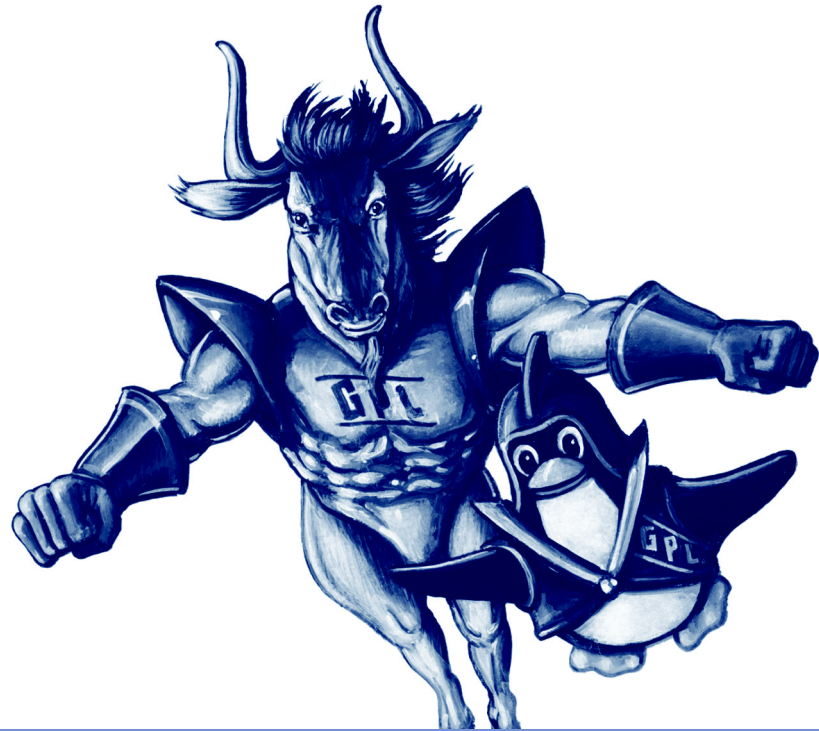


Programari lliure

Francesc Aulí Llinàs
Roger Baig Viñas

71Z799C02MO



Sistema operatiu GNU/Linux bàsic

David Megías Jiménez

Coordinador

Enginyer en Informàtica per la Universitat Autònoma de Barcelona. Màster en Tècniques Avançades d'Automatització de Processos per la Universitat Autònoma de Barcelona.

Doctor en Informàtica per la Universitat Autònoma de Barcelona. Professor dels Estudis d'Informàtica i Multimèdia de la Universitat Oberta de Catalunya.

Jordi Mas

Coordinador

Coordinador general de Softcatalà i desenvolupador del processador de textos lliures Abiword.

Membre fundador de Softcatalà i de la xarxa telemàtica RedBBS.

En qualitat de consultor, ha treballat en empreses com Menta, Telépolis, Vodafone, Lotus, eresMas, Amena i Terra España.

Francesc Aulí Llinàs

Autor

Enginyer en Informàtica per la Universitat Autònoma de Barcelona. Premi extraordinari. Concessió d'una beca FPI de la Generalitat de Catalunya.

Roger Baig i Viñas

Autor

Enginyer Tècnic Superior Industrial per la Universitat Politècnica de Catalunya i enginyer en Electrònica i Automàtica Industrial per la Universitat Politècnica de Catalunya. Professor associat del Departament de Telecomunicacions i Enginyeria de Sistemes de la Universitat Autònoma de Barcelona.

Primera edició: maig de 2005
© Fundació per a la Universitat Oberta de Catalunya. Av. Tibidabo, 39-43, 08035 Barcelona
Material realitzat per Eureka Media, SL
© Autors: Roger Baig i Viñas i Francesc Aulí Llinàs.
Dipòsit legal: B-15.564-2005
ISBN: 84-9788-266-0

Es garanteix permís per a copiar, distribuir i modificar aquest document segons els termes de la *GNU Free Documentation License, Version 1.2* o qualsevol de posterior publicada per la Free Software Foundation, sense seccions invariants ni texts de coberta anterior o posterior. Es disposa d'una còpia de la llicència en l'apartat "GNU Free Documentation License" d'aquest document.

Índex

Agraïments	9
Introducció	11
1. Presentació	13
1.1. Què és el GNU?	13
1.2. Què és el GNU/Linux?	16
1.3. Distribucions	18
1.4. Programes i documentació	20
2. Conceptes i comandaments bàsics	25
2.1. Introducció	25
2.2. Usuaris i grups	26
2.2.1. La jerarquia del sistema de fitxers	32
2.3. El sistema de fitxers	32
2.3.1. Directoris del sistema	33
2.3.2. Movent-nos	34
2.3.3. Enllaços	35
2.3.4. Permisos	37
2.3.5. Manipulació, patrons i recerques	39
2.3.6. Tipus i contingut de fitxers	41
2.4. Els processos	42
2.4.1. L'ajuda del sistema	46
2.5. Altres comandaments útils	46
2.5.1. Empaquetatge i compressió	47
2.5.2. Operacions de disc	48
2.5.3. Redireccionaments	51
2.6. Operacions amb comandaments	51
2.6.1. Comandaments específics del bash	53
2.6.2. <i>Shell scripts</i> amb bash	55
3. Taller de KNOPPIX	59
3.1. Introducció	59
3.2. Arrencada del sistema	60
3.3. Aturada del sistema	62

3.4. Configuració del teclat	62
3.5. Inspecció del sistema	63
3.6. Maneig de directoris i fitxers	68
3.7. Administració d'usuaris	74
3.8. Gestió de processos	78
3.9. Activació i ús del ratolí	80
3.10. Altres operacions	82
3.11. Conclusió	83
4. Instal·lació de GNU/Linux	85
4.1. Introducció	85
4.2. Arrencant	85
4.3. Particionant el disc	86
4.4. Instal·lació de mòduls	89
4.5. Configuració bàsica de la xarxa	90
4.6. Sistema d'arrencada	91
4.7. Elecció de paquets	91
4.8. Altres aspectes	92
5. Taller d'instal·lació de Debian Woody	95
5.1. Introducció	95
5.1.1. Sistemes d'instal·lació	96
5.1.2. Tipus de paquets	98
5.1.3. Estat de desenvolupament dels paquets	98
5.1.4. Sabors o <i>flavours</i> de Debian Woody	99
5.2. Instal·lació de Debian Woody	99
5.2.1. CD-ROM de Debian Woody i els seus diferents <i>flavours</i>	100
5.2.2. Installing Debian GNU/Linux 3.0 For Intel x86	101
5.3. Instal·lació de Debian Woody mitjançant CD-ROM	101
5.3.1. Abans de començar la instal·lació	101
5.3.2. Arrencada del sistema d'instal·lació	102
5.3.3. Configuració de l'idioma d'instal·lació	105
5.3.4. Menú principal d'instal·lació	105
5.3.5. Configuració del teclat	105
5.3.6. Partició del disc dur	106
5.3.7. Inicialització i activació de la partició <i>swap</i>	108
5.3.8. Inicialització i activació d'una partició Linux	108
5.3.9. Inicialització i activació d'altres particions Linux	109
5.3.10. Instal·lació del nucli	109

5.3.11. Configuració de mòduls	110
5.3.12. Configuració del <i>hostname</i>	110
5.3.13. Instal·lació del sistema base	111
5.3.14. Creació d'un disc d'arrencada	111
5.3.15. Instal·lació de LILO	111
5.3.16. Reinicialització del sistema	112
5.3.17. Arrencada del sistema base	112
5.3.18. Configuració horària	113
5.3.19. Configuració geogràfica	113
5.3.20. Establiment de la política de contrasenyes ..	113
5.3.21. Últimes configuracions	114
5.3.22. Configuració d'apt	115
5.3.23. <i>tasksel</i> i <i>dselect</i>	115
5.3.24. Particularitats d'una instal·lació per xarxa	117
5.4. Instal·lació de Debian Woody per xarxa	117
5.4.1. Aspectes comuns dels diferents mètodes d'instal·lació	118
5.4.2. Instal·lació del mòdul de xarxa	118
5.4.3. Configuració de la xarxa	120
5.4.4. Configuració d'apt	120
5.5. Conclusió	121
6. Configuracions bàsiques	123
6.1. El sistema de connexió	123
6.2. Explorant el <i>bash</i>	124
6.3. El sistema d'arrencada	126
6.3.1. LILO	128
6.3.2. GRUB	132
6.4. Accés a altres particions i dispositius	134
6.5. Configuració de dispositius	137
6.5.1. El teclat	138
6.5.2. Targeta de xarxa (tipus Ethernet)	139
6.5.3. Targeta WiFi	142
6.5.4. Mòdems	143
6.5.5. Targeta de so	144
6.5.6. Impressora	144
7. DAEMONS i <i>runlevels</i>	147
7.1. Els DAEMONS	147
7.2. Els <i>runlevels</i>	150
7.2.1. Logs de sistema (<i>syslogd</i>)	154
7.3. L'arrencada del sistema	154
7.4. DAEMONS bàsics	154
7.4.1. Logs de sistema (<i>syslogd</i>)	
7.4.2. Execucions periòdiques (<i>cron</i>)	157
7.4.3. Execucions retardades (<i>at</i> i <i>batch</i>)	159

8. Instal·lació d'aplicacions	161
8.1. Introducció	161
8.2. El sistema de paquets Debian	162
8.3. Compilació de programes nous	166
9. Taller de configuracions bàsiques	171
9.1. Introducció	171
9.2. El gestor d'arrencada	171
9.2.1. Instal·lació de LILO	172
9.2.2. Instal·lació de GRUB	173
9.3. El sistema de paquets	176
9.3.1. /etc/apt/sources.list	177
9.3.2. apt	179
9.3.3. dpkg	183
9.3.4. dselect	184
9.3.5. aptitude	184
9.4. Configuració regional: locales.....	184
9.5. Configuració de man i el paginador.....	186
9.6. L'arxiu principal d'arrencada, /etc/inittab ..	187
9.7. Muntatge de dispositius, /etc/fstab	187
9.8. Configuració de dispositius	189
9.8.1. Configuració del ratolí	190
9.8.2. Configuració de mòdems	192
9.8.3. Configuració de mòdems DSL	195
9.8.4. Configuració de targetes de xarxa	195
9.8.5. Configuració d'impressores	198
9.8.6. Configuració de targetes de so	200
9.9. Conclusió	200
10. Arquitectura X-Window	203
10.1. Què és X-Window?	203
10.2. Configuració	208
10.3. <i>X display manager</i>	212
11. Taller d'X-Window	215
11.1. Introducció	215
11.1.1. Diferents estratègies per a instal·lar els paquets	216
11.1.2. Instal·lació de paquets bàsics	216
11.2. Instal·lació del sistema bàsic	216
11.2.1. Inicialització del servidor	219
11.2.2. El fitxer de log	221
11.2.3. El servidor de fonts	221
11.3. <i>Window managers</i>	222
11.4. <i>X Session manager</i>	224

11.5. X Display manager	224
11.6. Desktop managers	225
11.6.1. GNOME	226
11.6.2. KDE	228
11.6.3. Personalització d'aspectes locals	229
11.7. Personalització d'alguns aspectes	229
11.7.1. Personalització d'aspectes de xarxa	231
11.8. Configuració d'impressores	232
11.9. OpenOffice	233
11.10. Conclusió	234
A. Taules de comandaments	237
A.1. Sistema de fitxers	237
A.2. Ajuda del sistema	237
A.3. Permisos dels fitxers	238
A.4. Còpia i esborrament de fitxers	238
A.5. Parada o reinici	238
A.6. Operacions amb fitxers	239
A.7. Compresió de fitxers i còpies de seguretat	240
A.8. Operacions de disc	240
A.9. Usuaris i grups	241
A.10. Gestió de processos	241
B. L'editor vi	243
B.1. Introducció	243
B.2. Modes del vi	243
C. Eines d'administració	247
C.1. Introducció	247
C.2. Linuxconf	249
C.3. Webmin	251
GNU Free Documentation License	254

Agraïments

Els autors agraeixen a la Fundació per a la Universitat Oberta de Catalunya (<http://www.uoc.edu>) el finançament de la primera edició d'aquesta obra, emmarcada en el Màster Internacional de Programari Lliure ofert per aquesta institució.

Introducció

Encara que ja fa més de vint anys que el programari lliure existeix, fins als darrers no s'ha perfilat com una alternativa vàlida per a molts usuaris, empreses i, cada vegada més, també, per a institucions i governs. Actualment, GNU/Linux és un dels sistemes operatius més fiables i eficients que podem trobar. Encara que la seva naturalesa de programari lliure va crear inicialment certes reticències en usuaris i empreses, GNU/Linux ha demostrat que està a l'altura de qualsevol altre sistema operatiu existent.

L'objectiu d'aquest curs és iniciar-nos en el món del GNU/Linux. Hi obtindrem les claus per entendre la filosofia del codi lliure, aprendrem a usar-lo i manipular-lo al nostre gust i entendrem com funcionen de les eines necessàries per a poder moure'ns fàcilment en aquest nou món. Cal tenir present, però, que aquest document no pretén ser un manual de referència imprescindible per a administradors i/o usuaris; per a això ja existeixen centenars de manuals, HOWTO i multitud d'altres referències que ocupen milers de pàgines. Aquí pretenem aprendre a fer els primers passos en aquesta terra poc explorada encara per a massa usuaris i administradors, alhora que ensenyarem com plantejar i resoldre per nosaltres mateixos els problemes que puguin aparèixer.

El curs no pretén basar-se en cap distribució en particular, però en la majoria d'exemples i activitats és necessari concretar específicament algunes accions, per a les quals s'utilitzarà Debian GNU/Linux (versió 3.0 –Woody–). Encara que no és una distribució tan intuïtiva i fàcil d'utilitzar com d'altres, ens servirà per a explicar pas a pas totes les característiques d'un sistema operatiu basat en GNU/Linux. A més, la seva extraordinària qualitat, estabilitat i seguretat la fan una de les opcions actualment més vàlides. D'altra banda, tampoc no hem d'oblidar el suport (Debian està desenvolupada per voluntaris i no dóna cap classe de suport) que es dóna en altres distribucions i que en el cas de moltes empreses és imprescindible. Per aquesta raó, hem inclòs un apèndix en el qual mostrem el procés d'instal·lació i les principals característiques de RedHat Linux (versió 9.0).

Esperem que el curs sigui del vostre grat i serveixi per a obrir-vos les portes del món del programari lliure. Com més usuaris siguem, més programari i de millor qualitat tindrem.

Benvinguts al GNU/Linux!

1. Presentació

1.1. Què és el GNU?

Per a entendre tot el moviment del programari lliure, ens hem de situar a la darrera de la dècada dels seixanta i a la primera dels setanta. En aquells temps les grans companyies d'ordinadors no donaven el valor que avui dia es dona al programari. En la seva gran majoria eren els mateixos fabricants d'ordinadors, els quals obtenien els seus principals ingressos venent les seves grans màquines, els que incorporaven algun tipus de sistema operatiu i aplicacions a aquestes sense donar-los cap valor. Les universitats tenien permís per a agafar i estudiar el codi font del sistema operatiu per a finalitats docents. Els mateixos usuaris podien demanar el codi font dels programes de control i de les aplicacions per adaptar-los a les seves necessitats. Es considerava que el programari no tenia valor per si mateix si no s'acompanyava del maquinari sobre el qual s'implementava. En aquest entorn, els laboratoris Bell (AT&T) van dissenyar un sistema operatiu anomenat UNIX, caracteritzat per la bona gestió dels recursos del sistema, la seva estabilitat i la seva compatibilitat amb el maquinari de diferents fabricants (per a homogeneïtzar tots els seus sistemes). Aquest últim fet va ser importantíssim (fins llavors tots els fabricants tenien els seus propis sistemes operatius incompatibles amb els altres), ja que esdevingué el factor que li va proporcionar una popularitat extraordinària.

A poc a poc, les grans empreses van començar a prendre consciència del valor que tenia el programari per si mateix: primer va ser IBM la que el 1965 va deixar de donar el codi font del seu sistema operatiu, a la darrera dels setanta, Digital Research va començar a vendre el seu, etc. Aquest fet va fer que totes les companyies s'adonessin que el programari podia ser molt rendible i els podia aportar grans beneficis. A partir d'aquest fet, la majoria d'empreses van començar a posar reticències a deixar el codi font dels seus programes i sistemes operatius i van començar a vendre els seus programes com un valor afegit al seu maquinari. En aquest entorn cada vegada més tant, Richard Stallman (que treballava en el MIT, Institut de Tecnologia de

Nota

El mateix Stallman explica com a anècdota com es va enfadar en descobrir que la companyia que els havia venut una nova impressora per al laboratori on treballava no li volia facilitar el codi font dels programes de control (*drivers*). Ell només els volia modificar perquè la impressora l'avisés automàticament quan s'encallava el paper! La companyia es va negar a proporcionar-los-hi.

Contingut complementari

El nom que va donar al projecte significa GNU, Not UNIX, afegint-se a la moda dels noms/bromes recursives d'aquell temps.

Massachusetts o Massachusetts Institute of Technology en anglès) es va indignar en comprovar que cada vegada era més difícil aconseguir el codi font dels programes que utilitzava per a adaptar-los a les seves necessitats, tal com havia fet fins aleshores.

A partir d'aquell moment, Stallman va decidir, essent conseqüent amb els seus ideals, iniciar un gran projecte per intentar obrir una altra vegada el codi font dels programes. Conscient que no podria aconseguir que les companyies cedissin en aquest punt, es va proposar crear el seu propi sistema operatiu i les aplicacions tot iniciant un projecte anomenat GNU.

És molt interessant per a entendre els motius que van portar Stallman a iniciar GNU el seu primer manifest, document on va explicar a tota la comunitat en què consistiria el projecte, com l'orientaria i per què l'havia de fer. Hi va començar a descriure el concepte de programari lliure i per a què creia necessari que programadors i desenvolupadors de tot el món hi contribuïssin. Encara que moltes vegades es confon el concepte de programari lliure amb el de programari gratuït (en anglès, free té els dos significats), en documents posteriors s'ha deixat molt clar que el programari lliure no té per què ser gratuït. Hem d'entendre com a programari lliure programes dels quals podem aconseguir el codi font, estudiar-lo, modificar-lo i redistribuir-lo sense que ens obliguin a pagar per això. El que hem de tenir clar és que sí que podem demanar els diners que vulguem pels programes i el seu codi font, el suport que podem oferir als usuaris, els llibres que venguem o el material que proporcionem, tal com fan moltes companyies que distribueixen GNU/Linux. Tanmateix, en cap moment no podem obligar que els usuaris no distribueixin el programari que els hem venut. Aquest ha de poder ser distribuït lliurement. És una manera diferent d'entendre el programari de la manera a què estem acostumats. En molts dels textos de l'FSF (Free Software Foundation) es parla més de filosofia que d'enginyeria. Hem d'entendre tot aquest moviment més com una manera de pensar o fer les coses que com una companyia més de programari.

La filosofia que en l'FSF es té del programari es defineix amb les quatre llibertats següents:

- La llibertat 0 es refereix a la llibertat de poder usar el programa per a qualsevol propòsit.

- La llibertat 1 és la que permet estudiar com funciona el programa i adaptar-lo a les pròpies necessitats. L'accés al codi font és una condició necessària per a garantir aquesta llibertat.
- La segona llibertat és la que permet distribuir lliurement còpies del programari, amb la qual cosa s'ajuda el veí.
- L'última llibertat és la que permet millorar el programa i fer públiques les pròpies millores, en benefici de tota la comunitat. L'accés al codi font, per tant, és un requisit imprescindible per a assegurar aquesta llibertat.

Per a donar totes aquestes llibertats al programari que es desenvolupava en el projecte i als seus usuaris finals es va escriure la llicència, amb la qual s'ha protegit tot aquest tipus de codi, la GPL (*General Public License*). Aquesta llicència posa per escrit les idees comentades anteriorment.

El projecte va començar a produir programari a partir del 1984, començant amb el desenvolupament de totes les eines necessàries per a poder implementar un sistema operatiu complet. Encara que fer un projecte d'aquestes característiques sigui un procés llarg i complex, des del principi molts programadors i desenvolupadors de programari es van sentir captivats per la idea de Stallman i hi van començar a col·laborar de manera massiva gratuïtament. La comunitat no va parar de créixer, i a poc a poc es va començar a disposar de les eines necessàries (editors, compiladors, etc.) per a implementar el nucli del sistema operatiu, que era la tasca que requeria les eines que es desenvolupaven. Des del primer moment es va voler crear un sistema operatiu semblant a UNIX i seguint les normes POSIX (*portable operating system interface*). Si bé UNIX també tenia problemes i mancances, era, i continua essent, prou bo per a adaptar-se a la majoria de les necessitats. La tasca de dissenyar i escriure el nucli del sistema operatiu va ser la que es va deixar per al final del procés. Encara actualment s'ha d'acabar definitivament i el nucli del GNU, anomenat Hurd, continua en fase de desenvolupament.

Activitats

1. Llegiu el primer missatge escrit per Stallman el 1983 en què anunciava el seu projecte (traduït al castellà): <http://www.gnu.org/gnu/initial-announcement.es.html>.

Nota

Podeu trobar una còpia de la GPL en anglès a l'adreça següent:

<http://www.gnu.org/licenses/gpl.html>.

Podeu trobar una traducció no oficial al català de la a GPL a l'adreça següent:

<http://ca.dodds.net/gnu/gpl.ca.html>.

Contingut complementari

Com el seu nom indica, el nucli (*kernel*) d'un sistema operatiu és el cor amb què pot funcionar. És el nucli, el programari que gestiona els recursos de l'ordinador: es comunica amb els dispositius i les aplicacions instal·lades, administra la memòria adequadament, reparteix temps de processament entre tots els programes, es comunica amb els dispositius d'emmagatzemament per desar els arxius, etc.

Contingut complementari

La tecnologia microkernel es basa a dividir les diferents funcionalitats del nucli d'un sistema operatiu en programes totalment separats i que es comuniquin entre si. Això el fa molt modular, facilita molt el test, la detecció i la correcció d'errors, el manteniment, etc. Actualment, alguns sistemes operatius com Amoeba, Chorus, Mach o Windows NT™ han incorporat aquest tipus de tecnologia.

2. Llegiu "El manifest GNU" original de Stallman (traduït al castellà): <http://www.gnu.org/gnu/manifesto.es.html>.
3. Llegiu la (General Public License): <http://www.gnu.org/licenses/gpl.html>.

1.2. Què és el GNU/Linux?

En aquest context, i quan l'FSF encara no tenia cap nucli estable per al seu sistema operatiu, un professor de la Universitat d'Holanda, Andrew Tanenbaum, va decidir escriure un sistema operatiu perquè els seus estudiants el poguessin estudiar. Igual que Stallman, fins llavors havia pogut utilitzar el codi font de UNIX d'AT&T perquè els seus alumnes aprenguessin a dissenyar sistemes operatius. La seva idea era escriure un sistema operatiu que el pogués estudiar i modificar qualsevol. El 1987 va posar fil a l'agulla i va anomenar el seu projecte mini UNIX, creant Minix. En no utilitzar ni una sola línia de codi de UNIX d'AT&T, no hi ha cap restricció per a agafar el codi, utilitzar-lo i modificar-lo lliurement.

Tanenbaum va voler crear un sistema orientat a finalitats docents, per la qual cosa el va dissenyar utilitzant una arquitectura microkernel, ideal per a una comprensió fàcil i aportant una tecnologia molt nova per a l'època que li permetia versatilitat, multiplataforma, etc. Aquest ha estat un dels punts forts i dèbils alhora del Minix: encara que el sistema sigui una petita joia per al seu estudi i disseny, és molt probable que mai no es pugui utilitzar en entorns reals. Es va optar perquè fos entenedor, modular i molt pedagògic, però no ràpid. De tota manera, Tanenbaum tampoc no pretenia això; al llarg dels anys Minix ha evolucionat i realment avui en dia encara continua existint i és estudiat per molts alumnes d'universitats d'arreu del món.

Aquí és quan entra en joc Linux. Mentre l'FSF seguia proporcionant eines per a construir un sistema operatiu amb el seu gran projecte, Tanenbaum orientava Minix a finalitats docents i moltes empreses continuaven fent evolucionar les seves pròpies versions de UNIX. Linus Torvalds, estudiant de la Universitat de Hèlsinki, decideix crear

l'agost del 1991 el seu propi nucli per a un sistema operatiu nou, Linux. La seva idea era crear un UNIX per a PC perquè tots els que volguessin el poguessin utilitzar al seu ordinador. La primera aparició en escena que va fer va ser en un debat sobre Minix i sistemes operatius, en què va exposar les idees següents:

```

Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Date: 25 Aug. 91 20:57:08 GMT
Organization: University of Helsinki
Hello everybody out there using minix.
I'm doing a (free) operating system (just a hobby,
won't be big and professional like gnu) for 386(486)
AT clones. This has been brewing since april, and
is starting to get ready. I'd like any feedback on
things people like/dislike in minix, as my OS res-
embles it somewhat (same physical layout of the
file-system (due to practical reasons) among other
things).
I've currently ported bash(1.08) and gcc(1.40), and
things seem to work.
This implies that I'll get something practical
within a few months, and I'd like to know what fea-
tures most people would want. Any suggestions are
welcome, but I won't promise I'll implement them :-)
```

Si accedíssim al fòrum de debat on va aparèixer aquest primer missatge, veuríem que de seguida gent de tot el món es va començar a interessar per aquest sistema nou, que en utilitzar el compilador i l'interpret de comandaments de GNU (gcc i bash) com a peces fonamentals, també tenia les característiques de programari lliure. Encara que en paraules del mateix Torvalds, si ell hagués sabut la quantitat de feina necessària per a aconseguir que la seva idea funcionés, mai no ho hauria fet: esforços de molts experts en informàtica de tot el món van fer possible aquest projecte.

De fet, en els primers anys de la seva existència, GNU/Linux s'identificava com el sistema operatiu dels hackers. Que fos difícil d'instal·lar i manipular, i la falta de programadors de control el feien una eina apta únicament per a gent molt entesa en el tema. Van ser aquests primers usuaris els que van dissenyar els programadors de control per als discos, les impressores, les targetes, etc., i els que van començar a fer co-

Contingut complementari

Linux, el nucli de GNU/Linux, és de tipus monolític. Això indica que no se separen les seves diverses funcionalitats en mòduls diferents, sinó que totes formen part d'un mateix programa. El principal inconvenient d'aquest tipus de disseny és que la localització d'errors i el seu manteniment són molt costosos. En contrapartida, el rendiment que s'aconsegueix és molt més gran que en altres tipus de disseny.

nèixer al món aquest sistema. A poc a poc, el nombre d'usuaris va començar a créixer i actualment ja hi ha moltes empreses i grups d'usuaris que creen les seves pròpies distribucions de GNU/Linux.

1.3. Distribucions

Actualment, existeixen moltes distribucions diferents basades en GNU/Linux. N'hi ha per a tota classe d'ordinadors i dispositius electrònics: ordinadors portàtils o de sobretaula, ordinador de butxaca (*pocketPC*) o PDA, punts d'accés de xarxes sense fil (o *wireless*), etc. La naturalesa del programari lliure permet això: qualsevol pot agafar el codi desenvolupat fins llavors i adaptar-lo a les seves pròpies necessitats. És un fet que, cada vegada més, empreses i usuaris trien sistemes basats en GNU/Linux per les seves elevades prestacions i la quantitat de programari disponible.

De tota manera, encara que hi hagi desenes de distribucions, n'hi ha de més populars que s'han estès molt. La filosofia del programari lliure fa que moltes empreses que han creat les seves pròpies distribucions de GNU/Linux no restringeixin l'accés al seu codi. Tot i així, el suport que ofereixen i el material que venen els aporta beneficis, que en permeten la subsistència. Així mateix, es pot considerar que en moltes d'aquestes distribucions s'inclou programari propietari que alguns usuaris prefereixen, si bé en molts casos hi ha programes homòlegs amb llicència Free Software.

A continuació farem una breu descripció d'algunes de les distribucions de GNU/Linux:

- Slackware: una de les primeres distribucions que van aparèixer. Va ser creada per Patrick Volkerding i va tenir un gran èxit en els seus primers anys d'existència.



- Debian GNU/Linux: una de les primeres distribucions de GNU/Linux que van aparèixer i encara continuen existint i evolucionat.

Contingut complementari

Encara que moltes distribucions de GNU/Linux es denominin només Linux, és important que diferenciem que realment Linux és el nucli del sistema operatiu i que el projecte GNU és el que realment hi ha aportat molta estructura perquè funcioni.

El sistema de paquets ens permet diferenciar clarament el programari lliure del que no ho és, i d'aquesta manera disposar de tot el sistema només amb programes de llicència Free Software. La desenvolupen un grup de col·laboradors distribuïts per tot el món i no té el suport de cap empresa. Encara que és de les més estables i segures que hi ha, el seu sistema d'instal·lació i configuració necessita coneixements previs.



- RedHat Linux: juntament amb SuSE, és una de les distribucions més populars. L'ha creat una empresa dels Estats Units i aporta programari de gran qualitat. Té un entorn molt intuïtiu que en facilita molt la instal·lació i configuració.



- SuSE Linux: encara que és una distribució creada bastant recentment, ha tingut una gran difusió. La va desenvolupar una empresa alemanya, aportant molt programari propietari de qualitat. És molt completa i fàcil d'instal·lar i mantenir, encara que en alguns aspectes no se segueixen alguns dels estàndards de la comunitat.



- KNOPPIX: distribució en un CD-live basada en Debian. Detecta automàticament tot tipus de maquinari i aporta l'últim escritori de KDE i la suite OpenOffice.org. Molt útil per a demostracions i usuaris novells en el sistema.



Tampoc no podem oblidar que hi ha altres sistemes operatius compatibles amb UNIX i els estàndards que se segueixen actualment. Molts dels conceptes i eines que veurem al llarg del curs també serviràn per a aquests altres. Especialment hem de destacar GNU/Hurd (nucli desenvolupat pel projecte GNU) i FreeBSD.

Activitat

4. Llegiu la descripció d'algunes de les distribucions actuals basades en GNU/Linux:
<http://www.linuxhq.com/dist.html>.

1.4. Programes i documentació

Internet ha estat sempre el principal mitjà de comunicació entre els desenvolupadors i usuaris del programari lliure. És per aquesta raó pel que ja des del principi de la gran expansió de GNU/Linux s'ha pogut trobar a la Xarxa molta informació sobre l'operatiu. La majoria dels programes els podem baixar d'Internet, empaquetats amb algun dels sistemes més comuns o bé directament a partir del seu codi font perquè el puguem compilar en el nostre sistema. A més, la majoria de les distribucions també es poden baixar de la Xarxa sense necessitat de comprar cap paquet especial de les revistes especialitzades o de les mateixes empreses que el produeixen. També és cert que si volem el suport que ofereixen algunes de les distribucions, el millor és comprar tot el material que es proporciona (CD, manuals, etc.) i registrar-s'hi.

A mesura que ens introduïm en el món del programari lliure i del GNU/Linux, veurem que un dels aspectes clau per a moure-s'hi és saber trobar la documentació que ens interessa. Quan ens trobem davant d'un problema, abans de començar a pensar com resoldre'l, hem de pensar que és molt probable que molta altra gent com nosaltres també s'hagi trobat amb la mateixa dificultat o amb una de similar. Buscar i trobar la documentació que s'adapti millor als problemes que se'ns plantegin ens estalviarà molt de temps i esforç. La comunitat del programari lliure genera centenars de documents que ens podem baixar lliurement d'Internet, a més dels fòrums de discussió, pàgines de rumors i notícies, etc.

Algunes de les referències més populars, i que més ens poden ajudar són:

- **Documentació**

<http://www.tldp.org>: The Linux Documentation Project. La majoria de guies, HOWTO, PMF o FAQ, etc. existents les podem trobar en aquest lloc, que a més està en diversos idiomes.

<http://lucas.linux.org.mx>: LinUx en CASTellano. Gran projecte de documentació en castellà per als HOWT, guies, etc. de GNU/Linux.

<http://www.linuxpowered.com/HOWTO/HOWTO-INDEX>: El HOWTO dels HOWTO.

<http://www.linux.com>: pàgina amb diferents seccions de notícies, documentació, etc.

<http://www.debian.org/doc>: documentació per a Debian GNU/Linux.

- **Notícies**

<http://slashdot.com>: notícies i rumors del món GNU/Linux. En anglès.

<http://barrapunto.com>: la rèplica de slashdot en castellà.

<http://puntbarra.com>: La rèplica de slashdot en català.

<http://bulmalug.net>: usuaris novells de Linux de Mallorca i els voltants. Notícies i seccions dedicades a temes concrets.

<http://www.es.gnu.org/gnuticias>: notícies de GNU en espanyol.

<http://linuxtoday.com>: una altra pàgina de notícies molt pràctica per a estar al corrent d'aquest tema.

<http://libertonia.escomposlinux.org>: pàgina de notícies. És especialment interessant la seva secció de "Fuentes de Noticias", on hi ha multitud d'altres enllaços a altres pàgines del mateix estil.

- **Fòrums**

<http://www.foroslinux.org>: diversos fòrums de GNU/Linux dedicats a tot tipus de temes.

<http://www.linuxsecurity.com/resources/forums-1.html>: fòrums centrats en temes de seguretat i similars.

- **Recerca**

<http://www.google.com/linux>: el cercador més gran del món també per a GNU/Linux.

<http://www.buscadoc.org>: cercador de documentació informàtica en castellà.

- **Distribucions**

<http://www.fsf.org>: la pàgina oficial de la Free Software Foundation.

<http://www.debian.org>: pàgina oficial de Debian GNU/Linux.

<http://www.redhat.com>: pàgina oficial de RedHat Linux.

<http://www.suse.com>: pàgina oficial de SuSE.

<http://www.slackware.com>: pàgina oficial de Slackware Linux.

<http://www.knoppix.com>: pàgina oficial de KNOPPIX.

- **Baixades**

<http://sourceforge.net>: la pàgina més gran amb projectes de programari lliure.

<http://www.softonic.com/index.phtml?n id=4>: secció de baixada per a GNU/Linux d'una de les múltiples pàgines de downloading.

<http://download.com>: pàgina de baixades.

- **D'altres**

<http://www.linuxsecurity.com>: pàgina molt actual centrada en tot tipus de temes de seguretat en GNU/Linux.

<http://www.linuxhq.com>: informació general sobre distribucions de GNU/Linux, seguretat, etc.

<http://www.linuxjournal.org>: pàgina de notícies i articles sobre GNU/Linux.

<http://www.linuxgazette.com>: revista de GNU/Linux.

<http://www.linux-mag.com>: revista de GNU/Linux.

<http://www.xfree86.org>: pàgina oficial del projecte XFree86.

<http://www.x.org/>: pàgina principal del projecte XOrg.

2. Conceptes i comandaments bàsics

2.1. Introducció

En aquest capítol aprendrem les idees i instruccions bàsiques per a moure'ns adequadament pel sistema. Si no estem acostumats a utilitzar la línia de comandaments per a manipular el sistema operatiu, al principi ens pot semblar una mica complicat, però a mesura que la utilitzem veurem que és molt útil i ens permet realitzar qualsevol tasca que vulguem fer. A més, el fet de saber utilitzar correctament els comandaments ens serà molt pràctic quan necessitem connectar-nos de manera remota a una màquina i podrem programar, així mateix, petits programes (*shell scripts*) per automatitzar les tasques d'administració més comunes.

La majoria dels comandaments que veurem en aquest capítol formen part de l'estàndard (normes IEEE POSIX) i són comuns a tots els sistemes GNU/Linux i a UNIX. Encara que cada distribució té les seves pròpies aplicacions d'administració i gestió, moltes de les accions que es fan a partir d'aquelles també es poden fer amb els comandaments que veurem. A partir dels comandaments, podrem manipular gairebé tots els aspectes del sistema i moure'ns-hi eficientment. Aprenent a utilitzar correctament aquests comandaments, aprendrem a navegar per qualsevol sistema basat en GNU/Linux, sense que importi quina distribució sigui.

Cada un dels comandaments del sistema sol tenir una multitud de paràmetres diferents. Amb la utilització dels paràmetres podem, amb un mateix comandament, fer moltes accions diferents, encara que totes siguin d'un mateix estil. En aquest document no especificarem els diferents paràmetres de cada un dels comandaments que veurem, ja que estendríem el text més enllà del permisible i tampoc no té sentit conèixer exactament la totalitat dels paràmetres possibles per a cada un. Tots disposen d'un ampli manual, on s'especifiquen totes les seves opcions, de manera que sempre que necessitem fer alguna acció en concret hi podrem recórrer. Als tallers distribuïts al llarg del curs

Contingut complementari

Un comandament és un programa que fa una acció determinada relacionada amb el sistema operatiu.

Contingut complementari

Un paràmetre no és més que una opció determinada d'un comandament, que hi afegim a continuació, precedit per un espai i, moltes vegades, per un guió. Per exemple, si un comandament fos fer una llista, podríem passar-li un paràmetre com "listar tot".

Nota

Una política de noms molt utilitzada sol ser posar com a *login* la primera inicial del nom de l'usuari seguit del seu cognom.

Contingut complementari

NIS són una sèrie d'aplicacions que ens permeten gestionar tots els usuaris d'un mateix laboratori centralitzadament en un sol servidor.

sí que veurem algunes d'aquestes opcions, encara que és important saber que amb el manual sempre en podrem descobrir moltes altres, que ens poden ajudar a fer tot el que necessitem.

2.2. Usuaris i grups

Actualment, la majoria dels sistemes operatius existents són multiusuari i multitasca. Això implica que més d'un usuari pot treballar en el sistema simultàniament a d'altres, executant una o més tasques a l'hora. Per aquest motiu, és molt important que el mateix sistema operatiu incorpori mecanismes per a manipular i controlar correctament els usuaris: el sistema d'entrada i identificació (*login*), els programes que pot executar cada un, mecanismes de seguretat per a protegir el maquinari de l'ordinador, protecció per als fitxers dels usuaris, etc.

Els sistemes operatius basats en UNIX organitzen tota aquesta informació per usuaris i grups. En entrar en el sistema, ens hem d'identificar amb un inici de sessió (*login*) i una contrasenya. L'inici de sessió sol ser un nom que identifica de manera inequívoca l'**usuari**. En sistemes on hi ha més d'uns quants usuaris, és important disposar d'una bona política de noms per a poder identificar-los tots de manera clara. La contrasenya ha de ser una combinació de lletres, nombres i caràcters especials. No ha d'estar formada per cap paraula de diccionari o similars perquè pot representar un problema de seguretat important. El sistema de contrasenyes és de tipus unidireccional. Això vol dir que la nostra contrasenya no és emmagatzemada com a text, sinó que és xifrada i desada. Quan entrem en el sistema i escrivim la nostra contrasenya, aquesta és xifrada i és comparada amb la que està emmagatzemada. Si coincideixen, la identificació és positiva, si no coincideixen, no. L'important de tot aquest sistema és que a partir del xifratge no podem aconseguir, de cap manera, la clau original. Els programes que intenten trencar les contrasenyes dels usuaris l'únic que poden fer és xifrar paraules a partir de diccionaris (amb sistemes automàtics per a derivar-les i buscar variants) i provar si coincideixen amb el xifratge d'alguna de les contrasenyes d'usuari. És per aquest motiu que hem d'escollir acuradament les nostres contrasenyes; si no ho fem, comprometrem tota la seguretat del sistema.

Actualment, en els sistemes GNU/Linux podem escollir dos tipus de xifratge possibles per a les contrasenyes d'usuari. El que s'usa des dels inicis de UNIX és el 3DES. L'únic inconvenient d'aquest tipus de xifratge és que només ens permet contrasenyes de vuit lletres (si n'escriuim més, s'ignoren), a diferència de l'altre tipus de xifratge, anomenat MD5, amb què podem usar contrasenyes de la longitud que vulguem (de fet, MD5 és un sistema de hashing, però també es pot utilitzar per a xifrar contrasenyes unidireccionalment). Com més llarga sigui la contrasenya, més segura serà, fet pel qual es recomana utilitzar el segon tipus de xifratge. De tota manera hem de considerar que, si necessitem usar alguns programes especials per a la gestió d'usuaris, com el NIS, pot ser que no siguin compatibles amb MD5.

Si bé un usuari és un individu particular que pot entrar al sistema, un grup és un conjunt d'usuaris amb accés al sistema que comparteixen unes mateixes característiques, de manera que ens és útil agrupar-los per a poder donar-los una sèrie de permisos especials al sistema. Un usuari ha de pertànyer, almenys, a un grup, encara que pot ser de més d'un. El sistema també utilitza tot aquest mecanisme d'usuaris i grups per a gestionar els servidors d'aplicacions instal·lats i altres mecanismes. Per aquesta raó, a més dels usuaris reals, en un sistema n'hi haurà molts altres de vinculats a altres tasques que s'han de fer en l'operatiu. Generalment, aquest tipus d'usuari no podrà entrar (amb una identificació o login normal) al sistema.

En tot sistema operatiu hi ha d'haver un superusuari (*root*). Aquest serà l'usuari que tindrà tots els permisos, el que tindrà els privilegis màxims que li permetran efectuar qualsevol operació sobre el sistema. És necessari que aquest existeixi, ja que serà qui s'encarregarà de tota l'administració i gestió de servidors, grups, etc. Aquest compte no s'ha d'utilitzar per a treballar normalment en el sistema. Només hauríem d'entrar com a *root* quan sigui realment necessari, utilitzant altres comptes per al treball normal dels usuaris. D'aquesta manera mai no podrem danyar el sistema amb operacions errònies o amb la prova de programes maliciosos, etc.

Tota la informació d'usuaris i grups es desa en els arxius següents:

- `/etc/passwd`: informació (nom, directori de l'usuari, etc.) de l'usuari.

Contingut complementari

Un servidor és un programa que s'encarrega de proporcionar algun tipus de servei (com servir pàgines web, deixar que els usuaris es connectin remotament, etc.), generalment vinculat a la Xarxa.

Contingut complementari

També és possible configurar el sistema perquè s'utilitzi un fitxer `shadow` per als grups (en cas que sigui necessari posar-los contrasenya). Aquest fitxer s'anomenaria `etc/gshadow`. Generalment, la configuració de contrasenyes s'indica en instal·lar el sistema, encara que tot es pot canviar i adaptar al nostre gust utilitzant els mòduls PAM (*pluggable authentication modules for Linux*), que són els programes que s'encarreguen de tot el sistema d'autenticació d'usuaris.

Contingut complementari

Trencar una contrasenya significa aconseguir la paraula clau utilitzant programes especials per a això. Aquests programes també els usen els administradors de sistemes per a descobrir quins usuaris utilitzen contrasenyes massa fàcils de descobrir (les contrasenyes bones no es poden trencar de cap manera sense fer servir grans supercomputadors).

- `/etc/group`: informació sobre els grups d'usuaris.
- `/etc/shadow`: contrasenyes xifrades dels usuaris i configuració per a la seva validesa, canvi, etc.

Utilitzar l'arxiu de shadow és opcional. Al principi, les contrasenyes xifrades dels usuaris es desaven al mateix fitxer de `passwd`, però, per raons de seguretat (molts mecanismes han de poder llegir aquest fitxer, amb la qual cosa era molt fàcil aconseguir-lo i intentar violar les contrasenyes) es va optar per canviar aquest mecanisme per a fer que el fitxer de `shadow` només fos accessible per a alguns usuaris amb privilegis especials en el sistema. Aquesta opció és configurable en el procés d'instal·lació del sistema i se sol recomanar utilitzar-la. Tots aquests fitxers estan organitzats per línies, on cada una identifica un usuari o grup (depenent del fitxer). En cada línia hi ha diversos camps separats pel caràcter ":". En tasques d'administració, és important saber què són aquests camps, per la qual cosa els explorarem amb una mica més de detall:

- `passwd`
 1. Inici de sessió (*login*): l'àlies de l'usuari. No hi pot haver dos àlies iguals, encara que algun sí que pot coincidir amb un grup del sistema.
 2. Contrasenya xifrada: si no s'utilitza el fitxer de `shadow`, les contrasenyes xifrades s'emmagatzemen en aquest camp. Si utilitzem el fitxer de `shadow`, tots els usuaris existents en aquest fitxer han d'existir també en el de `shadow` i en aquest camp es posa el caràcter "x".
 3. Identificador d'usuari o user ID: número d'identificació de l'usuari. És el número amb el qual el sistema identifica l'usuari. El 0 és l'únic que està reservat per al `root`.
 4. Group ID: el número del grup al qual pertany l'usuari. Com que un usuari pot pertànyer a més d'un grup, aquest grup es denomina primari.
 5. Comentaris: camp reservat per a introduir els comentaris que vulguem sobre l'usuari. Se sol utilitzar per a posar el nom complet o algun tipus d'identificació personal.

6. Directori de l'usuari (home): el directori de l'usuari és on aquest podrà desar tots els seus fitxers. Se solen posar tots en alguna carpeta del sistema (generalment /home/) i organitzats per grups.
7. Intèrpret de comandaments: un **intèrpret de comandaments (shell)** és un programa que s'encarrega de llegir tot el que escrivim al teclat i executar els programes o comandaments que li indiquem. N'hi ha desenes, encara que el més utilitzat és, sens dubte, el bash (GNU Bourne-Again SHell). Si en aquest camp escrivim /bin/false no permetrem que l'usuari executi cap comandament en el sistema, encara que hi estigui donat d'alta.

- **group**

1. Nom del grup.
2. Contrasenya xifrada: la contrasenya d'un grup s'utilitza per a permetre que els usuaris d'un determinat grup es puguin canviar a un altre o per a executar alguns programes amb permisos d'un altre grup (sempre que es disposi de la contrasenya).
3. *Group ID*: número d'identificació del grup. És el número amb el qual el sistema identifica internament els grups. El 0 és l'únic que està reservat per al grup del *root* (els administradors).
4. Llista d'usuaris: els noms dels usuaris que pertanyen al grup, separats per comes. Encara que tots els usuaris hagin de pertànyer a un grup determinat (especificat en el quart camp del fitxer de *passwd*), aquest camp es pot utilitzar perquè usuaris d'altres grups també disposin dels mateixos permisos que té el que es referencia.

- **shadow**

1. Inici de sessió (*login*): ha de ser el mateix nom que s'utilitza al fitxer de *passwd*.
2. Contrasenya xifrada.
3. Dies que han passat, des de l'1 de gener del 1970, fins que la contrasenya ha estat canviada per última vegada.
4. Dies que han de passar fins que la contrasenya es pugui canviar.

Contingut complementari

En sistemes UNIX és molt comú representar les dates a partir del nombre de segons transcorreguts des de l'1 de gener del 1970.

Contingut complementari

En sistemes on hi ha centenars d'usuaris, és usual posar algun tipus de mecanisme per a restringir l'espai de disc que pot utilitzar cada un. En els sistemes GNU/Linux aquest sistema es diu quota.

5. Dies que han de passar fins que la contrasenya s'hagi de canviar.
6. Dies abans que caduqui la contrasenya en què s'avisarà l'usuari que l'ha de canviar.
7. Dies que poden passar després que la contrasenya caduqui, abans de deshabilitar el compte de l'usuari (si no es canvia la contrasenya).
8. Dies, des de l'1 de gener del 1970, des que el compte està deshabilitat.
9. Camp reservat.

Quan un usuari entra al sistema, se'l situa en el seu directori i s'executa l'interpret de comandaments configurat. D'aquesta manera ja pot començar a treballar. Només l'administrador (*root*) del sistema (o els usuaris del seu grup) tenen permís per a manipular la informació dels usuaris i grups, donar-los d'alta, de baixa, etc. Existeixen molts comandaments per a manipular tot això. Cada un té, a més, diversos paràmetres diferents per a gestionar tots els camps que hem vist anteriorment de forma amena. A continuació mostrem alguns d'aquests comandaments:

- **adduser**: ens serveix per a afegir un usuari nou al sistema. La forma com aquest s'afegeix (si no li especifiquem res) es pot configurar en el fitxer `/etc/adduser.conf`. Se li poden passar multitud d'opcions diferents per a especificar el directori de l'usuari, l'interpret de comandaments que cal utilitzar, etc.
- **useradd**: crea un nou usuari o en canvia la configuració per defecte. Aquest comandament i l'anterior ens poden servir per a fer les mateixes accions.
- **usermod**: amb aquest comandament podem modificar la majoria dels camps que es troben al fitxer de `passwd` i `shadow`, com el directori de l'usuari, l'interpret de comandaments, l'expiració de la contrasenya, etc.
- **chfn**: canvia la informació personal de l'usuari, continguda en el camp de comentaris del fitxer de `passwd`.
- **chsh**: canvia l'interpret de comandaments de l'usuari.

- `deluser`: elimina un usuari del sistema, esborrant o desant tots els seus fitxers segons els paràmetres que li passem, fent-ne còpia de seguretat o no, etc. La configuració que s'utilitzarà per defecte amb aquest comandament està especificada en el fitxer `/etc/deluser.conf`.
- `userdel`: comandament amb les mateixes possibilitats que l'anterior.
- `passwd`: ens serveix per a canviar la contrasenya d'un usuari, la seva informació d'expiració o per a bloquejar o desbloquejar un compte determinat.
- `addgroup`: permet afegir un grup al sistema.
- `groupadd`: el mateix que el comandament anterior, però amb paràmetres diferents.
- `groupmod`: ens permet modificar la informació (nom i GID) d'un grup determinat.
- `delgroup`: elimina un grup determinat. Si algun usuari encara el té com a primari, no es podrà eliminar.
- `groupdel`: igual que en el cas anterior.
- `gpasswd`: ens serveix per a canviar la contrasenya del grup.

Per a saber quin usuari som, podem utilitzar el comandament `whoami`, que ens mostrarà el nostre inici de sessió (*login*). `groups` ens serveix per a saber a quins grups pertanyem i `ID` ens mostrarà l'usuari i els grups. També és interessant poder convertir-nos en un altre usuari sense haver de sortir de la sessió (comandament `login` o `su`) o canviar-nos de grup amb el comandament `newgrp`. Aquest últim comandament l'hem d'utilitzar només quan no pertanyem al grup en qüestió i en sabem la contrasenya (que ha d'estar activada al fitxer de *group*). Si només necessitem els permisos del grup en qüestió per a executar un comandament determinat, també podem utilitzar `sg`.

Tal com hem dit anteriorment, GNU/Linux és un sistema operatiu multiusuari, per la qual cosa en un mateix moment hi pot haver diversos usuaris connectats al sistema simultàniament. Per a saber quins usuaris hi ha en un moment determinat, podem utilitzar el comandament `who`, que ens mostra la llista d'usuaris dins del sistema.

Contingut complementari

Com veiem, en GNU/Linux tenim més d'una manera per a fer una acció determinada. Aquesta és la tònica general que se segueix en el sistema: podem editar directament els fitxers i modificar-los nosaltres mateixos, utilitzar alguns dels comandaments que hi ha, crear-los nosaltres mateixos, etc. En definitiva, tenim la possibilitat de triar què és el que més ens agrada.

Contingut complementari

El sistema de fitxers és el programa (o mòduls del nucli de l'operatiu) que s'encarrega de fer totes les operacions relacionades amb l'emmagatzemament i la manipulació dels arxius. Són les funcions que tracten amb els dispositius físics d'emmagatzemament de l'ordinador, com el disc dur.

w, a més, ens mostra què és el que fan. Ens hi podem comunicar utilitzant el comandament `write`, amb el qual apareix el missatge que hàgim escrit a la pantalla de l'usuari indicada o `wall`, que escriu el contingut del fitxer que hàgim especificat a tots els usuaris dins del sistema. Per a activar o desactivar l'opció de rebre missatges tenim el comandament `mesg`. També podem fer un xat personal amb algun usuari a partir del comandament `talk`.

2.3. El sistema de fitxers

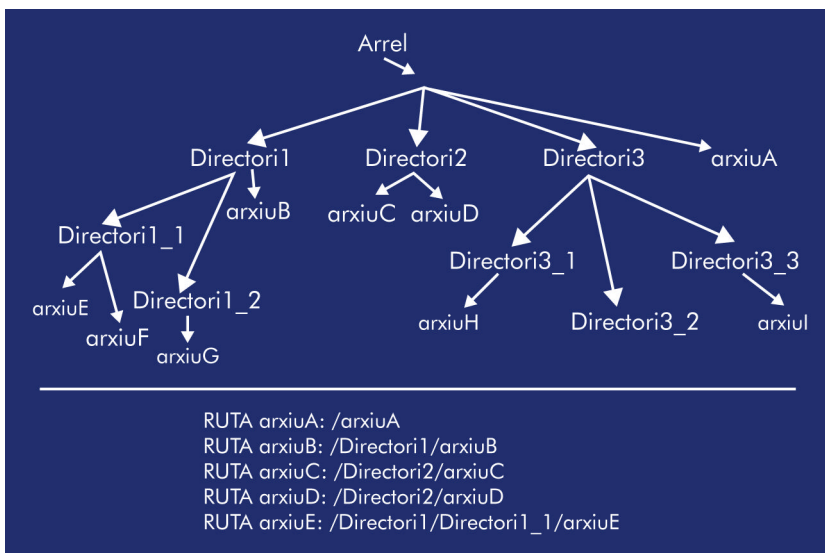
2.3.1. La jerarquia del sistema de fitxers

Tot sistema operatiu necessita desar multitud d'arxius: des dels de la configuració del sistema, els de log, els dels usuaris, etc. En general, cada operatiu utilitza el seu propi sistema de fitxers, caracteritzant-lo en molts aspectes com poden ser el rendiment, la seguretat, la fiabilitat, etc. GNU/Linux és capaç de llegir/escriure arxius amb qualsevol dels sistemes de fitxers que actualment existeixen, encara que per a la seva pròpia arrel i directoris principals és necessari un sistema de fitxers que li permeti certes operacions. Generalment, se sol utilitzar el tipus `ext2`, `ext3` o `ReiserFS`. L'`ext2` és el més típic i estès. El seu rendiment és bastant bo, incorpora tot tipus de mecanismes de seguretat i tuning i és molt fiable. `ext3` n'és l'evolució, i incorpora una tecnologia anomenada de *journaling*. Un dels principals avantatges d'aquesta tecnologia és que si hi ha un tall en el subministrament d'energia elèctrica i l'ordinador s'apaga sense tancar-se adequadament, els sistemes de recuperació de fitxers són més efectius. `ReiserFS` és un nou tipus de sistema que incorpora noves tecnologies de disseny que li permeten ser més ràpid. En el procés d'instal·lació del sistema operatiu se'ns preguntarà quin d'aquests tres volem usar. Generalment se sol utilitzar `ext2` o `ext3` perquè s'han provat més que el `ReiserFS`.

Una característica molt important de tots els sistemes operatius basats en UNIX és que tots els dispositius del sistema es poden tractar com si fossin fitxers. Igualment, quan vulguem accedir al contingut d'un CD, disquet o qualsevol altre dispositiu d'emmagatzematge, haurem de muntar-lo en un directori ja existent en el sistema i hi navegarem com si es tractés d'una carpeta més (l'ús de diferents unitats

–A:, B:, C:, D:, etc. és un esquema existent únicament en sistemes operatius tipus WindowsTM).

El primer que hem de tenir clar és que tot el sistema de fitxers parteix d'una mateixa arrel, a la qual ens referirem amb el caràcter "/". És l'origen de tot el sistema de fitxers i només n'existeix una. Per a organitzar els fitxers adequadament, el sistema proporciona el que anomenarem directoris (o carpetes), dins de les quals podem posar arxius i més directoris. D'aquesta manera aconseguim una organització jeràrquica com la que veiem en la figura següent:



2.3.2. Directoris del sistema

La majoria dels sistemes operatius del mercat segueixen l'estàndard FHS (<http://www.pathname.com/fhs/>), en què s'especifiquen les principals característiques que hauria de tenir qualsevol sistema operatiu. Entre aquestes hi ha la distribució en directoris que hem de fer dels nostres arxius per a tenir-los organitzats correctament i poder-los localitzar ràpidament i senzillament. En la majoria de distribucions basades en GNU/Linux se segueixen aquestes recomanacions, i es troben els directoris principals següents:

- /bin/: comandaments bàsics per a tots els usuaris del sistema.
- /boot/: arxius estàtics necessaris per a l'arrencada del sistema.

Contingut complementari

El sistema de fitxers ext2 ha estat dissenyat per a manejar molt ràpidament fitxers petits, que és el que més sol tenir un sistema operatiu. Amb el maneig i la manipulació de grans fitxers multimèdia, no se'n surt tan bé, encara que sempre es pot fer una mica de tuning per a adaptar-lo més a les nostres necessitats.

- `/dev/`: dispositius del sistema.
- `/etc/`: arxius de configuració del sistema i de les aplicacions que hi estan instal·lades.
- `/home/`: directori per a posar les carpetes home dels usuaris.
- `/libra/`: llibreries essencials per al nucli del sistema i els seus mòduls.
- `/mnt/`: punt de muntatge temporal per a dispositius.
- `/proc/`: processos i variables del nucli del sistema.
- `/root/`: directori de l'usuari (home) per al root del sistema.
- `/sbin/`: comandaments especials per al root del sistema.
- `/tmp/`: arxius temporals. Segons la distribució utilitzada (o la configuració que utilitzem) s'esborren en engegar el sistema o cada cert període de temps.
- `/usr/`: segona estructura jeràrquica, utilitzada per a emmagatzemar tot el programari instal·lat en el sistema.
- `/var/`: directori per als gestors de cues (spoolers) d'impressió, fitxers de diari (log), etc.

És molt recomanable conservar i no eliminar cap d'aquests directoris (o els que per defecte ens crea la distribució que utilitzem), ja que són bàsics per al bon funcionament del sistema. Generalment, els processos d'instal·lació d'aplicacions noves necessiten que existeixi l'organització donada i molts dels arxius de configuració dels programes han de ser en uns directoris determinats. El que sí que podem fer sense cap tipus de restricció és crear directoris nous en l'arrel del sistema o en qualsevol altra carpeta.

2.3.3. Movent-nos

Per a moure'ns per l'estructura de directoris hem d'utilitzar els comandaments per a fer llistes de continguts i canviar de carpeta. Quan entrem en el sistema, és usual que l'inici de la sessió (login) ens situï en el nostre directori de l'usuari (home), que generalment se sol referenciar amb el caràcter "`~`". Si volem veure el que hi ha en el

directori on estem situats, podem fer una llista dels continguts utilitzant el comandament `ls`. Hem de tenir en compte que per defecte el comandament no ens mostra els arxius que comencen per un punt. Amb el paràmetre `-a` sí que ens mostraria absolutament tots els fitxers. En tots els directoris hi ha una entrada `.` i una altra `..`. El punt és la referència al directori actual, mentre que els dos punts seguits fan referència al directori immediatament superior (en l'arbre de jerarquies) a l'actual. Naturalment, quan estem situats en l'arrel del sistema de fitxers, l'entrada `..` no existirà perquè ens trobem en el nivell superior.

Per a canviar de directori podem utilitzar el comandament `cd`. Si no li passem cap paràmetre, per defecte ens situarà en el nostre directori d'usuari (*home*). Generalment, se li sol indicar on volem anar, passant-lo de forma absoluta o relativa. De forma relativa significa que partirem del directori on som en el moment d'executar el comandament. Per exemple, si som en el directori `/usr/bin/` i volem anar al `/root/`, haurem d'introduir el comandament següent: `cd ../../root` (els dos primers punts indiquen `/usr/` i els següents l'arrel `/` del sistema, a partir de la qual ja podem accedir a `/root/`). De forma absoluta sempre partim de l'arrel, de manera que el comandament que utilitzaríem per a l'exemple anterior seria: `cd /root`. Per a saber en quin directori som, podem utilitzar el comandament `pwd`.

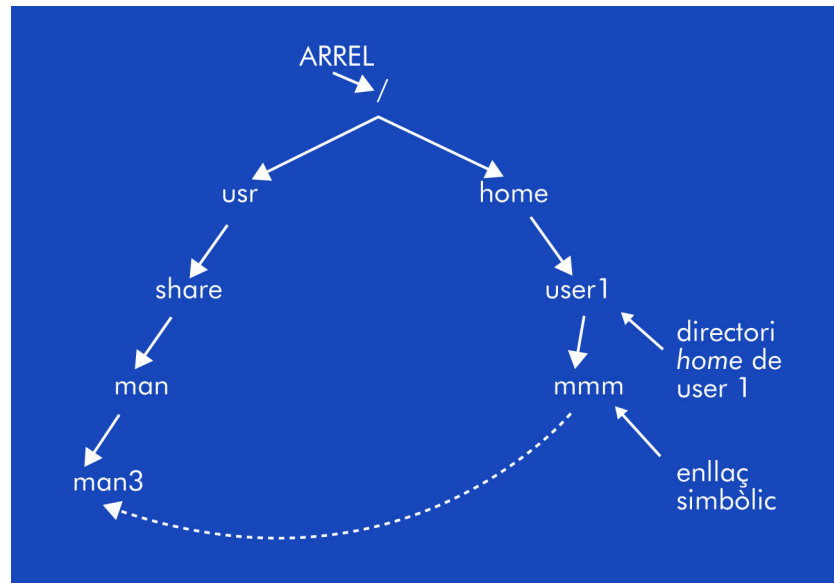
2.3.4. Enllaços

Altres mecanismes que ens proporcionen la gran majoria de sistemes de fitxers són els que anomenem enllaços. Un enllaç és un pont a un arxiu o directori pertanyent al sistema; una referència que podem posar en qualsevol lloc que ens interressi i que actua com un accés directe a qualsevol altre. Aquest mecanisme ens permet accedir a carpetes o arxius més ràpidament i còmodament, sense haver de desplaçar-nos per la jerarquia de directoris. Ho veurem amb un exemple: imaginem que som un usuari (`user1`) que necessita accedir freqüentment al directori `/usr/share/man/man3/`. En lloc d'escriure el llarg comandament que ens situaria en el directori en qüestió cada vegada que necessitèssim desplaçar-nos-hi, podem crear un enllaç en el nostre propi directori que ens redireccioni directament cap allà. El comandament `ln -s /usr/share/man/man3 mmm` ens crearia aquest pont, que hem anomenat `mmm`. L'usuari només

Contingut complementari

El fet que `ls` no ens mostri (per defecte) els arxius que comencen per punt, és perquè cada vegada que fem una llista del contingut del directori no hàgim de veure els fitxers i directoris de configuració de les aplicacions que utilitzem (que solen començar per aquest caràcter) i les entrades del directori actual i anterior, que sempre existeixen.

hauria d'escriure (des del seu directori d'usuari) "cd mmm" i automàticament el sistema el redirigiria cap a /usr/share/man/man3/. És important tenir en compte que en fer un "cd .." per anar al directori superior, tornariem al directori d'usuari i no a usr/share/man/, ja que hi hem accedit a partir del nostre enllaç. Podem veure aquest esquema gràficament en la figura següent:



Contingut complementari

Un enllaç fort només es pot crear entre fitxers o directoris d'una mateixa unitat a causa del mecanisme intern que s'utilitza per a gestionar-los.

