

GNU/Linux bàsic

P07/M2002/02688

Índex

1. Presentació	7
1.1. Què és el GNU?	7
1.2. Què és GNU/Linux?	9
1.3. Distribucions	11
1.4. Programes i documentació	13
2. Conceptes i ordres bàsics	17
2.1. Introducció	17
2.2. Usuaris i grups	18
2.3. El sistema de fitxers	23
2.3.1. La jerarquia del sistema de fitxers	23
2.3.2. Directoris del sistema	24
2.3.3. Moure'ns-hi	25
2.3.4. Enllaços	26
2.3.5. Permisos	27
2.3.6. Manipulació, patrons i cerques	29
2.3.7. Tipus i contingut de fitxers	31
2.4. Els processos	32
2.5. Altres ordres útils	35
2.5.1. L'ajuda del sistema	35
2.5.2. Empaquetat i compressió	36
2.5.3. Operacions de disc	37
2.6. Operacions amb ordres	39
2.6.1. Readreçaments	39
2.6.2. Ordres específiques del <i>bash</i>	40
2.6.3. <i>Shell scripts</i> amb <i>bash</i>	42
3. Taller de Knoppix	44
3.1. Introducció	44
3.2. Arrencada del sistema	45
3.3. Aturada del sistema	48
3.4. Configuració del teclat	48
3.5. Inspecció del sistema	51
3.6. Maneig de directoris i fitxers	56
3.7. Administració d'usuaris	62
3.8. Gestió de processos	65
3.9. Activació i ús del ratolí	67
3.10. Altres operacions	68
3.11. Conclusió	70
4. Instal·lació de GNU/Linux	71
4.1. Introducció	71

4.2. Arrencada	71
4.3. Particionar el disc	72
4.4. Instal·lació de mòduls	74
4.5. Configuració bàsica de la xarxa	75
4.6. Sistema d'arrencada	76
4.7. Elecció de paquets	76
4.8. Altres aspectes	77
5. Taller d'instal·lació de Debian Etch.....	78
5.1. Introducció	78
5.1.1. Sistemes d'instal·lació	80
5.1.2. Tipus de paquets	82
5.1.3. Estat de desenvolupament dels paquets	83
5.2. Instal·lació de Debian Etch	83
5.2.1. Sabors de Debian Etch	84
5.2.2. <i>Installing Debian GNU/Linux 4.0 For Intel x86.....</i>	84
5.3. Instal·lació de Debian Etch mitjançant DVD-ROM	84
5.3.1. Abans de començar la instal·lació	84
5.3.2. Arrencada del sistema d'instal·lació	86
5.3.3. Configuració de l'idioma d'instal·lació	88
5.3.4. Configuració del teclat	90
5.3.5. Detectar i muntar el CD-ROM	91
5.3.6. Configuració de la xarxa	92
5.3.7. Partició del disc dur	93
5.3.8. Configuració horària	98
5.3.9. Configurar usuaris i contrasenyes	98
5.3.10. Instal·lació del sistema base	102
5.3.11. Configurar el gestor de paquets	103
5.3.12. Seleccionar i instal·lar programes	104
5.3.13. Instal·lació de GRUB	104
5.3.14. Reinicialització del sistema	104
5.3.15. Arrencada del sistema base	104
5.3.16. Configuració d'apt	105
5.3.17. Tasksel	107
5.4. Instal·lació de Debian Etch per xarxa	109
5.4.1. Particularitats d'una instal·lació per xarxa	109
5.4.2. Aspectes comuns dels diferents mètodes d'instal·lació	109
5.4.3. Instal·lació del mòdul de xarxa	110
5.4.4. Configuració de la xarxa	111
5.4.5. Configuració d'apt	111
5.5. Conclusió	112
6. Configuracions bàsiques.....	113
6.1. El sistema de <i>login</i>	113
6.2. Explorar el <i>bash</i>	114
6.3. El sistema d'arrencada	116

6.3.1. Grub	117
6.4. Accés a altres particions i dispositius	120
6.5. Configuració de dispositius	123
6.5.1. El teclat	123
6.5.2. Targeta de xarxa (tipus Ethernet)	125
6.5.3. Targeta Wi-Fi	127
6.5.4. Mòdems	128
6.5.5. ADSL i PPPoE	129
6.5.6. Targeta de so	130
6.5.7. Impressora	130
7. Daemons i runlevels.....	132
7.1. Els <i>daemons</i>	132
7.2. Els <i>runlevels</i>	135
7.3. L'arrencada del sistema	138
7.4. <i>Daemons</i> bàsics	138
7.4.1. <i>Logs</i> de sistema (<i>sysklogd</i>)	138
7.4.2. Execucions periòdiques (<i>cron</i>)	141
7.4.3. Execucions retardades (<i>at</i> i <i>batch</i>)	142
8. Instal·lació d'aplicacions.....	144
8.1. Introducció	144
8.2. El sistema de paquets Debian	145
8.3. Compilació de nous programes	148
9. Taller de configuracions bàsiques.....	151
9.1. Introducció	151
9.2. El gestor d'arrencada	151
9.2.1. Instal·lació de Grub	152
9.3. El sistema de paquets	154
9.3.1. <i>/etc/apt/sources.list</i>	154
9.3.2. <i>apt</i>	156
9.3.3. <i>dpkg</i>	160
9.3.4. <i>dselect</i>	160
9.3.5. <i>aptitude</i>	160
9.4. <i>locales</i> : configuració regional	161
9.5. L'arxiu principal d'arrencada, <i>/etc/inittab</i>	161
9.6. Muntatge de dispositius, <i>/etc/fstab</i>	162
9.7. Configuració de dispositius	163
9.7.1. Configuració del ratolí	164
9.7.2. Configuració de mòdems	166
9.7.3. Configuració de mòdems DSL	167
9.7.4. Configuració de targetes de xarxa	168
9.7.5. Configuració d'impressores	171
9.7.6. Configuració de targetes de so	172
9.8. Conclusió	173

