

Curs de GNU/Linux

1. Introducció

Una mica d'història

En els inicis de la computació moderna, el programari havia d'acompanyar a l'ordinador en el moment de la seva adquisició i no era habitual que els programes fossin una mercaderia al marge del maquinari (equip). Clar, estem situats en l'època dels anys seixanta quan encara quedava molt lluny el concepte d'ordinador *personal*.

La companyia IBM era capdavantera en el mercat i subministrava equips als centres de càlcul de les grans empreses, universitats i altres organismes públics.

No existia, per tant, la noció de programari lliure tal com es coneix a partir dels anys vuitanta. No era necessari aquest concepte, doncs el programari era amb tota naturalitat, lliure. Es distribuïa habitualment al costat del seu codi font i els programadors compartien el seu codi en trobades, congressos i revistes especialitzades, com succeeix normalment amb el coneixement acadèmic i científic.

En aquest ambient, el ritme de desenvolupament de les aplicacions era molt elevat. Els *hackers* o programadors desenvolupaven la seva tasca en els departaments d'investigació de les grans empreses, universitats i altres organismes com l'Institut Tecnològic de Massachusetts, conegut també com MIT, que liderava en aquella època la implantació i el desenvolupament de solucions informàtiques.

Unix va ser un dels primers sistemes operatius evolucionats, sobretot per la seva capacitat de ser instal·lat en diferents tipus d'ordinadors. Va aparèixer a principis dels anys setanta en els Laboratoris Bell de la companyia AT&T, fruit del treball de Thompson i Ritchie, entre altres desenvolupadors. Es va estendre ràpidament per les universitats gràcies a una llicència que permetia el seu ús per a entitats educatives i, d'aquesta forma, gràcies a la disponibilitat del codi font, fou estudiat, corregit i millorat. La Universitat de Califòrnia, en Berkeley va polaritzar aquest desenvolupament creant i distribuint una variant lliure, Free BSD, que encara avui dia gaudeix de plena vigència i salut.

El túnel fosc

És evident que la situació va canviar aviat. La companyia IBM va anunciar que vendria part dels seus programes independentment dels equips, per tant, el programari va començar a prendre valor comercial per si mateix, al marge de l'ordinador on s'instal·lés. Com a conseqüència van començar les restriccions d'accés al codi font i les aplicacions es van tancar, impedit als usuaris l'estudi i la redistribució.

Això succeïa a principis de l'any 1970. Tota una indústria naixia al reclam de la nova forma de negoci i molt aviat es va convertir en absolutament normal la distribució comercial de qualsevol tipus de programari, sense accés al codi font i protegit pel marc legal establert.

En aquells moments, la comunitat de programadors assistia atònita al fenomen que s'estava produint al seu voltant, la majoria va ser "devorada" per la maquinària industrial i els més partidaris de la llibertat en el programari es van refugiar en la variant BSD d'Unix nascuda en Berkeley. No havia, per tant, un posicionament organitzat que plantés cara a aquesta situació i no apareixeria formalment fins a intervinguda la dècada dels vuitanta quan Richard Stallman lidera la formació del projecte GNU.

El moviment GNU

Richard Stallman treballava en el Laboratori d'Intel·ligència Artificial del MIT i era un dels hackers més prestigiosos i amb un futur més prometedori dintre de l'elit dels desenvolupadors. No obstant això, no estava satisfet amb l'ambient de treball que envoltava al creador de programari. Donat el caràcter comercial de les aplicacions, les empreses tractaven per tots els mitjans d'evitar que el codi estigués disponible i, per a això, feien signar als programadors uns acords de "no revelació" similars als utilitzats en la indústria per preservar les patents. Stallman no es trobava còmode en aquest

ambient. Havia de compartir el seu codi i creia en la naturalesa científica d'aquest i, per tant, en la seva disponibilitat i lliure circulació.

Aquest extracte del projecte GNU redactat per Stallman reflecteix perfectament aquesta circumstància (<http://www.gnu.org/gnu/thegnuproject.ca.html>):

"Al desaparèixer la meua comunitat, es va fer impossible continuar com abans. En lloc d'això, em vaig enfrontar a una elecció moral severa. L'elecció fàcil era unir-me al món del programari propietari, signar els acords de no revelar, i prometre que no aniria en ajuda del meu amic hacker. És molt probable que desenvolupés programari que es lliuraria sota acords de no revelar i d'aquesta manera incrementés també les pressions sobre altra gent perquè traixin als seus companys. Hauria fet diners d'aquesta manera, i tal vegada m'hagués divertit escrivint codi. Però sabia que al final de la meua carrera, al mirar enrere als anys construint parets per a dividir a la gent, sentiria que vaig usar la meua vida per a empitjorar el món."

Finalment va decidir abandonar el seu treball en el MIT per a no sentir-se obligat per cap acord de "no revelació" i dedicar-se a escriure un sistema lliure complet. Va néixer el projecte [GNU](#).

La principal preocupació de Stallman fou la llibertat del programari que produiria el projecte. No era suficient amb que els destinataris rebessin i compartissin els programes, sinó que la seva pretensió anava més enllà, volia assegurar-se que les aplicacions romandrien sempre amb el mateix caràcter, és a dir, havien de continuar sent redistribuïdes lliurement després de les millores que poguessin experimentar.

Per a això va idear un tipus especial de llicència, anomenada GPL (*Llicència Pública General*), que impedia de tancar el codi i patentar el programari emparat per ella i les seves posteriors evolucions. Aquesta llicència de caràcter "enganxós" fou denominada *copyleft*, en contraposició a les restrictives condicions de còpia habituals.

Si desitges més informació sobre aquest tema, has de consultar els diversos tipus de llicència GNU en l'adreça: <http://www.gnu.org/licenses/licenses.ca.html>

També es va crear la Fundació per al Programari Lliure ([FSF](#)), organització sense ànim de lucre amb l'objectiu de recaptar fons que financessin la producció d'aquestes aplicacions lliures. Aquesta fundació va subsistir amb les aportacions de particulars, empreses i institucions, al costat dels beneficis de la venda de cintes i manuals dels programes que produïen.

Tot aquest moviment va començar a produir programari lliure d'una gran qualitat i a un ritme vertiginós fins al punt que, des dels seus inicis, allà per l'any 1984, fins a finals de 1990, es trobaven molt prop d'aconseguir el seu objectiu. Tan sols faltava el nucli del sistema operatiu per disposar d'una implementació lliure i completa que fos capaç de fer funcionar i produir a un ordinador.

A la recerca d'un kernel

Mentrestant, el grup de hackers acollits en la llibertat de la variant Unix de la Universitat de Califòrnia (BSD) es trobaven també a un pas d'obtenir un kernel lliure, de fet, els faltaven només sis fitxers per culminar aquest assoliment.

Finalment, Bill Jolitz acaba d'escriure'ls i allibera 386BSD entorn de l'any 1992, que més tard acabaria en l'actual OpenBSD; no obstant això, la Universitat fou demandada i molts usuaris van abandonar el projecte per temor a una eventual derrota judicial.

En l'altra part del món, Linus Torvalds, un finlandès estudiant d'informàtica, llança un missatge a la xarxa Internet anunciant que estava treballant en un nucli construït a partir d'un rudimentari sistema Minix i havia aconseguit que funcionés el compilador gcc (creat per la FSF). Aquest anunci i les facilitats de comunicació que permetia la incipient Internet, va fer que nombrosos desenvolupadors es bolquessin amb el projecte i va començar una tasca frenètica on les versions noves s'alliberaven i es publicaven cada setmana pràcticament, fins a l'extrem que al març de 1994 va aparèixer la versió 1.0, la primera estable i totalment operativa. Aquest kernel es va llicenciar sota la GPL i va començar a integrar-se sobre ell tot el programari lliure existent, apareixent així el

concepte de *distribució*, l'objectiu de la qual és oferir a l'usuari un sistema i un conjunt d'aplicacions finals llistes per ser instal·lades. Aquestes circumstàncies expliquen que sigui més correcte i més just denominar a aquest sistema com GNU/Linux.

L'expansió de GNU/Linux

Encara estem en una època on la informàtica no ha arribat massivament a les llars i el programari lliure està reservat a programadors i usuaris amb certs coneixements del codi. No obstant això, el desenvolupament d'Internet i, sobretot, l'aparició dels moderns entorns d'escriptori, fonamentalment KDE i GNOME, a mitjans dels anys noranta, permeten al programari lliure estendre's al territori de les petites instal·lacions i a l'usuari domèstic, deixant així de ser patrimoni exclusiu de les grans xarxes corporatives, servidors d'Internet i altres funcions de gamma alta.

Altre factor a tenir molt en compte en el desenvolupament de GNU/Linux és l'aparició de les *distribucions*, un sistema operatiu i una col·lecció d'aplicacions llistes per usar, al costat d'un grup d'eines de configuració, instal·lació, gestió i administració.

Tot això, orientat a usuaris sense coneixements informàtics que han de disposar d'un entorn de producció fàcil d'instal·lar i mantenir.

La llavor de la llibertat en el programari va arrelant cada vegada més en la societat fins al punt que, en l'actualitat, no pocs observadors es pregunten si ens trobem davant un nou model de negoci que canviarà les estructures d'una indústria molt jove, però molt potent. De fet, la majoria de les multinacionals ja s'han posicionat i aproximat a aquest moviment, els fabricants de maquinari comencen a subministrar controladors que permetin usar els seus productes sota GNU/Linux. Els projectes de nous desenvolupaments lliures se succeeixen i es multipliquen aconseguint uns nivells de qualitat i uns ritmes de desenvolupament molt elevats.

Característiques del Programari Lliure

Tal vegada una de les formes més còmodes de caracteritzar al programari lliure sigui contraposar-lo a les condicions habituals de les llicències dels programes propietaris, analitzar cadascuna de les restriccions que solen imposar aquestes llicències i concloure que el programari es considera lliure quan està permès realitzar tot el prohibit en el propietari. Vegem amb una mica més de detall els requisits perquè una aplicació sigui qualificada de programari lliure.

La llibertat, en aquest context, es refereix que els usuaris poden executar, copiar, distribuir, canviar i millorar el programa. Dita així, pot fer l'efecte que tot, absolutament tot, està permès, no obstant això, no és cert.

Les llicències lliures tracten de protegir aquesta llibertat, però també solen imposar certes restriccions per garantir-la, per exemple, poden obligar-nos a mantenir els crèdits originals del programa quan realitzem una redistribució o alliberar aplicacions que estiguin basades en programari lliure millorat. De manera que la denominació programari lliure, tal com fou concebuda pel moviment GNU, ha de preservar quatre llibertats fonamentals per als usuaris:

1. Llibertat per a executar el programa amb qualsevol propòsit, en qualsevol lloc i per a sempre.
2. Llibertat per a estudiar el seu funcionament i adaptar-lo a les nostres necessitats.
3. Llibertat per a distribuir còpies.
4. Llibertat per a millorar el programa i publicar aquestes millores perquè la resta de la comunitat es beneficiï d'elles.

Aquestes quatre llibertats són les que protegeix la llicència que acompanya als programes distribuïts sota la denominació de programari lliure. Lògicament, la garantia d'aquestes llibertats exigeix l'accés al codi font de les aplicacions, en cas contrari, serien impossibles de ser dutes a la pràctica.

Existeixen diversos tipus de llicència que emparen productes considerats programari lliure, no obstant això, altres llicències poden resultar confuses i estar redactades en

termes interessadament ambigus per aprofitar la creixent popularitat i acceptació que està prenent el programari lliure. La *Fundació per al Programari Lliure* proposa un senzill test per comprovar si una llicència o un programa compleix les condicions necessàries per ser considerat lliure. Consisteix en plantejar-se les següents qüestions:

1. Disposes de les fonts del programa?
2. Pots modificar aquestes fonts?
3. Està permès distribuir el que modifiques?
4. Pots vendre les modificacions al preu que vulguis?
5. Has d'incloure les fonts obligatòriament al distribuir?

Si la resposta és afirmativa a les quatre primeres preguntes, el programa s'ha de considerar programari lliure, si a més la cinquena és també afirmativa, llavors estarà protegit amb *copyleft*, que impedirà als distribuïdors incloure restriccions addicionals que puguin convertir en propietari aquest programari.

També és important assenyalar que programari lliure no significa "no comercial", de fet, el programari comercial lliure té cada vegada més presència en els mercats, malgrat que qualsevol pot redistribuir-lo sense cost algun.

Diversos "cognoms" del programari

Normalment trobarem diferents categories d'aplicacions per a l'ordinador, si ens atenim a les seves condicions d'ús, modificació, còpia i distribució expressades en la seva llicència. Moltes d'aquestes denominacions poden prestar-se a confusió, per això és molt convenient aclarir el que signifiquen els diferents qualificatius aplicats al programari.

Programari comercial és el desenvolupat per una entitat amb intenció de vendre'l i obtenir uns beneficis. Ja hem advertit que no ha de confondre's amb "propietari", ja que hi ha programari lliure que és comercial, encara que és cert que la majoria del programari comercial és propietari.

Programari semillliure és aquell que manté les mateixes característiques que el programari lliure per als usuaris individuals, entitats educatives o sense ànim de lucre, no obstant això, prohibeix aquestes llibertats per al seu ús comercial o empresarial. Algú pot preguntar-se si una aplicació semillliure hauria de ser inclosa en les distribucions habituals de programari lliure, ja que el seu comportament és idèntic quan no mitja ànim de lucre. Per descomptat, això no és possible, almenys en les distribucions tipus GNU, per a això, s'addueixen dues raons fonamentals que impedeixen la redistribució d'aplicacions semillliures. En primer lloc, per una qüestió de principis ètics, ja que la llibertat en el programari, segons aquesta postura, ha de ser absoluta, incloses les empreses; més encara, quan les restriccions dels programes semillliures han d'obeir a objectius merament egoistes. En segon lloc, perquè la vocació de les distribucions de programari GNU abasta també entre els seus usuaris potencials a les empreses, incloent, per què no, un caràcter comercial en la relació.

Programari propietari és aquell que no és lliure ni semillliure, per tant, la seva redistribució, modificació i còpia estan prohibides o, almenys, tan restringides que és impossible fer-les efectives. Per descomptat, no està disponible el seu codi font. És el cas més comú del programari comercial que es distribueix actualment.

Freeware no té una definició clara i precisa, no obstant això, sol usar-se

per classificar al programari gratuït que pot redistribuir-se lliurement encara que no pugui modificar-se, entre altres coses, perquè no està disponible el seu codi font. Han de ser programes que s'usen per promocionar una versió comercial del mateix o altres serveis oferts pel desenvolupador. El *freeware* no és programari lliure.

Shareware és un programari que permet la seva redistribució, no obstant això, no va acompanyat del seu codi font i, per tant, no pot ser modificat. A més, passat un període de temps, normalment és necessari pagar una llicència per continuar usant-lo, ja que el programa deixa de funcionar, perd alguna funcionalitat o et recorda de forma insistent que és una versió sense registrar. Tampoc és programari lliure.

Open source (o programari de font oberta) és una denominació creada per la *Open Source Initiative* ([OSI](#)) en 1998 amb la intenció d'evitar l'ambigüitat del vocable *free* (lliure/gratis) en llengua anglesa, i aconseguir la introducció del programari lliure en l'àmbit comercial amb aquest terme considerat més "políticament correcte". Pràcticament, les característiques del *open source* i del programari lliure són equivalents, de fet, la OSI va utilitzar una versió de la usada pel projecte Debian –un dels més fidels a l'esperit GNU- per a definir els requisits de les aplicacions de font oberta, no obstant això, els partidaris d'una o altra denominació protagonitzen un dels debats més polèmics dintre del moviment pel programari lliure.

Els avantatges d'usar programari lliure

Arribats a aquest punt, la majoria dels avantatges de l'ús de programari lliure estan ressenyades en algun dels apartats anteriors, bé directament o bé deduïbles de les característiques i naturalesa d'aquest tipus de programari. No es tracta tampoc de mostrar una exhaustiva llista de beneficis socials i individuals derivats de l'adopció de solucions lliures per als sistemes informàtics. Sí tractaré de ressenyar algunes de les conseqüències més reveladores en aquest sentit.

La implantació i ús de programari lliure amb sistemes del tipus GNU/Linux, no ha de ser contemplada al marge de la seva filosofia sobre la llibertat. Sense entrar en consideracions d'altre tipus, aquesta llibertat porta amb si una evident millora de la qualitat com a conseqüència del model generatiu d'aplicacions. Els projectes estan contínuament auditats per desenvolupadors, verificadors i usuaris en general, constituint un potent estímul per a l'optimització del codi, combinant-se la col·laboració amb la competència, a la recerca de la millor solució final.

En un aspecte, potser més social, els avantatges vénen des del costat cooperatiu, pel que suposa un sistema que fomenta la lliure circulació d'informació i coneixement, veritables motors del progrés científic i humà. També el seu caràcter gratuït permet el lliure accés a aquestes tecnologies per part de sectors més desfavorits socialment. Per descomptat, en els entorns acadèmics aquesta elecció sembla fora de tot dubte.

La pròpia disponibilitat del codi font propícia característiques com l'examen per prevenir forats de seguretat o funcionalitats no documentades, ja ressenyada en l'apartat anterior. A més, li permet comptar amb la xarxa de suport més àmplia que cap sistema comercial pot oferir. Multitud de grups en tot el món, organitzats i molt actius, ofereixen uns canals de participació i fòrums d'ajuda que ens resoldran qualsevol dificultat que pogués presentar-se instal·lant, configurant o usant qualsevol aplicació lliure. Per altra banda, no s'ha d'oblidar la possibilitat que tenim d'avaluar les característiques d'un programa amb la llibertat d'instal·lar-lo o desinstal·lar-lo sense cap tipus de compromís, és més, en un entorn de producció comercial es pot adaptar o corregir el programari sense dependre del desenvolupador, és a dir, el control passa del proveïdor a l'usuari.

El programari lliure posseeix implementacions per a la majoria d'arquitectures i fabricadores d'equips de tot el món: Intel, AMD, Apple, Sun, ... Si posseeixes una computadora basada en una arquitectura comercialitzada per qualsevol fabricant, probablement hi ha una distribució GNU/Linux orientada al teu equip.

Per acabar aquest passeig per les conseqüències més cridaneres derivades de l'ús d'aquest tipus de programari, insistiré de nou en la llibertat, estesa a la decisió de l'usuari individual de poder triar el tipus de programa que prefereixi o que més s'ajusti a les seves necessitats.

Plantejaré també la peculiaritat que envolta aquesta decisió per a les administracions públiques: Mentre que una entitat comercial privada o un usuari, a títol personal, pot decidir lliurement i respectable el tipus de programari a utilitzar, no ocorre així amb una administració pública, ja que ha de substituir-se la llibertat d'elecció per la responsabilitat, en aquest cas. Qualsevol administració pública és, a més, un usuari de programes amb unes connotacions molt especials a causa de les obligacions concretes amb els administrats. Ha de proporcionar accessos universals als serveis que presta, custodiar dades sensibles de la població i no ha de caure en mans d'estratègies comercials que li coartin la seva independència tecnològica. Per a això, ha de respectar escrupolosament els estàndards, mantenir les dades de forma transparent que assegurin la seva confidencialitat i usar un programari sense hipoteques.

A més, li correspon un paper de lideratge, també en el sector tecnològic, que sigui capaç de generar riquesa en el seu àmbit de gestió. Totes aquestes circumstàncies l'obliguen a triar i potenciar el programari lliure en el seu àmbit competencial.

2. El sistema operatiu GNU/Linux

Definició de sistema operatiu

Un sistema operatiu és un conjunt de programes destinat a permetre la comunicació de l'usuari amb un ordinador i gestionar els seus recursos de manera còmoda i eficient. Comença a treballar quan s'encén l'ordinador, i gestiona el maquinari des dels nivells més bàsics.

Característiques del sistema operatiu GNU/Linux

1. Com sistema operatiu és molt eficient i té un excel·lent disseny.
2. És multitasca, multiusuari, multiplataforma i multiprocessador.

Multitasca	La paraula multitasca descriu l'habilitat d'executar diversos programes al mateix temps. GNU/Linux utilitza la anomenada multitasca preventiva, la qual assegura que tots els programes que s'estan utilitzant en un moment donat seran executats, sent el sistema operatiu l'encarregat de cedir temps de microprocessador a cada programa.
Multiusuari	Molts usuaris usant la mateixa maquina al mateix temps.
Multiplataforma	Les plataformes en les quals en principi es pot utilitzar GNU/Linux són: Alpha, Arm, HPPA, i386, IA64, m68k, Mips, PPC, S390 i Sparc.
Multiprocessador	Suport per a sistemes amb mes d'un processador. Funciona en manera protegida 386.

3. Protegeix la memòria entre processos, de manera que un d'ells no pugui fer caure a la resta del sistema.
4. Carrega només les parts d'un programa que s'usen. GNU/Linux només llegeix del disc aquelles parts d'un programa que estan sent usades actualment.
5. Comparteix la memòria entre programes augmentant la velocitat i disminuint l'ús de memòria: Política de còpia en escriptura per a la compartició de pàgines entre executables: això significa que diversos processos poden usar la mateixa zona de memòria per executar-se. Quan algun intenta escriure en aquesta memòria, la pàgina (4Kb de memòria) es copia a altre lloc.
6. Usa un sistema de memòria virtual per pàgines.
7. La memòria es gestiona com un recurs unificat per als programes d'usuari i per al cau de disc, de tal forma que tota la memòria lliure pot ser usada per a cau i aquesta pot al seu torn ser reduïda quan s'executin grans programes.
8. Permet usar biblioteques enllaçades tant estàticament com dinàmica.
9. Tot el codi font està disponible, incloent el nucli complet i tots els drivers, les eines de desenvolupament i tots els programes d'usuari; a més tot això es pot distribuir lliurement. Hi ha alguns programes comercials que estan sent oferts per a GNU/Linux actualment sense codi font, però tot el que ha estat gratuït segueix sent gratuït.
10. Emulació de 387 en el nucli, de tal forma que els programes no hagin de fer la seva pròpia emulació matemàtica. Qualsevol màquina que executi GNU/Linux semblarà dotada de coprocessador matemàtic. Per descomptat, si l'ordinador ja té una FPU (unitat de coma flotant), aquesta serà usada en lloc de l'emulació, podent fins i tot compilar el teu propi kernel sense l'emulació matemàtica i aconseguir un petit estalvi de memòria.
11. Suport per a molts teclats nacionals o adaptats i és bastant fàcil afegir nous dinàmicament.

