columna UID: Nom de l'usuari que ha posat en marxa el procés (o si és massa llarg, UID)
Columna PPID: PID del procés pare. Fixar-se sobre tot en el 0 (scheduler) i 1 (systemd)
Columna STIME: Instant d'inici del procés
Columna C : Part sencera del % d'ús de la CPU (veure més avall)
-F Mostra les mateixes columnes que -f més tres més:
Columna SZ: Tamany màxim de la memòria (en unitats-pàgina, veure més avall) que el procés podria arribar a utilitzar (però a la qual no sol arribar perquè aquí està contada la possible memòria swap que podrà fer servir, la memòria compartida i, sobre tot, l'espai reservat per carregar funcions del codi que poden no invocar-se mai). Aquest tamany és assignat pel kernel a petició del que demana el propi procés.
Columna RSS: Tamany de la memòria RAM (en KB) que tot el procés (codi+dades+pila) està utilitzant en aquest moment, sense comptar la swap. Cal tenir en compte, però, que en el total de cada procés s'inclou la possible memòria compartida amb altres processos (degut a l'ús de llibreries comunes) i com que aquestes llibreries només es carreguen una vegada independentment del número de processos que les facin servir, el còmput global d'aquesta columna sol ser sobreestimat.
NOTA: En aquest sentit, pot ser més útil el valor PSS, el qual és similar a RSS però amb un canvi: en sumar al còmput total la memòria compartida, es divideix aquesta memòria entre el número de diferents processos que la comparteixen, donant per tant un resultat més realista (encara que tampoc exacte perquè es fa una divisió equitativa però no sol ser així en veritat)
Columna PSR: Nucli que està executant ara mateix el procés
-L Combinat amb -f, mostra dues columnes més, relacionades amb "threads":
Columna LWP: Identificador del "thread" en qüestió
Columna NLWP: Nº de "thread" que conté el procés al que pertany
-M Mostra columna extra amb el contexte SELinux del procés en qüestió. Igual que -Z.

Existeix una columna -invisible per ara- del *ps* anomenada **VSZ**. El seu significat és idèntic al de la columna **SZ**, però les unitats utilitzades per mesurar el valor mostrat són diferents; en el cas de **SZ** són "pàgines" i en el cas de **VSZ** són Kilobytes. Una pàgina és el bloc mínim de memòria que Linux pot tractar individualment (de fet, el concepte és similar al de "bloc" en un disc dur) ; per saber el tamany d'una pàgina es pot executar la comanda: *getconf PAGE_SIZE*. En sistemes actuals sol ser de 4KB (4096 bytes) i, per tant, en aquests casos es compleix que **VSZ**=**SZ***4

-ww	Si una línia és més llarga que l'amplada de la pantalla, no la talla
-M	Mostra una columna sobre l'estat de SELinux per cada procés
-o nomcol,...	Mostra només les columnes especificades (jen minúscules!) El títol es pot canviar així: "cmd=Comanda,tty=Terminal" o esborrar-lo així "cmd="
	L'amplada de la columna es pot canviar així: "cmd:42"
	Es poden combinar les dues possibilitats, així: "cmd:42=Terminal"
--sort [-]nomcol	Mostra els processos ordenats per la columna especificada (PID, USER, %CPU, etc) El guió davant del nom de la columna indica que s'ordenarà en ordre invers

Els valors que poden acompanyar als paràmetres *-o* i *--sort* no es limiten només als noms de les columnes mostrades "de sèrie" pel *ps* sinó que poden ser molts més. La llista sencera es troba a la secció "Standard format specifiers" de la pàgina del manual de *ps* però a continuació destaquem els següents:

"%cpu" o "pcpu"	:ràtio entre el temps d'ús efectiu de la cpu i el temps que porta funcionant el procés des de què es va iniciar. Per veure un valor alt es pot executar per exemple <i>cat /dev/urandom > /dev/null</i>
"%mem" o "pmem"	:ràtio entre rss i el tamany de la memòria física de la màquina
"comm"	:similar a "command" -o a "args"- però sense mostrar els paràmetres de l'executable
"start" o "stime"	: moment d'inici del procés
"etime"	: temps "físic real" que porta el procés funcionant des de que es va iniciar (per tant, podem deduir que $pcpu=(time/etime)*100$). També està el valor "etimes", que és el mateix però en n° de segons
"s" o "state"	: estat del procés, el qual pot ser: R (running) S (sleeping interrompible -perquè s'espera algun event-. Se solen usar senyals per interrompre) D (sleeping ininterrompible -perquè s'espera una resposta d'I/O normalment a una falla de catxé-) Z (zombie: procés fill que ha mort però no ho ha notificat al pare i per tant encara apareixen) T (stopped, per una senyal SIGxxx o per estar sent depurat, rar de veure) SW (swapped; rar de veure)

