

Comandes Linux (II)

***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

-H Mostra l'arbre de processos. També està l'opció --forest

-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 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

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 <code>cat /dev/urandom > /dev/null</code>
"%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 $pcpu=(time/etime)*100$).
"s" o "state"	: estat del procés, el qual pot ser: R (running) S (sleeping interrompible -perquè s'espera algun event-) D (sleeping ininterrompible -perquè s'espera una resposta d'I/O-) Z (zombie: mort però el pare no ho sap i per això encara està voltant; l'acabarà eliminant el kernel) T (stopped, per una senyal SIGxxx o per estar sent depurat, rar de veure) SW (swapped; rar de veure)

<i>ps</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

<i>pidof</i> nomProg	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</i> expReg	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 + paràmetres
-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
-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 "l"

strace -e open uptime -o fichero : llamada al sistema, parámetros y lo que devuelve (-1)

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

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- compartida amb altres processos). 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-"drecceres":

P : ordena segons ús de CPU -per defecte-
M : ordena segons ús de memòria
N : ordena segons n° 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 endavant.

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

-d n° (similar a la tecla "d")
-u usuari (similar a la tecla "u")
-n n° (serveix per indicar el n° 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)

<i>htop</i>	Comanda similar a <i>top</i> però més amigable (cal instal·lar-la) Les seves tecles són autodescriptives: F2(molt interessant!), F3, F5, F6,F9... <i>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).</i> El seu arxiu de configuració és ~/.config/htop/htoprc (els canvis es guarden sols)
<i>w</i>	Similar a <i>who</i> , 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 <i>top</i> 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
<i>time comanda</i>	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).
<i>kill -s n° PID ...</i>	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.

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

<i>killall -s n° nomprog ...</i>	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
<i>pkill -s n° expReg</i>	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	En comptes d'escriure <i>expReg</i> , mata tots els processos d'aquell usuari
-t terminal	En comptes d'escriure <i>expReg</i> , mata tots els processos d'aquella terminal

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'executa 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) &*

Carpeta /proc:

11.-La carpeta /proc (a l'igual que passa amb la carpeta /dev) no és una carpeta "de veritat". En realitat és una manera que té el kernel d'accedir a determinada informació que manega en temps real, a través d'un mecanisme que imita un sistema de fitxers (podeu consultar la pàgina "man proc" perquè és super-completa). Observa el contingut dels següents "fitxers" i digues què contenen:

procfs, a "pseudo" file system dynamically generated by the kernel to provide information about the status of entries in its process table

```
/proc/cpuinfo
/proc/meminfo
/proc/partitions
/proc/filesystems
/proc/mounts
/proc/version
/proc/cmdline-->/etc/default/grub
/proc/uptime (temps encés, temps idle suma de los cores)
/proc/modules (pregunta al professor)
/proc/swaps (pregunta al professor)
```

Contingut de les carpetes /proc/nºPID (en concret, els arxius cmdline i status, per exemple)

```
tr '\0' '\n' < /proc/.../cmdline (o cwd o exe) ...
```

Contingut de les carpetes /proc/acpi/battery, /proc/acpi/ac_adapter, /proc/acpi/processor i /proc/acpi/fan

En realitat es tracta d'una interfàç que permet l'accés a la informació que gestiona el kernel en temps real, a través d'un mecanisme que imita un sistema de fitxers. La pàgina man proc és super-completa

/proc/PID : conté informació relativa al procés concret que s'està executant amb el PID concret

cmdline: conté la comanda que inicià el procés corresponent, amb tots els paràmetres

cwd : symlink al directori de treball actual del procés; exe: links al procés executable; root: al seudirectori root

environ: mostra les variables d'entorn pel procés

exe: symlink a l'arxiu executable original, si encara existeix

fd: mostra els descriptors dels fitxers usats pel procés

root: symlink a la carpeta arrel tal com és vista pel procés (normalment és / a no ser que s'hagi chrooteat)

maps, statm, mem : té a veure amb la memòria utilitzada pel procés

oom_score: <http://totaki.com/poesiabinaria/2010/04/cuando-un-proceso-se-come-la-memoria-de-nuestro-sistema/> --> /proc/sys/vm/oom_victim -> sysctl kernel.sysrq=1 (alt+sysrq+f)

stat, status : dóna informació sobre l'estat del procés (status és més clar que stat). De tota la informació, las variables relacionadas con el uso de memoria son las siguientes:

- **VmPeak:** Tamaño máximo que el proceso ha sido capaz de direccionar.
- **VmSize:** Tamaño máximo que ha direccionado sin contar regiones reservadas.
- **VmLck:** Tamaño de las páginas bloqueadas (que no se pueden swapper).
- **VmHWM:** Tamaño máximo que de memoria que ha ocupado el proceso.
- **VmRSS:** Tamaño de todas las paginas que pertenecen al proceso.
- **VmData:** Tamaño de memoria del proceso para datos sin contar la memoria compartida ni el tamaño de la pila.
- **VmStk:** Tamaño de la pila.
- **VmExe:** Tamaño de código ejecutable.
- **VmLib:** Tamaño de código ejecutable excluyendo el código del proceso: El tamaño del código de librerías.
- **VmPTE:** Tamaño de la tabla de páginas del proceso

task: directori que conté enllaços durs a les tasques iniciades pel procés (que seria llavors el pare)

limits : mostra els límits establerts pel procés (d'ús de Cpu, d'obrir fitxers, etc)

mountinfo, mounts, mountstats

net/ : ...