12. Consoles virtuals múltiples: diverses sessions de login a través de la consola entre les quals es pot canviar amb les combinacions adequades de tecles (totalment independent del maquinari de vídeo). Es creen dinàmicament i pot tenir fins a 64.
13. Suport per a diversos sistemes d'arxiu comuns, incloent minix-1, Xenix. Té un avançat sistema d'arxius propi amb una capacitat de fins a 4 Tb i noms d'arxius de fins a 255 caràcters de longitud. Accés transparent a particions MS-DOS (o a particions OS/2 FAT) mitjançant un sistema d'arxius especial: no és necessari cap ordre especial per usar la partició MS-DOS, aquesta sembla un sistema d'arxius normal d'Unix (excepte per algunes restriccions en els noms d'arxiu, permisos, i aquestes coses). Suport per a FAT32.
14. Sistema d'arxius de CDROM que llegeix tots els formats estàndard de CDROM. TCP/IP, incloent ftp, telnet, NFS, etc.
15. Diversos protocols de xarxa inclosos en el kernel: TCP, IPv4, IPv6, AX.25, X.25, IPX, DDP, Netrom, etc.

Distribucions

Una distribució és un conjunt d'aplicacions reunides per un grup, empresa o persona per permetre instal·lar fàcilment un sistema GNU/Linux.

Existeixen nombroses distribucions GNU/Linux. Algunes de les distribucions més conegudes són (per ordre alfabètic): CentOS, Damn Small, Debian, Fedora, Gentoo, KANOTIX, KNOPPIX, Kubuntu, Mandriva, MEPIS, openSuSE, PCLinuxOS, Slackware, Ubuntu, Vector, i Xandros. Cadascuna d'elles pot incloure qualsevol quantitat de programari addicional (lliure o no), com alguns que faciliten la instal·lació del sistema i una enorme varietat d'aplicacions, entre ells, entorns gràfics, suites ofimàtiques, servidors web, servidors de correu, servidors FTP, etcètera. La base de cada distribució inclou el nucli GNU/Linux, amb les biblioteques i eines del projecte GNU i de molts altres projectes/grups de programari, com BSD. Usualment s'utilitza la plataforma XFree86, Xorg o, darrerament, Xgl per sostenir interfícies gràfiques.

Una distribució és una manera de facilitar la instal·lació, la configuració i el manteniment d'un sistema GNU/Linux. Al principi, les distribucions es limitaven a recopilar programari lliure, empaquetar-lo en disquets o CDROM i fer la redistribució o vendre'l. Ara les grans distribucions RedHat, SuSE, Mandriva, ... són potents empreses que competeixen entre si per incloure l'últim programari, de vegades també programari propietari, amb instal·lacions gràfiques capaces d'autodetectar el maquinari i que instal·len un sistema sencer en uns quants minuts sense tot just preguntes. Entre les distribucions de GNU/Linux, destaca el projecte Debian. Debian neix com una iniciativa no comercial de la FSF, encara que després s'independitza d'aquesta i va més enllà del propi sistema GNU/Linux. És l'única de les grans distribucions que no té interessos comercials ni empresarials. Són els seus propis usuaris, molt actius, qui mantenen la distribució de manera comunitària, incloses totes les seves estructures de decisió i funcionament. El seu objectiu és recopilar, difondre i promoure l'ús del programari lliure. Reuneix el major catàleg de programari lliure, tots ells provats, mantinguts i documentats per algun desenvolupador voluntari.

distrowatch.com

Si el que voleu és consultar totes les distribucions de GNU/Linux, el seu historial, les diverses versions, els paquets que inclou cada versió d'una distribució, les notícies relacionades amb cada distribució, les anàlisis que han fet experts sobre una o altra distribució, articles, estadístiques, etc., la millor web és <http://distrowatch.com>

Tipus de cd's: d'instal·lació i live-cd

Hi ha dos tipus de cd's: els d'instal·lació del sistema operatiu, que poden ser un o més, i els live-cd, traduït en ocasions com CD viu o CD autònom, és un sistema operatiu (normalment acompanyat d'un conjunt d'aplicacions) emmagatzemat en un mitjà extraïble, tradicionalment un CD (d'aquí el seu nom), que pot executar-se des d'aquest

sense necessitat d'instal·lar-lo en el disc dur de l'ordinador, per a això usa la memòria RAM com disc dur virtual. Alguns live-cd's inclouen una eina que permet instal·lar-los en el disc dur. Altra característica és que en general no s'efectuen canvis en la computadora utilitzada, encara que alguns poden emmagatzemar preferències si així ho desitges. Per usar un live-cd és necessari obtenir-ne un (molts d'ells es troben com una imatge ISO que pot baixar-se d'Internet i gravar-se en disc) i configurar l'ordinador perquè arrenqui des de la unitat lectora, reiniciant després l'ordinador amb el disc en la lectora, amb la qual cosa el live-cd s'iniciarà automàticament.

Com aconseguir els cd's

Els cd's es poden descarregar des de la pàgina web oficial de la distribució. En el cas d'Ubuntu des de <http://www.ubuntu.com/download>

També els podem demanar i que ens els facin arribar a casa sense cap tipus de cost (ni tan sols les despeses de correu) des de <https://shipit.ubuntu.com>

Aquest servei no és exclusiu de la distribució Ubuntu. La Junta de Andalucia, sense anar més lluny, ofereix la possibilitat d'obtenir el cd d'instal·lació des de la seva web <http://www.guadalinex.org/cds.php> sempre que ens hàgim registrat prèviament.

Per últim indicar la manera més simple d'aconseguir una còpia dels cd's que és demanar-li a un amic.

3. La distribució Ubuntu

El projecte va néixer per iniciativa d'alguns programadors dels projectes [Debian](#), [Gnome](#) i [Arch](#) que es trobaven decebuts amb la manera d'operar del projecte Debian.

D'acord amb els seus fundadors, Debian era un projecte massa burocràtic on no existien responsabilitats definides i on qualsevol proposta interessant s'ofegava en un mar de discussions. Així mateix, Debian no posava èmfasis a estabilitzar el desenvolupament de les seves versions de prova i només proporcionava auditories de seguretat a la seva versió estable, la qual era utilitzada només per una minoria a causa de la poca o nul·la vigència que posseïa en termes de la tecnologia GNU/Linux actual.

Després de formar un grup multidisciplinari, els programadors van decidir buscar el suport econòmic de [Mark Shuttleworth](#), un emprenedor sud-africà que va veure amb simpatia el projecte i va decidir convertir-lo en una iniciativa autosostenible, combinant la seva experiència en la creació de noves empreses amb el talent i l'experiència dels programadors de la plataforma GNU/Linux. D'aquesta forma va néixer l'empresa Canonical, la qual s'encarrega de sostenir econòmicament el projecte mitjançant la comercialització de serveis i suport tècnic a altres empreses. Mentre els programadors armaven el sistema, Shuttleworth va aprofitar l'ocasió per aplicar una petita campanya de màrqueting per despertar interès en "la distribució-sense-nom" (en anglès: the no-name-distro).

Després de diversos mesos de treball i un breu període de proves, la primera versió d'Ubuntu (Warty Warthog) va ser llançada el 20 d'octubre de 2004.

Característiques

- Basada en la distribució [Debian](#).
- Disponible en 3 arquitectures: Intel [x86](#), [AMD64](#), [PowerPC](#). (S'estan plantejant si mantenir l'arquitectura PowerPC o abandonar-la.)
- Els desenvolupadors d'Ubuntu es basen en gran mesura en el treball de les comunitats de Debian i [GNOME](#).
- Les versions estables s'alliberen cada 6 mesos i es mantenen actualitzades en matèria de seguretat fins a 18 mesos després del seu llançament.
- L'escriptori oficial és [Gnome](#) i se sincronitzen amb els seus alliberaments.
- Per centrar-se a solucionar ràpidament els bugs, conflictes de paquets, etc. es decideix eliminar certs paquets del component main, ja que no són populars o simplement es van triar de forma arbitrària o les seves bases de suport al programari lliure. Per tals motius inicialment KDE no es trobava amb més suport del que lliuraven els mantenidors de Debian en els seus repositoris, raó per la qual es va sumar la comunitat de [KDE](#) distribuint una distribució anomenada [Kubuntu](#).
- Hi ha una versió d'Ubuntu basada en l'entorn d'escriptori [Xfce](#), el nom del qual és [Xubuntu](#).
- El navegador web oficial és [Mozilla Firefox](#).
- El sistema inclou funcions avançades de seguretat i entre les seves polítiques es troba el no activar processos latents per omissió al moment d'instal·lar-se. Per això mateix, no hi ha un [firewall](#) predeterminat, ja que no existeixen serveis que puguin atemptar a la seguretat del sistema.
- Per a tasques administratives inclou una eina anomenada [sudo](#) (similar al [Mac OS X](#)), amb la qual s'evita l'ús de l'usuari [root](#).
- Millorar l'accessibilitat i la internacionalització, de manera que el programari estigui disponible per a tanta gent com sigui possible. En la versió 5.04, és l'[UTF-8](#) la codificació de caràcters per defecte.
- No només té com llaç a Debian l'ús del mateix format de paquets [deb](#), Ubuntu té unions molt fortes amb aquesta comunitat, contribuint qualsevol canvi directament i immediatament, més que anunciant-los. Això succeeix en els temps de llançament. Molts dels desenvolupadors d'Ubuntu són també responsables dels paquets importants dintre de la distribució de Debian.

- Tots els llançaments d'Ubuntu es proporcionen sense cost algun. Els [cd's](#) de la distribució s'envien de forma gratuïta a qualsevol persona que els sol·liciti. També és possible descarregar les [imatges ISO](#) dels discos per transferència directa o sota la tecnologia [Bittorrent](#).
- Ubuntu no cobra honoraris per la subscripció de les millores de la "Edició Enterprise".

Organització de paquets

Ubuntu divideix tot el programari en quatre seccions, anomenades els "components", per reflectir diferències en llicències i la prioritat amb la qual s'atenen els problemes que informen els usuaris. Per defecte, s'instal·la una selecció de paquets que cobreix les necessitats bàsiques de la majoria dels usuaris de computadores.

main	El component main conté solament els paquets que compleixen els requisits de la llicència d'Ubuntu, i per als quals hi ha suport disponible per part del seu equip. Aquest està pensat perquè inclogui tot el necessari per a la majoria dels sistemes GNU/Linux d'ús general. Els paquets d'aquest component posseeixen ajuda tècnica garantida i millores de seguretat oportunes.
restricted	El component restricted conté el programari que està suportat pels desenvolupadors d'Ubuntu a causa de la seva importància, però que no està disponible sota cap tipus de llicència lliure per a incloure en main. En aquest lloc s'inclouen els paquets tals com els controladors propietaris d'algunes targetes gràfiques, com per exemple, els de nVIDIA. El nivell de l'ajuda és més limitat que per a main, ja que els desenvolupadors poden no tenir accés al codi font.
universe	El component universe conté una àmplia gamma del programari, que pot o no tenir una llicència restrictiva, però que no rep suport per part de l'equip d'Ubuntu. Això permet que els usuaris instal·lin tota classe de programes en el sistema, però els guarda en un lloc a part dels paquets suportats: main i restricted.
multiverse	Finalment, es troba el component multiverse, que conté els paquets sense suport degut al fet que no compleixen els requisits del Programari Lliure.

Els paquets d'Ubuntu generalment es basen en els paquets de la branca inestable (Sid) de Debian.

Llançaments

Una nova versió d'Ubuntu és llançada cada sis mesos, i cada llançament té un "nom" i un número de versió. El número de versió està basat en la data de llançament, per tant, per exemple, la versió 5.04 va ser llançada a l'abril (04) de 2005 (5). Sota hi ha una llista de les versions prèvies.

Versió	Data de llançament	Nom
4.10	20 d'octubre, 2004	The Warty Warthog
5.04	8 d'abril, 2005	The Hoary Hedgehog
5.10	13 d'octubre, 2005	The Breezy Badger
6.06 LTS	1 de juny, 2006	The Dapper Drake
6.10	26 d'octubre, 2006	The Edgy Eft
7.04	19 d'abril, 2007	The Feisty Fawn

Versió	Data de llançament	Nom
7.10	19 d'octubre, 2007	The Gutsy Gibbon
8.04 LTS	24 d'abril, 2008	The Hardy Heron
8.10	Octubre, 2008	The Intrepid Ibex

En el futur, haurà un llançament anomenat "The Grumpy Groundhog" que quedarà permanentment en versió "inestable" només per a propòsits de proves.

Web oficial <http://www.ubuntu.com> o <http://www.ubuntulinux.org>

4. Administració i configuració

L'estructura de directoris

L'estructura de directoris que segueix GNU/Linux és semblant a la de qualsevol sistema UNIX. No tenim una unitat per cada dispositiu físic com passa amb Windows, sinó que tots els discs durs, particions, etc., pegen d'un sistema de directoris jeràrquics en forma d'arbre.

A continuació, s'explica breument el contingut i el significat dels directoris més comuns:

Directori	Descripció
/	Arrel de l'estructura de directoris
/bin	Conté les comandes bàsiques i algunes aplicacions
/dev	Conté enllaços a la major part del maquinari
/etc	Conté els arxius de configuració de la majoria de programes
/home	Conté els arxius personals dels usuaris
/lib	Conté diverses llibreries
/mnt	Conté enllaços a altres dispositius de dades (CDROM, Floppy, etc.)
/proc	Conté informació temporal del sistema.
/sbin	Conté comandes administratives
/tmp	Conté arxius temporals
/usr	Conté arxius i programes
/var	Conté dades diverses normalment de mida creixent (arxius de log, correus, etc.)

L'interpret d'ordres o SHELL

Quan activem una consola de text el sistema ens convida a iniciar la sessió:

```

Ubuntu 7.10 "Edgy Eft" NOMDELAMAQUINA tty1

NOMDELAMAQUINA Login:
Password:

```

L'interpret d'ordres és el programa que fa d'interfase entre l'usuari interactiu i el sistema operatiu. El `shell` instal·lat s'anomena `bash` i és una evolució del `Bourne shell`. És l'interpret d'ordres que incorpora GNU/Linux i que ens indica que espera que li introduïm un usuari del sistema creat prèviament (o `root`, que és el superusuari del sistema) junt amb la seva contrasenya.

Un cop li haguem donat les dues dades, ens apareixerà la següent línia de text:

```
usuari@maquina:directoriactual$ si l'usuari amb que hem entrat no era root, o
maquina:directoriactual# si l'usuari amb que hem entrat era root amb
aquesta línia, el sistema ens indica que està llest
per treballar i esperant una comanda per executar.
```

Els dos objectes bàsics amb els que un usuari treballa un cop inicia una sessió són els fitxers i els processos (un procés és un programa en execució). La `shell` és capaç de llançar a executar altres programes (crear altres processos anomenats processos fills). L'execució d'aquests nous processos pot ser asíncrona (el procés `shell` es bloqueja en espera que finalitzi la execució del programa que ha llançat) o síncrona (el procés pare `-shell-` i el procés fill s'executen concurrentment). Per indicar la execució síncrona s'ha d'incloure al final de la línia de comandes el metacaracter `'&'`.

Per qualsevol consulta relacionada amb el `shell` es recomana utilitzar les pàgines del manual online (comanda `man`).

Visió d'usuari d'un sistema GNU/Linux

Linux és un sistema operatiu multiusuari, multitasca. El fet de que sigui multitasca permet l'execució concurrent de més d'un programa a la vegada. El fet que sigui multiusuari permet protegir les dades entre els diferents usuaris que tenen accés al sistema aconseguint d'aquesta l'accés controlat a tots els recursos del sistema.

Tipus d'usuari

Super-user	S'identifica pel nom <code>root</code> i té tots els privilegis sobre tots els recursos del sistema, pot gestionar (destruir, modificar...) informació pròpia i de qualsevol altre usuari sense cap tipus de restriccions.
Usuari interactiu	Usuari que té accés al sistema amb un identificador igual a <code>login_name</code> i un codi secret d'accés <code>password</code> . Aquest tipus d'usuari té l'accés restringit i controlat als recursos del sistema (dispositius, fitxers, processos...). Pot gestionar (crear, destruir, modificar...) informació seva sense cap tipus de restriccions, mentre que té accés restringit a la informació que pertany a altres usuaris.
Altres usuaris	<code>ftp</code> , <code>anonymous</code> , <code>sync</code> , etc. Aquests usuaris no tenen, en general, <code>password</code> . L'accés al recursos sol ser bastant restringit, normalment tant sols es poden llegir el contingut de certs directoris públics a tots els usuaris de la xarxa o executar una funció molt concreta del sistema

Tots els usuaris pertanyen a un grup d'usuaris com a mínim. El grup estableix uns privilegis o drets d'accés comuns a tots els usuaris que hi pertanyen.

La informació de tots els usuaris que tenen accés al sistema es troba dintre d'un fitxer que té per nom `/etc/passwd`. En aquest fitxer per cada usuari hi ha emmagatzemat un conjunt d'informació entre la que podem trobar dos identificadors associats a l'usuari en qüestió, aquests identificadors són:

<code>uid</code>	identificador d'usuari, aquest és únic per cada usuari que té accés
<code>gid</code>	identificador de grup, aquest identificador permet agrupar usuaris dintre d'un mateix grup i assignar els drets d'accés comuns com a grup

Aquests identificadors determinaran per a cada usuari a quins recursos poden accedir (fitxers, directoris, dispositius...) i quin tipus d'accés poden realitzar sobre aquests recursos (llegir, modificar, crear, esborrar, etc.)

L'entorn

Els noms dels directoris i les carpetes, en GNU/Linux poden tenir més de 200 caràcters, sense que obligatòriament hagin de tenir una extensió concreta. A part, s'ha de tenir en compte que són "case sensitive", és a dir, que diferencien entre majúscules i minúscules (per exemple: `readme` i `Readme` són dos arxius diferents).

Els noms es poden compondre de qualsevol caràcter, encara que es recomana utilitzar els caràcters A-Z, a-z, els números (0-9), el guió (-) i el guió baix (_).

Hi ha una sèrie de caràcters que tenen un significat especial:

- (~) caràcter que equival a `/home/nomusuari` (per usuaris diferents a `root`) o `/root` (per l'usuari `root`).
- (*) comodí que equival a qualsevol conjunt de caràcters. Inclou també el conjunt buit.
- (?) comodí que equival a qualsevol caràcter.

- (.) si apareix en un camí de directoris, es refereix al directori actual.
- (..) si apareix en un camí de directoris, es refereix al directori anterior.

Al igual que un conjunt de tecles:

(TAB)	permet auto completar la comanda.
(Cursor amunt o avall)	permet visualitzar les comandes que s'han anat introduint.
(Insert) o (AltGr+4)	escriu el símbol (~)
(ALT+F1..F6)	canvia de sessió / terminal.
(Shift+AVPG..REPG)	puja o baixa la pantalla de text actual.
(Ctrl + d)	logout
(Ctrl + c)	posa fi a l'execució del programa

Resum de les ordres bàsiques

- cd** Serveix per canviar al directori especificat com a paràmetre.
Sintaxi: cd directori
Exemple: cd /etc
Descripció de l'exemple: Ens situa per treballar en el directori /etc
Casos per observar: cd ., cd .., cd /
- pwd** Ens informa del directori on estem treballant actualment.
Sintaxi: pwd
Exemple: cd /etc; pwd
Descripció de l'exemple: Ens informa que ens trobem treballant al directori /etc
- man** Serveix per obtenir l'ajuda de qualsevol comanda.
Sintaxi: man comanda
Exemple: man pwd
Descripció de l'exemple: Ens dona l'ajuda de la comanda vista anteriorment: pwd
- mkdir** Serveix per crear un directori a partir del directori actual.
Sintaxi: mkdir directori
Exemple: mkdir apunts
Descripció de l'exemple: crea el directori apunts a partir de l'actual.
- rmdir** Serveix per esborrar un directori. Si s'especifica el paràmetre -r, permet eliminar directoris sense necessitar que estiguin buits.
Sintaxi: rmdir paràmetres directori
Exemple: rmdir apunts
Descripció de l'exemple: elimina el directori apunts sempre i quan estigui buit (sense cap arxiu ni directori dintre seu).
- cp** Serveix per copiar arxius d'un directori a un altre.
Sintaxi: cp paràmetres origen destí
Exemple: cp /etc/network/interfaces ~
Descripció de l'exemple: copia l'arxiu interfaces al directori de treball de l'usuari.
Casos per observar: cp *, cp ~, cp -p,
- ls** Serveix per llistar el contingut d'un directori
Sintaxi: ls paràmetres directori
Exemple: ls -la
Descripció de l'exemple: llista en una columna tots els arxius del directori actual.

Casos per observar: `ls -la`, `ls -color`

`mv` Serveix per moure arxius i/o directoris de l'origen al destí.
Sintaxi: `mv paràmetres origen destí`
Exemple: `mv apunts apunts_linux`
Descripció de l'exemple: mou del contingut del directori `apunts` al directori `apunts_linux`.