<i>ps tree</i>	Mostra una "foto" de l'arbre de processos actual
-a	Mostra a més els paràmetres de cada procés
-p	Mostra a més el PID de cada procés
-Z	Mostra a més contexte SELinux

<i>pidof nomProg</i>	Mostra el PID (o PIDs) associats al nom del procés indicat
-s	Només mostra un PID -si n'hi hagués més d'ún-

<i>pgrep expReg</i>	Mostra el PID/PIDs associats als processos el nom dels quals concordi amb l'exp reg indicada
-u usuari,usuari,...	Mostra el PID/PIDs que a més de concordar amb <i>expReg</i> , siguin dels usuari
-l	Mostra, a més del PID, la ruta del procés
-a	Mostra, a més del PID, la ruta del procés + paràmetres
-f	Mostra el PID/PIDs si <i>expReg</i> concorda amb el nom <u>o paràmetres</u>
-o	Si hi ha varis PIDs, mostra el més antic
-n	Si hi ha varis PIDs, mostra el més nou
-c	Conta els processos coincidents
-v	Mostra el resultat invers (els processos que NO coincideixen amb el patró)
-d ":"	Defineix el caràcter ":" com a separador entre PIDs (per defecte, és \n)

top

Mostra (una "pel·lícula" de l'estat de) tots els processos executant-se al sistema (incloent dimonis). Un "equivalent" similar podria ser *watch -n 3 "ps -e"*

La primera línia de la capçalera mostra, per defecte, l'hora actual, el temps que porta encés la màquina, la quantitat d'inicis de sessió oberts en aquest moment i la mitjana de càrrega en l'últim, 5 i 15 minuts. La mitjana de càrrega és un número que representa el número de processos en la cua d'execució de la CPU (és a dir, actius o esperant a ser executats). És una mesura més fiable que el %cpu per mesurar el rendiment real de la CPU ja que la mitjana de càrrega no té en compte els temps d'espera per I/O (discs, xarxa, bases de dades, etc). A la pràctica, si la mitjana de càrrega coincideix amb el número de CPUs del sistema, significa que aquestes CPUs tenen un ús perfectament ajustat a la demanda dels processos existents; si la mitjana és inferior, significa que alguna de les CPUs està sent infrautilitzada i si la mitjana és superior significa que hi ha processos sense atendre perquè estan forçats a haver-se d'esperar a què finalitzin els que ja s'hi estan executant. Les mitjanes a més temps d'un minut indiquen la tendència d'ús.

Aquesta línia és exactament la mateixa que la que es veu en executar la comanda *uptime* i també la primera que mostra la comanda *w*

Aquesta línia es pot fer desaparèixer (o tornar-la a mostrar visible, alternativament) pulsant, dins del *top*, la tecla "I"

La segona línia de la capçalera mostra, per defecte, el número total de processos i aquest número desglossat pels diversos estats que poden haver (R,S,T,Z)

La tercera línia de la capçalera mostra, per defecte, un còmput global de l'ús de la CPU, desglossat d'aquesta forma:

% de temps de cpu emprat en processos d'usuari (%us)

% de temps de cpu emprat en processos de kernel (%sy)

% de temps de cpu sense "treballar" (%id, d'"idle")

% de temps de cpu emprat en processos amb un nivell positiu de "niceness" (%ni, ho estudiarem properament)

% de temps de cpu que passen els processos esperant a que acabi una I/O -de disc, xarxa...- És temps malgastat (%wa)

% de temps de cpu emprat en atendre interrupcions hardware (%hi)

% de temps de cpu emprat en atendre interrupcions software -com les provocades pels drivers- (%si)

% de temps de cpu que una CPU virtual espera per entrar a la CPU real mentre l'hipervisor (KVM, generalment) està servint a una altra CPU virtual; només té sentit en sistemes virtualitzats. (%st, d'"steal time")

Aquesta línia (juntament amb la segona) es pot fer desaparèixer (o tornar-la a mostrar visible, alternativament) pulsant dins de *top* la tecla "t". Aquesta tecla també permet, seguint la roda de pulsacions, mostrar els % de temps d'una manera més gràfica, en forma de barres (de fet, hi ha dos tipus de barres, cadascuna visible amb una pulsació diferent de "t", però totes dues mostren el mateix: %us, %sy i total).