/proc/cpuinfo : conté informació sobre el tipus de processador de la màquina . registres, catxé, uname -m

grep "processor" /proc/cpuinfo | wc -l et dóna el número de nuclis del procesador

grep "physical id" /proc/cpuinfo | uniq | wc -l et dóna el número de procesadors físics

grep "cpu cores" /proc/cpuinfo et dóna el número de cores que tenen els procesadors físics. Si els processadors són single-core i el número de procesadors virtuals és major que el número de procesadors físics, vol dir que la cpu està usant hyper-threading, la qual està suportada si el flag "ht" apareix i s'està utilitzant un kernel SMP

grep "cache_alignment" (ó "clflush size") /proc/cpuinfo et diu si el processador és de 32 ó 64 bits. ¡No, en realitat es veu si és de 64bits si apareix el flag "lm" -"long mode"-!! (en cpufeature.h se expliquen los distintos flags)

`grep "svm|vmx" /proc/cpuinfo` : si sale alguna de las dos ("svm" para Amd y "vmx" para Intel), se tiene una cpu que permite el uso de KVM (virtualización por hardware).
`grep "cpu MHz" /proc/cpuinfo` : indica la velocidad actual del procesador. Amb "bogomips" es veu el nº d'instruc/s

Interesante ver esto (`watch -n 0.1 "cat /proc/cpuinfo | grep -i mhz"`) mientras se hace algo intensivo
Más detalles puñeteros: <http://www.binarytides.com/linux-check-processor/>
`lscpu`

`nproc`
`hwloc` interesante!!!

`man proc`
`mount | grep proc`
`lscpu --> MHz lm svm|vmx "processor" (nuclis) "physical id"(físic) "cpu cores" (nucli/físic)`

`/proc/meminfo`: conté informació sobre el tipus de memòria de la màquina. Aquesta informació pot ser entre altres:

- MemShared : sempre val 0, està per raons de compatibilitat
- Buffers : memòria en buffers, inútil casi en les estadístiques actuals
- Cached : memòria en catxé de disc menys SwapCache
- SwapCache: memory that once was swapped out, is swapped in but still also is in the swapfile. It saves i/o
- Active: zona de la memòria catxé on estan les dades més demandades
- Inact_clean: part de la memòria catxé inactiva (Zona divisió) fàcilment sobreescrible
- Inact_dirty: part de la memòria catxé inactiva difícil de sobreesciure (arxius que no s'han escrit encara, p.ex)
- Inact_target: umbral que quan es passa el kernel no mou pàgines de la memòria activa a la inactiva
- HighTotal: memòria Ram per sobre de 860M
- LowTotal: memòria Ram per sota de 860M, que el kernel pot direccionar directament sense trucs
- LowFree: evident

Dirty!!

- SwapTotal: quantitat total de memòria catxé física
- SwapFree: quantitat total de memòria catxé lliure
- Committed_AS: estimació de la quantitat de Ram necessària per a què no falti mai per la càrrega actual

`/proc/swaps` : conté la memòria swap reconeguda pel kernel
`/proc/partitions` : conté les particions reconegudes pel kernel (nodev)
`/proc/filesystems`: conté informació sobre els sistemes de fitxers disponibles (reconeguts pel kernel)
`/proc/mounts` : conté la llista de sistemes de fitxers que estan muntats actualment
`/proc/self/mountinfo`: contains information about mountpoints hierarchy as well as chronological order.
`/proc/version` : versió del kernel i versió del compilador utilitzat per generar-lo (`uname -a`)-->/sys/kernel/ostype, osrelease, version
`/proc/cmdline`: paràmetres que se li han passat al kernel en arrencar
`/proc/modules` : conté la llista de mòduls carregats actualment pel kernel
`/proc/devices`: conté la llista dels diferents tipus de dispositius de caràcter i bloc que reconeix el kernel
`/proc/interrupts`: línies IRQ assignades a dispositius. L'última columna és el mòdul del kernel que se n'encarrega. Han d'aparèixer allà el número total de Cpus que hi ha al sistema, en forma de columnes.
`/proc/ioports`: ports d'e/s assignats a dispositius. `/proc/dma` mostra els canals DMA en ús.
`/proc/irq`: hi ha una subcarpeta per cada Irq (0,1,2...)
`/proc/kcore`: representa el contingut de la memòria RAM. Si s'envia a `/dev/dsp` la Cpu fa molt soroll!
`/proc/uptime` : conté el número de segons que està el sistema encés, i el número de segons que està sense fer res (idle).
`/proc/loadavg` : mitjana de nº de processos planificats per executar-se en el l'últim minut, 5m i 15m, i el nº de processos¿R+D?/total lastPIDused ---> sleep &y se ve el PID que se le asigna
`/proc/stat` : estadístiques des de l'últim inici de sistema. You can try to use `/proc/stat` to calculate CPU utilization.

Comandes Linux (III)

***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.

rutaPuntMuntatge : Mostra només els punts de muntatge amb la ruta indicada
/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

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,... :Especificar les columnes desitjades (*lsblk -help* mostra llista completa)

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 tipussistfixx 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 varies 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 desconectar el dispositiu és bona idea executar la comanda *sync* (l'"extracció segura")
- *"noatime" (per a què cada cop que s'accedeixi a un fitxer 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").
- *"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/mtab** 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*.

Carpetes 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 escriptura 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)

`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=nºbloccacopiar` Augmentant el tamany del bloc es pot accelerar el procés de copia (es pot posar els sufixes k,m...).

`skip=nºbloccsesaltenalorigenabansdeferlacopia`

`seek=nºbloccsesaltenaldestiabansdeferlacopia`

`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 ocurrences)

`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`

`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)

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`.

fdisk -l [/dev/disk] 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/disk 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
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

mkfs.{ext4|vfat|xfs|jfs} /dev/disk1 Formateja amb el sistema de fitxers indicat la partició especificada

- L etiqueta : Estableix una etiqueta a la partició
- b tamany : Estableix el tamany (en bytes) dels blocs per aquesta partició
- c : Fa una comprovació prèvia de blocs
- v : Mode verbós
- F {12|16|32} : Indica el tipus de sistema a usar:fat-12,fat-16,fat-32 (només per vfat)

Per saber el tamany concret d'un bloc d'una determinada partició ja formatada, es pot fer: *blockdev --getbsz /dev/disk1*
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

df -h [/dev/disk1] Mostra les particions, l'espai total, disponible i ocupat i els seus punts de muntatge