En crear l'enllaç de l'exemple anterior hem passat el paràmetre "-s" al comandament. Això indica que volem crear un enllaç simbòlic. Els enllaços simbòlics signifiquen que només creem un apuntador o pont cap al fitxer o directori, de manera que si esborréssim el fitxer destinat, l'enllaç no apuntaria enlloc. Si no posem el paràmetre "-s" es crearia el que anomenem un enllaç fort (*hard link*) que, a diferència de l'anterior, fa un duplicat del fitxer. De fet, internament no és exactament un duplicat, és com dues entrades que apunten a les mateixes dades. D'aquesta manera, si en modifiquem un o altre, tots dos queden iguals. L'avantatge d'aquest tipus d'enllaç és que si esborrem qualsevol de les dues còpies del fitxer l'altra encara es conserva. Aquest tipus d'enllaç no s'utilitza massa perquè complica la gestió i manipulació dels fitxers (sempre és millor tenir una sola còpia dels arxius). A més, si fem un enllaç fort d'un directori, tots els arxius i subdirectoris que contingui també s'hauran de referenciar. Per aquesta raó només el root del sistema pot fer enllaços forts de directoris. Una altra diferència és que amb un enllaç simbòlic podem veure a quin fitxer apuntem, mentre que amb un de fort no ho podem veure (a causa del mecanisme que s'utilitza internament per a ells).

2.3.5. Permisos

En qualsevol sistema operatiu multiusuari necessitem que els fitxers que desem al nostre disc puguin tenir una sèrie de propietats que ens permetin veure'ls, modificar-los o executar-los per als usuaris que nosaltres definim. Encara que hi ha diverses alternatives per a fer això, GNU/Linux utilitza el sistema clàssic de UNIX, que, combinat amb tots els mecanismes de gestió d'usuaris i grups, ens permet qualsevol configuració possible. El que interessa és definir, per a cada fitxer o directori, a quin usuari i grup pertany i quins permisos té per a cada un d'ells, i també per a la resta d'usuaris del sistema. Executant `ls -l` veurem que a cada arxiu del directori on som apareix una línia semblant a la següent:

```
-rwxr-xr-x 1 user1 grup1 128931 Feb 19 2000 gp1.txt
```

Els deu primers caràcters (començant per l'esquerra) ens indiquen els permisos del fitxer de la manera següent:

- Caràcter 1: aquesta entrada ens indica si és un fitxer o un directori. En cas de ser un fitxer, apareix el caràcter "-", mentre que per als directoris apareix una "d".
- Caràcters 2, 3, 4: ens indiquen, respectivament, els permisos de lectura, escriptura i execució per al propietari del fitxer. En cas de no tenir el permís corresponent activat, trobem el caràcter "-" i si no "r", "w" o "x", segons si el podem llegir (*Read*), escriure (*Write*) o executar (*eXecute*). En el tercer caràcter, a més, ens podem trobar una "s", que ens indica si l'arxiu és de tipus *SetUserId*, que vol dir que en executar-lo obtindrà els permisos del propietari del fitxer. Si només té el permís "x", quan el programa s'executa ho fa amb els permisos de qui l'hagi llançat.
- Caràcters 5, 6, 7: aquests caràcters tenen exactament el mateix significat que anteriorment, però fan referència als permisos concedits als usuaris del grup a què pertany el fitxer.
- Caràcters 8, 9, 10: igual que en el cas anterior, però per als altres usuaris del sistema.

Contingut complementari

El mecanisme de *SetUserId* és molt útil quan un programa necessita tenir els permisos del seu propietari per a accedir a certs arxius o fer algun tipus d'operació en el sistema. De tota manera, hem de vigilar molt amb aquest tipus de fitxers perquè poden comportar errors de seguretat en el sistema si s'utilitzen malament.

Després d'aquests deu caràcters trobem una xifra que ens indica el nombre d'enllaços forts que té el fitxer. Per als directoris, aquest nombre indica quantes carpetes hi ha dins seu a més dels enllaços forts que té (quan no n'hi ha cap, el número és 2, a causa de la gestió interna de l'operatiu). A continuació veiem el propietari i el grup de l'arxiu, seguit de la mida (en bytes) que ocupa i la data de l'última modificació. A tots els fitxers es desa la seva data de creació, últim accés i modificació, que podem manipular, amb el comandament `touch`. Al final hi ha el nom del fitxer, on es diferencien minúscules de majúscules i podem tenir tot tipus de caràcters sense cap problema.

Per a canviar els permisos d'un arxiu determinat podem utilitzar el comandament `chmod`. Hem de tenir en compte que només el propietari de l'arxiu (o el `root`) pot canviar aquests permisos, ja que si no, el mecanisme no tindria cap sentit. Podem utilitzar aquest comandament de moltes maneres diferents, però les dues més freqüents són les següents:

- La primera manera d'utilitzar-lo és de l'estil `"chmod XXX nombre-Archivo"`. Les "X" han de ser tres números entre el 0 i el 7. El primer número indica els permisos que volem establir per a l'usuari, el segon, per al grup, i el tercer, per a la resta. Per a interpretar correctament els permisos que donarem utilitzant els números del 0 al 7, hem de fer ús de la representació binària del número en qüestió, de manera que el primer dígit indicarà el permís d'escriptura, el segon, el de lectura, i el tercer, el d'execució. En cada cas un 0 indica que no es dóna el permís en qüestió i l'1 indica que sí que es dóna. En la taula següent podem veure aquesta relació:

Representació decimal	Representació binària	Significat
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rxw

- L'altra manera d'utilitzar el comandament és indicant explícitament quin permís volem donar o quin permís volem eliminar del fitxer. La manera de fer-ho és indicant, primer, si ens referim als permisos de l'usuari, grup o a la resta amb les lletres "u", "g" o "o" respectivament. Seguidament, hem d'afegir un "+" o "-" segons si volem afegir o eliminar l'atribut, que indicarem amb "r", "w", "x" o "s" (aquest últim per al `SetUserId`). A més, podem fer totes les combinacions possibles, referint-nos a més d'un permís i/o usuaris. Per exemple, "`chmod go+r gp1.txt`" donaria el permís de lectura al grup i als altres usuaris per al fitxer `gp1.txt`.

Perquè canviï el propietari d'un fitxer existeix el comandament `chown`, que només pot utilitzar el `root` per raons de seguretat. Per a canviar el grup d'un arxiu determinat, es pot utilitzar el comandament `chgrp`. Com podem suposar, quan un usuari crea un arxiu nou, el sistema posa com a propietari l'usuari que l'ha creat i el dona com a pertanyent al grup primari del mateix usuari. Els permisos que es posen per defecte en crear un nou arxiu els podem configurar amb el comandament `umask`, al qual hem de passar la mateixa notació de tres nombres decimals entre el 0 i el 7 que hem vist anteriorment però complementats. Per exemple, si volem que els nostres fitxers s'inicialitzin amb els permisos "`rw- r--r--`", hauríem d'escriure "`umask 133`".

2.3.6. Manipulació, patrons i recerques

Ara que ja ens sabem moure correctament per la jerarquia de directors, també necessitem saber com copiar, eliminar i manipular correctament altres aspectes dels fitxers. El comandament `rm` és el que s'encarrega d'eliminar els arxius que li indiquem. Per a eliminar un directori, podem utilitzar el comandament `rmdir`, encara que només l'esborrarà quan aquest sigui buit (si volguéssim esborrar completament un directori i tot el seu contingut, podríem utilitzar "`rm - r`"). Per a copiar arxius d'un lloc a un altre tenim el comandament `cp`, al qual sempre hem d'indicar el fitxer o directori origen i el lloc o nom de destinació, encara que sigui en el directori actual. D'aquesta manera, si volem copiar l'arxiu `/home/user1/gp1.txt` en el directori actual (i amb el mateix nom) hauríem d'escriure "`cp /home/user1/gp1.txt`". Si en lloc de copiar els arxius volem moure'ls de lloc, podem utilitzar el comandament `mv`.

Contingut complementari

Si es permetés als usuaris canviar el propietari dels seus fitxers, la seguretat del sistema quedaria compromesa, perquè es podrien fer accions malicioses i després canviar el propietari dels arxius utilitzats inculpanent altres usuaris.

Contingut complementari

La sintaxi dels patrons (*patterns*) pot arribar a ser molt complexa, i ens permet referenciar qualsevol conjunt d'arxius que vulguem.

Un mecanisme molt útil que ens proporciona el sistema són els patrons (*patterns*). Fins ara hem vist com aplicar certes operacions sobre un arxiu determinat. Quan manipulem un sistema, en molts casos ens interessarà aplicar alguna de les operacions que hem vist però sobre un grup gran de fitxers. Els patrons ens permetran aplicar les operacions que vulguem especificant en una sola instrucció diversos fitxers que compleixin una sèrie de característiques concretes. Els hem de veure com a plantilles de número, de manera que el caràcter "*" significa qualsevol cadena de caràcters possibles i el "?" ens serveix com a comodí en qualsevol caràcter. D'aquesta manera, si volem fer una llista de tots els arxius que comencin per "s", que després tingui qualsevol altre caràcter, els segueixi una "a", i després qualsevol altra cadena, podríem utilitzar "ls s?a*". Entre "[" i "]" podem incloure altres caràcters, indicant que el patró té èxit si se'n troba algun en el nom. Per exemple, si volguéssim referenciar tots els arxius que comencin per "a" o per "b" i que continuïn amb qualsevol altra cadena, podríem escriure el patró (pattern) "[ab]*". Si després de "[" poséssim el caràcter "!" ("![ab]*") indicariem que el patró coincideix amb qualsevol arxiu que no comenci per "a" o "b". Finalment, per a facilitar certes recerques, dins de "[" i "]" podem especificar classes de caràcters de la manera següent: "[:class:]", on la "class" pot ser qualsevol de les esmentades en la taula següent:

Classe	Significat	Classe	Significat
alnum	[A-Za-z0-9]	alpha	[A-Za-z]
blank	[\]	cntrl	cars de control
digit	[0-9A-Fa-f]	graph	cars imprimibles (sense espais)
lower	[a-z]	print	cars imprimibles (amb espais)
punct	[.,!@?;:] . . .	space	[]
upper	[A-Z]	xdigit	[0-9A-Fa-f]

A-Z indica caràcters de la A a la Z, \t és el tabulador i \n és un salt de línia.

Naturalment, els patrons els podem utilitzar amb qualsevol dels comandaments que hem vist i la majoria dels que veurem a continuació. A més, la major part dels comandaments de fer una llista, eliminació, còpia, etc. de fitxers també permeten que se'ls passi un paràmetre (generalment "-r") per a fer les accions respectives de manera recursiva. D'aquesta manera, s'entrarà i s'executarà la instruc-

ció corresponent a tots els arxius i directoris, a partir d'on ens trobem i fins a arribar a l'últim nivell de la jerarquia.

Un altre tipus d'operació molt útil és la recerca de fitxers. Tenim diversos comandaments que ens permeten fer recerques de diferents tipus sobre tots els fitxers del sistema.

<code>find</code>	És el comandament més versàtil per a fer aquesta acció. Ens permet filtrar els fitxers per a trobar-hi els que tenen un nom determinat, els modificats o creats a partir d'una certa data, els que tenen uns permisos determinats, etc. El seu únic desavantatge és que no utilitza cap tipus de mecanisme per a accelerar la recerca, fet pel qual poden trigar bastant.
<code>locate</code>	Es tracta d'un altre comandament, però, a diferència de l'anterior, utilitza una base de dades interna que s'actualitza periòdicament i ens permet fer recerques bastant més ràpides. Hem de tenir en compte, tanmateix, que els resultats no sempre estaran actualitzats, a més que no podrem fer recerques tan versàtils com amb <code>find</code> .
<code>whereis</code>	Finalment, <code>whereis</code> està orientat a la recerca dels arxius binaris (els executables), d'ajuda o els de codi font d'un programa determinat.

2.3.7. Tipus i contingut de fitxers

Els arxius que tenim en el nostre sistema poden ser de tipus molts diferents: executables, de text, de dades, etc. A diferència d'altres sistemes que utilitzen l'extensió de l'arxiu per a determinar de quin tipus són, GNU/Linux utilitza un sistema denominat de magic numbers, que determina amb un número màgic el tipus de fitxer segons les seves dades (es passen una sèrie de tests que intenten determinar de quin tipus és el fitxer). El comandament `file` ens ho indica.

Si necessitem veure el contingut d'un fitxer, un dels comandaments bàsics és el `cat`. Passant-li el nom o noms dels arxius que volem veure, es mostra el seu contingut per pantalla. Hem d'intentar no mostrar fitxers executables o de dades per pantalla, ja que la impressió de caràcters no imprimibles ens deixaria la consola amb caràcters no comprensibles (sempre la podem reiniciar teclejant `reset` o `tset`). Per a fitxers molt extensos, ens aniran molt millor els comandaments `less` o `more`, que ens permeten desplaçar pel fitxer progressivament. Si el tipus de fitxer és binari i volem veure què conté, podem utilitzar els comandaments `hexdump` o `od` per a veure el contingut de

Nota

Si volem actualitzar la base de dades interna que utilitza el comandament `locate`, podem utilitzar `updatedb`.

Contingut complementari

Utilitzar l'extensió per a determinar el tipus d'un arxiu no és un sistema gaire eficaç, ja que qualsevol la pot canviar i generar confusions i errors al sistema.

forma hexadecimal o altres representacions. `strings` ens buscarà les cadenes de caràcters dins d'un fitxer binari i les mostrarà per pantalla.

Un altre tipus de comandaments molt útils són els que ens busquen un cert patró en el contingut dels fitxers. Amb el comandament `grep` podem passar com a segon paràmetre el nom de l'arxiu i com a primer el patró que vulguem buscar (amb la sintaxi que hem vist anteriorment, estesa a altres opcions). A més, el comandament ens permet altres múltiples accions, com ara comptar el nombre de línies on apareix el patró (paràmetre `-c`), etc. Amb `cut` podem separar en camps el contingut de cada línia del fitxer especificant quin caràcter és el separador, molt útil en tasques d'administració del sistema per a la seva automatització. També podem agafar un nombre determinat de línies del principi o final d'un arxiu amb els comandaments `head` i `tail` respectivament. Amb `wc` podem comptar el nombre de línies o paraules, la màxima longitud de línia d'un fitxer, etc.

Finalment, per acabar aquesta secció de manipulació de fitxers, l'únic que ens falta per veure és com comparar arxius diferents. Igual que amb les altres operacions, tenim diversos comandaments que ens ho permeten fer. `diff`, `cmp` i `comm` fan comparacions de diferents formes i mètodes als fitxers que indiquem. `sdiff`, a més, permet barrejar-los a la nostra elecció.

2.4. Els processos

El fet que el sistema operatiu sigui multitasca implica que podem llançar més d'un programa alhora. Un procés no és més que un programa o aplicació que es troba carregat en la memòria i en procés d'execució. Encara que el nostre ordinador només disposi d'una CPU, el sistema operatiu s'encarrega de repartir el seu temps de processament perquè diversos processos puguin fer les seves operacions, donant la sensació que s'executen tots alhora.

Per a identificar inequívocament cada procés, el nucli del sistema els assigna un número anomenat PID (*process identification*). Encara que podríem pensar que solament amb el nom ja els podríem referenciar, és imprescindible disposar d'aquest número perquè podem

Contingut complementari

La gestió de processos és un aspecte vital en tot sistema operatiu, ja que determina el temps de resposta de les nostres aplicacions, l'eficiència amb què s'utilitza la memòria i la CPU, etc.

executar un mateix programa tantes vegades com vulguem, al mateix temps que se n'executin diferents instàncies. Per a saber quins processos s'executen, podem utilitzar el comandament `ps`. Per a explorar una mica més tot aquest mecanisme de processos, explicarem amb més detall alguns dels paràmetres que podem passar a aquest comandament:

- `"T"`: aquesta opció ve per defecte i ens indica que només es mostraran els processos que s'executen al terminal on ens trobem o que s'hagin llançat a partir d'ell.
- `"-a"`: ens mostra els processos de tots els terminals del sistema.
- `"-A"`: ens mostra tots els processos del sistema. Si executem el comandament, veurem que, a part dels programes que els usuaris executen, n'hi ha d'altres. Molts d'aquests executen les funcions necessàries perquè l'operatiu funcioni correctament, d'altres són els servidors d'aplicacions configurats, etc.
- `"-l"`: ensenya informació estesa per a cada procés, com el temps de CPU que ha utilitzat, el terminal on s'executa, etc. En la segona columna també podem veure l'estat del procés. Encara que el sistema tingui molts processos executant-se en un mateix instant de temps, això no implica que tots necessitin temps de CPU constantment. Per exemple, quan un servidor de pàgines web no té cap petició, no és necessari que faci absolutament cap operació. Encara que estigui en memòria preparat per a executar-se en rebre una petició, és millor que no passi en cap moment per la CPU, ja que aquesta es pot utilitzar per a altres processos que sí que la necessiten. Internament, el sistema operatiu té implementats una sèrie de mecanismes molt eficaços per a gestionar tota aquesta classe d'operacions. D'aquesta manera, un procés pot estar en els estats següents (mostrats amb el caràcter corresponent):
 - `"D"`: procés ininterrompible. Aquest tipus de procés generalment sol pertànyer a l'entrada/sortida d'algun dispositiu que es faria malbé si deixés de ser atès.
 - `"R"`: procés que en el moment d'executar el comandament també s'executa, és a dir, tots aquells que són a la cua d'execució. La cua d'execució de processos és on es posen tots aquells que es reparteixen el temps de la CPU.

Contingut complementari

Amb els comandaments de manipulació de processos podem fer qualsevol acció que ens interressi: des d'interrompre els processos d'un usuari concret, eliminar aquells que no ens interessin o fer que alguns ocupin més temps la CPU perquè vagin més ràpid.

Contingut complementari

Per a tractar els senyals en un *shell script* (vegeu més endavant com programarlos), podem utilitzar el comandament `trap`.

- "S": procés adormit o esperant que ocorri algun tipus d'esdeveniment que el sistema el desperti i el posi a la cua d'execució.
- "T": procés que ha estat detingut per l'usuari o el sistema.
- "Z": procés zombie. Aquest estat indica que el procés ha tingut algun error i no funciona correctament. Generalment és millor eliminar aquest tipus de processos.

Un altre comandament molt útil és el `top`, que ens informa interactivament dels processos del sistema, de l'estat d'utilització de la CPU, la memòria utilitzada i lliure, la RAM que utilitza cada procés, etc. Aquest programa és molt indicat quan el sistema no respon adequadament o hi notem alguna disfunció estranya, ja que ens permet localitzar ràpidament quin procés afecta negativament el rendiment del sistema.

Com veiem, el sistema ens informa sobre tots els aspectes possibles dels processos del sistema. A més d'això, podem enviar certs senyals als processos per a informar-los d'algun esdeveniment, podem treure'ls de la cua d'execució, eliminar-los, donar-los més prioritat, etc. Saber manipular correctament tots aquests aspectes també és molt important, ja que ens permetrà utilitzar el nostre ordinador de manera més eficient. Per exemple, si som administradors d'un centre de càlcul on la majoria d'aplicacions que s'executen necessiten molt de temps de CPU, podríem configurar el sistema per fer que els més urgents s'executessin amb més prioritat que d'altres i acabessin primer. El comandament `kill` ens permet enviar senyals als processos que ens interessin. En general, tots els programes es dissenyen perquè puguin rebre aquest tipus de senyals. D'aquesta manera, segons el tipus de senyal rebut saben que han de fer unes operacions o d'altres. Hi ha molts tipus diferents de senyals, que podem veure en el manual de `kill`, encara que els més utilitzats són els que ens serveixen per a obligar un procés que n'acabi o n'interrompi l'execució. Amb el senyal `TERM` ("`kill -15 PID`"), indiquem al procés que volem que acabi, de manera que en rebre el senyal haurà de desartot el necessari i acabar la seva execució. Si hi ha algun tipus de problema o el programa no està preparat per a rebre aquest tipus de senyal, podem utilitzar `KILL` ("`kill -9 PID`"), que automàticament l'expulsa de la cua d'execució. `killall` serveix per a referir-nos al nom de diversos processos alhora en lloc de referenciar-los

pel seu PID *i*, d'aquesta manera, enviar-los un senyal a tots alhora. Amb el comandament `skill` també podem enviar senyals als processos, però amb una sintaxi diferent. Per exemple, si volem detenir totes les execucions d'un usuari determinat, podem utilitzar "`skill -STOP -u nomLogin`", que fa aturar tots els processos d'aquest usuari. Per a reiniciar-los de nou, podem passar el senyal de `CONT`. Quan executem algun programa en una consola i li volem passar el senyal de `TERM`, podem utilitzar la combinació de tecles `CTRL + C`. Amb `CTRL + Z` podem interrompre un programa i reviure'l amb `fg`.

Una altra manera de veure els processos és per la seva jerarquia. Igual que en el sistema de fitxers, els processos segueixen una certa jerarquia de pares a fills. Tot procés ha de ser llançat a partir d'un altre (tant si és el mateix intèrpret de comandaments, com l'entorn gràfic, etc.), de manera que es creï una relació de pares a fills. Amb el comandament `pstree` podem veure aquesta jerarquia gràficament. Si l'executem, veurem que el pare de tots els processos n'és un que s'anomena `init`. A partir d'aquest parteixen tots els altres, que alhora poden tenir més fills. Aquesta estructura jeràrquica és molt útil, ja que, per exemple, matant un procés pare que conté molts altres fills, també matem tots els seus fills. També ens pot servir per a identificar d'on parteixen certs processos, etc. Si no passem cap paràmetre al comandament, per defecte compacta tots els processos amb un mateix nom per a no mostrar una estructura massa gran, encara que això també és configurable a partir dels seus paràmetres.

Tots els processos del sistema tenen una certa prioritat. Com hem dit abans, aquesta prioritat indica el temps de CPU que es deixarà al procés. Com més prioritari sigui el procés, més temps d'execució tindrà respecte als altres. El rang de prioritats va des del `-20` al `19`, de més gran a més petit. Per a llançar un procés amb una prioritat determinada, podem utilitzar el comandament `nice`. Si volem donar una prioritat diferent d'un procés que ja estigui en execució, podem utilitzar `renice`. Només el `root` pot utilitzar el rang de prioritats negatives; així, el sistema s'assegura que el `root` compti sempre amb la possibilitat d'executar processos més ràpidament que els usuaris. Per defecte, la prioritat amb què s'executen els programes és `0`. Un aspecte que caldrà considerar és que amb tot aquest mecanisme de prioritats no podem mesurar el temps d'execució real d'un procés perquè la CPU es reparteix entre tots els que tinguem en la cua d'execució. En centres de càlcul on es factura segons el temps d'utilització de les

màquines, és molt important poder mesurar adequadament aquest aspecte. Per aquest motiu, el sistema ens proporciona el comandament `time`, el qual, en passar-lo al programa que volem mesurar, ens torna el temps real de CPU que ha utilitzat.

2.5. Altres comandaments útils

2.5.1. L'ajuda del sistema

Com hem dit al llarg del document, tots els comandaments tenen multitud d'opcions i paràmetres diferents que ens permeten manipular-los a la nostra elecció. Des del principi es va tenir molt en compte que és imprescindible disposar d'una bona documentació per a tots ells. Igualment, tota aquesta informació és necessària per als fitxers de configuració del sistema, les noves aplicacions que utilitzem, etc. És per això que el sistema mateix incorpora un mecanisme de manuals amb el qual podem consultar gairebé tots els aspectes dels programes, utilitats, comandaments i configuracions existents. El comandament més utilitzat és el `man`, que ens ensenya el manual del programa que li indiquem com a paràmetre. Per defecte, aquesta documentació es mostra per mitjà del programa `less`, amb el qual ens podem desplaçar cap endavant i cap enrere amb les tecles "AvPág" i "RePág", buscar una paraula amb el caràcter "/" seguit de la paraula ("n" ens serveix per a buscar les ocurrències següents i "N" per a les anteriors), "q" per a sortir, etc. Els manuals del sistema estan dividits en diferents seccions segons la seva naturalesa:

1. Programes executables (aplicacions, comandaments, etc.).
2. Crides al sistema proporcionades per l'interpret de comandaments.
3. Crides a llibreries del sistema.
4. Arxius especials (generalment els de dispositiu).
5. Format dels arxius de configuració.
6. Jocs.
7. Paquets de macro.

8. Comandaments d'administració del sistema (generalment aquells que només el `root` pot utilitzar)
9. Rutines del nucli.

Si hi ha més d'un manual disponible per a una mateixa paraula, ho podem especificar passant el nombre corresponent de la secció que ens interessa abans de la paraula, per exemple `man 3 printf`. Com els altres comandaments, `man` té multitud d'opcions diferents documentades en el seu propi manual (`man man`), a partir de les quals podem fer recerques automàtiques, crear un fitxer del manual en format imprimible, etc. Una d'aquestes opcions, que ens pot anar molt bé quan no sapiguem exactament el programa que busquem, és `-k` (el comandament `apropos` fa gairebé exactament el mateix). Amb `man -k` seguit d'una paraula que faci referència a l'acció que vulguem fer es buscarà entre tots els manuals del sistema i es mostraran els que en la seva descripció o nom aparegui la paraula indicada. Així, podem trobar el que volem sense haver de recórrer a cap llibre o referència externa al sistema.

Si el manual no ens proporciona tota la informació que necessitem, podem usar el comandament `info`, que és el mateix que el manual però encara més estès. Si l'únic que volem és tenir una breu referència del que fa un determinat programa/llibreria/ etc., podem utilitzar el comandament `whatis`.

2.5.2. Empaquetatge i compressió

Comprimir un arxiu, agrupar-ne diversos en un de sol o veure què conté un arxiu comprimit són tasques que efectuarem freqüentment per a fer còpies de seguretat, transportar arxius d'un lloc a un altre, etc. Encara que hi ha multitud de programes diferents que ens permeten dur a terme aquesta classe d'operacions, generalment en tots els sistemes GNU/Linux trobarem l'eina `tar`. Aquest programa ens permet manipular de qualsevol manera un o diversos arxius per a comprimir-los, agrupar-los, etc. Encara que les seves múltiples opcions són inacabables i té molta flexibilitat, aquí només n'explicarem algunes de les més bàsiques per a fer-nos una idea del que podem fer amb aquest programa. La sintaxi que utilitza és la següent: `tar opcions arxiuDestinacio arxiusOrigen`, on

Contingut complementari

Per a fer les seves recerques ràpidament, l'aplicació `man` utilitza una base de dades interna que buscarà pels arxius que contenen els manuals i els indexarà adequadament. Si volem actualitzar aquest manual (encara que normalment el mateix sistema ja ho fa automàticament), podem utilitzar el comandament `mandb`.

l'arxiu de destinació serà el nou fitxer que volem crear i els d'origen seran els que s'agruparan o comprimiran. És important tenir en compte que si volem agrupar tota una carpeta, per defecte el procés és recursiu, de manera que en empaquetar-la aquesta recorrerà tots els seus nivells i agruparà tot el que contingui. Per a crear un arxiu nou, li hem de passar el paràmetre "c", i si el volem desar en un arxiu, li hem de passar el paràmetre "f". D'aquesta manera, "tar cf final.tar o*" empaquetarà tots els arxius del directori actual que comencin per "o". Si a més volguéssim comprimir-los, podríem utilitzar "czf", amb la qual cosa s'utilitzaria el programa gzip després d'empaquetar-los. Per a desempaquetar un arxiu determinat, el paràmetre necessari és "x", de manera que hauríem d'escriure "tar xf", que indica el fitxer empaquetat. Si estigués comprimit, hauríem de posar "xzf".

Encara que amb el mateix tar podem comprimir arxius, l'aplicació en si mateixa no és de compressió. Com hem dit, per a això utilitza programes externs com el gzip. El gzip utilitza un format de compressió propi i diferent del tan popularitzat zip, que també podem utilitzar instal·lant l'aplicació corresponent. Una altra aplicació de compressió bastant utilitzada i que proporciona molt bons resultats és el bzip2. En la taula següent podem veure l'extensió que se sol fer servir per a identificar quin format utilitza un arxiu comprimit o empaquetat:

Extensió	Format
".tar"	tar
".gz"	gzip
".tgz"	tar + gzip
".bz2"	bzip2
".zip"	zip
".z"	compress

2.5.3. Operacions de disc

La gestió i manipulació dels discos durs de l'ordinador és un altre aspecte fonamental en les tasques d'administració del sistema. Encara que més endavant veurem com configurar adequadament els discos que tinguem instal·lats a l'ordinador, en aquesta secció ex-

plicarem quins són els comandaments necessaris per a veure informació que hi fa referència. Tot disc dur està dividit en particions, a les quals podem accedir com si d'un dispositiu independent es tractés, i que denominarem *unitat*. Això és molt útil perquè ens permet separar adequadament la informació que tinguem en el sistema, per a tenir més d'un sistema operatiu instal·lat al mateix disc, etc. El comandament `df` ens mostra, de cada unitat muntada en el sistema, l'espai que s'ha utilitzat i el que és lliure. Interpretarem la sortida següent de `df`:

Filesystem	1k-blocks	Used	Available
/dev/hda1	7787712	421288	6970828
/dev/hdb1	19541504	5742384	13799120
/dev/hdc	664432	664432	0

Com podem veure, per cada partició o dispositiu muntat en el sistema el comandament ens mostra la quantitat de blocs disponibles i utilitzats. El bloc de disc és una unitat que s'utilitza internament en els dispositius d'emmagatzematge perquè el seu maneig sigui més efectiu. Per defecte, aquest comandament ens ensenya la informació per blocs d'1 K, encara que passant-li el paràmetre "`-h`" (*human readable*) el podríem veure de manera més amena. La primera línia sempre ens mostra l'arrel del sistema de fitxers (el *root filesystem*) i després els altres dispositius. Fixem-nos que també ens mostra el seu punt d'ancoratge (en l'última columna), que és la carpeta on hauríem d'anar per a poder-ne veure el contingut.

Un altre comandament molt útil és `du`, que ens mostra realment el que ens ocupa un fitxer en disc. Per a entendre clarament què volem dir amb això, hem d'aprofundir una mica més en l'organització interna dels discos i en com el sistema operatiu els manipula. Tal com hem dit anteriorment, per raons d'eficiència el sistema operatiu divideix l'espai del disc en petits trossos anomenats *blocs*. La mida del bloc és configurable i generalment depèn de la mida del disc, encara que també el podem configurar per adaptar-lo millor a les nostres necessitats. Cada vegada que volem afegir un arxiu nou, el sistema operatiu li assigna un bloc. D'aquesta manera, en llegir o operar sobre l'arxiu, l'operatiu pot llegir directament tot un bloc (de la mida configurada) en un sol pas. Quan el fitxer ocupa més d'un bloc, se li

Contingut complementari

La mida del bloc i molts altres paràmetres es poden configurar en formatar una partició del disc dur (amb el sistema `ext2` o `ext3`). Aquests paràmetres es poden ajustar per a fer que el sistema s'adapti millor a les nostres necessitats i aconseguir més eficiència.

Use%	Mounted on
6%	/
29%	/info
100%	/CD-ROM

Contingut complementari

La desfragmentació d'un disc no és més que la reorganització dels blocs dels fitxers perquè quedin en llocs consecutius i el seu accés sigui més ràpid. En els sistemes de fitxers que utilitzem amb GNU/Linux no fa falta desfragmentar els discos (encara que hi ha programes amb aquesta finalitat) perquè el sistema s'encarrega automàticament de la seva bona organització.

n'assignen més, intentant que quedin tan junts com sigui possible, de manera que es puguin llegir consecutivament i incrementant, així, la velocitat de lectura. L'únic inconvenient d'aquest sistema és el desaprofitament que es fa dels blocs quan els fitxers són molt petits, ja que si un arxiu determinat no ocupa tot el bloc, l'espai restant no es pot aprofitar per a cap altre. De tota manera, aquest tipus d'organització és la que utilitzen tots els sistemes de fitxers existents, ja que és el més rendible per a aprofitar el disc dur. El comandament `du`, doncs, ens mostra el nombre de blocs que realment utilitza un arxiu determinat al disc.

Per a saber els paràmetres que tenim configurats en les nostres unitats de disc formatades amb `ext2` o `ext3`, podem utilitzar el comandament `dump2fs`, passant-li la partició concreta. Veurem que hi ha multitud d'opcions diferents que ens en permeten ajustar molt bé el comportament (en el manual trobarem què significa cada opció). De tota manera, una vegada hàgim formatat una partició, ja no podrem modificar gairebé cap d'aquestes opcions. Si volguéssim canviar-les, hauríem de copiar tota la informació de la partició, formatar de nou i tornar a copiar els arxius originals.

Les funcions del nucli que s'encarreguen de la gestió de fitxers utilitzen una sèrie de mètodes per a agilitar-ne els processos de lectura i escriptura. Un d'aquests és la utilització d'una memòria de disc, de manera que no s'hagi d'estar constantment llegint i escrivint al dispositiu físic, que resulta un procés lent i costós. L'únic que fa el mecanisme de memòria és mantenir una còpia del fitxer amb què es treballa en la memòria RAM (molt més ràpida), de manera que el procés sigui transparent per a l'usuari (la còpia en disc es fa segons algun tipus de política implementada en el nucli). L'únic problema d'aquesta gestió és que si tenim un tall en l'alimentació i no hem tancat correctament el sistema, és possible que alguns fitxers no s'hagin pogut desar al disc físic i tinguem alguna inconsistència en el sistema de fitxers. El programa `fsck` comprova i arregla un sistema de fitxers que hagi quedat en aquest estat. Encara que el podem executar quan vulguem, generalment el mateix sistema operatiu l'executa quan en el procés d'arrencada detecta que el sistema no es va tancar adequadament (abans d'apagar l'ordinador, hem d'executar el comandament `shutdown`, que s'encarrega de llançar tots els processos necessaris perquè els programes acabin, es desmunti el sistema de fitxers, etc.). En aquest sentit, el sistema de fitxers `ext3` és més eficaç

que el seu predecessor, ja que el *journaling* li permet recuperar més informació dels fitxers perduts i més ràpidament.

Naturalment, si els fitxers que tractem en el nostre sistema són molt crítics i no ens podem permetre, en cap cas, el fet de perdre'ls, també podem configurar l'operatiu perquè no utilitzi el sistema de memòria de disc. De tota manera, és molt recomanable utilitzar aquest mecanisme perquè incrementa molt el rendiment del sistema. Si en algun moment ens interessa sincronitzar la informació de la memòria de disc amb el disc físic, podem utilitzar el comandament `sync`. Finalment, també podem comprovar la integritat física d'una partició utilitzant el comandament `badblocks`, que duu a terme una revisió sobre el dispositiu indicat per a comprovar que no hi hagi cap zona danyada.

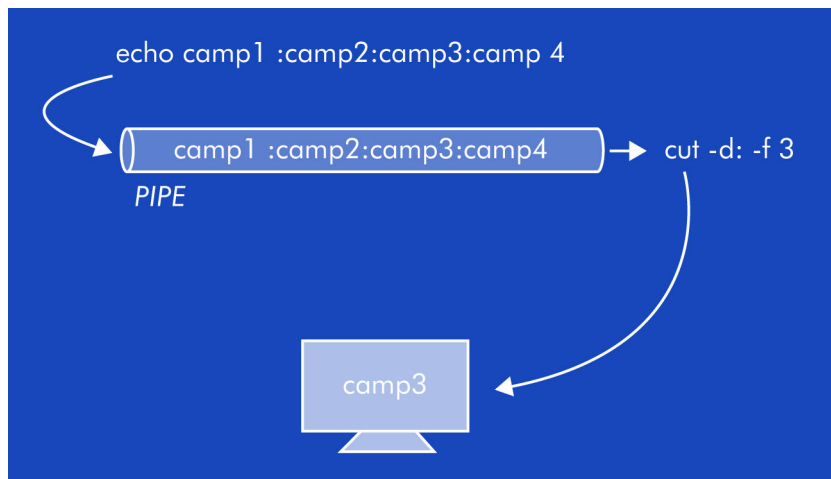
La majoria dels comandaments exposats en aquesta secció necessiten permisos especials per a executar-se, de manera que només el `root` els podrà utilitzar.

2.6. Operacions amb comandaments

2.6.1. Redireccionaments

Una vegada hem après a utilitzar alguns dels comandaments del sistema, és molt probable que en alguns casos ens interressi utilitzar-los simultàniament per a agilitar les accions que volem dur a terme. Una operació molt interessant consisteix a poder agafar la sortida d'un comandament perquè serveixi d'entrada a un altre i processar-la adequadament. El sistema operatiu utilitza un mecanisme de *pipes* (canonades), que ens permet redirigir les sortides de qualsevol comandament o programa cap on vulguem. El seu funcionament és molt simple: es tracta de posar el caràcter "`|`" entre els comandaments, de manera que la sortida del primer serveixi com a entrada per al segon. Ho veurem amb un exemple: en escriure el comandament "`echo camp1 : camp2 : camp3 : camp4`", l'únic que aconseguiríem seria que per pantalla ens aparegués "`camp1 : camp2 : camp3 : camp4`". Si d'aquesta sortida només volguéssim agafar el "`camp3`", podríem redirigir-la amb un *pipe* cap al comandament `cut`, perquè seleccioni

únicament el camp que ens interessa de la manera següent: "echo camp1 :camp2:camp3:camp4 | cut -d: -f 3". En la figura següent podem veure aquest exemple gràficament:



Naturalment, podem connectar tants pipes com necessitem per a fer accions més pràctiques que la que acabem de veure. Un altre tipus de redireccionaments molt pràctics són aquells que estan relacionats amb els fitxers. Aquest tipus de redireccionament ens permet agafar tota la sortida d'un comandament o programa i desar-la en un fitxer utilitzant el caràcter ">", igual que fèiem amb "|". Per exemple, si volem desar en un nou fitxer tot el que escrivim fins que premem CTRL + C, podríem utilitzar el següent: "cat > prova.txt". Amb ">>" podem fer exactament el mateix, però en lloc de crear sempre el fitxer nou, si aquest ja existís, s'afegiria la informació al final. Amb "<" el redireccionament es fa en sentit contrari, de manera que el contingut del fitxer que li indiquem es dirigirà cap al comandament o programa assenyalat.

Un aspecte molt interessant que hem de conèixer és que en sistemes tipus UNIX se separa la sortida normal d'un programa amb la dels errors. Encara que per defecte les dues sortides estan dirigides a la consola on s'ha executat el programa, les podem manipular perquè es dirigeixin cap on vulguem. Per a veure això de manera pràctica, provem d'esborrar un fitxer que no existeixi amb la instrucció següent: "rm fitxer > resultats". Encara que redireccionem la sortida del comandament cap al fitxer de resultats, per pantalla ens apareixerà un missatge d'error que indica que no s'ha trobat el fitxer. Això és perquè per defecte els redireccionaments només accepten la sortida estàndard del programa i no la d'error, que per defecte també es mostra per pantalla. Per a redirigir la sortida d'error, hauríem

d'indicar, abans del caràcter ">", el número "2", que és la sortida d'error ("1" és la normal). D'aquesta manera, executant `rm fitxer 2 > resultats` sí que aconseguiríem que la sortida es redirigís a l'arxiu de resultats. També podem desar la sortida normal i la d'errors en dos fitxers diferents: `rm fitxer 1 > resultats 2 > errors`. Si al contrari, volguéssim que totes les sortides es dirigissin cap a un mateix arxiu, podríem utilitzar ">&". A més, amb el caràcter "&" podem encaminar sortides d'un tipus cap a les altres; per exemple, si volguéssim encaminar la sortida d'errors cap a la normal, ho podríem indicar de la manera següent: `2>&1`.

És important tenir en compte que l'ordre dels redireccionaments és significatiu: sempre s'executen d'esquerra a dreta.

2.6.2. Comandaments específics del bash

Encara que alguns dels comandaments que hem vist ja són específics del bash, aquest intèrpret de comandaments en disposa d'altres que ens poden servir per a fer moltes altres operacions interessants. Un mecanisme molt útil és el d'executar processos en el que s'anomena *mode background*. Aquesta manera indica senzillament que el procés s'executa, però que l'intèrpret de comandaments ens torna la línia de comandaments per poder continuar executant altres programes. Per a indicar això al bash, hem d'escriure el caràcter "&" després del comandament o programa que executarem. Una vegada s'ha llançat el procés en *mode background*, es mostra una línia on se'ns indica el número de treball i el PID del procés llançat.

Amb el comandament `jobs` podem veure quins processos s'han llançat en *mode background* (passant-li el paràmetre "-1" també podem veure el seu PID). Si volem passar un d'aquests processos en *mode foreground* (com si l'haguéssim llançat des de la línia de comandaments sense el caràcter "&"), podem utilitzar el comandament `fg` indicant el PID del procés. També existeix `bg`, que ens envia un determinat procés en *mode background*. Aquest últim és útil quan, per exemple, executem un programa en manera *foreground* i l'interrompem amb CTRL + Z. Si després executem `bg` indicant-li el seu PID, el procés continuarà la seva execució en *mode background*. Tal com hem vist en seccions anteriors, els processos també tenen una jerarquia de pares a fills. Quan executem algun programa en *mode background* no inter-

Contingut complementari

El bash ens proporciona infinitud d'eines per regular qualsevol aspecte de l'interpret de comandaments. En el seu extens manual podem trobar la documentació necessària per a aprendre a manipular-les correctament.

ferim en aquesta jerarquia, de manera que si sortim de la sessió, tots aquests processos s'acabaran perquè el pare (l'interpret de comandaments des d'on els hem llançat) ja no estarà en execució. Si volem desvincular un procés del seu pare, podem utilitzar `disown`.

Un altre mecanisme molt útil del bash és la història de comandaments. És normal que utilitzant el sistema hàgim de repetir moltes instruccions escrites anteriorment. Amb les tecles del cursor amunt i avall podem veure tots els comandaments que hem utilitzat i repetir-ne algun prement RETURN. També podem utilitzar `history`, amb el qual es mostraran per pantalla tots els comandaments executats, enumerats segons la seva aparició. Escrivint "`!NUM`" s'executarà el que es correspongui amb aquesta història. També podem escriure "`!`" seguit de les lletres inicials d'algun programa executat anteriorment i el programa buscarà el més recent per executar-lo.

El bash disposa, així mateix, de tecles d'accés ràpid que ens permeten executar certes accions sense ni tan sols escriure-les. Algunes de les més freqüents són:

- **TAB**: no és necessari escriure el nom d'un fitxer, directori o comandament enterament. Si escrivim els primers caràcters i després premem la tecla del tabulador ens acabarà d'escriure la resta. Si hi hagués més d'una coincidència ens mostraria les diferents possibilitats.
- **CTRL + L**: neteja la pantalla (igual que el comandament `clear`).
- **SHIFT + REPAG**: ensenya mitja pantalla anterior.
- **SHIFT + AVPAG**: ensenya mitja pantalla posterior.
- **CTRL + W**: elimina l'última paraula escrita.
- **CTRL + T**: intercanvia l'ordre dels últims caràcters.
- **CTRL + U**: esborra tots els caràcters anteriors al cursor.
- **CTRL + D**: surt de l'interpret de comandaments (equivalent a fer un `logout`).
- `ulimit` és un comandament que ens permet configurar alguns dels aspectes interns relacionats amb el bash. Per exemple, per-

met indicar la quantitat de memòria que pot utilitzar l'interpret de comandaments, el nombre màxim d'arxius que es poden obrir, etc. Aquest comandament ens pot servir per a restringir una mica les accions que poden fer els usuaris del nostre sistema (en cas d'administrar servidors amb molts usuaris).

2.6.3. Shell scripts amb bash

Els *shell scripts* són fitxers on escrivim una sèrie de comandaments (qualsevol dels que hem vist en aquest capítol) perquè s'executin. Encara que la seva sintaxi pugui arribar a ser molt complexa i hauríem d'entrar en aspectes de programació per a entendre-la clarament, en aquesta secció explicarem d'una manera resumida algunes de les seves característiques essencials per tal que puguem entendre'ls i utilitzar-los mínimament (si volem aprofundir-hi més, podem recórrer al manual del bash). La primera línia del *shell script* ha d'especificar l'interpret de comandaments que s'utilitza:

```
#!/bin/bash
```

Després d'aquesta línia ja podem començar a escriure els comandaments que volem executar, un a cada línia. Com en tot llenguatge de programació, podem utilitzar variables, estructures condicionals i bucles. Per a declarar una variable utilitzarem la sintaxi següent:

```
nomVariable=contingut
```

Si el contingut és una cadena de caràcters, hem de posar-lo entre cometes; si és un número, no cal posar-hi res, i si volem desar en la variable la sortida d'un comandament, l'hauríem de posar entre caràcters. Per a referir-nos al contingut de la variable en altres instruccions, sempre hem de precedir el nom amb el caràcter "\$". Per a les instruccions condicionals podem utilitzar les estructures següents:

```
if condicio; then
    instruccions
else
    instruccions
fi
```

Contingut complementari

El comandament `fc` ens permet, igual que els *shell scripts*, escriure una sèrie de comandaments perquè s'executin però sense haver de desar l'arxiu.

on condició pot fer referència a un arxiu, fer alguna operació de comparació aritmètica (entre caràcters “(())”), etc. És especialment útil el comandament `test`, que ens permet fer comprovacions d'arxius, directoris, etc. i ens torna un booleà. D'aquesta manera, per exemple, si volguéssim fer una acció o una altra segons si existís un arxiu determinat, podríem utilitzar l'estructura següent:

```
if test -f /etc/inittab; then
    echo "El fitxer inittab existeix."
else
    echo "El fitxer inittab NO existeix."
fi
```

Una altra estructura condicional és la de selecció:

```
case paraula in
    cas1)
        instruccions
        ;;
    cas2)
        instruccions
        ;;
    *)
        instruccions
esac
```

En aquesta estructura es compara `paraula` amb `cas1`, `cas2`, etc., fins a trobar la que coincideixi, en la qual se n'executaran les instruccions. Si no se'n trobés cap, es passaria a la secció “*”)”, que és opcional. Aquesta estructura ens pot anar molt bé quan, per exemple, vulguem que un determinat *script* faci unes accions o unes altres segons el paràmetre que li passem. Els paràmetres els podem referenciar a partir de “\$1” per al primer, “\$2” per al segon i així successivament. Per als bucles podem utilitzar alguna de les estructures següents:

```
#BUCLE TIPUS FOR
for i in llista; do
    instruccions
done
#BUCLE TIPUS WHILE
```

Contingut complementari

Per a escriure comentaris en els *shell scripts* podem utilitzar el caràcter “#” seguit del comentari que vulguem. Aquest serà vàlid fins al final de línia.

```
while condicio; do
    instruccions
done
```

Naturalment, abans de poder executar un *shell script* hem de donar el permís d'execució al fitxer corresponent (comandament `chmod 750 nomFitxer`).

3. Taller de KNOPPIX

3.1. Introducció

Aquest taller pretén arribar a ser la vostra primera experiència amb un entorn UNIX. Per aquesta raó, el seu desenvolupament es guia pas a pas, deixant, sens dubte, la porta oberta als més curiosos perquè investiguin per compte propi. Amb aquest taller es pretén mostrar de manera pràctica tot el que s'ha exposat fins ara teòricament.

Tot el taller es pot desenvolupar sobre qualsevol PC, ja que el risc de danyar la informació que hi podem tenir és mínim. S'ha escollit aquesta distribució, KNOPPIX, perquè per a engegar-la no es requereixen coneixements previs del sistema operatiu, i perquè, una vegada detingut el sistema, no deixa rastre, tret que nosaltres el forcem (KNOPPIX, per defecte, no munta en el sistema els discos durs, així que les nostres dades estan fora de perill), a l'ordinador per on s'ha fet córrer. Òbviament, si ja es disposa d'un sistema operatiu tipus UNIX, es pot usar per a fer el seguiment del taller.

El fet de ser una distribució arrencable (*bootable*) des d'un CD-ROM i de no deixar rastre a l'ordinador on s'ha executat una vegada ha acabat el procés de parada fa que, tot i que es basi en Debian, el sistema de fitxers no pugui seguir el que marca la Debian Policy sobre aquest tema. No obstant això, aquestes diferències no afectaran el desenvolupament del taller, i tot el que aprenguem serà vàlid per als tallers posteriors. A més, és bo que des del principi ens acostumem a treballar amb diferents distribucions, i aprenguem a distingir entre el que és comú a tots els sistemes basats en UNIX i el que és propi de cada distribució.

Abans de començar només un consell: endavant amb les nostres iniciatives pròpies, intentem respondre nosaltres mateixos a les nostres inquietuds, consultem els man, fem proves, fallem i analitzem el perquè ho hem fet, intentem-ho una i una altra vegada, fins a aconse-

Nota

Una altra distribució molt popular en el moment de l'edició en català d'aquests materials és Ubuntu (<http://www.ubuntulinux.org>). Aquesta distribució, basada en Debian, es caracteritza per tenir un suport financer molt fort, fet que li permet disposar d'actualitzacions molt freqüentment. A més, se'n distribueixen dues versions diferents: una tipus Live-CD, que permet veure-la en acció sense haver-la d'instal·lar, i una altra per a fer-ne la instal·lació.

guir els resultats que volem; és així com s'aprèn UNIX, sense por, traient partit dels propis errors.

3.2. Arrencada del sistema

En primer lloc ens hem d'assegurar que el nostre ordinador s'engegarà des del CD-ROM. Per a això, entrarem en la BIOS (*Basic Input Output System*), generalment prement la tecla "Supr" durant el procés de revisió de la memòria RAM, i comprovarem que el CD-ROM està configurat com a primer dispositiu d'arrencada; si és així, ja podem sortir de la BIOS sense necessitat de desar res i posar el CD-ROM de KNOPPIX al lector. Si no fos així, faríem els canvis pertinents i els desaríem abans de sortir de la BIOS.

Després de reiniciar l'ordinador, transcorreguts uns quants segons ens apareixerà la pantalla d'arrencada de KNOPPIX amb les línies següents a la part inferior:

```
F2 for help
boot:
```

Podem prémer la tecla F2 per entrar a la pantalla on se'ns mostren les opcions que accepta KNOPPIX per arrencar. Podríem arrencar, per exemple, amb teclat espanyol i llengua castellana per a la interacció, amb Window Maker com a Windows manager i activar l'*scroll wheel* del ratolí; per a fer-ho, n'hi hauria prou de teclejar a la línia de comandaments (boot:) "knoppix lang=es wmaker wheelmouse". Però ara no és el cas; l'exemple anterior era només per a mostrar la potència de KNOPPIX; nosaltres, després d'inserir a la unitat de disquets un disquet nou formatat –sobre el qual farem els nostres primers passos en Linux, garantint així la resta d'informació que puguem tenir als nostres discs durs– només escriurem "knoppix 2" i polsarem "Intro" per a engegar el sistema en mode text.

Inicialment, el teclat està configurat per als Estats Units (us), per tant alguns caràcters no es correspondran amb el teclat espanyol; això ho arreglarem de seguida; tot i així, pot ser interessant saber on es tro-

ben alguns caràcters us al teclat espanyol: "=" és a la tecla $\hat{=}$, "/" a $\hat{=}$ i "-" a la tecla $\hat{-}$.

Una vegada pulsats INTRO, KNOPPIX començarà a carregar el sistema operatiu, tornant per pantalla alguns dels resultats dels tests que executa per a l'autoconfiguració.

Acabat aquest procés, s'obtindrà la línia de comandaments root@tty1[/]#:

```
Welcome to the KNOPPIX live Linux-on-CD!

Found SCSI device(s) handled by atp870u.o.
  Accessing KNOPPIX CD-ROM at /dev/scd0...
Total memory found: 515888 kB
Creating /ramdisk (dynamic size=407928k) on /dev/shm...Done.
Creating directories and symlinks on ramdisk...Done.
Starting init process.
INIT: version 2.78-knoppix booting
  Processor 0 is AMD Athlon(TM) XP 2100+ 1732MHz, 256 KB Cache
  APM Bios found, power management functions enabled.
  USB found, managed by hotplug.
  Enabling hotplug manager.
Autoconfiguring devices... Done.
  Mouse is Generic 3 Button Mouse (PS/2) at /dev/psaux
  Soundcard: CM8738, driver=cmpci
  AGP bridge detected.
  Video is ATI|Radeon 7500 QW, using XFree86(radeon) Server
  Monitor is LTN 020e, H:31-60kHz, V:55-75Hz
  Using Modes "1024x768" "800x600" "640x480"
Enabling DMA acceleration for: hde.
Scanning for Harddisk partitions and creating /etc/fstab... Done.
  Network device eth0 detected, DHCP broadcasting for IP.
  (Backgrounding)
  Automounter started for: floppy CD-ROM CD-ROM1.
INIT: Entering runlevel: 2
root@tty1[/]#
```

Ja som dins del sistema. En aquest cas no ha fet falta ni usuari ni contrasenya, hem entrat com a *root* directament; ho notem perquè el *prompt* acaba amb el caràcter "#". Per a qualsevol altre usuari diferent del de *root*, l'últim caràcter seria "\$".

3.3. Aturada del sistema

Un cop dins, el primer que hem de saber, tal com ja s'ha remarcat, és com aturar-lo. Recordem una vegada més que no podem parar l'ordinador sense haver aturat abans el sistema operatiu. Hi ha múltiples maneres de fer-ho. Les més comunes són: polsar la combinació de tecles CTRL + ALT + SUPR o mitjançant el comandament `halt` (n'hi ha moltes més, mitjançant el comandament `reboot`, canviant de `runlevel` a 0 o 6, etc.). Una vegada hàgim donat l'ordre al sistema perquè s'aturi, aquest començarà a executar les instruccions pertinents d'aturada (desmuntar dispositius, aturar processos, etc.), i al final de tot aquest procés, KNOPPIX expulsarà el CD-ROM i ens demanarà que polsem INTRO per parar l'ordinador.

3.4. Configuració del teclat

Una vegada sabem com aturar el sistema i tornem a ser-hi seguint els mateixos passos d'arrencada anteriors, el primer que hem de fer és configurar el mapatge del teclat correctament. Hi ha dues maneres de fer-ho, manualment:

```
root@tty1[/]#loadkeys /usr/share/keymaps/i386/qwerty/es.kmap.gz
```

o bé gràficament amb el comandament `kbdconfig` i escollint l'opció "es es".

És important recordar que sempre tenim l'opció d'autocompletar polsant "Tab" una vegada, si només hi ha una opció possible, i que s'autocompleti; o dues vegades seguides, si hi ha més d'una opció, i que se'ns mostrin les possibilitats. Ho podem provar amb el comandament `loadkeys` mateix. Teclegem només una "1" i polsem TAB una vegada, i el sistema emet un xiulet; i una segona per obtenir el següent:

```
root@tty1[/]# 1
Display all 143 possibilities? (y or n)
```

En aquest cas no volem que se'ns mostrin les cent quaranta-tres possibilitats. Polsem, doncs, "n", hi afegim una "o" i repetim l'operació anterior de polsar dues vegades TAB. Ara obtenim un resultat diferent:

```
root@tty1[/#]# lo
loadkeys          locale-gen      lockfile-create  logger          logredo         lookbib
loadshlib         localedef       lockfile-remove  login           logresolve      lorder
local             locate          lockfile-touch   logname         logrotate       losetup
locale            lockfile        logdump          logout          look
```

```
root@tty1[/#]# lo
```

Veiem que només falta afegir "adk" i tornar a polsar TAB perquè funcioni l'autocompletat i obtenim:

```
root@tty1[/#]# loadkeys
```

L'autocompletat també serveix per a fer referència a fitxers i directoris. N'hi ha prou de teclejar "loadkeys /u" i polsar TAB per a obtenir:

```
root@tty1[/#]# loadkeys /usr
```

i tornar a polsar TAB per a obtenir:

```
root@tty1[/#]# loadkeys /usr/
```

L'autocompletat és una eina molt útil, no solament perquè estalvia teclejar, sinó perquè a més serveix per a verificar que hem escrit correctament tant els comandaments com els directoris i els fitxers. És una eina l'absència de la qual es fa notar una vegada ens hem acostumat a usar-la.

3.5. Inspecció del sistema

Una vegada tenim el teclat configurat, podem procedir a inspeccionar una mica el sistema. En primer lloc, a quin lloc de l'estructura de

directoris ens trobem actualment? Ho podem saber mitjançant el comandament `pwd`:

```
root@tty1[ / ]# pwd
/
```

Som a l'arrel del sistema, ho sabem pel retorn del comandament `pwd`, però també ho hauríem pogut saber llegint la informació que ens dóna el *prompt*: "`root@tty1[/]#`". Anteriorment ja hem entès el significat del caràcter "#", coneguem ara la resta d'informació que ens dóna:

"`root`": en aquest camp apareix el nom d'usuari, `root` en aquest cas. Aquesta informació és una mica redundant, ja que el caràcter "#" indica el mateix, tot i que per a la resta d'usuaris és interessant.

Després del caràcter "@", que simplement serveix per a separar camps, hi ha "`tty1`", el qual ens indica a quina terminal ens trobem, i és que, tal com ja s'ha dit, Linux és multiusuari i multiprocés; així doncs, no és estrany que puguem accedir al sistema des de diversos terminals. Concretament KNOPPIX ofereix, per defecte, quatre terminals, als quals podem accedir mitjançant les combinacions de tecles ALT + F1 (`tty1`), ALT + F2 (`tty2`), ALT + F3 (`tty3`) i ALT + F4 (`tty4`); això és extremadament útil, ja que ens permet tenir fins a quatre sessions de treball obertes simultàniament i poder provar, per exemple, un comandament en una, en una altra tenir el *man* del comandament en qüestió per a llegir-ne les opcions, mentre que en una altra podem tenir un editor obert per a prendre nota del que fem (moltes distribucions ofereixen sis terminals mode text i reserven el setè per al terminal gràfic; en KNOPPIX el terminal gràfic, si s'ha habilitat, i no és el nostre cas, es trobaria en el cinquè terminal). Així doncs, provem la combinació ALT + F2: observem ara que en el camp que estudiem apareix "`tty2`", i executem un comandament, per exemple "`man man`", per a llegir el *man* de l'aplicació `man`.

De moment la deixem allà i tornem a la `tty1`, i comprovem que tot està com ho havíem deixat.

Finalment, i continuant amb la interpretació del *prompt*, abans del camp ja conegut "#" trobem "[/]": aquí se'ns indica el directori ac-

tual, l'arrel en aquest cas, tal com ens havia informat el comandament `pwd`. La diferència és que el *prompt* només ens informa del directori actual, mentre que `pwd` ens torna la ruta completa.

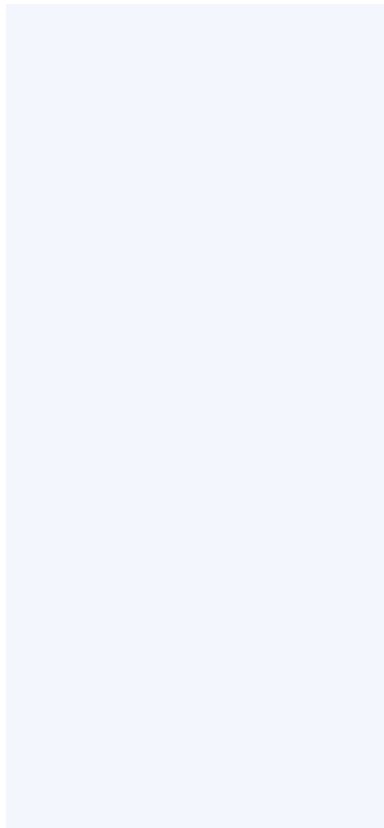
Podem fer ara una llista dels continguts del directori en què som. Per a fer-ho, utilitzarem el comandament `ls`, primer sense passar-li cap opció, i després passant-li'n dues per a indicar-li que ens torni informació més detallada (normalment les opcions van precedides de "-", encara que hi ha comandaments que no ho exigeixen; a més a més, una vegada posat el "-" es poden concatenar com es fa en l'exemple):

```
root@tty1[/]# ls
KNOPPIX bin boot CD-ROM dev etc home lib mnt opt proc
ramdisk sbin tmp usr var
root@tty1[/]# ls -la
total 45
drwxr-xr-x  9          root    root    1024 Mar  2  18:37 .
drwxr-xr-x  9          root    root    1024 Mar  2  18:37 ..
drwxr-xr-x 20          root    root    4096 Jan 18  18:05 KNOPPIX
lrwxrwxrwx  1          root    root     12 Jan 18  17:58 bin -> /KNOPPIX/bin
drwxr-xr-x  9          root    root    1024 Mar  2  18:37 .
lrwxrwxrwx  1          root    root     13 Jan 18  17:58 boot -> /KNOPPIX/boot
dr-xr-xr-x  5          root    root    2048 Jan  4  03:34 CD-ROM
drwxr-xr-x  3          root    root    29696 Mar  2  17:58 dev
drwxr-xr-x 139          root    root    6144 Mar  2  17:49 etc
lrwxrwxrwx  1          root    root     13 Mar  2  18:37 home -> /ramdisk/home
lrwxrwxrwx  1          root    root     12 Jan 18  17:58 lib -> /KNOPPIX/lib
drwxr-xr-x 13          root    root    1024 Mar  2  17:38 mnt
lrwxrwxrwx  1          root    root     12 Jan 18  17:58 opt -> /KNOPPIX/opt
dr-xr-xr-x 30          root    root     0 Mar  2  18:37 proc
drwxrwxrwt  4          root    root     80 Mar  2  18:37 ramdisk
lrwxrwxrwx  1          root    root     13 Jan 18  17:58 sbin -> /KNOPPIX/sbin
lrwxrwxrwx  1          root    root     8 Jan 18  17:58 tmp-> /var/tmp
lrwxrwxrwx  1          root    root     12 Jan 18  17:58 usr -> /KNOPPIX/usr
```

Fixem-nos en els diferents colors amb què se'ns mostren els resultats (podem observar els colors esmentats al nostre PC a mesura que anem progressant en el taller de KNOPPIX): blau marí per als direc-

toris; magenta per als enllaços simbòlics, la destinació dels quals se'ns mostra després de la combinació de caràcters “->”; verd per als *scripts* i executables (encara que, òbviament, en el directori arrel no n'esperem trobar cap, ja que, tal com hem vist, en UNIX l'ordre dins del sistema de fitxers és molt rígid), etc.