Movent-nos pels directoris - cd

L'ordre per moure'ns per l'estructura de la xarxa és '`cd`' i el seu format és el següent:

```
cd [director]
```

On '`cd`' és l'ordre i el que està entre claudàtors és l'argument, a l'estar entre claudàtors és opcional.

Per moure'ns d'un directori a un altre usem aquesta ordre, tenim 2 formes de nomenar un directori: la primera és per la seva ruta absoluta i la segona és per la seva ruta relativa. L'absoluta es refereix al nom complet començant des de '/', un exemple seria anar a `/etc` (joestic en `/home/ramon`).

```
ramon@portatil:~$ cd /etc
ramon@portatil:/etc$
```

Això té un problema, ja que sempre hem d'escriure la ruta completa on vulguem anar. Per no haver de fer-lo, usarem les rutes relatives: posem el nom del directori que volem entrar però a partir des del que estem. Per veure això tornarem a `/home/ramon` (en el vostre cas el directori de treball vostre) i d'allí anirem a `/home/ramon/docs`.

```
ramon@portatil:/etc$ cd /home/ramon
ramon@portatil:~$ cd docs
ramon@portatil:~/docs$
```

En el primer cas hem usat de nou una ruta absoluta i en el segon com ja estàvem en `/home/ramon` hem posat solament `docs` per entrar en `/home/ramon/docs`, que seria la seva ruta absoluta.

Dintre de tot directori existeixen dos directoris especials que són '.' i '..'. El primer fa referència al directori actual, és a dir, si fas '`cd .`' et quedes on estàs (el directori especial '.' ja veureu més endavant l'útil que és), i el segon fa referència al directori pare, o sigui, si estem en `/home/ramon` i fem '`cd ..`' acabarem en `/home`.

```
ramon@portatil:~$ cd .
ramon@portatil:~$ cd ..
ramon@portatil:~/home$
```

Aquests dos directoris especials els usarem en la rutes relatives, és el mateix fer '`cd ./docs`' que '`cd docs`' que per a aquesta ordre en concret ens dona el mateix però en uns altres pot ser que necessiteu usar './'. Imagineu-vos que des de `/home/ramon` voleu entrar (si tinguéssiu altre usuari anomenat així) en `/home/pepe`, per no haver d'escriure la ruta absoluta recorrerem a '`cd ../pepe`' i aconseguirem mitjançant una ruta relativa donar un salt endarrere en l'arbre i entrar en 'pepe'

```
ramon@portatil:~$ cd docs
ramon@portatil:~/docs$ cd ..
ramon@portatil:~$ cd ./docs
ramon@portatil:~/docs$ cd ..
ramon@portatil:~$ cd ../pepe
ramon@portatil:~/home/pepe$
```

Bé doncs no es vagin encara, àdhuc hi ha més. Hi ha una referència especial, estiguem on estiguem si fem '`cd`' o bé '`cd ~`' tornem al nostre directori de treball ja que `~` (sota

consola ho aconseguim prement la tecla AltGr + 4, bé la tecla Insert o Alt + 126) és una referència al nostre directori de treball.

```
ramon@portatil:~$ cd /usr/bin
ramon@portatil:~$ cd ~
ramon@portatil:~$
```

Amb això ja hem vist aquesta ordre tan simple però útil.

M'he perdut. On estic? - pwd

Si ens perdem, bé per que ens hem mogut molt movent-nos per l'arbre de directoris o bé perquè el nostre prompt no ens indica a cada moment on estem, tenim una ordre ben senzilla:

pwd

```
ramon@portatil:~$ pwd
/home/ramon
ramon@portatil:~$
```

Abans de continuar cal comentar una cosa, per evitar confusions explicaré una mica els 'alias'. L'interpret d'ordres ens dona l'opció de definir-nos el que denominarem 'alias', per exemple, pots fer que la paraula 'casa' executi 'cd', sembla una bonica forma de crear ordres al nostre gust. Algunes de les ordres que aquí explicaré tenen definits alies en moltes distribucions pel que el resultat pot ser que no sigui el mateix. Si això us ocorre teclegeu 'alias ORDREQUEFALLA=ORDREQUEFALLA' (compte, sense espais en l'igual), per exemple, si quan féssiu 'cd' us manés a /usr/bin (és un exemple rar, però bé) bastaria amb executar 'alias cd=cd'.

Per veure quins alies teniu definits executeu 'alias -p'

```
ramon@portatil:~$ alias -p
. alias ls='ls --color=auto'
ramon@portatil:~$
```

El que veurem a continuació és molt probable que hagi de fer-lo perquè surti el mateix que t'estic explicant. L'ús d'alies sobretot s'utilitza per executar per defecte opcions molt útils sense haver d'escriure-les sempre, fins i tot si vens del món de MS-DOS pots definir totes les teves ordres de sempre amb els anàlegs en GNU/Linux, útil, veritat? ;) (encara que això últim no ho recomano, no és bo agafar vicis dolents).

Nota: els alies que definim s'esfumaran quan tanquem la nostra sessió o reiniciem la màquina, però no tingueu por. Ja explicarem més endavant com definir les coses permanentment.

Llistant arxius - ls

Bé, ja sabem moure'ns, però això no ens serveix de molt si no sabem què hi ha en els nostres directoris. Per a això tenim una bona ordre:

ls [opcions][directori]

Ara executarem 'ls' tal qual i ens trobarem amb la següent sortida:

```
ramon@portatil:/documents$ ls
bookmarks.html  Open Office  Reconèixer SATA en l'arrencada
ramon@portatil:/documents$
```

Ens trobem amb una sortida una mica pobra, perquè en alguns casos no sabrem ni diferenciar què és cada cosa: un fitxer? un directori?. Amb l'opció '-F' obtindrem més informació:

```
ramon@portatil:/documents$ ls -F
bookmarks.html  Open Office/  Reconèixer SATA en l'arrencada
```

```
ramon@portatil:/documents$
```

Ara podem apreciar que als directoris al final li afegixen '/'. A 'ls' també li podem indicar el directori que volem mirar:

```
ramon@portatil:/documents$ ls -F /usr/local/bin
mencoder*  mplayer*
ramon@portatil:/documents$
```

En aquest cas a més veiem un * al final d'aquests fitxers, el que ens indica que són executables. També podem trobar-nos amb una @ indicant que és un enllaç. Altra opció interessant i que ens serà útil és '-a' que ens mostra els fitxers ocults (recorda que en GNU/Linux tot són fitxers).

```
ramon@portatil:/documents$ ls -F -a
./  ../  bookmarks.html  Open Office/  Reconèixer SATA en
l'arrencada
ramon@portatil:/documents$
```

Podem observar que apareixen més fitxers que estaven ocults. En GNU/Linux, qualsevol fitxer que comenci per '.' és un fitxer ocult contant els directoris especials '.' i '..' que ja us vaig comentar abans. Una opció realment útil i que us interessarà és '-l':

```
ramon@portatil:/documents$ ls -Fa -l
.  total 60
drwxr-xr-x   3 ramon ramon   4096 2005-03-31 10:43 ./
drwxr-xr-x  37 ramon ramon   4096 2005-03-31 10:42 ../
-rw-r--r--   1 ramon ramon  44409 2005-03-27 23:55 bookmarks.html
drwxr-xr-x   2 ramon ramon   4096 1970-01-01 01:00 Open Office/
-rw-r--r--   1 ramon ramon   1735 2005-03-30 02:05 Reconèixer SATA en
l'arrencada
ramon@portatil:/documents$
```

Per començar veureu que es poden barrejar opcions i que '-l' ens dona una gran quantitat d'informació. La primera indica el tipus de fitxer i els permisos (per exemple, drwxr-xr-x), els permisos no els explicaré ara per que no toca, però si comentar que la primera lletra indica quin tipus de fitxer és ('d' ens indica que és directori, '-' que és un fitxer normal, 'c' que és un dispositiu orientat a caràcter, 'b' dispositiu orientat a bloc i 'l' indica que és un enllaç). El següent ens indica: el nombre d'enllaços que existeixen a ell, el propietari, el grup propietari, la grandària en bytes, la data i el nom.

Si quan fem un 'ls' la llista és molt llarga podem moure'ns endarrere amb Shift+RePág, o bé usar la canonada i 'more'. Un exemple:

```
ramon@portatil:/bin$ ls -la | more
total 3140
drwxr-xr-x   2 root root   4096 2005-03-29 10:25 .
drwxr-xr-x  22 root root   4096 2005-03-27 18:52 ..
-rwxr-xr-x   1 root root   2756 2005-03-17 20:09 arch
-rwxr-xr-x   1 root root 643852 2005-01-13 22:44 bash
-rwxr-xr-x   1 root root  15960 2005-03-18 14:22 cat
..... (ometo arxius perquè no ocupi tant).....
-rwxr-xr-x   1 root root  11256 2005-03-18 14:22 false
-rwxr-xr-x   1 root root   5568 2005-03-18 16:37 fgconsole
--Més--
```

Finalment, ja només ensenyar-vos altra opció que us agradarà moltíssim '--color':

```
ramon@portatil:/documents$ ls --color
bookmarks.html  Open Office  Reconèixer SATA en l'arrencada
ramon@portatil:/documents$
```

La veritat s'explica per ella sola, no? No seguirem amb més opcions de 'ls' perquè hi ha gairebé tantes com lletres de l'abecedari, tant en majúscules com en minúscules.

Creant directoris - mkdir

Per començar, després d'haver après a moure'ns pel sistema el primer que veurem és com crear un directori, per a això tenim la següent instrucció:

```
mkdir [opcions] directori [directori2 ... directori3]
```

On com podeu apreciar podem crear d'un a diversos directoris alhora, així que ara toca provar-lo.

```
ramon@portatil:$ mkdir curs
ramon@portatil:$ cd curs
ramon@portatil:/curs$ mkdir dir1 dir2
ramon@portatil:/curs$ ls -l
. total 8
drwxr-xr-x  2 ramon ramon 4096 2005-03-31 10:52 dir1
drwxr-xr-x  2 ramon ramon 4096 2005-03-31 10:52 dir2
ramon@portatil:/curs$
```

Com podeu veure l'ús és ben senzill, com opcions destacarem '-p' que crea els directoris pares que no existeixen, com veiem en el següent exemple.

```
ramon@portatil:/curs$ mkdir -p ./dir1/NOEXISTEIXO/ops
ramon@portatil:/curs$ cd dir1
ramon@portatil:/curs/dir1$ ls
NOEXISTEIXO
ramon@portatil:/curs/dir1$ cd NOEXISTEIXO/
ramon@portatil:/curs/dir1/NOEXISTEIXO$ ls
ops
ramon@portatil:/curs/dir1/NOEXISTEIXO$
```

I finalment l'opció '-m', que ens dóna la possibilitat d'indicar-li els permisos que tindran els nostres directoris, però... com encara no ho hem vist en el curs queda pendent que ho proveu vosaltres mateixos quan vegem com són els permisos. L'opció '-m' va seguit de la manera abans de posar els directoris.

Eliminant directoris - rmdir

Ja que hem après a crear-los cal aprendre l'opció contrària, i ja que mkdir venia de make directory ara necessitem un remove directory:

```
rmdir [opcions] directori [directori2 ... directori3]
```

Així que el que ens queda és començar a provar-lo:

```
ramon@portatil:/curs/dir1/NOEXISTEIXO$ rmdir ops
ramon@portatil:/curs/dir1/NOEXISTEIXO$ cd ..
ramon@portatil:/curs/dir1$ cd ..
ramon@portatil:/curs$ rmdir ./dir1/NOEXISTEIXO/
ramon@portatil:/curs$ ls
dir1  dir2
ramon@portatil:/curs$ rmdir dir1 dir2
ramon@portatil:/curs$ ls
ramon@portatil:/curs$
```

Com podeu veure es poden esborrar diversos alhora, igual que amb mkdir creàvem diversos alhora. En aquest cas per a rmdir solament comentaré una opció interessant i és '-p'. Aquesta opció el que fa és esborrar recursivament la ruta completa que li indiquem, per exemple, si tenim /a/b/c un 'rmdir -p /a/b/c' és equivalent a 'rmdir /a/b/c' + 'rmdir /a/b' + rmdir 'a'. Anem a l'exemple:

```
ramon@portatil:/curs$ mkdir -p ./dir1/u/dos
ramon@portatil:/curs$ ls
dir1
ramon@portatil:/curs$ ls dir1
u
```

```
ramon@portatil:/curs$ ls ./dir1/u
dos
ramon@portatil:/curs$ rmdir -p dir1/u/dos
ramon@portatil:/curs$
```

Tampoc m'agrada teclejar tant - [TAB]

Això potser ho havia d'haver explicat abans, però m'acabo de'n recordar ara mateix. L'interpret d'ordres, per facilitar-nos el nostre treball, emplena per nosaltres amb el més lògic els arxius i les ordres, simplement començant a escriure'ls i prement tabulador. Per exemple, no ens recordem de com s'escrivia l'ordre 'mkdir' i solament recordem de 'mkd' premem el tabulador i ens ho emplena automàticament o ens indicarà quines ordres comencen per 'mdk'. En el meu cas tabulo i em xiula, no em surt res, torno a tabular i ara sí em dóna les ordres que comencen per mkd:

```
ramon@portatil:/bin$ mkd
mkdir      mkdirhier  mkdosfs
ramon@portatil:/bin$ mkd
```

Ara si afegeixo una 'i' em completa amb 'mkdir', ja que 'mkdir' està contingut dintre del nom de 'mkdirhier'. Ara tenim el que buscàvem, però si tornem a tabular ens escriurà 'mkdirhier'. Amb el que arribem a la conclusió que emplenarà quan: a) No hi ha altre ordre que coincideixi amb la cadena que hàgim escrit, i b) Quan el nom d'una ordre estigui contingut en altre podent seguir tabulant per arribar a l'altre.

Quan existeixen moltes possibilitats, tantes que no caben en pantalla, abans ens preguntarà si mostrar-lo o no. Per exemple, executar:

```
ramon@portatil:/bin$ m
Display all 117 possibilities? (y or n)
ramon@portatil:/bin$ m
```

En el meu cas he contestat amb 'n' ja que no vull veure la llista completa. Per exemple, si volguéssim veure tots els executables accessibles no tindríem més que tabular dues vegades sense escriure res.

```
ramon@portatil:/bin$
Display all 2176 possibilities? (y or n)
ramon@portatil:/bin$
```

Copiant - cp

I una vegada vist això arribem al moment en el qual crearem còpies dels arxius, per a això tenim la següent ordre:

cp [Opcions] Origen Destinació

L'ús és ben senzill, només cal indicar l'origen i la destinació del que volem copiar:

```
ramon@portatil:$ cp /curs/dir1/bookmarks.html /curs/dir2
ramon@portatil:$ cd curs/dir2
ramon@portatil:/curs/dir2$ ls
bookmarks.html
ramon@portatil:/curs/dir2$
```

Com veiem no se'ns demana confirmació per copiar, i pogués ocórrer que ens haguéssim equivocat i sobreescrivíssim un arxiu que ja tingués sense que ens preguntés, ja que 'cp' també renombra incloent el nom en la destinació.

'cp' és encara més versàtil, ja que amb l'opció '-r' podem copiar directoris sencers encara que és preferible usar l'opció '-R' perquè '-r' no té un comportament definit (o sigui que pot passar qualsevol cosa) si es copien fitxers que no són normals com poden ser un dispositiu. Vegem un exemple:

```
ramon@portatil:/curs$ mkdir copiadebin
```

```

ramon@portatil:/curs$ cp -R /bin ./copiadebin
ramon@portatil:/curs$ cd copiadebin
ramon@portatil:/curs/copiadebin$ ls
bin
ramon@portatil:/curs/copiadebin$ cd bin
ramon@portatil:/curs/copiadebin/bin$ ls
arch      dmesg          hostname      mkdir         pidof
sleep     zcmp
bash      dnsdomainname ip             mknod        ping
stty      zdiff
..... (ometo arxius perquè no ocupi tant).....
dir       gzip           mkbimage     netstat       sh
zcat
ramon@portatil:/curs/copiadebin/bin$

```

I per a rematar només queda comentar que 'cp' també té l'opció '-p', que és igual que en 'mkdir' i 'rmdir' així que sobren les explicacions. Per cert, 'cp' admet més d'un origen, així que pots copiar diversos arxius en una sola línia a una mateixa destinació.

Esborrant - rm

Si 'cp' ve de copy llavors alguns ja haureu endevinat quin ordre és la d'esborrar:

rm [opcions] llistadefitxersaesborrar

Vegem l'ús més simple de rm:

```

ramon@portatil:/curs/dir1$ cp bookmarks.html tesborrare
ramon@portatil:/curs/dir1$ ls
bookmarks.html tesborrare
ramon@portatil:/curs/dir1$ rm tesborrare
ramon@portatil:/curs/dir1$ ls
bookmarks.html
ramon@portatil:/curs/dir1$

```

Igual que 'cp', 'rm' també té l'opció '-i' i també pot esborrar directoris sencers amb '-r' i '-R' (en aquest cas ambdós són iguals), encara que també s'esborren amb '-d', però '-d' no té un caràcter recursiu i deixa desenllaçats els fitxers que conté (o sigui que és una cagada, així que millor no usar-la) el que vol dir que es queden com diríem sense cap directori que els contingui, pel que cal tenir molta cura amb aquesta opció.

```

ramon@portatil:/curs$ rm -ri dir1 dir2
rm: descendir al directori «dir1»? (s/n) s
. rm: esborrar el fitxer regular «dir1/bookmarks.html»? (s/n) s
. rm: esborrar el directori «dir1»? (s/n) s
. rm: descendir al directori «dir2»? (s/n) s
. rm: esborrar el fitxer regular «dir2/bookmarks.html»? (s/n) s
. rm: esborrar el directori «dir2»? (s/n) s
. ramon@portatil:/curs$

```

Solament queda apuntar que, tant 'cp' com 'rm', si '-i' fa que preguntí l'opció contrària és '-f' que no preguntarà res de res.

Movent - mv

Vinga, que segur que alguns ja s'han imaginat que per moure és:

mv [Opcions] origen destinació

Bé, 'mv' equival a copiar i esborrar, i igual que 'cp' admet diversos orígens i un directori destinació. Així que amb els exemples vists abans amb 'cp' sobren els exemples, excepte per al cas que l'origen i la destinació són el mateix, però en la destinació s'indica un nom de fitxer amb la qual cosa el que fem és renombrar:

```

ramon@portatil:/curs/dir1$ ls
bookmarks.html
ramon@portatil:/curs/dir1$ mv bookmarks.html enllacos

```

```
ramon@portatil:/curs/dir1$ ls
enllacos
ramon@portatil:/curs/dir1$
```

Com veieu el directori origen i destinació és el mateix, per la qual cosa en la destinació hem indicat un canvi de nom.

Comodins - *

Veurem l'ús dels comodins, que no és ni més ni menys que una característica de l'interpret d'ordres que ens permet referir-nos a un conjunt de fitxers alhora.

Començarem veient primer el comodí '*': el comodí * fa referència a qualsevol caràcter o cadena de caràcters (és a dir, substitueix a un, cap o molts caràcters). Per a entendre'l bé, ja que l'explicació pot ser una mica confusa, vegem com sempre un exemple simple:

```
ramon@portatil:/curs/dir1$ cd /dev
ramon@portatil:/dev$ ls l*
.  log  lp0  lvm

loop:
0.
ramon@portatil:/dev$
```

El que acabem d'indicar-li a ls amb l'ús de * és que ens llisti tots els fitxers que comencin per 'l' seguit de qualsevol cosa, fins i tot ens llista el contingut d'un directori que comença per 'l'. Altre exemple, perquè ens quedi definitivament clar, pot ser aquest:

```
ramon@portatil:/dev$ ls *rr*
stderr
ramon@portatil:/dev$
```

En aquest cas el que hem fet és dir-li que ens llisti tots els fitxers que continguin la cadena 'rr' (fins i tot els que comencin o acabin per 'rr' ja que * fins i tot pot ser cap caràcter).

Si sou observadors i/o curiosos veureu que no funciona amb els fitxers ocults (recordeu que comencen per '.'), perquè afectaria entre altres coses als directoris especials '.' i '..', i segons amb quines ordres ho utilitzem pot provocar veritables problemes (com per exemple rm).

El següent i últim comodí és '?' que fa referència a qualsevol caràcter, però en aquest cas només a un (no pot ser cap com el cas de *). Per veure'l clar vegem un exemple (per variar):

```
ramon@portatil:/bin$ ls ?a??
bash  dash  date  nano
ramon@portatil:/bin$
```

En aquest cas li preguntem a ls per tots aquells arxius que tenen longitud 4 i el segon caràcter és una 'a', senzill, veritat?

Pots combinar tant * com ? per aconseguir el resultat desitjat:

```
ramon@portatil:/bin$ ls ?a?*
bash  cat  dash  dati  false  nano  tar
ramon@portatil:/bin$
```

El que hem fet és dir-li a 'ls' que ens mostri tots aquells arxius que de segon caràcter tenen una 'a' sent tres el nombre mínim de caràcters. Potser amb 'ls' no li vegeu molta utilitat a aquests comodins, però imagineu-vos amb cp, rm i mv per exemple. Podeu fer tantes combinacions com s'us acudeixin, així que a provar vosaltres mateixos.