10. Arquitectura X-Window	174
10.1. Què és X-Window?	174
10.2. Configuració	179
10.3. <i>X display manager</i>	182
11. Taller d'X-Window	185
11.1. Introducció	185
11.2. Instal·lació i configuració del servidor X	185
11.2.1. Diferents estratègies per a la instal·lació dels paquets ...	185
11.2.2. Instal·lació de paquets bàsics	186
11.3. Configuració d'X	188
11.3.1. Secció "Device"	189
11.3.2. Secció "Monitor"	189
11.3.3. Secció "Screen"	190
11.3.4. Secció "InputDevice"	191
11.3.5. Secció "ServerLayout"	193
11.3.6. Secció "DRI"	193
11.3.7. Secció "Files"	193
11.3.8. Inicialització del servidor	193
11.3.9. El fitxer de <i>log</i>	194
11.4. <i>Window managers</i>	194
11.5. X Session manager	196
11.6. X Display manager	196
11.7. <i>Desktop managers</i>	197
11.7.1. GNOME	198
11.7.2. KDE	199
11.8. Personalització d'alguns aspectes	199
11.8.1. Personalització d'aspectes locals	200
11.8.2. Personalització d'aspectes de xarxa	201
11.9. Configuració d'impressores	202
11.10. OpenOffice	205
11.11. Conclusió	206

1. Presentació

1.1. Què és el GNU?

Per a entendre tot el moviment del programari lliure, ens hem de situar al final de la dècada dels seixanta, començament de la dels setanta. En aquells temps les grans companyies d'ordinadors no atorgaven el valor que avui dia es dona al programari. La gran majoria eren fabricants d'ordinadors que obtenien els seus principals ingressos venent les seves grans màquines, a les quals incorporaven algun tipus de sistema operatiu i aplicacions. Les universitats tenien permís per a agafar i estudiar el codi font del sistema operatiu amb finalitats docents. Els mateixos usuaris podien demanar el codi font de *drivers* i programes per tal d'adaptar-los a les seves necessitats. Es considerava que el programari no tenia valor per si mateix si no era acompanyat pel maquinari que el suportava. 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 operatius incompatibles amb els altres), ja que es va convertir en el factor que li va proporcionar molta popularitat.

Stallman, sense *drivers*

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 *drivers*. Ell només volia modificar-los perquè l'avisés automàticament quan s'encallava el paper! La companyia es va negar a proporcionar-los-hi.

A poc a poc, les grans empreses van començar a prendre consciència del valor del programari: primer va ser IBM la que el 1965 va deixar de donar el codi font del seu sistema operatiu, al final de la dècada 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 de les empreses va començar a posar reticències a deixar el codi font dels seus programes i sistemes operatius, i va començar a vendre els seus programes com un valor afegit al seu maquinari. En aquest entorn cada vegada més tancat, Richard Stallman (que treballava al MIT, Institut Tecnològic de Massachusetts) es va sentir indignat 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 llavors.

A partir d'aquell moment, Stallman va decidir ser conseqüent amb els seus ideals i 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 aplicacions iniciant un projecte denominat GNU.

D'especial interès per a entendre els motius que van portar Stallman a iniciar GNU és el seu primer manifest, el document en què 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 amb ell. 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 ha de ser gratuït necessàriament. 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 els usuaris a no distribuir el programari que els hem venut. Aquest ha de poder ser distribuït de manera lliure. 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 el 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 llibertat 2 és la que permet distribuir lliurement còpies del programari, i ajudar 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, així mateix, 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 GPL (General Public License), amb la qual s'ha protegit tot aquest tipus de programes. Aquesta llicència posa per escrit les idees anteriorment comentades.

El projecte va començar a produir programari a partir de 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 és un procés llarg i complex, des del principi molts programadors i desenvolupadors de programari es van veure captivats per la idea de Stallman i van començar a col·laborar-hi de manera gratuïta. La comunitat no va parar de créixer, i a poc a poc va començar a disposar de les eines necessàries (editors, compiladors, etc.) per a implementar el nucli del sistema operatiu, que era la tasca que requerien les eines que s'estaven desenvolupant. 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 els seus 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 queda per finalitzar definitivament i el nucli del GNU, anomenat Hurd, roman en fase de desenvolupament.

Kernel

Com el seu nom indica, el nucli (*kernel*) d'un sistema operatiu és el cor amb el qual pot funcionar. És el nucli del 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 per a tots els programes, es comunica amb els dispositius d'emmagatzematge per a desar els arxius, etc.

Activitat

1.1. Llegiu el primer missatge escrit per Stallman el 1983 que anuncia el seu projecte (traduït al castellà): <http://www.fsf.org/gnu/initial-announcement.es.html>.

Activitat

1.2. Llegiu el manifest GNU original de Stallman (traduït al castellà).

Activitat

1.3. Llegiu la General Public License.

1.2. Què és 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 al moment havia pogut utilitzar el codi font de l'UNIX d'AT&T perquè els seus alumnes aprenguessin a dissenyar sistemes operatius. La seva idea era escriure un sistema operatiu que pogués ser estudiat i modificat per qui volgués fer-ho. El 1987 es va posar mans a l'obra

i va anomenar el seu projecte mini-UNIX, que va donar lloc a **MINIX**. En no utilitzar ni una sola línia de codi de l'UNIX d'AT&T, no hi ha cap restricció a agafar el codi, utilitzar-lo i modificar-lo lliurement.

Tecnologia *microkernel*

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

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 fàcil comprensió, i aportant una tecnologia molt nova per a l'època que li permetia versatilitat, multiplataforma, etc. Aquest ha estat un dels punts forts i febles alhora del MINIX: encara que el sistema és una petita joia per al seu estudi i disseny, és molt probable que mai no es pugui utilitzar en entorns reals. Es va optar per fer-lo entenedor, modular i molt pedagògic, però no ràpid. De tota manera, Tanenbaum tampoc no pretenia això; al llarg dels anys MINIX ha anat evolucionant i realment avui en dia encara segueix existint i l'estudien molts alumnes d'universitats de tot el món.