Entrem en algun directori on segur que trobarem executables, per exemple `/usr/bin/` (per a fer-ho, executem “`cd /usr/bin/`”). En el cas d'aquesta distribució, el directori `/usr/` és, en realitat, un enllaç simbòlic sobre `/KNOPPIX/usr/`; podríem accedir, doncs, als seus continguts entrant-hi directament mitjançant el comandament `cd`, o bé seguir mitjançant el mateix comandament la seva destinació real. Tanmateix, escollim la primera opció, ja que quan ens trobem en un sistema UNIX instal·lat en un disc dur aquest directori serà un directori real:



```
root@tty1[bin]# ls -la
total 171831
drwxr-xr-x      2 root      root      311296 Jan 18 18:04 .
drwxr-xr-x     14 root      root       2048 Aug 17 2002 ..
-rwxr-xr-x      1 root      root       5380 Apr 1 2002 411toppm
-rwxr-xr-x      1 root      root       1649 Sep 15 20:36 822-date
-rwxr-xr-x      1 root      root       1971 Jun 5 2002 AbiWord
.
.
.
-rwxr-xr-x      1 root      root       17992 Sep 16 22:34 zsoelim
-rwxr-xr-x      1 root      root        966 Sep 4 13:53 zxpdf
```



Probablement no hàgim pogut veure més que les últimes línies que han creuat la pantalla. Podem visualitzar alguns resultats més polsant SHIFT + REPAG per a retrocedir en la llista i SHIFT + AVPAG per a avançar; tot i així, la memòria intermèdia del terminal (*buffer*) té un límit, i probablement tampoc no podrem visualitzar, mitjançant aquesta tècnica, tota la informació que ens ha tornat el comandament `ls`. Hem de recórrer, doncs, a altres tècniques, el redireccionament de la sortida (en aquest cas sobre un fitxer) o l'ús de *pipes* “|” i un paginador (`less` en aquest cas, que usa les tecles REPAG i AVPAG per a desplaçar el contingut mostrat per pantalla, i la tecla “q” per a sortir).

ANOTACIONS

Utilitzarem aquesta última, ja que no ens interessa desfer una llista del contingut d'un directori; aprofitarem, així mateix, per a posar en pràctica una altra utilitat que ens estalviarà teclejar: polsant els cursors d'amunt i avall ens podem desplaçar per totes les línies de comandaments que hem passat al sistema; així doncs, per a obtenir la línia desitjada, `ls -la | less`, n'hi haurà prou de polsar una vegada el cursor d'amunt i afegir-hi `| less`.

`less` és un paginador molt potent que KNOPPIX, i la majoria de distribucions, usa per a mostrar els continguts dels `man` (encara que, com la majoria de serveis en Linux, el podem configurar al nostre gust). Així doncs, podem fer un `man less` per informar-nos una mica sobre aquest comandament, i de passada acostumar-nos a la morfologia d'aquesta ajuda. Òbviament, atès que som sota `less`, per a sortir de `man` n'hi ha prou de prémer la tecla "q".

Fixem-nos que en totes les llistes de continguts de directoris que hem anat fent sempre apareixen, al principi de tot, dos directoris una mica especials: `."` i `..`; en crear directoris, aquests són creats automàticament pel sistema com a subdirectoris del directori creat, i contenen informació molt especial: `."` fa referència al directori mateix (de manera que, si fem, per exemple, un `ls -la .`, obtindrem la mateixa informació que fent un `ls -la`) i `..` fa referència al directori pare (fent un `ls -la ..` obtindrem la llista de continguts del directori pare).

Activitat

5. Feu un `man` del comandament `ls` i intenteu comprendre quines funcions tenen els paràmetres `-a` i `-l`.

Recordeu on som dins del sistema de fitxers? No? Doncs el comandament `pwd` us ho recordarà. Torneu al directori arrel. Hi ha bàsicament dues maneres de fer-ho, i una altra d'especial per a aquest cas; de tota manera, totes es basen en l'ús del comandament `cd`, i es diferencien entre elles per l'argument que li passem en cada un dels casos. La primera manera d'anar-hi és tornant als directoris pare pas a pas, mitjançant `cd ..` (atenció, el sistema no entén `cd..`, ja que interpreta que intentem executar un comandament anomenat `cd..`, el qual no existeix), iterativament fins a arribar al directori arrel (el veurem en el `prompt`). La segona és més eficaç, ja que amb una sola línia de comandaments aconseguirem el nostre propòsit,

anar al directori desitjat, passant-lo com a argument al comandament `cd`, "`cd /`" en aquest cas; aquesta segona opció és molt més potent, ja que permet passar directament a directoris que pertanyen a diferents branques, per la qual cosa podríem haver fet, per exemple, "`cd /etc/rcS.d/`" (l'última barra és opcional) per anar a aquest directori. Com hem dit, per a aquest cas hi ha una tercera opció per aconseguir la mateixa finalitat amb variacions: es tracta d'executar `cd` sense passar-li cap argument, i ens situaríem directament a l'arrel del sistema, ja que aquest és, en el cas de KNOPPIX (cas excepcional i una mica peculiar, ja que en la majoria de distribucions, i segons el Filesystem Hierarchy Standard (<http://www.pathname.com/fhs>), `root` ha de tenir un inici o home propi `/root`), el directori d'usuari de `root`; una variació d'aquest mètode és "`cd ~`", ja que "`~`" és equivalent al directori d'inici de l'usuari, el directori arrel en aquest cas. Encara hi ha una altra opció per tornar al directori arrel, o més ben dit, per tornar al directori del qual venim, l'arrel, ja que havíem executat "`cd /usr/bin`", i que és "`cd -`", ja que a "`-`" es desa l'últim directori on hem estat abans de l'actual.

3.6. Maneig de directoris i fitxers

Després d'examinar diferents opcions per a fer el mateix, ens trobem encara en el directori arrel. Podem aprendre ara a crear directoris i fitxers, moure'ls, copiar-los, esborrar-los, etc. Ho farem sobre el disquet que hem introduït en engegar KNOPPIX, i que el sistema mateix, en arrencar, l'ha automuntat (més endavant aprendrem com muntar i desmuntar dispositius). Recordem que en UNIX abans de poder usar els dispositius s'han de muntar, i que abans de retirar-ne el suport han de ser desmuntats. Aquest últim punt és extremadament important, ja que si no, la integritat de les dades no està en absolut garantida. Podem, per exemple, abans de procedir a treballar sobre el disquet, intentar retirar el CD-ROM prement el botó d'expulsió. Oh, sorpresa!, no es pot; no pensem que s'ha espatllat el dispositiu i menys que ho ha fet Linux. Simplement el que succeeix és que aquest dispositiu també ha estat muntat automàticament pel sistema durant l'arrencada, i que, per tant, ha passat a formar-ne part; no tindria cap sentit que poguéssim retirar el CD-ROM sense abans informar-ne el sistema perquè aquest el pugui desmuntar correctament; així doncs, aquest ha pres el control d'aquest dispositiu i, entre altres coses, ha deshabilitat l'efecte del botó d'expulsió del CD-ROM, preci-

sament a fi que accidentalment aquest sigui retirat sense que abans hagi estat desmuntat. No passa el mateix amb la unitat de disquets: en aquest dispositiu l'expulsió es fa de manera purament mecànica, així que és impossible que el sistema pugui impedir que nosaltres expulem el disquet sense informar-lo abans, però això no és en absolut recomanable, tal com ja s'ha dit, perquè pot representar la pèrdua de totes les dades que pogués contenir. Així doncs, mentre no aprenguem a muntar i desmuntar dispositius de suport de dades, el disquet ha de romandre a la unitat de disquets des que el sistema arrenca fins que s'atura, ja que durant el procés de parada, entre altres operacions, s'efectuaran les de desmuntat d'aquests dispositius. Anem al disquet: però, on és?, on l'ha muntat el sistema? Per a respondre aquestes preguntes executem el comandament `mount` sense arguments ni opcions addicionals (precisament aquest és el comandament que s'utilitza per a muntar dispositius, i `umount` per a desmuntar-los, tal com veurem més endavant):

```
root@tty1[/# mount
/dev/root on / type ext2 (rw)
/dev/CD-ROM on /CD-ROM type iso9660 (ro)
/dev/cloop on /KNOPPIX type iso9660 (ro)
/dev/shm on /ramdisk type tmpfs (rw,size=407928k)
none on /proc/bus/usb type usbdevfs (rw,devmode=0666)
automount(pid443) on /mnt/auto type autofs (rw,fd=6,pgrp=443,minproto=2,maxproto=4)
/dev/fd0 on /mnt/auto/floppy type vfat (rw,nosuid,nodev,uid=1000,gid=1000,umask=000)
```

Aquest comandament, cridat *sense arguments*, ens mostra els dispositius muntats en el sistema en el moment d'executar-lo juntament amb alguna informació addicional sobre ells (aquesta mateixa informació es pot trobar en el fitxer `/etc/mntab`; en conseqüència, fent un `cat /etc/mntab` aconseguiríem els mateixos resultats). En la segona línia retornada pel comandament ja podem veure que, efectivament, el CD-ROM `/dev/CD-ROM` està muntat en el sistema, i a més podem veure que està muntat sobre el directori `/CD-ROM/`. L'última línia retornada respon les preguntes que ens formulàvem anteriorment: el sistema ha muntat el disquet, `/dev/fd0/`, a `/mnt/auto/floppy/`. Canviem, doncs, a aquest directori per començar a treballar sobre el disquet i comprovem que efectivament sigui buit:

```
root@tty1[/# cd /mnt/auto/floppy
root@tty1[floppy]# ls -la
total 8
drwxrwxrwx 3 knoppix knoppix 7168 Jan 1 1970 .
drwxr-xr-x 3 root root 0 Mar 3 19:34 ..
```

Creem el nostre primer directori, hi entrem i hi creem un parell de subdirectoris:

```
root@tty1[floppy]# mkdir dir00
root@tty1[floppy]# cd dir00/
root@tty1[dir00]# mkdir subdir00 subdir01
root@tty1[dir00]# ls
subdir00 subdir01
```

Entrem al primer subdirectori i creem el nostre primer fitxer:

```
root@tty1[dir00]# cd subdir00
root@tty1[subdir00]# touch file00
root@tty1[subdir00]# ls -la
total 1
drwxrwxrwx 2 knoppix knoppix 512 Mar 3 20:21 .
drwxrwxrwx 4 knoppix knoppix 512 Mar 3 20:21 ..
-rwxrwxrwx 1 knoppix knoppix  0 Mar 3 20:21 file00
```

En realitat *touch* no serveix per a crear fitxers buits, encara que ho faci si el fitxer no existeix, sinó que serveix per a canviar les informacions relatives a dates i hores dels arxius. Podem posar algun contingut al nostre primer fitxer i comprovar que efectivament ha quedat registrat:

```
root@tty1[subdir00]# echo "el meu primer hello world en Linux" > file00
root@tty1[subdir00]# cat file00
el meu primer hello world en Linux
```

Hi afegim una mica més de text:

```
root@tty1[subdir00]# echo "Molt de gust de conèixer-te, serem bons amics" >> file00
root@tty1[subdir00]# cat file00
el meu primer hello world in Linux
Molt de gust de conèixer-te, serem bons amics
```

No hem d'oblidar d'usar els cursors per a estalviar-nos teclejar, ni l'autocompletat. Ara podem utilitzar un editor de text per a crear el nostre segon fitxer. Per a això usarem el *vi*. Recordem que aquest editor té dos modes: el de comandaments, en el qual s'entra quan s'engega, i el mode d'edició. Per a entrar al mode d'edició n'hi ha

prou de prémer la tecla “i”, i per a passar al mode de comandaments cal pulsar ESC. Una vegada en mode comandament “w” per a desar i “q” per a sortir (òbviament, vi disposa de molts més comandaments, però per ara amb aquests dos n’hi ha prou). És molt interessant conèixer aquests comandaments bàsics de vi, ja que aquest editor l’incorporen gairebé tots els paquets bàsics d’instal·lació de qualsevol distribució, i si aquesta fallés en algun moment, ens podria ser útil per a modificar algun fitxer i prosseguir aquesta instal·lació. Executem, doncs, vi seguit del nom que volem donar al nostre segon fitxer, premem la tecla “i”, escrivim el que ens sembli oportú, polsem “ESC” i “:wq” per a desar i sortir. Mitjançant more, un altre paginador, comprovem que tot ha anat com esperàvem:

```
root@tty1[subdir00]# vi file01
sembla que estem en el camí correcte, amic!!!
Hi estic d'acord.
~
~
~
:wq
root@tty1[subdir00]# more file01
sembla que estem en el camí correcte, amic!!!
Hi estic d'acord.
```

Ara intentem esborrar el nostre primer fitxer. Ho farem mitjançant el comandament rm:

```
root@tty1[subdir00]# rm file00
rm: remove regular file 'file00'?
```

El cert és que aquesta pregunta per part del sistema no ens l’esperàvem. Polsem CTRL + C per cancel·lar, “n” per a no esborrar o “y” per a fer-ho. L’important ara és que entenguem per què el sistema ens ha formulat aquesta pregunta. Si llegim el man del comandament rm veurem que aquest hauria d’esborrar l’arxiu sense fer cap pregunta (recordem que podem executar aquest man en una altra tty, i que en la tty2 encara tenim obert el man de man, si no n’hem sortit mitjançant “q”, o si no hem apagat l’ordinador des que l’hem invocat). Però fixem-nos en l’explicació de l’opció “-i” d’aquest mateix man. Què passa? Sembla com si aquesta opció estigués activada per defecte. En realitat no és exactament així, el que passa és que el sistema per

Nota

Vegeu l’apèndix B per aprendre més coses sobre el vi.

Nota

En sistemes bàsics tipus UNIX, si no es té vi es pot provar si es troben instal·lats altres editors de text més elementals com poden ser nano, pico, etc.

defecte hi ha establert uns àlies; per a veure'ls tots, executem el comandament `alias`:

```
root@tty1[etc]# alias
alias ..='cd ..'
alias cp='cp -i'
alias l='ls -a --color=auto'
alias la='ls -la --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias mv='mv -i'
alias rm='rm -i'
alias where='type -all'
alias which='type -path'
```

Aquí comencem a entendre moltes més coses, com per exemple per què el retorn del comandament `ls` és en colors, o que per a fer un “`ls -la`” n’hi ha prou de teclejar `la`. I també entenem per què, quan hem executat l’`rm` anterior, el sistema ens ha preguntat si realment volíem fer-ho. Mitjançant l’`alias` podem establir i modificar el comportament per defecte de comandaments o fins i tot crear-ne d’altres de nous:

```
root@tty1[subdir00]# alias hi='echo "et dic hola"'
root@tty1[subdir00]# hi
et dic hola
```

Continuant amb el `man` del comandament `rm`, ens disposem a llegir per a què serveixen les opcions `-f` i `-r`. La primera serveix perquè el comandament s’executi sense mostrar *prompt*, o el que és el mateix, desobeint l’opció `-i`, així que, si no havíem esborrat el nostre primer fitxer, ho podem fer ara mitjançant “`rm -f file00`”. La segona opció força la recursivitat, és a dir, que l’ordre s’estengui sobre els possibles subdirectoris i els seus continguts. Aquestes opcions són comunes a la majoria de comandaments bàsics destinats a la manipulació de directoris i fitxers; així doncs, podem crear un segon directori en l’arrel del disquet amb tots els continguts del primer que hem creat mitjançant el comandament `cp`; per a fer-ho, hem d’acudir a la recursivitat:

```
root@tty1[subdir00]# cp /mnt/auto/floppy/dir00/ /mnt/auto/floppy/dir01 -r
```

En aquest cas hem usat el direccionament absolut –tant per a especificar l’origen com la destinació– per a fer l’operació; és a dir, hem indicat, partint del directori arrel, la ruta completa, tant per a l’origen com per a la destinació. De la mateixa manera, podríem haver utilitzat el direccionament relatiu per a especificar l’origen de l’operació, la seva destinació o ambdues coses. Quan usem el direccionament relatiu, l’origen del direccionament és la posició actual dins del filesystem. Com la majoria de vegades, el sistema ens ofereix la possibilitat d’obtenir els mateixos resultats emprant mètodes diferents. Es tracta de conèixer-ne com més millor, i saber escollir el més efectiu per a cada cas en particular. Així doncs, hauríem obtingut el mateix resultat fent “`cp ../../dir00/ ../../dir01 -r`”, és a dir, comunicant a `cp` que l’origen és el directori `/dir00/`, el qual es troba dues branques per sota de la nostra posició actual, i que la destinació és `/dir01/`, al que també volem situar dues branques per sota de la nostra posició. Així mateix, hauria estat perfectament vàlida la línia “`cp -r ../ ../ ../dir02/`” per a aconseguir els mateixos objectius.

Activitat

6. Expliqueu el perquè d’aquesta última asseveració.

Ara podem descendir un subdirectori, amb la qual cosa ens situarem a `/mnt/autofloppy/dir00/` i podrem copiar el segon fitxer que hem creat en el subdirectori `/subdir00` en aquest mateix directori:

```
root@tty1[dir00]# cp subdir00/file01.
```

Hem d’entendre completament el significat i el perquè de la línia anterior: en primer lloc especifiquem l’origen del fitxer que volem copiar (també hauria estat vàlid, entre moltes altres opcions, “`./subdir00/file00`”), i en segon lloc cal especificar obligatòriament la destinació, la qual és la posició actual, és a dir “`./`”. Mitjançant un `ls` podem comprovar si hem obtingut el resultat que volíem. Ara ens podem situar al directori pare, on es troba muntat el disquet, i esborrar tot el que hem generat fins ara. Per a fer-ho, utilitzarem el *wildcard* “`*`” (existeix també el *wildcard* “`?`”, que serveix per a un sol caràcter, així com diferents mètodes per a especificar rangs de caràcters, i per a referir-se, en general, a més d’un fitxer o directori) per a estalviar teclejar:

```
root@tty1[floppy]# rm -rf *
```

Nota

Vigileu en executar el comandament “`rm -rf*`” com a `root` ja que pot tenir conseqüències nefastes.

Activitat

7. Llegiu el man d'rm i intenteu comprendre per què "rm -rf*" és tan perillós.

3.7. Administració d'usuaris

Ens ha d'alarmar el fet d'haver estat fent totes les tasques anteriors com a *root*, ja que ja tenim molt clar que el mode de superusuari només s'ha d'utilitzar en cas que sigui estrictament necessari.

No obstant això, queda justificat, ja que fins ara no havíem tingut cap contacte directe amb un sistema UNIX, i això ens ha servit per a familiaritzar-nos una mica amb els comandaments bàsics i amb el seu sistema de fitxers. Ha arribat el moment de començar a treballar com s'ha de fer sempre a partir d'ara, és a dir, utilitzant el compte de *root* només quan no hi hagi una altra possibilitat, com en el primer pas que farem a continuació, crear un nou usuari del sistema i assignar-li un compte, ja que aquesta operació només la pot fer el *root*. Procedim, doncs, a crear un usuari.

```
root@tty1[etc]# useradd user00
```

Hem creat el nostre primer usuari, però el procés ha estat una mica opac, ja que el sistema no ens ha retornat cap informació respecte a aquest comandament que acabem d'executar. No sabem, doncs, ni a quin grup o grups pertany aquest usuari, on té aquest el directori d'inici (*home*), quin intèrpret de comandaments se li ha assignat per defecte, etc. Molta d'aquesta informació la trobarem al fitxer de contrasenyes, `/etc/passwd`, així que n'analitzem, per camps, el contingut:

```
root@tty1[/#]# cd /etc
root@tty1[etc]# cat passwd
root:x:0:0:root:/home/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
.
.
.
partimag:x:104:65534:./home/partimag:/bin/false
user00:x:1001:100:./home/user00:/bin/bash
```

La informació que volem la trobem en l'última línia del fitxer, ja que el nostre usuari ha estat l'últim que s'hi ha afegit. Analitzem el contingut d'aquesta línia:

- `user00` és el nom de l'usuari, que ja coneixíem, ja que l'hem creat nosaltres mateixos.
- `x` és la seva contrasenya es troba al fitxer de shadow, `/etc/shadow`.
- `1001` és el seu número d'identificació com a usuari (UID).
- No té cap comentari associat.
- `/home/user00` és el seu directori de l'usuari (*home*), tal com era d'esperar, ja que tots els usuaris tenen el seu inici (*home*) a `/home/<usuari>`.
- `/bin/bash` l'interpret de comandaments amb el qual l'usuari es trobarà en iniciar la sessió és el bash, el mateix que hem utilitzat fins ara.

La pregunta que ens fem immediatament davant d'aquestes informacions és relativa a la contrasenya de l'usuari. Hem de consultar el fitxer de shadow, ja que així ho indica el fitxer `/etc/passwd`. De passada, introduïrem un concepte nou, el de filtre, que en aquest cas es materialitza amb el comandament `grep` (*general regular expression processor*). Els filtres són eines extremadament potents i molt usades en UNIX, i el seu camp d'acció s'estén molt més enllà del que comunament s'entén per filtre. En aquest cas, passarem, mitjançant un pipe, la sortida de `cat` a `grep`, perquè aquest mostri per pantalla aquelles línies que continguin la paraula que li passem com a paràmetre, `user00`:

```
root@tty1[etc]# cat shadow | grep user00
user00:!:12115:0:99999:7:::
```

A partir d'ara no ens hem de preocupar més: el segon camp ens indica que el nostre usuari no té contrasenya. Si n'hagués tingut, no la podríem conèixer, ja que tal com s'ha dit, les contrasenyes es desenxifrades i el xifratge és unidireccional. Però això tampoc no ens hauria de preocupar, ja que `root`, mitjançant el comandament `passwd`,

pot canviar la contrasenya de qualsevol altre usuari. Així doncs, l'únic que realment hem d'intentar és no oblidar mai la contrasenya de `root`, que tampoc no té contrasenya en aquest cas, tal com `/etc/shadow` ens indica, encara que si això succeís, també té solució. Continuem amb la nostra investigació sobre les propietats del nostre nou usuari. `/etc/group` ens ajudarà a saber exactament a quins grups pertany:

```
root@tty1[etc]# cat group | grep user00
root@tty1[etc]# cat group | grep 100
users:x:100:knoppix
knoppix:x:1000:
```

Amb la primera línia de comandaments ens preguntem, amb resposta negativa, si `/etc/group` té alguna referència explícita a `user00` (també es podria haver fet mitjançant el comandament `groups`, "`groups user00`"). Amb la segona busquem el revers del grup 100, grup primari de `user00` segons `/etc/passwd`, grup que es diu `users` en aquest cas. Així doncs, ja podem afirmar que el nostre nou usuari pertany a un sol grup, i que aquest és `user`. El fitxer `/etc/group` també es pot editar per a afegir usuaris a grups i crear grups nous, però cal remarcar que en cap cas en el camp d'usuaris pertanyents a un grup en concret no es pot posar el nom de cap altre grup amb la intenció d'afegir tots els usuaris d'aquest últim al primer. Però aquesta pràctica no és massa recomanable, ja que existeixen comandaments específics per a treballar amb grups (`newgrp`, `addgroup`, etc.).

Queda només una última qüestió per resoldre: si fem una llista dels continguts de `/home/`, veurem que el subdirectori `/user00/` no existeix. L'hem de crear, doncs, i assignar-li les propietats i atributs pertinents manualment:

```
root@tty1[etc]#mkdir /home/user00
root@tty1[etc]#chown user00 -R /home/user00
root@tty1[etc]#chgrp users -R /home/user00
root@tty1[etc]# cd /home
root@tty1[home]# ls -la
total 0
drwxr-xr-x 5 root    root    100 Mar  4  08:12 .
drwxrwxrwt 4 root    root     80 Mar  4  08:35 ..
drwxr-xr-x 2 knoppix knoppix  40 Mar  4  08:35 knoppix
```



```
drwxr-xr-x 2 root    root      40 Mar 4 08:35 root
drwxr-xr-x2 user00 users 60Mar408:12user00
```

Activitat

8. Mitjançant `man`, compreneu el funcionament dels comandaments `chown` i `chgrp`.

Ha arribat el moment d'entrar al sistema com a usuari nou. Ho farem mitjançant el comandament `su`, el qual obre un procés fill amb una connexió amb l'usuari que li hàgim passat. Com a `root` podem utilitzar sempre aquest mecanisme per a entrar com un altre usuari sense necessitat de conèixer la seva contrasenya, ja que en executar `su` com a `root` aquesta no és necessària; a més, es dóna el cas que l'usuari que hem creat no té contrasenya, i per tant mai no serà demanada. Però si provéssim d'executar `su` en qualsevol altra circumstància sí que se'ns demanaria la contrasenya de l'usuari:

```
root@tty1[home]# su user00
su(pam_unix)[4312]: session opened for user user00 by (uid=0)
user00@tty1[home]$
```

Notem primerament el canvi d'aspecte del *prompt*. Hem deixat de ser usuari `root`, per passar a ser `user00`, per la qual cosa hem perdut els privilegis de `root`, fet que es manifesta pel canvi de l'últim caràcter del *prompt*. Podem accedir ara a entrar al nostre directori d'usuari, hi creem un fitxer i li canviem els atributs perquè tan sols `user00` el pugui llegir, executar i escriure-hi. De moment, el permís d'execució no té gaire sentit per a nosaltres, ja que no sabem encara crear fitxers executables, però és bo conèixer-lo per endavant per quan sigui necessari utilitzar-lo:

```
user00@tty1[home]$ cd
user00@tty1[user00]$ echo "només user00 pot llegir, escriure i executar aquest fitxer." > user00file
user00@tty1[user00]$ chmod 700 user00file
```

Nota

El procés d'addició d'usuaris nous al sistema mitjançant Debian és molt més senzill, com es veurà més endavant. Però crear-ne un mitjançant KNOPPIX no ha estat una tasca en va, ja que ens ha servit per a familiaritzar-nos amb el sistema i aprendre comandaments nous.

Activitat

9. Creeu un usuari nou seguint els passos anteriorment descrits i comproveu que, efectivament, aquest usuari nou no pot llegir el fitxer creat nou ni escriure-hi.

3.8. Gestió de processos

UNIX es caracteritza per una excel·lent gestió dels processos que s'executen sobre el sistema. En primer lloc hem d'aprendre a detectar quins processos corren sobre el sistema i les seves particularitats (manera com corren, recursos que consumeixen, qui els executa, etc.). Executarem el comandament `ps` (*process status*) passant-li diversos paràmetres per a veure com incideixen sobre la informació tornada pel comandament:

```
root@tty1[/# ps
PID TTY TIME CMD
481 tty1 00:00:00 bash
          1559 tty1      00:00:00 ps
```

Sense arguments `ps` ens informa sobre els processos que corren sobre el terminal on s'executa; naturalment el primer procés sempre correspondrà a l'interpret de comandaments.

Podríem manar executar un procés en *background*, `sleep`, per exemple (procés que simplement espera que transcorri el nombre de segons que li passem com a paràmetre per a acabar) i observar el retorn de `ps`:

```
root@tty1[/# sleep 300 &
[1] 1703
root@tty1[/# ps
  PID TTY      TIME CMD
  481 tty1  00:00:00 bash
 1703 tty1  00:00:00 sleep
 1705 tty1  00:00:00 ps
```

Ens podem preguntar ara per tots els processos que corren sobre el sistema:

```
root@tty1[/# ps aux
USER    PID    %CPU   %MEM   VSZ    RSS    TTY    STAT   START   TIME   COMMAND
root     1      0.4    0.0    72     72     ?      S      15:41   0:07   init [2]
root     2      0.0    0.0     0      0      ?      SW     15:41   0:00   [keventd]
root     3      0.0    0.0     0      0      ?      SWN    15:41   0:00   [ksoftirqd_CPU0]
root     4      0.0    0.0     0      0      ?      SW     15:41   0:00   [kswapd]
root     5      0.0    0.0     0      0      ?      SW     15:41   0:00   [bdflush]
root     6      0.0    0.0     0      0      ?      SW     15:41   0:00   [kupdated]
root    52     0.0    0.0     0      0      ?      SW     15:41   0:00   [kapmd]
root    59     0.0    0.0     0      0      ?      SW     15:41   0:00   [khubd]
root   433     0.0    0.1   1368   616    ?      S      15:41   0:00   pump -i eth0
root   475     0.0    0.1   1316   596    ?      S      15:41   0:00   /usr/sbin/automount
root   481     0.0    0.3   2864  1988   tty1    S      15:41   0:00   /bin/bash -login
root   482     0.0    0.3   2864  1988   tty2    S      15:41   0:00   /bin/bash -login
root   483     0.0    0.3   2856  1952   tty3    S      15:41   0:00   /bin/bash -login
root   484     0.0    0.3   2856  1976   tty4    S      15:41   0:00   /bin/bash -login
root   2086    0.0    0.3   3436  1552   tty1    R      16:06   0:00   ps aux
```

També pot ser interessant en determinades situacions executar el comandament `top`, el qual ens mostra l'activitat de la CPU, l'ús de memòria, etc. interactivament.

Ara engegarem un procés, el detindrem i el farem executar en *background*. Per a fer-ho, farem un `sleep 20` i polsarem la combinació de tecles `CTRL + Z` abans de vint segons per a parar-lo, `jobs` per a assegurar-nos que efectivament s'ha aturat i conèixer el seu número d'identificació, i `bg` juntament amb el seu número d'identificació per a fer-lo executar en manera *background*:

```
root@tty1[/# sleep 20
<Ctrl+Z>
[1]+  Stopped                  sleep 20
\intro

root@tty1[/# jobs
[1]+  Stopped                  sleep 20
root@tty1[/# bg 1
[1]+  sleep 20 &
[1]+  Done                    sleep 20
root@tty1[/#
```

Tenim que, després d'haver transcorregut vint segons, el procés ha acabat estant encara en *background*, i se'ns ha informat per pantalla. Podem comprovar mitjançant `ps` que efectivament no hi ha cap procés corrent. En aquest punt, podem parar un procés i tornar-lo a *foreground* mitjançant `fg`:

```
<Ctrl+z>
[1]+  Stopped                  man ls
root@tty1[/]# jobs
[1]+  Stopped                  man ls
root@tty1[/]# bg 1
```

3.9. Activació i ús del ratolí

El ratolí és una eina extremadament útil per a treballar en mode consola, ja que permet seleccionar un text i enganxar-lo (així és com s'han capturat totes les entrades i sortides per a redactar el text d'aquest taller, per exemple).

Per tant, el primer que farem és configurar el ratolí, i ho farem amb el programa més comunament utilitzat, el `gpm`, el qual automàticament ja corre en mode *background*. (Això és a causa que `gpm` és un DAEMON, terme que s'analitza amb detall en seccions posteriors.)

Hi ha molts tipus de ratolins, però els més corrents són els de tres botons connectats a l'ordinador via port sèrie o via port PS2. Si tenim un ratolí del primer tipus, executarem la línia següent (es pressuposa que el ratolí està connectat al primer port sèrie de l'ordinador; si no fos així, només s'hauria de canviar el *device* `ttys0` (`/dev/ttS0`) per `ttys1` (`/dev/ttS1`) en cas de ser al segon port, per `ttys2` per al tercer port o per `ttys3` per al quart port):

```
#gpm -m /dev/ttys0 -t msc
```

Si el que tenim és un ratolí connectat al port PS2, per a activar-lo s'usarà el mateix procediment anterior variant únicament el *device* i el tipus de ratolí:

```
#gpm -m /dev/psaux -t imps2
```

En principi, després d'executar una de les dues línies de comandaments anteriors, com a *root*, movent el ratolí hauríem de veure, en tots els terminals, l'indicador de posicionament del ratolí, i en executar:

```
$roger@etseisa7:~$ ps aux | grep gpm
```

hauríem d'obtenir una resposta del tipus:

```
root 182 0.0 0.1 1552 524 ? S Mar11 0:00 /usr/sbin/gpm -m /dev/psaux -t imps2
```

segons *gpm* està corrent en *background*.

Si no fos així, és que el nostre ratolí no és estàndard. En aquest cas haurem de llegir amb atenció el *man* de *gpm* i executar "*gpm -t help*" per tractar d'identificar el tipus de ratolí que s'adapti al que tenim.

Una vegada configurat mitjançant el botó de l'esquerra, podem seleccionar la part de text que ens interessi, i prement el botó del mig enganxarem el contingut seleccionat en la posició actual del cursor. Si s'està acostumat a usar el ratolí per a marcar el posicionament del cursor, pot ser que els resultats que obtinguem no siguin precisament els desitjats, però practicant una mica, de seguida ens acostumarem a aquesta manera d'operar.

Cal destacar que el contingut de la memòria intermèdia del terminal (*buffer*) del ratolí es conserva en passar d'una tty a l'altra.

Activitat

10. Traient profit del fet que el contingut de la memòria intermèdia del terminal (*buffer*) del ratolí es manté entre terminals, obriu en una sessió de *vi* i en una altra situeu-vos en l'arrel del sistema de fitxers, feu una llista dels continguts amb detall, i porteu aquestes dades a l'editor de text. Deseu els continguts del fitxer de text i mitjançant *cat* compareu que efectivament hem obtingut els resultats volguts.

3.10. Altres operacions

Per a finalitzar aquest primer taller, executarem una sèrie de comandaments per a seguir el procés de familiarització amb el sistema i adquirir recursos per a poder solucionar futurs problemes que puguin sorgir.

Ens hem d'informar sobre el sistema, el tipus de màquina sobre el qual corre, el maquinari que tinguem instal·lat, etc. (els retorns dels comandaments que segueixen seran, naturalment, diferents per a cada cas en particular):

```

root@tty1[/]# uname -a
Linux Knoppix 2.4.20-xfs #1 SMP Die Dez 10 20:07:25 CET 2002 i686 AMD Athlon(TM) XP 2100+
                                                AuthenticAMD GNU/Linux

root@tty1[/]# cat /proc/cpuinfo
processor          : 0
vendor_id        : AuthenticAMD
cpu family       : 6
model            : 8
model name       : AMD Athlon(TM) XP 2100+
stepping         : 1
cpu MHz          : 1732.890
cache size       : 256 KB
fdiv_bug         : no
hlt_bug          : no
f00f_bug        : no
coma_bug         : no
fpu              : yes
fpu_exception    : yes
cpuid level      : 1
wp               : yes
flags            : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
                  cmov pat pse36 mmx fxsr sse syscall mmxext 3dnowext 3dnow
bogomips         : 3460.30

root@tty1[/]# lspci
00:00.0 Host bridge: VIA Technologies, Inc. VT8367 [KT266]
00:01.0 PCI bridge: VIA Technologies, Inc. VT8367 [KT266 AGP]
00:05.0 Multimedia audio controller: C-Media Electronics Inc CM8738 (rev 10)
00:06.0 RAID bus controller: Promise Technology, Inc. PDC20276 IDE (rev 01)
00:07.0 FireWire (IEEE 1394): Texas Instruments TSB43AB22 1394a-2000 Controller
00:09.0 USB Controller: VIA Technologies, Inc. USB 6(rev 50)
00:09.1 USB Controller: VIA Technologies, Inc. USB (rev 50)
00:09.2 USB Controller: VIA Technologies, Inc. USB 2.0 (rev 51)
00:0d.0 SCSI storage controller: Artop Electronic Corp AEC6712D SCSI (rev 01)

```

```

00:0f.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8029(AS)
00:11.0 ISA bridge: VIA Technologies, Inc. VT8233A ISA Bridge
00:11.1 IDE interface: VIA Technologies, Inc. Bus Master IDE (rev 06)
00:11.2 USB Controller: VIA Technologies, Inc. USB (rev 23)
00:11.3 USB Controller: VIA Technologies, Inc. USB (rev 23)
01:00.0 VGA compatible controller: ATI Technologies Inc Radeon 7500 QW
root@tty1[/]# df
Filesystem      1K-blocks      Used    Available    Use%    Mounted on
/dev/root        1971           1516         455         77%     /
/dev/CD-ROM     715744        715744         0         100%    /CD-ROM
/dev/cloop     1837536      1837536         0         100%    /KNOPPIX
/dev/shm        407928         40          407888         1%     /ramdisk
/dev/fd0        1424           56          1368         4%     /mnt/auto/floppy

```

Activitat

11. En el directori virtual `/proc/` i en els seus subdirectoris trobarem molta informació relativa al nostre sistema en fitxers de text. Situeu-vos en aquest directori, i mitjançant `more` o `cat` exploreu la informació d'aquests fitxers.

3.11. Conclusió

Aquest ha estat el nostre primer contacte amb un entorn UNIX. Ha servit una mica per a trencar el mite que treballar en aquest tipus de sistemes operatius és complicat i difícil. És més, ha servit perquè en comencem a intuir la potència i versatilitat. Hem après a moure'ns pel seu sistema de fitxers i a fer-hi operacions, a buscar informació, etc. Tot el que hem après fins ara ens serà de gran utilitat d'ara endavant, i ens servirà de base per a ampliar els nostres coneixements.

D'altra banda, la gran potència d'autodetecció i configuració de maquinari de KNOPPIX fa que aquest ens pugui ser molt útil a l'hora d'instal·lar Debian al pròxim taller si sorgissin problemes de maquinari. Sempre podrem tornar a arrancar KNOPPIX i estudiar com ha solucionat aquest el problema (bàsicament estudiant els arxius de configuració).

Com a última activitat del taller, podem tornar a engegar KNOPPIX en mode gràfic, és a dir, sense passar-hi el paràmetre 2 en iniciar-lo (una manera de fer-ho pot ser passant els paràmetres "knoppix lang=es wheelmouse").

4. Instal·lació de GNU/Linux

4.1. Introducció

En aquest capítol veurem els passos essencials que se segueixen en la majoria de processos d'instal·lació de GNU/Linux. Encara que cada distribució tingui el seu propi entorn d'instal·lació, en totes hi ha d'haver uns passos bàsics. En aquest capítol veurem aquests passos i explicarem els conceptes necessaris per a superar-los correctament. És important que abans d'instal·lar un nou sistema operatiu coneguem adequadament els components principals que tenim instal·lats al nostre ordinador per a poder-ho configurar tot adequadament, encara que la distribució que utilitzem incorpori detecció de maquinari.

És possible que en un sol disc dur tinguem instal·lats dos o més sistemes operatius totalment independents. Encara que el procés d'instal·lació d'un altre operatiu al mateix disc (o en un altre d'instal·lat a l'ordinador) no hauria d'interferir en les particions dels altres, és aconsellable fer còpies de seguretat de tots els nostres documents. Tanmateix, si seguim adequadament els passos que detallarem a continuació és pràcticament impossible perdre informació; malgrat això, sempre és recomanable la prudència i fer còpies dels arxius que realment ens importin.

4.2. Arrencant

Generalment, totes les distribucions de GNU/Linux proporcionen algun tipus de mitjà per a l'arrencada del procés d'instal·lació. Actualment se sol proporcionar un CD o DVD d'arrencada o, en cas que el nostre ordinador no tingui el lector corresponent o vulguem fer una instal·lació remota, es proporciona un disquet d'arrencada. Per a començar amb el procés, hem d'introduir el mitjà inclòs al seu dispositiu i engegar l'ordinador, configurant el seu BIOS o EFI perquè arrenqui des de la unitat que ens interessa.

Nota

Abans de començar el procés d'instal·lació és aconsellable conèixer la marca i model de la targeta gràfica i de so que tenim instal·lada, la targeta de xarxa, la marca, el tipus i les característiques del monitor, i qualsevol altre maquinari especial que tinguem. Generalment, per a la placa base, la CPU i la memòria RAM, no sol ser necessari conèixer-ne les característiques. Per a obtenir aquesta informació, podem recórrer als manuals lliurats en comprar l'ordinador o, si tenim un altre sistema operatiu instal·lat, recórrer-hi per a aquesta finalitat (en sistemes WindowsTM ho podem aconseguir a partir del tauler de control).

El primer pas del procés sol ser escollir quin tipus d'instal·lació volem fer. Generalment, l'elecció sol ser per a identificar quin tipus d'usuari està instal·lant el sistema per, segons aquest, proporcionar-li més o menys informació i deixar-lo configurar amb més precisió el sistema o no. Si tenim els coneixements suficients, és recomanable fer la instal·lació en mode expert (o similar) per a poder adaptar més el sistema a les nostres necessitats específiques. En algunes distribucions, aquest primer pas serveix per a seleccionar la versió del nucli que volem instal·lar o per a configurar el procés d'instal·lació en mode gràfic, text, etc. És imprescindible llegir atentament la informació que se'ns proporciona en aquesta primera fase per a poder triar adequadament el que més s'ajusti a la destinació que vulguem donar al sistema.

Seguidament, la majoria de distribucions ens deixen escollir el tipus de teclat (o configuració similar) que volem utilitzar. Si bé n'hi ha de moltes classes diferents, els més freqüents són els *qwerty* (els primers caràcters començant per dalt i l'esquerra del teclat), amb el qual hauríem de seleccionar *qwerty/es* (Spain).

Contingut complementari

Algunes distribucions de GNU/Linux ens permeten fer la instal·lació des de qualsevol mitjà: CD, DVD, FTP, HTTP, disc dur, NFS, etc. També hi ha mètodes d'instal·lació que permeten arrencar el procés des d'altres sistemes operatius.

Contingut complementari

Encara que hi hagi maneres d'instal·lar GNU/Linux utilitzant el sistema de fitxers d'un altre d'operatiu, no és recomanable utilitzar aquest tipus d'instal·lació perquè el rendiment del sistema baixa considerablement.

4.3. Particionant el disc

Particionar el disc dur és una de les parts més crítiques de tot el procés. Aquest pas significa dividir el disc dur en diverses seccions, per la qual cosa cada una es pren com a unitat independent. Si ja tenim un sistema operatiu instal·lat al nostre ordinador, el disc estarà particionat en una o diverses particions, però si el disc és nou, generalment no.

Per a instal·lar GNU/Linux hem de disposar, almenys, d'una partició per a ell. El fet que en modificar la mida d'una partició l'hàgim d'eliminar i crear-la de nou implica perdre tota la informació que hi teníem. Per aquest i altres motius, existeixen programes que ens permeten modificar la mida sense haver-les d'eliminar. El *fips* és un programa amb llicència GPL que ens permet redimensionar les nostres particions formatades amb FAT (per a sistemes WindowsTM de la família no NT) sense perdre'n la informació. També existeixen altres programes comercials que ens permeten efectuar aquest tipus d'operació amb qualsevol altre sistema de fitxers, de manera que si

no volem perdre la informació dels nostres sistemes, n'haurem d'utilitzar algun abans de començar amb tot el procés.

És recomanable que GNU/Linux utilitzi dues particions al disc dur. Una servirà per a desar els fitxers del sistema i l'altra per al *swap*. El *swap* és una zona d'intercanvi entre la memòria RAM de l'ordinador i el disc dur. Serveix quan el sistema operatiu té tota la memòria RAM ocupada i els programes en execució en demanen més. És en aquest moment quan es comença a utilitzar el *swap* per a desar zones de RAM que no s'utilitzen, intercanviant-les perquè les aplicacions no es quedin sense memòria disponible. També és possible prescindir de *swap*, però no és recomanable perquè el sistema no podrà gestionar tan adequadament els seus recursos; a més, en utilitzar simultàniament moltes aplicacions, aquestes es quedaran sense memòria amb més facilitat. Encara que la mida del *swap* pugui ser tan gran com vulguem, es recomana que aquest sigui el doble de la RAM instal·lada a l'ordinador si tenim 64 MB o menys, i igual si en tenim més. Aquests càlculs es basen en proves de rendiment del sistema que ens demostren que arriba un punt en el qual, si les aplicacions necessiten utilitzar massa memòria *swap*, el rendiment decreix molt, cosa que fa que el sistema quedi pràcticament saturat (per a veure quina quantitat de memòria RAM i *swap* s'utilitza, el sistema ens proporciona el comandament *free*).

Existeixen diverses aplicacions per a fragmentar el disc. Una de les primeres que va aparèixer va ser *fdisk*, encara que actualment n'existeixen d'altres, com *cdisk*, *diskDruid*, etc. En alguns processos d'instal·lació es pot escollir quin volem utilitzar, encara que totes permetin fer exactament el mateix canviant, això sí, la presentació, l'entorn, etc. La manera com GNU/Linux identifica els discos és `/dev/hdX` per als discos IDE i `/dev/sdX` per als SCSI, on en ambdós casos la "X" és una lletra, corresponent amb el disc al qual ens vulguem referir de la manera següent:

Dispositiu	Significat
<code>/dev/hda</code>	Mestre del primer canal IDE
<code>/dev/hdb</code>	Esclau del primer canal IDE
<code>/dev/hdc</code>	Mestre del segon canal IDE
<code>/dev/hdd</code>	Esclau del segon canal IDE

Contingut complementari

Si bé amb una o dues particions és suficient per a poder instal·lar GNU/Linux, és interessant dividir el disc en més particions i situar certs directoris en diferents unitats per a poder fer una gestió més eficient dels recursos, evitar caigudes del sistema per saturació de disc, etc. De tota manera, aquests aspectes els deixarem per a cursos més avançats d'administració.

Contingut complementari

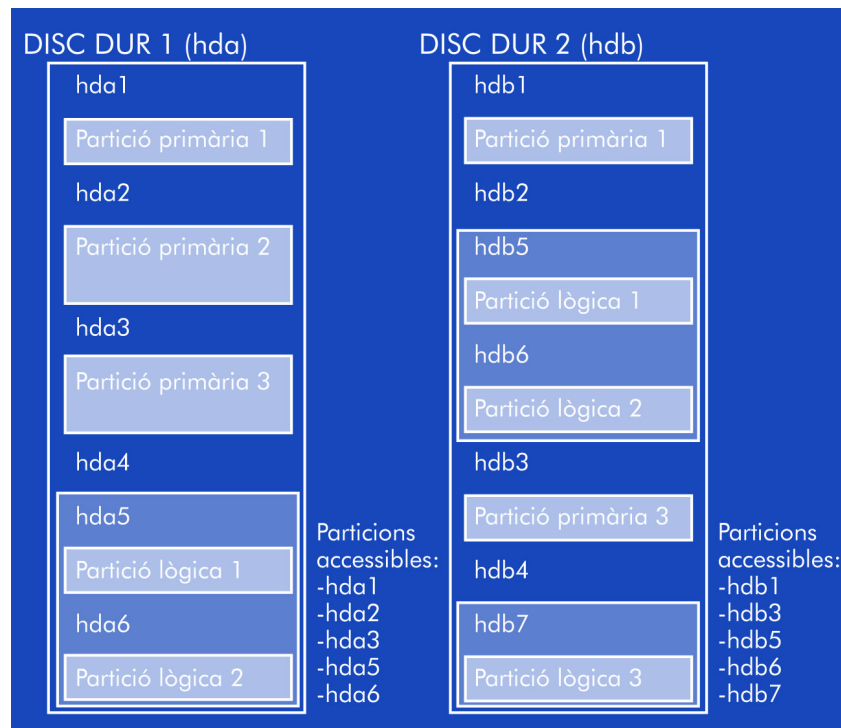
Per als ordinadors personals existeixen dues classes de discos durs: els IDE i els SCSI. Les controladores IDE (*integrated drive electronics*) són les que porten la majoria de plaques base. Amb aquest controlador podem connectar dos dispositius compatibles (discos, CD, etc.) en cada un dels dos canals. Les controladores SCSI (*small/smart computer system interface*) permeten fins a vuit dispositius i tenen taxes de transferència més altes, encara que el seu preu també sigui bastant més elevat que el dels IDE.

Contingut complementari

Per a desar la informació de particions i el programa d'arrencada, els discos tenen una zona de dades reservada anomenada MBR (*master boot record*).

Dispositiu	Significat
/dev/sda	Primer disc de la controladora SCSI
/dev/sdb	Segon disc de la controladora SCSI
...	

Si tenim més d'un disc al nostre ordinador, abans d'entrar al programa de particionament podrem triar sobre quin d'aquests volem operar. Quan creem una partició se'ns preguntarà si ha de ser primària o lògica. En un disc dur podem tenir fins a quatre particions primàries i fins a seixanta-quatre de lògiques. Si no necessitem més de quatre particions, podem escollir qualsevol dels dos tipus. Si en necessitem més, haurem de tenir en compte que les lògiques se situen dins d'una de primària (fins a un màxim de setze per a cada una), de manera que no podem tenir quatre particions primàries creades i després afegir-ne d'altres de lògiques. En aquest cas, n'hauríem de crear tres de primàries i fins a setze de lògiques en la quarta partició primària. En la figura següent podem veure un exemple de manera gràfica:



Quan creem una partició, hem d'indicar quin sistema de fitxers utilitzarem (`Linux ext2`, `Linux ext3` o `Linux swap`). Una vegada fetes les particions, desarem la configuració i hem d'indicar al procés

d'instal·lació on volem situar l'arrel del sistema de fitxers (*root filesystem*) i el swap del sistema. Una vegada efectuats aquests passos, ja podem continuar amb la resta del procés d'instal·lació.

4.4. Instal·lació de mòduls

Els mòduls del nucli són parts de programari especialitzades en alguna tasca concreta. Com que hi ha tota classe de dispositius diferents i desenes de funcions per a múltiples sistemes de fitxers, operacions de gestió en xarxa, etc., es va decidir no incloure de forma estàndard tots aquests programadors de control i les funcions. En les primeres versions de Linux, quan el nucli que teníem no incloïa alguna funció, havíem de recompilar-lo completament i generar-ne un de nou adaptat a les nostres necessitats. Amb la incorporació dels mòduls, tot aquest procés no és necessari i podem seleccionar directament les funcions que necessitem.

Generalment, les distribucions ja incorporen centenars de mòduls diferents per al nucli Linux. Se solen organitzar en diferents categories per a facilitar-ne la localització i, normalment, amb el mateix nom del mòdul o la breu descripció que se'n facilita ja sabem per a quin dispositiu està dissenyat o quina funció fa. Per aquesta raó dèiem anteriorment que era important conèixer el maquinari del nostre ordinador: en aquest pas podem triar amb més precisió els mòduls que necessitem. En alguns d'aquests és obligatori passar algun paràmetre especial (com la direcció d'E/S, la interrupció que utilitza el dispositiu, etc.), informació que podem obtenir per mitjà de la BIOS o EFI del sistema o pels procediments que anteriorment hem comentat.

També és cert que si el procés d'instal·lació porta algun tipus de programa per al reconeixement automàtic de maquinari, tot el procés de selecció i càrrega de mòduls no és necessari perquè ja es fa automàticament. Tanmateix, és possible que no es detecti adequadament algun dels dispositius que tinguem instal·lats; en aquest cas sí que hem d'incloure manualment el mòdul corresponent.

Si en el moment d'instal·lar el sistema ens oblidem d'incloure algun mòdul (o instal·lem algun dispositiu nou) sempre podem recórrer als

Contingut complementari

De totes les particions d'un disc dur en podem triar una perquè sigui l'activa. Aquest *flag* serveix per a indicar a la BIOS o a l'EFI del sistema quina és la partició que ha d'iniciar si en l'MBR no troba cap programa d'arrencada.

Contingut complementari

El fitxer de configuració dels mòduls se sol trobar a `/etc/modules`.

Contingut complementari

Una adreça IP és la identificació d'un ordinador dins d'una xarxa quan utilitzem el protocol TCP/IP, el que més s'utilitza a Internet.

comandaments `insmod` (per a agregar un mòdul nou), `lsmod` (per a fer una llista dels instal·lats), `rmmmod` (per a eliminar-ne algun) i `modprobe` (per a provar-ne algun i, si funciona correctament, incloure'l en el nucli). Tots aquests mòduls no són més que fitxers binaris que acostumem a trobar en el directori `/lib/modules/` del sistema.

Si tenim problemes per a descobrir quin maquinari tenim instal·lat a l'ordinador, una utilitat molt pràctica que se sol incloure en les distribucions és `discover`. Així podrem saber exactament quins components tenim i quins mòduls els corresponen.

4.5. Configuració bàsica de la xarxa

Després de configurar els mòduls que s'inclouran en el nucli del sistema operatiu haurem de configurar la xarxa (si tenim la targeta necessària). Encara que en aquest document no entrarem en detalls sobre xarxes de computadors, donarem les idees necessàries per a poder fer aquest pas sense complicacions.

El primer que es demanarà és el nom que volem donar al sistema. Aquest nom servirà per a referir-nos-hi sense haver de recordar-ne sempre l'adreça IP. Una vegada entrat el nom, se sol demanar si a la nostra xarxa utilitzem un mecanisme anomenat *DHCP* o *BOOTP*, que consisteix a tenir un servidor especial que s'encarrega d'assignar automàticament les IP als ordinadors que engeguen. Si utilitzem aquest mecanisme, ho hem d'indicar i si no, se'ns preguntarà per l'IP i la màscara del nostre ordinador. Si no coneixem aquestes dades, ens hem de dirigir a l'administrador de la nostra xarxa. Seguidament hem d'introduir la IP de la passarel·la o *gateway* de la nostra xarxa. La passarel·la és un dispositiu o ordinador que actua de pont entre la nostra xarxa local i Internet (si no tenim cap dispositiu d'aquest tipus, podem deixar en blanc aquest camp). A continuació, hem d'especificar el servidor o servidors de noms que utilitzem. Un servidor de noms (servidor DNS) és una màquina que ens proporciona l'equivalència entre un nom i una adreça IP. Si no sabem quin usem, també haurem de recórrer a l'administrador de la xarxa.

Si som en una xarxa local, n'haurem de consultar l'administrador perquè ens proporcioni tota la informació necessària sobre això. Si

tenim un altre operatiu instal·lat a l'ordinador, també podem accedir a la seva configuració per a obtenir-la. Tanmateix, en cap cas no ens podem inventar aquests valors perquè el més probable és que no es configuri adequadament i provoquem problemes a la xarxa local.

4.6. Sistema d'arrencada

Una vegada configurats tots aquests aspectes, hem d'instal·lar un petit programa al disc dur perquè en el procés d'arrencada de l'ordinador puguem triar quin sistema operatiu dels que tenim instal·lats volem arrencar. Encara que només hàgim instal·lat GNU/Linux, també haurem d'instal·lar aquest programa, ja que el seu sistema d'arrencada el necessita.

Existeixen diverses aplicacions per a fer aquest procés. Les més usuals són el LILO (*Linux LOader*) i el GRUB (*GNU GRand Unified Bootloader*). L'únic que fan aquestes aplicacions és iniciar el procés de càrrega i execució del nucli del sistema operatiu que li indiquem. Generalment, totes les distribucions detecten si tenim algun altre sistema operatiu instal·lat als discos durs i ens configuren automàticament el sistema d'arrencada. L'únic que hem de tenir en compte és que caldrà situar aquest programa correctament perquè s'executi en arrencar l'ordinador. Normalment se sol posar en l'MBR del disc mestre del primer canal IDE o SCSI, que és el primer lloc que la BIOS o EFI de l'ordinador inspecciona buscant un programa d'aquestes característiques.

4.7. Elecció de paquets

La majoria de processos d'instal·lació inclouen dues maneres de seleccionar els programes que instal·larem en el sistema: bàsic o expert. Amb el procés de selecció bàsic generalment s'agrupen els paquets disponibles per a grans grups de programes: administració, desenvolupament de programari, ofimàtica, matemàtiques, etc. Si encara no coneixem exactament els programes que utilitzarem, aquest tipus d'instal·lació és la més adequada perquè podem

Contingut complementari

Un paquet està format per un o diversos programes/l·libreries relacionats entre si, que s'agrupen per a formar només un bloc. La majoria de distribucions inclouen utilitats per al maneig de paquets (instal·lació, eliminació, configuració, etc.). Aquesta classe d'organització és molt útil perquè en necessitar un programa/utilitat/etc. podem instal·lar-ho tot alhora.

Contingut complementari

RedHat també ha adoptat un sistema de baixada/actualització del mateix estil que l'apt de Debian.

escollir, de manera molt general i sense entrar en detalls, quin tipus de programes volem fer servir.

Quan coneguem una mica més el sistema i sapiguem què és exactament el que utilitzarem, és millor la selecció experta de paquets. Amb aquest tipus de selecció podrem ajustar molt millor quins programes necessitem, estalviant-nos espai al disc i evitant que el sistema carregui més programes del necessari. En instal·lar un sistema per a un ús específic (servidor http, cvs, etc.) és molt recomanable escollir només aquells programes que realment utilitzarem per a evitar problemes de seguretat (com menys portes deixem obertes a l'exterior, més segur serà el sistema).

Algunes distribucions també admeten la possibilitat d'obtenir els paquets des d'ubicacions diferents, com un o diversos CD, des de servidors a Internet, etc. Debian GNU/Linux va ser la primera a tenir un sistema d'aquest tipus, anomenat `apt`. Aquest tipus de gestió és molt útil perquè ens proporciona molta flexibilitat, i ens manté informats de les últimes correccions i actualitzacions dels programes.

4.8. Altres aspectes

Hem de saber què hem de fer si hem tingut algun problema amb la instal·lació o en algun moment el sistema d'arrencada no ens deixa carregar el sistema operatiu. Generalment, en el procés d'instal·lació hi ha algun pas en el qual se'ns pregunta si volem crear un disquet de recuperació. Sempre és molt recomanable generar aquest disquet, ja que ens permet carregar l'operatiu i accedir al sistema de fitxers per a arreglar el que faci falta.

En algunes distribucions, amb el mateix CD d'arrencada podrem fer el mateix. En el cas de Debian, per exemple, amb el mateix procés d'arrencada s'inclou una consola (prement CTRL + F2 hi podem accedir) amb els comandaments bàsics perquè puguem fer operacions essencials de recuperació. De tota manera, si ens trobem amb algun inconvenient greu que no ens permeti arrencar el sistema correctament i volem totes les eines usals, també podem arrencar amb un CD-live de KNOPPIX o qualsevol altra distribució i muntar la unitat on

tinguem instal·lat el sistema per a arreglar la disfunció. El que sí que és important és disposar d'alguna d'aquestes eines i haver-la provat abans que ocorri algun problema seriós per a estar sempre preparat.

La majoria de processos d'instal·lació utilitzen una aplicació denominada `bootstrap`, que també podem instal·lar en el sistema amb el paquet corresponent. Amb aquesta aplicació ens podríem crear el nostre propi procés d'instal·lació o veure com fa realment el sistema aquestes operacions d'instal·lació i configuració. La majoria les podem reproduir amb el programa `base-config` o algun dels altres que el programa mateix crida (més informació al seu manual), encara que sempre podem recórrer als mateixos fitxers de configuració de l'operatiu per a fer-ho manualment.

5. Taller d'instal·lació de Debian Woody

5.1. Introducció

Si el primer taller ens servia per a fer els nostres primers passos sobre un sistema tipus UNIX, i per a això utilitzàvem una distribució que no deixés rastre al nostre PC, ja que s'engegava i s'executava des d'un CD-ROM, aquest ens ha de servir per a aprendre a instal·lar un sistema bàsic GNU/Linux al nostre ordinador. La distribució escollida per al desenvolupament del taller ha estat la Debian 3.0r1 (les versions 3.0 també es coneixen amb el nom de *Debian Woody*).

La decisió d'escollir Debian davant RedHat per al desenvolupament d'aquest taller, i dels dos restants, no ha estat gens fàcil, ja que RedHat ofereix productes comercials que tenen molt bona acollida en el món empresarial, i en general es considera molt més senzill instal·lar una RedHat que una Debian. El fet que s'hagi optat per Debian es deu principalment al següent: en primer lloc, perquè aquesta distribució és la que segueix més fidelment la filosofia GNU/Linux i tot es fa gràcies al treball voluntari de milers de persones; en segon lloc, perquè probablement és una de les distribucions que, tant a l'hora d'instal·lar com de mantenir, deixa més llibertat als usuaris; i com a últim punt fonamental, pel seu sistema de paquets, que probablement és el més consistent que existeix actualment.

S'ha optat, doncs, per Debian; no perquè es consideri millor que RedHat (segur que hi ha tantes opinions sobre això com usuaris d'ambdues distribucions), sinó perquè simplement s'ha cregut que la seva flexibilitat la converteix en una distribució molt apta per a finalitats didàctiques. Però sent conscients del fort arrelament que té RedHat, s'ha dedicat un apèndix a aquesta altra distribució, perquè el lector també la pugui conèixer i es pugui formar la seva pròpia opinió.

Tampoc no s'ha d'oblidar que, a part d'aquestes dues distribucions, n'hi ha moltes més. Però creiem que, quan ja s'ha tingut un primer contacte amb alguna, és tasca de cada usuari provar diferents distri-

Lectura complementària<http://www.debian.org/doc/><http://www.debian.org/distrib/cd><http://www.debian.org/distrib/netinst>

bucions i forjar-se una opinió pròpia de quina és la que s'adapta millor a les seves necessitats o exigències. Per aquest motiu, una vegada més animem a fer tot tipus d'experiments i proves, i que cadascú dirigeixi els seus esforços cap on cregui més interessant. GNU/Linux no és un sistema tancat, sinó al contrari, GNU/Linux és sinònim de llibertat, i per aquest motiu la intenció bàsica d'aquest mòdul és, en general, que una vegada acabat s'hagin establert les bases de coneixement necessàries perquè es pugui exercir sense cap tipus d'impediment aquesta llibertat, amb el benefici de tots els seus avantatges i assumint totes les seves conseqüències.

Activitat

12. Ja que s'instal·larà Debian, és recomanable visitar el seu web i familiaritzar-se una mica amb els seus continguts. Així doncs, es proposa visitar el web <http://www.debian.org> i els seus subapartats. Una vegada més s'insta el lector a deixar-se guiar per la seva curiositat i, en aquesta ocasió, a seguir els enllaços que li semblin interessants.

5.1.1. Sistemes d'instal·lació

Per sistema d'instal·lació s'entén quins recursos o dispositius s'utilitzaran per a fer la instal·lació del sistema. Actualment, gairebé totes les distribucions ofereixen diverses possibilitats per fer-la, però essencialment es pot distingir entre dos sistemes d'instal·lació: mitjançant CD-ROM o per xarxa. A més, és possible, en general, combinar diferents sistemes.