Consultant informació - man

Moltes vegades necessitem conèixer més sobre una ordre determinada o de l'ús del propi intèrpret i ens agradaria tenir un manual, en el cas de GNU/Linux disposem d'un manual en línia molt útil denominat 'man'. La seva sintaxi bàsica és la següent:

```
man [secció] ordre
```

Per exemple per saber tot de 'cp' no hi ha més que consultar 'man cp' ens mourem amb els cursors dalt o baix i per sortir premem la lletra 'q'. La barra espaiadora passa pàgina a pàgina i podem usar Re Pag i Av Pag del nostre teclat per moure'ns.

El manual en línia està dividit en seccions, concretament de la 1 a la 9, cadascuna referides a una cosa diferent. Per exemple, la secció 2 és la de programació de C. Per defecte no posarem secció i trobarem el que busquem ja que busca en totes, però en ocasions (com per exemple quan es programa) hi ha funcions que es diuen igual que alguna ordre de GNU/Linux i per defecte ens donarà l'ordre en comptes de la funció. En aquest cas, per exemple, li especificarem que volem la secció 2. Per veure per vosaltres mateixos com s'usa més a fons 'man' proveu amb 'man man'.

Altra font útil d'ajuda en línia és sens dubte l'ordre 'info', que té un maneig per visualitzar la informació igual que 'man'.

Històric - history

Per a no haver d'escriure una vegada i una altra les mateixes ordres, l'intèrpret d'ordres manté un històric de les ordres que introduïm. Per mostrar-les useu les tecles de cursor dalt i baix.

Per veure tot l'historial de cop tenim l'ordre 'history', que ens mostrarà numerades totes i cadascuna de les ordres que li hem donat a l'intèrpret d'ordres. Això és especialment útil per executar directament una ordre de l'historial usant '!' seguit del nombre d'ordre vist amb 'history'. Per exemple, per a l'ordre 5 de l'historial executeu '!5'.

En el cas de bash l'historial és guardat en el nostre propi directori de treball, en el fitxer '.bash_history', en el qual podem visualitzar i manipular el seu contingut al nostre gust. (Recordeu que és un fitxer ocult).

Tancament - halt

Ja per finalitzar només indicar-vos que, si esteu en manera consola, per tancar la sessió cal teclejar 'exit', 'logout', o Ctrl+D, per parar la màquina 'halt', i per reiniciar-la prémer 'ctrl+alt+sup' o escriure 'reboot'.

Entrada i sortida estàndard - cat

Normalment les ordres usen el que és conegut com entrada i sortida estàndard (stdin i stdout respectivament, ho pots trobar en /dev/stdin i /dev/stdout) que no són ni més ni menys que el teclat i la pantalla.

Per veure això usarem l'ordre:

```
cat [opcions] [llistadefitxers]
```

'cat' en si és un concatenador de fitxers que imprimeix la sortida per la sortida estàndard, les opcions no les veurem ja que no són molt interessants. Ens servirà per visualitzar els fitxers i si escrivim 'cat' sense paràmetres llegirà de l'entrada estàndard i escriurà en la sortida estàndard.

Redirecció d'entrada i sortida - >

L'intèrpret d'ordres ens permet redireccionar la sortida estàndard a un fitxer usant el símbol '>'. Per veure com funciona què millor que un exemple:

```
ramon@portatil:/directori$ ls
dir1 dir2
ramon@portatil:/directori$ ls >~/kk
ramon@portatil:/directori$ cat ~/kk
dir1
```

```
dir2
ramon@portatil:/directori$
```

La redirecció d'entrada és similar, només que s'usa el caràcter '<' Vegem un exemple:

```
ramon@portatil:/directori$ cat ~/kk
dir1
dir2
ramon@portatil:/directori$ cat < ~/kk
dir1
dir2
ramon@portatil:~/directori$
```

En aquest exemple veiem que el resultat és el mateix, òbviament.

Haig de destacar que la redirecció de sortida és una redirecció destructiva, amb això vull dir que si no existeix l'arxiu el crea i si existeix el sobreescrui, i això potser no ho desitgem.

Redirecció de sortida no destructiva - >>

Imaginem que volem fer una llista de la compra, llavors escriuríem alguna cosa així:

```
ramon@portatil:/directori$ cat > llistadelacompra
xampu
gominoles
cervesa
jb
(Aquí vaig prémer Ctrl+D)
ramon@portatil:/directori$
```

I se'ns obliden les aspirines:

```
ramon@portatil:/directori$ cat > llistadelacompra
aspirines
ramon@portatil:/directori$ cat llistadelacompra
aspirines
ramon@portatil:/directori$
```

Doncs ara ho acabem d'espantllar, ja que la llista la hi vam donar a Manolito i només ens va portar aspirines. Per evitar alguna cosa tan absurda i surrealista com això tenim la redirecció de sortida no destructiva, per a això usem '>>', vegem l'exemple:

```
ramon@portatil:/directori$ cat > llistadelacompra
xampu
gominoles
cervesa
jb
ramon@portatil:/directori$ cat >> llistadelacompra
aspirines
ramon@portatil:/directori$ cat llistadelacompra
xampu
gominoles
cervesa
jb
aspirines
ramon@portatil:/directori$
```

Amb això ja ens estalviem el problema. Suposo que ja haureu deduït avantatges de la redirecció a part de fer la llista de la compra, no? (tan simples com guardar el llistat de 'ls' en un fitxer i tan complicat com estalviar-nos teclejar les ordres que ens demanen moltes dades).

Pipes (canonades) - |

Per al següent exemple veurem altra ordre que el que fa és ordenar el que li entra i retornar-lo ordenat.

```
sort [opcions] [l·listadefitxers]
```

No veurem les opcions de moment (sempre pots fer 'man sort' si t'interessa). Vegem abans de totes maneres com es comporta sort sense opcions:

```
ramon@portatil:/directori$ sort l·listadelacompra
aspirines
cervesa
gominoles
jb
xampu
ramon@portatil:/directori$
```

Observem que ens retorna tot ordenat.

Seria molt interessant poder unir la sortida d'un programa amb l'entrada d'un altre i construir una cadena d'ordres. Imagineu-vos que en el cas anterior si a Manolito no li ordenem la llista es perd (no es molestin els Manolos, però algun nom hauria de tenir la víctima), el millor seria fer la llista amb 'cat', unir la sortida amb 'sort' i redireccionar al fitxer on volem guardar-lo, i per a això usem el caràcter '|' (la del AltGr +1). El que fem amb aquest símbol és crear una pipe, que és com he explicat unir la sortida d'una ordre amb l'entrada d'una altra. Millor un exemple:

```
ramon@portatil:/directori$ cat < l·lista_de_la_compra | sort > novallista
ramon@portatil:/directori$ cat novallista
aspirines
cervesa
gominoles
jb
xampu
ramon@portatil:/directori$
```

Observem que redireccionem l'entrada de 'cat' amb la llista ja creada i construïm una pipe per unir la sortida de 'cat' amb l'entrada de 'sort' i acabem redireccionant la sortida a un nou fitxer. El que aconseguim és una llista ordenada, que si bé podíem haver fet el mateix amb 'sort < lista_compra > novallista' havia de fer-lo de l'altra forma per explicar-lo.

Els permisos dels fitxers - -rwxrwxrwx

Com GNU/Linux és un sistema multiusuari, hem de protegir els fitxers de la manipulació per part d'uns altres. GNU/Linux ens proporciona per a això els permisos de fitxers.

```
ramon@portatil:~/temporal/ls -l
total 12
-rw-r--r--  1 ramon ramon 82 2005-04-07 00:16 tmp
-rw-r--r--  1 ramon ramon 82 2005-04-07 00:16 tmp2
-rw-r--r--  1 ramon ramon 82 2005-04-07 00:16 tmp3
ramon@portatil:~/temporal/
```

Els permisos de fitxers els podem establir en 3 nivells, permisos del propietari (usuari o user), permisos per al grup (group) i permís per a la resta (others).

Cada fitxer és de l'usuari que ho crea o bé els obté perquè algú li ha canviat la propietat (per exemple, root pot fer això). Només el propietari del fitxer i el superusuari (root) poden canviar els permisos dels fitxers.

Cada usuari pot pertànyer a un o a diversos grups d'usuaris, i és als usuaris que estan dintre del mateix grup que el propietari del fitxer a qui afecten els permisos de grup. I evidentment els permisos per a la resta afecten a tots els altres.

I alhora de tot això hi ha 3 tipus de permisos: permisos de lectura, permisos d'escriptura i permís d'execució.

Interpretant els permisos

Bé, com ja hem vist amb l'opció '-l' de 'ls' podem observar els permisos que té cada fitxer assignat, sent una cadena de 10 caràcters (per exemple -rw-rw-r--). El primer caràcter no ho explicarem.

Amb això ja ens queden 9, que ho dividirem en parts de 3: el primer per a la lectura, el segon per a l'escriptura i el tercer per a l'execució (si sou espavilats ja us haureu adonat que enGNU/Linux no s'executen els fitxers per tenir un tipus d'expressió, si no per tenir o no aquest permís). I l'ordre és el mateix: primer els 3 d'usuari, els 3 de grup i els 3 dels altres. Un '-' indica que aquest permís no està donat.

Perquè quedi clar vegem uns exemples:

-rw-rw-r-- L'usuari pot llegir i escriure, el grup pot llegir i escriure i la resta només llegir.

```
----- Ningú pot fer res.
-rwxrwxrwx Tots poden fer tot.
-rwx----- L'usuari pot fer tot.
---x--x--x El fitxer solament pot executar-se per tots.
-rwxr----- L'usuari pot fer-lo tot i el grup només llegir
```

Cal indicar que els permisos dels fitxers depenen dels permisos en si del directori que els conté, i que de res serveix tenir '-rwxrwxrwx' en un fitxer si el directori només té permisos '-r-----', amb la qual cosa només podríem llegir-lo i res més. Fins i tot si el directori no tingués permís de lectura per a ningú, no podríem ni tan sols llistar el contingut del directori, per veure això amb un usuari normal intenteu feu el següent:

```
ramon@portatil:$ ls /root
ls: /root: Permís denegat
ramon@portatil:$
```

Segons la configuració del vostre sistema pot ser que us deixi, si és així segurament no tindreu una bona seguretat en el vostre sistema.

Com curiositat a això, per exemple, podeu crear una bústia on tots poden entrar els fitxers que vulguin però només el teu pots veure'l i manipular-lo, seria alguna cosa així com '-rwx-w--w-'.

Canviant permisos - chmod

L'ordre que ens permet canviar els permisos als fitxers és:

chmod [opcions] manera fitxer

Com opció únicament comentarem '-R' que fa que el canvi de permisos sigui recursiu, per exemple, per donar els permisos a tots els fitxers d'un directori i els seus descendents.

Hi ha dues formes per a 'manera' a) amb un número b) amb lletres. Per a això descriurem la manera com {a,u,g,o}{+,-}{rwx} on:

- a Indica que s'aplicarà a tots.(all)
- u Indica que s'aplicarà a l'usuari.(user)
- g Indica que s'aplicarà al grup.(group)
- o Indica que s'aplicarà a uns altres.(other)
- + Indica que s'afegeix el permís.
- Indica que es lleva el permís.
- r Indica permís de lectura.
- w Indica permís d'escriptura.
- x Indica permís d'execució.

Així que primer indiquem a qui aplicarem el permís, i després quin permís posarem o llevarem. Veiem exemples de la manera:

a+r Permisos de lectura per a tots.
+r Igual que abans, si no s'indica se suposa 'a'.
og-x Lleva permís d'execució a tots menys a l'usuari.
o+rwX Dóna tots els permisos a l'usuari.
o-rwx Lleva els permisos als altres.

Amb tot això només queda fer un exemple de l'execució de 'chmod':

```

ramon@portatil:~/temporal/ls -l
total 12
-rw-r--r--  1 ramon ramon 82 2005-04-07 00:16 tmp
-rw-r--r--  1 ramon ramon 82 2005-04-07 00:16 tmp2
-rw-r--r--  1 ramon ramon 82 2005-04-07 00:16 tmp3
ramon@portatil:~/temporal/chmod a+rwx tmp
ramon@portatil:~/temporal/ls -l
total 12
-rwxrwxrwx  1 ramon ramon 82 2005-04-07 00:16 tmp
-rw-r--r--  1 ramon ramon 82 2005-04-07 00:16 tmp2
-rw-r--r--  1 ramon ramon 82 2005-04-07 00:16 tmp3
ramon@portatil:~/temporal/

```

En el cas dels números hem de pensar que cada grup de rwx es comporten com els nombres 421. Així si jo vull que el propietari tingui els permisos rwx assignaré el nombre 7 ($4+2+1=7$).

Si vull que el grup tingui només lectura r-- assignaré el nombre 4 ($4+0+0=4$). L'arxiu tmp2 de l'exemple anterior té els següents permisos '-rw-r--r--'. És a dir:

rw- per al propietari $4+2+0=6$
 r-- per al grup $4+0+0=4$
 r-- per als altres $4+0+0=4$

Si vull assignar aquests permisos a l'arxiu tmp ho faré de la següent manera:

```

ramon@portatil:~/temporal/chmod 644 tmp
ramon@portatil:~/temporal/ls -l
total 12
-rw-r--r--  1 ramon ramon 82 2005-04-07 00:16 tmp
-rw-r--r--  1 ramon ramon 82 2005-04-07 00:16 tmp2
-rw-r--r--  1 ramon ramon 82 2005-04-07 00:16 tmp3
ramon@portatil:~/temporal/

```

Canviant propietari i/o grup - chown

L'ordre chown canvia el propietari i/o grup de cada fitxer a propietari i/o grup.

chown [opció] propietari[:[grup]] fitxer

Exemple:

```

root@portatil:/home/ramon/temporal # ls -l
total 12
-rw-r--r--  1 ramon ramon 82 2005-04-07 00:16 tmp
-rw-r--r--  1 ramon ramon 82 2005-04-07 00:16 tmp2
-rw-r--r--  1 ramon ramon 82 2005-04-07 00:16 tmp3
root@portatil:/home/ramon/temporal # chown root:root
/home/ramon/temporal/tmp
root@portatil:/home/ramon/temporal # ls -l
total 12
-rw-r--r--  1 root  root  82 2005-04-07 00:16 tmp
-rw-r--r--  1 ramon ramon 82 2005-04-07 00:16 tmp2
-rw-r--r--  1 ramon ramon 82 2005-04-07 00:16 tmp3
root@portatil:/home/ramon/temporal #

```

Gestió d'usuaris

Encara que tu siguis l'únic usuari dintre del teu sistema, has de tenir un compte de root. No obstant això, no és gens aconsellable usar sempre root com si fóssim un usuari de Windows, és aconsellable tenir també un compte d'usuari normal. Si optes per usar sempre root per comoditat o perquè no vols adaptar-te a un ús més segur del sistema... tu mateix, però t'arrisques a carregar-te el sistema fàcilment, ja que pots fer "tot" i moltes vegades el sistema ni et demanarà una confirmació, i davant un atac donaràs a l'atacant més facilitat ja que, si aconseguix que s'executi codi en la teva màquina, aquest codi té via lliure al que vulgui fer.

El millor és usar root simplement quan t'és imprescindible (com canviar una configuració) i després torna al teu usuari normal, això ja ho vam veure amb el comando 'su'. Fins i tot si entres com root en entorn gràfic, quan acabis surt i entra com un usuari normal.

Conceptes de gestió d'usuaris.

El sistema guarda unes dades relatives sobre cada usuari, i aquesta informació es troba en el fitxer /etc/passwd. Cada línia conté informació sobre un usuari en el següent format:

nom:clauencriptada:UID:GID:GECOS:directoriinicial:intèrpret

El primer camp "nom" és el nom de l'usuari, seguidament està la clau encryptada encara que el que tots trobareu segurament és una 'x', i abans de res veurem perquè. Els ':' són simplement un separador entre els camps.

Després ens trobem amb el UID i GID, el primer és l'identificador d'usuari per al sistema i el segon l'identificador de grup primari al que l'usuari pertany, ja que un usuari pot pertànyer a molts grups.

GECOS és un camp opcional per a guardar informació. Normalment conté el nom complet de l'usuari. GECOS significa General Electric Comprehensive Operating System. En aquest camp podem posar el que vulguem ja que el sistema no ho usa per a cap gestió. Es pot ometre sense posar entre els ':'.

El directori inicial és on començarà l'usuari una cop 'loguejat', i l'intèrpret indica quin interpreti d'ordres usarà (bash, sh, tcsh...). Si en el camp intèrpret trobem '/sbin/nologin' és un usuari que no pot loguejar-se, això té sentit ja que el propi sistema té usuaris amb privilegis per donar serveis al sistema com poden ser 'bin', 'daemon', 'adm'...

L'usuari root sempre té com UID i com GID un '0', i qualsevol usuari que ho tingui tindrà els mateixos privilegis que ell. Cal recordar que un usuari només pot tenir un UID, però diversos usuaris poden tenir el mateix UID, cosa totalment desaconsellable perquè pot provocar forats de seguretat importants. Si veus un usuari amb UID=0 que no sigui root, el més segur és que hagi estat atacat per algun cracker.

Normalment el sistema guarda els UID baixos (per exemple per sota de 500 o per sota de 1000) per als usuaris del propi sistema, els quals necessita per als serveis que ofereix (per exemple, si tens un servidor de ftp el normal és tenir un usuari en el sistema que es digui 'ftp'), i els UID alts s'utilitzen per als usuaris normals.

Si on hauria d'aparèixer la clau apareix ':::', això indica que no hi ha clau i per descomptat no és gens bo. Si apareix una 'x' és que les claus o la clau d'aquest usuari estan gestionades en l'arxiu /etc/shadow ja que la majoria dels sistemes usen "claus en ombra". Això és degut al fet que /etc/passwd ha de ser visible a tots, perquè si no certes ordres com pot ser 'ls' deixarien de funcionar, i encara que la clau està encryptada no és bo que es trobi a la vista per l'existència de programes de força bruta que es dediquen a descriptar-les, i qualsevol usuari amb accés al nostre sistema tindria les claus usant aquests programes. En comptes d'això s'usa el fitxer /etc/shadow, que només és visible per root.

Com no és aconsellable per res tocar aquests fitxers manualment, no continuarem amb la descripció del fitxer shadow (qui vulgui veure'l que usi 'man 5 shadow'), en lloc de tocar els fitxers usarem les ordres que ens permeten manegar tot això.

Afegir usuaris - adduser

adduser usuari

```
root@portatil:/ # adduser pepe
Afegint usuari pepe...
```

```

Adding new group `pepe' (1001).
Adding new user `pepe' (1001) with group `pepe'.
Creant el directori home /home/pepe.
Copiant arxius des de /etc/skel
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Canviant la informació d'usuari per a pepe
Introdueixi el nou valor, o pressioni ENTER per al predeterminat
    Nom complet []:
    Nombre d'habitació []:
    Telèfon del treball []:
    Telèfon de casa []:
    Un altre []:
És correcta la informació? [y/N] y
root@portatil:/ #

```

Modificant a l'usuari - usermod

```

usermod [-c comentari] [-d home] [-i data] [-f dies] [-g grup] [-G llista de grups] [-m]
[-n] [-p passwd] [-s shell] [-o uid [-o] ] usuari [-L | -O] usuari

```

Jo puc canviar-me la meva contrasenya? - passwd

Imagina't que no estàs en un ordinador teu, sinó que estàs en un sistema en el qual tens un compte d'usuari, seria una molèstia no poder canviar la teva pròpia contrasenya i haver de demanar-ho a l'administrador (a part de no agradar-te que l'administrador sàpiga la teu clau). Doncs per a això la següent ordre:

```
passwd [opcions] [usuari]
```

```

root@portatil:/ # su pepe
pepe@portatil:/$ passwd
Changing password for pepe
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
pepe@portatil:/$

```

Com veieu ens demana el vell password i 2 vegades el nou (per no equivocar-nos). Pot ser que no ens deixi posar una clau senzilla, si vols fer-lo usa `passwd' amb l'opció `-o'. L'únic que pot canviar una contrasenya que no sigui seva és `root'.

```

root@portatil:/ # passwd pepe
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@portatil:/ #

```

Eliminant usuaris - userdel

```
userdel [-r] usuari
```

Simplement elimina a l'usuari que li indiquem (mireu què senzill aquesta vegada, ja se sap que destruir és més fàcil que crear), i com única opció tenim `-r', que elimina el seu directori home amb tots els seus fitxers.

```

root@portatil:/ # userdel -r pepe
root@portatil:/ # ls /home/pepe
ls: /home/pepe: No existeix el fitxer o el directori
root@portatil:/ #

```

Gestió de grups

Cada usuari pertany a un o més grups (encara que només un és el principal), i per això cada fitxer té un usuari i un "grup propietari" (el principal de l'usuari), per dir-lo així.

Igual que hi ha usuaris del sistema, també existeixen grups del sistema tals com grup root, grup bin... i en general, excepte rares excepcions, els usuaris mai han de pertànyer a cap d'aquests grups.

La informació dels grups es guarda en `/etc/group`, i igual que els usuaris també poden tenir i és normal tenir les claus sota shadow, concretament el fitxer de shadow per als grups és `/etc/gshadow`, i igual que en els usuaris només és accessible pel root. De la mateixa manera que amb els usuaris només explicaré el contingut del primer.

El format de `/etc/group` és el següent:

nomgrup:clau:GID:llista d'usuaris membres

El primer i el segon camp no necessiten explicació ja que és igual que en el d'usuaris (recordeu que si hi ha una x és que la clau està sota shadow). GID és l'identificador de grup i sol coincidir que el UID de l'usuari amb el mateix nom, però no té perquè ser així. I finalment tenim la llista de tots els usuaris que pertanyen a aquest grup.

No és normal que els grups d'usuaris tinguin clau, si coneixeu com és `/etc/gshadow` veureu que el camp de la clau està buit (de fet mireu-lo i veureu que gairebé tots els camps estan en blanc menys el primer i l'últim), no se solen usar mai les claus en els grups (la veritat és que seria molt aclaparador que et demanés la clau de grup també).