- T : També mostra el format dels sistemes de fitxers
- i : En comptes d'informar sobre l'espai en disc, informa sobre l'ús dels inodes

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)

genisoimage -R -J -o arxiu.iso carpeta Genera una iso amb el contingut de la carpeta especificada

wodim -v dev=/dev/sr0 arxiu.iso Grava un arxiu iso en un cd en blanc (col.locat a la gravadora /dev/sr0)

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

<i>gzip unarxiu</i>	Comprimeix l'arxiu (anomenant-lo "unarxiu.gz") amb l'algoritme GZ
<i>bzip2 unarxiu</i>	Comprimeix l'arxiu (anomenant-lo "unarxiu.bz2") amb l'algoritme BZ
-1 ... -9	Especifica el nivell de compressió: -1 = menys (però més ràpid) i -9 més (més lent)
-t	Comprova la integritat dels fitxers
-v	Mode verbós
-r	Si s'especifica una carpeta en comptes d'un arxiu, comprimeix el seus arxius interns
-l	Només mostra informació sobre l'arxiu especificat i l'eventual resultat comprimit

<i>gunzip unarxiu.gz</i>	Descomprimeix l'arxiu GZ (anomenant-lo "unarxiu")
<i>bunzip2 unarxiu.bz2</i>	Descomprimeix l'arxiu BZ (anomenant-lo "unarxiu")
-v	Mode verbós
-t	Comprova que tot surti bé

Una alternativa a *gunzip* és fer: *gzip -d arxiu.tar.gz*. Descomprimeix igual.
 El compressor *bzip2* té el paràmetre *-c* per comprimir i el *-d* per descomprimir.
 El compressor *compress* (o *ncompress*) també té l'opció *-d* per descomprimir (ó *uncompress*), i *-v* per mode verbós.
 El compressor *zip* també es pot fer servir per comprimir i descomprimir (i amb el paràmetre *-e*, també es pot afegir una contrasenya al paquet *zip* per protegir-lo)

<i>tar -cf unarxiu.tar unacarpeta</i>	Crea un arxiu TAR amb el contingut d'"unacarpeta" (no comprimeix)
-z	Comprimeix amb l'algoritme GZ. L'arxiu es denominarà "unarxiu.tgz".
-j	Comprimeix amb l'algoritme BZ. L'arxiu es denominarà "unarxiu.tbz"
-p	Manté els permisos i el propietari originals
-P	En crear el tar especificant rutes absolutes, es mantenen aquestes (no es treu la / del principi)
-L	Segueix els enllaços simbòlics
-v	Mode verbós

L'opció *f* és imprescindible si s'indica un arxiu "arxiu.tar" però es pot posar un arxiu de dispositiu, o bé la sortida/entrada estàndard (escrivint llavors el signe "-"). Per exemple, per tal d'entubar el resultat a un altre lloc, es pot fer: *tar cf - . | (cd /una/ruta; tar xvf -)*

Es pot fer el tar normal i posteriorment utilitzar *gzip* (així, *gzip arxiu.tar*) per a comprimir-ho (és a dir, en dues passes separades). El fitxer resultant tindrà l'extensió *tar.gz*. Per descomprimir aquest tipus d'arxius s'ha d'utilitzar el descompressor *gunzip* (o *bunzip2*) per obtenir el fitxer *.tar* i després s'ha d'utilitzar l'opció corresponent del tar normal.

<i>tar -rf unarxiu.tar unnouarxiu ...</i>	Afegeix a un arxiu TAR ja existent els arxius indicats
<i>tar -uf unarxiu.tar unnouarxiu ...</i>	Afegeix a un tar ja creat, només si són més actuals que els ja presents

<i>tar -xf arxiu.tar</i>	Extreu el contingut de l'arxiu TAR a la carpeta actual
-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
-C /ruta/carpeta/on/sextreuen/els/fitxers	(si no es posa res, s'extreuen a la carpeta actual)
-m	Canvia la data de modificació dels fitxers extrets per la data actual
-v	Mode verbós

<i>tar -tf arxiu.tar</i>	Mostra el contingut de l'arxiu TAR
--------------------------	------------------------------------

Recerca de fitxers

updatedb && locate nomfitxer Busca el fitxer indicat al disc dur

locate:

paraula

(busca qualsevol coincidència de la paraula dins la ruta dels fitxers presents al sistema, a partir d'una base de dades pròpia, generada amb updatedb (comanda que es configura al seu torn a /etc/updatedb.conf, entre altres coses per especificar gràcies a la línia PRUNEPATH a quins directoris no es buscarà). Es poden fer servir comodins. I expressions regulars si es fa servir el paràmetre -r)

-q : no mostra errors (tipus "permís denegat"...)

-n 20 : només mostra 20 resultats

find unacarpeta -name "hola"

Mostra les rutes dels fitxers anomenats "hola". Es poden usar comodins.

La recerca comença a partir de "unacarpeta" fins al final, però no es busca fora d'aquesta carpeta

-type d Només busca directoris, no fitxers

-size +1M Busca fitxers amb més d'1 mega de tamany. Amb un "-", buscaria tamanyos menors

-mtime +2 Busca fitxers modificats fa més de 2 dies. Amb un "-" buscaria en menys de 2 dies

-mmin +2 Busca fitxers modificats fa més de 2 minuts. Amb un "-" buscaria en menys de 2m

-perm 444 Busca fitxers que tenen els permisos especificats

-user pepito Busca fitxers que tinguin el propietari especificat

-samefile /ruta/fitxer Busca fitxers que siguin hard links de l'indicat

Es poden utilitzar diferents paràmetres a la vegada per fer recerques més complexes, unint-los amb la partícula -or ("o") i -and ("i"). Per exemple: find / -name "hola" -a -perm 654 busca tots els fitxers i directoris que s'anomenen "hola" i que a més tenen els permisos 654. També es pot fer servir la partícula -not per negar una condició; per exemple: find / -not -name "hola" busca tots els fitxers que no s'anomenen "hola".