Aquesta línia es pot desglossar en estadístiques per CPUs individuals (en comptes de mostrar el còmput global com es fa per defecte) si es pulsa la tecla "1". Això es pot combinar amb la tecla "t" esmenada en el paràgraf anterior.

La quarta línia de la capçalera mostra, per defecte (i en KB), la quantitat total de memòria RAM de la màquina, la que està sent utilitzada en aquest moment, la que roman lliure i l'emprada en buffers+memòria catxé. Sempre es complirà que total=utilitzada+lliure+buff/cache.

Un "buffer" és una ubicació de memòria utilitzada per un procés en particular per emmagatzemar temporalment dades que esperen ser enviades a la seva ubicació permanent; són molt utilitzats per ajustar les possibles diferents velocitats de rebuda/enviament de dades que hi poden haver entre diferents elements del sistema, com ara la RAM, els discos durs, les tarjes de xarxa, etc. La memòria "catxé" és una ubicació de memòria utilitzada per emmagatzemar dades que es preveuen que es tornaran a fer servir aviat i/o freqüentment per part d'un o més processos indistintament; la idea és no haver d'accedir al disc cada cop que es vulgui llegir la mateixa dada ja que aquest és un procediment molt lent: així, si la primera vegada la dada en qüestió es guarda a la catxé, les següents vegades el seu accés serà molt més ràpid.

La cinquena línia de la capçalera mostra, per defecte, (i en KB), la quantitat total de memòria swap de la màquina, la que està sent utilitzada en aquest moment i la que roman lliure.

Aquestes línies són similars a les que es veuen en executar la comanda *free*, de la qual aconsello llegir el seu manual però de la qual només destacarem els paràmetres *-h* i *-s n°segons*

NOTA: Cal dir que aquesta eina mostra una columna extra anomenada "Shared" que representa la memòria usada per *tmpfs*

Aquestes línies es poden fer desaparèixer (o tornar-la a mostrar visible, alternativament) pulsant, dins del *top*, la tecla "m". Aquesta tecla també permet, seguint la roda de pulsacions, mostrar les quantitats rellevants de memòria d'una

manera més gràfica, en forma de barres (de fet, hi ha dos tipus de barres, cadascuna visible amb una pulsació diferent de "m", però totes dues mostren el mateix: % m. usada i m. Total.

NOTA: Per saber més sobre la capçalera de *top*, recomano llegir la secció "Summary Area Fields" del seu manual

Les columnes que es mostren per defecte són: PID, USER, PR, NI, VIRT, RES, SHR, S, %CPU,%MEM, TIME+, COMMAND. Les que encara no coneixem són PR (que indica la prioritat del procés), NI (que indica la "niceness" del procés) i SHR (que indica la quantitat de memòria RES -que és el mateix que RSS- potencialment compartible amb altres processos degut a què conté codi de llibreries comunes, etc). La columna TIME+ és igual a la columna TIME de la comanda *ps* però la seva precisió és major perquè arriba fins les centèsimes de segon.

Ja dins de la comanda *top*, es poden fer servir les següents tecles per ordenar els valors mostrats:

x : mostra la columna que es farà servir per ordenar. A partir d'aquí...
>: ...canvia la columna d'ordenació a la columna a la dreta de l'actual.
<: ...canvia la columna d'ordenació a la columna de l'esquerra de l'actual.
R : reverteix l'ordre (per defecte és descendent)

A més a més, hi ha diferents tecles-"dreceres":

P : ordena segons ús de CPU -per defecte-
M : ordena segons ús de memòria
N : ordena segons nº PID
T : ordena segons el temps que porten funcionant els processos

També es poden implementar filtres pulsant les següents tecles:

u : mostra només els processos de l'usuari que s'especifiqui
i : mostra només els processos actius (amb %cpu>0)
o: serveix per definir el filtre que es vulgui escrivint el nom d'una columna i un valor, així:
 COMMAND=nom
 %CPU>1,0
 ...
Per veure els filtres activats, cal pulsar CTRL+o
Per desfer qualsevol filtre que s'hagi establert prèviament amb o, cal pulsar "="

Altres tecles interessants són:

f: serveix per entrar en una pantalla especial que ens permet especificar dues coses:
 *Quines columnes es volen mostrar (de/sel.leccionant-les de la llista amb "d")
 *Quina columna es vol fer servir com criteri d'ordenació (sel.leccionant-la amb "s")
 Un cop haguem acabat, s'ha de pulsar "q" per tornar a la pantalla principal
c: alterna entre mostrar la ruta+paràmetres dels processos o només el nom de l'executable
V: alterna entre mode normal i mode arbre
d: modifica l'interval de temps en què s'actualitza la sortida de la comanda *top* (per defecte 3s)
W: grava la configuració de sortida actual de la comanda *top* (en el fitxer ~/.toprc)
h: mostra l'ajuda (un altre h mostra una segona pantalla)
y: ressaltar en negrita les línies corresponents als processos en estat "ready" en aquest moment
q: surt

Les tecles k (mata un procés preguntant abans el seu PID) i r (fa un renice preguntant el seu PID i la prioritat) les estudiarem més endavan.