Afegint grups - addgroup

`addgroup grup`

Crearem un grup anomenat, per exemple, `administratius`:

```
root@portatil:/home/ramon/temporal # addgroup administratius
Adding group `administratius' (1001)...
Fet.
root@portatil:/home/ramon/temporal #
```

Eliminat els grups - groupdel

Per a eliminar els nous grups que hem creat usem el senzill comando:

`groupdel grup`

Amb aquesta simple ordre que no té ni opcions eliminarem els grups:

```
root@portatil:/home/ramon/temporal # groupdel administratius
root@portatil:/home/ramon/temporal #
```

Per acabar la gestió d'usuaris veurem dos ordres que ens poden ser útils:

`users`

`groups usuari`

El primer indica els usuaris que estan accedint al sistema, i el segon indica tots els grups als quals pertany un usuari. Exemple:

```
root@portatil:/home/ramon/temporal # users
ramon ramon
root@portatil:/home/ramon/temporal # groups ramon
ramon : ramon adm dialout cdrom floppy àudio dip video plugdev lpadmin
scanner
root@portatil:/home/ramon/temporal #
```

Altres ordres

du [opció]... [fitxer]...

Mostra un resum de l'ús de disc per a cada fitxer, recursivament per a directoris.

`-h`, mostra el resultat en format "humà" (ex., 1K 234M 2G)

`-s`, mostra solament un total per a cada argument

```
ramon@portatil:~$ du -hs *
48K   bookmarks.html
70M   descàrregues
32K   Desktop
47M   docs
148K  documents
```

```

9,9M   fotos
9,9M   proferamon.com
16K    ~/temporal
33M    uoc
ramon@portatil:~$

```

df [opció]... [fitxer]...

Mostra informació sobre el sistema de fitxers en el qual resideix cada fitxer, o per omissió sobre tots els sistemes de fitxers.

```

ramon@portatil:~$ df -h
S. fitxers      Grandària Usada  Disp Ús% Muntat en
/dev/hda7       11G  2,9G  7,0G  29% /
tmpfs           126M    0  126M   0% /dev/shm
/dev/hda1       10G  3,8G  6,3G  38% /mitjana/windows/c
/dev/hda5       7,0G  3,0G  4,1G  42% /mitjana/windows/d
/dev            11G  2,9G  7,0G  29% /.dev
none            5,0M  2,8M  2,3M  56% /dev
ramon@portatil:~$

```

find

Busca arxius

opció -iname patró

```

ramon@portatil:~$ find -iname *.html
./mozilla/firefox/7ovdoc0g.default/bookmarks.html
./bookmarks.html
./documents/bookmarks.html
./docs/bookmarks.html
ramon@portatil:~$

```

Control de tasques

El control de tasques és una utilitat inclosa en la majoria dels intèrprets d'ordres, i permeten el control de multitud de tasques o ordres alhora en un sol terminal.

Els processos poden estar en Primer Pla o en Segon Pla. En primer pla solament pot haver un procés alhora i aquest és precisament el que veiem, en altres paraules el que interactua amb nosaltres donant-nos una sortida o demanant dades, el qual rep les ordres del teclat. En canvi el procés en segon pla no rep senyal des del teclat i s'executa en silenci.

Si un programa triga molt a acabar i no mostra res interessant per pantalla el més lògic seria executar-lo en segon pla, com pogués ser la compressió d'un arxiu enorme. Si ho fem així podrem seguir usant l'ordinador per fer qualsevol altra cosa.

Un procés pot ser suspès (dormit), el que indica que temporalment estarà desocupat i més tard podríem reprendre'l per a quan ho necessitem.

O bé quan ja no ens interressi un procés podrem interrompre'l i acabar amb ell.

Veurem una ordre que per a mi és gairebé totalment inútil però que ens servirà molt bé per a les explicacions, és l'ordre:

yes [string]

Aquesta ordre simplement repetirà el que li indiquem en 'string' indefinidament, i en defecte d'això ens donarà una llista de 'y' si no li indiquem res.

Per provar-lo teclegem yes i a veure què passa:

```

ramon@portatil:~$ yes
y
y
y
....

```

Avorrits de veure 'y' sense parar el que farem és matar-lo per avorrit ;) per a això premerem 'Ctrl + c' i ja està.

Com al fer les proves no ens interessa estar mirant un munt de `y` el que farem és redirigir la sortida de l'ordre `yes` cap a `/dev/null` que és alguna cosa així com un forat negre, és a dir, tot el que allí enviem desapareix (més d'un voldria un `/dev/null` en la vida real).

Enviant al segon pla - &

Ara el que volem fer és arrencar l'ordre `yes` amb la redirecció a `/dev/null` i arrencar-lo en segon pla aconseguint una bonica despesa de CPU :). Per a enviar-lo a segon pla hem d'afegir el símbol `&` al final de la línia, com veiem en l'exemple.

```
ramon@portatil:$ yes > /dev/null &
[1] 8042
ramon@portatil:$
```

Aquest és el resultat, tornem a tenir el prompt per a nosaltres i dos curiosos nombres. Aquests dos nombres són les referències a la tasca, el `[1]` representa el nombre de tasca i el 8042 és l'identificador de procés o PID, que és ni més ni menys el nombre que li assigna el sistema al procés.

Però on s'ha ficat? - jobs

I ara què fem per veure què ha passat amb la nostra ordre? Doncs simple, usarem una de les ordres internes de l'interpret d'ordres, aquesta és `jobs` que dona un llistat de les tasques que tenim en execució i el seu estat:

```
ramon@portatil:$ jobs
[1]+  Running                  yes > /dev/null &
ramon@portatil:$
```

Com veieu ens indica el nombre de tasca, l'estat (en aquest cas corrent o millor dit executant-se) i la forma que ho cridem.

Més formes de matar - kill

Seguint amb les nostres ganes d'eliminar a `yes` del nostre sistema (ara no ens val `Ctrl + c`) tenim 2 formes de fer-lo amb una mateixa ordre, aquesta és `kill` que així sense més envia un senyal de fi de procés al procés que li indiquem.

Per fer-lo tenim dues formes de referenciar-la, la més còmoda seria amb el nombre de tasca i per a això usem el símbol `%` de la següent manera:

```
ramon@portatil:$ kill %1
ramon@portatil:$
```

Així que recordeu, per usar el nombre de tasca cal usar un `%` davant i per a usar el PID simplement posarem el PID. Així que `kill %1` i `kill 8042` serien equivalents.

I per a cerciorar-nos de la seva defunció consultem al nostre `jobs`:

```
ramon@portatil:$ jobs
[1]+  Acabat                   yes > /dev/null
ramon@portatil:$
```

Cal destacar que `jobs` guarda la informació del procés mort per a mostrar-la 1 vegada, ja que si executem de nou `jobs` ja res se sabrà de la tasca morta:

```
ramon@portatil:$ jobs
ramon@portatil:$
```

També disposem de l'ordre `killall` per matar els processos. La usem seguida del nom del procés que volem matar. Vegem un exemple:

```
ramon@portatil:$ jobs
[2]+  Stopped                  yes > /dev/null
[3]-  Running                  yes > /dev/null &
ramon@portatil:$ killall yes
```



```

ramon@portatil:$ jobs
[2]+  Stopped                  yes > /dev/null
[3]-  Acabat                   yes > /dev/null
ramon@portatil:$

```

Aturar i continuar – Ctrl + z

Hi ha altra forma d'enviar una tasca a segon pla i és executar-la normalment i fer-la suspendre's o dormir-se, per a això usarem la combinació de tecles 'Ctrl+z':

```

ramon@portatil:$ yes > /dev/null
(Aquí vaig prémer Ctrl + z)
[1]+  Stopped                  yes > /dev/null
ramon@portatil:$ jobs
[1]+  Stopped                  yes > /dev/null
ramon@portatil:$

```

Per tornar-la a reprendre en primer pla usarem l'ordre interna 'fg' (foreground) continuant la tasca just on es va deixar (cal recalcar que quan una tasca està parada no usa temps de CPU).

```

ramon@portatil:$ fg
yes > /dev/null
[1]+  Stopped                  yes > /dev/null
ramon@portatil:$

```

Veiem com l'hem reprès i l'hem tornat a parar, però en aquest cas ja sí que l'enviarem al segon pla amb l'ordre interna 'bg' (background), que continuarà amb l'execució de l'ordre en segon pla.

```

ramon@portatil:$ bg
[1]+ yes > /dev/null &
ramon@portatil:$

```

Recapitulant

Les tasques poden estar en primer i segon pla, i aquestes poden estar bé executant-se, desocupades o acabades (mortes). Les tasques en segon pla no poden parar-se ni eliminar-les mitjançant 'Ctrl + c' sinó que abans cal passar-les a primer pla per poder realitzar això.

Una tasca parada no consumeix temps de CPU, però segueix mantenint la memòria que estigués usant, simplement espera que se li doni l'oportunitat de continuar amb la seva tasca.

Encara que una tasca en segon pla no pot rebre res des del teclat, sí que pot enviar dades a la pantalla i pot resultar molt molest.

Per exemple, si se't hagués ocorregut executar 'yes &' no podrídes detenir-lo amb 'Ctrl + c' o amb 'Ctrl + z' ja que està en segon pla, però ens està aclaparant amb una tira interminable de 'y' que no ens deixa veure el nostre prompt, amb el que per remeiar això hem de teclejar a cegues 'fg' per portar-nos la tasca al primer pla i aquesta vegada sí la podem parar per aquest mètode (també podríem haver-la eliminat amb un 'kill' a cegues amb el seu nombre de tasca).

Més sobre 'fg' i 'bg'

Aquestes dues ordres actuen sobre l'últim procés parat o creat, indicat per un '+' quan executem 'jobs' (mireu els exemples anteriors). Si tinguéssim diverses tasques per indicar-li a 'fg' i 'bg' sobre quin actuar utilitzaríem, com amb 'kill', el símbol '%' més el seu nombre de tasca.

```

ramon@portatil:$ yes > /dev/null &
[2] 8142
ramon@portatil:$ yes > /dev/null &
[3] 8143
ramon@portatil:$ yes > /dev/null &

```

```

[4] 8145
ramon@portatil:$ jobs
[1]  Running          yes > /dev/null &
[2]  Running          yes > /dev/null &
[3]- Running         yes > /dev/null &
[4]+ Running         yes > /dev/null &
ramon@portatil:$ fg %2
yes > /dev/null
(Aquí vaig prémer Ctrl + z)
[2]+ Stopped         yes > /dev/null
ramon@portatil:$ jobs
[1]  Running          yes > /dev/null &
[2]+ Stopped         yes > /dev/null
[3]  Running          yes > /dev/null &
[4]- Running         yes > /dev/null &
ramon@portatil:$ kill %1
ramon@portatil:$ kill %4
[1]  Acabat          yes > /dev/null
ramon@portatil:$
ramon@portatil:$ jobs
[2]+ Stopped         yes > /dev/null
[3]  Running          yes > /dev/null &
[4]- Acabat          yes > /dev/null
ramon@portatil:$

```

Ordres associades - ps, top, nice, renice, pidof, nohup

ps mostra els processos del sistema. Amb les opcions a, x i l ens mostra tots els processos d'una forma ampliada.

```

ramon@portatil:$ ps axl
F  UID  PID  PPID  PRI  NI   VSZ   RSS  WCHAN  STAT  TTY          TIME
COMMAND
5  105  7837    1  26  10   1740   740  -      SNs  ?           0:00
/sbin/syslo
0  1000  7938    1  16   0  33628  15272  -      S1   ?           0:16
gedit
0  1000  8006    1  15   0  39212  14372  -      S1   ?           0:04
gnome-termi
0  1000  8007  8006  18   0   2224   656  -      S    ?           0:00
gnome-pty-h
0  1000  8008  8006  16   0   4800   1836  wait   Ss   pts/0       0:00 bash
0  1000  8142  8008  25   0   3348   428  finish T    pts/0       0:11 yes
0  1000  8143  8008  25   0   3348   428  -      R    pts/0       1:58 yes
0  1000  8182  8008  18   0   4148   732  -      R+   pts/0       0:00 ps
axl
ramon@portatil:$

```

top llista els processos en execució de forma interactiva

```

ramon@portatil:$ top
Tasks:  86 total,   2 running,  83 sleeping,   1 stopped,   0 zombie
Cpu(s): 99.0% us,  1.0% sy,  0.0% ni,  0.0% id,  0.0% wa,  0.0% hi,  0.0%
si
Mem:    256300k total,  242352k used,  13948k free,  4884k buffers
Swap:   522072k total,  11648k used,  510424k free,  61944k cached

   PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
  8143 ramon    25   0  3348  428  364  R  95.1   0.2   5:51.42  yes
  6244 root     15   0 56152  23m  8780  S   2.3   9.2   0:43.86  Xorg
  8006 ramon    15   0 39300  14m  9464  S   2.3   5.7   0:05.24  gnome-
terminal
  7502 ramon    15   0 20812   9m  7868  S   0.3   4.0   0:00.72  mixerapplet2
     1 root     16   0  1552   488  444  S   0.0   0.2   0:00.56  init
     2 root     34  19     0     0     0  S   0.0   0.0   0:00.00  ksoftirqd/0

```

L'ordre nice estableix la prioritats que ha de tenir un procés que encara no està en execució. Si se li passa com paràmetre una ordre juntament amb la prioritats desitjada, ajusta la prioritats de l'ordre a l'especificada. Exemple: (el color vermell només és per resaltar l'exemple)

```
root@portatil:/home/ramon # nice -10 find / > /dev/null &
[1] 8531
root@portatil:/home/ramon # top

top - 03:00:47 up 51 min,  2 users,  lloeu average: 0.73, 0.72, 0.67
Tasks:  87 total,   1 running,  86 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.3% us,   8.3% sy,   0.7% ni,   0.0% id,  89.7% wa,   1.0% hi,   0.0%
si
Mem:    256300k total,   253152k used,    3148k free,    29216k buffers
Swap:   522072k total,   11648k used,    510424k free,   44212k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 8531 root        26   10  3384   556  400  D   8.0   0.2   0:02.56 find
  108 root         5  -10    0     0     0  S   0.3   0.0   0:00.94 kblockd/0
 6244 root        15    0 56920  23m 9548  S   0.3   9.5   1:23.54 Xorg
```

renice canvia la prioritats d'un procés que s'està executant

```
root@portatil:/home/ramon # nice -10 find / > /dev/null &
[1] 8565
root@portatil:/home/ramon # renice -0 8565
8565: prioritats antiga 10, nova prioritats 0
root@portatil:/home/ramon # top

top - 03:03:30 up 54 min,  2 users,  lloeu average: 1.05, 0.89, 0.74
Tasks:  87 total,   2 running,  85 sleeping,   0 stopped,   0 zombie
Cpu(s):  4.0% us,   9.0% sy,   0.0% ni,   0.0% id,  84.7% wa,   2.3% hi,   0.0%
si
Mem:    256300k total,   253256k used,    3044k free,    27932k buffers
Swap:   522072k total,   11648k used,    510424k free,   44076k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 8565 root        18    0  3384   556  400  R   7.3   0.2   0:03.60 find
 8006 ramon       16    0 39668  14m 9516  S   2.7   5.9   0:35.85 gnome-
terminal
```

pidof ens retorna el PID d'un procés passant-li el nom del procés

```
root@portatil:/home/ramon # find / > /dev/null &
[1] 8616
root@portatil:/home/ramon # pidof find
8616
```

nohup executa ordres immunes a que l'ordinador es pengi i als tancaments de sessió.

```
ramon@portatil:~$ nohup find / > /dev/null &
[1] 8637
ramon@portatil:~$
```

i si a continuació féssim 'exit' el procés continuaria executant-se.

Tasques programades: AT i CRON

AT

at ens serveix per programar tasques que s'han d'executar abans que la màquina s'apagui.

atd és el dimoni que s'encarrega de gestionar els treballs d'at.

at gestiona treballs per a una posterior execució.

atq llista les tasques pendents.

atrm seguit del nombre de tasca esborra la tasca pendent.
Exemples:

```
root@portatil:/home/ramon # date
mar abr 12 03:22:31 CEST 2005
root@portatil:/home/ramon # at 03:30 today
warning: commands will be executed using /bin/sh
at> poweroff
at> <EOT> (Aquest EOT surt quan prems 'Ctrl + D')
job 1 at 2005-04-12 03:30
root@portatil:/home/ramon # atq
1          2005-04-12 03:30 a root
root@portatil:/home/ramon # atrm 1
root@portatil:/home/ramon # atq
root@portatil:/home/ramon #
```

CRON

cron és el dimoni que s'encarrega de gestionar els treballs de crontab.
crontab -e edita tasques programades.
crontab -l llista les tasques programades.
crontab -r esborra tasques programades.

minuts	hora	dia	mes	mes
dia setmana	usuari	ordre		
de 0 a 59	de 0 a 23	d'1 a 30/31	d'1 a 12	d'1 a 7

Exemple:

perquè cada dia la màquina es pari a les 21:00 escriuríem alguna cosa així:

```
0 21 * * * root poweroff
```

Si volem que una tasca s'executi cada 5 hores hem d'usar la barra '/' i escriure alguna cosa semblat a això:

```
* /5 * * * usuari tasca
```

Existeixen quatre carpetes on podem posar executables o enllaços a executables perquè s'executin amb la periodicitat que nosaltres vulguem (horària, diària, setmanal o mensual). Aquestes carpetes són:

```
/etc/cron.hourly
/etc/cron.daily
/etc/cron.weekly
/etc/cron.monthly
```

```
root@portatil:/home/ramon # cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file.
# This file also has a username field, that none of the other crontabs
do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || run-parts --report /
etc/cron.daily
47 6 * * 7 root    test -x /usr/sbin/anacron || run-parts --report /
etc/cron.weekly
52 6 1 * * root    test -x /usr/sbin/anacron || run-parts --report /
etc/cron.monthly
#
```

Enllaços - ln

Perquè serveixen els enllaços?

Els enllaços serveixen per donar a un fitxer múltiples noms (no confongueu els que veniu del món de Windows amb accessos directes). Internament per al sistema els fitxers són identificats per un nombre que es diu inode, el que vol dir que el nom d'un fitxer està associat a un inode que és el que usa realment el sistema operatiu, el qual és l'únic identificador per al sistema. Amb això podríem dir que simplement els noms aquests que veiem al fer un 'ls' són enllaços a inodes. I com podreu deduir un directori no és més que una llista de inodes amb els seus respectius noms de fitxers.

Enllaços durs

Un enllaç dur consisteix a assignar un nom de fitxer a un inode, amb la qual cosa podem referenciar a un mateix fitxer amb diversos noms i un canvi en un d'ells implica un canvi en la resta ja que es tracta del mateix fitxer realment. Per a això usem l'ordre:

ln [Opcions] Origen Destinació

On l'origen és el nom del fitxer al que volem fer un enllaç dur, i per veure el nombre d'inode usarem l'opció '-i' de 'ls'. Perquè tot quedi clar un exemple:

```
ramon@portatil:/curs$ ls -i
261358 e1 261476 e2
ramon@portatil:/curs$ ln e2 e3
ramon@portatil:/curs$ ls -i
261358 e1 261476 e2 261476 e3
ramon@portatil:/curs$
```

El primer que fem és mirar tots els inodes dels fitxers continguts en el nostre directori, creem un enllaç a e2 anomenat e3 i comprovarem que efectivament tenen el mateix inode.

Suposo que ja us ho esteu preguntant: què passa si esborro un d'ells? Doncs si esborres un d'ells simplement el comptador disminueix en 1, i quan no hi ha més enllaços és esborrat realment. Vegem-lo millor amb un exemple:

```
ramon@portatil:/curs$ ls -l
total 24
-rw-r--r-- 1 ramon ramon 13475 2005-04-19 00:56 e1
-rw-r--r-- 2 ramon ramon 792 2005-04-19 00:57 e2
-rw-r--r-- 2 ramon ramon 792 2005-04-19 00:57 e3
ramon@portatil:/curs$ rm e3
ramon@portatil:/curs$ ls -l
total 20
-rw-r--r-- 1 ramon ramon 13475 2005-04-19 00:56 e1
-rw-r--r-- 1 ramon ramon 792 2005-04-19 00:57 e2
ramon@portatil:/curs$
```

Podem observar amb el primer 'ls -l' que e2 i e3 tenen ambdós un 2 ja que a l'inode al que apunten té 2 enllaços, i com ambdós apunten al mateix doncs ambdós mostren que tenen 2. I a l'esborrar un d'ells simplement s'elimina l'enllaç i disminueix el comptador. Fàcil, veritat?

Cal ressaltar que els enllaços durs tenen una limitació i és que només es poden fer enllaços d'aquest tipus en un mateix sistema de fitxers. Amb això vull dir que si tens per exemple dues particions no pots fer enllaços durs entre particions. Si intentes fer un enllaç dur entre particions reps un missatge d'error més o menys així:

```
ramon@portatil:/curs$ ln e2 /etc/e3
ln: creant l'enllaç dur `/etc/e3' a `e2': Enllaç creuat entre dispositius
no permès
ramon@portatil:/curs$
```

Enllaços simbòlics

Aquests són altres tipus d'enllaços bastant diferents i sense les limitacions anteriorment explicades. Aquest tipus d'enllaços permeten donar a un fitxer el nom d'un altre, però no enllaça el fitxer amb un inode. El que fa en aquest cas és realment apuntar al nom del fitxer amb qui enllaça, per la qual cosa tindran inodes diferents i si un s'elimina no s'elimina l'altre (això és ja més semblat al que és un accés directe de Windows, encara que no ho és per res).