Aquí és quan entra en joc **Linux**. Mentre l'FSF seguia amb el seu gran projecte proporcionant eines per a la construcció d'un sistema operatiu, Tanenbaum orientava MINIX a finalitats docents i moltes empreses continuaven fent evolucionar les seves pròpies versions d'UNIX. **Linus Torvalds**, estudiant de la Universitat d'Hèlsinki, decideix crear l'agost de 1991 el seu propi nucli per a un nou sistema operatiu, Linux. La seva idea era crear un UNIX per a PC a fi que 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 el qual 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 en el qual va aparèixer aquest primer missatge, veuríem com, ràpidament, gent de tot el món es va començar a interessar per aquest nou sistema, que en utilitzar el compilador i intèrpret d'ordres 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 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.

Kernel monolític de Linux

Linux, el nucli de GNU/Linux, és de tipus monolític. Això indica que no se separen les seves diferents funcionalitats en diferents mòduls, sinó que tot forma 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.

De fet, en els primers anys de la seva existència, GNU/Linux s'identificava com el sistema operatiu dels *hackers*. La seva difícil instal·lació, manipulació i falta de *drivers* el feien una eina apta únicament per a gent molt entesa en el tema. Van ser aquests primers usuaris els que van dissenyar els *drivers* per als discos, impressores, targetes, etc. i els que van començar a donar a conèixer 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

En l'actualitat, hi ha 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 o *pocket PC* o PDA, punts d'accés de xarxes sense fil, etc. La naturalesa del programari lliure permet això: qualsevol pot prendre el codi desenvolupat fins al moment 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 tots maneres, encara que hi ha desenes de distribucions, n'hi ha algunes de més populars que s'han estès molt. La filosofia de programari lliure provoca 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 aporten beneficis, la qual cosa permet la seva subsistència. Així mateix, es pot considerar que en moltes d'aquestes distribucions s'inclou programari de propietat 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:

Linux i el projecte GNU

Tot i que moltes distribucions de GNU/Linux es denominen només Linux, és important que diferenciem que realment Linux és el nucli del sistema operatiu i que el projecte GNU és el que realment ha aportat molta de l'estructura per al seu funcionament.

- Slackware: una de les primeres distribucions que van sorgir. 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 va aparèixer i encara continuen existint i evolucionat. El sistema de paquets ens permet diferenciar clarament el programari lliure del que no ho és, la qual cosa ens permet disposar de tot el sistema només amb programes de llicència Free Software. És desenvolupada per 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 de més popularitat. És creada per 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. És desenvolupada per una empresa alemanya i aporta molt programari de propietat 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 *live-CD* basada en Debian. Detecta automàticament tot tipus de maquinari i aporta l'últim escriptori de KDE i la *suite* OpenOffice.org. És molt útil per a demostracions i usuaris novells en el sistema.



- Ubuntu: és una distribució Linux que ofereix un sistema operatiu predominantment enfocat a ordinadors d'escriptori, encara que també proporciona suport per a servidors. Basada en Debian GNU/Linux, Ubuntu concentra el seu objectiu en la facilitat d'ús, la llibertat en la restricció d'ús, els llançaments regulars (cada sis mesos) i la facilitat en la instal·lació. Existeix tant en format *CD-live* com en format instal·lable. Detecta automàticament tot tipus de maquinari (fins i tot el més modern).



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

Activitat

1.4. 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. Per aquesta raó, 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 pot 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 anem introduint en el món del programari lliure i del GNU/Linux, veurem com 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 donar voltes sobre com resoldre'l, hem de pensar que és molt probable que una altra persona com nosaltres s'hagi trobat amb el mateix problema o amb un problema similar. Buscar i trobar la documentació que s'adapti millor als problemes que se'ns vagin plantejant 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
 - The Linux Documentation Project. La majoria de les guies, HOWTO, PMF, etc. existents les podem trobar en aquest lloc, que a més està en diversos idiomes. <http://www.tdlp.org>
 - LinUx en CASTellano. Gran projecte de documentació en castellà per als HOWTO, guies, etc. de GNU/Linux. <http://lucas.linux.org.mx>
 - El HOWTO dels HOWTO. <http://lucas.linux.org.mx>
 - Linux.com. Pàgina amb diferents seccions de notícies, documentació, etc. <http://www.linux.com>
 - Documentació per a Debian GNU/Linux. <http://www.debian.org/doc/>
- Notícies
 - Slashdot. Notícies i rumors del món GNU/Linux. En anglès. <http://slashdot.org>
 - Barrapunto. La rèplica de Slashdot en castellà. <http://barrapunto.com>
 - Puntbarra. La rèplica de Slashdot en català. <http://puntbarra.com>
 - Bulmalug. Usuaris novells de Linux de Mallorca i voltants. Notícies i seccions dedicades a temes concrets. <http://bulmalug.net>

- Notícies de GNU en castellà. <http://www.es.gnu.org>
- Linuxtoday. Una altra pàgina de notícies molt pràctica per a estar a l'última moda. <http://www.linuxtoday.com>
- Libertonía. Pàgina de notícies. D'especial interès és la seva secció "Fonts de notícies", en què hi ha multitud d'altres enllaços a altres pàgines del mateix estil. <http://libertonía.escomposlinux.org>
- Fòrums
 - Foroslinux.org. Diversos fòrums de GNU/Linux dedicats a tot tipus de temes.
 - Fòrums de Linux Security. Fòrums centrats en temes de seguretat i similars.
- Cerca
 - Linux en Google. El cercador més gran del món també per a GNU/Linux. <http://www.google.com/linux>
 - Buscadoc. Cercador de documentació informàtica en castellà.
- Distribucions
 - La pàgina oficial de la Free Software Foundation. <http://www.fsf.org>
 - Pàgina oficial de Debian GNU/Linux. <http://www.debian.org>
 - Pàgina oficial de RedHat Linux. <http://www.redhat.com>
 - Pàgina oficial de SuSE. <http://www.novell.com/es-es//linux/>
 - Pàgina oficial de Slackware Linux. <http://www.slackware.com>
 - Pàgina oficial de Knoppix. <http://www.knoppix.com>
 - Pàgina oficial d'Ubuntu. <http://ubuntu.org>
- Baixades
 - Sourceforge. La major pàgina amb projectes de programari lliure. <http://sourceforge.net>
 - Linux en Softonic. Secció de baixada per a GNU/Linux d'una de les múltiples pàgines de *downloading*. <http://www.softonic.com>
 - Download. Pàgina de baixades. <http://www.download.com>
- Altres