Debian ofereix tres tipus d'instal·lació: mitjançant un joc de CD-ROM, amb un únic CD-ROM, el qual conté el programari d'instal·lació i els paquets bàsics, deixant que la resta de la instal·lació es faci per xarxa, i exclusivament per xarxa. Encara que, òbviament, si s'opta per aquest últim sistema, farà falta algun tipus de suport inicial per a arrencar el sistema d'instal·lació (normalment un joc de disquets, entre dos i disset, depenent del tipus i sistema d'instal·lació escollits, o un CD-ROM, del mateix joc de CD de la distribució, encara que també és possible usar altres mètodes, com el DHCP per a fer una instal·lació completament remota).

El grau d'interacció que requereix una instal·lació també depèn del sistema i del tipus d'instal·lació escollit. Tal com és d'esperar, cada un d'aquests sistemes té els seus avantatges i els seus inconvenients: mentre que una instal·lació estàndard ens permet anar pas a pas, fet que ens serà extremadament útil per a adequar el sistema a les nostres necessitats i possibilitats, un sistema d'instal·lació totalment automàtic requereix unes infraestructures i uns coneixements més avançats, i per tant una inversió, tant en temps com en diners, que queda justificada només si el nombre de sistemes a muntar és molt gran (per exemple, es podria plantejar la implementació d'aquest tipus d'instal·lació en un departament on convinguessin ordinadors destinats a ús personal, amb d'altres destinats a la paral·lelització, i on el seu nombre augmenta sovint).

A l'hora d'escollir un sistema i tipus d'instal·lació, hem de considerar, doncs, diversos factors, com ara: quantes instal·lacions farem?, quantes instal·lacions diferents farem?, quin grau d'experiència tenim?, etc. El fet que sigui aquesta la nostra primera instal·lació ens porta immediatament al tipus d'instal·lació més interactiva i més utilitzada: la instal·lació interactiva estàndard. Ara només falta per determinar el sistema d'instal·lació, i això dependrà fonamentalment de si disposem de connexió a Internet i de la seva velocitat. Òbviament, si no disposem de connexió a Internet o si la velocitat d'accés que tenim és baixa (com la que ofereixen els mòdems estàndards) no tenim més opció que escollir una instal·lació basada en un joc de CD-ROM; si al contrari, disposem d'una velocitat d'accés a Internet mitjanament acceptable (com la que poden oferir les línies basades en tecnologia ADSL) o alta (connexió directa a Internet per passarel·la), la millor opció serà decantar-se per una instal·lació per xarxa.

Una instal·lació efectuada per xarxa representa molts avantatges sobre una d'efectuada mitjançant CD-ROM, i especialment en Debian, ja que això ens permetrà instal·lar les últimes versions disponibles dels paquets; d'altra banda, actualitzar tots els paquets instal·lats en el sistema serà tan simple com executar una sola instrucció. Però aquest fet no ens ha de fer abandonar la instal·lació utilitzant CD-ROM, si, com hem dit, no disposem de connexió a Internet o la que tenim és molt lenta (no s'han de subestimar tampoc els paquets que contenen aquests CD-ROM, ja que Debian Stable es caracteritza per ser una distribució on només s'inclouen paquets que han estat provats exhaustivament).

Lectura complementaria

<http://www.debian.org/distrib/packages>
http://www.debian.org/social_contract#guidelines

5.1.2. Tipus de paquets

A continuació ens centrarem en Debian i en el seu sistema de paquets. Un paquet de Debian és identificable per la seva extensió `.deb`. La distribució Debian diferencia els seus paquets en quatre classes diferents. Els paquets propis de la distribució són en la classe *main*, mentre que les classes *contrib*, *non-free* i *non-US* els proveeix l'organització Debian en benefici dels seus usuaris:

- **main.** Paquets que compleixen les Debian Free Software Guidelines, és a dir, que se'n garanteix l'ús i redistribució lliure, tant de tots els binaris que els componen com del seu codi font complet.
- **contrib.** Paquets que, tot i ser lliures, i per tant a banda dels binaris –també tenen disponible el seu codi font–, depenen d'altres paquets que no ho són.
- **non-free.** Paquets que, fins i tot encara que no costin diners, estan sota condicions oneroses que en restringeixen d'alguna manera l'ús o la redistribució.
- **non-US/main non-US/non-free.** Paquets que no poden ser exportats fora dels Estats Units perquè contenen programari xifrat o programari que pot afectar assumptes relacionats amb patents.

5.1.3. Estat de desenvolupament dels paquets

L'estat de desenvolupament dels paquets marcarà el tipus de distribució que instal·lem. Així doncs, es parla de tres tipus de distribució:

- **stable.** Distribució dels paquets amb la versió oficial més recent de la distribució. Consta de programari estable i ben provat, i canvia només en incorporar correccions importants de seguretat o d'usabilitat.
- **testing.** Distribució que conté els paquets que s'espera que formin part de la pròxima distribució estable. Hi ha una sèrie de requisits molt estrictes que ha de complir cada paquet abans de deixar de ser *unstable* per a passar a ser *testing*.

Lectura complementaria

<http://www.debian.org/releases/testing/>

- **unstable.** En aquesta distribució hi ha els paquets més recents de Debian i en conseqüència els menys provats. Per aquesta raó, poden contenir problemes prou greus per a afectar l'estabilitat del sistema.

5.2. Instal·lació de Debian Woody

Com s'ha dit, en essència Debian presenta tres tipus d'instal·lació: per joc de CD-ROM, per CD-ROM mínim, i instal·lació per xarxa. Si es dominen la primera i l'última, utilitzar la segona és trivial. Així doncs, començarem per instal·lar el nostre primer sistema mitjançant el joc de CD-ROM facilitat pas a pas i, una vegada acabat, analitzarem les diferències entre aquest procés i el d'instal·lació per xarxa.

Cal afegir que les imatges del CD-ROM mínim d'instal·lació no són oficials, si bé està admès com a tal per part de Debian. Aquestes imatges són creades per compte propi, generalment per gent pertanyent a la Debian people, i només estan referenciades al web oficial de Debian, en l'apartat d'instal·lació mitjançant CD-ROM mínim. Tot i així, en general aquest sistema d'instal·lació és molt pràctic, i ha estat provat amb bons resultats, amb imatges de diferents fonts, en diversos ordinadors.

5.2.1. Sabors o *flavours* de Debian Woody

Es coneix amb el nom de *flavours* o sabors els diferents nuclis pre-compilats, destinats a suportar diferents tipus de maquinari de les distribucions. Debian Woody presenta quatre *flavours* diferents:

- **vanilla.** Aquest nucli està basat en la sèrie 2.2, en el qual s'han inclòs molts programes de control destinats a suportar maquinari antic (com poden ser els dispositius ISA), a més de donar suport al port USB.
- **compact.** Aquest nucli està basat en la sèrie 2.2 i principalment orientat a donar suport a dispositius PCI i sistemes IDE i SCSI.

Nota

És possible incorporar paquets a una instal·lació basada en una distribució procedent d'una altra distribució; això en general implica seriosos problemes de compatibilitats i és per aquest motiu que només es recorre a aquesta pràctica en casos puntuals.

Lectura complementària

<http://db.debian.org/>
<http://www.debian.org/CD/netinst/>

- **idepci**. Aquest nucli està basat en la sèrie 2.2 i està concebut per a donar suport al nombre de maquinari més gran possible. Per aquesta raó es tracta d'un nucli bastant gran i poc optimitzat.
- **bf24**. Nucli basat en la sèrie 2.4 amb suport per a ext3 i ReiserFS orientat a donar suport a maquinari nou, com poden ser els teclats USB. També dóna suport a dispositius SCSI i IDE ATA.

En aquest taller es farà la instal·lació utilitzant el *flavour* bf24, ja que l'experiència diu que dóna molt bons resultats, a part d'usar un nucli basat en la sèrie 2.4. Si amb aquest *flavour* es donés el cas que durant el procés d'arrencada l'ordinador es quedés penjat o notéssim comportaments estranys, com podria ser no reconèixer el disc dur, hauríem de canviar de *flavour*, intentant trobar el que s'adaptés millor a les nostres necessitats, o el que simplement ens deixés arrencar sense problemes; en aquest cas la primera opció que s'hauria de provar és la *idepci*.

5.2.2. CD-ROM de Debian Woody i els seus diferents *flavours*

Dels set CD-ROM que componen Debian Woody, cinc porten associats un o més *flavours*, des dels quals, per tant, es pot engegar el sistema d'instal·lació. La relació entre CD i *flavours* és la següent:

- **CD1**. Aquest CD-ROM és *multiboot*, és a dir, en arrencar se li pot passar com a paràmetre el *flavour* que ens interessa entre d'altres. Si no se li passa cap paràmetre, transcorreguts uns quants segons arrencarà com a *idepci*.
- **CD2**. *vanilla*.
- **CD3**. *compact*.
- **CD4**. *idepci*.
- **CD5**. *bf24*.

Així doncs, per a inicialitzar la nostra instal·lació n'hi haurà prou de "bootar" l'ordinador des de la unitat de CD-ROM amb el CD1 i teclajar `bf24` o amb el CD5 i pulsar "Intro".

5.2.3. Installing Debian GNU/Linux 3.0 For Intel x86

El document bàsic que ens proporciona Debian per a la seva instal·lació és *l'Installing Debian GNU/Linux 3.0 For Intel x86*, inclòs en el CD1 (`/install/doc/install.en.html`) disponible en diferents idiomes, entre els quals hi ha el català (`/install/doc/ca/install.ca.html`). És recomanable llegir aquest document i tenir-ne una còpia a mà per si sorgís cap problema, com podria ser no disposar d'una unitat de CD-ROM *bootable*.

5.3. Instal·lació de Debian Woody mitjançant CD-ROM

Aquesta instal·lació es farà en un PC estàndard sobre un disc dur IDE connectat com a màster sobre el port IDE primari estàndard. Si es disposa d'un altre tipus de connectors a la placa base més ràpids (ATA-100 o ATA-133), probablement també seran suportats per Debian, però en general es requereix configuracions especials i, en conseqüència, queden fora d'aquest taller. Pel que es refereix a controladores i discos SCSI, si aquests són mitjanament estàndards, seran detectats sense cap problema durant el procés d'arrencada.

5.3.1. Abans de començar la instal·lació

Abans de començar la instal·lació pròpiament dita, caldrà assegurar-se que disposem d'un espai mínim al nostre disc dur on fer-la (es recomana disposar, com a mínim, d'entre dos i tres gigabytes d'espai lliure). Si aquest és nou, podem començar directament amb la instal·lació, malgrat que també hi pensem instal·lar un altre sistema operatiu (n'hi haurà prou de reservar l'espai que considerem per a aquest amb el tipus de partició que requereixi).

Si disposem d'un espai que prèviament havíem reservat –ja que ja havíem pensat instal·lar GNU/Linux– o tenim una partició de qualsevol altre sistema operatiu on volem instal·lar-lo, també podem prosseguir amb la instal·lació, és a dir, arrencar des del CD-ROM.

Si al contrari, tenim tot el disc dur ocupat i amb una sola partició (cosa molt poc recomanable, ja que en general això fa disminuir sensiblement el rendiment de qualsevol sistema operatiu, i en especial aquells amb sistemes de fitxers poc consistents), hem d'alliberar espai per a poder instal·lar Debian Woody. La dificultat de fer aquesta operació dependrà estrictament de quin sistema de fitxers sigui el que contingui aquesta partició.

Nota

L'aplicació `Qtparted` està integrada en `KNOPPIX`, caldrà, però, arrencar-lo en mode gràfic, ja que l'aplicació està pensada per a aquest entorn.

Probablement, el sistema operatiu en qüestió serà de la família de productes de Microsoft™. Si el sistema de fitxers és de tipus FAT o FAT32 (utilitzats per `MSDOS™`, `Windows95™` i `Windows98™`), el problema és relativament senzill de resoldre, ja que amb la mateixa distribució se'ns facilita una aplicació [`fips20.exe` en el CD1 (`/tools/fips20.zip`)] que ens assistirà a l'hora de repartir el disc dur i crear espai per a instal·lar GNU/Linux (és molt recomanable que abans de fer córrer `fips` es desfragmenti el disc). Si el sistema de fitxers és de tipus NTFS (`WindowsNT™`, `Windows2000™` o `WindowsXP™`), podem fer servir `Qtparted`.

5.3.2. Arrencada del sistema d'instal·lació

Arribats en aquest punt, podem començar la instal·lació pròpiament dita. Per a això, engegarem l'ordinador, ens assegurarem que el primer dispositiu a l'hora de "bootar" sigui la unitat de CD-ROM (entrant en la BIOS), i hi posarem el CD1 (ja que, en aquest cas concret, usarem el *flavour* bñ24. També podríem haver arrencat amb el CD5 amb els mateixos resultats). Al cap d'uns moments ens apareixerà una pantalla de benvinguda com la següent:

```
Welcome to Debian GNU/Linux 3.0!
```

```
This is a Debian CD-ROM. Keep it available once you have installed
your system, as you can boot from it to repair the system on your hard
disk if that ever becomes necessary (press <F3>for details).
```

For a "safe" installation with kernel 2.2.20, you can press <ENTER> to begin. If you want additional features like modern hardware support, specify a different boot flavor at the boot prompt (press <F3> to get an overview). If you run into trouble or if you already have questions, press <F1> for quick installation help.

WARNING: You should completely back up all of your hard disks before proceeding. The installation procedure can completely and irreversibly erase them! If you haven't made backups yet, remove the CD-ROM from the drive and press <RESET> or <Control-Alt-Del> to get back to your old system.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. For copyright information, press <F10>.

Press <F1> for help, or <ENTER> to boot.

boot:

Podem pulsar F1, sense demorar-nos massa temps, ja que si no, la instal·lació començarà automàticament i amb el *flavour* idepci, *flavour* que s'usa per defecte per a accedir a l'índex d'ajuda:

HELP INDEX:

Press the function keys <F1> through <F10> for helpful information.

KEY TOPIC

<F1> This page, the help index.
 <F2> Prerequisites for installing this system.
 <F3> Boot methods for special ways of using this Rescue disk.
 <F4> Special boot arguments, overview.
 <F5> Special boot arguments for special machines.
 <F6> Special boot arguments for selected disk controllers.
 <F7> Special boot arguments for the install system.
 <F8> How to get help.
 <F9> Debian e-mail lists. <F10> Copyrights and warranties.

Press <ENTER> or type boot method, arguments, and <ENTER> to boot. Press any of <F2> through <F10> for quick installation help.

boot:

Podem polsar qualsevol de les opcions que ens ofereix el menú, però la més destacable és l'F3, en la qual trobarem els diferents *flavours* que admet el mode d'arrencada, i també com utilitzar aquest mateix CD1 per a usar-lo com a disc de rescat:

BOOT METHODS

The Rescue disk can be used for system recovery as well as for installation. The rescue method will mount an existing root partition, while the other methods will give you a small, standalone root system.

Available boot methods:

linux or idepci

Start the installation -this is the default.

bf24

Start the installation with a Linux 2.4 kernel.

compact

Start the installation with a kernel including PCI SCSI and IDE drivers.

vanilla

Start the installation with a kernel including many drivers for older hardware (such as systems with ISA-based hardware).

rescue

Boot and mount any root filesystem. The root filesystem must be given at the prompt, so e.g., type rescue root=/dev/hda1. (You can also use rescbf24, resccomp, or rescvanl.)

Press <ENTER> or type boot method, arguments, and <ENTER> to boot.

Press function key <F1> for the help index.

boot:

Teclegem, doncs, bf24, i comencem amb la instal·lació.

Una vegada fet això, es carregarà el nucli (veurem durant uns instants les sortides per pantalla de les diferents proves que es fan) i immediatament ens apareixerà una pantalla, *Choose The Language*, perquè mitjançant els cursors seleccionem l'idioma d'instal·lació.

A partir d'ara ja disposem, polsant la combinació de tecles ALT + F2 i INTRO, d'un intèrpret d'ordres, que, encara que molt bàsic, es pot usar en qualsevol moment de la instal·lació.

En la tty3 (ALT + F3) el nucli deixa els seus missatges (és a dir, el contingut del fitxer /var/log/messages).

5.3.3. Configuració de l'idioma d'instal·lació

És molt recomanable seleccionar l'opció a `Choose this and press Enter to proceed in English` per a evitar possibles errors de traducció. En general, i si és possible, sempre és millor treballar amb l'idioma original, i així es farà durant la resta del taller; si bé, una vegada més, es deixa llibertat al lector perquè decideixi ell mateix el que li sembli més adequat; en aquest cas, que sigui ell qui triï l'idioma que utilitzarà durant la instal·lació.

Escollim l'idioma anglès i polsem "Intro". A la pantalla *Choose Language Variant* següent es pot escollir quin anglès es vol utilitzar: nosaltres usarem l'anglès `English (United States)`.

5.3.4. Menú principal d'instal·lació

Després de la pantalla de crèdits *Release Notes*, i després de polsar "Intro", entrarem en la interfície *Debian GNU/Linux Installation Main Menu*, que ens servirà per a configurar la resta de la instal·lació. Aquesta interfície opera segons uns criteris preestablerts; així, sempre se'ns suggerirà el pròxim pas a donar per a una instal·lació estàndard, seguit, si cal, d'alternatives a aquesta opció. No obstant això, mitjançant els cursors ens podem desplaçar i seleccionar qualsevol opció de les que se'ns ofereixen després d'una línia en blanc (aquí apareix una llista de totes les opcions possibles d'instal·lació, encara que algunes no estaran disponibles si en primer lloc no n'hem utilitzat d'altres; en principi, llevat de casos excepcionals i perquè és la nostra primera instal·lació, mai no usarem aquestes opcions i deixarem que la mateixa interfície d'instal·lació ens hi guïi mitjançant el suggeriment que ens faci en cada moment).

5.3.5. Configuració del teclat

Com a primer suggeriment, la interfície d'instal·lació ens proposa que configurem el teclat `Configure the Keyboard`. Així ho fem. Polsem "Intro" per a accedir a la pantalla *Select a Keyboard*, i seleccionem el teclat, `qwerty/es:Spanish`; polsem `INTRO` perquè el dispositiu d'entrada respongui al mapat de caràcters del teclat espanyol. En la `tty2` podem comprovar que, efectivament, ja disposem d'accents, per exemple.

Nota

L'idioma d'instal·lació no condiciona de cap manera la configuració del teclat, que fins al moment és `us`, ni l'idioma d'interacció amb el sistema ni altres configuracions. Aquesta opció només és vàlida durant el procés d'instal·lació.

5.3.6. Partició del disc dur

Una vegada seleccionat el teclat, el procés d'instal·lació ens torna al menú principal i ens proposa que procedim a particionar el disc dur, `Partition a Hard Disk`. Així ho fem. Polsem `INTRO` i entrem a la pantalla de selecció de dispositius que cal particionar, `Select Disk Drive`. Si disposem de més d'un disc dur, mitjançant els cursors podem seleccionar sobre quin d'aquests volem operar. Una vegada feta la selecció, polsem `INTRO`. Després d'una pantalla d'avertència referent a les limitacions de `LILLO`, *LILLO Limitations*, i polsant `INTRO` entrem en una altra pantalla de consell sobre com s'ha de calcular la mida de les particions si es té intenció d'utilitzar el Reiser filesystem, *Note on additional space for Reiserfs Journal*. Polsem `INTRO` novament i s'arrenca l'aplicació `cfdisk` per a crear i gestionar les particions.

Les mides, les seves característiques i el nombre de particions depenen en gran manera del tipus d'ús i de la quantitat de disc dur de què es disposi. Com que es tracta d'una instal·lació amb finalitats educatives, es facilita seguidament la informació sobre les particions que es crearan suposant que es treballi sobre un espai d'entre els cinc i els quinze gigabytes destinats a la nova instal·lació.

Com a mínim cal crear dues particions: una primera per a muntar el sistema i una altra de `swap` (és cert que es pot fer instal·lacions sense `swap`, però, tal com ja s'ha dit, no és en absolut recomanable). Per a augmentar l'eficiència del sistema, nosaltres crearem cinc particions.

- La primera destinada a allotjar la *root partition*; aquesta no ha de ser gaire gran, per això se li destinarà entorn d'un deu per cent del disc dur, preferentment en una partició primària; però si no en disposem, la podem crear com a partició lògica sense donar-hi més importància.
- La segona es destinarà a `swap`. Es recomana que aquesta tingui una mida d'entre 512 Mb i 1 Gb; amb aquestes dimensions ens assegurarem, excepte en casos excepcionals, que mai no s'arribarà a saturar; aquesta partició també és preferible que sigui primària, però si ha de ser lògica, tampoc no repercutirà en el rendiment del sistema.
- La tercera partició estarà destinada a allotjar el directori `/home/`, la finalitat del qual és emmagatzemar dades d'usuari,

i, depenent de la mida del disc dur, se li pot assignar entre un deu i un vint per cent del disc dur, segons el nombre d'usuaris i de l'ús que es farà del sistema.

- La quarta partició serà per al directori `/usr/`; cal tenir present que aquesta partició allotjarà gran part del programari que s'instal·li, per la qual cosa haurà de tenir una mida significativa, aproximadament un quaranta per cent del disc.
- L'espai restant es destinarà al directori `/var/`, on s'allotgen llibreries, fitxers de `log`, etc.

La distribució de particions anterior és només una proposta que té dos objectius: d'una banda, pretén millorar el rendiment que ofereix una instal·lació basada únicament en dues particions i, de l'altra, dona més robustesa al sistema. Entre altres avantatges, tenir les dades repartides entre diferents particions fa que la corrupció d'una d'aquestes no impliqui automàticament la pèrdua de tota la informació del sistema. Òbviament, es poden crear altres particions o ometre'n algunes de les proposades. (L'ordre de les particions no afecta el comportament del sistema.)

Pel que es refereix a `cdisk`, el seu funcionament es basa en l'ús dels cursors, tant per a moure's per les particions del disc (part superior de la pantalla, mitjançant els cursors `Up` i `Down`), com per a seleccionar les possibles operacions que es puguin fer en cada moment (part inferior de la pantalla, amb `Left` i `Right`), ja que aquestes variaran segons l'estat de la partició seleccionada. Per a crear una nova partició, cal seleccionar l'opció `[New]`; a continuació, si encara es poden crear particions primàries (el nombre màxim són quatre), se'ns preguntarà si la volem crear com a partició primària o lògica; després hem d'especificar, en Mb, la mida de la partició; i finalment la seva ubicació física al disc (és recomanable que abans de començar a particionar el disc fem un petit esquema de com ho volem fer i crear les particions a partir d'aquest, per a poder respondre així sempre `[Beginning]` aquesta pregunta).

Una vegada s'han creat les particions, els cal assignar el tipus de sistema de fitxers. Per defecte les particions es creen de tipus Linux, és a dir, amb codi `0x83`; en general, doncs, aquesta opció per defecte ja ens serà útil excepte per a la partició destinada a `swap`. Per a aquesta partició en concret caldrà seleccionar l'opció `[Type]`,

amb la qual cosa primer se'ns mostrarà, en hexadecimal, la relació entre codis i *filesystem types*, i després se'ns preguntarà quin tipus de sistema de fitxers volem assignar a la partició seleccionada. Per defecte se'ns proposa la 0 x 82, és a dir, *Linux swap*, que és l'opció que volem. Si tenim més d'una instal·lació de GNU/Linux al mateix ordinador, es pot utilitzar la mateixa partició swap per a totes, ja que la informació que s'hi pugui emmagatzemar durant el funcionament del sistema és totalment volàtil. N'hi haurà prou, doncs, de polsar "Intro" i se'ns retornarà a la pantalla principal de `cfdisk` mostrant-nos que, efectivament, s'ha fet el canvi.

Una vegada tinguem construïda la taula de particions a la nostra mida, només caldrà seleccionar l'opció [Write] perquè aquesta, després d'una pregunta de confirmació final a què respondrem *yes*, s'escrigui en l'MBR i es puguin fer efectius els canvis. Ara, doncs, ja podem abandonar `cfdisk` amb l'opció [Quit] i prosseguir la instal·lació.

5.3.7. Inicialització i activació de la partició swap

Una vegada hem sortit del `cfdisk`, retornem al menú principal. Aquesta vegada se'ns proposa que inicialitzem i activem la partició de swap, *Initialize an Activate a Swap Partition*. Polsem INTRO per a entrar en una pantalla, *Scan fo Bad Blocks?*, on se'ns pregunta si volem revisar la partició a la recerca de possibles blocs defectuosos. Aquesta mateixa pantalla ens apareix cada vegada que inicialitzem i activem una partició. Cal advertir que el procés de revisió és lent; així doncs, aquesta vegada l'activarem per veure la seva manera de procedir, però per a la resta de particions deixarem que cada u prengui la decisió que estimi més encertada. L'experiència posa de manifest que per a discos nous, o en bon estat, no és necessari executar aquest procés de prova, ja que els resultats sempre solen ser positius.

Una vegada acabat el procés de revisió, entrem en una pantalla de confirmació final, *Are You Sure?*, abans de procedir a la inicialització i a l'activació de la partició. Polsem INTRO per a portar a terme el nostre objectiu.

5.3.8. Inicialització i activació d'una partició Linux

De tornada al menú principal, aquest ens suggereix que inicialitzem i activem una partició de Linux, *Initialize and Active a*

Swap Partition. Aquesta operació és indispensable, ja que és en aquest punt, quan inicialitzarem, activarem i muntarem la partició *root (/)*. Polsem **INTRO** per entrar a la pantalla de selecció del tipus de sistema de fitxers que volem per a aquesta partició, *Choose Filesystem Type*. Nosaltres decidim que sigui de tipus *ext3*. Una vegada feta aquesta selecció, entrem a la pantalla, on triarem la partició sobre la qual volem operar, *Select Partition*. Com que es tracta de muntar la partició *root*, escollirem la primera. Una vegada fet així, entrarem a la pantalla ja coneguda de revisió de blocs defectuosos, i a continuació a la pantalla de confirmació d'operació. Després de l'operació, el sistema d'instal·lació ens pregunta si volem muntar en aquesta partició el *root filesystem*, *Mount as the Root Filesystem?* Òbviament, ja que així ho havíem planejat, respondrem afirmativament.

5.3.9. Inicialització i activació d'altres particions Linux

Ara el menú principal ens suggereix, com a opció principal, que instal·lem el nucli i els seus mòduls en la *root partition*, *Install Kernel and Driver Modules*. Nosaltres encara no ho farem, ja que, tal com havíem planejat, hem d'inicialitzar, activar i muntar la resta de particions que hem creat. Triem, doncs, la primera alternativa que ens ofereix el menú principal *Initialize a Linux Partition*. Fet això, entrarem a la pantalla de selecció de *filesystem*, *ext3* novament, i entrarem a la pantalla de selecció de partició. Optem per la primera que se'ns suggereix, i passem a les pantalles de revisió de blocs i de confirmació. Una vegada acabada l'operació, se'ns pregunta quina partició volem muntar *Select Mount Partition*. Aquí la primera vegada optarem per la de */home*, seguint el plantejament inicial.

De nou en el menú principal, repetirem les accions anteriors per a muntar les particions de */usr* i de */var*.

5.3.10. Instal·lació del nucli

Una vegada acabat el procés d'inicialització, activació i muntatge de totes les particions que havíem creat, podem procedir al pròxim pas

que el menú d'instal·lació ja fa estona que ens suggereix, és a dir, instal·lar el nucli i els mòduls `Install kernel and Driver Modules`. Entrem ara a la pantalla de selecció del mitjà d'instal·lació *Select Installation Medium*. Com que instal·lem el sistema utilitzant el joc de CD, triem la primera opció. Se'ns demana ara el primer CD, *Please insert the CD-ROM*, però com que ja és al lector, podem continuar el procés.

Entrem ara en una pantalla en la qual se'ns demana on ha d'anar a buscar el sistema els fitxers del nucli per a la instal·lació, però com que fem la instal·lació des de CD-ROM, només se'ns dóna una opció. Així doncs, polsem `INTRO` per a instal·lar el nucli al disc dur, en la *root partition*.

5.3.11. Configuració de mòduls

Un cop aquí, el menú principal ens suggereix la configuració dels mòduls del nucli `Configure Device Driver Modules`. Seguint aquest pas, entrem en una pantalla d'advertència sobre els mòduls que ja porta incorporat el nucli que s'ha instal·lat *Note about loaded drivers*, el qual ens indica que si no trobem cap mòdul que sigui del tot necessari podem prosseguir amb la instal·lació. En principi no ens hem de preocupar massa pels mòduls, ja que per ara el nostre objectiu principal és instal·lar un sistema mínim al nostre PC, i després ja l'anirem ampliant i millorant. Així doncs, podem fer un tomb pel menú de mòduls a fi d'adquirir coneixements, però, en principi, no n'activarem cap (una vegada instal·lat el sistema bàsic, podrem entrar en aquest mateix menú i activar qualsevol mòdul mitjançant el comandament `modconf`).

5.3.12. Configuració del hostname

És el moment de donar un nom al sistema que instal·lem `Configure the hostname`. Per defecte se'ns suggereix el nom de *Debian, Choose the Hostname*. Podem esborrar aquest suggeriment i donar un nom d'una sola paraula que ens agradi més. (En aquest punt només cal especificar el nom local del sistema. Si aquest formarà part d'una xarxa, i per tant necessitarà domini, ja s'especificarà en l'apartat de configuració de la xarxa.)

5.3.13. Instal·lació del sistema base

Amb tota aquesta informació, el sistema ja pot fer la instal·lació del sistema base; és el que ens proposa el menú principal, `Install the System Base`. Acceptem, doncs, aquesta proposta. Novament se'ns pregunta pel mitjà d'instal·lació, CD-ROM, l'opció per defecte i seguidament se'ns demana de nou el CD1 per a prosseguir; polsem, doncs, `INTRO` i deixem que el sistema llegeixi els fitxers necessaris.

El pas següent és especificar el directori des d'on s'instal·larà el sistema base `Select Archive Path`. Només se'ns mostra una opció, `instmnt`, que és el directori virtual que s'ha creat per a aquesta finalitat (en realitat aquesta informació s'obté del CD1, a partir d'enllaços simbòlics d'aquest directori al directori `/CD-ROM/`). Així doncs, polsem `INTRO` i deixem que finalment comenci la instal·lació del sistema base al disc dur.

5.3.14. Creació d'un disc d'arrencada

Una vegada acabat el procés d'instal·lació del sistema base, el menú principal ens suggereix fer el sistema `bootable`, `Make System Bootable`. Tanmateix, com a mesura de seguretat ens decidim per la primera alternativa que se'ns ofereix, és a dir, crear un disc d'arrencada. Per tant, seleccionem aquesta opció, `Make a Boot Floppy`, i polsem `INTRO`. Inserim un disquet verge a la unitat de disquets i procedim a crear aquest disc.

5.3.15. Instal·lació de LILO

Una vegada creat el disc d'arrencada, el menú principal ens suggereix que tornem a arrencar el sistema. Però també podem fer el sistema `bootable`, com se'ns suggeria anteriorment, és a dir, deixar que escriu en l'MBR la informació necessària perquè es pugui arrencar la instal·lació que acabem de fer des del mateix disc dur. Per a això, hem de seleccionar l'opció `Make System Bootable` i contestar afirmativament, en general, les propostes que ens fa el sistema d'instal·lació.

Fer aquesta operació en aquest moment està plenament justificat si al nostre PC només hem instal·lat GNU/Linux, o si creiem que el ges-

tor d'arrencada LILO serà capaç d'arrencar altres possibles sistemes operatius que tinguem instal·lats (si es tracta de sistemes MicrosoftTM, es pot instal·lar LILO, ja que l'experiència mostra que LILO és perfectament capaç de gestionar-ne l'arrencada). Però no hem de dur a terme aquesta operació, si ja disposem d'un altre gestor d'arrencada (el qual caldrà configurar perquè pugui engegar el sistema que acabem d'instal·lar), o si volem instal·lar un altre gestor d'arrencada com pot ser GRUB, o si no estem plenament segurs que LILO serà capaç de gestionar correctament l'arrencada de la resta de sistemes operatius que tinguem instal·lats.

Nota

Trobareu més informació sobre GRUB i LILO en taller següent.

Si no hem instal·lat LILO, de moment podem engegar el sistema operatiu mitjançant el disquet que hem creat anteriorment, i al pròxim taller avaluarem la possibilitat d'instal·lar el gestor d'arrencada des del mateix sistema.

5.3.16. Reinicialització del sistema

Ha arribat el moment de reinicialitzar el sistema per a engegar el sistema base que hem instal·lat al nostre disc dur, i que a partir d'ell comenci a personalitzar la nostra primera instal·lació de GNU/Linux. Així doncs, retirem el CD1, o quan es produeixi la reinicialització, entrem en la BIOS per reestablir la seqüència de dispositius d'arrencada, preferiblement unitat de disquets, disc dur, CD-ROM, etc. per a un sistema estàndard, i el disquet si hem instal·lat LILO, i seleccionem l'opció `Reboot the System`. Després d'una pantalla final d'avertències i recomanacions que hem de llegir atentament, podem procedir a la reinicialització. Si no s'ha instal·lat LILO, el disquet haurà de romandre a la unitat de disquets, i ens hem d'assegurar que el PC s'engegarà des d'ell.

5.3.17. Arrencada del sistema base

Si tot ha anat com cal, després de reiniciar el sistema observarem, tant si hem arrencat íntegrament des del disc dur com si s'ha fet inicialment mitjançant disquet, que el sistema, en arrencar, ja usa informació del disc dur. Transcorreguts uns moments, en els quals apareixeran per pantalla els resultats dels diferents processos que s'executen durant l'arrencada, el sistema entrarà en una pantalla de benvinguda, ens

convidarà a prosseguir la instal·lació i ens recordarà que podrem repetir aquest procés en qualsevol moment executant el comandament `base-config` com a *root*. Polsem `INTRO` i prosseguim.

5.3.18. Configuració horària

Immediatament després de la pantalla inicial de configuració, entrem en la de la configuració horària del sistema, *Time Zone Configuration*. En aquesta se'ns mostrarà l'hora de la BIOS. Si aquesta hora coincideix amb l'hora del nostre fus horari, hem de contestar que no; al contrari, si aquesta hora està establerta respecte al meridià de Greenwich, hem de contestar que sí, perquè el sistema faci els càlculs necessaris per a situar-nos en l'hora del nostre fus horari.

5.3.19. Configuració geogràfica

Una vegada establert el criteri per al càlcul horari, hem d'especificar, a les dues pantalles següents, *Time Zone Configuration*, el nostre emplaçament geogràfic. Si l'hora de la BIOS està sintonitzada segons l'hora local, i així ho hem especificat en l'apartat anterior, les respostes a aquestes preguntes no tindran cap rellevància. Sí que és important contestar correctament aquestes preguntes si l'horari de la BIOS s'ha establert segons el criteri GTM. (Sempre és possible reconfigurar aquests paràmetres mitjançant el comandament `tzconfig`.)

5.3.20. Establiment de la política de contrasenyes

Una vegada acabats els procediments per a la configuració horària del sistema, ha arribat el torn d'establir la política de contrasenyes, *Password Setup*. Aquesta es compon de la possible habilitació dels sistemes *md5* i *shadow*, i de l'establiment de la contrasenya de *root*, i de la creació d'un compte d'usuari.

Sistema md5

En primer lloc el sistema ens pregunta si volem instal·lar el sistema de contrasenyes *md5*. Si no tenim sospita de problemes d'incompa-

tibilitats (NIS, etc.), podem contestar que sí, ja que és bona idea poder usar contrasenyes amb més de vuit caràcters.

Sistema shadow

Una vegada més, si no tenim sospita de possibles incompatibilitats (aquest sistema no sol implicar problemes d'incompatibilitats), és bo habilitar aquesta opció, ja que així ens assegurem que el fitxer que conté les contrasenyes xifrades (`/etc/shadow`) només serà accessible pel root. Això representa una mesura més de protecció per al nostre sistema.

Contrasenya de root

Ara hem d'entrar la contrasenya de root. És important seguir les recomanacions per a seleccionar aquesta contrasenya i recordar-les. Amb l'objectiu de confirmar que s'ha entrat la contrasenya que volem i no hi ha hagut possibles errors de tecleig, se'ns demana que la tornem a escriure a tall de confirmació.

Creació d'un compte d'usuari

Tal com ja s'ha remarcat, treballar sempre com a root és una mala política per diferents motius. A conseqüència d'això, el sistema ens recomana crear un compte d'usuari normal. Se'ns demanarà el nom d'aquest compte, si hi volem afegir dades suplementàries (per defecte, Debian user), i la seva contrasenya, que, per la mateixa raó, l'hem d'escriure dues vegades.

5.3.21. Últimes configuracions

Paquets PCMCIA

Una vegada creat l'usuari normal, si no disposem de dispositius PCMCIA (típics en ordinadors portàtils), el sistema ho detectarà automàticament i ens suggerirà que confirmem que els paquets associats a aquests dispositius poden ser eliminats.

Nota

Davant l'oblit de la contrasenya de root, hi ha diferents estratègies a seguir. Una pot ser utilitzar el CD-ROM d'instal·lació, inicialitzar una instal·lació i, just després d'haver configurat el teclat, accedir a la tty2; una vegada allà, podem muntar la *root partition* del nostre disc dur i mitjançant un editor eliminar el contingut del camp de contrasenya xifrada del fitxer `/etc/passwd`. Quan tornem a reiniciar el sistema, el superusuari no tindrà contrasenya; per aquesta raó, el primer que hem de fer és assignar-li'n immediatament una.

Configuració d'un sistema PPP

Si disposem d'un mòdem, pot ser bona idea que procedim a configurar-lo, encara que també es pot fer una vegada estiguem en el sistema mitjançant l'aplicació `pppconfig` executada com a `root`. Òbviament, si no disposem d'aquest dispositiu, respondrem negativament aquesta pregunta. Si no necessitem els serveis de PPP, podem desinstal·lar els seus paquets: `ppp`, `pppconfig`, `pppoe` i `pppoeconf`.

5.3.22. Configuració d'apt

En aquesta secció se'ns pregunta sobre les fonts on `apt` haurà d'anar a buscar la informació per construir la seva base de dades sobre els paquets (dependències, estat d'instal·lació, etc.). Com que fem una instal·lació íntegrament basada en CD-ROM, escollim la primera opció, i, després de posar el CD1 al lector, acceptem la ruta que ens suggereixi per accedir-hi. Després de transcórrer uns moments en els quals `apt` extraurà la informació relativa als paquets que continguin aquest CD, se'ns preguntarà si volem afegir el contingut d'un altre CD a la base de dades. Una vegada hàgim introduït el CD2 al lector, contestarem `yes` aquesta pregunta. Repetirem aquesta operació fins que `apt` hagi processat el contingut del setè CD-ROM. Les operacions sobre la gestió de continguts dels CD-ROM es poden fer també des del sistema mitjançant l'aplicació `aptCD-ROM`.

Una vegada acabat l'escaneig dels continguts dels set CD-ROM, se'ns preguntarà si volem afegir alguna altra font d'on `apt` pugui obtenir paquets. De moment contestarem que no, igual que a la pregunta següent respecte de la revisió de seguretat dels continguts dels paquets (no és necessari de moment, ja que, en usar únicament paquets de la distribució oficial, en principi els seus continguts ja són segurs).

5.3.23. tasksel i dselect

A continuació se'ns pregunta si volem que s'executin els programes de selecció de paquets, primer `tasksel` i després `dselect`. Respondrem que no, ja que aquests programes també els podem exe-

cutar des de la línia de comandaments una vegada hàgim acabat la instal·lació.

Aquí acaba la instal·lació bàsica; com a últim punt, i ja responent preguntes formulades per `apt`, se'ns informarà dels paquets que s'instal·laran i els que es desinstal·laran. Una vegada acabat aquest procés, se'ns preguntarà si volem esborrar els paquets `.deb` del nostre disc dur. (Aquests arxius s'emmagatzemen a `/var/cache/`.) Com que en disposarem en CD-ROM, podrem contestar que sí, ja que la seva presència només implica l'ocupació d'espai al disc dur.

Automàticament, s'engega el programa de configuració d'`exim`. Com que la configuració d'aquest programa de gestió de correu no és objecte d'aquest curs, escollim l'opció 5 per a sortir del programa. Finalment, després d'uns moments ens apareixerà per pantalla:

```
Debian GNU/Linux 3.0 nom_de_la_màquina tty1
nom_de_la_màquina login:
```

Així, tenim instal·lat un sistema bàsic estàndard al nostre ordinador. Ara comença la tasca de construir un sistema a la nostra mida. És molt important, arribats en aquest punt, distingir entre dos tipus de sistemes: els de desenvolupament i els de producció.

Un sistema de desenvolupament és aquell que està destinat a l'experimentació, on es fan proves i experiments, i on no prevalen ni l'estabilitat ni l'eficiència; és, en definitiva, un sistema destinat a l'adquisició de coneixements sobre el sistema mateix. Contràriament, en un sistema destinat a la producció, l'eficiència i l'estabilitat són les característiques que més pes tenen. Així doncs, ens hem d'assegurar que contingui únicament i exclusivament aquells paquets estrictament necessaris, ja que el fet de tenir paquets innecessaris instal·lats va en detriment de l'eficiència. L'estratègia que caldrà seguir abans de muntar un sistema de producció (un servidor de web, un servidor d'aplicacions, etc.) passa sempre per treballar primer sobre un sistema de desenvolupament, on s'assajaran i provaran diferents tàctiques per a poder treure conclusions amb vista a muntar el sistema de producció.

Nota

En el camp de la informàtica, el terme *desenvolupament* té més d'una acceptió; sovint s'utilitza en el sentit de 'creació o evolució d'una aplicació'.

El nostre primer sistema serà, evidentment, un sistema de desenvolupament, tant perquè no l'hem pensat perquè serveixi per a cobrir cap necessitat en concret, com perquè és la nostra primera experiència de muntatge d'un sistema, la finalitat del qual és, únicament, obtenir coneixements. Instal·larem i desinstal·larem diferents paquets, farem proves amb diferents configuracions, etc. I això, és clar, perjudica directament l'eficiència i l'estabilitat del sistema. Així doncs, animem el lector que, una vegada finalitzat el mòdul i hagi obtingut ja una idea global del que és instal·lar un sistema, amb tot el que això implica, reinstal·li de nou tot el sistema partint de zero per adaptar-lo estrictament a les seves necessitats.

5.4. Instal·lació de Debian Woody per xarxa

Molts dels passos per a fer una instal·lació per xarxa són comuns als d'una instal·lació mitjançant CD-ROM; així doncs, destacarem aquells aspectes que difereixen del procés descrit anteriorment i ens hi centrarem.

5.4.1. Particularitats d'una instal·lació per xarxa

La diferència bàsica entre una instal·lació feta mitjançant un joc de CD-ROM i una per xarxa resideix en la ubicació d'on s'extreuen els paquets a l'hora d'instal·lar-los. Mentre que en una instal·lació feta a partir de CD-ROM en caldrà inserir-ne al lector almenys un cada vegada que s'instal·li un paquet nou per a poder extreure'n les dades necessàries, en una instal·lació per xarxa els paquets s'obtenen remotament, fet que ens permet, en primer lloc, accedir a l'última versió d'aquests, i poder actualitzar tots els que tinguem instal·lats amb una simple línia de comandaments (`apt-get upgrade`).

D'altra banda, per a fer una instal·lació per xarxa, generalment n'hi ha prou amb un joc de disquets, o amb un únic CD-ROM, que contingui la informació necessària per a poder engegar un sistema operatiu bàsic sobre el qual es farà córrer el programa d'instal·lació (que ja està inclòs en el joc de disquets o en el CD-ROM, junta-

ment amb els mòduls que ens puguin ser necessaris per a configurar l'accés a la xarxa), i a partir d'aquí, la resta d'informació s'obtindrà remotament.

5.4.2. Aspectes comuns dels diferents mètodes d'instal·lació

Com ja s'ha dit, molts dels passos que cal seguir per a fer una instal·lació per xarxa són comuns a les dades per a fer una instal·lació mitjançant el joc de CD-ROM de la distribució. Així doncs, entrem al programa d'instal·lació de la mateixa manera com ho hem fet anteriorment (5.3.1 i 5.3.2) i inicialment seguim els mateixos passos per configurar l'idioma d'instal·lació, el teclat, particionar el disc dur i activar les particions (5.3.3, 5.3.5, 5.3.6, 5.3.7, 5.3.8 i 5.3.9), i per instal·lar el nucli (5.3.10). En l'apartat `Select Installation Medium` optem per l'opció `/dev/fd0`, i immediatament se'ns demanarà que inserim el `rescue disk` seguit dels discos de programadors de control. Les imatges d'aquests discos són a `/CD-ROM/dists/woody/main/disks-i386/3.0.23-2002-05-21/images-1.44/bf24`. Un cop aquí, haurem de comprovar si el nucli ha reconegut la nostra targeta de xarxa i, si no, procedirem a instal·lar i configurar el mòdul pertinent per a fer-la operativa. En molts casos la configuració, pas de paràmetres, es pot fer de manera automàtica mitjançant `modprobe`, programa que es pot llançar des de la mateixa interfície d'instal·lació després de la selecció d'un mòdul. Una vegada fet això, hem de configurar la xarxa, especificar de quin lloc obtindrem les fonts, i a partir d'aquest punt seguirem els mateixos passos que amb el CD-ROM per acabar la seva instal·lació.

5.4.3. Instal·lació del mòdul de xarxa

Aquest és un punt clau per a poder fer la instal·lació per xarxa, ja que és aquí on, si el programa de control de la nostra targeta de xarxa no ha estat compilat dins del nucli, hem de seleccionar el mòdul necessari per a tenir-hi accés. En primer lloc, doncs, hem d'esbrinar si la nostra targeta de xarxa ja ha estat detectada durant el procés d'arrencada i s'ha carregat el seu programa de control corresponent. Per a fer-ho, accedim al segon terminal (`ALT + F2`) i executem el comandament `amesg`. Ara hem de buscar, entre les moltes línies que ens ha tornat aquest comandament, si n'hi ha algunes que facin referèn-

cia a la nostra targeta de xarxa. A tall d'exemple, per a una targeta RTL-8029 (Realtek Semiconductors) s'obté:

```
~# dmesg
.
.
.
ne2k-pci.c:v1.02 10/19/2000 D. Becker/P. Gortmaker
  http://www.scyld.com/network/ne2k-pci.html
PCI: Found IRQ 11 for device 00:0b.0
PCI: Sharing IRQ 11 with 00:07.2
eth0: RealTek RTL-8029 found at 0xe400, IRQ 11, 52:54:00:DB:FB:D4.
.
.
.
```

Si la recerca ha resultat infructuosa, en primer lloc hem de determinar quina targeta de xarxa tenim. Per a això, el millor és acudir a la documentació que venia adjunta amb la targeta o a la inspeccionar-la visualment. Si això no és possible, hi ha altres estratègies per a determinar quina és la nostra targeta, com pot ser polsar ALT + F2 per accedir a la consola i investigar el contingut del fitxer `/proc/pci` (mitjançant `cat`, per exemple), o podem recórrer a la informació que ens pugui proporcionar algun altre sistema operatiu que tinguem instal·lat a l'ordinador.

Una vegada coneguem quin tipus de targeta de xarxa és la que tenim, hem d'esbrinar quin mòdul és el que ens servirà per a accedir a la targeta. L'estratègia més segura per a fer-ho és recórrer a qual-sevol cercador, per exemple <http://www.google.es>, entrar paraules clau sobre la nostra targeta (*referència de la targeta NIC linux module*, per exemple) i llegir algunes de les pàgines trobades. També es pot recórrer a les pàgines de les principals distribucions de Linux, i entrar la referència de la targeta als seus cercadors. Com a últim recurs, es pot recórrer a la documentació de mòduls de xarxa del nucli, on s'especifica, per a totes les targetes suportades, el mòdul corresponent. També és bo saber si el fabricant ha desenvolupat el seu propi mòdul. Arribar a trobar un mòdul per a una targeta pot ser una tasca molt complicada, fins i tot impossible, ja que pot succeir que no hi hagi suport per a aquesta targeta o que calgui recórrer a

mètodes avançats per a poder-lo configurar; per aquest motiu, es recomana utilitzar sempre targetes com més estàndards millor.

Una vegada hàgim esbrinat el mòdul que necessitem, després d'instal·lar el nucli (5.3.10), hem de seleccionar la proposta que ens suggereixi el menú principal `Configure Device Driver Modules`. Després d'una pantalla d'avertiment, on se'ns recorda que molts programes de control ja estan inclosos en el nucli, entrarem a la pantalla de selecció de mòduls `Select Category` (podem accedir a aquesta interfície en qualsevol moment, executant el comandament `modconf`). Aquest comandament serveix com a *forntend* per a l'administració de programes de control que han estat compilats de manera modular juntament amb el nucli) i mitjançant els cursors seleccionarem l'opció `kernel/drivers/net`. Una vegada dins de la pantalla de selecció de mòduls de targeta de xarxa `Select Kernel/drivers/net modules`, seleccionem una altra vegada amb els cursors el mòdul que necessitem. Després de respondre que sí a la pregunta sobre si realment volem instal·lar aquest mòdul, podem deixar que l'`autoprobe` configuri el mòdul per a nosaltres, si no s'ha de passar cap paràmetre en concret al mòdul en qüestió. Passats uns instants, rebrem el missatge que ens indica si el mòdul s'ha instal·lat correctament o no.

5.4.4. Configuració de la xarxa

Una vegada sabem que la targeta de xarxa és operativa, al menú principal de la instal·lació seguirem el pas suggerit `Configure the Network` per procedir a la configuració de la xarxa. En primer lloc haurem de configurar el nom de node de l'ordinador, sense entrar el domini. A continuació, haurem d'entrar l'IP, la màscara de xarxa, la passarel·la, el nom de domini, i els servidors de DNS, fins a un nombre de tres, separats per espais.

5.4.5. Configuració d'apt

Una vegada s'ha configurat la xarxa, els passos següents que cal seguir són idèntics als seguits en la instal·lació per CD-ROM, fins a arribar a la configuració d'apt *Apt Configuration*. Llavors, en comptes de triar l'opció de CD-ROM, optarem per l'opció de xarxa que més ens convingui. A efectes pràctics és el mateix seleccionar el protocol

http que l'ftp. Després de fer la selecció se'ns preguntarà si volem usar paquets de tipus non-US; en principi, si estem fora dels Estats Units hem de respondre que no. Respecte a la resposta sobre la pregunta següent, referent a l'ús de paquets non-free, ja es deixa que el mateix lector prengui la decisió segons els seus principis ètics. Tot seguit se'ns pregunta de quin estat ha de ser la rèplica o *mirror* del qual apt haurà d'obtenir els paquets: hem d'escollir sempre el que ens sigui accessible de manera més ràpida, que acostuma a ser el més pròxim a nosaltres geogràficament parlant. Una vegada seleccionat l'estat, se'ns demana que seleccionem un servidor en concret (l'aplicació `netselect` destinada a facilitar l'elecció de servidors de paquets segons el criteri de velocitat d'accés). Quan s'ha resolt aquesta qüestió, se'ns mostra la pantalla de configuració d'accés per mitjà de servidor intermediari o *proxy*; si no hem d'usar aquest servei, deixarem en blanc la línia.

És important que els paquets crítics, quant a seguretat, s'obtinguin de servidors segurs. Per aquesta raó, es recomana que s'obtinguin en concret de `security.debian.org`.

Quan hàgim fet tots aquests passos, apt connectarà amb la rèplica que li hem especificat per configurar la seva base de dades. A partir d'aquest punt, per a la resta de la instal·lació seguirem els mateixos passos que en la instal·lació feta mitjançant CD-ROM.

5.5. Conclusió

En aquest taller hem après a instal·lar GNU/Linux al nostre ordinador. Encara que per ara el nostre sistema sigui molt bàsic, l'objectiu del taller s'ha complert plenament, ja que hem establert les bases per poder començar a treure partit de la flexibilitat i potència d'aquest sistema operatiu. En el pròxim taller aprendrem com configurar el sistema i a instal·lar-hi aplicacions noves per adaptar-lo i dotar-lo de totes les eines que estimem necessàries per a cobrir les nostres necessitats.

Nota

Si s'utilitza una connexió ADSL, pot ser que el nostre ISP disposi d'algun tipus de dispositiu tallafoc que faci impossible l'obtenció de paquets mitjançant http. Si ens trobem en aquest cas, el primer que hauríem de provar és si l'ftp no està filtrat.

6. Configuracions bàsiques

6.1. El sistema de connexió

Si encara no tenim configurat l'entorn gràfic, quan engeguem un sistema GNU/Linux ens apareixerà una pantalla de connexió (*login*) on es demanarà a l'usuari que s'identifiqui abans de començar a utilitzar el sistema. De fet, la majoria de distribucions llancen diverses consoles a les quals podem accedir a partir de ALT + F1, ALT + F2, etc. Això ens permet treballar simultàniament amb diferents comptes alhora, tenir diverses sessions obertes per a executar programes diferents, etc.

El programa que s'encarrega de gestionar cada una d'aquestes consoles és el `getty`. L'única cosa que fa aquest programa és obrir una connexió amb el dispositiu adequat (en el cas de les consoles de la pantalla, és el `/dev/ttyX`, on la "X" és el número de consola) i llançar l'aplicació de connexió. Aquest mecanisme ens permet molta flexibilitat, ja que el mateix programa `getty` permet comunicar amb diferents dispositius, de manera que podríem connectar un terminal pel port sèrie de l'ordinador, muntar una consola utilitzant la línia telefònica i un mòdem, etc.

Abans de llançar l'aplicació de connexió, es mostra un missatge de benvinguda per pantalla. Aquest missatge el podem configurar en el fitxer `/etc/issue`, escrivint el que vulguem. En aquest mateix fitxer també podem mostrar algunes de les variables del sistema referenciant-les de la manera següent:

"\d"	a data actual
"\s"	el nom del sistema operatiu
"\l"	el número de consola
"\m"	l'arquitectura de l'ordinador
"\n"	el nom de l'ordinador
"\o"	el nom del domini
"\r"	la versió del sistema operatiu
"\t"	l'hora actual
"\u"	el nombre d'usuaris actius en el sistema

Una vegada entrem al sistema, el programa de connexió s'encarrega de mostrar-nos el missatge del dia. Aquest missatge és el que hi ha escrit en el fitxer `/etc/motd`, que també podem canviar. Aquest mecanisme és molt útil per a informar tots els usuaris d'algun esdeveniment determinat, avisar-los d'algun problema, etc. Si un usuari vol suprimir aquest missatge, ho pot fer creant un fitxer buit anomenat `.hushlogin` en el seu directori d'usuari. Després de mostrar aquest missatge, el procés de connexió llança l'interpret d'ordres configurat per defecte per a l'usuari. El primer que fa l'interpret de comandaments és executar el contingut del fitxer `.profile` (que ha de ser en el directori de l'usuari). Aquest fitxer serveix perquè s'executin les instruccions configurades sempre que l'usuari entri al sistema. A més d'aquest `~/.profile`, també tenim el `/etc/profile`, que s'executa per a tots els usuaris del sistema i resulta molt útil per a poder configurar totes les opcions que volem sense haver de posar les instruccions dins de cada un dels `.profile` dels usuaris.

6.2. Explorant el bash

Si bé el fitxer `.profile` s'executa sigui quin sigui el directori d'ordres que utilitzem, els arxius `.bashrc` o `.bash_profile` se solen executar només quan utilitzem el bash (encara que es pot configurar a partir del mateix `.profile` de l'usuari, que és on es crida a l'execució d'aquest arxiu). Veurem algunes de les instruccions que podem trobar en aquests fitxers:

```
#CONFIGURACIONS BASIQUES
msg n
umask 022
#PATH
export PATH=      /usr/local/sbin:/usr/local/bin:/usr/
                  sbin:/usr/bin:/sbin:/bin:/usr/bin/X11

#PROMPT
export PS1='\h:\w\$ '
#ALIES D'USUARI
alias l='ls --color=auto'
alias ll='ls --color=auto -al'
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
alias v='vim'
```


Com veiem, en aquest arxiu podem definir el que vulguem. Les dues primeres instruccions del fitxer anul·len l'entrada de missatges d'altres usuaris i configuren els permisos que tindran els nous fitxers que crearem. La instrucció següent és la definició del *path*. El *path* són els directoris on tenim els comandaments, programes, aplicacions, etc. que volem poder cridar des de qualsevol lloc de la jerarquia de sistema de fitxers sense necessitat d'escriure la seva ruta completa (cada directori del *path* el separem amb ":"). La declaració següent és la del *prompt* del sistema. El *prompt* és la línia que apareix en el directori d'ordres abans del caràcter "#" (per al *root*) o "\$" (per als altres usuaris). Ens podem configurar aquest *prompt* de la manera que vulguem utilitzant les variables del sistema següents:

"\d"	la data del sistema
"\h"	el nom de la màquina
"\s"	el directori d'ordres que utilitzem
"\u"	el nom de l'usuari
"\v"	la versió del bash
"\w"	el directori actual
"\!"	el número d'història del comandament
"\\$"	apareix "#" si som el <i>root</i> o "\$" per als altres usuaris

Finalment, tenim els àlies d'usuari. Els àlies són sinònims, generalment per als comandaments que més utilitzem (per a no haver-los d'escriure completament). En un dels àlies que teníem en l'exemple anterior definíem que en escriure "l" s'executés "ls --color=auto". D'aquesta manera, podem utilitzar llargs comandaments sense haver-los d'escriure tots cada vegada que els utilitzem.

Tant en la definició del *path* com en la del *prompt* hem utilitzat el comandament `export`. Aquest comandament ens permet definir el que anomenem una *variable d'entorn*. Aquestes variables són utilitzades pel directori d'ordres per a fer certes operacions, desar algun tipus d'informació, etc. Podem veure totes les que hi ha declarades amb el mateix comandament `export`. Amb `set` i `unset` també podem manipular altres atributs que té l'interpret de comandaments. Algunes d'aquestes variables i atributs, que té per defecte el bash, són:

- `PWD`: directori actual.

Nota

Si volem executar els programes del directori des d'on estem situats sense necessitat de posar "." al principi, podem afegir aquesta entrada en la declaració del *path*. Igualment, si en el *path* no hi ha el programa que necessitem executar, en podem especificar la ruta completa en la línia de comandaments. De tota manera, no és recomanable afegir "." al *path* perquè pot representar un forat de seguretat.

Nota

Podem veure tots els àlies definits a partir del mateix comandament `alias`.

Nota

"echo \$NomVariable" permet veure el contingut d'aquestes variables i atributs.

- `BASH_VERSION`: versió del bash que utilitzem.
- `RANDOM`: genera un número aleatori diferent cada vegada que mostrem el seu contingut.
- `SECONDS`: nombre de segons que han passat des que hem obert el directori d'ordres.
- `HOSTNAME`: nom del sistema.
- `OSTYPE`: tipus de sistema operatiu que utilitzem.
- `MACHTYPE`: arquitectura de l'ordinador.
- `HOME`: directori de l'usuari.
- `HISTFILESIZE`: mida del fitxer d'història (nombre de comandaments que es desen).
- `HISTCMD`: número de comandament actual en la història.
- `HISTFILE`: fitxer on es desa la història de comandaments (generalment `.bash_history` del directori de l'usuari).

Amb la manipulació d'aquestes variables podem personalitzar molt més el nostre intèrpret de comandaments per adaptar-lo als nostres gustos i necessitats.

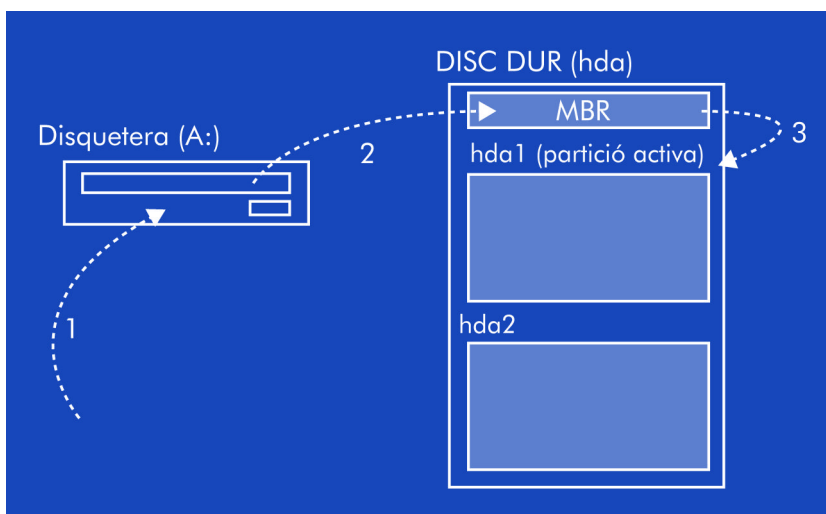
6.3. El sistema d'arrencada

Encara que amb la instal·lació del sistema operatiu ja es duu a terme la configuració i instal·lació d'un sistema d'arrencada, en aquesta secció veurem amb detall quines opcions ens proporcionen i com els hem de personalitzar i adaptar a les nostres necessitats. Encara que n'hi ha molts, el LILO i el GRUB són els més utilitzats en els entorns GNU/Linux, per la qual cosa ens centrarem en aquests dos.

Abans d'entrar en detalls sobre la configuració d'aquests dos programes, explicarem com funciona el sistema d'arrencada d'un PC estàndard. Tal com ja sabem, a partir de la BIOS o EFI de l'ordinador en podem configurar la seqüència d'arrencada. En general, aquesta seqüència sol començar buscant a la unitat de disquets i continua

amb el CD/DVD i el disc dur. Encara que puguem instal·lar el LILO o el GRUB en un disquet o al sector d'arrencada d'un CD, és més usual instal·lar-lo al disc dur per no haver d'introduir el disc cada vegada que engeguem el nostre ordinador.

Quan el sistema d'arrencada de l'ordinador busca al disc dur, el primer que inspecciona és si l'MBR (*Master Boot Record*) del primer disc dur (màster del primer canal IDE o el primer disc del canal SCSI) conté alguna indicació del sistema que cal carregar. L'MBR és la primera pista del disc dur, allà on es desa la informació de les particions configurades i, opcionalment, el programa encarregat d'iniciar el sistema operatiu. Si aquí no es troba aquest programa, s'inspecciona el sector d'arrencada de la partició activa del disc. Sempre que vulguem instal·lar un programa en el sistema d'arrencada de l'ordinador, l'hem de situar en alguna d'aquestes zones. En la figura següent podem veure tot aquest procés quan en la seqüència d'arrencada primer és la unitat de disquets i després el disc:



Sempre que instal·lem un sistema d'arrencada, hem de tenir en compte que l'ordre amb què es fa aquesta seqüència és important: si n'instal·lem un en l'MBR i un altre en la partició activa, s'executarà el de l'MBR perquè la BIOS o EFI inspecciona primer aquesta zona. Si no tenim partició activa, hem de situar el programa d'arrencada en l'MBR. De tota manera, el més recomanable és instal·lar sempre el programa en l'MBR perquè és el primer que s'inspecciona. Encara que puguem tenir instal·lats altres sistemes operatius en altres discos, hem d'instal·lar el LILO o GRUB en alguna d'aquestes zones. En la

configuració del programa ja indicarem on estan situats els operatius que vulguem carregar.