Per realitzar aquest tipus d'enllaços usem l'opció '-s' de 'ln' i per a entendre'l millor com sempre un exemple:

```
ramon@portatil:/curs$ ls -li
total 20
261358 -rw-r--r--  1 ramon ramon 13475 2005-04-19 00:56 e1
261476 -rw-r--r--  1 ramon ramon   792 2005-04-19 00:57 e2
ramon@portatil:/curs$ ln -s e2 e3
ramon@portatil:/curs$ ls -li
total 20
261358 -rw-r--r--  1 ramon ramon 13475 2005-04-19 00:56 e1
261476 -rw-r--r--  1 ramon ramon   792 2005-04-19 00:57 e2
261477 lrwxrwxrwx  1 ramon ramon    2 2005-04-19 01:05 e3 -> e2
ramon@portatil:/curs$
```

Podem veure en aquest exemple com els inodes són totalment diferents (vermell), i com e3 és un enllaç a e2 d'una manera molt senzilla (verd). I per veure que l'anteriorment dit és cert eliminarem e2 i quedarà e3

```
ramon@portatil:/curs$ rm e2
ramon@portatil:/curs$ ls -li
total 16
-rw-r--r--  1 ramon ramon 13475 2005-04-19 00:56 e1
lrwxrwxrwx  1 ramon ramon    2 2005-04-19 01:05 e3 -> e2
ramon@portatil:/curs$
```

I endevineu ja què passa si fem 'cat' a e3, o fa falta un exemple? ;)

Muntant unitats - mount

Què necessitem saber i repassar?

Quant a les unitats d'emmagatzematge com són discos durs, disquetes, unitats òptiques, pen usb i altres, ja siguin locals o de forma remota (a través de la xarxa per exemple), sapiguen que no s'accedeix directament a elles pel seu dispositiu.

En els sistemes Unix no existeixen les unitats com en Windows (A:, B:, C: ...). Existeix un directori '/' a partir del qual penja tot. Per accedir a una unitat (com una partició) cal muntar-la en un directori dintre del nostre arbre de directoris. Amb això aconseguim un 'tot' uniforme i accedim a tot de la mateixa manera, ja siguin particions FAT, les pròpies de GNU/Linux, o una targeta de memòria o la memòria d'una càmera digital, estiguin dintre del teu ordinador o en la Xina. És important entendre això i el perquè es munten i es desmunten unitats, que no és més que enllaçar i desenllaçar amb l'arbre de directoris, per no escoltar coses tan absurdes com que "Linux està molt endarrerit perquè cal muntar unitats".

Per conveni, consens o costum, els dispositius són muntats en un directori dintre '/mnt', i en moltes distribucions els dispositius removibles com disquetes i dispositius òptics dintre de '/media'. Encara que realment podem muntar-los en qualsevol altre directori que no sigui '/', si bé és completament absurd muntar unitats en llocs del sistema com '/boot', '/bin', '/usr' i similars.

I finalment, un dispositiu es munta en un únic directori per a ell sol, res de muntar tot en el mateix lloc. Si s'us ocorre muntar en un directori que no estigui buit observareu que l'anterior desapareix, realment el que succeeix és que queda ocult i inaccessible fins que es desmunti el dispositiu.

Per muntar unitats tenim una ordre molt interessant i útil:

`mount [-o Opcions] [-t tipus] [dispositiu] [directori]`

Evidentment aquesta no és l'única forma d'utilitzar `mount` ja que existeixen més opcions i paràmetres, però sí la que ens serà d'utilitat per a nosaltres que estem aprenent. El primer que veurem és l'ús de `mount` sense paràmetres:

```
ramon@portatil:/curs$ mount
/dev/hda7 on / type ext3 (rw,errors=remount-ro)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
/dev/hda1 on /mnt/windows/c type vfat (rw,umask=000)
/dev/hda5 on /mnt/windows/d type vfat (rw,umask=000)
/dev on /.dev type unknown (rw,bind)
none on /dev type tmpfs (rw,size=5M,mode=0755)
usbfs on /proc/bus/usb type usbfs (rw)
ramon@portatil:/curs$
```

El que aquí se'ns mostra és el que tenim muntat actualment en el nostre sistema, de la forma:

Dispositiu on Directori type Tipus Opcions

on el Tipus indica el sistema de fitxers que usa, i Opcions ens indica les opcions amb les quals va ser muntat, més endavant veurem què signifiquen.

Si esteu atents us adonareu que existeixen Directoris als quals no hi ha assignats un dispositiu real, com per exemple "`proc on /proc type proc (rw)`":

```
ramon@portatil:/curs$ ls /dev/proc
ls: /dev/proc: No existeix el fitxer o el directori
ramon@portatil:/curs$
```

Això és degut al fet que existeixen sistemes de fitxers que no estan associats a un dispositiu real, és més, en el cas `'/proc'` d'aquest exemple ni tan sols existeix físicament, és una representació en memòria dels processos que corren pel sistema i més informació útil que ens proporciona el kernel. Però de moment donarem de costat això i ens centrarem en el que ens interessa.

El kernel de GNU/Linux és capaç de reconèixer molts sistemes de fitxers, i entre ells els que més ens interessin són els següents:

vfat	Per a particions pròpies de Windows tipus FAT ja sigui FAT16, FAT32 o FAT12 dels disquets
ntfs	Per a particions pròpies dels Windows NT/2000/XP
ext2	Particions de GNU/Linux corresponents a l'antic estàndard, però àdhuc molt en ús
ext3	Particions de GNU/Linux corresponents al nou estàndard
swap	Per a particions swap de GNU/Linux
reiserfs	Particions de tipus ReiserFs molt utilitzades en GNU/Linux
iso9660	Un sistema de fitxers local usat per discos CDROM
ufs	Un sistema de fitxers local usat per discos CDROM

Però hi ha més com poden ser `minix`, `ext`, `xiafs`, `hpfs`, `msdos`, `umdos`, `proc`, `nfs`, `smbfs`, `ncpfs`, `affs`, `ufs`, `romfs`, `sysv`, `xenix`, `coherent` ... Com veieu, una gran quantitat de possibilitats impossibles d'aconseguir amb un Windows.

I ara queda la gran prova: muntar un dispositiu, i encara que la majoria de vosaltres tindreu ja facilitat la forma de muntar-lo ho farem manualment. Per a això usarem l'opció `'-t'` que serveix per indicar el tipus de sistema a usar, jo per referenciar al cdrom ho faré com `/dev/hdXY`, encara que pots usar `/dev/cdrom`:

```
root@portatil:/home/ramon # mount -t iso9660 /dev/hdc /home/ramon/curs
mount: dispositiu de blocs /dev/hdc està protegit contra escriptura; es
munta com només lectura
```

```
root@portatil:/home/ramon #
```

Com veieu només root pot muntar unitats, i ara podem veure el contingut:

```
root@portatil:/home/ramon # cd curs
root@portatil:/home/ramon/curs # ls
dists  install  md5sum.txt  pool      README.diskdefines  ubuntu
doc    isolinux  pics        preseed   tools
root@portatil:/home/ramon/curs #
```

I ara com trec el meu CD? No em deixa! - umount

Una unitat muntada mai ha de desconnectar-se o retirar-se sense abans desmuntar-la, i això és molt important ja que GNU/Linux usa un sistema de buffers (memòria intermèdia entre el dispositiu i el processador, ja que els sistemes d'emmagatzematge són moltíssim més lents que la memòria principal) en memòria que pot no estar actualitzat pel que fa a la unitat (encara que en aquest cas sigui de només lectura), cada cert temps el contingut d'aquests buffers és sincronitzat amb el dispositiu i al desmuntar-se el que es fa és descarregar tots aquests buffers per poder retirar de forma segura el dispositiu (o apagar la màquina sense perdre dades).

Perquè això és així? Per raons d'eficiència, encara que això suposi pèrdua de facilitat (cosa que actualment no és així, ja que les distribucions més amigables estan preparades per no haver de muntar i desmuntar, tal com una Mandriva per posar un exemple).

L'ordre que necessitem és:

umount dispositiu | directori

I aquesta ordre és ben senzilla, per desmuntar-lo simplement executem 'umount' i li indiquem o bé el dispositiu que volem desmuntar o bé el directori.

```
root@portatil:/home/ramon # umount /dev/hdc
root@portatil:/home/ramon #
```

No em desmunta la unitat, què faig? - eject

Algunes vegades ens ocorre que no es desmunten les unitats, cosa que pot arribar a desesperar-nos, el primer que cal mirar és que no estigui executant-se res que usi el que tenim muntat, la major part de les vegades ho solucionem amb això (és evident que si alguna cosa està en ús no ho podem llevar). Altre problema i més gros és que un procés camini per aquí bloquejat i no alliberi el dispositiu. Aquest problema és més complicat, ja vam veure a la [sessió 5](http://www.proferamon.com/1104.html#cont) (<http://www.proferamon.com/1104.html#cont>) com gestionar i eliminar processos.

Si apareix aquest problema contem amb una ordre bastant útil que ens pot solucionar la papereta, i el seu ús bàsic és:

eject [dispositiu|directori]

No és necessari en aquest cas ser root, per defecte si no se li indica res es referirà a '/dev/cdrom' així que ho executarem per al nostre exemple:

```
root@portatil:/home/ramon # eject hdc
root@portatil:/home/ramon #
```

Com veieu és tan amable que fins i tot ens expulsa el CD, i no és necessari posar abans '/dev/'. Per cert, no sigueu il·lusos i espereu que us expulsi un disquet ;).

Instal·lació de programes

Què és això d'apt

Apt és un sistema de gestió de paquets de programari (utilitzar aquest tipus d'eines és la forma més indolora d'instal·lar paquets). És originària de Debian i està portada a diverses distribucions no descendents d'aquesta, com RedHat, Fedora, Mandriva, Suse i multitud d'elles.

El gran avantatge d'apt, igual que aplicacions similars, és que tu li dius què instal·lar i ell no només t'ho instal·la, sino que a més et resol i instal·la totes les dependències, així que

adéu a les instal·lacions difícils i benvingut a instal·lacions més senzilles, ordenades i netes que el següent, següent, següent de Windows. No és una meravella?
En qualsevol cas, apt és una interfície senzilla de 'dpkg' (o 'rpm' segons el cas).

D'on obtinc el necessari per al conjunt d'eines d'apt?

Els que usin Debian o derivades d'ella com Ubuntu, Knoppix, Gnoppix, Linex, Guadalinux i un grapat gran d'elles més, no tindran problemes perquè ja els ve de sèrie. Per als que no, ja poden fer recerques en google amb les paraules apt-get i la distribució que vols, encara que és possible que no ho trobin per a la seva.

Per als que usin alguna de RedHat, Fedora o openSuse ja els estalvio jo una mica el treball:

<http://apt.freshrpms.net/> per a RedHat i Fedora

<http://linux01.gwdg.de/apt4rpm/home.htm> per a openSuse

I la resta pot mirar <http://rpm.pbone.net/> on hi ha apt-get per a un munt d'elles.

Bé, ja està instal·lat, però no corris. Necessitem saber una mica més

Bé, en aquest punt ja hauries de tenir instal·lat apt, però encara ens queden coses abans de començar. Apt en si no és una ordre, sinó un conjunt d'eines de les quals veurem bàsicament apt-get i apt-cache, que són les que normalment cobriran totes les nostres necessitats.

Apt, per treballar amb els paquets, li dóna igual que siguin deb o rpm, per aquest motiu ha estat portat a moltes distribucions. El que apt necessita és que li diguis on té els repositoris de paquets, els repositoris són els "llocs" on es troben els paquets dels quals volem disposar, i pot ser qualsevol font: cdrom, ftp, http o d'altres.

Nosaltres de moment optarem per instal·lar i actualitzar amb apt via ftp o http, perquè aconseguirem el programari més actual ja que els repositoris solen actualitzar-se freqüentment. Per configurar això hem d'editar (com root) /etc/apt/sources.list i tindrà un aspecte com aquest:

```
#deb file:///cdrom/ sarge main
#deb cdrom:[Debian GNU/Linux testing _#Sarge_ - #Official Snapshot i386
Binary-1 (20050318)]/ unstable contrib main

#testing SARGE
deb ftp://ftp.es.debian.org/debian/ testing main non-free contrib
#deb-src ftp://ftp.es.debian.org/debian/ testing main non-free contrib
deb http://non-us.debian.org/debian-non-US testing/non-US main contrib
non-free
#deb-src http://non-us.debian.org/debian-non-US testing/non-US main
contrib non-free

#inestable SID
#deb ftp://ftp.es.debian.org/debian/ unstable main non-free contrib
#deb-src ftp://ftp.es.debian.org/debian/ unstable main non-free contrib
#deb http://non-us.debian.org/debian-non-US unstable/non-US main contrib
non-free
#deb-src http://non-us.debian.org/debian-non-US unstable/non-US main
contrib non-free
```

Aquest és un sources.list ja configurat per a una Debian Sarge, i pot ser que el vostre estigui ja configurat o no, però ens servirà per explicar què significa cada cosa.

1) Tot el que es trobi després de '#' es considera un comentari i serà ignorat. Com podeu veure jo tinc comentades moltes línies.

2) El format de les línies és el següent:

tipus_de_paquet font distribució [component1] [component2] [...]

En "tipus de paquet" va o bé 'deb', 'deb-src' o bé 'rpm', 'rpm-src' depenent del tipus de paquets que usem. En "font" indiquem on està el repositori, que sol ser o bé alguna cosa com:

cdrom:[Debian GNU/Linux testing _#Sarge_ - #Official Snapshot i386 Binary-1 (20050318)]/ el que ens indica que és un cd i en aquest cas concret el d'instal·lació.

O bé és una adreça http o ftp, com podeu veure en l'exemple. El següent és la distribució, en el meu cas a l'usar Debian apareixen unstable i testing, que són 2 de les 3 branques que té Debian (stable, testing i unstable amb els noms de Woody, Sarge i Sid respectivament). I finalment, els components que poden existir o no, en el meu cas es corresponen amb la forma d'organitzar els paquets de Debian. En cada distribució aquestes coses canvien, depèn de com s'estructurin internament.

Un altre exemple amb un sources.list de la versió Hoary de la distribució Ubuntu:

```
# cdrom
deb cdrom:[Ubuntu 5.04 _Hoary Hedgehog_ - Release i386 (20050407)]/ hoary
main restricted

# http
deb http://es.archive.ubuntu.com/ubuntu hoary main restricted
deb-src http://es.archive.ubuntu.com/ubuntu hoary main restricted

deb http://es.archive.ubuntu.com/ubuntu hoary universe
deb-src http://es.archive.ubuntu.com/ubuntu hoary universe

# paquets de seguretat
deb http://security.ubuntu.com/ubuntu hoary-security main restricted
deb-src http://security.ubuntu.com/ubuntu hoary-security main restricted

deb http://security.ubuntu.com/ubuntu hoary-security universe
deb-src http://security.ubuntu.com/ubuntu hoary-security universe
```

Però tranquils, que encara que com veieu és fàcil, no haureu de crear les línies vosaltres mateixos (però us ho explico perquè és bo que sapiguen què signifiquen), ni investigar com és internament la vostra distribució. Nosaltres només ens limitarem a buscar per internet quines són les línies que hem d'afegir. Oi que és fàcil?

Així que només us queda buscar aquestes línies, preferiblement agafant sempre de les fonts oficials. Tranquils, és molt fàcil i la xarxa està plena d'elles ;) i depenent de quins programes o aplicacions us caldran més o menys repositoris.

Per fer aquesta recerca disposem d'una eina molt útil:

apt-setup

Aquesta eina ens busca els repositoris ftp o http del país que nosaltres triem i ens afegeix les línies a l'arxiu 'sources.list', és a dir, ens ho fa tot.

Sens dubte, en això de repositoris i nombre de paquets guanya per golejada Debian, que és una de les grans virtuts d'aquesta distribució, encara que no sigui precisament la més fàcil d'instal·lar i engegar. Sabent buscar bé repositoris és possible que mai necessiteu compilar res, o molt poc, per a aquesta distribució.

Ja ho tinc tot, que comenci la festa

Ja hem arribat a l'interessant, i no em digueu que us ha estat difícil arribar fins a aquí perquè no m'ho crec. Per començar amb el nostre sources.list ja configurat, el que hem de fer és sincronitzar l'índex de paquets respecte a les fonts (els repositoris apuntats pel nostre sources.list):

apt-get update

Res més fàcil que executar això cada vegada que canviem les fonts del sources.list. Com els repositoris solen actualitzar-se molt sovint, sobretot els de Debian, és convenient que executeu això amb freqüència per a no trobar problemes a l'hora d'instal·lar. Vegem un exemple:

```
ramon@portatil:~$ su
Password:
root@portatil:/home/ramon # apt-get update
Obj http://non-us.debian.org testing/non-US/main Packages
Des:1 http://debian.tu-bs.de unstable/main Packages [17,6kB]
Obj http://non-us.debian.org testing/non-US/main Release
Obj http://non-us.debian.org testing/non-US/contrib Packages
```

```

Obj http://non-us.debian.org testing/non-US/contrib Release
Obj http://non-us.debian.org testing/non-US/non-free Packages
Obj http://non-us.debian.org testing/non-US/non-free Release
Des:2 http://debian.tu-bs.de unstable/main Release [112B]
Des:3 ftp://ftp.es.debian.org testing/main Packages [3151kB]
Des:4 ftp://ftp.es.debian.org testing/main Release [81B]
Des:5 ftp://ftp.es.debian.org testing/non-free Packages [58,6kB]
Des:6 ftp://ftp.es.debian.org testing/non-free Release [85B]
Des:7 ftp://ftp.es.debian.org testing/contrib Packages [55,8kB]
Des:8 ftp://ftp.es.debian.org testing/contrib Release [84B]
 Descarregats 3284kB en 3m32s (15,5kB/s)
 Llegint llista de paquets... Fet
root@portatil:/home/ramon #

```

Veieu què fàcil, no heu de fer res per la vostra banda, només esperar.

Sempre que feu un 'apt-get update' és bo fer:

apt-get check

que no és més que una eina de diagnòstic. Actualitza la caché de paquets, torna a crear un nou arbre de dependències i busca dependències impossibles de resoldre, encara que si no enteneu res d'això no importa, simplement sapigheu que és bo realitzar-lo, sobretot per saber si tot ha sortit bé:

```

root@portatil:/home/ramon # apt-get check
Llegint llista de paquets... Fet
Creant arbre de dependències... Fet
root@portatil:/home/ramon #

```

Instal·lar i actualitzar, 2 en 1

Si un paquet ja està instal·lat, apt ho actualitza a l'última versió, i en cas contrari ho instal·la. El mateix amb totes les seves dependències, que es resolen automàticament. Quines més podem demanar quan és especial per a còmodes :).

apt-get install llista_de_paquets

Per veure un exemple d'això instal·larem avui un petit editor de text compatible en gran mesura amb els '.doc' de Windows, Abiword (encara que si voleu un bon paquet ofimàtic ja sabeu, OpenOffice.org):

```

root@portatil:/home/ramon # apt-get install abiword
Llegint llista de paquets... Fet
Creant arbre de dependències... Fet
S'instal·laran els següents paquets extres:
  abiword-common libenchant1
Paquets suggerits:
  abiword-plugins abiword-plugins-gnome abiword-doc
Paquets recomanats
  abiword-help
S'instal·laran els següents paquets NOUS:
  abiword abiword-common libenchant1
0 actualitzats, 3 s'instal·laran, 0 per eliminar i 3 no actualitzats.
Necessito descarregar 3974kB d'arxius.
S'utilitzaran 13,6MB d'espai de disc addicional després de desempaquetar.
Desitja continuar [S/n]? s
Des:1 ftp://ftp.és.debian.org testing/main abiword 2.0.14-1 [1885kB]
Des:2 ftp://ftp.és.debian.org testing/main gnome-desktop-environment 61
[9960B]
Des:3 ftp://ftp.és.debian.org testing/main gnome-core 61 [9782B]
 Descarregats 1905kB en 1m28s (21,4kB/s)
 (Llegint la base de dades ...)
...
...
Desempaquetant el reemplaçament de gnome-core ...
Configurant abiword (2.0.14-1) ...
Configurant gnome-core (61) ...
Configurant gnome-desktop-environment (61) ...
root@portatil:/home/ramon #

```

Podeu comprovar que 'apt-get' amb la seva opció 'install' s'ha ocupat de tot, fins i tot d'eliminar paquets que ja no seran necessaris, cosa que no he comentat abans. Com veieu és el millor per mantenir net el sistema.

Desinstal·lar té també 2 formes

Desinstal·lar té 2 formes, encara que només per als que usem paquets 'deb'.

```
apt-get remove llista_de_paquets
```

```
apt-get --purge remove llista_de_paquets
```

El primer és el que elimina el paquet que li indiques, i el segon el que a més elimina els fitxers de configuració del paquet, que ens seran necessaris quan instal·lem de forma errònia, desinstal·lem i instal·lem de nou. Per a això, per variar, altre exemple:

```
root@portatil:/home/ramon # apt-get remove abiword
Llegint llista de paquets... Fet
Creant arbre de dependències... Fet
Els següents paquets s'ELIMINARAN:
abiword abiword-common
0 actualitzats, 0 s'instal·laran, 2 per a eliminar i 22 no actualitzats.
Necessito descarregar 0B d'arxius.
S'alliberaran 11,0MB després de desempaquetar.
Desitja continuar? [S/n] s
(Llegint la base de dades ...)
146960 fitxers i directoris instal·lats actualment.)
Desinstal·lant abiword-common ...
Desinstal·lant abiword ...
root@portatil:/home/ramon #
```

```
root@portatil:/home/ramon # apt-get --purge remove abiword
Llegint llista de paquets... Fet
Creant arbre de dependències... Fet
Els següents paquets s'ELIMINARAN:
abiword* abiword-common*
0 actualitzats, 0 s'instal·laran, 2 per a eliminar i 22 no actualitzats.
Necessito descarregar 0B d'arxius.
S'alliberaran 11,0MB després de desempaquetar.
Desitja continuar? [S/n] s
(Llegint la base de dades ...)
146960 fitxers i directoris instal·lats actualment.)
Desinstal·lant abiword-common ...
Desinstal·lant abiword ...
root@portatil:/home/ramon #
```

Com veieu 'apt-get remove' també elimina tot el que ja no necessitem, incloses les dependències, que ja no són usades per cap altre paquet.