D'altra banda, a l'hora d'executar la comanda *top*, també podem indicar alguns paràmetres interessants:

-d nº (similar a la tecla "d")
-u usuari (similar a la tecla "u")
-n nº (serveix per indicar el nº d'actualitzacions a mostrar abans de tancar *top*)
-b (serveix per mostrar les actualitzacions en pàgines diferents -tipus log- en comptes de refrescar la mateixa pantalla; útil combinat amb -n)
-p PID1,PID2,... (només mostra la informació dels processos indicats. Útil junt amb pgrep)
-H (mostra els threads)

htop

Comanda similar a *top* però més amigable (cal instal·lar-la)

Les seves tecles són autodescriptives: F2(molt interessant!), F3, F5, F6,F9...

Convé, per exemple, activar l'opció "Hide userland threads" a F2 per a què no es mostrin els threads com a processos independents (que és el que passa per defecte).

El seu arxiu de configuració és ~/.config/htop/htoprc (els canvis es guarden sols)

Pulsant "h" es mostra una ajuda descrivint els codis de colors i més tecles possibles (com per exemple "l" per mostrar els fitxers oberts pel procés sel.leccionat -com lsof-, "e" per mostrar les variables d'entorn usades pel procés sel.leccionat, "u" per mostrar només els processos d'un determinat usuari, "a" per mostrar només els processos adjuntats a un determinat core, "H" per veure els threads d'usuari(o no) y "K" per veure els thread del kernel (o no), "p" per veure la ruta absoluta dels processos (o no), etc)

NOTA: **VIRT** stands for the virtual size of a process, which is the sum of memory it is actually using, memory it has mapped into itself (for instance the video card's RAM for the X server), files on disk that have been mapped into it (most notably shared libraries), and memory shared with other processes. **VIRT** represents how much memory the program is able to access at the present moment. **RES** stands for the resident size, which is an accurate representation of how much actual physical memory a process is consuming. (This also corresponds directly to the **%MEM** column) **SHR** indicates how much of the **VIRT** size is actually sharable memory or libraries. In the case of libraries, it does not necessarily mean that the entire library is resident. For example, if a program only uses a few functions in a library, the whole library is mapped and will be counted in **VIRT** and **SHR**, but only the parts of the library file containing the functions being used will actually be loaded in and be counted under **RES**.

w

Similar a *who*, però a més diu quina comanda està executant cada usuari

La primera línia que mostra és idèntica a la primera línia que mostra la comanda *top*

Després apareix una línia per cada login obert informant de: l'usuari connectat, en quin terminal està, des de quin ordinador (si és un accés remot), des de quina hora, el temps que està funcionant el procés executat per l'usuari en qüestió, el JCPU (temps de cpu gastat per tots els processos d'aquell terminal), el PCPU (temps de cpu gastat per aquest procés concret) i quin procés concret es tracta.

-h : fa que no es mostri la primera línia general, ni les capçaleres de les columnes

usuari : només mostra la informació corresponent a l'usuari especificat

time comanda

Executa la comanda indicada i, seguidament, retorna el temps transcorregut en l'execució en mode "user", el temps gastat en mode "system" i el temps total des de que es va posar en marxa fins que ha finalitzat (temps que sempre serà superior a la suma dels dos anteriors perquè aquí s'ha de tenir en compte el temps que el procés roman a l'espera del seu torn de CPU).

kill -s n° PID ...

Atura el/s procés/sos amb el PID/s especificat/s fent servir una determinada senyal indicada pel seu número.

Per defecte, la comanda *kill* empra la senyal n°15 (SIGTERM), la qual pot no funcionar degut a què el procés ha d'haver sigut programat específicament per atendre aquesta ordre (és a dir, és possible que el procés ignori la senyal que demana que es mori). Per assegurar-se de què el procés es mor, cal llavors enviar la senyal n°9 (SIGKILL), que mai és ignorable i, per tant, sempre aconsegueix matar al procés en qüestió (perquè ho fa d'una forma "brusca", sense donar temps al programa a "reaccionar"). Una altra senyal interessant és la n°2 (SIGINT), que és la que es genera en pulsar CTRL+C (també pot ser ignorada, però).