6.3.1. LILO

El LILO ens permet múltiples configuracions diferents per a fer gairebé qualsevol acció amb el sistema d'arrencada de l'ordinador. Per a no estendre'ns innecessàriament, en aquesta secció només explicarem les opcions més importants per poder configurar un sistema d'arrencada de múltiples operatius tenint en compte la seva seguretat. En el manual del programa i del seu fitxer de configuració ("man lilo" i "man lilo.conf") trobarem l'especificació completa per a totes les altres opcions. El fitxer de configuració del LILO és el `/etc/lilo.conf`. Quan executem `lilo`, el programa busca tota la informació en aquest fitxer i instal·la el programa d'arrencada a la zona que li indiquem. En aquest fitxer de configuració tenim dues seccions: la global i les locals a cada sistema operatiu. En la secció global trobem totes les configuracions generals per al programa. En cada línia podem posar una directiva de configuració per indicar algun tipus d'informació al programa. Les directives més importants d'aquesta secció són les següents:

- `boot = DISPOSITIU`: li indiquem a quin dispositiu volem instal·lar el LILO.

Si volem instal·lar-lo en l'MBR del primer disc del canal IDE, a `DISPOSITIU` posarem `/dev/hda`; si ho volem en la primera partició, `/dev/hda1`, etc. També tenim l'opció d'instal·lar-lo en un disquet, indicant el dispositiu `/dev/fd0`.

- `default = NOM`: indiquem el sistema operatiu que es carregarà per defecte. El `NOM` ha de correspondre a alguna de les etiquetes configurades en cada sistema de les seccions locals.
- `prompt`: amb aquesta directiva indiquem al programa que abans de carregar el sistema mostri un *prompt* per pantalla perquè puguem seleccionar el que vulguem. Si no activem aquesta opció, no podrem seleccionar l'operatiu que cal carregar i automàticament s'iniciarà el que tinguem per defecte.

- `delay = TSECS`: especifica el nombre de dècimes de segon que el LILO esperarà abans de carregar el sistema operatiu per defecte (la directiva `default` ha d'existir). Aquesta opció és vàlida quan només tenim un sistema operatiu al disc; si utilitzem `prompt`, haurem de configurar-lo amb `timeout`.
- `timeout = TSECS`: dècimes de segon que el LILO esperarà en el `prompt` abans d'engegar el sistema operatiu per defecte (han d'existir la directiva de `prompt` i `default`).
- `lba32`: aquesta opció ens serveix per a poder arrencar qualsevol partició del disc dur.

Si desactivem aquesta directiva, no podrem engegar sistemes operatius en particions situades en sectors superiors del disc (a partir del cilindre 1.024) i haurem de posar l'arrencada del sistema en una petita partició a l'inici del disc (més informació a "`man lilo.conf`"). Totes les BIOS a partir del 1998 suporten aquesta opció, de manera que és molt recomanable utilitzar-la per a estalviar-nos problemes.

Les directives locals són les que corresponen a cada un dels sistemes operatius instal·lats en les particions dels nostres discos. La primera directiva ha de ser o bé `image` o `other`. Les expliquem a continuació (juntament amb altres directives possibles):

- `image = CAMI`: el `CAMI` ens indica l'arxiu que conté la imatge nucli del sistema operatiu. Aquesta opció només és vàlida per a sistemes GNU/Linux o similars. Generalment, les distribucions creen un directori `/boot/` on es posa la imatge del nucli del sistema, encara que també és usual posar un enllaç en l'arrel anomenat `/vmlinuz` (o similar). En aquesta directiva podem especificar qualsevol dels dos.
- `other = CAMI`: dispositiu que conté un sistema operatiu diferent de GNU/Linux o similar.
- `label = NOM`: etiqueta per al sistema operatiu. Aquesta directiva és obligatòria per a totes les seccions locals que tinguem configurades, ja que si no, no les podríem identificar de cap manera.
- `alias = NOM`: sinònim per a la mateixa secció local. En podem tenir tants com vulguem.

Contingut complementari

El sistema d'arrencada d'un sistema GNU/Linux o similar ens permet, amb un mateix sistema de fitxers, carregar diversos nuclis diferents. D'aquesta manera, sense la necessitat d'instal·lar de nou el sistema, podem treballar amb nuclis diferents. Per a configurar-lo només hauríem d'especificar dues seccions locals posant, en cada una, quin nucli utilitzaríem.

Contingut complementari

Amb el LILO o el GRUB podem passar paràmetres al nucli Linux en el moment d'arrencar. Això és molt útil quan volem fer alguna operació específica en el sistema; per exemple, passant `single` o `1` s'iniciaria el sistema en el `runlevel 1`, amb `root=/dev/hda3` especificaríem l'arrel del sistema de fitxers, etc.

- `root = DISPOSITIU`: aquesta opció només és vàlida per als sistemes GNU/Linux o similars. Se li especifica a quin dispositiu és situada l'arrel del sistema de fitxers (*root filesystem*). Aquesta opció es passa com a argument en carregar el nucli del sistema i és opcional; si no se li passa res, s'agafa la que té configurada per defecte la imatge del nucli.

Hi ha algunes de les directives del LILO que també es poden configurar directament sobre la imatge del nucli de l'operatiu. Amb el comandament `rdev` podem configurar les possibles opcions d'una imatge nucli Linux. Algunes són l'arrel del sistema de fitxers (si no el configurem amb el LILO, s'agafa el que hi ha en la imatge), el mode `vga`, etc.

Si volem protegir adequadament el sistema d'arrencada de l'ordinador, hem d'afegir algunes directives més en aquest fitxer de configuració. El primer que hem de fer és protegir la BIOS o EFI amb una contrasenya i configurar la seqüència d'arrencada perquè només es pugui fer a partir del disc dur. Amb això passem tota la responsabilitat al LILO. A partir de les directives `password =CONTRASENYA` i `restricted`, en podem configurar la seguretat de cinc maneres diferents:

Contrasenya global	Hem de posar la directiva de contrasenya a la secció global i en engegar qualsevol dels sistemes es demanarà la contrasenya.
Contrasenyes locals	En els sistemes que vulguem que ens demanin contrasenya hem de posar la directiva <code>password</code> i quan s'iniciïn en la demanaran.
Contrasenya global restringida	Hem de posar la directiva de <code>password</code> i <code>restricted</code> a la secció global i només quan engeguem algun sistema passant algun paràmetre al nucli es demanarà la contrasenya.
Contrasenya global i local restringida	La contrasenya es posa a la secció de global i només posem el <code>restricted</code> en els sistemes en què volem que es demani contrasenya en passar algun paràmetre al nucli.
Contrasenya global i local restringides	És el mateix que la configuració anterior, però amb l'avantatge que podem configurar diferents contrasenyes per als diferents sistemes que volem engegar.

Hem de tenir en compte que si utilitzem la directiva `password`, la contrasenya s'escriu al fitxer com un text, de manera que haurem d'eliminar els permisos de lectura del `lilo.conf` per a tots els usuaris menys el `root`. Finalment, després d'escriure aquest fitxer de configuració podem instal·lar el LILO al sector d'arrencada configurat. Per a fer-ho, només hem d'executar "lilo". Si el volguéssim desinstal·lar, li hauríem de passar el paràmetre "-u".

A continuació mostrem un exemple d'aquest fitxer de configuració, preparat per a engegar un sistema GNU/Linux i un altre WindowsTM:

```
lba32
boot = /dev/hda
prompt
timeout = 50
message = /etc/message
default = debian
restricted
password = contrasenya

    image = /vmlinuz
    label = debian
    root = /dev/hda1

    other = /dev/hda3
    label = w2000
```

Tota aquesta configuració serveix per a una arrencada estàndard del sistema. També hi ha un altre tipus d'arrencada que utilitza una imatge anomenada de *RAM Disk* (`initrd`). Aquest altre tipus d'arrencada serveix per a fer una configuració modular del nucli Linux. És molt utilitzat quan necessitem un nucli amb alguna configuració especial, per a incloure mòduls en el mateix nucli, per a fer una imatge d'arrencada per a un CD *live*, per a tenir una mateixa imatge per a tots els ordinadors d'un laboratori emmagatzemada en un únic servidor, etc. De tota manera, les instal·lacions estàndard del sistema operatiu no utilitzen gairebé mai aquest tipus d'arrencada. Si ens volem crear una imatge d'aquest tipus, ens en podem informar en el manual d'`initrd` i en el del programa `mkinitrd`.

**Contingut
complementari**

El GRUB (GRand Unified Bootloader) és el programa d'arrencada del projecte GNU.

6.3.2. GRUB

Igual que el LILO, el GRUB també ens serveix per a instal·lar un programa a la zona d'arrencada que vulguem de l'ordinador. A diferència del LILO, té moltes més possibilitats que el fan molt versàtil: permet tenir un petit intèrpret de comandaments en arrencar l'ordinador, ens permet accedir als arxius de les particions del disc sense carregar cap operatiu, etc. Com en el cas anterior, en aquesta secció només en veurem la configuració bàsica. Si volguéssim aprofundir més en el seu ús, podríem recórrer al seu manual o al HOWTO corresponent.

El sistema d'arrencada del GRUB es carrega en dues fases. Generalment, amb la instal·lació del paquet, s'incorporen dos fitxers corresponents a aquestes dues fases. Si volem que en arrencar el GRUB no ens mostri cap menú per a seleccionar l'operatiu que volem carregar, només hem d'executar el programa `grub` i executar-lo en l'intèrpret de comandaments que ens mostra

```
$ install (hd0,0)/PATH/stage1 d (hd0) (hd0,0)/PATH/stage2
```

Aquesta instrucció instal·la el GRUB en l'MBR del disc mestre del primer canal IDE. La manera com es referencien els discos varia una mica de com es fa en GNU/Linux i amb el LILO. En "hdX" la "X", en lloc de "a", "b", .. ., és "0", "1", etc. Per a les particions també es comença amb el número "0" per a denominar la primera i a diferència de `hda1`, s'ha d'escriure `(hd0,0)` i consecutivament per a les altres. Llegint la instrucció d'aquesta manera ens fixem que el primer paràmetre serveix per a designar on és l'arxiu de la primera fase del GRUB (li indiquem la partició corresponent, directori `-PATH-` i fitxer `-stage1-`). Generalment, quan instal·lem el paquet del GRUB també s'afegeixen aquests dos fitxers per a cada una de les fases de càrrega (solen estar situats a `/usr/share/GRUB/i386-pc/`). El paràmetre `d (hd0)` indica que la primera fase del GRUB s'instal·larà en l'MBR del primer disc. L'última opció especifica on és situat el fitxer per a la segona fase de càrrega, que és executada per la primera.

Amb aquesta configuració, en reiniciar l'ordinador apareixerà, per defecte, l'interpret de comandaments del GRUB: podem manipular molts aspectes del disc, arrencar el sistema operatiu que vulguem, etc. Si volem engegar un sistema GNU/Linux hem d'escriure les instruccions següents:

```
$ kernel (hd0,0)/vmlinuz root=/dev/hda1 $ boot
```

Amb la primera indiquem on se situa la imatge nucli (amb els paràmetres que vulguem) i amb la segona iniciem el procés de càrrega de l'operatiu. Si optem per un menú de selecció per no haver d'escriure aquests comandaments cada vegada que engeguem l'ordinador, podem generar un fitxer de menú com el següent (els comentaris comencen per "#"):

```
#Especificació de l'operatiu que es carregarà per defecte.
#Aquest número està en correspondència amb l'ordre dels
# sistemes que hi ha en les seleccions locals als operatius.
default 0
#Indiquem que esperi 10 segons abans de carregar el sistema
# configurat per defecte
timeout 10
#Configuració d'arrencada per a un sistema GNU/Linux
title Debian GNU/Linux
kernel (hd0,0)/vmlinuz root=/dev/hda1
#Configuració d'arrencada per a un sistema Windows
title W2000
root (hd0,2)
makeactive
```

Per a instal·lar el GRUB amb aquest menú d'arrencada, hauríem d'executar la mateixa instrucció que anteriorment però afegint-hi el paràmetre `p (hd0,0)/PATH/menu.lst` amb el disc, camí i fitxer de menú. Per a protegir el sistema d'arrencada (que desprotegir és encara més perillós que amb el LILO), podem posar la di-

Nota

Per a veure tots els comandaments disponibles en el directori d'ordres del GRUB podem prémer TAB. També s'inclouen ajuts per a tenir una referència completa de tots els comandaments.

Nota

Una altra manera d'instal·lar el GRUB és utilitzant el programa `grub-install`.

directiva de `password CONTRASENYA` a la secció global del fitxer de configuració. D'aquesta manera, quan des del menú es vulgui entrar en el directori d'ordres del GRUB es demanarà la contrasenya. Com en el cas del LILO, si utilitzem aquesta directiva és molt important que només el `root` pugui llegir aquest arxiu de configuració (encara que en aquest cas també hi ha l'opció de posar la contrasenya xifrada amb md5).

6.4. Accés a altres particions i dispositius

Els sistemes tipus UNIX tracten tots els dispositius de l'ordinador com si fossin fitxers. Això ens permet molta flexibilitat, ja que podem aprofitar tots els mecanismes i funcions que utilitzàvem amb els fitxers i aplicar-los als dispositius. En el directori `/dev/` tenim tots els dispositius reconeguts pel sistema. Si el sistema no reconeix adequadament un dispositiu o en volem crear un d'especial, el comandament `mknod` ens permet fer aquesta classe d'operacions, encara que és important saber exactament què volem fer abans d'utilitzar-lo, ja que el seu mal ús podria danyar parts del sistema.

Per a les unitats d'emmagatzematge, el sistema ens proporciona un altre tipus d'operació per poder accedir als seus sistemes d'arxius, la de muntatge. Per a aquesta operació utilitzarem els comandaments `mount` i `umount`, que situen (munten) o desmunten tot el sistema de fitxers d'un dispositiu/unitat determinat en un directori existent del sistema. La manera bàsica d'utilitzar el comandament és "mount dispositiu directori", on dispositiu pot referenciar qualsevol dispositiu del canal IDE o SCSI (`/dev/hdXX`, `/dev/sdXX`), la unitat de disquets (`dev/fdX`), cintes de *backup*, etc., i directori és la ubicació on muntarem l'estructura de fitxers del dispositiu. És recomanable que el directori on muntem aquests dispositius sigui buit, ja que quan s'utilitza com a punt de muntatge no s'hi pot accedir.

Per a desmuntar un d'aquests dispositius podem utilitzar "umount directori", on directori ha de ser el punt de muntatge utilitzat. Si muntem dispositius com un disquet o CD, és important no treure el dispositiu del suport, ja que abans hem d'avisar el sistema perquè actualitzi la memòria del sistema de fitxers del dispositiu. Igualment, tampoc no podem desmuntar el dispositiu si algun usuari o aplicació

Contingut complementari

Per a la unitat de disquets i CD/DVD moltes distribucions ja creen un directori per defecte on muntar-los (`/floppy/` o `/mnt/floppy/` i `/CD-ROM/` o `/mnt/CD-ROM/`). També se sol proporcionar el directori `/mnt/`, on podem crear directoris per a altres dispositius que tinguem en el sistema.

utilitza algun dels seus arxius o directoris (en intentar-ho, el sistema donaria un missatge d'error).

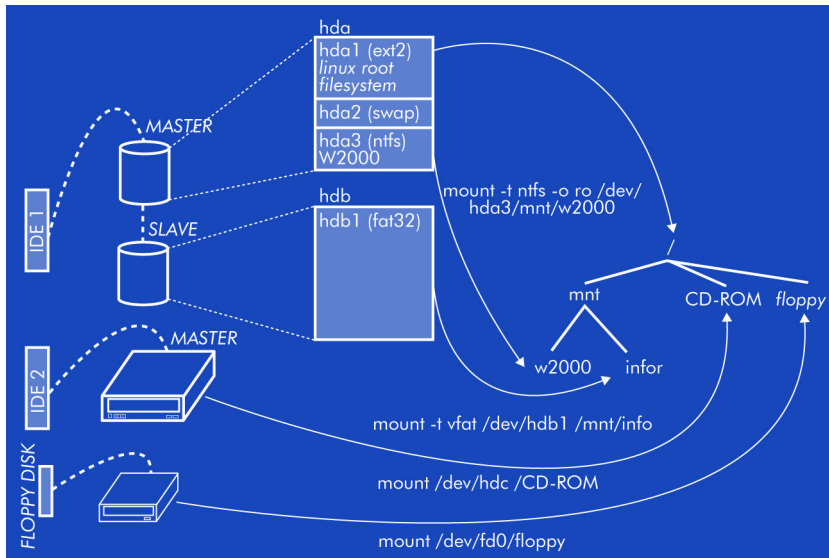
Ens fixem que amb el comandament de muntatge no especifiquem en cap moment el tipus de sistema de fitxers utilitzat en la unitat que volem muntar, de manera que s'haurà de determinar automàticament. Si ho volem especificar manualment, podem passar al comandament `mount` el paràmetre “-t tipus”, on tipus podria ser algun dels de la taula següent (en podeu veure en el manual de `mount` la llista completa):

Tipus	Sistema
ext	GNU/Linux (versions de nucli anteriors a 2.1)
ext2	GNU/Linux (versions de nucli posteriors a 2.1)
ext3	GNU/Linux (versions de nucli posteriors a 2.2 o 2.4)
swap	Sistema de swap de GNU/Linux
sysv	Sistemes tipus UNIX
minix	MINIX
iso9660	Sistema de fitxers que utilitzen la majoria de CD
nfs	Sistema de fitxers remot (<i>network file system</i>)
smbfs	Sistema de fitxers remot en xarxes Windows TM (<i>samba file system</i>)
ntfs	Família de WindowsNT TM
msdos	MS-DOS TM
vfat	Família de Windows95 TM

A més de passar el tipus de sistema de fitxers utilitzat per la unitat que vulguem muntar, també podem indicar altres opcions que ens puguin ser molt útils en determinades situacions (sempre precedides per “-o” i, si volem passar-ne més d’una, separades per comes):

Significat de l'opció	Permetem	No permetem
Execució de binaris	exec	noexec
Ús del bit de SetUserld	suid	nosuid
Fitxers de només lectura	ro	rw
Sistema sincronitzat (ús de memòria de disc)	sync	async
Interpretació de caràcters o blocs especials	dev	nodev
Permís que qualsevol usuari munti o desmunti el dispositiu	user	nouser
Sistema de tipus swap	sw	
(si passéssim <code>defaults</code> s'utilitzarien les opcions <code>rw</code> , <code>suid</code> , <code>dev</code> , <code>exec</code> , <code>auto</code> , <code>nouser</code> i <code>async</code>)		

En la figura següent podem veure un exemple d'utilització d'aquest comandament per a muntar diversos dispositius diferents:



A més d'aquests comandaments per a muntar i desmuntar unitats, el sistema operatiu ens proporciona una altra manera de fer el mateix i tenir sempre una configuració determinada segons la unitat. En el fitxer `/etc/fstab` podem desar aquesta informació de manera que cada línia indicarà una unitat amb el seu directori de muntatge i les opcions que vulguem configurar. La sintaxi de cada una d'aquestes línies serà:

```
<disp> <dir> <tipusSistema> <opcions> <dump> <ordre>
```

En el primer camp hem d'especificar el dispositiu tal com fèiem amb el comandament de `mount` i el segon serà el directori on vulguem muntar la unitat indicada. En el camp de `tipus Sistema` podem especificar el sistema de fitxers que utilitza la unitat o bé `auto` perquè el detecti automàticament. A `opcions` podem escriure les mateixes que utilitzàvem amb el comandament de `mount`, separades per comes si en posem més d'una. Una opció útil d'aquest camp és la configuració d'`auto` o `noauto`, amb la qual cosa indiquem al sistema que munti automàticament (o no) la unitat en arrencar. El camp de `dump` indica si volem fer còpies de seguretat (més informació en el manual de `dump`). Si no utilitzem aquest sistema, podem posar un "0". L'últim camp serveix per a indicar l'ordre de muntatge de les unitats. Si li posem un "0", indiquem que l'ordre no és important.

Nota

Amb el comandament "mount -a" es muntarien tots els dispositius que tinguessin l'opció d'`auto` activada (que ho està per defecte) d'aquest fitxer de configuració.

ANOTACIONS

L'arrel del sistema de fitxers és el primer que s'ha de muntar, per la qual cosa en aquest camp hi hauria d'haver un "1".

Una entrada que sempre veurem en aquest fitxer i que ens pot sorprendre és la del directori `/proc/`, que té un significat especial. Realment, el que hi ha en aquest directori no són fitxers, sinó el valor de moltes de les variables que utilitza el nucli del sistema. Seguint la mateixa polífrica de l'operatiu, amb la qual tot s'ha de poder referenciar com un arxiu, en el directori `/proc/` també podem manipular variables internes del nucli com si es tractés de fitxers.

Encara que tot aquest disseny per a muntar les unitats en l'estructura jeràrquica de directoris és molt potent i ens permet molta flexibilitat, en alguns casos no és gaire pràctic. Per exemple, cada vegada que volem copiar un arxiu en un disquet haurem de muntar la unitat, copiar-la i desmuntar-la de nou. Per aquesta raó, existeixen algunes altres aplicacions que ens faciliten tot aquest procés per a estalviar-nos alguns passos. Una d'aquestes són les `mttools`, que és un paquet amb diverses aplicacions que ens permeten copiar directament arxius en un disquet i des d'un disquet, i algunes altres eines interessants. També existeix un paquet anomenat `autoifs`, que detecta automàticament la inserció d'algun dispositiu en el sistema i els munta sense necessitat d'escriure cap comandament.

6.5. Configuració de dispositius

Encara que en els inicis de GNU/Linux no era així, actualment cada vegada més fabricants proporcionen programadors de control per als seus dispositius especials de GNU/Linux. Abans d'intentar configurar-los, n'hem de buscar informació en els mateixos manuals del sistema, en els mòduls, a Internet, etc. per estalviar-nos problemes en la seva posada en marxa. Encara que actualment la majoria de dispositius tenen HOWTO, manuals o algun tipus de documentació, és important que abans de comprar-ne un de nou, ens informem adequadament de si existeix algun programador de control disponible per a ell.

Encara que en aquesta secció només veurem com configurar alguns dels dispositius més freqüentment utilitzats als ordinadors personals,

segons la distribució que utilitzem, el procés pot variar significativament. En tota la secció ens hem basat en el sistema que s'utilitza en Debian GNU/Linux, encara que si aprenem des de la base com funciona aquest procés, no hauríem de tenir problemes per a buscar quina és la configuració que s'utilitza en altres distribucions.

6.5.1. El teclat

La configuració correcta del teclat és un aspecte molt important per a poder solucionar qualsevol problema que ens sorgeixi amb ell. En primer lloc, hem de saber que quan el sistema arrenca, es carrega un mapa de caràcters corresponent al configurat en el procés d'instal·lació. Aquest mapa de caràcters se sol trobar situat a `/etc/console/boottime.kmap.gz` o en algun altre directori de `/etc/`. Si canviéssim de teclat, només hauríem de canviar aquest fitxer amb el que correspongués al nou teclat i en arrencar de nou ja es carregaria el mapa nou. Tots els mapes de caràcters existents se solen situar dins del directori `/usr/share/keymaps/` ordenats per arquitectures d'ordinadors i països. Concretament, el que utilitzem a Espanya amb ordinadors basats en l'arquitectura i386 i un teclat estàndard el trobaríem a `i386/qwerty/es.kmap.gz`.

Encara que tots aquests mapes de caràcters estiguin comprimits en format gzip, podem descomprimir-los i canviar alguna de les seves entrades per adaptar-les a les nostres necessitats. En cada línia trobem la directiva `keycode`, que indica que definim una tecla, indicada en el número que segueix la directiva (podem saber quin número correspon a cada una de les tecles a partir del comandament `showkey`). Després d'aquesta definició, tenim els significats que té la tecla prement-la sola, amb el SHIFT, etc.

Vegem-ho amb un exemple:

```
keycode 53 = minus underscore control
keycode 53 = Delete
```

En la primera línia s'indica quin caràcter correspon en prémer la tecla sola (minus) o prement el SHIFT (underscore). La segona ens mostra la funció que s'executarà en cas de prémer la tecla juntament amb la de CTRL (s'eliminarà el caràcter següent). Tota la informació

Contingut complementari

Amb el programa `consolechars` podem carregar la font que volem al terminal. Hem de diferenciar clarament el que és una font i el que és el mapa de caràcters: el mapa ens determina quin significat té cada tecla, mentre que la font és només la representació gràfica que li donem. Tota la configuració de les fonts de caràcters se sol trobar a `/etc/console-tools/config`.

necessària per a configurar correctament un arxiu d'aquest tipus la podem trobar en el manual de `keymaps`, molt útil quan ens trobem amb algun problema amb teclats especials o d'altres països. Si no volem reiniciar el sistema en canviar aquest fitxer de mapa, podem utilitzar el comandament `loadkeys` (`dumpkeys` ens mostra les configurades).

Un altre aspecte relacionat amb el teclat és el tema de les dièresis, accents, etc. Tot això ho podem configurar a partir del fitxer de `/etc/inputrc` (totes les directives possibles d'aquest fitxer les tenim especificades en el manual de `readline`). La que ens pot ser més útil és la de `convert-meta`, que desactivant-la (`set convert-meta off`) ens permet utilitzar els accents i dièresi (per al català).

Finalment, una altra configuració important (indirectament relacionada amb el teclat) és la de `locales`. Amb `locales` podem configurar la zona o zones geogràfiques que hi ha per a poder utilitzar tecles especials del teclat, veure les dates en el format a què estem acostumats, etc. Aquesta configuració la utilitzen moltes de les llibreries del sistema, de manera que en molts comandaments i aplicacions del sistema s'utilitzarà la seva configuració per a adaptar algunes funcions al nostre entorn local. La seva configuració la podem trobar a `/etc/locale.gen` i podem utilitzar els comandaments `locale-gen` i `locale` per a veure-la o actualitzar-la.

6.5.2. Tarjeta de xarxa (tipus Ethernet)

Per a configurar una nova tarjeta de xarxa (tipus Ethernet), el primer que hem de fer és afegir el mòdul del nucli necessari perquè es reconegui adequadament. Encara que en algunes targetes és possible que no hàgim de fer aquest pas perquè el mateix nucli ja pot estar compilat per a reconèixer les més habituals, ens hem d'assegurar (abans de comprar la tarjeta) que existeix el programador de control o mòdul necessari per a ella.

Una vegada el sistema reconeix la tarjeta, ja la podem configurar com vulguem.

En el fitxer `/etc/network/interfaces` podem especificar tota la seva configuració, on també tindrem la de les altres interfícies del sis-

Contingut complementari

Amb `discover -module ethernet` podem saber quin mòdul necessita la nostra tarjeta de xarxa. Si volem deixar-lo configurat perquè es carregui sempre l'hauríem d'escriure a `/etc/modules` (si no, el podem inserir amb `modprobe` o `insmode`).

Nota

Una altra manera de denominar les targetes de xarxa és amb la sigla de *network interface card* (NIC).

Nota

Una altra manera de reconfigurar el *keymap* i les locals a Debian és utilitzant "apt-reconfigure console-data" o "apt-reconfigure locales" respectivament.

Nota

A partir de les versions 2.2 del nucli Linux, ja es pot utilitzar una nova infraestructura de xarxa denominada *iproute2*, amb la qual podem manipular tots els aspectes relacionats amb les nostres NIC i les taules internes que utilitza l'operatiu per a manejar tot el que està relacionat amb la xarxa.

tema. Una interfície és un dispositiu (real o lògic) relacionat amb la xarxa a partir del qual el sistema es pot comunicar amb altres ordinadors, oferir uns serveis determinats, etc. Són les portes que té el sistema per a poder-se comunicar. Per a cada interfície reconeguda en el sistema, en aquest fitxer se li especifiquen les directives necessàries per al seu funcionament correcte.

Ho veurem amb un exemple:

```
#Interfície de loopback
auto lo
iface lo inet loopback

#NIC
auto eth0
iface eth0 inet static
    address 192.168.0.10
    netmask 255.255.255.0
    network 192.168.0.0           #opcional
    broadcast 192.168.0.255     #opcional
    gateway 192.168.0.1        #opcional
```

La primera entrada que trobarem en aquest fitxer sol ser per a la interfície de *loopback*. Aquesta interfície no es correspon amb cap targeta ni dispositiu real de l'ordinador, sinó que és un mecanisme de l'operatiu que li permet utilitzar els protocols de comunicació internament. D'aquesta manera, si provem funcions de la xarxa sense comunicar-nos amb cap altre ordinador, no fa falta ni tan sols tenir una targeta de xarxa instal·lada. En totes les entrades trobem la directiva d'*auto* abans d'especificar la configuració del dispositiu. Aquesta directiva indica que la targeta es pot muntar automàticament quan el sistema arrenca. La directiva d'*iface* especifica el tipus de targeta i protocol que s'utilitzarà per mitjà de la sintaxi següent: "*iface dispositiu familia-Protocol metodeConfiguracio*". Amb les targetes Ethernet el dispositiu serà *ethX*, on la "X" serà un número que comenci per "0", que indica el número de targeta instal·lada a l'ordinador. La família del protocol de comunicació utilitzat amb la targeta sol ser qualsevol de les següents:

- *inet*: IPv4, utilitzat a Internet i la majoria de xarxes locals.

- `inet6`: IPv6, la nova versió d'IPv4, que s'instaura a poc a poc.
- `ipx`: per a xarxes Novell™.

Finalment, en l'últim camp s'indica com s'obté la configuració de xarxa de la targeta (la seva adreça, la xarxa on és, la passarel·la que cal utilitzar, etc.). En la taula següent podem veure quines són aquestes opcions per a la família de protocols `inet`:

Config	Opcions	Descripció
<code>loopback</code>		Mètode per a definir la interfície de <i>loopback</i> (s'ha d'utilitzar amb la interfície <code>lo</code>).
<code>static</code>		Mètode per a configurar una NIC amb una adreça IP estàtica.
	<code>address</code>	Adreça IP de la interfície. Camp requerit.
	<code>netmask</code>	Màscara de l'adreça IP. Camp requerit.
	<code>broadcast</code>	Adreça de <i>broadcast</i> . Si no s'especifica, es calcula automàticament.
	<code>network</code>	Adreça d'identificació de xarxa. Només requerida per a versions del nucli 2.0.X.
	<code>gateway</code>	Adreça IP del <i>gateway</i> que utilitzem per a aquesta interfície.
<code>dhcp</code>		Mètode per a configurar remotament l'IP de tots els ordinadors d'una xarxa local (<i>dynamic host configuration protocol</i>).
	<code>hostname</code>	IP del servidor de DHCP.
	<code>leasehours</code>	Temps, en hores, de lloguer de l'IP (passat aquest temps, es renova).
	<code>leasetime</code>	Temps, en segons, de lloguer de l'IP.
	<code>vendor</code>	Identificador de tipus de servidor (en general, <code>dhcpd</code>).
	<code>client</code>	Identificador del tipus de client (en general, <code>dhcpd</code>).
<code>bootp</code>		Mètode per a configurar remotament l'IP de tots els ordinadors d'una xarxa local (<i>BOOT protocol</i>). Actualment s'utilitza més DHCP.
	<code>bootfile</code>	Fitxer que s'utilitza en el moment d'arrencada.
	<code>server</code>	Adreça IP del servidor de BOOTP.
	<code>hwaddr</code>	Adreça MAC del servidor BOOTP.
<code>ppp</code>		Mètode utilitzat amb el protocol punt a punt o <i>point to point protocol</i> , usat als mòdems.
	<code>provider</code>	Proveïdor del servei.

Encara que en aquest capítol no entrarem en xarxes de computadors, hem de saber que disposem de molts comandaments per a manejar la configuració de xarxa del sistema operatiu. Els més importants són

`ifconfig`, amb el qual podem veure els dispositius configurats, `ifdown` i `ifup`, que ens permet apagar o encendre la interfície que volem, i `route`, que ens mostra la taula d'encaminament o *routing* del sistema.

6.5.3. Targeta WiFi

Les xarxes de comunicació sense cables són cada vegada més freqüents, tant en instal·lacions domèstiques com en institucions, escoles o empreses. Per a instal·lar-les, s'ha de disposar del que denominem *punts d'accés*, que són uns dispositius connectats a la xarxa física de la institució. Aquests punts d'accés permeten que, a partir d'unes targetes PCMCIA, qualsevol ordinador del seu al voltant es pugui connectar a la xarxa. D'aquesta manera, se simplifica molt el cablatge dels edificis.

Perquè el nostre GNU/Linux detecti i configuri adequadament una targeta sense cable o *wireless* (de tipus PCMCIA) hem d'afegir al nucli del sistema els mòduls `orinoco_cs` i `hermes`, que en molts casos ja està compilat en el nucli mateix. Tota la configuració d'aquestes targetes es desa als fitxers del directori `/etc/pcmcia/`. L'únic que hem d'afegir en el fitxer `/etc/pcmcia/config.opts` és (depenent de la targeta hauríem de canviar la directiva de `card`):

```
card " Conceptronic Wireless"
    version "802.11", "11Mbps Wireless LAN Card"
    bind "orinoco_cs"
```

Amb això ja aconseguim que el sistema reconegui la targeta sense cable com un dispositiu més del sistema. Si no tenim cap altra targeta Ethernet instal·lada, es referenciarà com a `eth0`, `eth1` si ja en tenim una, etc. A continuació, l'únic que ens faltaria perquè la targeta es connecti al punt d'accés serà editar el fitxer `/etc/network/interfaces` i afegir-li la configuració necessària perquè se li assigni una IP. Naturalment, aquesta interfície del sistema la podem tractar com qualsevol altra, utilitzant els mecanismes de tallafoc o *firewall* del sistema, amb les aplicacions `iproute2`, etc.

6.5.4. Mòdems

Per a configurar un mòdem generalment se sol utilitzar l'aplicació `pppconfig`, que escriu els arxius de configuració necessaris per al DAEMON del sistema `ppp`, que és el programa encarregat d'establir la connexió a Internet. Amb `pppconfig` (o aplicacions similars) sempre s'han de seguir uns passos determinats, que detallem a continuació:

- 1) Nom del proveïdor: el proveïdor és l'empresa amb què tenim el contracte de connexió a Internet. Aquest nom serveix per a poder identificar cada connexió que configurem de manera única.
- 2) Configuració de servidors de noms: quan establim el contracte amb el nostre proveïdor, generalment se solen proporcionar l'IP o les IP dels servidors de noms que s'han d'utilitzar. Si tenim aquestes IP, hem d'indicar que utilitzem una configuració estàtica, i seguidament ens les demanaran. Només en cas que el nostre proveïdor ens indiqui que la configuració de DNS és dinàmica, hem d'escollir aquest altre tipus de configuració. Amb la tercera opció, que ens informa que DNS serà tractat per altres mitjans, podem utilitzar la configuració del fitxer `/etc/resolv.conf`.
- 3) Mètode d'autenticació: el mètode d'autenticació pot ser PAP o CHAP. Generalment, els proveïdors solen utilitzar el PAP (*peer authentication protocol*), encara que si no funcionés ens hauríem d'informar adequadament.
- 4) Nom d'usuari i contrasenya: aquesta és la informació que ens proporciona el proveïdor per poder-nos connectar i accedir als seus serveis.
- 5) Velocitat del mòdem: segons quin mòdem tinguem, podrem accedir a Internet a una velocitat més gran o més petita. Actualment, tots van a 115.200 bps, per la qual cosa és més recomanable deixar el valor "115200". Si tinguéssim un mòdem més lent, ja es detectaria i reconfiguraria automàticament en el moment de la connexió.
- 6) Trucada amb polsos o tons: la majoria de centraletes telefòniques ja funcionen amb tons, encara que en determinades zones rurals encara s'utilitza l'antic sistema de polsos.

Contingut complementari

Els paquets que contenen els programes necessaris per a la connexió a Internet amb un mòdem se solen anomenar `ppp` i `pppconfig`.

Contingut complementari

Amb `pppconfig` tenim un menú que ens permet afegir, modificar o eliminar connexions. Els passos que mostrem es corresponen amb la inserció d'una nova connexió.

- 7) Número de telèfon: aquest número també l'ha de proporcionar el proveïdor d'Internet.
- 8) Port de comunicació: el port de comunicació és el port en què tenim connectat el mòdem. Si li indiquem que el detecti automàticament, revisarà tots els ports i el configurarà automàticament. Si no, el podem indicar amb `/dev/ttySX`, on la "x" és un 0 per al COM1, un 1 per al COM2, etc.

Tota aquesta configuració se sol emmagatzemar als arxius situats en el directori `/etc/ppp/`. Encara que també puguem editar aquests fitxers i canviar les directives manualment, és més recomanable utilitzar alguna aplicació automàtica, ja que la seva configuració és bastant complexa. Per a establir la connexió amb el nostre proveïdor, hauríem d'iniciar el DAEMON executant `/etc/init.d/ppp start`. Per a parar-lo, podem utilitzar `/etc/init.d/ppp stop`.

6.5.5. Targeta de so

La targeta de so necessita que s'insereixi un mòdul del nucli del sistema per a poder funcionar correctament. Si tenim instal·lada l'aplicació `discover`, podem descobrir quin mòdul és el que correspon a la nostra targeta per mitjà del comandament `discover --module sound`. Per a instal·lar el mòdul, podem utilitzar els comandaments `insmod` o `modprobe`, i si el volem deixar configurat permanentment, l'hauríem d'escriure en el fitxer `/etc/modules`.

Encara que amb la inclusió del mòdul corresponent ja podrem utilitzar la targeta de so adequadament, generalment també se sol instal·lar la infraestructura de so ALSA (*advanced Linux sound architecture*). Normalment, la majoria de distribucions la solen incloure per defecte, encara que si no és així, es pot instal·lar amb el paquet corresponent.

6.5.6. Impressora

En GNU/Linux, la configuració d'impressores es pot fer amb moltes aplicacions diferents. Encara que l'`lpd` (Line Printer DAEMON) va ser un dels primers programes de gestió d'impressió que van aparèixer en els sistemes tipus UNIX, actualment n'hi ha molts de més

Contingut complementari

ALSA és un projecte que ha desenvolupat molt programari relacionat amb aplicacions de tractament de so, mòduls nous per al nucli Linux, etc.

fàcils de configurar i gestionar. A continuació, comentem alguns dels més utilitzats:

- `lpd`: un dels primers DAEMONS d'impressió dels sistemes tipus UNIX. La seva configuració s'ha de fer manualment.
- `lpr`: la versió de BSD de l'`lpd`. És molt recomanable utilitzar algun tipus de filtre automàtic com ara el `magicfilter` o l'`apsfilter` per a configurar les impressores. Aquest tipus de filtre detecta automàticament el tipus de fitxer que s'ha d'imprimir i prepara la impressió adequadament (utilitza un filtre anomenat *IFHP*).
- `lprng`: aplicacions basades en `lpr` amb l'avantatge que incorpora una eina de configuració denominada `lprngtool`, que permet fer la configuració de manera gràfica i senzilla.
- `gnulpr`: la versió de GNU del sistema d'impressió `lpr`. També incorpora eines gràfiques de configuració, gestió dels serveis, etc.
- CUPS: de Common UNIX Printing Systems, aquest conjunt d'aplicacions és compatible amb els comandaments de `lpr` i també serveix per a xarxes Windows™. Utilitza un conjunt de filtres propis i suporta la gran majoria d'impressores del mercat.

Encara que totes aquestes aplicacions tinguin els seus propis mètodes de configuració, totes utilitzen el fitxer `/etc/printcap` per a desar-la. Generalment, també utilitzen algun tipus de DAEMON perquè el sistema d'impressió sigui operatiu. El DAEMON es pot configurar perquè l'ordinador a què estigui connectada la impressora serveixi com a servidor d'impressió. D'aquesta manera, diversos ordinadors de la mateixa xarxa podran utilitzar la mateixa impressora, amb la qual cosa s'estalviaran recursos. Per als clients d'impressió es poden utilitzar els mateixos programes especificant, en la configuració, que la impressora és remota (generalment s'ha de proporcionar l'IP del servidor d'impressió i la cua).

Si volem configurar un servidor d'impressió per a xarxes Windows™ o configurar una impressora d'un servidor Windows™ des d'un client GNU/Linux, hem d'utilitzar un altre tipus de programes. Samba és un conjunt d'aplicacions de GNU/Linux que utilitza els protocols de les

Contingut complementari

En configurar un servidor d'impressió, és important que configurem adequadament des de quines màquines/usuaris permetem la impressió. D'una altra manera, un atacant podria aprofitar la vulnerabilitat i aprofitar els nostres recursos, deixar la impressora sense paper, etc.

**Contingut
complementari**

Swat (*samba web administration tool*) és una eina molt útil per a configurar un servidor de samba.

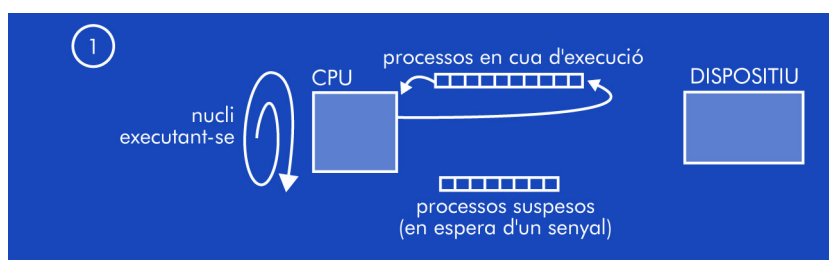
xarxes Windows™. Encara que les seves funcionalitats vagin molt més allà de la configuració d'un servidor o client d'impressió, per a poder utilitzar impressores en Windows™ haurem de fer servir aquest conjunt d'aplicacions o bé les que ens proporcionin CUPS.

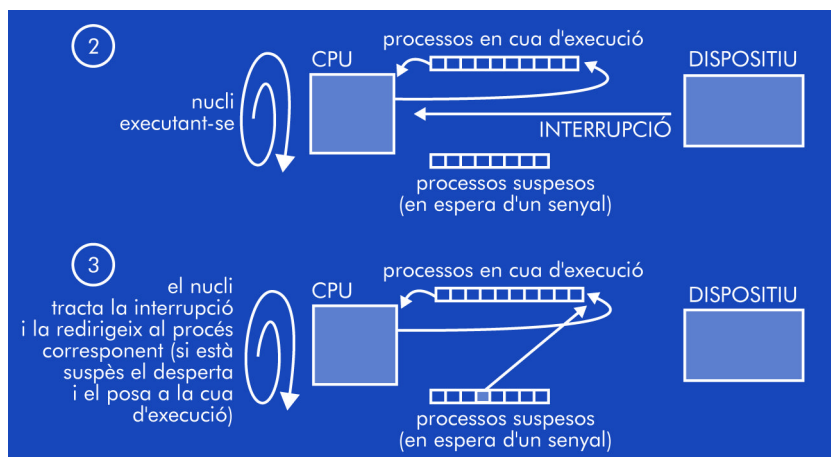
7. DAEMONS i runlevels

7.1. Els DAEMONS

Com ja sabem, GNU/Linux ens permet executar simultàniament tants processos com vulguem repartint equitativament el temps de la CPU entre ells. De fet, el mecanisme de maneig de processos també ha de tenir en compte el que s'anomenen *interrupcions*. Una interrupció és un senyal que arriba al nucli del sistema des de qualsevol dels dispositius que tenim instal·lats al nostre ordinador. Aquestes interrupcions solen estar vinculades a algun procés en concret, de manera que el nucli ha de despertar el procés en qüestió (si no està en execució) i redirigir-li la interrupció perquè la processi adequadament. Un exemple típic d'interrupció és quan premem una tecla del teclat o movem el ratolí: en fer-ho, el dispositiu envia un senyal que ha de ser redirigit cap a l'aplicació corresponent perquè sigui tractada adequadament.

Per a poder manejar adequadament totes les interrupcions que es produeixen, el nucli no escolta permanentment els dispositius del sistema esperant els seus senyals. En lloc de fer-ho, el sistema executa les operacions dels processos en la cua d'execució i només quan es produeix una interrupció atén el dispositiu que l'ha generat. Això s'ha de fer d'aquesta manera a causa de la gran diferència de velocitat entre els dispositius del sistema i la CPU. El tractament d'interrupcions és fonamental per a qualsevol sistema operatiu, ja que és aquest mecanisme, entre d'altres, el que ens permet mantenir en execució tants processos com vulguem i, quan arribin les interrupcions, despertar els processos que les esperen.





Un DAEMON (*Disk And Execution Monitor*) és un procés que, generalment, tenim carregat en la memòria, esperant algun senyal (provinent d'una interrupció de dispositiu o del nucli mateix) per a despertar-se i executar les funcions necessàries per a tractar-la. Encara que aquesta definició també pugui encaixar amb altres processos del sistema (llançats pels usuaris o pel mateix sistema), un DAEMON també se sol ajustar a aquesta forma d'execució (encara que en alguns casos especials, no). D'aquesta manera, els DAEMONS que tinguem carregats no ocuparan la CPU mentre no sigui estrictament necessari, i encara que en tinguem en memòria sempre podrem treballar amb l'ordinador sense problemes.

Encara que un DAEMON sigui un procés com qualsevol altre que s'executi en mode *background*, la manera com els organitzem i tractem sí que és diferent de la resta de comandaments i programes del sistema. Generalment, tots els DAEMONS tenen un *shell script* situat en el directori `/etc/init.d/` que ens permet iniciar-lo, parar-lo o veure el seu estat d'execució. Per a fer algunes d'aquestes funcions hem d'executar el *shell script* corresponent al DAEMON que vulguem tractar passant-li algun dels paràmetres següents:

- `start`: per a iniciar el DAEMON. Si aquest ja s'executa, es mostra un missatge d'error.
- `stop`: per a parar el DAEMON. Si no s'executa, es mostra un missatge d'error.
- `restart`: reinicia el DAEMON. Serveix perquè es tornin a llegir els seus arxius de configuració.

Contingut complementari

Els *shell scripts* dels DAEMONS no són més que una eina per a facilitar tot el seu procés d'arrencada, parada, etc. En alguns casos, també podem utilitzar el mecanisme i l'organització d'aquests per a poder executar certes operacions que ens interessin (escrivint un *shell script* que s'executi en entrar en un nivell d'execució determinat).

ANOTACIONS

- `reload`: encara que no tots els DAEMONS ho permetin, aquest paràmetre serveix per a poder recarregar els arxius de configuració sense haver-lo de parar.

La majoria d'aquests *scripts* utilitzen un programa anomenat `start-stop-daemon` que ens proporciona el sistema operatiu i que serveix per a tractar d'aquests processos. És habitual que en administrar un servidor hàgim de dissenyar-nos els nostres propis DAEMONS per a executar alguna tasca concreta. En el directori on se situen tots els *shell scripts* dels DAEMONS també se sol trobar un d'exemple (`/etc/init.d/skeleton`) perquè el puguem utilitzar quan necessitem configurar-ne un de nou que no sigui en la distribució. Generalment solen estar programats de la manera següent:

```
#!/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:
    /usr/sbin:/usr/bin
DAEMON=/usr/sbin/daemon
NAME=daemon

DESC="some daemon"

test -x $DAEMON || exit 0
set -e
case "$1" in
    start)
        echo -n "Starting $DESC: $NAME"
        start-stop-daemon --start --quiet --pidfile \
            /var/run/$NAME.pid --exec $DAEMON
        echo "."
    ;;
    stop)
        echo -n "Stopping $DESC: $NAME "
        start-stop-daemon --stop --quiet --pidfile \
            /var/run/$NAME.pid --exec $DAEMON
        echo "."
    ;;
    restart|force-reload)
        echo -n "Restarting $DESC: $NAME"
        start-stop-daemon --stop --quiet --pidfile \
            /var/run/$NAME.pid --exec $DAEMON
        sleep 1
        start-stop-daemon --start --quiet --pidfile \
```

Contingut complementari

Per a executar un DAEMON, l'hem de cridar amb la seva ruta completa `/etc/init.d/nombreDaemon` i passar-li el paràmetre que ens interessi. Algunes distribucions incorporen el comandament `service`, que permet fer el mateix sense haver d'especificar la ruta completa.

Contingut complementari

Encara que els siguin programes com qualsevol altre, la seva programació difereix una mica de les aplicacions d'usuari perquè han d'incloure funcions per a quedar suspesos i esperar senyals perquè siguin despertats, etc.

```

/var/run/$NAME.pid --exec $DAEMON
echo "."
;;
*)
N=/etc/init.d/$NAME
    echo " Usage: $N {start|stop|restart|
        force-reload}" >&2

exit 1
;;
esac
exit 0

```

Com hem vist, en les variables declarades a l'inici del *shell script* especifiquem quin `PATH` és necessari per al procés del `DAEMON`, el programa que s'ha d'executar (`DAEMON`), el nom que li donem (`Name`, que ha de ser igual que el nom del *shell script*) i la seva descripció (`Desc`). L'únic que fa el codi en arrencar el `DAEMON` és escriure en el directori `/var/run/` un fitxer amb el PID del procés. En parar-lo, es buscarà aquest PID i s'enviarà el senyal d'acabament al procés corresponent. Naturalment, trobarem *shell scripts* preparats per a fer moltes més operacions amb el `DAEMON` que hem de tractar, encara que com a mínim tots han de tenir aquesta estructura.

7.2. Els *runlevels*

Els `DAEMONS` que tinguem executant-se en un determinat moment ens marquen els serveis que el sistema operatiu ofereix i/o rep. El fet que puguem tenir tants `DAEMONS` diferents fa que n'hàgim de plantejar l'organització adequadament. Entendrem un *runlevel* (o nivell d'execució) com l'execució d'uns `DAEMONS` determinats que al seu torn proporcionen uns serveis concrets. En la instal·lació d'un servidor és habitual dissenyar una configuració perquè en moments concrets es puguin oferir uns serveis determinats i en d'altres no. Per a permetre aquest tipus de funcionament, el sistema operatiu ens proporciona diferents nivells d'execució que podem adaptar a les nostres necessitats.

Si bé podem configurar el nombre de nivells d'execució que volem i la funcionalitat de cada un d'ells, generalment els sistemes tipus UNIX ens en proporcionen sis de diferents amb les propietats següents:

Nivell	Funcionalitat
0	El nivell d'execució 0 està configurat per a aturar el sistema.
1	Aquest nivell és denominat <i>single user</i> , ja que només permet l'entrada al sistema al seu <i>root</i> . S'engeguen els DAEMONS mínims i serveix per a tasques de manteniment.
2~5	Els nivells del 2 al 5 estan destinats a ser configurats segons les necessitats de cada instal·lació. En instal·lar el sistema, per defecte tots són iguals. A aquests nivells també se'ls diu <i>multiusuari</i> , ja que, per defecte, permeten que més d'un usuari treballi en el sistema.
6	L'últim nivell està preparat per a reiniciar el sistema. És molt semblant al 0 però s'hi afegeix una funció de reinici.

El comandament necessari per a canviar de nivell d'execució és `init` (li passem com a paràmetre el nivell d'execució que vulguem) i per a veure en quin som, `runlevel`.

Els comandaments `halt`, `reboot`, `shutdown` o `poweroff` l'únic que fan és cridar el nivell d'execució 0 o 6 fent, abans, alguna operació concreta (vegeu-ne el manual per a més informació). Només el *root* del sistema pot utilitzar tots aquests comandaments.

La manera com s'organitzen aquests DAEMONS en cada nivell d'execució és molt simple. Cada nivell d'execució té un directori situat a `/etc/rcX.d/` on la "X" és el número de nivell. En aquests directoris trobem enllaços simbòlics als *shell scripts* dels DAEMONS situats a `/etc/init.d/`, que ens serveixen per a indicar al sistema si volem iniciar o parar el DAEMON que apunten. Amb el mateix nom de l'enllaç s'identifica l'acció que s'ha de fer: si l'enllaç comença per "S" (*Start*), indiquem que volem iniciar el DAEMON, mentre que si comença per "K" (*Kill*), indiquem que el volem aturar. Si el nom no comença per cap d'aquestes lletres, el sistema no fa res amb l'enllaç. Després d'aquesta lletra es posa un número de dues xifres entre "00" i "99", que n'indica l'ordre d'inici o d'aturada. Aquest ordre és important, ja que alguns DAEMONS necessiten que d'altres estiguin en execució abans de ser iniciats.

En canviar de nivell d'execució, el sistema inspeccionarà els DAEMONS del directori corresponent i començarà parant, primer, els DAEMONS indicats i després iniciarà els altres. L'únic que es fa és cridar el DAEMON passant-lo com a paràmetre `start` o `stop`, de manera que si en parem algun que no s'executi en el moment d'aturada, no passarà res perquè el mateix *shell script* ho té en compte. Això ens serveix per a poder canviar de nivell d'execució sense tenir en compte el nivell anterior en què érem. En la figura següent podem veure un exemple de configuració per a tres nivells d'execució:

Nivell d'execució 2		DAEMONS executant-se
K50sshd	S10syslogd	<i>syslogd</i>
K51apache-ssl	S12kerneld	<i>kerneld</i>
K52ftpd	S20dhcpd	<i>dhcpd</i>
K53telnet	S50proftpd	<i>proftpd</i>
	S90apache	<i>apache</i>
Nivell d'execució 3		
K50dhcpd	S10syslogd	<i>syslogd</i>
K51proftpd	S12kerneld	<i>kerneld</i>
K52apache	S20sshd	<i>sshd</i>
K53ftpd	S50apache-ssl	<i>apache-ssl</i>
K53telnet		
Nivell d'execució 4		
K50dhcpd	S10syslogd	<i>syslogd</i>
K51proftpd	S12kerneld	<i>kerneld</i>
K52apache	S20ftpd	<i>ftpd</i>
K53ftpd	S50telnet	<i>telnet</i>
K53telnet		

En el fitxer `/etc/inittab` tenim definida tota la configuració dels *runlevels*: el nivell d'execució per defecte, el nombre de consoles disponibles en cada un dels nivells, etc. Cada línia del fitxer és una directiva amb la sintaxi: "`<id> :<runlevels> : <action> : <process>`". El primer camp és l'identificador de la directiva; seguidament trobem en quins nivells d'execució és vàlida aquesta directiva, l'acció que ha de fer i el procés que ha de llançar. En l'exemple següent expliquem com configurar algunes d'aquestes directives:

```
# El nivell d'execució per defecte (en aquest cas, el 2)
id:2:initdefault:

# Scripts que s'han d'executar en arrencar el sistema (abans
#d'entrar en el nivell d'execució per defecte) si::sysinit:/etc/
init.d/rcS
```

```

# Programa que es crida en entrar al nivell d'execució

# single user (l'acció wait indica que es llança el
# procés i no es fa res més)
~~:S:wait:/sbin/sulogin

# Configuració dels diferents nivells d'execució
# disponibles en el sistema
10:0:wait:/etc/init.d/rc 0
11:1:wait:/etc/init.d/rc 1
12:2:wait:/etc/init.d/rc 2
13:3:wait:/etc/init.d/rc 3
14:4:wait:/etc/init.d/rc 4
15:5:wait:/etc/init.d/rc 5
16:6:wait:/etc/init.d/rc 6

# Comandament que s'ha d'executar en prémer CTRL + ALT + DEL
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now

# Definició de les consoles obertes a cada
# nivell d'execució (l'acció respawn indica
# que en acabar l'execució del procés
# getty es llanci una altra vegada)

1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6

```

Com veiem, en aquest fitxer es configura tot el que es refereix als nivells d'execució de manera molt flexible, de manera que podem canviar el que ens interressi per a adaptar-lo millor a les nostres necessitats. Ens fixem que, encara que aquí definim el nivell d'execució per defecte, també el podríem especificar en engegar el sistema amb el LILO o GRUB. Això és molt útil, per exemple, quan tenim problemes greus en el sistema que no ens permeten arreglar-los adequadament; si arrenquem amb el primer nivell (passant "1" o

Contingut complementari

En aquest fitxer també podríem configurar un terminal que es comunicués amb el sistema a partir d'un mòdem i un altre ordinador amb la directiva "T1:23:respawn:/sbin/mgetty -x0 -s 57600 ttyS1". D'aquesta manera, podríem tenir una consola del sistema en un altre terminal comunicant-nos amb una línia telefònica.

“single”), només s’iniciaran les funcions més necessàries i podrem entrar per a arreglar el que faci falta.

7.3. L’arrencada del sistema

El procés pare de tots els altres és l’`init`. Aquest procés s’encarrega d’engegar els altres que tinguem en el nivell d’execució configurat. Tanmateix, abans d’entrar en aquest nivell s’executen tots els *shell scripts* de `/etc/rcS.d/` (configurat a `/etc/inittab`), que poden ser o bé DAEMONS com els dels altres *runlevels* o simplement *shell scripts* necessaris per al sistema (càrrega del mapa de caràcters, càrrega dels mòduls del nucli, etc.). Si en volem eliminar algun, hem de saber exactament què fem, ja que generalment són imprescindibles per al bon funcionament de l’operatiu.

Una vegada s’han engegat aquests DAEMONS (o *shell scripts*), s’entra en el nivell d’execució configurat per defecte, parant i iniciant-hi els DAEMONS especificats. Quan s’hagi après tota aquesta organització, ja podrem adaptar l’arrencada del sistema a les nostres necessitats creant i situant els DAEMONS que vulguem a qualsevol dels llocs que hem vist.

7.4. DAEMONS bàsics

Segons la distribució de GNU/Linux que utilitzem, el mateix procés d’instal·lació ja configura uns DAEMONS o d’altres. Tot i així, totes les distribucions solen incorporar el DAEMON per al sistema de *logs* i el de l’execució periòdica i retardada d’aplicacions (encara que les seves configuracions poden variar una mica). En aquesta secció veurem com funcionen aquests tres DAEMONS bàsics i com els podem configurar. És important saber manejar-los perquè ens poden ajudar molt en algunes de les tasques d’administració.

7.4.1. Logs de sistema (sysklogd)

Els *logs* del sistema són fitxers de traça que un DAEMON de l’operatiu s’encarrega de generar perquè quedi constància de qualsevol acció que s’hi faci. El DAEMON encarregat de fer aquestes tasques és

Contingut complementari

El nucli del sistema també llança un DAEMON per gestionar els seus *logs* denominat `klogd`.

el `syslogd`, la configuració del qual trobem a `/etc/syslog.conf`. Cada línia d'aquest fitxer consisteix en una regla amb dos camps: el selector i l'acció. Amb el selector configurem de quin servei volem tractar els *logs* i el seu nivell de prioritat. L'acció serveix per a indicar cap on volem redirigir els *logs* (a un fitxer, a una consola, etc.). En les taules de la pàgina següent podem veure les diferents opcions vàlides per a aquests camps.

Generalment, tots els fitxers de *logs* del sistema se solen emmagatzemar en el directori `/var/log/`. Encara que la majoria de fitxers de *logs* siguin de text i els puguem veure amb qualsevol editor, en podem trobar algun d'especial que no desii les seves dades en aquest format. Generalment solen ser els fitxers `/var/log/wtmp` i `/var/log/btmp`, que són els *logs* d'entrada d'usuaris en el sistema i d'entrades errònies respectivament. Per a veure aquests dos fitxers, podem utilitzar els comandaments `last` i `lastb`. Si tinguéssim configurats aquests *logs* en algun altre fitxer, també els podríem veure passant el paràmetre `-f fitxer` al comandament `last`.

Contingut complementari

Per a veure els últims registres d'entrada dels usuaris també podem utilitzar el comandament `lastlog`.

Selector			
Servei	Significat	Prioritat	Significat
authpriv	Missatges d'autoritzacions o d'aspectes de seguretat.	emerg	El sistema és inutilitzable.
cron	DAEMON <code>crond</code> i atd.	alert	L'acció s'ha de fer tot seguit.
DAEMON	DAEMONS del sistema sense opcions de <i>logs</i> .	crit	Condicions crítiques
ftp	DAEMON del servidor FTP (<i>file transfer protocol</i>)	err	Condicions d'error
kern	Missatges del nucli del sistema.	warning	Condicions d'emergència.
lpr	Missatges del subsistema d'impressió.	notice	Notícies normals, però importants.
mail	Missatges del subsistema de correu (si el tenim configurat).	info	Missatges d'informació
news	Missatges del subsistema de notícies (si el tenim configurat).	debug	Missatges de <i>debugging</i> .
syslog	<i>Logs</i> generats pel mateix DAEMON <code>syslogd</code> .		
user	<i>Logs</i> d'aplicacions de nivell d'usuari.		
uucp	Missatges generats pel sistema d'UUCP (<i>unix-to-unix copy protocol</i>)		
local0~7	Reservats per al seu ús local.		

(Els serveis i prioritats es poden combinar com es vulgui.)

Acció	
Destinació	Explicació
Fitxer regular	S'especifica la ruta completa del fitxer. Posant un "-" al davant no es requereix sincronitzar el fitxer cada vegada que s'hi escriu (encara que es perdrà el log en cas de fallar l'alimentació).
Pipe anomenat	Aquest sistema permet que els missatges es redirigeixin cap a una canonada creada abans d'iniciar el DAEMON de <code>sysklogd</code> amb el comandament <code>mkfifo</code> . S'indica posant el caràcter " " abans del nom del fitxer. És molt útil per a operacions de <i>debugging</i> de programes.
Consola	Especificant <code>/dev/ttyX</code> , on "X" és un número de consola, o <code>/dev/console</code> , els logs es redirigeixen a la pantalla especificada.
Màquina remota	Per a especificar que els logs es redirigeixin a una màquina remota, hem de precedir el nom de l'ordinador central remot amb "@".
Usuaris	Especificant el nom d'usuari o usuaris (separats per comes) els logs corresponents s'hi redirigeixen.
Usuaris en línia	Amb "*" especificarem que els logs es redirigeixin a tots els usuaris que en el moment d'ocórrer el log estiguin dins del sistema. Això s'utilitza per a avisar tots els usuaris que ha passat alguna acció crítica en el sistema.
(Les accions es poden posar en tots els selectors que es vulgui.)	