- Linux Security. Pàgina molt actual centrada en tot tipus de temes de seguretat en GNU/Linux. <http://www.linuxsecurity.com>
- LinuxHQ. Informació general sobre distribucions de GNU/Linux, seguretat, etc.
- Linux Journal. Pàgina de notícies i articles sobre GNU/Linux. <http://www.linuxjournal.com>
- *Linux Gazette*. Revista de GNU/Linux. <http://www.linuxgazette.com>
- *Linux-mag*. Revista de GNU/Linux. <http://www.linux-mag.com>
- Pàgina oficial del projecte XFree86. <http://www.xfree86.org>

2. Conceptes i ordres bàsics

2.1. Introducció

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

La majoria de les ordres que veurem en aquest apartat formen part de l'estàndard (normes IEEE POSIX) i són comunes 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'elles també es poden efectuar amb les ordres que veurem. A partir d'aquestes, podrem manipular gairebé tots els aspectes del sistema i moure'ns-hi eficientment. Aprenent a utilitzar correctament aquestes ordres, aprendrem a navegar per qualsevol sistema basat en GNU/Linux, sense que importi quina distribució estiguem usant.

Cadascuna de les ordres del sistema sol tenir multitud de paràmetres diferents. Amb la utilització dels paràmetres podem, amb una mateixa ordre, fer moltes accions diferents, encara que totes siguin d'un mateix estil. En aquest document no especificarem els diferents paràmetres de cadascuna de les ordres 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 cadascuna. Totes elles disposen d'un ampli manual, en el qual s'especifiquen totes les seves opcions, de manera que sempre que necessitem fer alguna acció en concret hi podrem recórrer. En els tallers distribuïts al llarg del curs sí que veurem algunes d'aquestes opcions, encara que és important saber que amb el manual sempre podrem descobrir moltes altres que ens poden ajudar a fer tot el que necessitem.

Paràmetre d'una ordre

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

Ordre

Una ordre és un programa que fa una determinada acció relacionada amb el sistema operatiu.

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 de manera simultània a d'altres, i executar una o més tasques alhora. 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 cadascú, 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 *login* i una contrasenya. El *login* sol ser un nom que identifica de manera inequívoca l'**usuari**. En sistemes en què 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, números i caràcters especials. No ha de ser 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 significa que la nostra contrasenya no és emmagatzemada com a text, sinó que és xifrada i guardada tal com és. Quan entrem en el sistema i escrivim la nostra contrasenya, aquesta és xifrada i és comparada amb la que hi ha emmagatzemada. Si coincideixen, la identificació és positiva, si no coincideixen, no hi ha identificació. 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'única cosa que poden fer és xifrar paraules a partir de diccionaris (amb sistemes automàtics per a derivar-les i buscar-ne variants) i provar si coincideixen amb el xifratge d'alguna de les contrasenyes d'usuari. Per aquest motiu, hem d'escollir acuradament les nostres contrasenyes; altrament comprometrem tota la seguretat del sistema.

Actualment, en els sistemes GNU/Linux podem triar dos tipus de xifratge possibles per a les contrasenyes d'usuari. El que s'utilitza des dels inicis d'UNIX és el 3DES. L'únic inconvenient d'aquest tipus de xifratge és que només ens permet contrasenyes de 8 lletres (si n'escrivim més, s'ignoren), a diferència de l'altre tipus de xifratge, anomenat MD5, amb el qual 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 de manera unidireccional). Com més llarga sigui la contrasenya, resulta més segura, amb la qual cosa, es recomana utilitzar el segon tipus de xifratge. De totes maneres, 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.

Política de noms

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

NIS

NIS són una sèrie d'aplicacions que ens permeten gestionar tots els usuaris d'una mateixa xarxa de manera centralitzada en un sol servidor.

Si bé un usuari és un individu particular que pot entrar en el 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-los donar una sèrie de permisos especials en el 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 un *login* normal) al sistema.

En tot sistema operatiu hi ha d'haver un superusuari (*root*). Aquest té privilegis màxims que permetran que efectuï 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 i utilitzarem 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 *home*, etc.) de l'usuari.
- `/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 en el mateix fitxer de `passwd`, però, per raons de seguretat (molts mecanismes han de poder llegir aquest fitxer, amb la qual cosa era molt fàcil fer-se amb ell 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 sol ser recomanable utilitzar-la. Tots aquests fitxers estan organitzats per línies, cadascuna de les quals 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)*Login*: el nom de l'usuari. No hi pot haver dos noms iguals, encara que sí algun que coincideixi amb un grup del sistema.

Servidor

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.

Configuració de contrasenyes

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-hi 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.

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 de ser també en el de `shadow` i en aquest camp es posa el caràcter "x".

3) 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 és reservat per al `root`.

4) Group ID: el número de 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 home: el directori home de l'usuari és on aquest pot desar tots els seus fitxers. Se solen posar tots en alguna carpeta del sistema (generalment `/home/`) i organitzats per grups.

7) Intèrpret d'ordres: un intèrpret d'ordres (*shell*) és un programa que s'encarrega de llegir tot el que escrivim en el teclat i executar els programes o ordres que hi indiquem. N'hi ha desenes, encara que la més utilitzada és, sens dubte, el `bash` (GNU Bourne-Again Shell). Si en aquest camp escrivim `/bin/false` no permetrem que l'usuari executi cap ordre 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 és reservat per al grup del `root` (els administradors).

"Violar" una contrasenya

"Violar" 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 utilitzar grans supercomputadors).

4) Llista d'usuaris: els noms dels usuaris que pertanyen al grup, separats per comes. Encara que tots els usuaris han 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 s'està referenciant.