A banda de les senyals que maten, també són útils les senyals que aturen processos (estat "T"). Aquestes senyals són la n°19 (SIGSTOP) i la n°20 (SIGTSTP). La diferència entre elles és que la primera pot ser ignorada pel procés en qüestió però la segona no. Per desfer l'"aturament" i tornar el procés a primer pla cal enviar la senyal n°18 (SIGCONT).

NOTA: Una manera alternativa a la comanda *kill* per enviar la senyal n°19 és pulsant CTRL+Z sobre el terminal on s'estigui executant en aquest moment el procés a aturar. Una alternativa a la senyal n°18 són les comandes *fg* o *bg*

Obtindreu més informació sobre els diferents tipus de senyals existents i les seves aplicacions executant *man 7 signal*

killall -s n° nomprog ... Atura tots els processos que coincideixin amb el nom indicat fent servir una determinada senyal indicada pel seu número.

- I No diferencia entre minúscules i majúscules en el nom del programa
- r Interpreta el nom del programa com una expressió regular
- i Pregunta interactivament per cada procés concret que correspon a la comanda si es vol matar
- v Mode verbós

pkill -s n° expReg Atura tots els processos el nom del qual coincideixin amb l'expressió regular escrita, fent servir una determinada senyal indicada pel seu número.

- u usuari A més s'ha de complir que els processos coincidents hagin sigut executat perrr l'usuari indicat
- t terminal A més s'ha de complir que els processos coincidents s'executin al terminal indicat
- n Només mata el procés més nou de tots els possibles coincidents
- o Només mata el procés més antic de tots els possibles coincidents

trap "ordres;aexecutar;quanesdetecta;lasenyal;especificada" n°senyal ... Aquesta comanda és útil sobre tot en scripts. Serveix per "atrapar" senyal. Per exemple: si s'escriu al començament d'un script l'ordre *trap "echo adeu; exit" 2*, quan es pulsi CTRL+C durant la seva execució, apareixerà el missatge "adeu" i llavors acabarà perquè s'ha afegit l'exit.

*¿Què passarà doncs si executem la següent línia?:

trap "echo No em mataras" 2 15; while true do; sleep 60; done

*Si simplement es vol ignorar les senyals sense fer res més, es pot escriure *trap "" 2 15* (per ignorar en aquest cas les n°2 i n°15) o també es pot escriure *trap : 2 15*

*Per tornar a restaurar l'"escolta" a les senyals, es pot fer *trap - 2 15*

*Per veure quines senyals estan "interceptades al shell/script actual: *trap -p*

comanda & El símbol & serveix per executar una comanda de llarga duració i poder seguir treballant amb la consola. Si no s'escriu, no es podria fer servir la consola fins que la comanda acabés la seva execució. En paraules tècniques: inicia el procés directament en segon pla (és a dir, en mode "no interactiu", com si fos un demoni). Fixeu-vos que en executar la comanda així es mostra en pantalla entre claudàtors el número de treball de fons i a la seva dreta, el seu PID.

Si ja s'està executant un procés en primer pla i el que es vol és passar-lo a segon pla, el procediment és el següent: primer caldrà aturar-lo amb CTRL+Z i seguidament caldrà reanudar-lo (ja en segon pla) amb la comanda *bg %n* (on %n és el n° de treball de fons, i és un número que en fer el CTRL+Z apareix entre claudàtors).

Aquest número %n també es pot conèixer usant la comanda *jobs*, la qual mostra els processos parats (Stopped) i també els que estan en segon pla (Running) d'aquest shell en concret.

Per tornar a passar un procés en segon pla a primer pla, caldrà executar la comanda *fg %n*

La comanda *jobs* mostra amb un signe "+" el procés, d'entre tots els que hi ha a la llista, que serà posat en primer pla per defecte si s'executa la comanda *fg* sense paràmetres (correspon al procés que fa més temps que és a la llista); el signe "-" indica el següent procés, darrera el "+", que passaria a primer pla si s'executa *fg* sense paràmetres (correspon al segon procés que fa més temps que és a la llista). El paràmetre *-l* de la comanda *jobs* mostra els Pids.