Contingut complementari

Si volguéssim configurar una consola del sistema per a veure tots els logs que es generen, podríem afegir la línia `*.* /dev/ttySX` (on "X" és la consola on volem veure els logs) al fitxer `/etc/syslog.conf` i reiniciar el DAEMON `sysklogd`.

Contingut complementari

Segons el servidor que administrem haurem de tenir en compte la legalitat vigent (segons els països), que en alguns casos obliga a conservar els fitxers d'algun tipus de logs durant un determinat període de temps.

Aquesta manera de tractar els logs permet molta flexibilitat per a configurar-los adequadament quan instal·lem un servidor, tasca molt important per a tenir controlats els aspectes que més ens interessin del sistema. Tot i així, si haguéssim de desar tots els logs que es generen en un servidor, segurament al final saturaríem el disc per la mida sempre creixent d'aquests arxius. Per a evitar-ho s'utilitza un sistema de rotació de logs, que consisteix a comprimir, cada cert temps, aquests fitxers i desar-los només fins a una antiguitat determinada. Encara que generalment se solen comprimir cada setmana i es desen només els d'un o dos mesos anteriors, podem configurar tot això a partir del fitxer `/etc/logrotate.conf`. Els logs de certs servidors i/o aplicacions també es poden configurar explícitament per a tenir un control més adequat del que fan. La configuració personalitzada de logs per a aquestes aplicacions se sol situar a `/etc/logrotate.d/`.

Internament, el sistema utilitza uns programes per a manejar més amenament tot aquest sistema de logs. Amb `logger` podem escriure en el sistema de logs del sistema. `saveolog` i `logrotate` serveixen per a desar i, opcionalment, comprimir alguns dels fitxers de logs que tenim (amb el segon podem configurar més opcions que amb el primer). Aquests comandaments també es poden utilitzar per a crear els nostres propis fitxers de logs o, si és necessari, manipular

manualment els del sistema (amb el seu manual obtindrem més informació sobre el seu tractament i la seva manipulació).

7.4.2. Execucions periòdiques (cron)

Moltes de les tasques d'administració d'un servidor s'han de dur a terme periòdicament. També hi ha moltes accions, com l'actualització dels logs o les bases de dades internes que utilitzen certs comandaments, que s'han d'executar regularment perquè funcionin correctament. Per aquest motiu, és molt important que el mateix operatiu ens proporcioni alguna eina per poder configurar eficientment totes aquestes execucions periòdiques.

El DAEMON `cron` és el que s'encarrega de manejar tot el sistema d'execucions periòdiques. La seva organització és molt simple: en el fitxer `/etc/crontab` es desa la configuració interna del DAEMON i en els directoris `/etc/cron.daily/`, `/etc/cron.weekly/` i `/etc/cron.monthly/`, els *shell scripts* dels programes que s'executaran diàriament, setmanalment o mensualment respectivament. També existeix el `/etc/cron.d/`, on podem situar arxius amb un format especial per a configurar l'execució de determinats programes de manera més flexible.

Generalment, en el fitxer `/etc/crontab` trobem les directives següents:

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:
    /sbin:/bin:/usr/sbin:/usr/bin
#m h dom mon dow user  command
25 6 * * * root test -e /usr/sbin/anacron ||
    run-parts --report /etc/cron.daily
47 6 * * 7 root test -e /usr/sbin/anacron ||
    run-parts --report /etc/cron.weekly
52 6 1 * * root test -e /usr/sbin/anacron ||
    run-parts --report /etc/cron.monthly
```

La definició de les variables `SHELL` i `PATH` serveixen per a indicar al DAEMON quin intèrpret de comandaments ha d'utilitzar i quin serà el seu `PATH`. Les tres línies següents estan formades pels

Contingut complementari

Moltes de les aplicacions del sistema necessiten algun tipus d'actualització periòdica, generalment configurada a partir del `cron`. Si no tenim l'ordinador encès tot el dia, és important que el configurem adequadament perquè es facin en algun moment en què sapiguem que l'ordinador estarà encès.

Contingut complementari

Si instal·lem GNU/Linux en un ordinador que no està en funcionament tot el dia, és recomanable tenir instal·lat el programa *anacron* perquè executi els *scripts* configurats amb el *cron* adequadament (encara que sigui en hores diferents de les previstes).

Contingut complementari

Si utilitzem el fitxer */etc/crontab* per a configurar les nostres pròpies execucions periòdiques, cada vegada que el modifiquem haurem de reiniciar el DAEMON *cron*. Si utilitzem la seva estructura de directoris, no ens caldrà.

Contingut complementari

Utilitzant el comandament *crontab*, en gravar el fitxer es comprova que la sintaxi sigui correcta.

camps: "*<minut> <hora> <diaMes> <mes> <diaSetmana> <usuari> <comandament>*". Els cinc primers indiquen quan s'ha d'executar el comandament corresponent (han de ser coherents) i en el sisè trobem l'usuari que s'utilitzarà per a executar el comandament especificat en l'últim. Fixem-nos que en el fitxer de configuració els comandaments que s'executen una vegada al dia, una vegada a la setmana o una vegada al mes són els encarregats de llançar els *shell scripts* que es trobin en els directoris especificats. Si existeix el programa *anacron*, s'executen amb aquest; si no s'utilitza el *run-parts*, que encara que no tingui tantes funcionalitats com l'*anacron*, també serveix per a poder executar tots els *shell scripts* que es trobin en un directori determinat. Aquesta configuració és la que ens permet tota l'estructura de directoris que hem comentat anteriorment. Si volguéssim, podríem canviar les directives d'aquest arxiu per adaptar-les més a les nostres necessitats.

Encara que també podríem utilitzar aquest mateix fitxer de configuració per a posar els nostres propis comandaments, és més recomanable utilitzar l'estructura de directoris que ens proporciona el mateix DAEMON. L'únic que no apareix en aquesta configuració és el de */etc/cron.d/*, que el DAEMON ja té en compte automàticament. En aquest directori podem situar arxius exactament amb la mateixa sintaxi que en el */etc/crontab* per a programar execucions personalitzades. D'aquesta manera, la flexibilitat és total.

Si per a les tasques d'administració periòdiques del sistema és recomanable utilitzar tota aquesta estructura, quan són els usuaris els que volen configurar alguna tasca periòdica és més usual fer servir fitxers particulars per a cada un d'ells. Amb el comandament *crontab* podem passar els paràmetres "*-u USER -e*" i automàticament s'editarà el fitxer de configuració particular per a l'usuari especificat. Els fitxers particulars dels usuaris es desen en el directori */var/spool/cron/crontabs/* (de fet, el fitxer */etc/crontab* és el mateix que el particular del *root*). Per a poder limitar quins usuaris poden utilitzar aquest DAEMON, podem editar els fitxers */etc/cron.allow* i */etc/cron.deny*, on podem posar, respectivament, la llista dels usuaris als quals permetem utilitzar el *cron* i a qui no.

ANOTACIONS

7.4.3. Execucions retardades (at i batch)

Si bé el `cron` ens permet fer operacions cada cert període de temps, el DAEMON `atd` permet executar un comandament o aplicació en un moment determinat. Igual que amb el DAEMON anterior, podem configurar quins usuaris el poden utilitzar o no a partir dels fitxers `/etc/at.allow` i `/etc/at.deny`. En aquest cas, no tenim fitxer de configuració explícit per al DAEMON, sinó que és amb el comandament `at` amb què podem especificar en quin moment volem executar certa operació amb la sintaxi: “`at -f fitxer TEMPS`”. Generalment el fitxer sol ser un programa o *shell script* creat pel mateix usuari on s’escriuen totes les instruccions que es vulguin executar. L’especificació de `TEMPS` pot arribar a ser molt complexa, pot determinar una `HORA` amb el format “`hh:mm`”, un temps a partir del moment d’execució amb “`now + XX minutes`”, etc. (en el seu manual s’especifiquen tots els possibles formats).

Amb `atq` podem veure quines feines tenim retardades i amb `atrm` en podem esborrar alguna que sigui a la cua. Finalment, si volem executar tots els treballs a la cua de l’`at`, podem utilitzar el comandament `atrun`. Aquest ens permet passar el paràmetre “`-l LOADAVERAGE`”, on `LOADAVERAGE` ha de ser un número que indiqui a partir de quin moment de càrrega del sistema es podran executar els comandaments retardats. Això enllaça directament amb el comandament `batch`, que serveix exactament per al mateix que l’`at` i segueix la seva mateixa sintaxi, però sense necessitat d’especificar un temps concret d’execució. Les operacions configurades en aquesta cua es portaran a terme quan la càrrega del sistema baixi a menys d’1,5.

D’aquesta manera, quan necessitem executar un determinat comandament en una hora concreta, hauríem d’utilitzar l’`at`, mentre que per a operacions que vulguem fer sense que entorpeixin el funcionament normal de l’ordinador, hauríem d’utilitzar el `batch`. En els directoris `/var/spool/cron/atjobs/` i `/var/spool/cron/atspool/` es desen els fitxers corresponents a tots aquests treballs retardats.

Contingut complementari

Tot el sistema d’`at` i `batch` funciona amb uns mecanismes de cua d’execució (un per a cada un dels dos). Encara que en puguem configurar més, generalment només amb aquests ja en tenim prou per a fer qualsevol tipus d’execució retardada en el sistema.

Contingut complementari

La càrrega del sistema és un paràmetre que ens indica el grau d’activitat de l’ordinador. Amb el comandament `top` podem veure aquesta càrrega interactivament.

8. Instal·lació d'aplicacions

8.1. Introducció

La gestió i manipulació dels paquets és un aspecte fonamental en qualsevol distribució de GNU/Linux. Un paquet és un o diversos programes, llibreries o components de programari empaquetats en un sol arxiu preparat per a instal·lar-lo i integrar-lo al sistema operatiu. En el disseny de qualsevol distribució és molt important proporcionar les eines necessàries per a poder instal·lar i gestionar adequadament aquests paquets. També s'han de proporcionar eines, especialment per als desenvolupadors de programari, per a poder-ne crear d'altres de nous. En aquests paquets se solen incloure els executables del programa i les seves dependències i conflictes amb altres aplicacions. Les dependències indiquen, en instal·lar un paquet, si necessiten altres programes perquè l'aplicació funcioni correctament, mentre que els conflictes ens informen d'incompatibilitats entre programes instal·lats i el que volem instal·lar. Els sistemes de paquets estan dissenyats d'aquesta manera per a facilitar la instal·lació de les aplicacions noves, ja que algunes llibreries són utilitzades per més d'un programa i no tindria sentit que totes les aplicacions que les utilitzessin les instal·lessin de nou.

Actualment, la gran majoria de distribucions utilitzen un dels dos sistemes de paquets més estesos en el món del GNU/Linux: els deb o els rpm. D'una banda, els paquets deb són els que la distribució de Debian GNU/Linux utilitza en la seva distribució, mentre que els rpm (*redhat package manager*) són els nadius de RedHat. Les distribucions basades en alguna d'aquestes dues generalment adopten el sistema de paquets corresponent, encara que la majoria de les altres distribucions pròpies també han optat per incorporar algun dels dos sistemes, ja que actualment la gran majoria de programes s'empaqueten utilitzant aquests formats.

D'altra banda, els programes amb llicència GPL o similar també se solen distribuir amb el seu codi font (empaquetats i comprimits amb

Contingut complementari

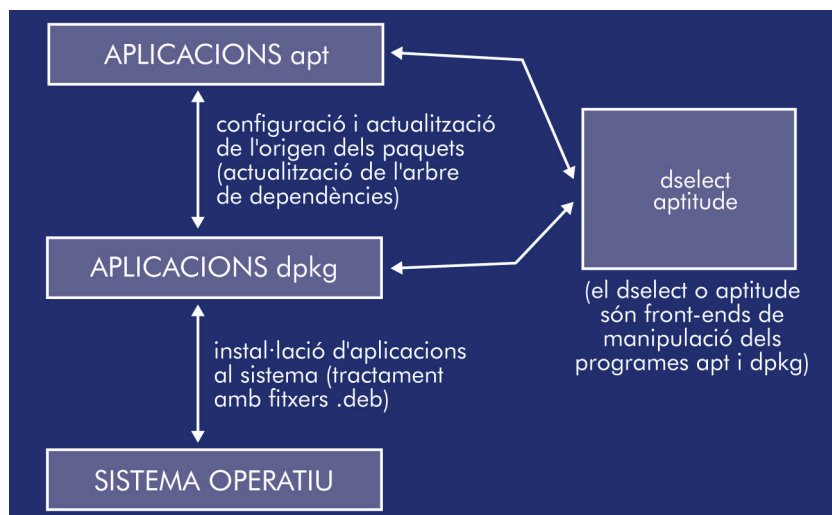
Debian GNU/Linux va ser la primera distribució que va crear un sistema de paquets.

algun format estàndard, com el tar). A partir d'aquest codi font, també podem instal·lar el programa en el nostre operatiu, compilant-lo i situant els executables al lloc on els correspongui.

En aquest capítol veurem com està organitzat el sistema de paquets de la distribució Debian per la gran quantitat d'eines que es proporcionen i la flexibilitat de la seva configuració. En l'última secció, aprendrem com instal·lar un programa a partir del seu codi font, ja que en alguns casos ens podem trobar que el programa que necessitem no estigui empaquetat. Aquest tipus d'instal·lació era el que s'utilitzava sempre abans que apareguessin els primers sistemes de paquets, que van sorgir per facilitar tot aquest procés.

8.2. El sistema de paquets Debian

Les aplicacions per a manipular el sistema de paquets de Debian GNU/Linux són, bàsicament, de dos tipus: els programes `apt` (*advanced packaging tool*) i els `dpkg` (*Debian package*). El conjunt d'aplicacions `apt` serveix per a configurar on aconseguim els paquets, quins són els que volem i resoldre dependències i conflictes amb d'altres. Els programes `dpkg` serveixen per a instal·lar els paquets, configurar-los, saber quins tenim instal·lats, etc. Hi ha altres aplicacions, com ara `dselect` o `aptitude`, que serveixen per a manipular els programes `apt` i `dpkg` proporcionant, en un sol entorn, eines interactives per a manipular-los. En la figura següent podem veure aquest esquema:



Els programes que en última instància s'encarreguen d'instal·lar les aplicacions són els `dpkg`. Aquests programes descomprimeixen el fitxer “.deb” i instal·len el programa. Les aplicacions `apt` ens ajuden a localitzar les últimes versions dels programes que necessitem, copien al disc els fitxers de les fonts d'on les hagin extret (FTP, CD-ROM, etc.) i comproven dependències i conflictes dels nous paquets perquè es puguin instal·lar correctament. Les principals aplicacions `apt` són les següents:

- `apt-config`: serveix per a configurar algunes de les opcions d'`apt` (l'arquitectura del nostre sistema, directori on es desen els arxius, etc.).
- `apt-setup`: aplicació per a configurar les fonts dels paquets (d'on els obtenim).
- `apt-cache`: gestió de la memòria de paquets (directori on es desen els arxius “.deb” abans d'instal·lar-los).
- `apt-CD-ROM`: aplicació per a gestionar CD-ROM que continguin paquets.
- `apt-get`: actualització, instal·lació o baixada dels paquets.

Tota la configuració d'`apt` és al directori `/etc/apt/`. En el fitxer `/etc/apt/sources.list` és on es desa la configuració de les fonts dels paquets. Amb totes aquestes fonts es genera una llista de paquets disponibles, que podem consultar i instal·lar sempre que ens interressi. Generalment, el format d'aquest arxiu segueix la sintaxi següent:

```
deb http://site.http.org/debian distribucio seccio1 seccio2 seccio3
deb-src http://site.http.org/debian distribucio seccio1 seccio2 seccio3
```

El primer camp de cada línia indica el tipus d'arxiu a què ens referim: binaris (`deb`) o codi font (`deb-src`). Seguidament trobem la referència de la font dels paquets, que pot ser un CD-ROM, una adreça d'Internet, etc. El camp de distribució indica a `apt` quina versió de Debian GNU/Linux utilitzem. Aquest camp és important perquè cada

Contingut complementari

Encara que amb les aplicacions `apt` també es puguin instal·lar paquets, l'únic que fan és cridar als programes `dpkg`.

Contingut complementari

Els sistemes de paquets també permeten crear paquets amb el codi font de les aplicacions. Si només ens interessa utilitzar l'aplicació, no fa falta que baixem els paquets de codi font.

versió de la distribució té els seus propis paquets. En els últims camps podem especificar quin tipus de paquets volem utilitzar.

Si canviéssim aquest fitxer manualment, podríem utilitzar el comandament `apt-get update` per a actualitzar tots els paquets disponibles en el sistema. Per a inserir els paquets d'un CD-ROM a la llista de paquets disponibles, podríem utilitzar `apt-CD-ROM add`, amb la qual cosa s'exploraria el CD inserit i s'actualitzaria la llista de paquets del sistema. Si algunes de les fonts continguessin paquets iguals, en instal·lar-lo la mateixa aplicació `apt` detectaria quin és més recent o quin triga menys temps a baixar-se i el baixaria de la font corresponent. Amb el programa `netselect`, a més, podríem configurar més àmpliament tot aquest sistema de baixada.

Una altra opció molt interessant que ens proporciona la majoria de distribucions és la de l'actualització de paquets en què s'hagi descobert algun tipus de vulnerabilitat o error en el seu funcionament. Amb Debian, tan sols hem d'afegir la línia següent en l'arxiu `/etc/apt/sources.list`:

```
deb http://security.debian.org/ stable/updates
main contrib non-free
```

A mesura que es detecten paquets crítics, es posen en aquesta font, de manera que només executant `apt-get update` s'avisava de les actualitzacions noves que hem de fer en el sistema i es reinstal·len els paquets necessaris.

Encara que amb els programes `dpkg` podem manipular qualsevol aspecte dels paquets instal·lats en el sistema, crear-ne de nous, modificar els instal·lats, etc., en aquest curs només repassarem els més importants des del punt de vista d'usuari, perquè hi puguem fer les operacions bàsiques. Els programes principals `dpkg` són els següents:

- `dpkg-divert`: ens serveix per a manipular el lloc d'instal·lació d'alguns dels paquets instal·lats en el sistema. Molt útil per a evitar alguns problemes de dependències.

- `dpkg-reconfigure`: amb un mateix paquet deb moltes vegades s'inclou algun mecanisme per a configurar algunes de les opcions de l'aplicació interactivament. Amb aquesta aplicació podem tornar a configurar el paquet que li indiquem amb els mateixos mecanismes utilitzats en la seva instal·lació.
- `dpkg-scanpackages`: aquest programa serveix per a escanejar un directori determinat del sistema que contingui arxius ".deb" perquè es generi un arxiu d'índex. Amb aquest arxiu d'índex podem incloure el directori com una font més d'apt. D'altra banda, és molt útil quan baixem programes no oficials de la distribució.
- `dpkg-scansource`: aplicació amb les mateixes funcionalitats que l'anterior però per a paquets de codi font.
- `dpkg-split`: programa per a dividir i unir un paquet a diversos arxius diferents.

Amb aquests programes podem manipular de qualsevol manera els nostres paquets. L'aplicació principal, `dpkg`, és la que ens permet instal·lar els paquets del sistema, fer-ne una llista o eliminar-los. Per a fer una llista de tots els paquets disponibles hi podem passar el paràmetre "-l", amb la qual cosa es mostrarà una llista completa dels paquets i el seu estat d'instal·lació (instal·lats, instal·lats però no configurats, etc.). Si volguéssim veure tota la informació d'un paquet determinat, podríem utilitzar el paràmetre "-p" seguit del nom del paquet, amb la qual cosa es mostren totes les dependències, conflictes amb altres paquets, versió, descripció, etc.

Per a instal·lar paquets nous podem utilitzar el paràmetre "-i" seguit del nom de l'arxiu. Si ens causa problemes de dependències, les podem ignorar amb "--ignoredepends=x", on la "x" indica la dependència, encara que hàgim de vigilar molt com utilitzem aquest paràmetre perquè en ignorar dependències és possible que el programa instal·lat no funcioni correctament. Si només volguéssim descomprimir l'arxiu ".deb" per veure què conté, també podríem utilitzar "-x". Per a eliminar els paquets, hem de passar "-r" seguit del nom del paquet, que l'elimina del sistema, però desant-ne els arxius de configuració (amb "-P" s'elimina tot).

Contingut complementari

Amb `dpkg` també podem utilitzar patrons per a seleccionar, instal·lar, eliminar, etc. els paquets del sistema.

Un altre paràmetre molt interessant és el de “`--force-things X`” (on la “`X`” és una de les opcions següents), que ens pot ajudar en algun dels casos que mostrem a continuació:

- “`auto-select`”: selecciona automàticament els paquets que s’han d’instal·lar o desinstal·lar amb el paquet nou que triem.
- “`downgrade`”: instal·la el paquet encara que n’hi hagi versions més noves.
- “`remove-essential`”: encara que el paquet es consideri essencial en el sistema, l’elimina.
- “`depends`”: no té en compte les dependències, les considera com si fossin alertes.
- “`depends-version`”: no té en compte dependències de versió dels paquets.
- “`conflicts`”: instal·la el paquet, encara que entri en conflicte amb algun altre del sistema.

Encara que tots els programes que hem anat comentat al llarg d’aquesta secció tenen moltes opcions i existeixen molts altres programes, amb els que hem especificat ja en tindrem prou, amb el sistema de paquets de la distribució que utilitzem, per a fer gairebé qualsevol tasca que sigui necessària. Si bé ja hem comentat que amb programes com el `dselect` o `aptitude` ja podem dur a terme les tasques bàsiques d’instal·lar i eliminar paquets, és important conèixer adequadament aquests altres comandaments perquè per a operacions específiques o per a automatitzar els processos de selecció i instal·lació poden ser molt útils.

8.3. Compilació de programes nous

En l’administració de qualsevol servidor és molt probable que en alguns casos ens trobem que hem d’utilitzar algun programa que la

nostra distribució no té o que necessitem l'última versió d'un servidor d'aplicacions que encara no està convenientment empaquetat, etc. En aquests casos, sempre ens podem baixar el codi font del programa i compilar-lo manualment. És important comprendre la diferència que hi ha entre compilar un programa per aconseguir el seu executable que baixar directament el binari. Quan compilem un programa, aquest utilitza les llibreries disponibles en el sistema, mentre que si el baixem directament, el més probable és que no funcioni adequadament perquè intentarà utilitzar alguna llibreria que no serà exactament igual que la que tinguem instal·lada en el sistema. És per això que, el més recomanable, quan necessitem instal·lar un nou programa del qual no disposem del paquet corresponent, és compilar-lo de nou.

En baixar les fonts d'una aplicació, ens trobarem amb un fitxer empaquetat i comprimit amb `tar` i `gzip` o similars. És usual afegir-hi un fitxer anomenat `README` en què s'explica pas a pas totes les accions necessàries per a compilar correctament el programa. Encara que sigui recomanable llegir-lo, en la majoria de casos, el procés d'instal·lació sempre és el mateix.

El primer que hem de fer per a compilar el nou programa és descomprimir-lo i desempaquetar-lo. Una vegada fet això, disposarem del codi font estructurat en diversos directoris. En la seva arrel, podem trobar (o no) un fitxer anomenat `Makefile`. Aquest arxiu indica al compilador quines llibreries s'utilitzen, com s'han de compilar els arxius de codi, etc. Si tenim aquest `Makefile`, ja podem compilar el programa executant `make`. No fa falta que li passem cap paràmetre perquè per defecte ja busca el fitxer de `Makefile` i executa les accions que s'hi especifiquen. Si el procés no ha donat cap error, ja podrem moure l'executable generat per a posar-lo en algun dels directoris del `PATH` configurat; d'aquesta manera, sempre que el vulguem executar no haurem d'escriure la seva ruta completa. Molts `Makefile` proporcionen, així mateix, instruccions perquè ens puguem estalviar aquest últim pas. Generalment, executant `make install` el mateix programa s'encarrega de situar adequadament els binaris i, si n'hi hagués, els arxius de documentació. Finalment, si no ens interessés desfer el codi font del programa, ja podríem eliminar tot el contingut dels directoris creats.

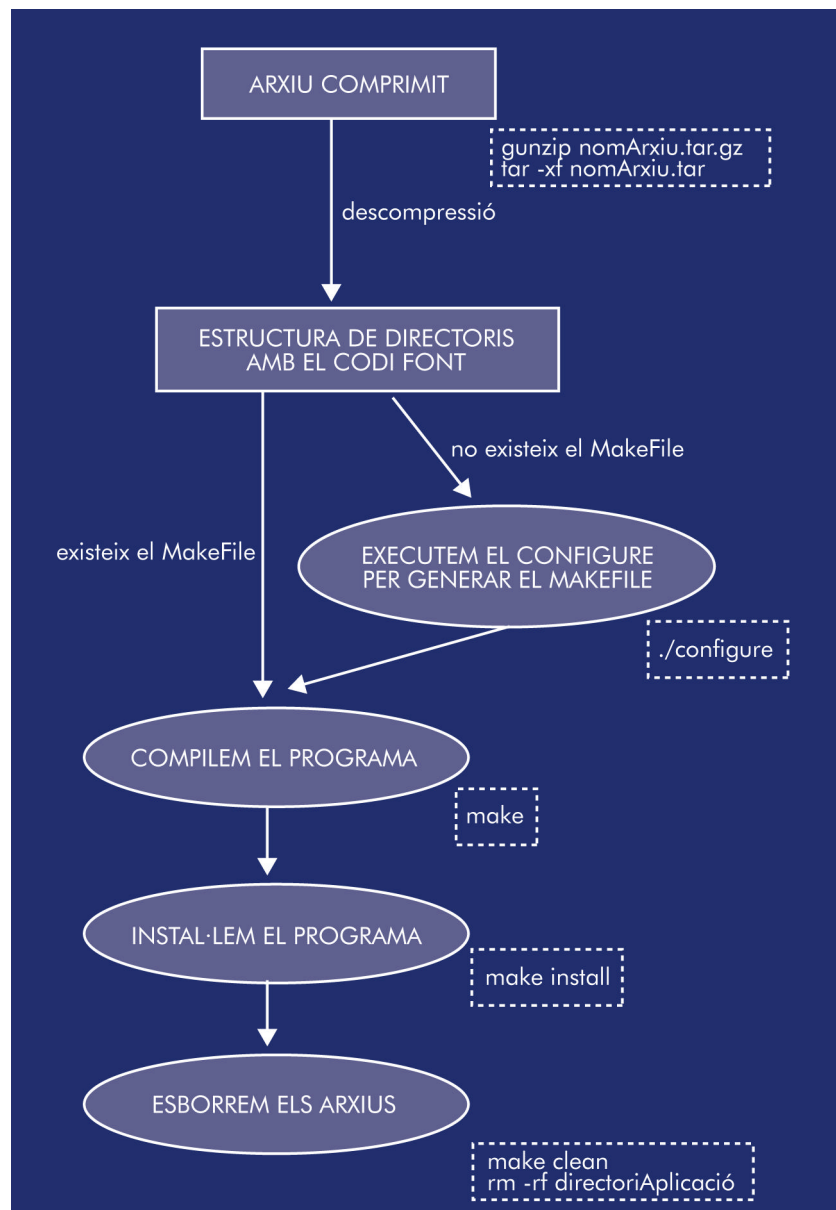
Contingut complementari

En compilar un programa és molt probable que necessitem tenir instal·lades en el sistema les fonts o les capçaleres de les llibreries que utilitza. Generalment, aquests paquets solen tenir el mateix nom que la llibreria, però afegint `-dev` (de desenvolupament) al final.

Contingut complementari

El procés de compilació d'un programa pot durar des de segons fins a hores, segons l'aplicació. Per exemple, la compilació del nucli Linux va des de cinc o deu minuts fins a les dues hores (segons la versió del nucli i la potència de l'ordinador).

Si el programa no incorpora l'arxiu de Makefile, generalment se sol incloure algun *shell script* per a generar automàticament aquest fitxer (habitualment, aquest *script* se sol anomenar *configure*). En executar aquest *shell script* es comprovarà que el sistema tingui instal·lades totes les llibreries necessàries per a una bona compilació i, si en faltés alguna, es donaria un missatge d'avís. Una vegada executat correctament aquest *shell script*, ja disposarem del *Makefile*, amb la qual cosa el procés torna a ser el mateix que anteriorment. En la figura següent podem veure tot això gràficament:



Encara que la majoria del codi font dels programes s'organitzi de la manera com hem exposat, també és possible que ens trobem amb

altres tipus d'instal·lació. N'hi ha algun que incorpora menús, entorns en X o altres mètodes més amens. Últimament, també comencen a aparèixer algunes aplicacions que permeten fer tot el procés d'instal·lació baixant el codi directament d'Internet.

9. Taller de configuracions bàsiques

9.1. Introducció

En el segon taller hem après a instal·lar un sistema bàsic perquè, a partir d'aquest, puguem començar a muntar un sistema a la mida de les nostres necessitats. Aquest serà l'objecte d'aquests dos últims tallers. En el primer, després de concretar alguns aspectes referents a la configuració del sistema d'instal·lació de paquets, quedaran fixades les bases per poder instal·lar les aplicacions necessàries. A continuació, aprendrem a utilitzar les diferents eines que ens ofereix Debian per a la gestió de paquets –és a dir, per a instal·lar, desinstal·lar, actualitzar, etc. aplicacions–, i n'instal·larem i en configurarem algunes, que són comunes i necessàries en la majoria de sistemes tipus UNIX. L'últim taller està plenament orientat al sistema gràfic. Hi aprendrem a instal·lar-lo, configurar-lo, adaptar-lo a les nostres preferències i, finalment, aprendrem a instal·lar i utilitzar aplicacions que se serveixen d'aquest sistema per a poder córrer.

9.2. El gestor d'arrencada

En primer lloc ens hem d'assegurar d'instal·lar un sistema d'arrencada que sigui capaç de gestionar sense problemes els possibles sistemes operatius que tinguem instal·lats al nostre ordinador. Aquesta qüestió ja es va abordar superficialment en el taller anterior, durant el procés d'instal·lació. Si ja tenim instal·lat el gestor d'arrencada i n'hem comprovat el funcionament correcte, podem passar a la secció següent; però si no és així, és a dir, si per a engegar el nostre sistema operatiu necessitem el disquet d'arrencada que hem creat durant el procés d'instal·lació, ha arribat el moment d'instal·lar un sistema de gestió d'arrencada al disc dur per a evitar la farragosa tasca d'utilitzar cada vegada el disquet. De qualsevol manera, sempre és interessant tenir instal·lat un sistema gestor d'arrencada, ja que ens permetrà, entre altres coses, poder engegar diversos nuclis.

Nota

MBR és l'acrònim de Master Boot Record. Aquesta és l'àrea del disc dur on s'emmagatzemen les instruccions bàsiques perquè l'ordinador pugui carregar el sistema operatiu.

Tal com s'ha exposat, hi ha diferents gestors d'arrencada, entre els quals es pot esmentar LILO i GRUB. GRUB és un poderós sistema gestor d'arrencada, del projecte GNU, que es caracteritza per poder gestionar correctament l'arrencada de pràcticament qualsevol sistema operatiu que tinguem instal·lat al nostre ordinador. No obstant això, el seu ús i la seva configuració són una mica complexos. LILO és el gestor que es va dissenyar inicialment per a gestionar les arrencades dels nuclis Linux, es caracteritza per tenir un sistema d'instal·lació més intuïtiu, a més d'oferir la possibilitat, mitjançant una sola línia de comandaments, de reestablir de manera senzilla la imatge de l'MBR anterior si alguna cosa ha anat malament; aspecte molt interessant, sobretot si al nostre disc dur conviuen diferents sistemes operatius, ja que escriure en l'MBR pot inutilitzar-ne algun, especialment si es tracta de sistemes Microsoft™.

De qualsevol manera, si no disposem de gestor d'arrencada i volem instal·lar-ne un, el primer que hem de fer és engegar el sistema operatiu que tenim i comprovarem si l'aplicació està instal·lada, fent, per exemple, un `man` de l'aplicació.

9.2.1. Instal·lació de LILO

L'arxiu de configuració de LILO es troba a `/etc/lilo.conf`. L'arxiu està ben documentat i, per tant, és fàcil modificar-lo mitjançant un editor de text i adaptar-lo a les nostres necessitats. Una vegada hàgim fet les modificacions que considerem oportunes, el que farem és executar com a `root` la instrucció `lilo` per a transferir a l'MBR el contingut del fitxer de configuració.

Com a mesura de precaució, el que farem immediatament després és reiniciar l'ordinador, per exemple mitjançant el comandament `reboot`, i comprovarem que tots els sistemes operatius que tinguem instal·lats arrenquin correctament. Si no fos així, el que s'ha de fer és tornar a arrencar el nostre sistema operatiu de nou instal·lat (mitjançant LILO, si aquest és capaç de fer-ho, o mitjançant el disquet d'arrencada –el que hem utilitzat fins al moment per a arrencar el sistema–) i executar la instrucció següent per reestablir la còpia de seguretat de l'MBR que LILO ha fet abans d'escriure so-

bre ella; amb això estarem en la mateixa situació que érem abans d'executar `lilo`:

```
~# lilo -U
```

Una vegada fet això, hem de procedir a estudiar el perquè dels errors i tractar de corregir-los; ha arribat el moment de començar a llegir documentació, començant pels mans, les PMF o FAQ, la documentació que podem trobar a `/usr/share/doc/lilo/` etc. Després de fer els canvis pertinents, repetirem el procés anterior.

Pot succeir que després de diversos intents arribem a la conclusió que o LILO no pot o no el sabem configurar correctament per a engegar tots els sistemes operatius que tenim instal·lats a l'ordinador. Ha arribat el moment de provar amb altres gestors, com pot ser GRUB.

9.2.2. Instal·lació de GRUB

GRUB no ve instal·lat per defecte; per aquest motiu, abans d'usarlo hem d'instal·lar-lo. Per a això, podem procedir a llegir les seccions posteriors per a informar-nos sobre el sistema gestor de paquets, la seva configuració i el seu ús; o simplement, en cas que durant la instal·lació hàgim introduït en la base de dades els continguts de tots els CD o que hàgim optat per una instal·lació per xarxa, executarem la línia següent:

```
~# apt-get install grub grub-doc
```

Si no volem instal·lar el paquet de documentació, es pot ometre l'últim paràmetre de la línia anterior.

Una manera recomanable de treballar amb GRUB és fer, abans d'escriure a l'MRB, les proves que s'estimin oportunes sobre un disquet i provar-ne el funcionament correcte arrencant des d'aquest. El primer que farem és copiar els fitxers necessaris a:

```
~# cp /usr/lib/grub/i386-pc/stage* /boot/  
~# cp /usr/share/doc/grub/examples/menu.lst /boot/
```

Una vegada fet això, editarem el fitxer de configuració de GRUB, `/boot/menu.lst`, i l'adaptarem a les nostres necessitats. Aquesta operació pot ser una mica complexa, a causa, entre altres coses, del canvi de nomenclatura pel que fa als discos durs; per tant per a identificar quin dispositiu es troba en un fitxer en concret, ens podem ajudar del comandament `find` una vegada hàgim entrat en el mode de comandaments de GRUB mitjançant `grub`. Així doncs, per exemple, per a localitzar un fitxer, la ubicació del qual segur que necessitarem especificar en el fitxer de configuració de GRUB, com pugui ser `/vmlinuz`, procedirem de la manera següent:

```
~# grub

      Grub version 0.91 (640K lower / 3072K upper memory)

[ Minimal BASH-like line editing is supported. For the
  first word, TAB lists possible command completions.
  Anywhere else TAB lists the possible completions
  of a device/filename. ]

grub> find /vmlinuz
      (hd2,0)
```

GRUB té molts més comandaments; per a veure una llista d'alguns n'hi ha prou de prémer la tecla TAB en la línia de comandaments per obtenir el següent:

```
grub>
      Possible commands are: blocklist boot cat chainloader
      cmp color configfile deb ug device displayapm displaymem
      embed find fstest geometry halt help hide impsp robe initrd
      install ioprobe kernel lock makeactive map md5crypt module
      moduleno unzip partnew parttype password pause quit read
      reboot root rootnoverify savede fault serial setkey setup
      terminal testload testvbe unhide uppermem vbeprobe

grub>
```

Per a obtenir ajuda d'algun comandament en concret, n'hi ha prou de teclejar `help` seguit del nom del comandament que busquem. Una bona manera de treballar sobre el nostre fitxer de configuració de GRUB `/boot/menu.lst` és obrint una nova sessió en una `tty` diferent i modificar aquest fitxer a mesura que treballem sobre la interfície de comandaments de GRUB.

Una vegada hàgim acabat d'editar el fitxer de configuració, i després d'introduir un disquet verge a la unitat de disquets, teclejarem –substituint el caràcter “X” pel número de disc dur, i “Y” per la partició corresponent– la línia de comandaments següent:

```
GRUB> install (hdX,Y)/boot/stage1 d (fd0) (hdX,Y)/boot/stage2 p (hdX,Y)/boot/menu.lst
```

Amb la línia anterior hem transferit al disquet (fd0) la informació de l'MBR (stage1), GRUB i la seva interfície de comandaments (stage2), i el menú d'arrencada que hàgim configurat en el fitxer /boot/menu.lst. Estem, doncs, en condicions de reiniciar la màquina, arrencar mitjançant el disquet que acabem de crear i comprovar si la nostra configuració és correcta.

És molt interessant disposar d'un disquet amb GRUB, ja que, mitjançant la seva interfície de comandaments, podem intentar engegar directament els diferents sistemes operatius que tinguem instal·lats, per tal de fer proves per a depurar el contingut del fitxer /boot/menu.lst. A tall d'exemple, s'exposa el procediment que cal seguir per a arrencar un sistema Microsoft™ instal·lat com a (hd0,0) i a continuació els comandaments necessaris per a arrencar un sistema GNU/Linux instal·lat com a (hd1,0):

Per a un sistema Microsoft™:

```
grub> rootnoverify (hd0,0)
grub> makeactive
grub> chainloader +1
grub> boot
```

Per a un sistema GNU/Linux:

```
grub> kernel (hd1,0) /vmlinuz root=/dev/hdc1
grub> boot
```

Una vegada tinguem al nostre disquet el sistema d'arrencada definitiva que vulguem implementar, simplement hem de transferir aquesta mateixa configuració a l'MBR del disc dur d'arrencada. Des de la mateixa línia de comandaments del disquet teclejarem:

```
GRUB> install (hdX,Y)/boot/stage1 d (hd0,0) (hdX,Y)/boot/stage2 p (hdX,Y)/boot/menu.lst
```

Com s'ha dit, utilitzar GRUB és una mica més complex que utilitzar LILO; per aquesta raó es recomana que abans d'embarcar-se en la seva instal·lació i posterior configuració, es llegeixi detingudament els man i la documentació que s'hagi instal·lat amb el paquet grub-doc. També és accessible a <http://www.gnu.org/software/grub/>.

9.3. El sistema de paquets

Ha arribat el moment d'analitzar i aprendre a utilitzar el sistema de paquets de Debian, probablement el més sòlid i fiable dels que hi ha en el món GNU.

En les subseccions següents aprendrem a configurar la seva base de dades, a manipular-la, a instal·lar paquets, actualitzar-los, etc. Moltes vegades hi ha diferents maneres d'obtenir el mateix resultat. Se n'exposaran algunes, amb dos objectius principals: el primer, perquè el lector pugui optar per la que més li interessi, i el segon, perquè és interessant conèixer sempre més d'una solució a un mateix problema, per si de cas alguna fallés.

Activitat

13. Per a comprendre plenament el funcionament del sistema de paquets Debian, es recomana la lectura de:

- APT HOWTO: <http://www.de.debian.org/doc/manuals/apt-howto/index.en.html>
- http://www.de.debian.org/doc/manuals/debian-faq/ch-pkg_basics.en.html
- <http://www.de.debian.org/doc/manuals/debian-faq/ch-pkgtools.en.html>
- Els man següents: `apt`, `apt-cache`, `apt-get`, `sources.list`, `dpkg` i `dselect`.

Nota: per a fer aquestes lectures més assequibles, podem esperar a llegir-les quan tinguem configurada la impressora i hàgim après a imprimir manuals.

9.3.1. [/etc/apt/sources.list](#)

L'arxiu `/etc/apt/sources.list` és el cor de la configuració del sistema de paquets de Debian. Com que es tracta d'un fitxer de text, com la majoria de fitxers de configuració en els sistemes UNIX, el podem editar manualment, o bé mitjançant algunes eines de les que disposa el sistema per a aquesta finalitat.

El contingut d'aquest fitxer dependrà en gran mesura de la velocitat amb què puguem accedir a Internet, si és que ho podem fer. Però en cap cas no hem d'oblidar executar la instrucció següent com a *root*, una vegada hàgim modificat el fitxer:

```
/etc/apt# apt-get update
```

Si no ho féssim, la base de dades del sistema de paquets no s'actualitzaria i, en conseqüència, cap dels canvis fets no produiria efecte. En una instal·lació on els paquets s'obtinguin remotament, aquest comandament ha de ser executat periòdicament per actualitzar la base de dades; per aquest motiu, no és mala idea incloure'l dins del sistema cron.

Cada línia d'aquest fitxer fa referència a una font de paquets i els camps van separats per un espai. En primer lloc especificarem si la font de paquets és de paquets binaris `deb` o si és de codi font `deb-src`. En el segon camp especificarem la manera d'accedir-hi: `CD-ROM`, `http`, `ftp`, etc. seguit de la l'adreça d'accés. Els camps restants fan referència al tipus de paquets a què volem accedir per aquesta línia.

`/etc/apt/sources.list` amb accés ràpid a Internet

En el millor dels casos disposarem d'un accés ràpid a Internet. Això, probablement, ja ens haurà permès fer la instal·lació del sistema bàsic per xarxa, a més de disposar sempre de les últimes versions dels paquets. Es tracta, a més, de la manera més còmoda de treballar amb paquets, ja que només ens hem de preocupar d'inserir el CD corresponent per fer una instal·lació.

Hem d'anar en compte, abans que res, que el contingut de `/etc/apt/sources.list` sigui correcte. A continuació se'n mostra un exemple:

```
deb http://ftp2.de.debian.org/debian/ stable main non-free contrib
deb-src http://ftp2.de.debian.org/debian/ stable main non-free contrib
deb http://security.debian.org/ stable/updates main contrib non-free
```

Totes les adreces anteriors són oficials; és a dir, reconegudes per Debian. A part d'aquestes fonts, també es poden utilitzar paquets no oficials, que encara que no siguin oficials no vol dir que no tinguin la qualitat suficient per a incloure'ls al nostre fitxer de configuració. Una bona adreça per a obtenir informació sobre la localització d'aquests paquets és <http://www.apt-get.org>.

Després d'editar i desar el fitxer, executarem el comandament `apt-get update` i després d'uns instants, en els quals el sistema de paquets reconfigurarà la seva base de dades i visualitzarà per pantalla els diferents accessos que es fan, ja podrem accedir als paquets nous.

`/etc/apt/sources.list` sense accés ràpid a Internet

En el cas de no disposar de connexió a Internet o de ser lenta, hem d'optar, sens dubte, per utilitzar el joc de CD de la distribució per a instal·lar els diferents paquets. Si durant el procés d'instal·lació no hem inserit tots els CD, haurà arribat el moment de fer-ho. Inserim el primer CD-ROM al lector i usem el comandament `apt-CD-ROM` per a incloure'n els continguts en la base de dades:

```
/etc/apt# apt-CD-ROM add
Using CD-ROM mount point /CD-ROM/
```

```
.
.
.
```

Repeat this process for the rest of the CDs in your set.

Llavors, repetirem el mateix procés per a tots i cada un de nostres CD de la distribució. Així mateix, es pot utilitzar el mateix procediment per a incorporar dades procedents de CD-ROM no oficials.

Una vegada tinguem configurat el nostre accés a Internet, si ho estimem oportú, podem incloure fonts de paquets d'accés remot. Per a això editarem el fitxer `/etc/apt/sources.list` i després d'executar `"apt-get update"`, tindrem disponibles els paquets nous.

9.3.2. [apt](#)

apt és l'acrònim d'*advanced package tool*, que, com ja s'ha dit diverses vegades, és el sistema bàsic encarregat de l'administració de paquets de les distribucions basades en Debian. *apt* posa a la nostra disposició essencialment dues eines: `apt-get` i `apt-cache`. El primer comandament el pot utilitzar únicament i exclusivament el `root` del sistema, ja que és l'eina de gestió de paquets: instal·lació, desinstal·lació, actualització, etc., mentre que el segon, com que és un comandament orientat a la recerca d'informació dins de la base de dades, tant si són paquets instal·lats com sense instal·lar, el pot utilitzar qualsevol usuari.

Per tal de facilitar el maneig de paquets, s'han desenvolupat altres aplicacions que corren per sobre d'*apt*, com pot ser el *middle-end* `dpkg` o els front-end `dselect` o `aptitude`. Però abans d'endinsar-nos en els diferents sistemes d'administració de paquets, n'hem de conèixer alguns conceptes i la seva relació amb el sistema i el sistema de gestió.

Tipus de paquets segons la seva prioritat

Dins del sistema de paquets se'n distingeixen cinc tipus diferents segons el seu grau de dependència amb el sistema mateix. Per ordre decreixent de prioritat es classifiquen com a:

Required. Es tracta de paquets indispensables per al funcionament correcte del sistema mateix.

Important. Es tracta de paquets que haurien de ser presents en qualsevol sistema tipus UNIX.

Standard. Es tracta de paquets que comunament es troben en un sistema GNU/Linux. En general són aplicacions de mida reduïda, però que ja no són indispensables per al sistema.

Optional. Es tracta de paquets que poden ser presents en un sistema GNU/Linux. Entre d'altres, dins d'aquest grup es troben tots els paquets referents al sistema gràfic, ja que aquest no es considera indispensable; en realitat, en molts servidors, per tal d'augmentar-ne el rendiment es prescindeix de l'entorn gràfic.

Extra. Són paquets que, o bé presenten conflictes amb paquets amb prioritat superior a la seva o bé requereixen configuracions especials que no els fan aptes per a integrar-los com a *optional*.

Podem determinar a quin grup pertany un paquet en concret mitjançant, per exemple, la sentència `apt-cache show <nom del paquet>` i consultar el contingut del camp `Priority`.

Grau de dependència entre paquets

apt es caracteritza per la seva gran consistència a l'hora de gestionar les dependències que hi ha entre paquets. Per exemple, pot ser que una aplicació determinada que vulguem instal·lar depengui d'una llibreria que no té el sistema, en conseqüència, d'un altre paquet que no tinguem instal·lat. En aquest cas, apt ens informarà d'aquesta dependència i ens preguntarà si volem que, juntament amb l'aplicació que instal·larem, s'instal·li el paquet que conté aquesta llibreria. Les relacions entre paquets es classifiquen de la manera següent:

depends. El paquet que volem instal·lar depèn d'aquests paquets i, per tant, si volem que aquest paquet funcioni correctament, hem de permetre que apt n'instal·li la resta.

recommends. El responsable del manteniment del paquet ha estimat que normalment els usuaris que vulguin instal·lar aquest paquet també usaran els que recomana.

suggests. Els paquets que se suggereixen permeten obtenir un rendiment més gran del paquet que instal·larem.

conflicts. El paquet que instal·larem no funcionarà correctament si aquests altres paquets són presents en el sistema.

replaces. La instal·lació d'aquest paquet implica la desinstal·lació d'altres paquets.

provides. El paquet que instal·larem incorpora tot el contingut d'aquests paquets.

Podem determinar les dependències d'un paquet, per exemple, la sentència `apt-cache depends <nom del paquet>`.

Accions sobre els paquets

Mitjançant els *flags* següents, `dpkg` o `dselect` ens informaran sobre el que l'usuari pretén fer amb aquests paquets:

unknown. Mai no s'ha fet referència a aquest paquet.

install. Es vol instal·lar o actualitzar el paquet.

remove. Es vol desinstal·lar el paquet, però mantenint-ne els fitxers de configuració (comunament situats a `/etc/`).

purge. Es vol desinstal·lar per complet el paquet, fins i tot els seus fitxers de configuració.

hold. Es vol indicar que no es vol fer cap operació sobre aquest paquet (és a dir, que se'n mantingui la versió i la configuració fins que s'avisi del contrari).

Estat d'instal·lació dels paquets

Dins del sistema un paquet es pot trobar:

installed. El paquet ha estat instal·lat i ha estat configurat correctament.

not-installed. El paquet no està instal·lat en el sistema.

unpacked. El paquet ha estat desempaquetat, però no ha estat configurat.

half-installed. El paquet ha estat desempaquetat i s'ha iniciat el procés de configuració, però per alguna raó aquest no ha acabat.

config-files. Només hi ha, en el sistema, els arxius de configuració d'aquest paquet.

apt-cache

Com ja s'ha dit, `apt-cache` és un comandament orientat a analitzar el sistema de paquets i, per tant, com que no és una arma potencialment perillosa per al sistema, és accessible a tots els seus usuaris. Els paràmetres més utilitzats per a aquest comandament són els següents:

search pattern. Busca en la base de dades els paquets que continguin *pattern* en el nom o que en la seva descripció aparegui *pattern* (si el resultat és una llista extensa a causa que *pattern* és molt general, es poden utilitzar *pipes* i `grep` per a filtrar aquests resultats).

show package. Informa sobre el paquet.

policy package. Informa sobre l'estat d'instal·lació, la versió i revisió del paquet, i la seva procedència.

depends package. Explicita les dependències del paquet.

show package. Mostra les dependències directes i indirectes del paquet.

apt-get

apt-get és el comandament que s'utilitza per a gestionar els paquets del sistema. Per aquest motiu el seu ús està restringit al `root` del sis-

tema. Els paràmetres més utilitzats per a aquest comandament són els següents:

install package. Instal·la el paquet. Si aquest depèn de paquets que no es troben en el sistema, apt ens n'informarà, i ens preguntarà si juntament amb el paquet en qüestió volem instal·lar els paquets de què depèn i que no estan instal·lats; en general és interessant seguir els consells d'apt.

update. Actualitza la base de dades d'apt. Aquest comandament s'ha d'executar cada vegada que es modifiqui l'arxiu `/etc/apt/sources.list`.

upgrade. Força l'actualització de tots els paquets instal·lats en el sistema en l'última versió disponible.

remove package. Elimina el paquet, sense eliminar els fitxers de configuració, amb vista a possibles reinstal·lacions.

remove -purge package. Elimina per complet el paquet, incloent-hi els seus arxius de configuració.

clean. Elimina les còpies caducades dels paquets que s'han instal·lat, procés en el qual s'emmagatzema de manera automàtica una còpia del paquet sense desempaquetar a `/var/cache/apt/archives` quan s'instal·la un paquet. Comandament molt útil amb vista a alliberar espai del disc dur, ocupat per fitxers que, probablement, mai més no s'utilitzaran.

autoclean. Elimina totes les còpies no desempaquetades dels paquets, independentment de la seva vigència.

9.3.3. dpkg

`dpkg` és un acrònim de *Debian package manager* i va ser concebut com a *back-end* d'apt. Els paràmetres més utilitzats són els següents:

- l. Per a fer una llista de tots els paquets de la base de dades i el seu estat d'instal·lació (generalment aquesta opció es combina amb `grep`).

- **L package**. Per a fer una llista dels fitxers continguts al paquet.
- **r package**. Té el mateix efecte que `apt-get remove package`.
- **P package**. Té el mateix efecte que `apt-get remove -- purge package`.
- **p package**. Té el mateix efecte que `apt-get show package`.
- **s package**. Descriu l'estat d'instal·lació del paquet.
- **S file**. Cerca a quins paquets pertany el fitxer.

9.3.4. dselect

`dselect` és una GUI (*graphical user interface*) que corre sobre `apt`. Per a entrar-hi, n'hi ha prou de teclejar el comandament `dselect`, i mitjançant els menús d'aquesta interfície seleccionar els diferents paquets sobre els quals volem operar i especificar quin tipus d'operació hi volem fer.

9.3.5. aptitude

`aptitude` és una altra GUI que corre sobre `apt`. Per defecte no ve instal·lada, per la qual cosa cal fer-ho abans de procedir a utilitzar-la:

```
/etc/apt# apt-get install aptitude
```

Una vegada instal·lada, la llancem mitjançant el comandament `aptitude`, i de seguida veurem que el seu ús és igual o més simple que el de `dselect`, ja que disposa de menús desplegable accessibles mitjançant F10.

9.4. Configuració regional: locals

Encara que aparentment el nostre teclat funcioni correctament, ja que podem utilitzar accents, dièresi i altres caràcters no anglesos, a mesura que adaptem el sistema a les nostres necessitats, i especial-

ment quan instal·lem el sistema gràfic i hi fem córrer aplicacions, en el pròxim taller ens adonarem que això no és així. Podem, doncs, en aquest punt procedir a configurar correctament aquests aspectes per no haver de fer-ho més endavant.

En primer lloc comprovarem si el paquet `locales` està instal·lat:

```
/# dpkg -l | grep locales
ii locales          2.2.5-11.2          GNU C Library: National Language (locale) da
```

Si no obtenim l'última línia, hem de procedir a instal·lar el paquet i configurar-lo:

```
/# apt-get install locales
```

I si ja en disposem, teclejarem la línia següent per a reconfigurar-lo:

```
/# dpkg-reconfigure locales
```

De les moltes opcions que se'ns ofereixen, triem `[*] ca_ES ISO-8859-1`, és a dir, ens situem sobre aquesta opció i polsem la barra espaiadora. Mitjançant `TAB`, ens situem sobre `OK` i premem `INTRO`. A la pròxima pantalla seleccionem `C`.

De tornada a la línia de comandaments, editem el fitxer `/etc/environment` per deixar-lo de la manera següent:

```
LC_ALL=ca_ES
LANGUAGE=en_US
LC_TYPE=ca_ES
LC_MESSAGES=ISO8859-1

LANG=C
```

Per a fer efectiu el canvi, n'hi ha prou de teclejar el comandament `locale-gen`, i sortirem de totes les sessions que tinguem obertes per a carregar la nova configuració.

Lectura complementària

Per a saber més sobre `locales`, es recomana visitar la pàgina:

<http://www.uniulm.de/~s-smasch/locale/>.

9.5. Configuració de *man* i el paginador

Pot ser que en intentar invocar `man` ens doni un error del tipus:

```
Reformatting man(1), please wait...
sh: /usr/bin/pager: No such file or directory
sh: exec: /usr/bin/pager: cannot execute: No such file or directory
man: command exited with status 32256: /usr/bin/zsoelim /tmp/zmanZpjojo |
      /usr/bin/tbl | /usr/bin/nroff -mandoc -Tlatin1 | exec /usr/bin/pager -s
```

En aquest cas, no tenim cap paginador assignat a `/usr/bin/pager`, de manera que `man` el pugui utilitzar per a mostrar-nos els continguts d'ajuda. És més, probablement no tindrem cap paginador instal·lat. En ambdós casos, el que farem és intentar instal·lar `less`, segurament el paginador més utilitzat per a `man`:

```
/# apt-get install less
```

En el cas que el paquet no estigui instal·lat, se'ns obrirà una pantalla de configuració, a la qual en principi, contestarem negativament, opció per defecte. Amb això ja hauréu instal·lat `less`, i probablement ja s'hagi assignat ell mateix com a paginador de `man`. Si no fos així (el podríem provar executant un `man man`, per exemple), o si `less` ja estigués instal·lat, utilitzaríem el comandament següent per a assignar-lo com a paginador (també es pot utilitzar per a canviar de paginador, per exemple, per a canviar a `more`, `jless`, o qualsevol altre paginador):

```
/# update-alternatives --config pager
```

El que fa el comandament anterior, en el cas del paginador, i en general amb la resta d'aspectes configurables mitjançant aquell, és crear enllaços simbòlics. Per al cas del paginador són aquests:

```
/etc/alternatives/pager -> /usr/bin/less
/usr/bin/pager -> /etc/alternatives/pager
```

9.6. L'arxiu principal d'arrencada, /etc/inittab

Encara que el procés d'arrencada d'un sistema GNU/Linux sigui complex, en aquesta secció només es pretén treballar sobre un dels fitxers principals d'aquest procés: `/etc/inittab`. Aquest arxiu indica al procés d'arrencada, entre d'altres, a quin *runlevel* s'entrarà, i definirà quins processos arrencaran automàticament durant el procés d'arrencada. Per a saber en quin *runlevel* ens trobem, n'hi ha prou de teclejar el comandament `runlevel`. Per a canviar de *runlevel*, com a *root*, usarem la instrucció `init <runlevel de destinació>`.

És interessant obrir aquest fitxer i familiaritzar-se amb el seu contingut, ja que això ens permetrà comprendre millor el procés d'arrencada d'un sistema GNU/Linux.

9.7. Muntatge de dispositius, /etc/fstab

`/etc/fstab` és el fitxer que conté la informació sobre les particions i els dispositius que es muntaran automàticament durant el procés d'arrencada, i les que es poden muntar posteriorment, i també estableix qui pot fer-ho. A continuació, es mostra el possible contingut d'aquest fitxer i es passa a analitzar-lo:

```
# /etc/fstab: static file system information.
#
# <file system>      <mount point>      <type>      <options>      <dump>      <pass>
/dev/hdg1            /                   ext3         errors=remount-ro 0           1
/dev/hdg2            none                swap         sw              0           0
proc                 /proc              proc         defaults        0           0
/dev/fd0             /floppy            auto         user,noauto     0           0
/dev/hdg5            /usr               ext3         defaults        0           2
/dev/hdg6            /var               ext3         defaults        0           2
/dev/hdg7            /home              ext3         defaults        0           2

/dev/CD-ROM /CD-ROM iso9660 ro,user,noauto 0 0

/dev/hdc1 /mnt/hdc1 ntfs ro,user,noauto,gid=windows,umask=0007,utf8 0 0
/dev/hde1 /mnt/hde1 ntfs ro,user,noauto,gid=windows,umask=0007,utf8 0 0
/dev/hde5 /mnt/hde5 ntfs ro,user,noauto,gid=windows,umask=0007,utf8 0 0

/dev/hde6 /mnt/hde6 vfat utf8,user,noauto 0 0
/dev/hde7 /mnt/hde7 vfat utf8,user,noauto 0 0

/dev/sda1 /mnt/sda vfat utf8,sync,user,noauto 0 0
```

Les primeres línies les ha generat automàticament el procés d'instal·lació i hi podem veure com estan distribuïts els diferents directoris dins de l'estructura purament GNU/Linux. Potser la línia que més crida l'atenció sigui la `proc /proc proc defaults 0 0`; aquesta és l'encarregada del muntatge del directori virtual `proc`, del qual ja es va parlar en el primer taller.

Són més interessants les línies tipus `/dev/hdc1 /mnt/hdc1 ntfs utf8,ro,noauto,user,gid=windows,umask=0007,utf8 0 0`. S'hi especifica el punt d'origen i el punt de muntatge de particions pertanyents al sistema operatiu Windows2000™, és a dir, de tipus `ntfs`. En aquestes particions no es pot escriure des de GNU/Linux, encara que sí que se'n pot llegir el contingut, cosa que es veu reflectit en les opcions `ro,noauto,user,gid=windows,umask=0007,utf8` (és fonamental no deixar cap espai en blanc entre opcions, ja que aquest caràcter és el que s'utilitza per a separar els camps en aquest fitxer). La primera indica que es tracta d'una partició de només lectura; la segona, que no es munta automàticament durant el procés d'arrencada del sistema; la tercera indica que aquesta partició la pot muntar qualsevol usuari; la quarta opció indica que només hi podran accedir els membres pertanyents al grup `windows` (definit en el fitxer `/etc/group`); la penúltima opció estableix l'antimàscara de muntatge, i l'última, la taula de codis que s'ha d'utilitzar.

Cal tenir present que per defecte el nucli que hem instal·lat no suporta el tipus `ntfs`. Per tant, cal carregar el mòdul corresponent mitjançant el comandament `modconf` i seleccionar l'opció `kernel/fs/ntfs`.

Les últimes línies del fitxer anterior van destinades a muntar particions `fat32`, sobre les quals sí que és possible escriure des de GNU/Linux. Per aquesta raó és bona idea disposar sempre d'una petita partició amb aquest tipus de format, ja que serà accessible tant des de GNU/Linux, com des de Windows™.

Si bé és cert que és possible muntar un sistema de fitxers des de la línia de comandaments, com per exemple es faria per a muntar el CD-ROM,

```
/etc/apt# mount /dev/CD-ROM /CD-ROM -t iso9660 ro
```


és molt més còmode tenir la informació introduïda en l'arxiu `/etc/fstab`, ja que això ens permetrà fer el mateix teclejant tan sols:

```
/etc/apt# mount /CD-ROM
```

9.8. Configuració de dispositius

Després d'establir les bases per a l'administració de paquets, podem abordar la tasca de començar a configurar el sistema a la mida de les nostres necessitats. Aquest procés consta, bàsicament, de dues parts: configuració dels diferents dispositius de maquinari que tinguem instal·lats a l'ordinador i instal·lació del programari que utilitzarem.

La configuració del maquinari del sistema sol ser la part que costa més esforç en general, ja que a cada ordinador trobarem dispositius diferents, i per tant cada ordinador serà un món. En general, en els sistemes GNU/Linux es pot configurar qualsevol dispositiu, per estrany que sigui, encara que segons el seu grau d'estandardització, això serà més o menys complicat. Però també és cert que durant aquest procés és quan més s'aprèn, ja que, en general, configurar un dispositiu implicarà sempre certes tasques prèvies, com informar-nos exactament de quin tipus de dispositiu disposem, llegir documentació sobre com aquests tipus de dispositius s'integren en els sistemes GNU/Linux, com es fa aquesta integració per al nostre dispositiu en particular, etc.

Atès que no tots els fabricants de maquinari donen suport als sistemes GNU/Linux, i n'hi ha que ni tan sols no faciliten la informació necessària perquè els desenvolupadors de la comunitat puguin escriure el codi necessari per a poder integrar aquests dispositius al sistema operatiu, es recomana sempre que a l'hora d'adquirir maquinari nou, ens informem sobre quin és exactament el producte que volem adquirir (informació en general molt més precisa que la que solen facilitar els proveïdors de maquinari, i de la qual, de vegades, no disposen), si aquest està plenament suportat, de quina informació disposem per a integrar el nou producte en el nostre sistema, etc. Els aspectes generals que cal considerar, a l'hora de fer una nova adquisició, són a grans trets: el grau d'estandardització i qualitat del producte. Pel que

fa a l'estandardització, com més estàndard sigui el producte, segur que més usuaris en disposaran i, per tant, es fa molt més probable que estigui plenament suportat. Quant a la qualitat del producte, ja fa alguns anys que fabricants de maquinari, per a reduir costos, van començar a substituir funcions que inicialment s'implementaven via maquinari per solucions programari (el cas més comunament conegut d'aquesta pràctica és, potser, el dels mòdems coneguts com a *winnòdems*). Això es tradueix, d'una banda, en una baixada de rendiment del sistema, ja que pressuposa carregar la CPU amb noves tasques, per a moltes de les quals ni tan sols no ha estat dissenyada i, de l'altra, la necessitat de disposar del programari que suplanti el maquinari eliminat i que, en general, només està desenvolupat per a cert tipus de sistemes operatius; per aquesta raó, es recomana en general defugir de qualsevol producte que es distingeixi per ser dissenyat per a un sistema operatiu determinat.

Al llarg de cada subsecció de configuració es comentaran alguns aspectes sobre els diferents dispositius que ens podem trobar, i quins problemes porten implícits.

Abans de començar a configurar els diferents dispositius del nostre sistema, recordarem algunes estratègies que ens poden ser d'utilitat per a aquesta finalitat. En primer lloc, mitjançant el comandament `lspci` podem obtenir molta informació sobre com aquests dispositius han estat reconeguts pel sistema durant el procés d'arrencada. Si aquesta informació no ens és suficient, sempre podem recórrer al directori virtual `/proc/`, on queda registrada tota la informació sobre el maquinari del sistema, entre d'altres. També poden ser d'utilitat els fitxers de *log*, ubicats a `/var/log/` (una pràctica interessant per veure com evoluciona temporalment el contingut d'aquests fitxers és utilitzar el comandament `tail` amb el paràmetre `"-f"` i redirigir la seva sortida a una tty que no usem; a tall d'exemple `"tail -f /var/log/messages > /dev/tty10"`).

9.8.1. Configuració del ratolí

Igual que s'ha fet en el taller de *Lnoppix*, el DAEMON que s'encarregarà de gestionar el ratolí serà *gpm*. Procedim, doncs, a instal·lar el paquet:

```
/etc/apt# apt-get install gpm
```

En finalitzar la instal·lació, s'arrenca automàticament un *script* per a assistir-nos en la configuració del ratolí: els paràmetres que li hem de passar són essencialment els mateixos que li vam passar en el seu moment en el taller de KNOPPIX, però si alguna cosa fallés, sempre podem tornar a llançar el programa de configuració mitjançant el comandament `gpmconfig` (en general la configuració "`-m /dev/psaux -t imps2`" hauria de ser vàlida per a la majoria de ratolins de tres botons PS2). La configuració que utilitzarà `gpm` cada vegada que arrenqui es desa a `/etc/gpm.conf`.

Es recomana que el ratolí tingui tres botons, ja que s'acostuma a assignar funcions a tots tres, en especial en els entorns gràfics. Per aquesta raó, si disposem d'un ratolí de tan sols dos botons, haurem d'emular el tercer polsant tots dos alhora.

Arrencada i parada de gpm

Com ja s'ha dit, el programa encarregat de gestionar el funcionament del ratolí és un DAEMON. Per això, tant per a llançar-lo com per a aturar-lo procedirem de la mateixa manera que es fa amb qualsevol DAEMON. Aquesta subsecció servirà d'exemple per a mostrar com s'engeguen i s'aturen els .

Tant l'arrencada com la parada d'un DAEMON es fa mitjançant un *script* resident a `/etc/init.d/`. En general, si s'invoca aquest *script* sense cap paràmetre, ell mateix ens mostrarà una línia d'ajuda per a orientar-nos en el seu ús. Procedim, doncs, a parar el DAEMON `gpm`:

```
/etc/init.d# ./gpm stop
Stopping mouse interface server: gpm
```

Mitjançant "`ps aux`" podem comprovar que, efectivament, no hi ha cap procés corrent anomenat `gpm`, i a més podem veure que si movem el ratolí, no es mostra res per pantalla. Ara procedim a engegar-lo:

```
/etc/init.d# ./gpm start
Starting mouse interface server: gpm
```

Si movem el ratolí observarem com a la pantalla apareix el seu punter. Ara procedim a analitzar si en engegar l'ordinador aquest DAEMON s'engegarà automàticament. El fitxer `/etc/inittab` ens indica en quin *runlevel* s'engegarà el sistema operatiu: per defecte, el 2; per tant, en aquest arxiu hauríem de trobar una línia com la que segueix:

```
# The default runlevel.
id:2:initdefault:
```

Comprovem, doncs, si en el directori `/etc/rc2.d/` hi ha un enllaç simbòlic a `/etc/init.d/gpm`:

```
/etc/init.d# ls -l ../rc2.d/ | grep gpm
lrwxrwxrwx    1 root    root          13 feb 21 13:03 S20gpm -> ../init.d/gpm
```

Nota

El comandament per a crear enllaços simbòlics entre el directori d'un *runlevel* determinat i `/etc/init.d/` és `update-rc.d`.

Si aquest enllaç simbòlic no existís, i volguéssim que `gpm` s'arrenqués automàticament durant el procés d'arrencada, l'hauríem de crear manualment mitjançant `ln -s`. Si en canvi l'enllaç existís i no volguéssim que `gpm` s'engegués, durant el procés d'arrencada n'hi hauria prou d'esborrar aquest enllaç simbòlic; no obstant això, no és recomanable esborrar els *scripts* de `/etc/init.d`, ja que són molt útils amb vista a arrencar i aturar.

9.8.2. Configuració de mòdems

Igual que la resta de maquinari, els mòdems es poden configurar de manera totalment manual, però aquesta pràctica, en general, ha passat a formar part del passat, ja que amb el temps s'han desenvolupat eines prou potents i fiables que ens poden ajudar a estalviar-nos la farragosa tasca de configurar un mòdem manualment. Una d'aquestes eines és `pppconfig`, que és el que es proposa en aquest text per a configurar el nostre mòdem.

Però abans de començar la configuració del nostre mòdem, cal posar de manifest que no són realment mòdems tots els dispositius que s'anuncien o es venen amb aquest nom. Com ja s'ha dit, molts fabricants, amb l'objectiu de reduir costos, han substituït components físics per programari, i són probablement els mòdems els primers

dispositius que històricament van ser víctimes d'aquestes pràctiques. Cal prestar una especial atenció als mòdems interns, ja que en realitat són pocs els que incorporen tot el maquinari propi d'aquests dispositius. Aquests són fàcilment recognoscibles per la diferència de preu respecte als falsos mòdems. Per aquesta raó, molts d'aquests dispositius s'han vist reduïts a mers esclaus del seu programari (per a més informació sobre aquests dispositius i de la seva integració en GNU/Linux vegeu:

- <http://www.tldp.org/HOWTO/Winmodems-and-Linux-HOWTO.html>,
- <http://www.tldp.org/HOWTO/Linmodem-HOWTO.html>,
- <http://www.idir.net/~gromitkc/winmodem.html>).

Per aquesta raó, en general es recomana utilitzar, sempre que sigui possible, mòdems externs. Independentment que es disposi d'un mòdem real o no, es recomana la lectura de <http://www.tldp.org/HOWTO/Modem-HOWTO.html>.

Atès que utilitzarem `pppconfig` per a configurar el nostre mòdem, si no l'instal·lem durant el procés d'instal·lació del sistema (es pot provar intentant llançar l'aplicació directament, és a dir, teclejant `pppconfig`, o mitjançant "`dpkg -l | grep pppconfig`"), el primer que hem de fer és instal·lar l'aplicació:

```
~# apt-get install ppp pppconfig
```

Una vegada ens trobem a la pantalla principal de la interfície d'instal·lació, seleccionarem l'opció `Create a connection`, i a la pantalla següent introduïrem el nom amb què ens referirem a aquesta configuració, ja que és possible configurar i gestionar més d'una connexió.

Després d'assignar un nom a la nova configuració, hem de configurar l'accés al DNS, podem escollir l'opció per defecte, assignació estàtica de DNS, o per assignació dinàmica de DNS, si sabem del cert que el nostre PSI durant el procés de connexió ens facilita les adreces dels DNS. Si escollim l'opció per defecte, se'ns demanarà que entrem les IP dels DNS que volem utilitzar, i que s'emmagatzemaran en el fitxer `/etc/ppp/resolv/`.

A la pantalla següent hem d'escollir el mètode d'autenticació per a establir la connexió. En general, i llevat de casos excepcionals, escollirem la primera opció, PAP. Seguidament, facilitarem el nom d'usuari i la contrasenya de connexió, i seleccionarem la velocitat d'accés; la que es proposa per defecte, 115200, en la majoria de connexions hauria de funcionar sense cap problema. Després d'especificar si la nostra línia telefònica va per polsos o per tons (actualment, la majoria ja van per tons), entrarem el número que s'ha de marcar i que ens ha d'haver facilitat el nostre PSI.

Arribats en aquest punt, pppconfig ens proposa l'execució d'autode-tecció del mòdem. Amb el mòdem en marxa podem deixar que sigui el mateix programa el que detecti a quin port està connectat el mòdem, o ho podem fer nosaltres manualment (recordant sempre la correspondència: primer port sèrie, COM1, /dev/ttyS0, segon port sèrie, COM2, /dev/ttyS1, etc.).

Una vegada hàgim entrat la ttyS a la qual està connectat el mòdem, accedirem a una pantalla de resum de les dades que hem entrat, la qual també ens ofereix la possibilitat d'establir opcions avançades (en general, no necessàries). Si les dades són correctes, podem escollir l'opció `Finished Write files and return to main menu`, i després de confirmar l'operació, de retorn al menú principal de la interfície, si no volem configurar cap altra connexió, triarem l'opció `Quit Exit this utility` per a tornar a la línia de comandaments. Una vegada en la línia de comandaments, podem comprovar que les dades s'han desat correctament a `/etc/ppp/peers/nomdeconfiguracio`.

Per als usuaris de connexions PPP també pot ser interessant instal·lar el paquet `pppstatus` per a monitoritzar el trànsit de la connexió, entre d'altres.

Establiment i finalització de connexió: `pon`, `pooff`

Per a establir la connexió, n'hi haurà prou de teclejar la instrucció `pon` seguida del nom de connexió que vulguem utilitzar; si només hem configurat una connexió, no serà necessari especificar-ne el nom. En principi, si no es restringeix l'ús, `pon` pot ser executat per qualsevol usuari.

Per a finalitzar la connexió n'hi haurà prou d'executar el comandament `poff`.

9.8.3. Configuració de mòdems DSL

De la mateixa manera que s'ha fet en la secció anterior per a la configuració de mòdems tradicionals, utilitzarem una eina per a configurar els mòdems DSL: `pppoeconf`. Però abans de començar es recomana la lectura de <http://www.tldp.org/HOWTO/DSL-HOWTO/> i de <http://www.tldp.org/HOWTO/ADSL-Bandwidth-Management-HOWTO/>.

9.8.4. Configuració de targetes de xarxa

Si pel motiu que sigui no hem configurat la targeta de xarxa durant el procés d'instal·lació, ara és el moment de fer-ho. Encara que pràcticament totes les targetes de xarxa estiguin suportades en els sistemes GNU/Linux, ja que aquestes són una peça clau per a un sistema operatiu orientat a xarxes, una vegada més es recomana l'ús de maquinari tan estàndard com sigui possible per a no tenir complicacions a l'hora de configurar-lo. Pot ser que la nostra targeta estigui suportada de dues maneres diferents: la primera és que el seu programador de control hagi estat compilat directament dins del mateix nucli, i la segona és que el programador de control hagi estat compilat en forma modular, perquè es carregui posteriorment.

La manera més senzilla de saber si el programador de control de la nostra targeta de xarxa ha estat compilat dins del mateix nucli és analitzant el missatge de retorn de `dmesg` tal com s'ha fet en el punt 5.4.3. La configuració completa del `flavour` `bf24` es pot trobar a <ftp://ftp.debian.org/debian/dists/woody/main/disks-i386/current/bf2.4/kernel-config>.

Si després que s'analitzi detalladament la sortida del comandament `dmesg` arribem a la conclusió que el programador de control per a la nostra targeta no ha estat carregat, podem executar el comandament `modconf`, que serveix per a carregar mòduls al nucli que han estat compilats juntament amb ell, i comprovar si el programador de control per a aquesta apareix en la subsecció `kernel/drivers/net`.

Si és així, n'hi haurà prou de seleccionar-ho per a carregar-lo en el nucli.

Si la nostra targeta de xarxa no està suportada per defecte, haurem de recórrer a la recompilació del nucli.

Una vegada fet això, editarem el fitxer `/etc/network/interfaces` per passar els paràmetres corresponents a la nostra xarxa a la targeta. Una possible configuració seria:

```
# /etc/network/interfaces -configuration file for ifup(8), ifdown(8)

# The loopback interface
auto lo
iface lo inet loopback

# The first network card this entry was created during the Debian installation
# (network, broadcast and gateway are optional)
auto eth0
iface eth0 inet static
    address 158.109.69.132
    netmask 255.255.0.0
    network 158.109.0.0
    broadcast 158.109.255.255
    gateway 158.109.0.3
```

Es recomana la lectura d'<http://www.tldp.org/HOWTO/Networking-Overview-HOWTO.html>, també <http://www.fokus.gmd.de/linux/HOWTO/Net-HOWTO/> i el man d'`interfaces`. Si no disposéssim de targeta de xarxa i volguéssim fer proves, sempre podríem recórrer al mòdul `dummy`; en aquest cas, el dispositiu en comptes de dir-se `eth0` s'anomenaria `dummy0`.

Si es vol configurar més d'una targeta de xarxa al mateix ordinador (pràctica molt habitual en *gateways*, entre d'altres), és necessari passar els paràmetres corresponents al nucli durant el procés d'arrencada (utilitzant `append` a LILO, per exemple) per a evitar conflictes entre dispositius.

Arrencada i parada de serveis de xarxa: `ifup`, `ifdown`

La reinicialització de tots els serveis de xarxa (els de `/etc/network/interfaces`) es pot fer mitjançant l'*script* `/etc/init.d/networking` amb el paràmetre `restart`.

`ifup` s'utilitza per a engegar els serveis de xarxa d'una interfície determinada, i `ifdown` per a parar-los. Així doncs, per a la configuració anterior, si volguéssim detenir i tornar a engegar els serveis d'`eth0`, el que faríem és el següent (s'utilitza el comandament `ifconfig` per a comprovar-ne els resultats):

```
~# ifconfig
eth0 Link encap:Ethernet HWaddr 00:01:02:B4:3A:61
      inet addr:158.109.69.132
      Bcast:158.109.255.255 Mask:255.255.0.0
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:36409683 errors:0 dropped:0 overruns:221 frame:0
      TX packets:35938 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      RX bytes:1489273710 (1.3 GiB) TX bytes:20116974 (19.1 MiB)
      Interrupt:5 Base address:0x9400

lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      UP LOOPBACK RUNNING MTU:16436 Metric:1
      RX packets:823 errors:0 dropped:0 overruns:0 frame:0
      TX packets:823 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:3169619 (3.0 MiB) TX bytes:3169619 (3.0 MiB)

~# ifdown eth0
~# ifconfig
lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      UP LOOPBACK RUNNING MTU:16436 Metric:1
      RX packets:823 errors:0 dropped:0 overruns:0 frame:0
      TX packets:823 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:3169619 (3.0 MiB) TX bytes:3169619 (3.0 MiB)
```

```

~# ifup eth0
~# ifconfig
eth0 Link encap:Ethernet HWaddr 00:01:02:B4:3A:61
inet addr:158.109.69.132 Bcast:158.109.255.255 Mask:255.255.0.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:36420981 errors:0 dropped:0 overruns:221 frame:0
TX packets:35965 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:1490867554 (1.3 GiB) TX bytes:20118868 (19.1 MiB)
Interrupt:5 Base address:0x9400

lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:823 errors:0 dropped:0 overruns:0 frame:0
TX packets:823 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:3169619 (3.0 MiB) TX bytes:3169619 (3.0 MiB)

```

9.8.5. Configuració d'impressores

Tenir configurada la impressora pot ser de gran utilitat, ja que això ens permetrà, entre altres coses, imprimir els fitxers de man, els de configuració, etc. per a poder-los estudiar més detingudament sobre format paper.

Pel que es refereix al tipus d'impressora preferible, en un entorn domèstic seria una que usés el port paral·lel (/dev/lpX, normalment /dev/lp0) amb l'objectiu de garantir que es tracta d'una impressora no dependent del seu programari, ja que la majoria de les noves que apareixen en el mercat, especialment les de port USB, són impressores generalment dissenyades per a un sistema operatiu determinat. Aquestes impressores són popularment conegudes com a *winprinters* (i la seva gènesi és semblant a la dels denominats *winmòdems*). En un entorn professional, l'òptim seria disposar d'una impressora que incorporés la seva pròpia interfície de xarxa i que, per tant, fos un node més d'aquesta. Per a més informació sobre impressores i la seva integració en els sistemes operatius GNU/Linux, es recomana visitar la pàgina <http://www.linuxprinting.org>, on trobarem una llista exhaustiva de les impressores existents en el mercat, i el seu grau de suport.

La topologia general del sistema d'impressió sota GNU/Linux és la de client-servidor. El servidor, `lpd` (en el sistema BSD), és tipus DAEMON, i en conseqüència el manipularem com a tal. L'arxiu de configuració del servidor és `/etc/printcap`. Hi podem configurar tantes impressores com vulguem. Davant de la presència de més d'una impressora configurada, els clients usen, normalment, el paràmetre “-p” seguit del nom de la impressora, per a referir-s'hi.

Com hem vist, configurar correctament una impressora pot ser una mica difícil. Per aquesta raó, han aparegut diversos projectes per desenvolupar interfícies que en faran més amena la configuració, alguns dels quals aporten el seu propi servidor d'impressió i les seves pròpies eines client.

Impressió de fitxers de text

Els formatadors són programes que s'utilitzen, principalment, per a transcriure fitxers en format text a format PostScript (el llenguatge PostScript és el que històricament ha tingut més implementació en el camp de la impressió). Aquests programes ens poden ser útils, ja que ens permeten passar a format paper l'ajuda dels comandaments, fitxers de configuració, etc. per poder-los estudiar d'una manera més còmoda. Entre aquests trobem `mpage` (dins del paquet amb el mateix nom), o `enscript` (també empaquetat amb aquest mateix nom, i més potent que l'anterior). A continuació, es detallen un parell de línies per exemplificar-ne l'ús:

```
man man | mpage -2 -o | lpr
```

Mitjançant aquesta línia redireccionem la sortida del `man` de `man` a `mpage`, que li dona format a dues columnes per full, sense marges, i redireccionem la seva sortida al client d'impressió `lpr`:

```
enscript /etc/fstab -B -fTimes-Roman7 -r
```

Amb aquesta línia farem que s'imprimeixi el contingut del fitxer `/etc/fstab` sense capçalera, utilitzant el tipus de caràcter Times-Roman de mida 7 i de forma apaïxada.

9.8.6. Configuració de targetes de so

A causa de la gran quantitat de targetes de so existents en el mercat, es fa gairebé impossible donar una descripció de com configurar-les totes. Es recomana la lectura d'<http://www.tldp.org/HOWTO/Sound-HOWTO/> i la visita a les pàgines dels dos projectes més destacats quant a so pel que fa a GNU/Linux: <http://www.open-sound.com/> y <http://www.alsa-project.org/>.

A continuació s'exposarà la manera de procedir per a configurar una targeta de so bastant comuna: SoundBlasterPCI (chipset ES1371). Per a aquest tipus de targeta, el comandament `lspci` ens tornarà una línia com la següent:

```
00:0d.0 Multimedia audio controller: Ensoniq 5880 AudioPCI
(rev 02)
```

En primer lloc, carregarem el mòdul corresponent a aquesta targeta de so mitjançant el comandament `modconf`, `kernel/drivers/sound`, `es1371`.

Seguidament, crearem el grup `audio` a `/etc/group` i hi inclourem tots els usuaris que vulguem que tinguin accés al dispositiu de so (si volem que tots els usuaris hi tinguin accés, podem obviar aquest pas, i donar tots els permisos als fitxers `/dev/dsp` i `/dev/mixer`); a continuació associarem els fitxers `/dev/dsp` i `/dev/mixer` al grup nou creat.

Amb això ja tenim configurada la targeta de so. Ara ho podem comprovar direccionant un fitxer d'àudio directament a `/dev/dsp`, com suggereix <http://www.tldp.org/HOWTO/Sound-HOWTO/>, o esperar que tinguem l'entorn gràfic configurat per poder instal·lar aplicacions d'àudio que hi corren a sobre.

9.9. Conclusió

En aquest taller hem après a treballar amb el sistema de paquets de Debian, fet fonamental, ja que això ens ha permès aprendre a ins-

tal·lar, desinstal·lar i gestionar aplicacions. També hem après a configurar diferents dispositius de maquinari, i amb això una cosa bàsica: que amb GNU/Linux això no és tan senzill com en altres sistemes operatius, ja que requereix, en general, tenir un coneixement més profund tant del mateix dispositiu com del sistema per si mateix. Però com a compensació, podem assegurar que, una vegada configurat un dispositiu, aquest funcionarà perfectament i no ens n'haurèm de preocupar més. I tot això considerant que encara no hem configurat l'entorn gràfic (probablement una de les missions més complicades en un entorn GNU/Linux). No obstant això, i per a animar els que a hores d'ara puguin pensar que endinsar-se en el món GNU/Linux ha estat una mala idea, una missió fallida i una pèrdua de temps, ens agradaria esmentar un paràgraf del llibre Debian GNU/Linux 2.1 de Mario Camou, John Goerzen i Aaron Van CouWenberghe, part I, capítol 1, *Why Linux Is Better*:

“Windows NT, however, learned to speak the language of the Internet just a few years ago. It is not refined by any stretch imagination, but plagued with bugs and inconsistencies. Bussinesses need a solution that they can put online and expect to remain available 100% of the time, but Windows NT cannot meet this need. On the contrary, a Windows NT administrator's most common job is crash recovery; he wastes too much time doing busywork, such as booting servers that have gone down [...]”

10. Arquitectura X-Window

10.1. Què és X-Window?

X-Window és una arquitectura de finestres dissenyada a mitjan anys vuitanta per a poder disposar d'un entorn gràfic en estacions de treball. A diferència d'altres entorns de finestres, l'arquitectura X-Window es va dissenyar per a ser independent de plataforma, de manera que es pogués instal·lar a qualsevol ordinador que correués un sistema tipus UNIX. Encara que l'arquitectura de finestres X-Window hagi tingut una història dilatada en la qual s'han utilitzat diferents tipus de llicències, diverses implementacions i molts equips de desenvolupament diferents, actualment s'utilitza, majoritàriament, la implementació que ha desenvolupat el projecte XFree86 Inc, anomenada XFree86. Aquesta implementació es distribueix amb llicència *open source*, que encara que no sigui exactament igual que la GPL o les seves variants, té característiques semblants que permeten accedir al seu codi font, la seva redistribució, etc. Per aquest motiu, en la majoria de distribucions de GNU/Linux (i cada vegada més en altres sistemes operatius –com Mac OS X–) s'incorpora aquesta implementació d'X-Window.

X-Window està dissenyat amb una arquitectura client/servidor. Aquest tipus d'arquitectura significa que el programari està estructurat en dos informes totalment independents (client i servidor) que es comuniquen a partir d'un enllaç de comunicació. Encara que això impliqui que el disseny i la codificació siguin una mica més complexos, aquesta arquitectura proporciona una flexibilitat total en el sentit que el client i el servidor poden estar ubicats en llocs diferents i utilitzant plataformes i/o sistemes operatius diferents. A més, podem aprofitar molt més un mateix client, ja que aquest podrà donar servei a més d'un servidor alhora. D'aquesta manera, els ordinadors servidors poden treballar amb un entorn gràfic i els recursos del client. Naturalment, aquesta arquitectura també ens permet treballar amb X-Window localment en la màquina on està situat el client, encara que no sigui indispensable.

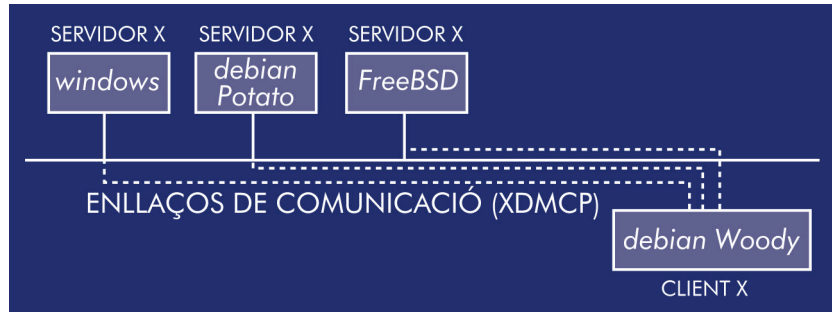
Contingut complementari

Una arquitectura de finestres (o sistema de finestres) és un entorn gràfic que ens proporciona la possibilitat de tenir diferents aplicacions localitzades en diverses regions de la pantalla, generalment delimitades per algun tipus de finestra. Aquests entorns solen proporcionar mecanismes per al desplaçament i la manipulació d'aquestes finestres de manera que la feina pugui ser més interactiva i amena.

Els components de què està compost X-Window són: client, servidor i enllaç de comunicació. Client i servidor estan dissenyats per a ser independents de plataforma i, en el cas de l'enllaç de comunicació, per a ser independent del protocol de xarxa.

D'aquesta manera, podem utilitzar X-Window en qualsevol tipus d'escenari; per exemple, podríem tenir el servidor instal·lat en un ordinador amb WindowsTM, connectant-se a un client amb GNU/Linux, i utilitzar com a canal de comunicació Internet (protocol IPv4). Encara que la configuració de cada un d'aquests components (sobretot el client) sí que depèn, en certa manera, de la plataforma on estigui instal·lat, l'enllaç de comunicació ens permet aïllar els components, donant-los un llenguatge propi perquè s'entenguin.

Aquest enllaç utilitza un protocol propi denominat XDMCP (*X display manager control protocol*), que és en un nivell superior al de la xarxa de comunicació utilitzada (per això és independent de xarxa).



En aquesta arquitectura, el servidor està ideat per a recollir els esdeveniments que es produeixen pels dispositius d'entrada com el teclat, el ratolí, etc. i enviar-los al client. El client processa aquests esdeveniments i respon al servidor, que mostra els resultats en els dispositius de sortida (generalment el monitor). Encara que la primera impressió que pot suggerir-nos aquest disseny és que el temps de resposta ha de ser molt lent, el protocol XDMCP està especialment dissenyat per a proporcionar un enllaç ràpid entre client i servidor, de manera que es pugui treballar realment de manera interactiva. En els únics escenaris en què podem notar aquest inconvenient és en connexions remotes utilitzant xarxes de comunicacions lentes.

En resum, doncs, les característiques i funcions principals de cada un dels components d'X-Window són les següents:

Client	Gestió de diferents servidors simultàniament
	Dependent de plataforma
	Processament de les aplicacions
Servidor	Control del <i>display</i> de l'usuari
	Independent de plataforma
	Processament dels dispositius d'entrada
Enllaç	Dissenyat per a poder treballar interactivament
	Pensat per a minimitzar el trànsit a la xarxa
	Transparent (independent de xarxa)

A mesura que les targetes gràfiques han evolucionat, cada vegada més aplicacions i jocs necessiten un processament en 2D o 3D més ràpid. Si bé l'arquitectura de finestres X-Window aporta molts avantatges, quan volem utilitzar aquest tipus d'aplicacions el disseny client/servidor no és el més adequat, ja que no aprofitem les funcions de processament 2D i 3D extremadament ràpid de les targetes gràfiques instal·lades al servidor. Per a solucionar aquest problema, a partir del 1998 va aparèixer una tecnologia anomenada *DRI* (*direct rendering infrastructure*), que permet aprofitar els xips de processament de les targetes per a estalviar feina al client X-Window. D'aquesta manera, continuem tenint tots els avantatges d'X-Window aprofitant els elements específics de les targetes gràfiques.

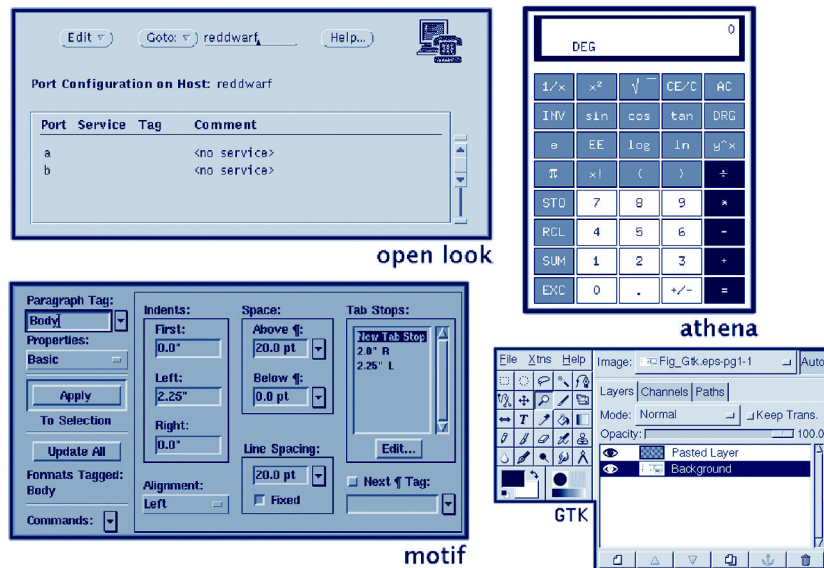
A diferència d'altres sistemes operatius on l'entorn gràfic està íntimament integrat amb la resta de funcions, l'arquitectura X-Window és totalment independent de l'operatiu i no ens limita a cap GUI (*grafic user interface*) determinat. De fet, l'arquitectura només ens proporciona eines gràfiques de baix nivell per manipular la sortida del monitor. Aquestes eines estan incloses en la llibreria Xlib i principalment són funcions per a crear i manipular finestres, operacions amb fonts de caràcters, detecció d'esdeveniments d'usuari i operacions gràfiques. Amb aquestes funcions podem dotar les nostres aplicacions del *look and feel* que vulguem, crear nous GUI, etc. De fet, això va representar una feina addicional per als primers desenvolupadors d'aplicacions en X-Window, ja que a més de programar l'aplicació havien de desenvolupar les seves pròpies llibreries per a crear menús, icones, etc. A mesura que X-Window va créixer, van aparèixer el que anomenem *jocs d'eines* (*toolkits*), que són llibreries generalment im-

Contingut complementari

El *look and feel* és el disseny utilitzat per als botons, barres de desplaçament, menús, etc. d'un entorn gràfic o una aplicació.

plementades amb Xlib i que proporciona un GUI particular. D'aquesta manera, en dissenyar una aplicació podem utilitzar algun d'aquests jocs d'eines que ja proporcionen les eines estàndard per a crear menús, botons, gestionar els *cut and paste*... i centrar-nos a programar l'aplicació en si mateixa. El fet de no marcar cap *look and feel* ha estat una altra de les claus de l'èxit de l'arquitectura X-Window, ja que cada fabricant o desenvolupador de programari se n'ha pogut dissenyar un de propi, marcant la diferència amb els altres.

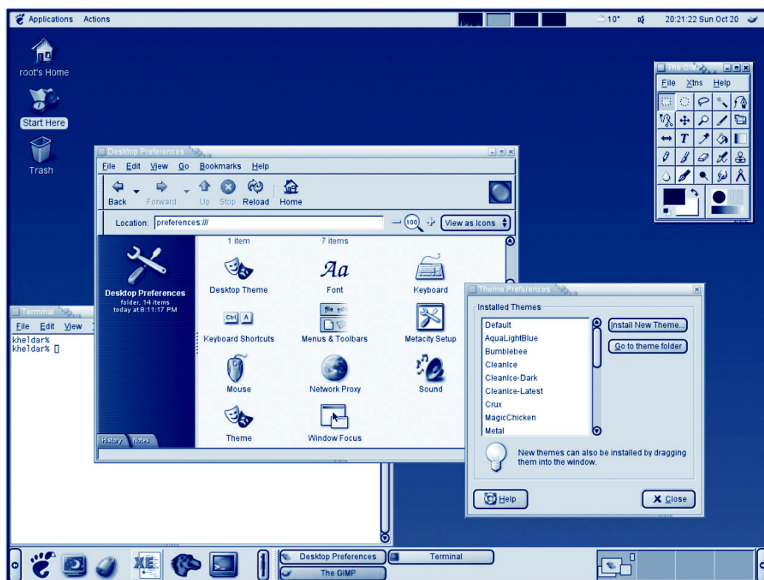
Encara que existeixin molts jocs d'eines diferents, en la figura següent en podem veure alguns dels més populars que s'han utilitzat al llarg de la història d'X-Window:



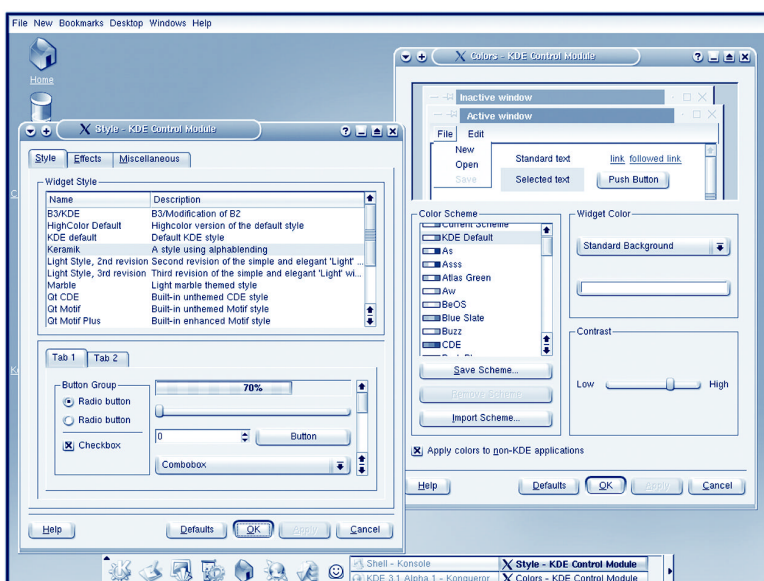
El *window manager* és un servidor especial d'X-Window que s'encarrega de gestionar totes les finestres, els escriptoris, les pantalles virtuals, etc. Naturalment, totes les aplicacions poden funcionar amb qualsevol *window manager*, ja que aquest només s'encarrega de gestionar la finestra on s'ubica el programa. Encara que la programació d'un *window manager* sigui molt diferent de la d'una aplicació, també se solen utilitzar jocs d'eines particulars que proporcionen un *look and feel* determinat. Actualment existeixen desenes de *window managers* diferents (*wmaker*, *sawmill*, *olwmm*, etc.), i així és el mateix usuari qui pot triar el que més li agradi.

Un altre tipus de programari molt relacionat amb X-Window és el que s'encarrega de proporcionar un entorn integrat per a les aplicacions, l'escriptori, les eines d'administració del sistema, etc. Els més populars

que existeixen actualment són el KDE (*the K Desktop Environment*) i el GNOME (*GNU Object Model Environment*). Tots dos proporcionen un joc d'eines particular, un entorn d'escriptori amb moltes funcionalitats i configuracions diferents i una llista d'aplicacions integrades que cada vegada creixen més. La majoria de distribucions de GNU/Linux i el UNIX proporcionen algun d'aquests dos entorns d'escriptori perquè són molt amigables i proporcionen eines i programari propi de gran qualitat que ajuden en gran manera l'usuari per a configurar el sistema i l'escriptori. Tots dos poden funcionar amb qualsevol *window manager* que compleixi una sèrie de característiques bàsiques. En la següent figura podem veure l'aspecte de tots dos:



GNOME



KDE

Finalment, un altre tipus d'aplicació que s'utilitza a X-Window és el *session manager*, que són una sèrie de programes que permeten desfer la configuració d'una sessió d'usuari determinada perquè en arrencar de nou X-Window es carreguin les aplicacions que tingui configurades. Generalment, en els entorns integrats ja s'incorporen aquestes eines automàticament; si no, podem recórrer al que la mateixa infraestructura d'X-Window proporciona: l'`xsm`.

Activitats

14. Llegeix la història d'X-Window en l'article:
http://www.linux-mag.com/2001-12/xfree86_01.html.
15. Vegeu alguns dels *window manager* i entorns d'escriptori existents a: <http://www.xwinman.org>.

10.2. Configuració

Actualment, les versions de la implementació XFree86 que més s'utilitzen són les 4.X, la configuració de les quals veurem en aquesta secció. Si bé la majoria de targetes gràfiques del mercat ja estan suportades, és possible que des del moment de l'aparició en el mercat d'una nova targeta fins que es doni suport en X-Window passin unes quantes setmanes o uns quants mesos. De tota manera, cada vegada més els mateixos fabricants donen suport a GNU/Linux i en alguns casos ja proporcionen els seus propis programadors de control per a aquest sistema operatiu. Tot i així, abans de comprar una nova targeta gràfica, sempre és recomanable comprovar si està disponible algun tipus de programador de control per a la distribució que utilitzem.

Per a instal·lar XFree86 al nostre ordinador el primer que haurem de fer és baixar-nos els paquets que contenen les eines bàsiques i el programari per al client i el servidor. Generalment, aquests paquets se solen denominar `xfree86-common`, `xfree86server`, etc., i porten implícites diverses dependències de fonts i algunes utilitats bàsiques per al maneig d'X-Window. Quan ja hàgim instal·lat aquests paquets, hem de configurar adequadament els dispositius

Contingut complementari

Alguns programes típics de configuració d'X-Window són `xf86config` o `XF86Setup`.

dels quals disposem per poder arrencar correctament el client i servidor X-Window. Segons la distribució que utilitzem, es fa ús d'un o un altre programa o, en alguns casos, amb la mateixa instal·lació dels paquets ja es llança una petita aplicació de configuració. No obstant això, aquesta configuració sempre ha de contenir uns passos determinats, que detallem a continuació, classificats segons el dispositiu que cal configurar:

1. Targeta gràfica

- Programador de control: les diferents famílies de targetes gràfiques porten uns microprocessadors específics i utilitzen unes funcions determinades per a fer les seves operacions. Per aquesta raó, hem d'indicar el programador de control adequat per a la nostra targeta. Si no el sabem, podem instal·lar algun tipus d'aplicació per a la detecció de maquinari automàtic; si utilitzem, per exemple, el `discover`, podem saber quin programador de control necessita la nostra targeta amb el comandament `discover --xdriver video`.
- Identificador: l'identificador de la targeta pot ser qualsevol nom amb què vulguem referir-nos a la nostra targeta. Aquest identificador s'utilitza internament per a poder referenciar adequadament les targetes que tenim instal·lades en el sistema.
- Quantitat de memòria: segons la quantitat de memòria de la targeta, podrem inicialitzar els gràfics amb més o menys resolució i amb profunditats de color més o menys elevades. Encara que no sigui imprescindible indicar aquesta quantitat (el sistema ho detecta automàticament), sí que és recomanable especificar-la en la configuració.
- Utilització del *framebuffer* del nucli: el *framebuffer* del nucli és un programador de control especial de Linux que permet fer algunes operacions sobre X-Window. Encara que no sigui obligatori utilitzar-lo, generalment s'usa perquè el servidor d'X-Window es pugui comunicar directament amb el nucli del sistema. De tota manera, si fos problemàtic, sempre es pot desactivar.

2. Teclat

- Regla XKB: perquè el servidor d'X-Window pugui manejar correctament el teclat, necessita saber quines regles hi pot aplicar. Per a la majoria de teclats estàndard dels PC, s'utilitza la regla "xfree86" i per a les estacions de treball Sun, se sol utilitzar la regla "sun".
- Model de teclat: el model de teclat generalment se sol identificar a partir del nombre de tecles que té. Els teclats dels PC estàndard que tenen les tecles de menú i logo solen tenir cent quatre tecles (els identifiquem amb el nom "pc104"). Els teclats que no porten aquestes tecles s'identifiquen com de cent una tecles ("pc101").
- *Keyboard layout*: en aquesta secció hem d'identificar el país del teclat amb la seva referència ISO 3166. En el cas d'Espanya és "es", per a França, "fr", etc.
- *Keyboard options*: opció per a personalitzar algunes tecles del teclat.

3. Ratolí

- Port: el port del ratolí és la connexió que utilitza per a comunicar-se amb l'ordinador. Quan comprem el ratolí, sempre s'indica si és de tipus PS/2, sèrie, etc. En cas que sigui de tipus PS/2, el port serà /dev/psaux, per als ratolins sèrie el port serà /dev/ttyS0 (COM1), /dev/ttyS1 (COM2) i consecutivament.
- Tipus: per a especificar el tipus del ratolí, se sol proporcionar una llista de la qual hem d'escollir el que més s'ajusti al nostre model i fabricant. Generalment, sabent el model del ratolí ja podem escollir adequadament l'opció que li correspon.
- Emulació de tres botons: en el cas que el nostre ratolí només tingui dos botons, es proporciona la possibilitat d'emular el tercer (el del mig) prement tots dos simultàniament. Si el nostre ratolí no té el botó del centre, és recomanable activar aquesta opció perquè alguns programes d'X-Window necessiten que el ratolí tingui els tres botons.

4. Monitor

- **Identificador:** igual que en el cas de la targeta gràfica, la identificació del monitor serveix perquè el sistema el pugui referenciar internament. Li podem posar el nom que vulguem.
- **Monitor tipus LCD:** en la majoria de processos de configuració se'ns preguntarà si el nostre monitor és de tipus LCD (pantalla TFT). És important respondre correctament a aquesta pregunta perquè el maneig d'un tipus o altre de monitor varia considerablement.
- **Característiques:** en la configuració de característiques es preguntarà les resolucions màximes que pot obtenir el nostre monitor, la freqüència de refresc, etc. Encara que segons el programa utilitzat per a configurar X-Window es plantejaran més o menys preguntes d'aquest estil, és important tenir a mà la informació del monitor i contestar adequadament per a poder aprofitar-ne al màxim les característiques.
- **Resolucions disponibles:** en aquest pas hem d'assenyalar quines resolucions volem poder mostrar al nostre monitor quan iniciem X-Window. També és habitual que se'ns pregunti la profunditat de color que volem utilitzar per defecte; el més recomanable és utilitzar-ne una d'alta (16 o 24 bits) per a poder veure nítidament tots els colors.

Una vegada contestades aquestes preguntes, que poden ser més o menys segons el programa que utilitzem, tota la configuració es desa al fitxer `/etc/X11/XF86Config-4`.

Aquest fitxer està organitzat en les diferents seccions que hem vist i, recorrent al seu manual, veurem que tenim moltes més possibilitats que ens donen una flexibilitat total per configurar de la manera com vulguem les nostres X-Window. Per a provar si realment funcionen, podem executar `X`, amb el qual ens hauria d'aparèixer una pantalla amb quadres blancs i negres molt petits i el punter del ratolí com una `X` (per a sortir-ne podem utilitzar `CTRL + ALT + BACKSPACE`).

Contingut complementari

Quan utilitzem `startx` o `xinit` s'executen les instruccions del fitxer `/etc/X11/Xsession`. Si en el directori de l'usuari que iniciés X-Window hi hagués un fitxer `.xsession`, s'executarien les instruccions d'aquest en lloc de l'altre.

Si tenim instal·lat algun *window manager*, el més habitual per a arrencar X-Window és utilitzar algun dels *shell scripts* `xinit` o `startx`. Aquests s'encarreguen de llançar el *window manager* configurat i fan algunes altres accions necessàries per a inicialitzar correctament X-Window. Una vegada tenim la pantalla en mode gràfic, en podem canviar la resolució amb les tecles `CTRL + ALT +` i `CTRL + ALT + -`, o tornar a les consoles de text amb `CTRL + ALT + F1`, `CTRL + ALT + F2`, etc. (amb `CTRL + ALT + F7` tornàriem a la gràfica).

Una altra característica important en la configuració d'X-Window és la de la configuració dels *Xwrappers*. Els *Xwrappers* ens permeten controlar quins usuaris poden iniciar una sessió amb X-Window. En el fitxer `/etc/X11/Xwrapper.config` es troba la directiva `"allowed users"`, amb la qual especifiquem qui està autoritzat per a engegar X-Window amb els valors:

- `"Console"`: qualsevol usuari que hi hagi en un consola local pot iniciar X-Window.
- `"Rootonly"`: només el `root` pot iniciar X-Window.
- `"Anybody"`: qualsevol usuari del sistema pot iniciar X-Window (encara que no hi estigui connectat localment).

Això és molt útil, sobretot, en administrar un servidor en què generalment no es permet que els usuaris treballin amb l'entorn gràfic per la despesa de recursos que això representa.

10.3. X display manager

En la secció anterior hem vist com configurar X-Window localment. Tal com hem comentat al llarg del capítol, l'arquitectura de finestres X-Window ens permet que client i servidor estiguin instal·lats en diferents ordinadors. Per a configurar el nostre ordinador de manera que faci les funcions d'un client X-Window, hem d'instal·lar algun tipus d'*X display manager*. Aquests programes obren un port de comunicacions que permeten que els seus clients es puguin connectar i treballar amb X-Window remotament. Encara que existeixen molts

programes d'aquest tipus, un dels primers que va aparèixer, i en què es basen molts d'altres, és l'`xdm`.

Els *X display manager* poden actuar tant localment com remotament. Entre altres funcions, el que fan és mostrar una pantalla (en l'entorn gràfic) perquè l'usuari s'identifiqui amb la seva connexió i contrasenya. Funcionen com qualsevol altre DAEMON del sistema, de manera que l'inici i la parada es poden configurar com vulguem (utilitzant els nivells d'execució que el sistema proporciona). Hem de tenir en compte que si el configurem perquè funcioni localment, en engegar el sistema ens trobarem amb la pantalla d'identificació gràfica i no les consoles a què estàvem acostumats (encara que continuen estant disponibles). Amb aquesta configuració ja no podrem utilitzar `startx` o `xinit` per a inicialitzar X-Window, ja que s'executaran per defecte.

Quan instal·lem `xdm`, tots els seus fitxers de configuració es deixaran en el directori `/etc/X11/xdm`. Repassarem què conté cada un d'aquests fitxers:

<code>xdm-config</code>	Localització dels arxius de configuració d' <code>xdm</code> .
<code>xdm.options</code>	Opcions globals de configuració.
<code>Xaccess</code>	Definició dels equips remots als quals deixem accedir.
<code>Xservers</code>	Servidors locals d' <code>xdm</code> .
<code>Xresources</code>	Configuració de la pantalla de connexió: colors, fonts, mida, etc.
<code>Xsetup</code>	<i>Script</i> que s'executarà quan s'engegui <code>xdm</code> .
<code>Xstartup</code>	<i>Script</i> que s'executarà quan un usuari entri a X-Window. Se solen posar accions relacionades amb l' <code>xdm</code> .
<code>Xsession</code>	<i>Script</i> que s'executarà en entrar en una sessió d'usuari. Se solen posar accions especials per als usuaris, encara que també se sol cridar a l'execució del fitxer <code>/etc/X11/Xsession</code> .
<code>Xreset</code>	<i>Script</i> que s'executarà en acabar una sessió d'usuari.

La configuració dels servidors locals la trobem al fitxer `Xservers`. Si volguéssim desactivar el servidor local, podríem comentar totes les línies d'aquest arxiu. D'aquesta manera, encara que tinguéssim instal·lat un client d'X-Window, per defecte no s'iniciaria a la màquina

local. Si al contrari, volguéssim instal·lar-ne més d'un, podríem editar el fitxer i afegir directives com les que segueixen:

```
:0 local /usr/X11R6/bin/X :0 vt7
:1 local /usr/X11R6/bin/X :1 vt8
```

Aquestes dues directives indiquen que volem dues instàncies d'X-Window, una a la consola 7 ("vt7") i l'altra a la 8 ("vt8"), accessibles amb CTRL + ALT + F7 i CTRL + ALT + F8 respectivament. Fixem-nos que cada directiva inclou un ": 0" o ": 1", que fan referència a la instància d'X-Window que manegen. Per defecte, sempre s'utilitza la 0, però en voler més d'un servidor local hem de referenciar-lo d'aquesta manera. Al final de cada una d'aquestes línies podríem afegir paràmetres especials per a cada servidor d'X-Window (a "man X" trobem tots els possibles), com la profunditat de color que volem per a cada un, la resolució de la pantalla, etc. D'aquesta manera, podríem treballar amb diferents sessions d'X-Window obertes tal com fèiem amb les consoles.

Generalment, la configuració per defecte d'xdm no permet connexions remotes per raons de seguretat. Si volguéssim activar aquestes connexions, podríem editar el fitxer `Xaccess` i, utilitzant la sintaxi que se'ns indica, afegir els servidors a què permetem donar aquest servei. També hauríem de comentar la línia `DisplayManager.requestPort: 0` del fitxer `xdm-config`, que per defecte inutilitza totes les connexions que es reben. Una vegada fets aquests canvis, reiniciant el DAEMON d'xdm, el client ja estaria preparat per a servir X-Window a qualsevol servidor que li ho demanés.

Per a les màquines on només volem instal·lar el servidor d'X-Window, hauríem d'instal·lar X-Window tal com hem vist en l'apartat anterior i utilitzar el comandament `X -query IP`, on l'IP hauria de ser la del client. De la mateixa manera que quan teníem més d'un servidor X-Window en una màquina local, si a la màquina ja tinguéssim una altra instància d'X-Window executant-se, hauríem d'utilitzar `X -query IP :1` per a la segona instància, ": 2", per a la tercera i consecutivament.

Contingut complementari

Si al fitxer de `Xaccess` hi ha una línia amb el caràcter `**`, indica que deixem que qualsevol servidor es connecti a l'X-Window del servidor. Utilitzar X-Window remotament sense cap tipus de xifratge pot representar un error de seguretat, amb la qual cosa és molt recomanable informar-se adequadament abans d'utilitzar-lo.

Contingut complementari

`Xnest` és un servidor d'X-Window que ens permet obrir en una finestra una altra instància d'X-Window.

ANOTACIONS

11. Taller de X-windows

11.1. Introducció

En el segon taller es van establir les bases per a manipular i gestionar paquets correctament, i vam aprendre a configurar alguns dispositius. Malgrat això, i a causa de la seva complexitat, no es va abordar el tema de la configuració de la targeta gràfica, o millor dit, de la instal·lació de l'entorn gràfic X. A causa de la complexitat de la seva estructura i configuració, s'ha optat per dedicar un taller monogràfic sobre l'X Window System. En aquest taller aprendrem a instal·lar, configurar i personalitzar aquest sistema. Però no es pretén fer un repàs exhaustiu de tot el sistema, ja que aquesta tasca seria probablement inabordable per raons molt diferents. Es pretenen establir les bases perquè cadascun sigui capaç de configurar el seu propi sistema en funció de la targeta gràfica de què disposi, dels seus gustos i de les seves preferències. En finalitzar el taller, hauríem de ser capaços d'instal·lar un entorn X, i de saber treure partit a la seva increïble potència.

Activitats

A causa de la complexitat del sistema X, es recomana la lectura dels documents següents per a fixar conceptes abans de començar a treballar sobre el sistema. A més, ens aportaran coneixements suplementaris, que es poden posar en pràctica al llarg del taller.

16. <http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/htmlsingle/XWindow-Overview-HOWTO.html>
Es tracta d'un document senzill, que serveix per a assimilar els conceptes bàsics que concerneixen el sistema.
17. <http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/htmlsingle/XWindow-User-Howto.html>
És un document amb continguts més avançats que l'anterior, però també és recomanable.

11.2. Instal·lació del sistema bàsic

Com ja s'ha dit, l'X-Window System és un sistema molt complex que integra moltes llibreries i aplicacions, algunes de les quals són fonamentals per al seu funcionament, encara que la majoria només s'han d'instal·lar si, per les nostres necessitats, les hem d'utilitzar. Aquesta és una de les raons per les quals, en Debian, el sistema ve distribuït a molts paquets diferents, dels quals només instal·larem aquells que siguin necessaris.

11.2.1. Diferents estratègies per a instal·lar els paquets

Com que la interdependència entre els diferents paquets és molt forta, podem aprofitar aquest fet perquè sigui el mateix sistema de gestió de paquets el que instal·li tots aquells que consideri necessaris per al funcionament correcte d'una aplicació d'alt nivell, entre els quals es trobaran, òbviament, tots els paquets bàsics del sistema. Així doncs, podríem utilitzar `dselect` o `apt-get` per a instal·lar una d'aquestes aplicacions. Però aquesta no és una bona estratègia, ja que implica la pèrdua del control dels paquets que instal·lem i a més pot implicar l'omissió d'alguns paquets essencials, dels quals, pel motiu que sigui, no s'hagi fet referència durant el càlcul de dependències.

Per aquest motiu, es recomana construir el sistema pas a pas, per a comprendre quins paquets s'instal·len en cada moment i per què.

11.2.2. Instal·lació de paquets bàsics

Els paquets bàsics del sistema es poden agrupar, generalment, en:

core. Paquets essencials i imprescindibles per al funcionament correcte del sistema (`x-window-system-core`).

system. Paquets essencials la presència dels quals és altament recomanable (`x-window-system`).

fonts. Paquets de fonts, alguns dels quals són imprescindibles per a disposar, almenys, d'un tipus de fonts en l'entorn.

utils. Paquets de programes que corran sobre el sistema gràfic. Són, en el fons, els que donen la raó d'existència del sistema.

libs. Paquets de llibreries de l'entorn gràfic, que proporcionen funcions a altres programes.

Podem instal·lar els paquets bàsics del sistema mitjançant l'ordre següent:

```
~# apt-get install x-window-system
```

Amb això es calcularan les dependències entre paquets i se'ns informarà que s'han d'instal·lar entorn d'una trentena de paquets. Per a procedir-hi, se'ns demana la confirmació, a la qual contestarem afirmativament.

Durant l'execució de l'*script* de configuració del paquet `x-server-common`, se'ns preguntarà si volem que sigui `debconf` qui configurei els *wrappers*. L'opció més senzilla i pràctica és permetre que la configuració es faci automàticament, i continuar endavant.

L'*script* de configuració d'`xserver-xfree86` ens situarà en la interfície de configuració del servidor. El primer que se'ns pregunta és si volem que s'executi un test d'autodetecció del programador de control que s'ha de carregar per a la nostra targeta. És una bona opció contestar afirmativament a aquesta pregunta, ja que si el test falla o detecta més d'una possibilitat, ens donarà accés a la pantalla de selecció dels programadors de control. Si hem de seleccionar el programador de control manualment, en primer lloc cal llegir la documentació de la targeta gràfica; si no en disposem, també podem utilitzar el comandament `lspci` per a informar-nos de com ha estat detectada pel nucli (generalment l'última línia de retorn). Quan ja tinguem aquesta informació, ens dirigirem a <http://xfree86.org/current/Status.html> per informar-nos de quin és el programador de control que hem de carregar. Amb això ens situem en la pantalla següent. En aquesta se'ns demanarà si volem que sigui `debconf` qui ens guiï en la configuració de l'arxiu `/etc/X11/XF86Config-4`, que és l'arxiu fonamental de configuració del servidor. No és mala idea contestar afirmativament, i, al final, comprovar si la configuració resultant és

Nota

Després que s'hagin desempaquetat els paquets, s'executaran automàticament els *scripts* de configuració de diversos d'ells. Sempre és possible interrompre l'execució d'aquests *scripts* amb la combinació de tecles CTRL + C, i reiniciar el procés tornant a executar el comandament anterior.

Nota

Des del punt de vista de la programació, un *wrapper* és un segment de codi A que combinat amb un altre codi B determina com s'executa aquest codi B. El *wrapper* actua com a interfície entre el codi B que fa les crides i el codi A del *wrapper*, ja que només es pot accedir al codi del *wrapper* a través del mateix *wrapper*. Aquesta estratègia s'utilitza, en qüestions de seguretat, per a evitar que el codi A faci, ell mateix, crides a certes funcions crítiques.

Nota

L'any 2003, un grup del projecte Xfree86 es va escindir a causa de certes discrepàncies amb la direcció pel que fa a polítiques de versions i conflictes dels codis.

L'X.org (<http://www.x.org>) va néixer amb el propòsit d'un model més obert i sense dependre d'interessos comercials, fet que va atraure de seguida molts desenvolupadors que ja feia temps que veien amb mals ulls la política de l'Xfree86. Actualment, gairebé totes les distribucions ja incorporen les aplicacions desenvolupades per l'X.org.

vàlida o no. En primer lloc se'ns pregunta si volem usar el *kernel framebuffer device interface*, i se'ns suggereix que ho fem; en principi, contestarem afirmativament, però tenint molt present que això és una font potencial de problemes, de manera que, una vegada hàgim finalitzat la configuració, si en intentar arrencar el servidor alguna cosa va malament, haurem d'estar atents al contingut de la línia, i provar si amb el valor "false" funciona:

```
#/etc/X11/XF86Config-4
Section "Device"
    Identifier      "Generic Video Card"
    Option          " UseFBDev"           "true"
```

El següent que configura l'*script* és el teclat. En general, i en especial sobre la plataforma PC, el suggeriment d'utilitzar el tipus `xfree86` ha de funcionar correctament, encara que a les pantalles posteriors n'acabarem d'ajustar el funcionament. En la pantalla següent contestarem quin tipus de teclat és el que tenim; en general `pc105` ha de funcionar correctament.

Ha arribat el moment de seleccionar l'idioma del teclat. En aquest camp entrarem `es`; quant al camp de variació idiomàtica cal deixar-lo en blanc, ja que el teclat espanyol no en presenta.

El camp següent, referent a la personalització del teclat, es deixa en blanc tret que es tingui algun propòsit específic. Amb això haurem acabat amb la configuració del teclat i passarem a la configuració del ratolí. Per a això, en primer lloc, se'ns demana a quin port està ubicat (actualment, la majoria a `/dev/psaux`, la primera opció). A continuació se'ns pregunta de quin tipus de ratolí disposem (novament, la primera opció, `IMPS/2`, sol ser la més habitual). Quan ja hem configurat el ratolí, entrem en la configuració de la pantalla. En primer lloc se'ns pregunta si disposem d'una pantalla tipus LCD. Tot seguit se'ns demanarà en quin grau de dificultat volem entrar els paràmetres referents al nostre monitor. La manera més simple (`Simple`) només ens demanarà la mida del monitor, mentre que en la més complexa (`Advanced`) ens demanarà les freqüències de refresc horitzontals i verticals del monitor, entre d'altres. Seguidament se'ns preguntarà amb quines resolucions volem treballar i, en funció dels paràmetres que hàgim entrat en l'apartat anterior, se'ns faran

suggeriments diferents. A continuació se'ns demanarà quina serà la resolució de color que s'usarà per defecte en engegar el servidor (actualment la majoria de targetes suporten, sense cap problema, resolucions de 24 bits). Amb això haurem acabat la configuració del servidor.

Actividad

18. Quan ja l'hàgim instal·lat, editeu el fitxer `/etc/X11/XF86Config-4`, i intenteu associar les diferents directrius als paràmetres que hem entrat durant l'execució de l'*script* de configuració del servidor.

Nota

Una bona manera d'aconseguir un fitxer `/etc/X11/XF86Config-4`, que funcioni, és copiar el que genera KNOPPIX quan entrem en mode gràfic; així doncs, podem fer una còpia d'aquest fitxer, en un disquet, per exemple, i comparar-lo amb el que hem generat mitjançant l'*script* de configuració del servidor.

11.2.3. Inicialització del servidor

Ha arribat el moment de comprovar si el fitxer de configuració del servidor és correcte, i en conseqüència el servidor s'engega com cal. Per a això n'hi ha prou d'executar el comandament `startx`.

Si tot funciona correctament, al cap d'uns moments la nostra pantalla adquirirà un fons de malla de colors grisencs, i en el centre de la pantalla apareixerà una aspa. Hem fet un gran pas endavant, ja que configurar el servidor perquè arrenqui és el més difícil de l'entorn X. Ara només és qüestió de temps per a acabar de donar a l'entorn l'aspecte que vulguem. Mitjançant el ratolí podem moure l'aspa, i prement els botons del mig i esquerre, podem explorar una mica les possibilitats d'aquest entorn gràfic una mica rudimentari. Per a sortir-ne i continuar amb la configuració del sistema, cal prémer el botó esquerre del ratolí, situar-nos sobre `Exit` opció `Yes, really quit`, o simplement polsar la combinació de tecles `CTRL-ALT-BACKSPACE`.

Si al contrari, al cap d'uns moments retornem a la consola alfanumèrica, és que el servidor no s'ha pogut engegar adequadament. Ha arribat el moment d'estudiar detingudament el fitxer de `log (/var/log/XFree86.0.log)` i intentar detectar les possibles fonts d'errors. Les més comunes solen ser: mala elecció del programador de control que cal carregar (si el procés de selecció l'hem deixat a les mans de l'*script*, haurem de consultar la pàgina abans esmentada per assegurar-nos que el programador de control que l'*script* ha es-

collit és el correcte), posar la directiva `UseFBDev` a `true`, quan ha d'estar en `false`, usar resolucions o freqüències de refresc més altes que les que la targeta pot suportar, etc.

Una altra possibilitat és que, fins i tot estant suportada la targeta pel Xfree86 Project, pot ser que el paquet que usem no la suporti. Davant aquesta situació, se suggereix que s'incorpori la línia següent en el fitxer `/etc/apt/sources.list`:

```
deb http://people.debian.org/~blade/woody/i386/ ./
```

i que, després d'executar el comandament `apt-get update`, es torni a executar el comandament d'instal·lació de l'`X-Window-System` (`apt-get install X-Window-System`). Amb això acudirem a una pàgina no oficial per obtenir els paquets de versions més actualitzades del sistema X Window, en els quals, probablement, ja es donarà suport a la nostra targeta.

Amb tot això, cal destacar que la configuració correcta d'una targeta de vídeo pot ser una tasca veritablement difícil. Per a això, acudirem a cercadors on entrarem paraules clau referents a la nostra targeta juntament amb "X" o "Linux" per a veure si trobem documentació sobre això.

Mitjançant el comandament següent, evitarem que cada vegada que engeguem l'ordinador entrem en el mode gràfic:

```
~# rm /etc/rc2.d/S99xdm
```

Això ens serà útil mentre no tinguem configurat totalment aquest entorn. Una vegada acabat, ja serà cada usuari el que decidirà si vol o no que en arrencar l'ordinador s'entri en el mode gràfic. Per a fer-ho, només caldrà tornar a crear l'enllaç simbòlic que hem esborrat amb el comandament anterior:

```
~# cd /etc/rc2.d
/etc/rc2.d# ln -s ../init.d/xdm S99xdm
```


11.2.4. El fitxer de log

Encara que l'arrencada del servidor hagi resultat reeixida, hem d'analitzar el contingut del fitxer principal de *log* del sistema X, `/var/log/XFree86.0.log`, ja que això ens ajudarà a depurar problemes i errors menors que, tot i que no han interromput l'arrencada del servidor, sí que en fan baixar el rendiment.

Alguns exemples típics poden ser: eliminar la línia del fitxer de configuració que faci referència a les fonts ciríl·liques, que mai no les utilitzem, ja que ni tan sols les hem instal·lades; eliminar d'aquest mateix fitxer tot el que faci referència al Generic Mouse, ja que ho fa incompatible amb el que hem configurat nosaltres, etc.

11.2.5. El servidor de fonts

El sistema gràfic per si mateix no porta associat cap tipus de base de dades sobre les fonts disponibles, ja que això no entra en el domini de les seves funcions. Aquesta és la tasca del servidor de fonts `xfst` (`/usr/bin/X11/xfst`); així, qualsevol programa client podrà usar les fonts disponibles, que s'engeguen automàticament en llançar el servidor X en forma de DAEMON. Existeixen multitud de paquets per a agregar més fonts al conjunt que hem instal·lat per defecte.

Una aplicació que pot ser útil per a conèixer quines fonts tenim instal·lades i quin aspecte tenen és `xfontsel`. Es tracta d'una aplicació escrita mitjançant llibreries d'athena (com la resta dels `xbase`-clients, grup al qual pertany `xfontsel`), en la qual, mitjançant menús desplegable se seleccionen les opcions disponibles de configuració de les fonts i ens mostra alguns caràcters d'exemple. Per a conèixer els caràcters d'una font es pot usar l'aplicació `xfd`.

Així doncs, una vegada sapiguem quina font volem utilitzar per a un programa determinat, podem llançar-lo des d'una `xterm` amb l'argument `-fn` seguit de la descripció de la font (altres programes poden usar noms diferents per a aquest argument). Un exemple pot ser:

```
xterm -fn -misc-fixed-medium-r-normal--20-140-100-100-c-100-iso8859-1 &
```

11.3. Window managers

Els *window managers* són programes client (en realitat se'ls anomena *metaclients*) encarregats de gestionar les diferents finestres que corren sobre l'entorn gràfic i la seva presentació, i també de llançar altres clients (aplicacions). A hores d'ara, ja tenim un *window manager* instal·lat, el *twm*, ja que una instal·lació completa d'un sistema X requereix com a mínim un *window manager*, encara que aquest no formi part del servidor, que corri sobre ell. Com ja hem vist, el *twm* és molt rudimentari; potser, doncs, ens interessarà instal·lar-ne alguns de més complexos, com pot ser *WindowMaker*, *BlackBox*, *qwm*, etc. N'instal·larem alguns i els provarem. L'objectiu final, seguint la filosofia GNU, és que cada usuari acabi usant el programari que prefereixi. Així doncs, es deixa que, una vegada coneguts alguns *window managers* existents, sigui el lector mateix qui decideixi quin usará. Òbviament, és possible tenir més d'un *window manager* instal·lat, encara que només se'n pugui fer córrer un per sessió. (Tanmateix, sí que podem, com a exemple de la flexibilitat del sistema X, tenir dos *window manager* corrent en un mateix ordinador, cada un en un terminal diferent.)

En primer lloc instal·larem, a tall d'exemple, el *qwm*. Es tracta, com veurem en llançar-lo, d'un *window manager* que simula un entorn que probablement ens sigui conegut. Per a fer-ho, n'hi ha prou d'executar la instrucció:

```
~# apt-get install qwm
```

En aquest moment ja tenim dos *window managers* instal·lats: el *twm* i el *qwm*. Per a fer-ne córrer un o altre n'hi haurà prou d'indicar la ruta completa d'ubicació del *window manager* que vulguem utilitzar després del comandament *startx* (recordem que el comandament *whereis* ens pot ser molt útil a l'hora de buscar la ubicació d'un fitxer). Així doncs, per a fer córrer el *window manager* que acabem d'instal·lar, n'hi haurà prou d'executar:

```
~# startx /usr/bin/X11/qwm
```

Per a utilitzar el *twm* se n'hauria d'haver especificat la ruta completa de la manera següent:

```
~# startx /usr/bin/X11/twm
```

Haver d'especificar cada vegada quin *window manager* volem utilitzar, una vegada ens hàgim decidit per un de concret, pot ser una mica pesat. Per a especificar quin *window manager* s'ha d'usar en cas que després del comandament `startx` no se n'especifiqui cap en concret, crearem l'arxiu `.xsession` en el directori arrel de l'usuari amb el contingut següent, en el cas que volguéssim que el *window manager* per defecte fos el `twm`, per exemple:

```
# ~/.xsession
exec twm
```

Si volguéssim que fos `qvwmm` el *window manager* per defecte, n'hi hauria prou de canviar `twm` per `qvwmm`. L'execució dels diferents processos que impliquen l'arrencada de l'entorn gràfic, i també els fitxers de configuració que es llegeixen durant aquest procés, estan fortament determinats. Així doncs, creant el fitxer anterior, el que hem fet és editar un dels últims fitxers (els que resideixen en el directori arrel de l'usuari) que es llegeixen abans d'entrar en l'entorn gràfic. Aquest fitxer, per tant, ens permet modificar alguns aspectes que s'han determinat per defecte dins del sistema i que estan definits en els fitxers residents a `/etc/X11` i els seus subdirectoris.

Per a finalitzar aquesta secció instal·larem un *window manager* molt utilitzat en el món GNU/Linux, que es caracteritza per la seva versatilitat i el seu escàs consum de recursos: el `WindowMaker`:

```
~# apt-get install wmaker
```

Hem instal·lat ja tres *window managers*, i segurament n'instal·larem més. A part del mètode anteriorment descrit per a preestablir quin volem executar per defecte, podem utilitzar el menú del comandament `update-alternatives` per a establir-ho:

```
~# update-alternatives x-window-manager
```

D'aquesta manera, si l'usuari no especifica el contrari al fitxer `~/.xsession`, tothom utilitzarà aquest *window manager* per defecte.

Lectura complementària

Animem el lector a familiaritzar-se una mica amb aquest *window manager* i a visitar-ne el web per a ampliar coneixements: <http://www.windowmaker.org/>.

11.4. X Session manager

Els *session managers* són programes que es poden fer córrer sobre una sessió gràfica i que ens en permetran establir i modificar paràmetres. *xms* és el *session manager* que ve per defecte amb la instal·lació que hem fet del servidor gràfic. El podem llançar des d'un terminal X (per a llançar una *xterm*, premerem el botó del mig del ratolí i seleccionarem Programs/Xshells/Xterm), mitjançant el comandament `xsm`.

Una vegada llançat `xsm`, mitjançant Checkpoint, podem desar la configuració de la sessió actual (essencialment referent a les aplicacions que tinguem corrent), gestionar els processos que corren mitjançant Client List, consultar el *log* de la sessió o tancar la sessió desant la configuració actual.

A part d'`xsm`, hi ha altres *session managers*. Aquests acostumen a ser una part més dels *desktop managers*, i hi estan tan integrats que de vegades resulta difícil reconèixer-ne les accions. Un exemple típic és la pregunta que se'ns formula sobre si volem desar la sessió en tancar KDE.

11.5. X Display manager

En finalitzar la instal·lació de l'*Xserver*, suggeríem que s'eliminés l'enllaç simbòlic `/etc/rc2.d/S99xdm` per a evitar que en tornar a arrencar el sistema quan entri al *runlevel 2* s'executés `xdm`, acrònim d'*X display manager*. Aquest és el *display manager* que el paquet *X-Window-System* instal·la per defecte. Els *display managers* són els programes encarregats de gestionar qui, des d'on, i com pot entrar un usuari a l'entorn gràfic. Per a llançar-lo, ho farem com amb qualsevol altre DAEMON:

```
~# /etc/init.d/xdm start
```

Per a parar-lo, també utilitzarem el mateix procediment que seguiríem per a detenir qualsevol altre DAEMON, amb l'excepció que hem de polsar la combinació de tecles `Ctrl + Alt + F1`, per a sortir de l'entorn gràfic i situar-nos en la `tty1`, per exemple, en comptes d'utilitzar la combinació que s'usa per a canviar de `ttys` en entorns alfanumèrics:

```
~# /etc/init.d/xdm stop
```

Com hem comprovat, el *display manager* ens demana el nom d'usuari i la contrasenya, les mateixes dades que utilitzem per a accedir al sistema per les tty's, si no és que hi hem imposat alguna restricció. Després de validar-nos, entrem en la manera gràfica de la mateixa manera com ho fèiem mitjançant el comandament `startx`. La diferència resideix en el fet que, quan acabem la sessió gràfica, el servidor no s'atura, sinó que continua corrent l'`xdm`.

Un dels inconvenients d'`xdm` és que no ens permet seleccionar amb quin *window manager* volem treballar. Però existeixen altres *display managers*, com poden ser `wm` (de WindowMaker), `gmd` (del projecte GNOME), o `kdm` (del projecte KDE), que sí que ho permeten.

Podem instal·lar el `wm`, per a veure'n l'aspecte, i per a conèixer un altre *display manager*:

```
~# apt-get install wm
```

En executar-se l'*script* de postinstal·lació, se'ns preguntarà quin *display manager* volem usar, `xm`, que ja el teníem instal·lat, o `wm`. Seleccionarem aquest últim, perquè es creï l'enllaç necessari perquè es llanci `wm` com a *display manager* durant el procés d'arrencada del sistema (si existeix el fitxer `/etc/rc2.d/S99xdm`, és millor esborrar-lo per a evitar missatges de *warning* en arrencar el sistema). Si no volem que s'engegui automàticament cap *display manager* en engegar el sistema, n'hi haurà prou d'eliminar els enllaços necessaris, és a dir, el fitxer `/etc/rc2.d/wm`. L'arxiu `/etc/X11/default-display-manager` marca el *display manager* que s'usa per defecte.

Quan ja s'hagi arrencat una sessió X des del *display manager* –és a dir, una vegada hàgim llançat el pertinent *window maker*–, pot ser interessant executar el comandament `ps tree` per a veure les relacions de dependència entre els diferents processos que corren en aquest moment, juntament amb la informació que ens aportarà la línia `ps aux`.

11.6. Desktop managers

L'aparició de diferents jocs d'eines, i també el desenvolupament de diversos projectes que desplegaven o usaven llibreries de l'entorn

gràfic, va fer aparèixer projectes que busquessin la unificació de tots aquests esforços. Va ser llavors quan va aparèixer un concepte nou en l'entorn X: el de *desktop manager*. Els *desktop managers* són projectes en que es pretén establir les bases per a unificar i estandaritzar, tant la presentació com les polítiques de programació i de desenvolupament d'aplicacions. Un dels primers que va aparèixer va ser el CDE (Common Desktop Manager), encara que actualment els dos projectes més destacats en aquest sentit són GNOME i KDE, als quals, a causa del seu alt grau d'implementació i de desenvolupament, dedicarem una subsecció respectivament. Però abans podem esmentar altres *desktop managers*, com poden ser: GNUStep, ROX, GTK+Xfce o UDE.

11.6.1. GNOME

GNOME és un projecte que forma part de GNU, que es caracteritza pel fet de no necessitar estrictament un *window manager* en concret, encara que es recomana que se n'usi algun que en garanteixi el funcionament correcte (a *GNOME-compliant window manager*) com poden ser: IceWM o Sawfish. Tot i així, per a respectar les preferències i la llibertat de l'usuari, GNOME, en el seu control tauler disposa sempre d'un *window manager selector*, que ens permet escollir quin *window manager* volem usar. GNOME està basat en *Gtk toolkit*, les llibreries pròpies desenvolupades dins del projecte, conegudes com a *gnome-libs específiques*.

Com tots els *desktop managers*, GNOME disposa del seu propi tauler, del seu gestor d'arxius: Nautilus, i del seu control tauler: GNOME Control Panel.

Per a fer una instal·lació bàsica de GNOME, instal·larem el paquet següent juntament amb totes les seves dependències:

```
~# apt-get install gnome-session
```

Tal com s'ha dit, encara que GNOME no exigeixi l'ús de cap *window manager* determinat, es recomana que aquest sigui *GNOME-compliant window manager*. Instal·larem Sawfish, el qual es va desenvolupar

estrictament per a complir aquest requisit. Instal·larem el paquet i totes les seves dependències:

```
~# apt-get install sawfish-gnome
```

Tenim, doncs, un altre *window manager* instal·lat. Detindrem el *display manager* i tornarem a llançar-lo perquè aquest nou *window manager* s'hi integri (GNOME també té el seu propi *display manager*, gdm, que podem instal·lar si volem). Ara tenim dues possibilitats per a aconseguir el nostre objectiu: fer córrer GNOME. La primera és arrencar Sawfish, des del *display manager* o mitjançant `startx` i, una vegada dins, llançar `gnome-session` des d'un terminal X, o bé operar de manera inversa; és a dir, arrencar GNOME pels mateixos procediments que Sawfish, i després llançar `sawfish` des d'un terminal X. Es recomana procedir de l'última manera si volem que la propera vegada que arrenquem GNOME s'executi Swafish (serà el mateix *session manager* de GNOME l'encarregat de fer i registrar els canvis necessaris perquè això s'esdevingui).

Una vegada familiaritzats una mica amb GNOME, el que podem fer és instal·lar alguns paquets que ens puguin ser útils, en concret `gnome-help` i `gnome-terminal`; el primer ens ofereix una interfície on podrem llegir *mans*, fitxers de text en un entorn gràfic, etc., i el segon instal·la l'`xterm` propi de GNOME.

Una aplicació que donarà un aspecte molt més amigable a GNOME serà Nautilus, concebut, en principi, com un gestor de fitxers, però que a més ens permetrà configurar el *desktop*. Per al seu funcionament correcte, Nautilus necessita Mozilla M18 i GNOME Helix. Per tant, els paquets que caldrà instal·lar seran els següents: `nautilus-suggested`, per a fer la instal·lació de Nautilus, i `mozilla`, per a fer una instal·lació completa del navegador (quant a la seva configuració, podem instal·lar les fonts True Type, i podem posar en `auto` el sound DAEMON's dsp wrapper).

GNOME és un projecte en constant desenvolupament i que ofereix una multitud d'aplicacions. Per això, una vegada instal·lat el seu sistema bàsic, suggerim al lector que visiti el seu web `http://www.gnome.org` i que sigui ell mateix qui explori, descobreixi i de-

Nota

En el moment de preparar l'edició catalana d'aquests materials, l'última versió estable de GNOME era la 2.10.

cideixi quina aplicació li pot interessar instal·lar. Només executant la línia següent ja ens podem adonar de la magnitud del projecte:

```
~# apt-cache search gnome
```

Debian Woody integra la versió 1.4 de GNOME. Com que ja està disponible la versió 2.2, proposem modificar el contingut de `/etc/apt/sources.list` afegint-hi les línies següents, per a poder actualitzar o instal·lar de soca-rel els paquets d'aquesta versió "debianitzats":

```
#GNOME2
deb http://ftp.acc.umu.se/mirror/mirrors.evilgeniuses.org.uk/debian/backports/woody gnome2.2/
deb-src http://ftp.acc.umu.se/mirror/mirrors.evilgeniuses.org.uk/debian/backports/woody gnome2.2/
```

Nota

Com que, probablement, els *sources* recomanats en aquestes seccions estaran obsolets quan el lector provi d'utilitzar-los, l'adreçem a l'adreça <http://www.apt-get.org> perquè pugui fer-hi les actualitzacions pertinents.

Una vegada hàgim fet això, n'hi haurà prou d'executar els comandaments següents per a disposar d'aquestes actualitzacions:

```
apt-get update apt-get install gnome
```

11.6.2. KDE

KDE, a diferència de GNOME, sí que necessita un *window manager* concret: es tracta de *kwm*, basat en *Qt toolkit* i en les pròpies *kdelibs*. També disposa del seu *launcher panel* *kpanel*, del seu propi gestor d'arxius *Konquest* i de la seva utilitat de configuració *Control Panel*. Òbviament, KDE pot estar instal·lat en el mateix sistema on hàgim instal·lat GNOME, i fins i tot hi ha aplicacions pertanyents a un *desktop manager* que poden córrer en l'altre. A més, KDE també té el seu propi *display manager* (*kdm*) juntament amb moltes més aplicacions. Novament es recomana al lector visitar el seu web per a conèixer-ne les possibilitats: <http://www.kde.org>, i el web de l'encarregat d'integrar KDE en Debian <http://people.debian.org/ccheney/>. Així mateix, podem executar la línia següent per a veure la integració de KDE en Debian:

```
~# apt-cache search kde
```


Els paquets bàsics de KDE són al paquet `kdebase`. Aquest serà, doncs, el primer que instal·larem:

```
~# apt-get install kdebase
```

Novament haurem de reinicialitzar el nostre *window manager* per a tenir accés al *desktop manager* acabat d'instal·lar. Una vegada fet això, podrem procedir a instal·lar el gestor d'arxius, paquet `konquest`. Mitjançant el paquet `kde-i18-ca` podem instal·lar els fitxers necessaris perquè KDE treballi en català.

A partir d'aquest punt, ja serà cada usuari el que instal·larà els diferents paquets del projecte que li siguin d'interès.

Quant a versions, Debian Woody integra la versió 2. Per a disposar de versions més recents, cal recórrer a paquets no oficials. Si es vol tenir accés a aquests paquets, podem afegir la línia següent a `/etc/apt/sources.list` i actualitzar la base de dades mitjançant el comandament `apt-get update`:

```
#KDE3  
deb http://people.debian.org/~schoepf/kde3/woody ./
```

Tal com ja fèiem, per a preestablir el *window maker* per defecte, utilitzarem el menú del comandament `update-alternatives` per a seleccionar el *session manager*:

```
~# update-alternatives x-session-manager
```

11.7. Personalització d'alguns aspectes

11.7.1. Personalització d'aspectes locals

En general els arxius de configuració es troben en el directori `/etc/X11/0` en alguns dels seus subdirectoris. Personalitzadament, cada usuari pot redefinir els paràmetres de configuració i afegir-ne de nous creant o editant en el seu directori de l'usuari els fitxers que porten el mateix nom que els de configuració general però precedits

d'un ". ". Es podran redefinir o establir tots aquells paràmetres que no requereixin permisos de superusuari, ja que els arxius d'inici es processen després dels de configuració general, i els paràmetres sempre prendran l'últim valor que se'ls assigni.

Xsession

`/etc/X11/Xsession` és un *script* que s'executa en entrar en una sessió d'usuari.

Aquest *script* és el que governa tot el procés d'arrencada de la sessió fins que podem començar a treballar, i també és l'encarregat de gestionar els missatges d'errors que es puguin produir durant aquest procés, els quals es registren a `$HOME/.xsession-errors`.

A `$HOME/.xsession` és on personalitzarem l'arrencada per a un usuari en particular. Així doncs, si volem que el *window manager* sigui *blackbox*, i que s'engegui automàticament *bbkeys* en *background* en iniciar la sessió, aquest contindrà les línies següents:

```
bbkeys &
blackbox
```

Xresources

A l'arxiu `$HOME/.Xresources` personalitzarem l'aspecte de les diferents aplicacions. La sintaxi és *application*parameter: value*. Així doncs, si volguéssim invertir els colors de l'aplicació *xterm*, afegiríem la línia següent al fitxer:

```
Xterm*reverseVideo: true
```

El comandament `xrdb` és l'encarregat de gestionar la base de dades d'*Xresources*. Mitjançant `"xrdb -query"` podem conèixer totes les propietats establertes i el seu valor, i mitjançant el paràmetre `"-display"` obtindrem una llista de tots els paràmetres que accepta el comandament. Si a aquest li passem com a paràmetre la ubicació d'un fitxer, en llegirà totes les definicions de paràmetres.

Xmodmap

El servidor gràfic usa la taula de codis de caràcters per a fer la conversió de senyals provinents del teclat (*server-independent*) a símbols del sistema (*server-dependent*).

La taula de conversió que cal usar ha estat seleccionada durant el procés de configuració del teclat, però el comandament `xmodmap` ens permet modificar el contingut. Un exemple del seu ús pot ser el següent:

```
~# xmosmap -e "keycode 127 = Delete"
~# xmosmap -e "keycode 22 = BackSpace"
```

Mitjançant els paràmetres `“-pk”` `xmodmap` ens tornarà tot el contingut de la taula de conversió que s’usa.

11.7.2. Personalització d’aspectes de xarxa

Els aspectes que es presenten aquí també són d’interès per a un sistema *stand alone*, ja que, com tot el sistema operatiu, el sistema X usa sempre un disseny orientat a xarxa.

\$DISPLAY

La variable `$DISPLAY` serveix per a indicar al client amb quin servidor s’ha de comunicar. La seva sintaxi és la següent:

```
hostname:display number.screen number.
```

Així doncs, si haguéssim definit un altre terminal gràfic al sistema X, afegint la línia següent a `/etc/X11/xdm/Xservers`:

```
:1 local /usr/X11R6/bin/X vt8
```

podríem llançar una aplicació gràfica des d’un `xterm` d’un terminal gràfic a l’altre definint la variable adequadament. Per això, si volguéssim llançar `xeyes` des del primer terminal gràfic, via `xterm`, i “`displayar-lo`” en el segon, procediríem de la manera següent:

```
~$ set DISPLAY :0.1; export DISPLAY
~$ xeyes &
```

Nota

Aquests comandaments es van utilitzar durant molt de temps a causa de la inversió d’assignació de símbols a les taules de conversió; actualment, tanmateix, aquest problema està resolt.

Si entrem en una sessió gràfica, obrim un xterm, canviem d'usuari mitjançant el comandament `su` i provem de llançar una aplicació gràfica, se'ns tornarà un missatge d'error que ens indica que es pot establir connexió amb el servidor. Una estratègia per a evitar aquest problema és utilitzar el paràmetre “-p” perquè s'exporti tot el conjunt de variables d'entorn, i evitar així que el servidor rebutgi la nostra petició de connexió. Aquesta pràctica pot ser molt útil per a llançar programes de configuració que necessitin permisos de *root*, ja que ens evitarà haver d'entrar en l'entorn gràfic com a *root* (pràctica no gaire recomanable, i que, encara que per defecte es permeti, moltes vegades es restringeix manualment).

xhost i xauth

El comandament `xhost` permet establir quins equips poden accedir al servidor gràfic remotament; és a dir, quines màquines client poden llançar una aplicació per ser “displayada” al servidor. La seva sintaxi és la següent: `xhost +hostname`. Si no s'especifica cap *hostname*, qualsevol màquina podrà llançar aplicacions sobre el servidor. Per defecte, no es permet la connexió des de cap equip remot.

El comandament `xauth` serveix per a determinar quins usuaris poden llançar aplicacions sobre el servidor gràfic. Així doncs, mitjançant la combinació d'aquests dos comandaments podrem establir una política de seguretat d'accés al servidor X bastant raonable.

11.8. Configuració d'impressores

La tasca de configurar impressores es pot facilitar a partir de l'entorn gràfic. Existeixen multitud d'aplicacions per a configurar el sistema d'impressió natiu (basat en `lpd` com a servidor i `lpr` com a client), i altres que substitueixen aquest sistema per un de propi, comunament també basat en l'estructura client servidor. En aquesta secció presentarem dues d'aquestes eines: `CUPS` i `LPRng`, ambdós amb el seu sistema propi de servidor d'impressió.

Per a instal·lar LPRng cal instal·lar necessàriament el paquet `lprng`. El paquet que conté la GUI és de configuració del fitxer `/etc/printcap` basat en LPRng és `lprngtool`. Òbviament, per a fer modificacions en aquest fitxer de configuració, caldrà executar el comandament `lprngtool` amb privilegis de `root`. Addicionalment, es recomana instal·lar el paquet `printop`, que aporta una interfície gràfica per a tot el sistema d'impressió, i el paquet `lprng-doc`, el qual conté tota la documentació d'aquest sistema d'impressió.

Per a instal·lar CUPS (Common Linux Printing System) caldrà instal·lar el paquet del servidor d'impressió, `cupsys`; i es recomana instal·lar, juntament amb aquest paquet, el de clients d'impressió, paquet `cupsys-client`. També es pot instal·lar el paquet `cupsys-bsd` per a disposar dels comandaments habituals en el sistema d'impressió de BSD. Pel que fa a programadors de control, els paquets `cupsys-driver-gimpprint` i `cupsomatic-ppd`, n'aporten una gran quantitat, juntament amb nombrosos algorismes de processament d'imatge, definició de mides de paper, etc.

11.9. OpenOffice

En aquesta secció, encara que s'allunyi una mica de l'objectiu principal del curs, presentarem una *suite* ofimàtica que pot ser de molta utilitat per a aquells que estiguin acostumats a utilitzar programari d'aquest tipus. Es tracta d'OpenOffice.org (<http://www.openoffice.org>), projecte derivat de StarOffice, de Sun Microsystems. Cal destacar que aquest projecte és una multiplataforma, i que per tant pot ser implementat en altres sistemes operatius diferents dels de tipus UNIX.

Si bé aquesta *suite* no està integrada a Debian, podem recórrer, com ja hem fet anteriorment, a fonts no oficials per a obtenir-ne els paquets:

```
#OpenOffice
deb http://ftp.freenet.de/pub/ftp.vpn-junkies.de/openoffice/ woody main contrib
```

Una vegada hàgim afegit la línia anterior a `/etc/apt/sources.list` i hàgim actualitzat la base de dades mitjançant `apt-get update`,

n'hi haurà prou d'executar la línia següent per a procedir a la instal·lació:

```
~# apt-get install openoffice.org openoffice.org-l10n-en
```

Només cal tenir present, durant el procés d'instal·lació (el qual s'ha d'executar sota un entorn gràfic), que cal fer la instal·lació per a xarxa. Així, només serà necessari que en el directori d'usuari hi hagi un petit directori on es desin les seves configuracions personals.

Una vegada s'hagi instal·lat el programa la primera vegada, cada usuari haurà d'executar el programa següent perquè es creï el seu directori:

```
/usr/lib/openoffice/program/setup
```

Quan s'hagi fet això i s'hagin respost algunes preguntes, mitjançant el comandament `openoffice` obrirem la *suite*.

11.10. Conclusió

Amb aquest taller s'acaba el material didàctic del segon mòdul del màster. Hi hem après a instal·lar l'entorn gràfic al sistema, que, com ja vam veure en el seu moment, no és una part fonamental dins del sistema operatiu. Però és clara la seva utilitat en molts casos, i no és fins que s'arriba a aquesta part que per a molta gent el sistema operatiu que ha estat objecte d'estudi en aquest mòdul pot esdevenir una alternativa seriosa a altres sistemes. Un cop arribats al final del mòdul, els autors voldríem manifestar el nostre convenciment absolut que GNU/Linux és un sistema operatiu extraordinari, no solament a causa del seu entorn gràfic, que és el que potser sorprèn més a primera vista, sinó per una infinitat d'arguments, d'entre els quals en podem destacar la filosofia, robustesa, adaptabilitat, potència, nivells potencials de seguretat, etc. Estem convençuts que aquest sistema operatiu és una aposta de futur per al qual, si bé ja ha demostrat que és capaç d'obrir-se un espai en el món dels sistemes operatius, només es pot esperar sorpreses positives.

Esperem haver proporcionat els coneixements suficients i haver transmès l'entusiasme necessari perquè el lector iniciï aquí el seu propi camí en el món del programari lliure, i haver sabut obrir-li les portes a una comunitat on tothom és benvingut i respectat.

A. Taules de comandaments

A.1. Sistema de fitxers

Comandament	Significat i paràmetres útils
ls [pattern]	<p>Llista dels continguts d'un directori determinat</p> <ul style="list-style-type: none"> ℘ "-a" mostra tots els continguts, fins i tot els que comencen per "." ℘ "-l" mostra la informació completa relativa als continguts ℘ "-h" acompanyada de "-l" mostra la mida dels arxius en unitats de bytes, KB, MB, GB ℘ "-S" ordena la llista per mida ℘ "-w" mostra la llista per columnes ℘ "-R" mostra la llista recursivament, passant per tots els subdirectoris ℘ "--color" acoloreix el text de la llista segons el tipus d'arxiu
cd [ruta]	Canvia al directori especificat. Si no se n'especifica cap, va al directori de l'usuari
pwd	Mostra la ruta completa fins al directori actual
find [paràmetres] [path]	<p>Busca un fitxer o directori determinat</p> <ul style="list-style-type: none"> ℘ "-iname pattern" cerca recursiva del <i>pattern</i> a partir del directori actual ℘ "-ilname pattern" com a "-iname" però fent la recerca insensible a majúscules o minúscules ℘ "-maxdepth numLevels" fa la recerca fins al nivell de profunditat especificat ℘ "-uid UID" força la coincidència d'UID ℘ "-gid GID" força la coincidència de GID
ln ruta [nomNouEnllaç]	<p>Crea un enllaç a un fitxer o directori. Si no es tracta d'un enllaç simbòlic, es fa una còpia exacta i es modifica davant qualsevol actualització (<i>hard link</i>)</p> <ul style="list-style-type: none"> ℘ "-s" crea un enllaç simbòlic

A.2. Ajuda del sistema

Comandament	Significat i paràmetres útils
man [nummanual] comandament	Mostra el contingut del manual del comandament mitjançant un paginador
info comandament	Una altra font d'ajuda, que de vegades complementa els continguts del <i>man</i> , i en d'altres és l'única documentació que es manté
apropos paraulaClau	Busca tots els comandaments que contenen la paraula clau en la descripció del seu manual

A.3. Permisos dels fitxers

Comandament	Significat i paràmetres útils
chmod mode fitxer	Estableix els permisos de fitxers i/o directoris <i>P</i> "-R" el canvi es fa recursivament; és a dir, es fa sobre tots els fitxers i subdirectoris del directori especificat
chown propietari[.grup] fitxer	Estableix el propietari i el grup (si aquest s'especifica) de fitxers i/o directoris <i>P</i> "-R" el comandament actua recursivament
chgrp grup fitxer	Estableix el grup a què pertanyen fitxers i/o directoris <i>P</i> "-R", el comandament actua recursivament
umask mode	Inicialitza els permisos de tots els fitxers creats a partir del moment en què s'executa el comandament

A.4. Còpia i eliminació de fitxers

Comandament	Significat i paràmetres útils
rm fitxer	El comandament per a esborrar fitxers i/o directoris <i>P</i> "-f" suprimeix els missatges de confirmació <i>P</i> "-R" el comandament actua recursivament <i>P</i> "-i" pregunta abans de procedir a esborrar
rmdir directori	Esborra directoris buits
cp fitxerOrigen fitxerDestinació	Copia arxius de l'origen a la destinació especificats <i>P</i> "-f" no demana confirmació <i>P</i> "-i" demana confirmació abans de copiar cada fitxer <i>P</i> "-l" fa enllaços en comptes de copiar <i>P</i> "-R" actua recursivament copiant els continguts dels subdirectoris <i>P</i> "-s" fa enllaços simbòlics en comptes de copiar
mv fitxerOrigen fitxerDestinació	Mou arxius de l'origen a la destinació especificats <i>P</i> "-f" no demana confirmació <i>P</i> "-i" demana confirmació abans de moure cada fitxer

A.5. Parada o reinici

Comandament	Significat i paràmetres útils
logout	Surt de l'interpret d'ordres i torna a la connexió
halt	Inicia el procés d'aturada del sistema (mai no hem d'apagar l'ordinador fins que no es "display" Power Down)
reboot	Reinicia el sistema, és a dir, procedeix a aturar-lo i després reinicialitza l'ordinador

A.6. Operacions amb fitxers

Comandament	Significat i paràmetres útils
cat [arxiu]	Mostra el contingut de l'arxiu o, si aquest no s'ha especificat, mostra per pantalla tot el que s'entra per teclat P "-n" enumera les línies mostrades
less arxiu	Paginador per a mostrar el contingut d'arxius. Les opcions que s'especifiquen a continuació no són paràmetres del comandament, sinó tecles d'accés directe que s'han d'utilitzar en el seu ús P "/pattern" dins de "less" podem fer una recerca del <i>pattern</i> especificat precedit de "/". Per a continuar la recerca cap endavant o cap enrere, usarem les tecles "n" i "N" respectivament P tecla FIN va al final de l'arxiu P tecla INICIO va al principi de l'arxiu P tecla AVPAG avança pàgina P tecla RETURN avança una línia P tecla SPACE avança una pàgina
more arxiu	Mostra el contingut de l'arxiu d'una manera similar a "less" però més simple (no permet, per exemple, l'opció de retrocedir)
grep [pattern] arxiu	Busca les línies que compleixen el <i>pattern</i> i les mostra per pantalla P "-b" mostra el nombre de bytes de cada línia trobada P "-c" no mostra les línies per pantalla, sinó que les compta i en torna el nombre P "-e pattern" "útil" quan el <i>pattern</i> comença pel caràcter "-" P "-i" ignora la diferència entre majúscules i minúscules P "-n" mostra el número de línia P "-v" inverteix el resultat de la cerca P "-w" selecciona només aquelles línies on el <i>pattern</i> és tota una paraula
cut arxiu	Mostra els camps de l'arxiu P "-d caràcter" indica quin caràcter és el delimitador de camp P "-f numCamp" indica quin camp ha de mostrar P "-b llistaBytes" del camp seleccionat, només es vol veure els bytes indicats, separats per comes
wc fitxer	Mostra el nombre de bytes, línies i paraules del fitxer P "-c" només mostra els bytes P "-l" només mostra el nombre de línies P "-w" només mostra el nombre de paraules
diff arxiu1 arxiu2	Mostra les diferències entre arxiu1 i arxiu2 P "-B" ignora les línies en blanc P "-i" ignora la diferència entre majúscules i minúscules P "-w" ignora els espais en blanc

A.7. Compressió de fitxers i còpies de seguretat

Comandament	Significat i paràmetres útils
tar opcions arxiu [arxiuOrigen]	<p>Uneix diferents arxius i directoris en un de sol (conegut com a <i>tarball</i>, els quals es caracteritzen perquè acaben en <i>.tar</i>)</p> <p>P "-cf" crea un arxiu a partir dels arxius i directoris origen</p> <p>P "-cvf" com en el cas anterior, però mostrant per pantalla el procés</p> <p>P "-cvzf" com en el cas anterior, però comprimint els arxius amb gzip</p> <p>P "-x" extreu el contingut d'un <i>tarball</i></p> <p>P "-xvzf" extreu i descomprimeix mostrant per pantalla els resultats del procés (els arxius comprimits es caracteritzen per acabar en ".tar.gz" o ".tgz")</p>
cpio	Per a fer còpies de seguretat
gzip arxiu	comprimeix l'arxiu i li dona l'extensió ".gz" P "-d" descomprimeix
gunzip arxiu	Descomprimeix l'arxiu
zip arxiuFinal arxiuOrigen	Comprimeix tots els arxius origen en arxiu final (el qual es caracteritza perquè acaben en ".zip") P "-e" xifra l'arxiu mitjançant una paraula clau que es demana en executar el comandament
unzip arxiu	Descomprimeix l'arxiu

A.8. Operacions de disc

Comandament	Significat i paràmetres útils
df	<p>Mostra l'estat de les particions muntades pel que es refereix a espai total, ocupat i lliure</p> <p>P "-h" mostra la informació en unitats de bytes, KB, MB, GB</p> <p>P "-m" mostra la informació en unitats de MB</p>
du [arxiu]	<p>Mostra els blocs que ocupa l'arxiu</p> <p>P "--block-size=MIDA" ens permet establir la mida de bloc que vulguem</p> <p>P "-h" mostra la informació en unitats de bytes, KB, MB, GB</p>
mkfs	Comandament per a crear un nou sistema de fitxers. A causa de la naturalesa del comandament, es recomana la lectura i la comprensió del manual del comandament
dumpe2fs dispositiu	Mostra informació del sistema d'arxius indicat
fsck dispositiu	Fa una revisió del sistema d'arxius i en repara els possibles errors. És important tenir en compte que el dispositiu no ha d'estar muntat en el sistema durant l'ús del comandament
sync	Sincronitza els arxius de memòria i disc dur
badblocks dispositiu	<p>Busca errors en la partició especificada</p> <p>P "-s" es visualitza informació del procés</p>

A.9. Usuaris i grups

Comandament	Significat i paràmetres útils
whoami	Mostra quin usuari ha executat el comandament
groups	Mostra el grup a què pertany l'usuari que ha executat el comandament
who	Mostra els usuaris que estan en el sistema en el moment de l'execució del comandament <i>▶</i> "-T" indica, per a cada usuari, si permet rebre missatges
w [nomUsuari]	Mostra informació extensa sobre els usuaris del sistema en el moment de l'execució. Si no s'especifica un usuari en concret, ens la mostra per a tots
write nomUsuari [consola]	Per a enviar un missatge a l'usuari especificat. Per a determinar la consola, podem utilitzar els comandaments "who" o "w"
talk nomUsuari [consola]	Obre una sessió de xat amb l'usuari especificat si aquest l'accepta
mesg [y-n]	Sense passar paràmetres, mostra si està o no activa la recepció de missatges. El paràmetre "y" l'activa i el paràmetre "n" la desactiva
adduser [nomUsuari]	Crea un usuari nou del sistema
userdel nomUsuari	Elimina l'usuari especificat del sistema
addgroup [nomGrup]	Crea un grup nou
groupdel nomGrup	Elimina un grup del sistema

A.10. Gestió de processos

Comandament	Significat i paràmetres útils
ps	Mostra els processos del sistema <i>▶</i> "-A" mostra tots els processos que s'executen en el sistema (igual que "-e") <i>▶</i> "-H" mostra els processos ordenats per jerarquia <i>▶</i> "-l" mostra informació extensa sobre els processos
top	Mostra els processos en execució interactivament
kill PID	Mana un senyal d'acabament del procés <i>▶</i> "-9" envia el senyal de matar el procés
killall nomProcés	Mana el senyal de final de procés a tots els processos indicats <i>▶</i> "-9" envia el senyal de matar els processos
nice [comandament]	Mostra el nivell de prioritat d'execució que s'utilitza per defecte, i si se li passa com a paràmetre un comandament juntament amb la prioritat que es vol, ajusta la prioritat del comandament a l'especificada <i>▶</i> "-n nouNivell" executarà el comandament amb la prioritat especificada
renice prioritat PID	Canvia la prioritat del PID especificat
at temps	Executa el comandament a l'hora especificat <i>▶</i> "-f comandament" especifica el comandament que s'ha d'executar
atq	Mostra la cua d'espera d'"at"
atrm numComandament	Elimina un comandament de la cua d'espera. El número de comandament és el que torna "atq"
batch	Executa els comandaments o processos quan la càrrega del sistema és baixa. <i>▶</i> "-f comandament" per a especificar el comandament

B. L'editor vi

B.1. Introducció

Saber utilitzar un editor de textos és imprescindible per a poder editar i modificar els fitxers del sistema. Encara que existeixin centenars d'editors diferents, el vi sempre ha estat l'editor per defecte dels sistemes tipus UNIX. Encara que al principi el vi ens pugui semblar un editor molt simple, a mesura que ens acostumem als seus comandaments veurem que té moltes utilitats que ens faciliten molt la manipulació dels fitxers. Encara que per a tasques llargues (com quan programem) existeixin altres editors més útils, la gran majoria d'administradors de sistemes utilitzen el vi per a moltes de les tasques d'administració. El fet que es tracti d'un editor en mode text (que permet la seva utilització a la consola del sistema) i que estigui disponible en tots els sistemes fan del vi l'editor ideal en els entorns UNIX.

Per a cridar el vi podem utilitzar algun dels mètodes que veiem en la taula següent:

<code>"vi arxiu"</code>	Edita el fitxer en manera "full screen"
<code>"vi -r arxiu"</code>	Recupera l'última versió desada del fitxer (per als casos en què no sortim correctament de l'editor i quedi un fitxer de swap)
<code>"vi + arxiu"</code>	Edita el fitxer i se situa en l'última línia
<code>"vi +numLínia arxiu"</code>	Edita el fitxer i se situa en la línia indicada
<code>"vi arxiu1 ...arxiuN"</code>	Edita tots els arxius especificats. Per a saltar d'un a l'altre hem d'escriure en mode comandament ":n". Amb ":n!" no desem les modificacions
<code>"vi +/string arxiu"</code>	Edita el fitxer i situa el cursor en la primera ocurrència de l'"string"

B.2. Modes del vi

El vi té dos modes d'utilització: mode comandament i mode inserció. En el mode comandament tot el que escriguem serà interpretat per l'editor per a fer accions concretes, mentre que el mode inserció s'uti-

Contingut complementari

En GNU/Linux se sol utilitzar més el vim (Vi IMproved), que és 99,9% compatible amb el vi però afegeix unes quantes funcionalitats més.

litz per a modificar el contingut de l'arxiu. Quan entrem en el vi, per defecte estem en mode comandament. Per a canviar a mode inserció, podem utilitzar qualsevol de les tecles de la taula següent:

"a"	Insereix després del caràcter on estem situats
"i"	Insereix abans del caràcter on estem situats
"A"	Afegeix al final de la línia actual
"I"	Afegeix al principi de la línia actual
"R"	Entra en mode inserció reemplaçant caràcters
"o"	Afegeix una línia en blanc sota de la nostra i passa a mode inserció
"O"	Afegeix una línia en blanc a sobre de l'actual i passa a mode inserció

Per a tornar a mode comandament, podem utilitzar la tecla ESC. En mode inserció l'únic que podem fer és escriure text, eliminar-lo o desplaçar-nos amb les tecles AVPAG i REPAG. El mode comandament ens permet moltes més accions. En les taules següents n'especifiquem algunes de les més comunes:

Movent-nos pel fitxer	
"j"	Línia següent
"k"	Línia anterior
"l"	Caràcter següent
"h"	Caràcter anterior
"["	Inci de l'arxiu
"]]"	Final de l'arxiu
"nG"	Anar a la línia "n"
"G"	Anar al final de l'arxiu
RETURN	Línia següent
CTRL+F	Pantalla següent
CTRL+B	Pantalla anterior
CTRL+D	Mitja pantalla següent
CTRL+U	Mitja pantalla anterior

Operacions amb arxius	
" :w "	Desa el fitxer
" :w nomArxiu "	Desa el fitxer amb el nom indicat
" :wq "	Desa el fitxer i surt de l'editor
" :x "	Desa el fitxer i surt de l'editor
" ZZ "	Desa el fitxer i surt de l'editor
" :q "	Surt si no hi ha hagut canvis al fitxer
" :q! "	Surt sense desar els canvis
" :e arxiu "	Edita l'arxiu indicat si no hi ha canvis en l'actual
" :e! arxiu "	Edita l'arxiu indicat i perde els canvis en l'actual, si n'hi ha
" :r arxiu "	Afegeix l'arxiu indicat després de la línia actual
" :Nr arxiu "	Afegeix l'arxiu indicat després de la línia "N"
" :sh "	Executa un intèrpret d'ordres sense sortir de l'editor; per a sortir de l'intèrpret d'ordres hem d'escriure "exit"
" :N, Mw! "	Desa des de la línia "N" a l'"M" i descarta totes les altres
" :N, M>>arxiu "	Afegeix des de la línia "N" a l'"M" l'arxiu indicat
" := "	Mostra la línia actual
CTRL+G	Mostra l'estat del fitxer

Copiar, esborrar, enganxar, buscar i reemplaçar	
" yy "	Copia la línia actual
" Nyy "	Copia les "N" línies a partir del cursor
" p "	Enganxa les línies copiades a sota de l'actual
" P "	Enganxa les línies copiades a sobre de l'actual
" x "	Esborra el caràcter de sota del cursor
" dw "	Esborra la paraula de sota del cursor
" dd "	Esborra la línia actual
" D "	Esborra des de la posició del cursor fins al final de la línia
" /string "	Busca l'"string" a partir de la posició actual. Per a continuar amb la recerca, es pot utilitzar "n" i "N" per a anar cap endavant o enrere respectivament
" ?string "	Com el comandament anterior però en manera invertida
" :set ic "	Fa les recerques ignorant majúscules/minúscules
" :set noic "	Fa les recerques amb majúscules/minúscules
" :g/HOLA/ s/ /ADÉU "	Substitueix tots els "HOLA" per "ADÉU"

En la línia inferior de l'editor veurem els comandaments que escrivim, que s'executaran en prémer el RETURN. A més, la majoria d'aquests comandaments permeten la repetició: tan sols hem d'escriure el nombre de vegades que vulguem que s'executin abans que el comandament. Per exemple, amb "dd" aconseguim que s'esborri la línia actual; si escrivíssim "3dd" en lloc de la línia actual, s'esborrarien les tres següents.

C. Eines d'administració

C.1. Introducció

Quan s'administra un sistema GNU/Linux és necessari conèixer una gran varietat d'aplicacions i programes diferents. Encara que abans d'instal·lar qualsevol aplicació sigui totalment imprescindible llegir-se detingudament la documentació que incorpori, en alguns casos les configuracions poden arribar a ser realment complexes. Per aquest motiu, des de fa ja diversos anys han aparegut eines d'administració més intuïtives que permeten manejar múltiples aplicacions i serveis de manera més amena.

Generalment, aquestes eines globals d'administració incorporen mecanismes per a poder configurar i manejar els aspectes bàsics del sistema i els fitxers de configuració de les aplicacions que utilitzem. Si bé és interessant saber que existeixen aquestes eines, no és recomanable que hi basem tota la configuració d'un servidor per diversos motius. En primer lloc, hem de pensar que aquests programes no sempre tenen en compte totes les possibilitats que els servidors proporcionen. Això pot provocar que deixem sense una configuració adequada alguna opció important per a les nostres necessitats, que no tinguem en compte algun sistema de seguretat, etc. En segon lloc, tampoc no podem oblidar que encara que l'entorn de configuració sigui més amè i, generalment, més fàcil d'utilitzar i manejar, hem de conèixer què és el que realment es fa quan activem les opcions dels diferents programes i serveis que configurem. Encara que l'entorn sigui molt intuïtiu, això no implica que no hàgim de saber què significa exactament cada opció. Si no tenim un coneixement extens del servei que manipulem, és molt fàcil generar errors que poden provocar un mal funcionament del sistema, forats de seguretat, etc. Finalment, un altre motiu per a no utilitzar únicament aquestes aplicacions és que en algun moment el sistema pot tenir errors que no ens permetin utilitzar-les o senzillament que en altres sistemes que hàgim d'administrar no estiguin instal·lades. Si no coneixem amb una mica de detall els fitxers de configuració de les aplicacions que utilitzem, ens

Contingut complementari

Quan instal·lem alguna eina general d'administració, és molt important que en limitem l'ús i accés a només el *root* del sistema; altrament qualsevol usuari podria modificar qualsevol aspecte del sistema. A més, també hem d'estar molt alerta dels forats de seguretat que hi puguin aparèixer, ja que en haver de manejar els programes instal·lats en el sistema, la majoria d'aquestes eines s'han d'executar amb els permisos de *root*, amb el perill que això comporta.

trobarem totalment indefensos davant de qualsevol petit problema que pugui sorgir.

Per tot això, s'han d'utilitzar aquestes eines amb compte i sabent exactament què és el que modifiquen. En alguns casos poden ser molt útils per a veure com podem fer algun tipus de configuració complicada o per a detectar errors que hàgim generat. La manera com les hauríem d'utilitzar ens ha de servir com a complement de la nostra administració, però mai basar-nos-hi totalment.

Totes les distribucions de GNU/Linux acostumen a incorporar les seves pròpies eines automàtiques d'administració. Això és una característica clarament diferenciadora de les diferents distribucions de GNU/Linux. En SuSE, per exemple s'incorpora una aplicació denominada *Yast2*, que ens permet fer gairebé qualsevol operació de configuració del sistema; RedHat incorpora múltiples programes diferents per a configurar la xarxa, els , els servidors d'aplicació, etc.; en instal·lar un paquet en Debian, ja es permet inicialitzar una configuració a partir de les respostes que donem a diverses pantalles de diàleg; algunes aplicacions porten els seus propis *scripts* per a permetre configuracions estàndard més ràpides, etc. Tot i així, si sabem què és el que realment fan aquestes aplicacions i a quins fitxers desentenen la seva configuració, en tenir problemes amb el sistema la seva reparació serà molt més fàcil. A més d'aquestes eines úniques per a la distribució que utilitzem, n'existeixen d'altres de generals que podem instal·lar en la majoria de distribucions existents. Encara que n'hi hagi unes quantes desenes i que cada administrador hagi d'escollir la que més li agradi o s'adapti a les seves necessitats, en aquest apèndix mostrem un parell de les més versàtils i populars: `linux-conf` i `webmin`.

C.2. Linuxconf

L'aplicació d'administració general `Linuxconf` es basa en un entorn de menús de text que podem utilitzar des de qualsevol consola del sistema. En la figura següent podem veure el menú principal:

```

-----Linuxconf-----
This is the main entry to Linux configuration.

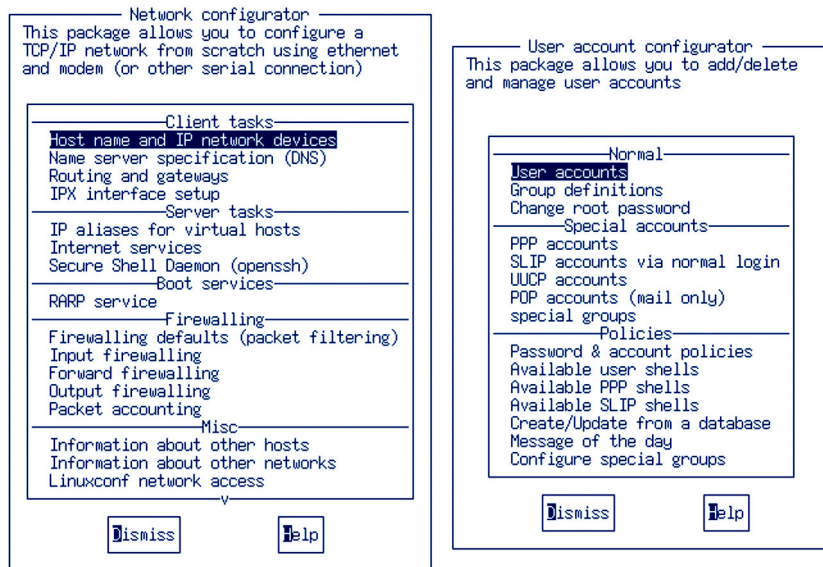
Use the TAB key to navigate between the field
section and the button bar at the bottom.
Check out the help for this screen. It is an
introduction to Linuxconf

-----Config-----
Networking
Users
File systems
Miscellaneous
Peripherals
Boot
cluster administration
-----Control-----
Control panel
Linuxconf management
Date & time
-----Status-----
Logs
System status
-----Tasks-----
Friendly helpers (Gurus)

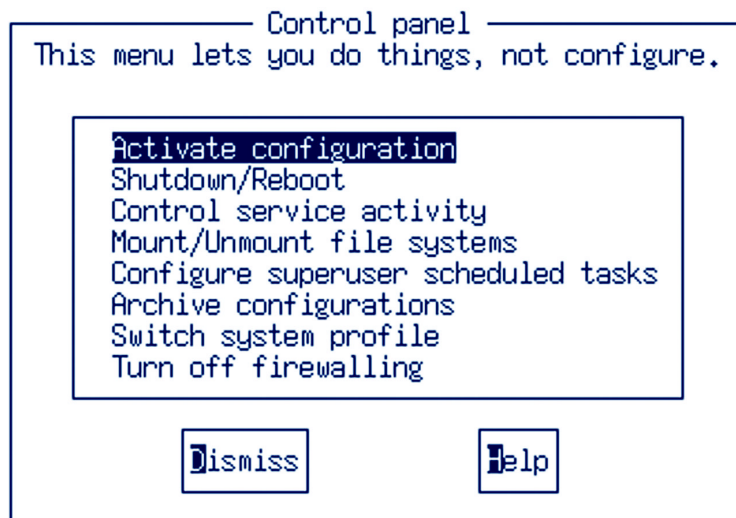
Quit      Help
  
```

Com podem apreciar en la imatge, `linuxconf` divideix les seves operacions en les seccions següents:

- Configuració: aquesta és la secció principal de `linuxconf`, on podem configurar la majoria d'aspectes del sistema com la xarxa, els usuaris, els perifèrics instal·lats, etc. En les figures següents podem veure el diàleg de configuració de la xarxa i d'usuari:



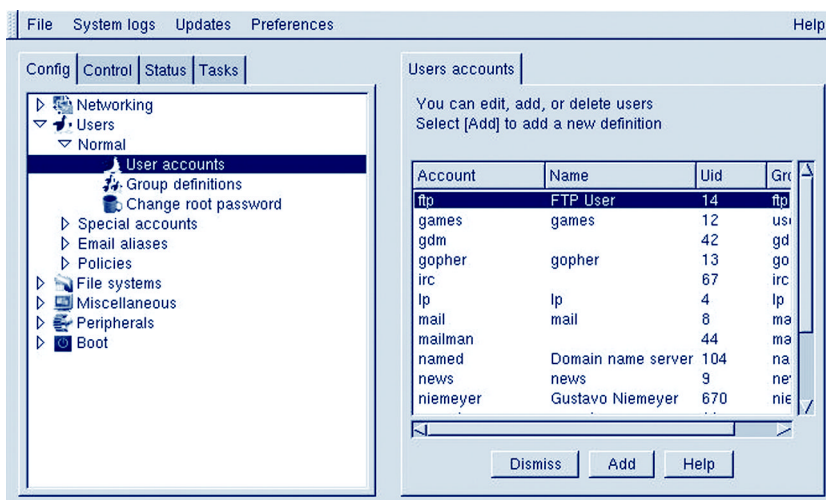
- Control: secció per a fer accions concretes en el sistema com ara muntar o desmuntar unitats, canviar l'hora del sistema, personalitzar els menús, etc. En la figura següent es poden apreciar algunes de les accions del menú de tauler de control:



- Estat: quan vulguem veure els *logs* o l'estat d'algun aspecte del sistema, podem recórrer als menús d'aquesta secció. S'hi utilitzen molts comandaments bàsics del sistema per a veure l'estat del disc, la memòria usada, etc.
- Tasques: altres diàlegs de configuració per a inicialitzar correctament un mòdem, la xarxa, etc.

Una altra manera d'utilitzar aquest programa és a partir d'un navegador web. Per defecte, l'accés via navegador està tancat, per la qual cosa abans d'utilitzar-lo l'haurem d'habilitar a partir del menú de `Networking`, `Linuxconf network access` i activant l'opció d'`Enable network access`. Obrint el navegador i accedint a `http://localhost:98/` tindrem els mateixos diàlegs i opcions del menú de `linuxconf` en format html. Per defecte, només es podrà accedir a aquest servei des de la mateixa màquina, encara que és recomanable activar-lo només quan el vulguem utilitzar.

Finalment, un altre projecte relacionat amb `linuxconf` és el `gnome-linuxconf`, que té les mateixes funcions que vèiem anteriorment però es pot utilitzar en les X. En la figura següent en podem veure l'aspecte:



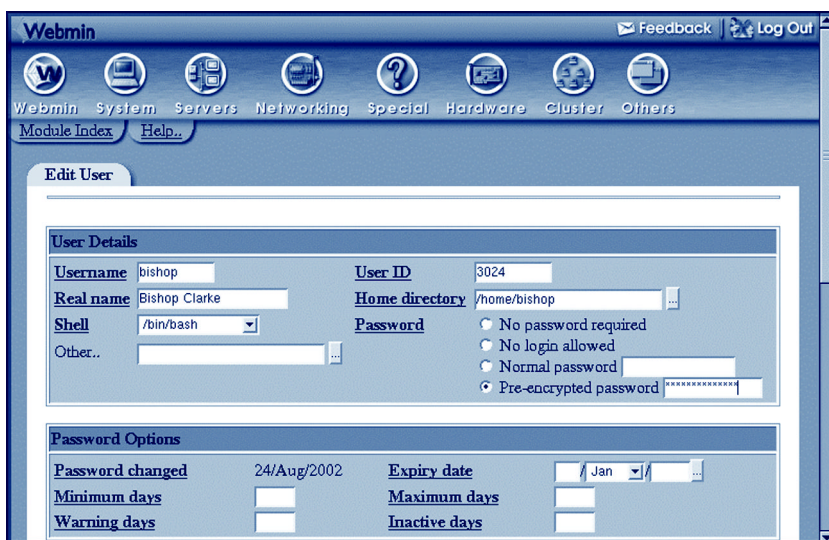
C.3. Webmin

Tal com el seu nom indica, `webmin` és una eina de configuració via web. El seu disseny està molt ben aconseguit i la majoria dels seus mètodes de configuració estan molt ben pensats. Ben utilitzada, és una eina que pot arribar a ser-nos molt útil. Per defecte, en instal·lar-la obre el port 10000 perquè hi puguem accedir a partir d'un navegador qualsevol. Abans d'entrar ens demanarà la contrasenya de l'administrador del sistema, encara que també té un sistema molt útil d'administració d'usuaris propis, per mitjà del qual podem especifi-

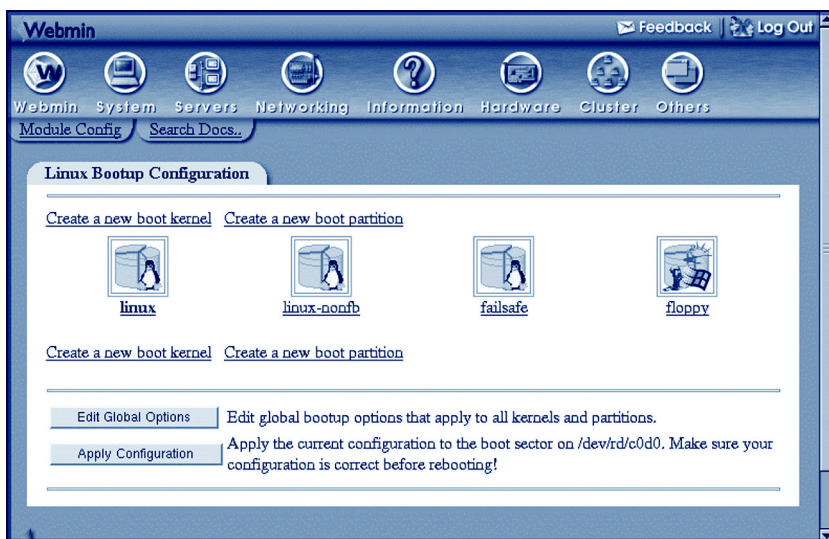
car quines accions pot fer cada un. Aquesta opció és molt interessant perquè permetrà configurar més d'un administrador de sistema cada un especialitzat en unes tasques.

Per a fer-nos una idea de l'aplicació, a continuació mostrem una sèrie de captures de diferents seccions:

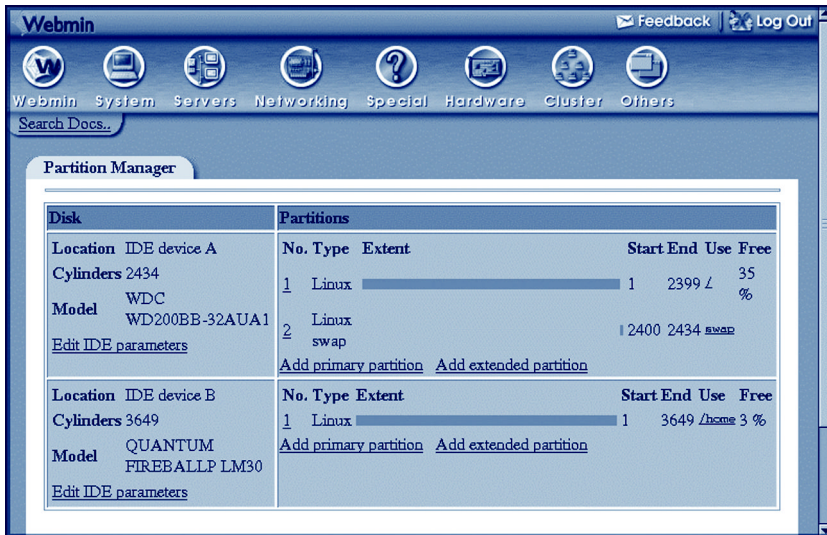
- Administració d'usuaris del sistema:



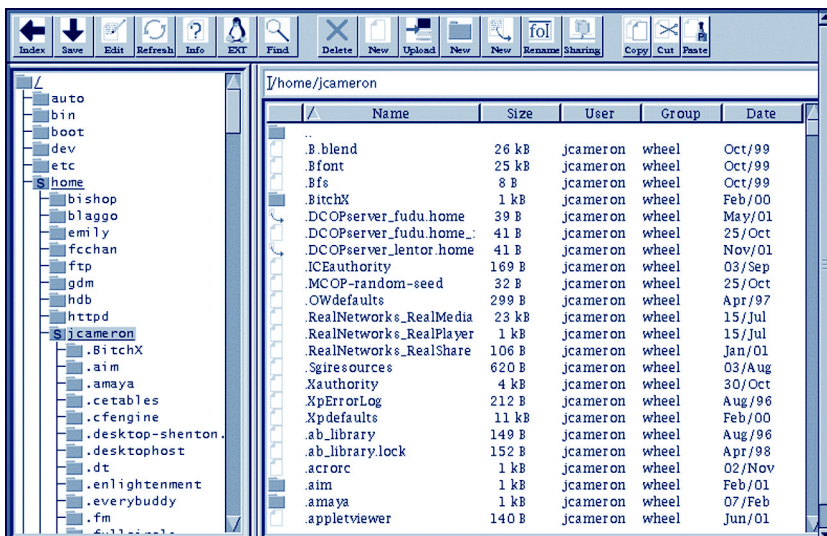
- Configuració d'arrencada del sistema:



- Informació de disc:



- Navegador d'arxius:



GNU Free Documentation License

GNU Free Documentation License
Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent.

An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material.

If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number.

Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit.

When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form.

Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the

translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.



La universidad
virtual

Formació de Postgrau