- shadow

1) *Login*: ha de ser el mateix nom que s'utilitza en el fitxer de `passwd`.

2) Contrasenya xifrada.

3) Dies que han passat, des de l'1 de gener de 1970, fins que la contrasenya ha estat canviada per última vegada.

4) Dies que han de passar fins que la contrasenya es pugui canviar.

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 de 1970, des que el compte és deshabilitat.

9) Camp reservat.

Quan un usuari entra en el sistema, se'l situa en el seu directori *home* i s'executa l'interpret d'ordres (*shell*) configurat. D'aquesta manera, ja pot començar a treballar. Només el *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. Hi ha moltes ordres per a manipular tot això. Cadascuna d'elles té, a més, diversos paràmetres diferents per a gestionar tots els camps que hem vist anteriorment de manera amena. A continuació, mostrem algunes d'aquestes ordres:

- `adduser`: ens serveix per a afegir un nou usuari al sistema. La manera en què aquest s'afegeix (si no li especifiquem res) es pot configurar en el fitxer `/etc/adduser.conf`. S'hi poden passar multitud d'opcions diferents per a especificar el directori *home*, el *shell* que cal utilitzar, etc.
- `useradd`: crea un nou usuari o en canvia la configuració per defecte. Aquesta ordre i l'anterior ens poden servir per a efectuar les mateixes accions.

Dates en sistemes UNIX

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

Quota de disc

En sistemes en els quals hi ha centenars d'usuaris, és freqüent posar algun tipus de mecanisme per a restringir l'espai de disc que en pot utilitzar cadascun. En els sistemes GNU/Linux aquest sistema s'anomena quota.

- `usermod`: amb aquesta ordre podem modificar la majoria dels camps que es troben en el fitxer de `passwd` i `shadow`, com el directori `home`, el `shell`, 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 el `shell` de l'usuari.
- `deluser`: elimina un usuari del sistema, i esborra o desa tots els seus fitxers segons els paràmetres que li passem, i en fa còpia de seguretat o no, etc. La configuració que s'utilitzarà per defecte amb aquesta ordre s'especifica en el fitxer `/etc/deluser.conf`.
- `userdel`: ordre 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 determinat compte.
- `addgroup`: permet afegir un grup al sistema.
- `groupadd`: el mateix que l'ordre anterior, però amb diferents paràmetres.
- `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 l'ordre `whoami`, que ens mostrarà el nostre `login`. `groups` ens serveix per a saber a quins grups pertanyem i `id` ens mostrarà usuari i grups. També és interessant poder-nos convertir en un altre usuari sense haver de sortir de la sessió (ordre `login` o `su`) o canviar-nos de grup amb l'ordre `newgrp`. Aquesta última ordre l'hem d'utilitzar només quan no pertanyem al grup en qüestió i sabem la seva contrasenya (que ha d'estar activada en el fitxer de `group`). Si només necessitem els permisos del grup en qüestió per a executar una ordre determinada, també podem utilitzar `sg`.

Versatilitat de GNU/Linux

Com veiem, en GNU/Linux tenim més d'una manera per tal de dur a terme 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 algunes de les ordres que existeixen, crear-los nosaltres mateixos, etc. En definitiva, tenim la possibilitat de triar què és el que més ens agrada.

Tal com dèiem anteriorment, GNU/Linux és un sistema operatiu multiusuari, per la qual cosa en un mateix moment hi pot haver diversos usuaris connectats al sistema de manera simultània. Per a saber quins usuaris hi ha en un moment determinat, podem utilitzar l'ordre `who`, que ens mostra la llista d'usuaris dins del sistema. `w`, a més, ens mostra què és el que estan fent. Ens hi podem comunicar utilitzant l'ordre `write`, amb la qual apareix el missatge que haguem escrit a la pantalla de l'usuari indicada o `wall`, que escriu el contingut del fitxer que haguem especificat a tots els usuaris dins del sistema. Per a activar o desactivar l'opció de rebre missatges tenim l'ordre `mesg`. També podem fer un xat personal amb algun usuari a partir de l'ordre `talk`.

2.3. El sistema de fitxers

2.3.1. La jerarquia del sistema de fitxers

Tot sistema operatiu necessita guardar 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 i el caracteritza 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 hi ha, 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` és l'evolució de l'anterior i incorpora una tecnologia denominada *journaling*. Un dels principals avantatges d'aquesta tecnologia és que si hi ha un tall en el subministrament d'energia 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`.

Sistema de fitxers

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'emmagatzematge i la manipulació dels arxius. Són les funcions que tracten amb els dispositius físics d'emmagatzematge de l'ordinador, com el disc dur.

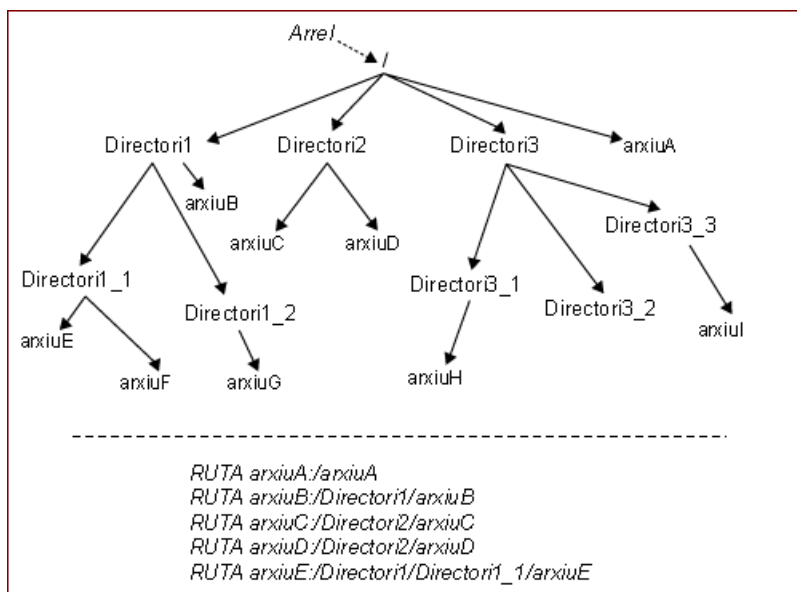
Una característica molt important de tots els sistemes operatius basats en UNIX és que tots els dispositius del sistema es poden tractar com si fossin fitxers. Igualment, quan vulguem accedir al contingut d'un CD, disquet o qualsevol altre dispositiu d'emmagatzematge, l'hauréu de muntar en un directori 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).