Oh no! Tenim un problema

Moltes vegades si no fem 'apt-get update' de forma habitual ens podem trobar amb el problema que no poden instal·lar-se els paquets perquè ja no es troben, o bé has instal·lat alguna cosa que no venia en els repositoris i van ocórrer dependències, en qualsevol cas sempre que es trobin dependències trencades. Com podreu suposar aquest problema és més probable que es doni amb els que usen 'deb' que amb els que usen 'rpm', ja que 'dpkg' instal·la encara que trobi dependències i a 'rpm' en canvi cal dir-li que forci la instal·lació ja que si troba dependències per defecte no instal·laria.

Però aquest és un problema d'allò més senzill de solucionar. En el cas que sigui per no sincronitzar amb 'apt-get update', realitzarem aquest pas. En qualsevol altre cas, realitzar:

```
apt-get -f install
```

El que fa és intentar arreglar un sistema amb dependències actualment trencades, i generalment amb un bon fitxer 'sources.list' el nostre problema desapareixerà. En un sistema sense dependències trencades com el meu, l'execució d'aquesta ordre donaria el següent resultat:

```

root@portatil:/home/ramon # apt-get -f install
Llegint llista de paquets... Fet
Creant arbre de dependències... Fet
0 actualitzats, 0 s'instal·laran, 0 per eliminar i 0 no actualitzats.
root@portatil:/home/ramon #

```

Posa't a l'última i actualitza el teu sistema complet

Sí llegeixes bé, amb apt és possible actualitzar el sistema complet i fins i tot la distribució completa a l'última versió, amb el que amb un bon sistema administrat només necessitaràs una instal·lació.

I ara que ho penso, com actualitzes un sistema Windows?, ah sí, fas copia de seguretat de les teves dades, gairebé segur que et convé formatar la teva partició i fins i tot depenent del cas canviar el format de sistema d'arxius i després comences a instal·lar el nou Windows (en la majoria dels casos pirata), i després a instal·lar de nou els programes des de zero. Si està clar... amb Windows són tot facilitats XD sobretot per mantenir una neteja, un ordre i sobretot per passar per caixa XD.

Hi ha dues formes d'actualitzar, cadascuna d'elles amb les seves peculiaritats, i ambdues són absurdes si abans no es realitza un 'apt-get update'. Vegem quins són i quins matisos té cadascuna d'elles:

```

apt-get upgrade
apt-get dist-upgrade

```

La primera es pot dir que és la més respectuosa, ja que s'usa per instal·lar la versió més nova de tots els paquets instal·lats en el sistema provinents d'alguna de les fonts llistades al 'sources.list'. Els paquets instal·lats amb una nova versió disponible són descarregats i actualitzats, sota cap circumstància es desinstal·laran paquets, o s'instal·laran paquets nous. Les noves versions de programes instal·lats que no puguin ser actualitzats sense canviar l'estat d'instal·lació d'altres paquets no s'instal·laran, mantenint-se la versió actual.

I la segona, a més de realitzar les accions de la primera, utilitza intel·ligentment els canvis de dependències deguts a noves versions de paquets, apt-get té un sofisticat sistema de resolució de conflictes, si és necessari tractarà d'actualitzar els paquets més importants a costa dels menys importants.

Normalment si es fa la segona no es necessitarà realitzar la primera, però si es fa la primera llavors la segona farà encara més actualitzacions. Vegem un exemple d'ambdues:

```

root@portatil:/home/ramon # apt-get upgrade
Llegint llista de paquets... Fet
Creant arbre de dependències... Fet
S'actualitzaran els següents paquets:
docbook-dsssl exim4 exim4-base exim4-config exim4-daemon-light fwbuilder
fwbuilder-common fwbuilder-linux gallery gftp gftp-common gftp-gtk gftp-
text
groff-base klogd lha libfwbuilder6 libmysqlclient12 libsensors3
libssl0.9.6
libtag1 libxine1 manpages menu mysql-client mysql-common mysql-server
sysklogd tasksel xprt-common xprt-xprintorg zlib1g zlib1g-dev
33 actualitzats, 0 s'instal·laran, 0 per a eliminar i 0 no actualitzats.
Necessito descarregar 23,9MB d'arxius.
S'utilitzaran 408kB d'espai de disc addicional després de desempaquetar.
Desitja continuar? [S/n] s
Des:1 ftp://ftp.es.debian.org testing/main zlib1g-dev 1:1.2.2-3 [488kB]
....
Des:33 ftp://ftp.es.debian.org testing/main libssl0.9.6 0.9.6m-1 [1755kB]
Descarregats 23,9MB en 18m31s (21,5kB/s)
Preconfiguring packages ...
....
Configurant exim4-base (4.34-7) ...
....
root@portatil:/home/ramon #

root@portatil:/home/ramon # apt-get dist-upgrade

```

```
Llegint llista de paquets... Fet
Creant arbre de dependències... Fet
Calculant l'actualització... Llest
0 actualitzats, 0 s'instal·laran, 0 per a eliminar i 0 no actualitzats.
root@portatil:/home/ramon #
```

En aquest cas 'apt-get dist-upgrade' no va fer res perquè no va ser necessari, ja que 'apt-get upgrade' va realitzar tot l'imprescindible.

Per actualitzar la teva distribució el normal és canviar els repositoris de la que uses per la que desitges, per exemple, per passar de Debian Sarge a Debian Sid canviarem en sources.list les fonts d'una per la de l'altra i ens llancem amb 'apt-get update', 'apt-get check' i 'apt-get dist-upgrade'. Encara que això funciona bé en Debian, compte, pot ser que en altres distribucions no sigui així i pugueu tenir problemes.

Neteja la cau de l'apt

Apt el que fa és baixar els paquets i després fer el que sigui necessari amb ells. Per a això usa un directori com cau situat en /var/cache/apt/archives i aquest mai es buida, a menys que es faci manualment esborrant els arxius, cosa que no recomano o bé mitjançant altres dues opcions de l'apt amb les seves petites diferències.

Si no ho esborres pot arribar a convertir-se amb el pas del temps en un problema d'espai, per exemple, jo que solc ser descurat ho tinc a la grandària d'1.4 GB, encara que és convenient també no tenir-la sempre buida, sobretot quan instal·les i tornes a instal·lar un mateix paquet, ja que el que trobi en la cau no ho baixarà.

Les dues formes són:

```
apt-get clean
apt-get autoclean
```

La primera esborra totalment el repositori local que conté els fitxers descarregats, i la segona només esborrarà aquells paquets que ja no poden ser descarregats, o són clarament inservibles. Això permet mantenir la cau durant llargs períodes de temps sense que augmenti la seva grandària sense control.

Veurem un exemple d'ambdues, encara que aquesta vegada començarem per la segona:

```
root@portatil:/home/ramon # apt-get autoclean
Llegint llista de paquets... Fet
Creant arbre de dependències... Fet
Del a2ps 1:4.13b-4 [715kB]
Del alsa-base 1.0.5a-3 [47,5kB]
...
Del gftp 2.0.17-6 [34,6kB]
Del manpages-dev 1.67-2 [1044kB]
Del mysql-server 4.0.21-6 [0B]
root@portatil:/home/ramon #
```

Amb això he passat d'una caché d'1.4GB a una de 776.7 MB.

```
root@portatil:/home/ramon # apt-get clean
root@portatil:/home/ramon #
```

I amb això la caché està buida.

Jo només vull que es descarreguin els paquets

Pot ser que ens interessi que a l'instal·lar o actualitzar només descarregui els paquets a la cau i no faci res més amb ells. Per a això usem l'opció '-d' quedant les ordres així:

```
apt-get -d install llista_de_paquets
apt-get -d upgrade
apt-get -d dist-upgrade
```

Això pot ser molt útil si vols instal·lar i no estaràs present com en el cas dels 'deb', que moltes vegades llancen preguntes sobre com s'han de configurar els paquets, o també si instal·lem alguna cosa i com tenim una connexió lenta volem copiar els paquets de la cau a un directori per no tornar-los a baixar mai més :)

Vegem l'exemple només amb la instal·lació:

```

root@portatil:/home/ramon # apt-get -d install abiword
Llegint llista de paquets... Fet
Creant arbre de dependències... Fet
S'instal·laran els següents paquets extres:
abiword-common
Paquets suggerits:
abiword-plugins abiword-plugins-gnome abiword-doc
Paquets recomanats
abiword-help
S'instal·laran els següents paquets NOUS:
abiword abiword-common
0 actualitzats, 2 s'instal·laran, 0 per eliminar i 0 no actualitzats.
Necessito descarregar 3371kB d'arxius.
S'utilitzaran 11,0MB d'espai de disc addicional després de desempaquetar.
Desitja continuar? [S/n] s
Des:1 ftp://ftp.es.debian.org testing/main abiword-common 2.0.14-1
[1486kB]
Des:2 ftp://ftp.es.debian.org testing/main abiword 2.0.14-1 [1885kB]
Descarregats 3371kB en 2m42s (20,7kB/s)
Descàrrega completa i en manera de només descarrega
root@portatil:/home/ramon #

root@portatil:/home/ramon # apt-get install abiword
Llegint llista de paquets... Fet
...
S'utilitzaran 11,0MB d'espai de disc addicional després de desempaquetar.
Desitja continuar? [S/n] s
Seleccionant el paquet abiword-common prèviament no seleccionat.
(Llegint la base de dades ...)
...
Configurant abiword-common (2.0.14-1) ...
root@portatil:/home/ramon #

```

Podeu observar com simplement el baixa i para. A l'executar la mateixa ordre sense '-d', com els troba en cau, ja no baixa res i en aquest cas s'instal·la.

Però si vull reinstal·lar alguna cosa ja instal·lada

Imaginem que se'ns ha corromput l'executable o fitxers d'alguna cosa que ja tenim instal·lada, la nostra primera ocurrència seria:

```

root@portatil:/home/ramon # apt-get install abiword
Llegint llista de paquets... Fet
Creant arbre de dependències... Fet
abiword ja està en la seva versió més recent.
0 actualitzats, 0 s'instal·laran, 0 per a eliminar i 0 no actualitzats.
root@portatil:/home/ramon #

```

Uhhh... realment un problema, ja que per defecte si està instal·lat ja no reinstal·la damunt, però això ho podem solucionar fàcilment amb l'opció '--reinstall'. Vegem l'exemple:

```

root@portatil:/home/ramon # apt-get --reinstall install abiword
Llegint llista de paquets... Fet
Creant arbre de dependències... Fet
0 actualitzats, 0 s'instal·laran, 1 reinstal·lats, 0 per eliminar i 0 no
actualitzats.
Es necessita descarregar 0B/1885kB d'arxius.
S'utilitzaran 0B d'espai de disc addicional després de desempaquetar.
Desitja continuar? [S/n] s
(Llegint la base de dades ...)
146924 fitxers i directoris instal·lats actualment.)
Preparant per reemplaçar abiword 2.0.14-1 (usant
.../abiword_2.0.14-1i386.deb) ...
Desempaquetant el reemplaçament de abiword ...
Configurant abiword (2.0.14-1) ...

```

```
root@portatil:/home/ramon #
```

Com veieu és fàcil guanyar el pols si es coneixen les eines necessàries.

D'acord, tot està bé però falta alguna cosa, I si no conec el nom del que vull instal·lar?

Doncs també tenim una ordre dintre de les utilitats d'apt del qual només veurem una opció, i és la següent:

```
apt-cache search llista_de_paraules
```

Simplement realitza una recerca de l'expressió regular donada en tot el text de tots els paquets disponibles. Busca l'expressió regular tant en el nom dels paquets com en la seva descripció, i mostra el nom del paquet i una petita descripció d'aquest. Així que cercarem jocs que siguin com el Bubble:

```
root@portatil:/home/ramon # apt-cache search bubble
bubblefishymon - system load dockapp with a duck
bubblemon - Bubbling Load Monitoring GNOME Applet
emacs-goodies-el - Miscellaneous add-ons for Emacs
fb-music-high - High quality, large music files for Frozen-Bubble
fb-music-low - Lower quality, small music files for Frozen-Bubble
frozen-bubble - Pop out the bubbles !
frozen-bubble-data - Data files for Frozen-Bubble
gdesklets-data - displays and sensors for gdesklets
gkrellm-bfm - system load plugin for gkrellm with a duck
junior-puzzle - Debian Jr. Puzzles
wmbubble - A system-load meter for Window Maker that features a duck
wmfishtime - Dockable clock app for WMaker, BlackBox, I, SawFish etc
xbubble - A nice Puzzle Bubble clone
xbubble-data - Data files for XBubble, a nice Puzzle Bubble clone
root@portatil:/home/ramon #
```

screen

És un gestor de pantalla amb emulació de terminal.

És un gestor de finestres a pantalla completa que multiplexa un terminal físic entre processos (típicament shells interactius). Quan es crida a screen, crea una finestra simple amb un shell en ella (o el comandament especificat) i llavors surt del seu estat de tal manera que puguis fer servir el programa. L'usuari pot crear noves finestres (a pantalla completa) amb altres programes en ella (inclosos més shells), tancar finestres existents, veure una llista de finestres, iniciar o tancar una sessió, copiar i enganxar text entre finestres, etc. En totes les finestres els programes funcionen completament independents uns d'uns altres. Els programes continuen funcionant encara que la seva finestra no sigui visible en aquest moment i fins i tot quan la sessió completa de screen és separada des del terminal. Quan un programa acaba, screen (per defecte) mata la finestra que la contenia. Si aquesta finestra estava en primer plànol, mostrarà la finestra prèvia, si no queda cap, screen surt.

ssh

És un programa que ens permet iniciar una sessió en una màquina remota o executar comandaments en una màquina remota. Proveeix comunicacions segures mitjançant encriptació entre dues màquines de no confiança sobre una xarxa insegura.

ssh connecta i inicia sessió en la màquina especificada (amb nom d'usuari opcional). L'usuari ha d'indicar la seva identitat a la màquina remota usant el mètode que depèn de la versió del protocol usat. Si s'especifica un comandament, aquest serà executat en la màquina remota en lloc d'un inici de sessió.

Exemple:

```
ssh 192.168.1.13
```

putty: és un programa que funciona en Windows que ens permet fer alguna cosa similar a ssh.

dselect

És la interfície del manipulador de paquets Debian per a consola.

dselect és la principal interfície per a la instal·lació, maneig i eliminació de paquets Debian GNU/Linux. En el menú principal de dselect, l'administrador del sistema pot:

Actualitzar la llista de paquets disponibles.

Veure l'estat dels paquets instal·lats i disponibles.

Alterar seleccions de paquets i manejar les dependències corresponents.

Instal·lar nous paquets o actualitzar-los a noves versions.

dselect opera com interfície de dpkg, l'eina gestora de paquets de Debian de baix nivell.

El seu avantatge és que posseeix una pantalla completa de selecció de paquets amb resolució òptima de dependències i conflictes. Quan s'executa amb permisos d'administrador, els paquets poden ser instal·lats, actualitzats o eliminats. Diversos mètodes d'accés poden ser configurats per saber les versions disponibles i instal·lables des dels repositoris de paquets corresponents. Depenent del mètode d'accés usat, aquests repositoris de paquets poden ser públics en servidors a través d'Internet, xarxes locals o bé en cdroms. El mètode d'accés recomanat és apt, proporcionat pel paquet apt.

Normalment dselect s'invoca sense paràmetres. Es presenta un menú interactiu, oferint a l'usuari una llista d'accions. Si es dona una acció com argument, llavors serà executada immediatament. Hi ha moltes ordres disponibles per modificar el comportament de dselect o també per mostrar informació addicional sobre el programa.

L'eina dselect presenta diverses fallades:

La interfície de selecció de dselect pot semblar confusa per a alguns usuaris principiants.

S'ha informat, que fins i tot experimentats desenvolupadors del kernel s'han queixat.

Manca de documentació.

No hi ha cap opció d'ajuda en el menú principal.

La llista de paquets disponibles no pot ser reduïda.

Els altres mètodes d'accés no tenen tanta qualitat com el mètode que proporciona apt, no implica que els altres no funcionin, sinó que aquest últim és molt més flexible, la qual cosa ho fa ser molt més recomanable.

aptitude

És un gestor de paquets, del sistema de paquets de Debian GNU/Linux, d'interfície d'alt nivell basada en text.

Permet a l'usuari veure la llista de paquets i realitzar tasques de gestió tals com instal·lar, actualitzar, i eliminar paquets. Les accions poden ser realitzades des de la interfície visual o des de la línia de comandaments.

Repositoris

Un repositori és un conjunt de paquets Debian organitzats en un directori en arbre especial, el qual també conté alguns arxius addicionals amb els índexs i informació dels paquets. Si un usuari afegix un repositori al seu arxiu sources.list, ell pot veure i instal·lar fàcilment tots els paquets disponibles en aquest igual que els paquets continguts en Debian. Els repositoris de la versió 6.06 són:

```
deb http://es.archive.ubuntu.com/ubuntu dapper main restricted
deb-src http://es.archive.ubuntu.com/ubuntu dapper main restricted

deb http://es.archive.ubuntu.com/ubuntu dapper-updates main restricted
deb-src http://es.archive.ubuntu.com/ubuntu dapper-updates main
restricted

deb http://es.archive.ubuntu.com/ubuntu dapper universe
deb-src http://es.archive.ubuntu.com/ubuntu dapper universe

deb http://security.ubuntu.com/ubuntu dapper-security main restricted
```

```
deb-src http://security.ubuntu.com/ubuntu dapper-security main restricted
deb http://security.ubuntu.com/ubuntu dapper-security universe
deb-src http://security.ubuntu.com/ubuntu dapper-security universe
```

Ordre d'arrencada

També podem canviar l'ordre d'arrencada dels sistemes operatius (sempre que tinguem instal·lats diversos). Per fer això editarem l'arxiu `/boot/grub/menu.lst`

```
ramon@portatil:~$ sudo gedit /boot/grub/menu.lst
```

Muntar altres particions en l'engegada

Si volem tenir a la nostra disposició fàcilment les particions dels altres sistemes operatius podem assignar-los directoris. En primer lloc crearem directoris dintre del punt de muntatge `/mnt` per a cada partició. Faig un exemple amb tres particions. Jo anomeno als directoris `sistema1`, `sistema2` i `sistema3`, però tu pots donar-los el nom que vulguis.

```
ramon@portatil:~$ sudo mkdir /mnt/sistema1
ramon@portatil:~$ sudo mkdir /mnt/sistema2
ramon@portatil:~$ sudo mkdir /mnt/sistema3
```

Ara editem l'arxiu `/etc/fstab`

```
ramon@portatil:~$ sudo gedit /etc/fstab
```

Afegim les següents línies i desem l'arxiu. (Substitueix X, Y, i Z pels números que pertoquin. Si tens dubtes consulta el tema [particions de la sessió 14/03](http://www.proferamon.com/1403.html/#particions) <http://www.proferamon.com/1403.html/#particions>)

```
/dev/hdaX /mnt/sistema1 vfat rw,user,noauto 0 0
/dev/hdaY /mnt/sistema2 nfts ro,user,noauto 0 0
/dev/hdaZ /mnt/sistema3 ext3 rw,user,noauto 0 0

/dev/hda5 /media/hda5 vfat uid=nomusuari,umask=0007,fmask=0117,utf8 0 0
#utf8 corregeix els accents
```

Tema

Si no ens agraden les icones, el fons d'escriptori, les vores de la finestra o els controls podem canviar-los al menú Sistema / Preferències / Tema. Ens podem descarregar fons d'escriptori, icones, etc de la web [Gnome - Artwork & Themes](http://art.gnome.org/) (<http://art.gnome.org/>)

Consola de java

Si no sou tan integristes com Richard Stallman potser vulgueu instal·lar un programa que, encara que no és programari lliure, és gratuït. Em refereixo a una consola de java. Moltes de les pàgines web la necessiten per mostrar els seus continguts. Començarem descarregant l'arxiu d'instal·lació de la web de descàrregues de programari [java per a GNU/Linux](http://java.com/es/download/linux_manual.jsp). (http://java.com/es/download/linux_manual.jsp) Heu de tenir present de prémer el botó GNU/Linux (archivo autoextraíble). L'altre botó que apareix és per a distribucions que fan servir paquets RPM (Fedora, SuSE, Mandriva, etc).