-delete

-exec

find ~/ -xdev -inum 1234567 (ls -i fichero.txt)

https://www.gnu.org/software/findutils/manual/html_node/find_html/Hard-Links.html

The reason for using the filesystem's mount point is that hard links are restricted to reside in the same filesystem as the physical file. This means that *file2* and *file3* must be in the same filesystem as *file1*. Therefore, to find ALL hard links to a file, you must start search at the mount point of the filesystem. The -xdev option instructs *find* NOT to traverse directories in other filesystems

find ./ -name ".csv" | parallel echo*

rutaIniciRecerca ABAST OPCIO ACCIO

On ABAST pot ser algun d'aquests:

-maxdepth n° : limita la recursivitat de la recerca al número de subdirectoris indicats

-mindepth n° : realitza la recerca sense baixar del nivell de subdirectorí especificat

-xdev : no busca en punts de muntatge que apuntin a sistemes de fitxers diferents del de la ruta d'inici.

On OPCIO pot ser alguna d'aquestes (o la seva combinació utilitzant els connectors -a ó -o):

-perm 444

-name "nomarxiu" (*admet comodins bash*). També està la opció case-insensitive -iname

-user propietari

-uid uid

-group propietari

-gid gid

-type {d|f|l} (*només buscar directoris, arxius o links, respectivament*)

-mtime [+|-]n (*s'ha modificat fa més o menys de n dies, comptant 24h enrera des de l'hora actual*)

-mmin [+|-]n (*s'ha modificat fa més o menys de n minuts*)

-atime [+|-]n (*s'ha accedit fa més o menys de n dies. També està -amin, on n són minuts*)

-ctime [+|-]n (*s'ha creat fa més o menys de n dies. També està -cmin, on n són minuts*)

-newer fitxer (*es vol buscar tot el que tingui una data de modificació més nova que el fitxer especificat*)

-anewer fitxer (*es vol buscar tot el que tingui una data d'accés més nova que el fitxer especificat*)

-size [+|-]n (*"n" és n° de blocs -512 bytes-. Si s'escriu després de "n" la lletra "k", "n" es n° de kbytes*)

- empty (*fitxer buit*)
- inum n^oinode (*troba la ruta de tots els enllaços durs que comparteixin el mateix número d'inode*)
- nouser (*busca tots els fitxers que pertanyin encara a algun usuari que ja no existeixi al sistema*)
- lname (*¿similar a -type l -name?*). També està la opció case-insensitive -ilname

Totes aquestes opcions poden ser negades si s'escriuen després d'un signe ! seguit d'un espai en blanc

On ACCIO pot ser alguna d'aquestes:

- print: retorna la ruta de l'arxiu trobat (acció per defecte)
- delete: esborra l'arxiu trobat. Habitualment s'utilitza juntament amb -depth, per esborrar de “fora a dins”
- exec comanda {} \; : executa la comanda amb cadascun dels arxius trobats o ';'
- ok comanda {} \; : igual que -exec però demanant confirmació a cada troballa
- printf "%TY-%Tm-%Td %TT %p\n" : imprimeix amb format (en este cas: data i hora de modif i ruta)

Exemple: find /usr/share/doc -exec grep "Adler" {} \; -exec ls {} \; 2>/dev/null ... mv {} /ruta/destino

fdupes:

rutaCarpeta1 [rutaCarpeta2] (busca fitxers duplicats en els directoris especificats)

-r : fa una recerca recursiva

-d : esborra tots els duplicats excepte un (pregunta a l'usuari a no ser que es posi el paràmetre -N)

-S : mostra el tamany dels fitxers duplicats

Comandes Linux (IV)

***Treballar amb comptes d'usuari:**

<i>id [nomusuari]</i>	Mostra el UID (identificador numèric d'usuari) i el GID (identificador numèric del grup al que pertany aquest usuari) de l'usuari actual (o bé l'usuari indicat, si s'és root). També mostra el seu nom i el nom del grup al que pertany. L'UID i el GID de l'usuari "root" sempre és 0 (que és el que realment defineix que un usuari sigui administrador o no). A més mostra el conjunt de grups secundaris als que pertany l'usuari indicat
-u	Només mostra el UID. Si s'afegeix el paràmetre -n , només mostra el nom de l'usuari actual
-g	Només mostra el GID. Si s'afegeix el paràmetre -n , només mostra el nom del grup
<i>useradd -m usuari</i>	Crea un nou usuari del sistema. El paràmetre -m crea a més la seva carpeta personal
-u uid	Assigna un UID concret a l'usuari (per defecte serà el sistema qui l'assignarà). Per defecte, els UIDs reservats a "persones" són a partir de 1000.
-g nomgrup gid	Indica a quin grup principal -ha d'existir prèviament- pertanyerà el nou usuari (per defecte es crea un grup nou amb el mateix nom al qual només pertany aquest usuari). Si s'especifica el grup "root", l'usuari tindrà privilegis d'administrador
-G nomgrup,...	Indica a quin/s grup/s secundari/ -ha/n d'existir prèviament- afegirem el nou usuari
-c "comentari"	Afegeix un comentari de l'usuari (gravat a /etc/passwd)
-d /ruta/home	Indica la ruta de la carpeta personal del nou usuari (ha d'acompanyar-se de -m) si no volem que sigui la ruta per defecte
-s /ruta/shell	Indica la shell utilitzada pel nou usuari si no volem l'assignada per defecte
-k /ruta/skel	La llista de shells instal·lades i usables al sistema està a l'arxiu /etc/shells
-e any-mes-dia	Quan es crea un usuari nou, automàticament es copia dins la seva carpeta personal tot el contingut de la carpeta /etc/skel, a mode de plantilla (per tant, si es vol que tot usuari nou tingui una sèrie de configuracions i arxius per defecte, el més fàcil és ubicar-los a /etc/skel). No obstant, si es vol que la plantilla ("l'skeleton") es copïi d'una altra carpeta, es pot indicar amb aquest paràmetre.
	Indica la data d'inhabilitació del compte, després de la qual no es podrà iniciar sessió

NOTA: Els paràmetres per defecte d'*useradd* es poden consultar i configurar a l'arxiu **/etc/default/useradd**. Allà trobem per exemple les línies SHELL, GROUP, HOME, EXPIRE o SKEL. El paràmetre -D de *useradd* mostra els valors actuals que hi ha definits en aquest fitxer.