Sistema de fitxers ext2

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

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'hi ha una. Per a organitzar els fitxers adequadament, el sistema proporciona el que denominarem directoris (o carpetes), dins dels quals podem posar arxius i més directoris. D'aquesta manera, aconseguim una organització jeràrquica com la que veiem en la figura següent:

Figura 2.1



2.3.2. Directoris del sistema

La majoria dels sistemes operatius del mercat segueix l'estàndard FHS, en el qual s'especifiquen les principals característiques que hauria de tenir qualsevol sistema operatiu. Entre elles es troba la distribució en directoris que hem d'efectuar dels nostres arxius per tal de tenir-los organitzats correctament i de poder-los localitzar de manera ràpida i senzilla. En la majoria de les distribucions basades en GNU/Linux se segueixen aquestes recomanacions i es troben els directoris principals següents:

- `/bin/`: ordres bàsiques per a tots els usuaris del sistema.
- `/boot/`: arxius estàtics necessaris per a l'arrencada del sistema.
- `/dev/`: dispositius del sistema.
- `/etc/`: arxius de configuració del sistema i de les aplicacions que hi ha instal·lades.

- `/home/`: directori per a posar les carpetes home dels usuaris.
- `/lib/`: biblioteques 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 home per al *root* del sistema.
- `/sbin/`: ordres especials per al *root* del sistema.
- `/tmp/`: arxius temporals. Segons la distribució utilitzada (o la configuració que utilitzem) s'esborren en arrencar 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 o *spoolers* d'impressió, fitxers de *log*, etc.

És molt recomanable conservar aquests directoris i no eliminar-ne cap (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 de noves aplicacions necessiten que hi hagi l'organització donada i molts dels arxius de configuració dels programes han de ser en determinats directoris. El que sí que podem fer sense cap tipus de restricció és crear nous directoris en l'arrel del sistema o en qualsevol altra carpeta.

2.3.3. Moure'ns-hi

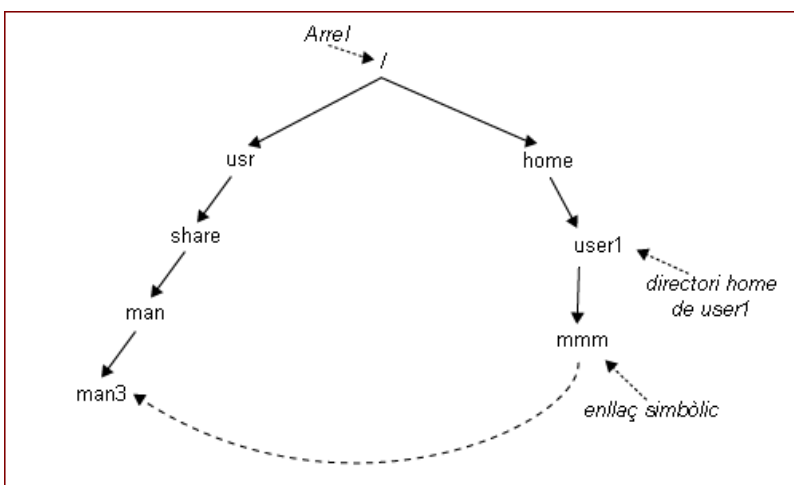
Per a moure'ns per l'estructura de directoris hem d'utilitzar les ordres per a fer una llista dels continguts i canviar de carpeta. Quan entrem en el sistema, és usual que el *login* ens situï en el nostre directori home, que generalment se sol referenciar amb el caràcter `~`. Si volem veure el que hi ha en el directori en el qual estem situats, podem fer una llista dels continguts utilitzant l'ordre `ls`. Hem de tenir en compte que per defecte l'ordre 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 som 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 l'ordre `cd`. Si no li passem cap paràmetre, per defecte ens situarà en el nostre directori home. Generalment, se li sol indicar on volem anar, passant-lo de manera absoluta o relativa. De manera relativa significa que partirem del directori on som en el moment d'executar l'ordre. Per exemple, si som en el directori `/usr/bin/` i volem anar al `/root/`, hauríem d'introduir l'ordre 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 manera absoluta sempre partim de l'arrel, de manera que l'ordre que utilitzaríem per a l'exemple anterior seria: `cd /root`. Per a saber en quin directori som, podem utilitzar l'ordre `pwd`.

2.3.4. Enllaços

Uns altres mecanismes que ens proporcionen la gran majoria de sistemes de fitxers són els que denominem enllaços. Un enllaç és un pont a un arxiu o directori que pertany al sistema; una referència que podem posar en qualsevol lloc que ens interessi i que actua com un accés directe a qualsevol altre. Aquest mecanisme ens permet accedir a carpetes o arxius de manera més ràpida i còmoda, sense haver-nos de desplaçar 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 l'ordre llarga 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 readreci directament cap allà. L'ordre `ln -s /usr/share/man/man3 mmm` ens crearia aquest pont, que hem denominat `mmm`. L'usuari només hauria d'escriure (des del seu directori home) `cd mmm` i, automàticament, el sistema el readreçaria cap a `/usr/share/man/man3/`. És important tenir en compte que en fer un `cd ..` per a anar al directori superior, tornariem al directori home i no a `usr/share/man/`, ja que hi hem accedit a partir del nostre enllaç. Podem veure aquest esquema de manera gràfica en la figura següent:

Figura 2.2



En crear l'enllaç de l'exemple anterior hem passat el paràmetre `-s` a l'ordre. Això indica que volem crear un enllaç simbòlic. Els enllaços simbòlics signifiquen que només estem creant un apuntador o pont cap al fitxer o directori, de manera que si esborrèssim el fitxer de destinació, l'enllaç no assenyalaria enlloc. Si no hi 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, són com dues entrades que assenyalen cap a les mateixes dades. D'aquesta manera, si en modifiquem l'un o l'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 contingués també s'haurien de referenciar. Per aquesta raó, només el *root* del sistema pot efectuar enllaços forts de directoris. Una altra diferència és que amb un enllaç simbòlic podem veure cap a quin fitxer estem assenyalant, mentre que amb un de fort no podem (a causa del mecanisme que s'utilitza internament per a ells).