La comanda *kill* també pot fer servir el número %n com a alternativa al PID

nohup comanda & Si es surt de la sessió/terminal (senyal n°1, SIGHUP), *nohup* fa que la comanda indicada continui executant-se (sempre que no s'apagui la màquina). El resultat de l'execució es guarda automàticament en l'arxiu "nohup.out", ubicat al directori on s'executi la comanda, o si no es pot escriure, a HOME (si és que no es redirecciona explícitament).

*Si es desitja aplicar un *nohup* sobre un procés que ja està en marxa (en segon pla), s'ha de fer així: *disown -h %n°job > resultat*

*Un *nohup* "caser" seria fer: *trap "" 1 15 ; comanda) &*

Comandes Linux (II)

***Treballar amb punts de muntatge, particions i fitxers:**

mount Mostra els punts de muntatge existents. A nosaltres només ens interessaran els que corresponen als dispositius /dev.. En aquests casos, la informació que obtenim és: ruta del dispositiu /dev, ruta del punt de muntatge corresponent, sistema de fitxer en què està formatat el dispositiu i, entre parèntesis, opcions diverses de muntatge

Una comanda que també serveix per mostrar els punts de muntatge però d'una forma molt més clara és *findmnt* : aquesta comanda per defecte mostra quatre columnes: ruta del punt de muntatge, ruta del dispositiu, sistema de fitxers i opcions de muntatge.

[-T] rutaPuntMuntatge : Mostra només els punts de muntatge amb la ruta indicada
[-S] /dev/disp1 : Mostra només els punts de muntatge del dispositiu indicat
-t tipussistfixer : Mostra només els punts de muntatge que tinguin el sistema de fitxers indicat
-O opcio,... : 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
-s : Només mostra els dispositius muntats mitjançant */etc/fstab* . Amb *-m* és *mstab*

Altres comandes interessants:

lsblk [/dev/disp|disp1] Mostra els dispositius de bloc en forma d'arbre. Les columnes per defecte 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, punt munt.
-o col1,col2... +colaafegir,... : Especifica les columnes desitjades (*lsblk -help* mostra llista completa)
-p : Mostra la ruta absoluta dels dispositius (normalment *"/dev/xxx"*)
-P : Fa que la sortida, en comptes de ser en columnes, sigui en línies
-x col1 : Ordena per la columna indicada

blkid [/dev/disp|disp1] Mostra els dispositius de bloc indicant la seva ruta de dispositiu, la seva etiqueta, el seu UUID i el seu sist. de fitxers En comptes d'indicar com a paràmetre la ruta del dispositiu a mostrar es pot especificar *-L label* (per indicar la seva etiqueta) o *-U uuid* (per indicar el seu format)

findfs {LABEL|UUID|}=valor Troba la ruta de dispositiu corresponent a l'etiqueta o uuid indicat

mount -t tipussistfix /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 per muntar 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), sysfs (conté informació relativa als dispositius).

En comptes d'indicar la ruta del dispositiu es pot escriure
-L label o *-U uuid*

-o ... Serveix per indicar diferents opcions de muntatge, 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.

umount /ruta/punt

Desmunta el punt de muntatge

-l : El desmuntatge es farà quan tots els recursos usats pel dispositiu siguin lliberats
-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)

Per a què es puguin muntar automàticament les particions en arrencar el sistema (com a mínim, el punt de muntatge / ha d'existir sempre) ha d'aparèixer la línia corresponent dins l'arxiu **/etc/fstab**. El format de les línies és:

/dev/disp1 /ruta/punt tipussistfitx opcions 0 0

Els dispositius també es poden especificar amb la seva etiqueta: (*LABEL=...*) ó amb el seu uuid (*UUID=...*)

Els tipus de sistemes de fitxers són els ja coneguts i també "auto", per fer que sigui el sistema qui l'endevini

Algunes de les opcions (si hi ha vàries han de estar separades per comes) són:

*"ro" (muntar com a només-lectura; el contrari és "rw")
*"noexec" (per evitar que es puguin executar programes en aquella partició; el contrari és "exec")
*"user" (per a què els usuaris que no siguin *root* puguin muntar aquella partició; el contrari és "nouser")
*"users" (per al mateix que "user" però a més permetent que qualsevol usuari pugui desmuntar el que ha muntat un altre usuari -l'opció "user" només permet desmuntar un punt al mateix usuari que el va muntar-
*"noauto" (per "deshabilitar" aquesta entrada però permet muntar-la manualment; el contrari és "auto")
*"sync" (per a què hi hagi sincronització a l'hora de gravar les dades al disc o no -o sigui, no hi ha memòria "dirty"- ; el contrari és "async"). En el mode sync no és molt preocupant la desconexió en calent; en el mode async, abans de desconnectar el dispositiu és bona idea executar la comanda *sync* (l'"extracció segura")
*"noatime" (per a què cada cop que s'accedeixi a un fitxer o carpeta del dispositiu -encara que sigui només lectura- no es guardi una marca de temps; d'aquesta manera, la I/O serà més ràpida; el contrari és "atime").
*"nodiratime": (per a què cada cop que s'accedeixi a una carpeta del dispositiu -encara que sigui només lectura- no es guardi una marca de temps; d'aquesta manera, la I/O serà més ràpida; el contrari és "atime").
*"relatime": (per a què només es guardi una marca de temps d'accés (és a dir, un nou valor atime quan, o bé es compleix que (*mtime|ctime*)>*atime* o bé quan el valor del atime actual és més antic d'un dia)
*"defaults" (un conjunt de diferents opcions. Concretament: *rw,exec,auto,nouser,async, suid,dev*)
NOTA: Per saber més opcions, es recomana llegir la pàgina del manual de la comanda *mount* i en especial, els apartats "mount options for..." i "filesystem-independent mount options"

El primer número (generalment un 0) que apareix després de les opcions és la freqüència amb la què s'executa el programa de còpies de seguretat *dump*: 0 és mai, 1 és diari, 2 és cada dos dies, etc.

El segon número que apareix després de les opcions estableix l'ordre amb què els dispositius seran xequjats per *fsck* en muntar-se: 0 és que no es xequja mai, 1 és el primer dispositiu de tots, 2 és el segon, etc.

El paràmetre -a de la comanda *mount* recarrega la informació continguda a l'arxiu */etc/fstab* sense que calgui reiniciar la màquina.

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

Carpets que estaria bé tenir en particions separades:

/home
/boot (on està el kernel)
/var (on estan els logs, temporals, mails, cues d'impressió...)
/tmp (on estan els arxius temporals)

Hi ha directoris aptes per ser compartits (no tenen informació específica de l'equip o dels programes locals) o no, i directoris estàtics (les dades no canvien sense la intervenció directa de l'administrador: documentació, programes...) ó variables (bases de dades, correus...). Així:

Apte per compartir i estàtic: poden ser muntats en només lectura i exportats (/usr, /opt)
Apte per " i variable: poden ser muntats en escritura i exportats (/home, /var/mail)
No apte per " i estàtic: muntats localment en només lectura (/etc,/boot)
No apte per " i variable: muntats localment (/run)

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

fdisk /dev/disp Permet manipular particions del dispositiu (disc!) especificat (com si fos el Gparted però amb comandes. Per saber quines comandes es poden fer servir, consulta l'ajuda del programa però les més bàsiques són:

m: veure opcions
o : crear la taula de particions de tipus **MSDOS**
g : crear la taula de particions de tipus **GPT**
n: crear nova partició (p: primària (1-4), e: extensa...i a partir d'aquí anar responnent)
a: fer-la activa
d: eliminar partició que es digui tot seguit
p: visualitza estat actual de la taula de particions (com fdisk -l)
w: surt desant els canvis
q: surt sense gravar

parted [/dev/disp] Alternativa a *fdisk*
select /dev/disp : Selecciona el dispositiu sobre el qual treballar
mklabel {msdos|gpt} : crea la taula de particions de tipus MSDOS o GPT
mkpart {primary|extended|logical} n°MBiniciFrom0 n°MBfinalFrom0 : crea partició
set n° boot on : establir la partició indicada com activa
rm n° : Elimina de forma interactiva la partició indicada
print : mostra la llista actual de particions del disc
quit: surt

mkfs.{ext4|vfat|xfs|jfs} /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

-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 el tamany (en bytes) dels blocs per aquesta partició ExtX o XFS

-S tamany : Estableix el tamany (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)

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".

Els valors per defecte, si no es canvien per algun paràmetre, els pren *mkfs* de l'arxiu de configuració */etc/mke2fs.conf*

Les etiquetes i els UUIDs es poden canviar a posteriori d'haver formatat una partició amb les següents comandes:

tune2fs { -L etiqueta | -U valoamà|random|time|clear } /dev/disp1 : Particions ExtX (desmuntades)

xfs_admin { -L etiqueta | -U valorama|generate } /dev/disp1 : Particions XFS

fatlabel /dev/disp1 etiqueta : Particions FAT32 (desmuntades)

swaponlabel { -L etiqueta | -U valoramà } /dev/disp1 : Particions swap (desactivades amb *swapoff /dev/disp1*)