L'arxiu **/etc/passwd** (propietat de l'usuari root però llegible per tothom) conté la llista d'usuaris del sistema (els quals inclouen no només els usuaris associats a "persones" -usuaris interactius- sinó tots els usuaris que existeixen per donar un context d'execució a determinats dimonis -usuaris "fantasmes"-). Cada línia d'aquest fitxer es correspon amb un usuari diferent, i conté una sèrie de dades separades per ":". Concretament, cada línia té set camps (per més informació, veure *man 5 passwd*): el nom de l'usuari, el seu uid, el seu gid, un comentari sobre ell, la ruta de la seva carpeta personal i el shell que utilitza per defecte:

nomusuari:x:uid:gid:comentari:/ruta/carpeta/home:/ruta/shell/usat/per/lusuari

El segon camp està marcat com "x" perquè és el valor que sol tenir; aquest valor indica que la contrasenya de l'usuari està guardada en un altre fitxer (/etc/shadow). Es pot comprovar, entre altres coses, com l'usuari root té sempre l'uid 0 i el gid 0, i que els usuaris interactius tenen un uid a partir de 1000 (o el valor especificat a /etc/login.defs, com veurem de seguida). Es pot distingir un usuari interactiu d'un usuari "fantasma" observant el valor de l'últim camp: si aquest és /sbin/nologin o /bin/false, és un usuari "fantasma" (perquè no pot obrir cap terminal).

<i>passwd</i> <i>usuari</i>	Li dóna una contrasenya nova a l'usuari especificat. És imprescindible fer aquest pas després d'executar la comanda <i>useradd</i>
-l	Bloqueja l'usuari especificat per tal de què no pugui iniciar sessió
-u	Desbloqueja l'usuari
-d	Esborra la contrasenya (la posa en blanc)
-e	Expira immediatament la contrasenya; al pròxim login es demanarà renovar-la
-n n°	Indica el n° de dies mínim que ha de passar per poder canviar la contrasenya (0=a tota hora)
-x n°	Indica el n° de dies màxim en què expirarà la contrasenya des de l'últim canvi
-w n°	Indica el n° de dies abans de la data d'expiració de la contrasenya que s'informarà a l'usuari
-i n°	Indica el n° de dies després de la data d'expiració de la contrasenya fins inhabilitar compte. Similar al paràmetre -f de <i>useradd</i> . Un valor de -1 indica que el compte no s'inhabilitarà mai
-S	Mostra el nom de l'usuari, si està bloquejat, sense contrasenya o amb (L,NP,P respectivament) la data del últim canvi de contrasenya, el n° min, el n° max, n° warning i el n° fins inhabilitar. Afegint el paràmetre -a no cal indicar cap usuari perquè es veu aquest informació de tots ells.
--stdin	Permet llegir la contrasenya d'un pipe (p.ex: echo "1234" <i>passwd</i> --stdin <i>usuari</i>). Només funciona a sistemes RedHat, però

Com que l'arxiu */etc/passwd* té permisos 644 degut a què diferents programes necessiten accedir-hi per llegir la informació que hi ha allà guardada, les contrasenyes s'han de guardar en un altre arxiu també propietat de root però protegit amb permisos 600, l'arxiu */etc/shadow*. Cada línia d'aquest fitxer es correspon amb una contrasenya diferent del sistema, i conté una sèrie de dades separades per ":". Concretament, cada línia té nou camps (per més informació, veure *man shadow*): el nom de l'usuari, la seva contrasenya encriptada, la data de l'últim canvi de contrasenya -expressada en forma de número de dies des de 1-1-1970 fins aquesta data-, el n° de dies mínim que ha de passar per poder canviar la contrasenya (paràmetre -n de *passwd*), el n° de dies màxim en què expirarà la contrasenya des de l'últim canvi (paràmetre -x de *passwd*), el n° de dies abans de la data d'expiració de la contrasenya que s'informarà a l'usuari (paràmetre -w de *passwd*), el n° de dies després de la data d'expiració de la contrasenya fins inhabilitar compte (paràmetre -i de *passwd*), la data d'inhabilitació del compte -expressada en forma de número de dies des de 1-1-1970 fins aquesta data- (paràmetre -e de *useradd*) i un darrer camp que no té significat.

nomusuari:contrasenyaencript:dataultimcanvi:n:x:w:i:datainhabilitacio::

L'algoritme d'encriptació utilitzat es pot conèixer mirant els primera caracters del segon camp de */etc/shadow*: si comença per \$1\$ l'algoritme és MD5, si comença per \$2\$ és Blowfish, si comença per \$5\$ és SHA256 i si és \$6\$ és SHA512. Una mateixa contrasenya no crea el mateix hash perquè cada cop s'utilitza una llavor diferent: aquesta llavor (que es genera aleatòriament) és el que es veu entre els caracters anterior i el següent signe \$; la contrasenya pròpiament dita es el que va després. L'algoritme escollit per encriptar les contrasenyes es defineix a */etc/pam.d/common-password* (o */etc/pam.d/system-auth* a Fedora) a la línia on s'indica el mòdul pam_unix.so. Allà també es pot especificar entre altres coses la longitud mínim de les contrasenyes, amb el paràmetre "minlen".

Podem fer que la contrasenya estigui en blanc si esborrem el segon camp de */etc/shadow* (truc útil si l'hem oblidada). També podem desactivar un compte d'usuari si afegim al davant del hash de la contrasenya el símbol "!" (això és el que fan, de fet les comandes *passwd -l* i *usermod -L*) i tornar-lo a activar treient-lo. El símbol "*" en comptes del hash de la contrasenya també desactiva el compte d'usuari però llavors caldrà tornar a donar una nova contrasenya amb la comanda *passwd* per tornar a habilitar l'usuari associat.