Creació d'enllaços forts

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.

2.3.5. Permisos

En qualsevol sistema operatiu multiusuari necessitem que els fitxers que desem en el nostre disc puguin tenir una sèrie de propietats que ens permetin veure'ls, modificar-los o executar-los per als usuaris que nosaltres hi definim. Encara que hi ha diverses alternatives per a dur a terme això, GNU/Linux utilitza el sistema clàssic 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 cadascun d'ells, així com per a la resta d'usuaris del sistema. En executar `ls -l` veurem com a cada arxiu del directori on som apareix una línia semblant a la següent:

```
-rwxr-xr-x 1 user1 grup1 128931 Febr. 19 2000 gpl.txt
```

Els primers deu 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 el cas de ser un fitxer, apareix el caràcter "`-`", mentre que per als directoris hi apareix una "`d`".
- Caràcters 2, 3, 4: ens indiquen, respectivament, els permisos de lectura, escriptura i execució per al propietari del fitxer. En el cas de no tenir el permís corresponent activat, hi 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 significa que en executar-lo obtindrà els permi-

sos 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 els anteriors, 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.

Després d'aquests 10 caràcters trobem una xifra que ens indica el nombre d'enllaços forts que té el fitxer. Per als directoris, aquest número 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, de l'últim accés i de l'última modificació, que podem manipular amb l'ordre `touch`. Al final es troba el nom del fitxer, en el qual es diferencien minúscules de majúscules i podem tenir tot tipus de caràcters sense cap problema.

El mecanisme de SetUserId

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 aquest tipus de fitxers perquè poden suposar errors de seguretat en el sistema si s'utilitzen malament.

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

- La primera manera d'utilitzar-lo és de l'estil `chmod XXX nombreArquivo`. Les *x* han de ser tres números entre 0 i 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:

Taula 2.1

Representació decimal	Representació binària	Significat
0	000	---
1	001	--x

Representació decimal	Representació binària	Significat
2	010	-w-
3	011	-wx
4	100	r---
5	101	r-x
6	110	rw-
7	111	rwX

- L'altra manera d'utilitzar l'ordre és indicar de manera explícita quin permís volem donar al fitxer o 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-hi o eliminar-ne l'atribut, que indicarem amb "r", "w", "x" o "s" (aquest últim per al SetUserId). A més, podem fer totes les combinacions possibles, i referir-nos a més d'un permís i/o usuaris. Per exemple, `chmod go+r gp1.txt` concediria el permís de lectura al grup i als altres usuaris per al fitxer `gp1.txt`.

Per a canviar el propietari d'un fitxer hi ha l'ordre `chown`, que només pot utilitzar el *root* per raons de seguretat. Per a canviar el grup d'un arxiu determinat, es pot utilitzar l'ordre `chgrp`. Com podem suposar, quan un usuari crea un nou arxiu, 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 l'ordre `umask`, a la qual hem de passar la mateixa notació de tres números decimals entre 0 i 7 que vèiem anteriorment però complementats. Per exemple, si volem que els nostres fitxers s'inicialitzin amb els permisos `rw-r--r--`, hauríem d'escriure `umask 133`.

Política de seguretat de fitxers

Si es permetés que els usuaris canviessin el propietari dels seus fitxers, la seguretat del sistema quedaria compromesa, ja que es podrien dur a terme accions malicioses i després canviar el propietari dels arxius utilitzats i inculpar-ne altres usuaris.

2.3.6. Manipulació, patrons i cerques

Ara que ja ens sabem moure correctament per la jerarquia de directoris, també necessitem saber com copiar, eliminar i manipular correctament altres aspectes dels fitxers. L'ordre `rm` és la que s'encarrega d'eliminar els arxius que li indiquem. Per a eliminar un directori, podem utilitzar l'ordre `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 l'ordre `cp`, a la 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/`

Sintaxi de patrons

La sintaxi dels patrons pot arribar a ser molt complexa, la qual cosa ens permet referenciar qualsevol conjunt d'arxius que vulguem.

`gp1.txt` en el directori actual (i amb el mateix nom) hauríem d'escriure `cp /home/user1/gp1.txt` (és a dir, usant el punt "."). Si en lloc de copiar els arxius, els volem moure de lloc, podem utilitzar l'ordre `mv`. Un mecanisme molt útil que ens proporciona el sistema són els patrons. Fins ara, hem vist com aplicar certes operacions sobre un arxiu determinat. Quan estem manipulant 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 noms, de manera que el caràcter "*" significa qualsevol cadena de caràcters possibles i el "?" ens serveix com a comodí de 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 "[" podem incloure altres caràcters, i indicar 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 continuen amb qualsevol altra cadena, podríem escriure el patró `[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 "[" podem especificar classes de caràcters de la manera següent: `[:class:]`, en què la `class` pot ser qualsevol de les indicades en la taula següent:

Taula 2.2

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 de les ordres que hem vist i la majoria de les que veurem a continuació. A més, la major part de les ordres de llista, eliminació, còpia, etc. de fitxers també permeten que se'ls passi de manera recursiva. D'aquesta manera, s'anirà entrant i executant la instrucció corresponent a tots els arxius i directoris, a partir del lloc 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 diverses ordres que ens permeten fer recerques de diferents tipus sobre tots els fitxers del sistema.