`fsck.{vfat|ext4|xfs|jfs} /dev/disk1`

Fa un xequi del sistema de fitxers de la partició especificada.
A particions NTFS s'ha d'usar la comanda *ntfsck*

- p : reparació automàtica, sense fer preguntes
- y : sempre respon que sí a les preguntes
- V : mode verbós
- n : no es faci cap canvi al sistema de fitxers (només informa, no repara)
- c : comprova si hi ha blocs erronis executant internament la comanda badblocks (només ext4)
- f : força la comprovació encara que el sistema estigui marcat com a net (només ext4)
- D : optimitza l'accés al contingut dels directoris, reindexant-lo i reordenant-lo (només ext4)

NOTA: En XFS està `xfs_check` i `xfs_repair`, que són realment els que fan la feina: `fsck.xfs` no fa res

La comanda *fsck* s'executa automàticament en iniciar si es va apagar la màquina sense desmuntar el sistema d'arxius.
En concret, comprova que:

- Un bloc no pertanyi a diferents arxius (inodes)
- No hi hagin blocs lliures que apareguin com ocupats
- No hi hagin blocs ocupats que apareguin com a lliures
- El contador d'enllaços durs no sigui erroni
- No hi hagi clústers perduts -> /lost+found
- No hi hagi inodes ilocalitzables
- No hi hagi dades inconsistents a la taula d'inodes)
- *Per Xfs hi ha `xfs_check` i `xfs_repair`, i també `xfs_growfs`.

`df -h [/dev/disk1]`

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
- a : Inclou a la llista els pseudo-sistemes de fitxers
- 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)

`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.
Un programa gràfic similar (però més versàtil) és el *baobab*

- s : Només mostra el tamany total de la carpeta
- x : No entra en directoris que pertanyin a un altre sistema de fitxers (punts de muntatge)
- b : Format 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.

`dd if=/dev/sda1 of=/dev/sda2`

Fa una còpia bit a bit de la partició 1 a la 2. La partició 1 també podria ser el dispositiu `/dev/zero` ó `/dev/urandom`, per exemple.

`dd if=/dev/sda of=/dev/sdb`

Fa una còpia bit a bit del disc sda en sdb.

`dd if=/dev/sda1 of=arxiu.iso`

Genera una iso bit a bit de la partició 1 . També podria ser a la inversa

*(Alguns) paràmetres del `dd`:

`bs=tamanyblocacopiar{KMG}` `count=nºbloscacopiar` Augmentant el tamany del bloc es pot accelerar el procés de copia (es pot posar els sufixes k,m...).

`skip=nºblocsquesaltenalorigenabansdeferlacopia`

`seek=nºblocsquesaltenaldestiabansdeferlacopia`

`conv=noerror,sync` El que fa és no detenir-se en detectar els errors de l'entrada (`noerror`) i substituir-los per zero a la sortida (`sync`). Altres opcions són: `notrunc` i `fsync`.

`status=progress` Mostra una barra de progrés mentre es realitza la còpia

*(Alguns) exemples del `dd`:

`dd if=/dev/zero of=/dev/sda`

(esborra un disc; `dd if=/dev/sda | hexdump -C | grep [^00]`)

`dd if=/dev/urandom of=/dev/sda`

(escriu dades aleatòries al disc, -també pot ser a un fitxer-)

`dd if=/dev/sr0 of=arxiu.iso`

(genera una iso d'un cd)

`dd if=/dev/sda | sed 's/Ana /Maria/g' | dd of=/dev/sda` (substitueix totes les ocurrencies)

`dd if=/dev/zero of=/dev/ram7 bs=1 count=16384` (crea un disc de memòria. Un cop formatat `/dev/ram7` amb `mkfs`, i muntat en una carpeta, es podrà fer servir com una partició extra MOLT més ràpida que una normal, però atenció, el contingut en anar-se la corrent és volàtil!. En aquest sentit també es podria usar com partició un fitxer, així:

`dd if=/dev/zero of=file.bin bs=1k count=100` (existeix una alternativa: `fallocate [-o offset] -l tamany fitxer`)

`mkfs.ext4 file.bin`

`mount -t ext4 file.bin /mnt`

... copiar a /mnt el necessari

`umount /mnt`

`xorriso -as mkisofs -R -J -o arxiu.iso carpeta` Genera una iso amb el contingut de la carpeta especificada

`xorriso -as cdrecord dev=/dev/sr0 arxiu.iso` Grava un arxiu iso en un cd en blanc (dins la gravadora `/dev/sr0`)