Els valors que s'usen per defecte a l'hora de crear un usuari relacionats amb la duració de les contrasenyes es troben a l'arxiu */etc/login.defs*. Allà hi ha per exemple les línies:

PASS_MIN_DAYS (valor per defecte equivalent al paràmetre -n)

PASS_MAX_DAYS (que estableix el valor per defecte equivalent al paràmetre -x)

PASS_WARN_AGE (valor per defecte equivalent al paràmetre -w)

També hi ha altres aspectes relacionats amb la creació d'usuaris, com ara:

UID_MIN i UID_MAX (uid mínim i màxim possibles per usuaris interactius)

LOGIN_TIMEOUT (número màxim de segons que *login* deixa introduir una contrasenya)

CREATE_HOME (si val "yes" permet no haver d'escriure el paràmetre -m d'*useradd* per crear la carpeta personal)

USERGROUP_ENAB (si val "yes" -per defecte-, en crear un usuari es crearà un grup homònim que el conté i si val "no", s'afegirà al grup indicat a l'opció GROUP de */etc/default/useradd*)

userdel -r usuari Elimina un usuari del sistema. El paràmetre *-r* esborra a més la seva carpeta personal. En general, és recomanable, abans d'eliminar un usuari, esborrar la seva bústia del sistema (*/var/spool/mail/usuari*), els seus treballs pendents de cron, els seus processos, els seus treballs d'impressió i treure'l dels diferents grups

La comanda *usermod* admet els mateixos paràmetres vistos que *useradd* (excepte *-k*) per tal de modificar algun aspecte de l'usuari indicat. Per tant, si fem per exemple *usermod -u uid usuari* canviarem el seu UID, si fem *usermod -g nougrup usuari* canviarem el seu grup primari, si fem *usermod -s /bin/fish usuari* canviarem el seu shell per defecte, si fem *usermod -m -d /ruta/nova usuari* mourem el contingut de la seva carpeta personal antiga a la nova ruta, si fem *usermod -e yyyy-mm-dd usuari* canviarem la seva data d'inhabilitació, etc. Aplicacions concretes no vistes fins ara:

<i>usermod -l nounom usuari</i>	Canvia el nom a un usuari ja existent
<i>usermod -a -G nomgrup usuari</i>	Afegeix l'usuari al grup secundari. El mateix fa <i>gpasswd -a usuari grup</i>
<i>usermod -G "" usuari</i>	Treu l'usuari de tots els grups secundaris. Si es vol treure només d'un grup secundari específic, caldrà fer <i>gpasswd -d usuari grup</i>
<i>usermod -L usuari</i>	Bloqueja el compte (similar a <i>passwd -l usuari</i>)
<i>usermod -U usuari</i>	Desbloqueja el compte (similar a <i>passwd -u usuari</i>)

groupadd nomgrup Crea un nou grup -buit- al sistema. En realitat, l'afegeix a la llista-arxiu */etc/group*
-g gid Especifica un GID concret en comptes de deixar que el sistema li assigni un automàtic

groupdel nomgrup Elimina el grup indicat (sempre i quan no hi hagi cap usuari que en pertanyi de forma primària)

groupmod -g gid nomgrup Modifica el GID d'un grup existent
groupmod -n nounom nomgrup Modifica el nom d'un grup existent

A l'arxiu */etc/group* s'estableix la pertinença dels usuaris a més grups apart del seu grup principal. El seu format és:

nomgrup:x:gid:usuariquehipertany,unaltreusuariquehipertany,...

El segon camp sempre és "x" (no s'usa actualment), el tercer correspon al GID del grup en qüestió i el quart és la llista d'usuaris (separats per comes) que (a més de pertànyer al seu grup principal corresponent) pertanyen també a aquest grup.

NOTA: Recordeu que amb la comanda *id* es pot saber l'UID i el GID del grup principal de l'usuari actual, a més dels GIDs dels altres grups als que pertany. La comanda *groups* mostra, en canvi només la llista dels noms de grups de l'usuari actual (o si s'és root, de l'usuari indicat)

NOTA: A <https://wiki.debian.org/SystemGroups> es llisten els grups que solen venir per defecte en un sistema Debian i quina és la seva funció.

last Mostra quins usuaris han iniciat sessió a la màquina, quan, quant de temps i des d'on. Per obtenir aquesta informació llegeix el fitxer */var/run/wtmp* (pels accessos anteriors) i */var/run/utmp* (pels accessos actualment actius). Els següents paràmetres serveixen per filtrar la informació mostrada (es poden combinar)

-n n° Mostra només les darreres *n* entrades
usuari Mostra només les entrades associades a l'usuari indicat
ttyX Mostra només les entrades associades al terminal indicat
-s yyyyymmddhhmmss Mostra només les entrades a partir de la data indicada
-t yyyyymmddhhmmss Mostra només les entrades fins la data indicada
-da Mostra una columna més que indica la IP de l'ordinador d'origen (si l'accés és remot)

lastb Mostra els darrers intents fallits de connexió. La sortida i els paràmetres que admet són similars als de la comanda *last*. En aquest cas, llegeix l'arxiu */var/run/btmp*

<i>lastlog</i>	Llegeix <code>/var/log/lastlog</code> , que conté tots els usuaris del sistema amb informació de la darrera connexió
<i>who</i>	Mostra quins usuaris tenen obertes sessions al sistema, a quins terminals, des d'on i des de quan
-q	Conta el número d'usuaris connectats en aquest moment.
-H	Imprimeix les capçaleres
-u	Afegeix dues columnes més: temps d'inactivitat i PID executat per l'usuari
<i>whoami</i>	Diu quin usuari sóc jo
<i>w</i>	Ja vista als apunts de processos
<i>users</i>	Mostra els usuaris que estan loguejats al sistema (com <i>who</i> , però mostrant molta menys informació). En realitat busca els usuaris a <code>/var/run/utmp</code>

Hi ha diferents arxius relacionats amb el comportament general dels usuaris del sistema que convé conèixer. P.ex:

**/etc/issue* : Missatge que es mostra als terminals virtuals abans d'iniciar sessió. Dins el seu contingut s'admeten diferents valors especials com `\d` (data actual) , `\t` (hora actual), `\s` (nom del sistema operatiu), `\r` (versió del sistema operatiu), `\l` (nom del terminal), `\m` (arquitectura de la màquina), `\n` (nom de la màquina), `\o` (domini de la màquina), `\u` (número d'usuaris actualment connectats a la màquina), etc

**/etc/motd* : Missatge que es mostra just després d'haver iniciat sessió a un terminal virtual. Si no es desitja mostrar cap missatge per un usuari en concret, es pot crear el fitxer (buit) `~/.hushlogin` (veure `man login.defs`).