Ara que el tenim descarregat creem un directori per fer la instal·lació i movem l'arxiu descarregat a aquest directori. Atenció: Jo he descarregat l'arxiu en el meu directori personal, potser vosaltres el teniu en altre directori.

```
ramon@portatil:~$ sudo mkdir /usr/java
ramon@portatil:~$ sudo mv jre-1_5_0_04-linux-i586.bin /usr/java
```

Ens desplaçem al directori d'instal·lació. Canviem els permisos de l'arxiu per poder-lo executar i l'executem.

```
ramon@portatil:~$ cd /usr/java
ramon@portatil:/usr/java$ sudo chmod a+x jre-1_5_0_04-linux-i586.bin
ramon@portatil:/usr/java$ sudo ./jre-1_5_0_04-linux-i586.bin
```

Ens ha creat un directori en el qual s'ha instal·lat la consola de java. Ara podem esborrar l'arxiu que ens havíem descarregat ja que ja no ho necessitem i crearem un enllaç simbòlic al plugin perquè funcioni en el nostre navegador. Jo ho faré només per a Mozilla Firefox, però ho heu de fer per a cada navegador que useu.

```
ramon@portatil:/usr/java$ sudo rm jre-1_5_0_04-linux-i586.bin
ramon@portatil:/usr/java$ cd /usr/lib/mozilla-firefox/plugins
ramon@portatil:/usr/lib/mozilla-firefox/plugins$ sudo ln -s
/usr/java/jre1.5.0_04/plugin/i386/ns7/libjavaplugin_oji.so
```

I ara podem [comprovar](http://www.java.com/en/download/help/testvm.xml) (http://www.java.com/en/download/help/testvm.xml) que estigui ben instal·lada. Si funciona ens apareixerà una imatge semblant a aquesta.

Test your JVM

```
Congratulations. The latest version is installed on your system.
Your Java configuration is:
  Vendor:      Sun Microsystems Inc.
  Version:    1.5.0_04
  Operating System: Linux
  OS version: 2.6.10-5-386
```



Plugin de flash

M'imagino que també voldreu instal·lar el plugin de flash. En primer lloc heu d'entrar en la web del [centre de descàrregues d'Adobe per a GNU/Linux](http://www.adobe.com/shockwave/download/download.cgi?P1_Prod_Version=ShockwaveFlash) (http://www.adobe.com/shockwave/download/download.cgi?P1_Prod_Version=ShockwaveFlash) i prémer sobre el botó 'Descargar ahora mismo'. D'aquesta manera haureu descarregat l'arxiu install_flash_player_7_linux.tar.gz Ara és el moment de tancar el navegador i obrir el terminal. Estant en el directori de descàrrega escriviu:

```
ramon@portatil:~$ sudo tar -zxvf install_flash_player_7_linux.tar.gz
```

Es crearà un directori anomenat install_flash_player_7_linux. Entreu en ell i crideu a l'arxiu instal·lador.

```
ramon@portatil:~/install_flash_player_7_linux$ sudo ./flashplayer-
installer
```

Diccionaris de l'OpenOffice.org 2.0

Descarregar els diccionaris del [ftp de diccionaris de l'Openoffice.org](http://ftp.services.openoffice.org/pub/OpenOffice.org/contrib/dictionaries/) (http://ftp.services.openoffice.org/pub/OpenOffice.org/contrib/dictionaries/)

En el cas català només trobareu el diccionari ortogràfic ca_ES.zip

En el cas espanyol són tres arxius es_ES.zip, hyph_es_ES.zip i thes_es_ES.zip

Descomprimir els arxius en /home/usuari/.openoffice/version/user/wordbook (substituïu usuari pel vostre nom d'usuari)

Editar l'arxiu /etc/openoffice/dictionary.lst i afegiu al final aquestes línies

```

DICT ca ES ca_ES
DICT es ES es_ES
HYPH es ES hyph_es_ES
THES es ES th_es_ES

```

Reproductor multimèdia vlc

Si vols poder reproduir arxius multimèdia (àudio o vídeo) i no tenir problemes sigui el que sigui el seu format has d'instal·lar el programa vlc.

```

ramon@portatil:~$ sudo apt-get install
vlc wxvlc vlc-plugin-esd

```

Configurar el compte de gmail amb gaim

Protocol: Jabber

Nom d'usuari: *****

Servidor: gmail.com

Contrasenya: *****

Opcions de Jabber

Port: 5222

Connectar amb el servidor: talk.google.com

Compartir arxius amb NFS (Entre màquinesGNU/Linux)

En primer lloc instal·lem els paquets nfs-common i nfs-user-server

```

ramon@portatil:~$ sudo apt-get install
nfs-common nfs-user-server

```

Editem l'arxiu /etc/exports per a indicar quines carpetes han de compartir-se, amb quines màquines i amb quins permisos.

Escric uns exemples:

```

/home/usuari          192.168.1.0/255.255.255.0      rw
/media/hda2          192.168.1.4/255.255.255.255   ro

```

En el primer cas compartim la carpeta /home/usuari amb totes les màquines de la xarxa i amb permisos de lectura i escriptura.

En el segon cas compartim la carpeta /media/hda2 amb la màquina que té la IP 192.168.1.4 i només amb permís de lectura.

Reiniciem el dimoni

```

ramon@portatil:~$ sudo /etc/init.d/nfs-user-server restart

```

Per accedir des d'altra màquina emprarem una sintaxi similar a aquesta mount ip:directori destinació. Vegem un exemple:

```

ramon@portatil:~$ sudo mount 192.168.1.3:/home/usuari /mnt/home/usuari

```

Canviar la imatge del splash

En primer lloc pots descarregar alguna imatge de art.gnome.org. Després des del terminal deses la imatge actual amb un altre nom i enllaçes amb la teua imatge.

```
ramon@portatil:~$ sudo mv /usr/share/pixmaps/splash/ubuntu-splash.png
/usr/share/pixmaps/splash/ubuntu-splash.png_backup
ramon@portatil:~$ sudo ln -s /ruta_al_teu_splash
/usr/share/pixmaps/splash/ubuntu-splash.png
```

Posar una imatge de fons en el menú del grub

El primer que has de fer és aconseguir una imatge ja sigui de la web o bé te la construeixes tu mateix. La imatge ha de tenir una resolució de 640x480 píxels i 14 colors com a màxim.

Ho pots fer amb una imatge qualsevol. L'obres amb el Gimp i amb 'Imatge / Escala la imatge' li canvies la resolució a 640x480.

Ara toca reduir els colors. Obres el menú 'Imatge / Mode / Indexat' i estableixes el número màxim de colors a 14.

Finalment deses una còpia de la imatge amb la extensió xpm fent 'Fitxer / Desa una còpia'

Un cop feta la imatge la comprimim amb gzip i la movem al directori del grub.

```
ramon@portatil:~$ gzip imatge.xpm
ramon@portatil:~$ sudo mv imatge.xpm.gz /boot/grub
```

Obrim l'arxiu /boot/grub/menu.lst per modificar-lo

```
ramon@portatil:~$ sudo gedit /boot/grub/menu.lst
```

i afegim la següent línia

```
splashimage (hd0,0)/boot/grub/imatge.xpm.gz
```

Tingues present que (hd0,0) es refereix al primer disc dur ide (primer zero) i la primera partició primària (segon zero). Si tens instal·lat el teu sistema en altra partició o un disc dur secundari hauràs de modificar aquests números.

Configuració d'un mòdem 56k.

pppconfig serveix per configurar connexions fil per randa a Internet.

pppconfig és un diàleg interactiu que ens ajuda a configurar la nostra connexió a Internet. Compte amb explicacions per a cada pas. Usa els arxius de configuració ppp estàndard. Algunes característiques són:

Múltiples proveïdors de serveis d'Internet amb servidors de noms separats.

Detecció del mòdem.

DNS dinàmic.

Marcats a petició.

Permet als usuaris no root executar ppp.

pon engega el log de connexions ppp, /etc/ppp/ppponboot, si existeix i és executable. La connexió ppp serà engegada usant la configuració de /etc/ppp/peers/provider llevat que se li passi l'argument isp-name (només serà necessari si tenim més d'una configuracions).

poff apaga el log de connexions ppp.

Configuració de la targeta de xarxa

L'arxiu `/etc/network/interfaces` conté informació de la configuració dels dispositius de xarxa disponibles i de com activar-los.

```
ramon@portatil:~$ cat /etc/network/interfaces
# This file describes the network interfícies available on your system
# and how to activate them. For more information, see interfícies(5).

# The loopback network interface
auto lo
iface lo inet loopback

# This is a list of hotpluggable network interfaces.
# They will be activated automatically by the hotplug subsystem.
mapping hotplug
    script grep
    map wlan0

# The primary network interface
iface wlan0 inet static
    address 192.168.1.10
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
    # wireless-* options are implemented by the wireless-tools
package
    wireless-mode managed
    wireless-essid 3Com
    # dns-* options are implemented by the resolvconf package, if
installed
    dns-nameservers 192.168.1.1
    dns-search site
    wirelessessid 3Com

auto wlan0
ramon@portatil:~$
```

Si volem que la configuració del dispositiu sigui estàtica farem les següents modificacions:

```
iface eth0 inet static
address 192.168.1.2
netmask 255.255.255.0
gateway 192.168.1.1
broadcast 192.168.1.255
network 192.168.1.0
```

Si volem que la configuració del dispositiu sigui dinàmica, és a dir, que prengui la configuració del DHCP del nostre router, farem les següents modificacions:

```
iface eth0 inet dhcp
```

L'arxiu `/etc/resolv.conf` conté informació dels DNS.

```
ramon@portatil:~$ cat /etc/resolv.conf
search site
nameserver 192.168.1.1
domain GRT
ramon@portatil:~$
```

Si volguéssim tenir uns DNS assignats per nosaltres i no pel router hauríem d'editar l'arxiu `/etc/resolv.conf` i indicar els DNS que volem usar. Per exemple:

```
nameserver 194.224.52.36
nameserver 194.224.52.37
```

Després d'haver modificat les configuracions dels dispositius de xarxa hem de parar-los i encendre'ls. Per a això usarem les ordres:

```
ifdown: apaga una interfície de xarxa
ifup: engega una interfície de xarxa
```

Per obtenir una llista de tots els dispositius pci connectats a la màquina disposem de l'ordre lspci

```
ramon@portatil:~$ lspci
0000:00:00.0 Host bridge: Intel Corp. 82845 845 (Brookdale) Chipset Host
Bridge (rev 04)
0000:00:01.0 PCI bridge: Intel Corp. 82845 845 (Brookdale) Chipset AGP
Bridge (rev 04)
0000:00:1i.0 PCI bridge: Intel Corp. 82801 PCI Bridge (rev 05)
0000:00:1f.0 ISA bridge: Intel Corp. 82801BA ISA Bridge (LPC) (rev 05)
0000:00:1f.1 IDE interface: Intel Corp. 82801BA IDE O100 (rev 05)
0000:00:1f.2 USB Controller: Intel Corp. 82801BA/BAM USB (Hub #1) (rev
05)
0000:00:1f.3 SMBus: Intel Corp. 82801BA/BAM SMBus (rev 05)
0000:00:1f.4 USB Controller: Intel Corp. 82801BA/BAM USB (Hub #2) (rev
05)
0000:00:1f.5 Multimedia audio controller: Intel Corp. 82801BA/BAM AC'97
Audio (rev 05)
0000:00:1f.6 Mòdem: Intel Corp. Intel 537 [82801BA/BAM AC'97 Modem] (rev
05)
0000:01:00.0 VGA compatible controller: ATI Technologies Inc Radeon
Mobility M6 LY
0000:02:00.0 CardBus bridge: Texas Instruments PCI1420
0000:02:00.1 CardBus bridge: Texas Instruments PCI1420
0000:02:08.0 Ethernet controller: Intel Corp. 82801BA/BAM/CA/CAM Ethernet
Controller (rev 03)
0000:03:00.0 Network controller: Texas Instruments ACX 111 54Mbps
Wireless Interface
ramon@portatil:~$
```

Amb l'opció `-vvx` obtindrem una informació detallada.

Com instal·lar més tipus de lletra

Ens descarreguem la lletra des de <http://apostrophiclab.pedroreina.net> i descomprimim l'arxiu en una carpeta

```
$ cd /usr/share/fonts/truetype
$ sudo mkdir nom_de_la_font
$ cd nom_de_la_font
$ sudo cp directori_de_la_font_descarregada/*.ttf ./
```

Com instal·lar les Microsoft TrueType Fonts

Si voleu tenir a la vostra disposició les fonts de Microsoft que habitualment s'usen a Windows heu d'instal·lar el paquet `msttcorefonts`.

```
$sudo apt-get install msttcorefonts
```

Aquest paquet proveeix les fonts de la llista:

- Andale Mono
- Arial Black
- Arial (Negreta, Cursiva, Negreta Cursiva)

- Comic Sans MS (Negreta)
- Courier New (Negreta, Cursiva, Negreta Cursiva)
- Georgia (Negreta, Cursiva, Negreta Cursiva)
- Impact
- Times New Roman (Negreta, Cursiva, Negreta Cursiva)
- Trebuchet (Negreta, Cursiva, Negreta Cursiva)
- Verdana (Negreta, Cursiva, Negreta Cursiva)
- Webdings

Com mostrar una imatge quan s'inicia el sistema

1. Necessitem un nou repositori del que descarregar el programa Splashy, pel que hauràs d'editar-lo:

```
$ sudo gedit /etc/apt/sources.list
```

2. I afegir la següent línia:

```
deb http://splashy.aliioth.debian.org/debian/ unstable main
```

3. Ara actualitza la base de dades de paquets:

```
$ sudo aptitude update
```

4. I instal·lem el programa:

```
$ sudo aptitude install splashy
```

5. Editar `/boot/grub/menu.lst` per afegir l'opció `vga=792` a la teva opció per defecte d'arrencada.

Per exemple:

```
title Ubuntu, kernel 2.6.10-5-k7 Default
root (hd0,0)
kernel /vmlinuz root=/dev/hde6 ro quiet splash vga=792
initrd /initrd.img
savedefault
boot
```

Amb això aconseguiràs arrencar a 1024x768 i milions de colors.

6. Pots descarregar més temes des d'aquí: <http://splashy.aliioth.debian.org/themes/>

7. Descomprimeix el tema:

```
$ tar xzf ubuntu_splashy_theme.tar.gz
```

8. Copia els fitxers al directori de splashy:

```
$ sudo cp -a ubuntu/ /etc/splashy/themes
```

9. Canviem el fitxer de configuració de splashy:

```
$ sudo mv /etc/splashy/config.xml /etc/splashy/config.xml.old
$ sudo mv /etc/splashy/themes/ubuntu/config.xml /etc/splashy
```

OK! Ja està tot! La pròxima vegada que reiniciïs veuràs la barra de progrés pujant. Per a tornar a la manera clàssica, prem F2.

Instal·lació de VirtualBox OSE (Open Source Edition)

VirtualBox OSE és un programa de virtualització que ens pot servir per executar altres sistemes operatius en màquines virtuals a dins del nostre GNU/Linux.

Instal·lem els paquets, creem el mòdul vboxdrv i l'afegim al nucli

```
ramon@portatil:~$ sudo aptitude install virtualbox-ose virtualbox-
ose-source module-assistant
ramon@portatil:~$ sudo m-a prepare
ramon@portatil:~$ sudo m-a a-i virtualbox-ose
ramon@portatil:~$ sudo modprobe vboxdrv
```

Editem l'arxiu `/etc/modules`

```
ramon@portatil:~$ sudo gedit /etc/modules
```

Afegim el mòdul vboxdrv perquè es carregui a l'inici

```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be
loaded
# at boot time, one per line. Lines beginning with "#" are
ignored.

fuse
lp
sbp2
vboxdrv
```

I ara només ens falta afegir-nos com a usuari del grup vboxusers

```
ramon@portatil:~$ sudo adduser nom_del_teu_usuari vboxusers
```

Per acabar, reiniciem l'ordinador.

5. Aplicacions

Navegadors web: Mozilla, Mozilla Firefox, Konqueror, Epiphany, Galeon
Client de correu: Thunderbird, Kmail
Suite de gestió d'informació personal: Evolution, Kontact
Programes P2P: aMule, Imule, xMule, Transmission, Bittorrent, azureus
Missatgeria instantània: Pidgin, Kopete, aMSN
Multimèdia àudio: CD Player, Banshee, Rhythmbox, Sound Juicer, Streamtuner, XMMS
Multimèdia vídeo: Vlc, Totem, Xine, gxine, mplayer
Descàrrega de vídeo: Miro
Edició de vídeo: Kino, pitivi, subtitle editor
Gravació i edició de so: Audacity, Sound Recorder
Partitures musicals: Denemo, Lilypond
Entrenament musical: Solfege
Gravació de cd's: Brasero, K3b, Gnomebaker, k9copy
Edició d'imatge: Gimp
Gràfics vectorials: Inkscape, Karbon
Animació 3d: Blender
Edició de diagrames: Dia, Inkscape, Skencil
Visualitzadors de documents PDF: Evince, Kpdf, Xpdf
Petites utilitats: Wget, Lynx, Gedit
Gestió: Gparted, Alacarte, Partimage, Krusader
Disseny web: Nvu, Bluefish, Quanta Plus, Mozilla Composer, Flash4linux
Programació: Gambas, Mono, Kdevelop
Ofimàtica: Openoffice, Abiword, Gnumeric, mySQL
Gestió de projectes: Planner, Project Management
Maquetació: Scribus
Simulació de màquines: VirtualBox, VMWare, Qemu
Producció: abanq (abans facturalux)
Cad: Qcad
VNC: Vinagre
Proxy: Squid
Tallafocs: Iptables, ufw
Servidor de webs: Apache
Antivirus: Clamav

Índex de contingut

Curs de GNU/Linux.....	1
1. Introducció.....	1
Una mica d'història.....	1
El túnel fosc.....	1
El moviment GNU.....	1
A la recerca d'un kernel.....	2
L'expansió de GNU/Linux.....	3
Característiques del Programari Lliure.....	3
Diversos "cognoms" del programari.....	4
Els avantatges d'usar programari lliure.....	5
2. El sistema operatiu GNU/Linux.....	7
Definició de sistema operatiu.....	7
Característiques del sistema operatiu GNU/Linux.....	7
Distribucions.....	8
distrowatch.com.....	8
Tipus de cd's: d'instal·lació i live-cd.....	8
Com aconseguir els cd's.....	9
3. La distribució Ubuntu.....	10
Característiques.....	10
Organització de paquets.....	11
Llançaments.....	11
4. Administració i configuració.....	13
L'estructura de directoris.....	13
L'interpret d'ordres o SHELL.....	13
Visió d'usuari d'un sistema GNU/Linux	14
Tipus d'usuari	14
L'entorn.....	14
Resum de les ordres bàsiques.....	15
Movent-nos pels directoris - cd.....	16
M'he perdut. On estic? - pwd.....	17
Llistant arxius - ls.....	17
Creant directoris - mkdir.....	19
Eliminant directoris - rmdir.....	19
Tampoc m'agrada teclejar tant - [TAB].....	20
Copiant - cp.....	20
Esborrant - rm.....	21
Movent - mv.....	21
Comodins - *.....	22
Consultant informació - man.....	22
Històric - history.....	23
Tancament - halt.....	23
Entrada i sortida estàndard - cat.....	23
Redirecció d'entrada i sortida - >.....	23
Redirecció de sortida no destructiva - >>.....	24
Pipes (canonades) - 	25
Els permisos dels fitxers - rwxrwxrwx.....	25
Interpretant els permisos.....	26
Canviant permisos - chmod.....	26
Canviant propietari i/o grup - chown.....	27
Gestió d'usuaris.....	27
Conceptes de gestió d'usuaris.	28
Afegir usuaris - adduser.....	28
Modificant a l'usuari - usermod.....	29
Jo puc canviar-me la meva contrasenya? - passwd.....	29
Eliminant usuaris - userdel.....	29

Gestió de grups.....	29
Afegint grups - addgroup.....	30
Eliminat els grups - groupdel.....	30
Altres ordres.....	30
Control de tasques.....	31
Enviant al segon pla - &.....	32
Però on s'ha ficat? - jobs.....	32
Més formes de matar - kill.....	32
Aturar i continuar - Ctrl + z.....	33
Recapitulant	33
Més sobre 'fg' i 'bg'	33
Ordres associades - ps, top, nice, renice, pidof, nohup	34
Tasques programades: AT i CRON.....	35
AT.....	35
CRON.....	36
Enllaços - ln.....	37
Enllaços durs	37
Enllaços simbòlics	38
Muntant unitats - mount.....	38
I ara com trec el meu CD? No em deixa! - umount.....	40
No em desmunta la unitat, què faig? - eject.....	40
Instal·lació de programes.....	40
Què és això d'apt	40
D'on obtinc el necessari per al conjunt d'eines d'apt?	41
Bé, ja està instal·lat, però no corris. Necessitem saber una mica més	41
Ja ho tinc tot, que comenci la festa	42
Instal·lar i actualitzar, 2 en 1	43
Desinstal·lar té també 2 formes	44
Oh no! Tenim un problema	44
Posa't a l'última i actualitza el teu sistema complet	45
Neteja la cau de l'apt	46
Jo només vull que es descarreguin els paquets	46
Però si vull reinstal·lar alguna cosa ja instal·lada	47
D'acord, tot està bé però falta alguna cosa, I si no conec el nom del que vull instal·lar? ...	48
screen.....	48
ssh.....	48
dselect.....	49
aptitude.....	49
Repositoris.....	49
Ordre d'arrencada.....	50
Muntar altres particions en l'engegada.....	50
Tema.....	50
Consola de java.....	50
Plugin de flash.....	51
Diccionaris de l'OpenOffice.org 2.0.....	51
Reproductor multimèdia vlc.....	52
Configurar el compte de gmail amb gaim.....	52
Compartir arxius amb NFS (Entre màquinesGNU/Linux).....	52
Canviar la imatge del splash.....	53
Posar una imatge de fons en el menú del grub.....	53
Configuració d'un mòdem 56k.....	53
Configuració de la targeta de xarxa.....	53
Com instal·lar més tipus de lletra.....	55
Com instal·lar les Microsoft TrueType Fonts.....	55
Com mostrar una imatge quan s'inicia el sistema.....	56
Instal·lació de VirtualBox OSE (Open Source Edition).....	57
5. Aplicacions.....	58