**/etc/profile* (i `~/.profile`) : Script que s'executa cada cop que s'inicia sessió en un terminal virtual o bé mitjançant *su*. Està l'script general per tots els usuaris i l'script particular per cada usuari, el qual pot sobrescriure el genèric. En general, aquests scripts serveixen per definir variables d'entorn i definir àlies.

**/etc/bash.bashrc* (i `~/.bashrc`) : Script que s'executa cada cop que s'obre un terminal Bash (en un terminal virtual o bé en un entorn gràfic mitjançant `gnome-terminal`, `konsole`, `xterm` o similars). Està l'script general per tots els usuaris i l'script particular per cada usuari, el qual pot sobrescriure el genèric. En sortir del terminal Bash es llegeix l'arxiu `~/.bash_logout`

**/etc/securetty* : Conté la llista de terminals virtuals des d'on es pot accedir com a root

**/etc/nologin* : Si existeix, deshabilita l'accés al sistema de tots els usuaris excepte el root. El seu contingut és el missatge que veuran els usuaris

Su i sudo:

<i>su -l [usuari]</i>	Canvia l'usuari de treball per l'usuari indicat (si no s'indica, s'entén que és el "root"). El paràmetre -l (o bé, que seria el mateix, un guió) és necessari per renovar, un cop s'és el nou usuari, el valor de les variables d'entorn i el directori personal (obrint-se a més la shell que sigui la definida al seu arxiu <code>/etc/passwd</code>); si no es posa es mantindrien els valors de l'usuari antic. Demana la contrasenya de l'usuari al que es vol convertir.). Per tornar a l'usuari anterior cal escriure <i>exit</i>
-c comanda	Executa la comanda indicada com l'usuari indicat (fa el mateix que <i>sudo</i>)
<i>sudo comanda</i>	Executa una comanda com si l'hagués executat el "root" (o com si l'hagués executat l'usuari indicat amb el paràmetre -u <i>nomusuari</i> uid Demana la contrasenya de l'usuari <u>actual</u> (i, per tant, a diferència de amb la comanda <i>su</i> , aquí no cal saber la contrasenya de "root" -ni tan sols cal que existeixi-). Aquesta comanda només la poden fer servir els usuaris que estan especificats dins de l'arxiu <code>/etc/sudoers</code> (que estudiarem tot seguit). Per defecte, l'usuari que es crea en la instal·lació d'un sistema Ubuntu hi és present, però els altres que es puguin crear posteriorment no. A la instal·lació d'un sistema Fedora es pot escollir fer servir la mateixa configuració (usuari amb sudo i usuari root deshabilitat) o bé utilitzar la configuració "clàssica" (usuari sense sudo i usuari root habilitat).
<i>variable=valor</i>	Executa la comanda amb el valor indicat de la variable d'entorn indicada (normalment les variables d'entorn es resetegen, així que d'aquesta manera s'assegura que la indicada té el valor que es vol). Existeix el paràmetre -E per fer que es mantingui el valors de totes les variables d'entorn antigues però no funciona si al fitxer <code>/etc/sudoers</code> està la paraula <code>NOSETENV</code> : (allà on es col·loca <code>NOPASSWD</code> ;, tal com veurem de seguida)
-H	Estableix el valor de \$HOME a la de l'usuari destí

Si es vol activar el compte de root en Ubuntu o similars, que el tenen desactivats, amb l'usuari que es va crear a l'instal·lació, s'ha de fer: *sudo passwd root* per donar una contrasenya al root. I ja està

<i>sudo</i>	
-l	Mostra quines accions estan permeses per l'usuari actual en executar el sudo
-v	Allarga 5 minuts més el temps sense preguntar contrasenya
-k	Oblida la contrasenya (per a què s'hagi de tornar a introduir)
-L	Llista les diferents possibilitats de la línia Defaults de l'arxiu <code>/etc/sudoers</code>
-i	Similar a <i>su -l root</i> . També es podria fer <i>sudo su -l root</i>
-s	Similar a <i>su root</i>

L'arxiu `/etc/sudoers` indica quin usuari pot executar sudo i quin no. La majoria de les seves línies tenen aquest format:

user_alias host_alias = (runas_alias) tag command_alias

on:

**user_alias* pot ser el nom d'un usuari, o bé el seu UID (si es posa primer un # davant del número) o bé un grup d'usuaris (si es posa primer un % davant del nom) o bé un conjunt de tot plegat (definit prèviament a la línia *User_Alias*, així: *User_Alias GENT = pepe,manolo,%grup*)

**host_alias* és el conjunt de computadors (generalment definit prèviament a la línia *Host_Alias* així: *Host_Alias MIPC=127.0.0.1, 192.168.1.3*) des d'on l'usuari podrà executar el *sudo*. Aquest valor és útil si l'arxiu `/etc/sudoers` està compartit per diverses màquines. La majoria de vegades, però, aquí veurem el valor especial *ALL*

**runas_alias* és el conjunt d'usuaris (directament separats per comes o bé definit prèviament a la línia *Runas_alias* així: *Runas_Alias PERSONES = mama,papa*) en què l'usuari que executi el *sudo* es podrà convertir. La majoria de vegades aquí veurem el valor especial *ALL*

**tag* representa una sèrie de valors opcionals que modifiquen el comportament de la línia. Per exemple, aquí pot haver el valor NOPASSWD: , la qual indica que no es demanarà mai contrasenya quan s'executi sudo

**command_alias* és el conjunt de comandes (directament separades per comes o bé definit prèviament a la línia Cmnd_Alias així: *Cmnd_Alias COMANDES = /bin/chmod, /bin/chown*) que es podran executar amb sudo. Si es vol que siguin qualsevol, aquí veurem el valor especial ALL. També es poden fer servir comodins

NOTA: No cal especificar programes concrets: es pot indicar un directori, i s'entendrà que són tots els programes d'allà dins. Per escriure un directori, s'ha d'acabar la ruta amb un / final

Així doncs, per fer que un usuari pugui executar sudo, l'únic que caldria fer és afegir una línia com la següent a l'arxiu /etc/sudoers (sota la línia ja existent *root ALL=(ALL) ALL*, per exemple): *nomusuari ALL=(ALL) ALL* Una alternativa també podria ser simplement afegir aquest usuari a un grup d'usuari que ja pugui executar sudo perquè ja estigui present a l'arxiu /etc/sudoers (exemples són el grup "admin" i el grup "sudo"). Així, fent *usermod -a -G sudo nomusuari* obtindríem el mateix resultat.

En qualsevol cas, posar un ! davant d'un valor és equivalent a dir "excepte"

Cal tenir en compte que les regles es llegeixen en ordre de dalt a baix fins el final...vigilar per tant amb les contradiccions.

A més de les línies anteriors, a l'arxiu /etc/sudoers també podem trobar altres línies més específiques, com ara:

La línia "Defaults timestamp_timeout" defineix el número de minuts sense demanar contrasenya

La línia "Defaults env_reset" reseteja les variables d'entorn a les de l'usuari destí

La línia "Defaults env_keep='VAR1 VAR2 VAR3'" manté les variables d'entorn antigues especificades

La línia "Defaults passprompt="frase_q_es_mostra_quan_se_solicita_la_contrasenya"

La línia "Defaults badpass_message="frase_q_es_mostra_quan_sha_ficat_la_contrasenya_malament"

logout, exit

Desconnecten i surten del consola, però no tanquen sessió gràfica.

shutdown now

Apaga la màquina. En comptes de "now" es pot especificar una hora del tipus hh:mm per apagar la màquina a aquella hora, o bé "+n", on n és el número de minuts d'espera abans de l'aturada.

"Wrappers" d'aquesta comanda són: reboot, halt i poweroff

-r Reinicia la màquina en comptes d'aturar-la

-c Cancela l'aturada/reinici

Comandes Linux (VI)

***Gestió de paquets:**

A sistemes Debian (Ubuntu és un d'ells), els paquets d'instal·lació tenen el format intern “deb”. En canvi, a sistemes RedHat (Fedora, OpenSuse, etc) els paquets d'instal·lació tenen el format intern “rpm”. Això implica que per gestionar els paquets s'utilitzen una sèrie de comandes que són diferents entre els dos sistemes. Concretament:

Sistemes Debian

La comanda aptitude és una de les que tenim (una altra famosa és apt-get) per connectar als repositoris, obtenir d'allà la informació necessària, descarregar els paquets i instal·lar-los (via dpkg), entre moltes altres coses:

aptitude install programa :Descarrega i instal·la un programa
aptitude remove programa: Esborra un programa, deixant els fitxers de configuració per un possible futur
aptitude purge programa: Esborra un programa completament
aptitude search programa: Busca un programa als repositoris, i informa de si està ja instal·lat o no
aptitude show programa: Mostra informació del programa (versió, per a què serveix, pàgina web...)
aptitude safe-upgrade: Actualitza tots els paquets que ho necessitin a la versió més moderna dels repositoris

La llista de repositoris es troba en un fitxer de text anomenat /etc/apt/sources.list. Si no tenim Internet, però tenim els paquets ja descarregats en un ordinador, podem fer servir la comanda dpkg:

dpkg -i paquet : Instal·la un paquet
dpkg -r paquet: Esborra un paquet, deixant els fitxers de configuració per un possible futur
dpkg -P paquet:Esborra un paquet completament
dpkg -I paquet: Mostra informació del paquet
dpkg -L paquet: Mostra la llista de fitxers inclosos dins del paquet
dpkg -S /ruta/fitxer : Diu a quin paquet pertany el fitxer especificat

Sistemes RedHat

La comanda yum és una de les que tenim (una altra famosa és zypper) per connectar als repositoris, obtenir d'allà la informació necessària, descarregar els paquets i instal·lar-los (via dpkg), entre moltes altres coses:

yum install programa : Descarrega i instal·la un programa
yum remove programa:Esborra un programa completament
yum search programa :Busca un programa als repositoris, i informa de si està ja instal·lat o no
yum info programa :Mostra informació del programa (versió, per a què serveix, pàgina web...)
yum update : : Actualitza tots els paquets que ho necessitin a la versió més moderna dels repositoris

La llista de repositoris es troba en diferents fitxers de text (un per cada repositori) amb extensió “repo” dins la carpeta /etc/yum.d. Si no tenim Internet, però tenim els paquets ja descarregats en un ordinador, podem fer servir la comanda rpm:

rpm -U paquet : Instal·la un paquet
rpm -e paquet: Esborra un paquet completament
rpm -qi paquet: Mostra informació del paquet
rpm -ql paquet: Mostra la llista de fitxers inclosos dins del paquet
rpm -qf /ruta/fitxer : Diu a quin paquet pertany el fitxer especificat

A vegades ens podem trobar amb que un programa no està ni en format “deb” ni en “rpm”, sinó que està directament en codi font. En aquest cas, el que s'ha de fer és llegir les instruccions d'instal·lació que venen dins l'arxiu README, ja que en aquests casos cada programa és un món.