Taula 2.3

find	És l'ordre més versàtil per a fer aquesta acció. Ens permet filtrar els fitxers per a trobar des dels que tenen un nom determinat, els modificats o creats a partir d'una certa data, els que tenen certs permisos, etc. El seu únic desavantatge és que no utilitza cap tipus de mecanisme per a accelerar la recerca, amb la qual cosa, aquesta pot trigar bastant.
locate	Es tracta d'una altra ordre que, 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 podem fer recerques tan versàtils com amb <code>find</code> .
whereis	Finalment, <code>whereis</code> s'orienta 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 molts tipus 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 *magic numbers*, que determina amb un nombre màgic el tipus de fitxer segons les seves dades (es passen una sèrie de tests que intenten determinar de quin tipus és el fitxer). L'ordre `file` ens ho indica.

Si necessitem veure el contingut d'un fitxer, una de les ordres bàsiques és `cat`. En passar-li el nom o els noms dels arxius que volem veure, es mostra per pantalla. Hem d'intentar no mostrar fitxers executables o de dades per pantalla, ja que l'abocament 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 les ordres `less` o `more`, que ens permet desplaçar-nos pel fitxer de manera progressiva. Si el tipus de fitxer és binari i volem veure què conté, podem utilitzar les ordres `hexdump` o `od` per a veure el contingut de 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 d'ordres molt útils són les que ens busquen un cert patró en el contingut dels fitxers. Amb l'ordre `grep` li podem passar com a segon paràmetre el nom de l'arxiu i com a primer, el patró que vulguem buscar (amb la sintaxi que vàiem anteriorment, estesa a altres opcions). A més, l'ordre ens permet múltiples accions més, com comptar el nombre de línies en què apareix el patró (paràmetre `-c`), etc. Amb `cut` podem separar en camps el contingut de cada línia del fitxer especificant quin caràcter n'és el separador, molt útil en tasques d'administració del sistema per a la seva automatització. També podem prendre un determinat nombre de línies del començament o del final d'un arxiu amb les ordres `head` i `tail`, respectivament. Amb `wc` podem comp-

Ordre `updatedb`

Si volem actualitzar la base de dades interna que utilitza l'ordre `locate`, podem utilitzar l'ordre `updatedb`.

Extensió d'arxius

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

tar el nombre de línies o paraules, la màxima longitud de línia d'un fitxer, etc. Finalment, per acabar aquest subapartat de manipulació de fitxers, l'únic que ens falta per veure és com comparar diferents arxius. Igual que amb les altres operacions, tenim diverses ordres que ens permeten fer-ho. `diff`, `cmp` i `comm` fan comparacions de diferents maneres i mètodes en els fitxers que indiquem. `sdiff`, a més, permet barrejar-los en 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 anar fent les seves operacions, amb la qual cosa fa la sensació que s'estan executant tots alhora.

Per a identificar de manera inequívoca cada procés, el nucli del sistema els assigna un número anomenat PID (Process IDentification). Encara que podríem pensar que únicament amb el nom ja els podríem referenciar, és imprescindible disposar d'aquest número per a poder executar un mateix programa tantes vegades com vulguem, al mateix temps que se n'executen diferents instàncies. Per a saber quins processos s'estan executant, podem utilitzar l'ordre `ps`. Per a explorar una mica més tot aquest mecanisme de processos, explicarem amb més detall alguns dels paràmetres que podem passar a aquesta ordre:

- `T`: aquesta opció ve per defecte i ens indica que només es mostraran els processos que s'estan executant al terminal en el qual ens trobem o que s'hagin llançant a partir d'ell.
- `-a`: ens mostra els processos de tots els terminals del sistema.
- `-:` ens mostra tots els processos del sistema. Si executem l'ordre, veurem que, a part dels programes que els usuaris executen, n'hi ha d'altres. Molts d'ells 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 en el qual 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 en absolut que faci 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 implementada una sèrie de mecanismes

Gestió de processos

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.

molt eficaços per a gestionar tota aquesta classe d'operacions. D'aquesta manera, un procés es pot trobar 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 es deixés d'atendre.
 - R: procés que en el moment d'executar l'ordre també s'està executant, és a dir, tots aquells que són en cua d'execució. La cua d'execució de processos és on es posen tots aquells que es van repartint el temps de la CPU.
 - S: procés adormit o que espera que ocorri algun tipus d'esdeveniment perquè 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 zombi. Aquest estat indica que el procés ha tingut algun error i no funciona correctament. Generalment, és millor eliminar aquest tipus de processos.
- Una altra ordre molt útil és `top`, que ens informa de manera interactiva 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 notem alguna disfunció estranya, ja que ens permet localitzar ràpidament quin procés està afectant 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 uns senyals als processos a fi d'informar-los d'algun esdeveniment, els podem treure 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 en el qual la majoria de les aplicacions que s'executen necessiten molt temps de CPU, podríem configurar el sistema per tal de fer que els més urgents s'executin amb més prioritat que d'altres i acabin primer. L'ordre `kill` ens permet enviar senyals als processos que ens interressi. 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 unes altres. Hi ha molts tipus diferents de senyals, que podem veure en el manual de `kill`, encara que les més utilitzades són les que ens serveixen per a obligar un procés que acabi o interrompi la seva execució. Amb el senyal `TERM` (`kill -15 PID`), indiquem al procés que volem que acabi, de manera que en rebre el senyal haurà de desfer tot 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`),

Ordre trap

Per a tractar els senyals en un *shell script* (vegeu més endavant com programar-los), podem utilitzar l'ordre `trap`.

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 l'ordre `skill` també podem enviar senyals als processos, però amb una sintaxi diferent. Per exemple, si volem aturar totes les execucions d'un usuari determinat, podríem utilitzar `skill -STOP -u nomLogin`, amb la qual cosa tots els processos d'aquest usuari s'aturarien. Per a reiniciar-los de nou, podríem passar el senyal de `CONT`. Quan estem executant 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`.

Manipulació de processos

Amb les ordres de manipulació de processos podem fer qualsevol acció que ens interessi: 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.

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, sigui el mateix intèrpret d'ordres, l'entorn gràfic, etc., de manera que es crea una relació de pares a fills. Amb l'ordre `pstree` podem veure aquesta jerarquia de manera gràfica. Si l'executem, veurem com el pare de tots els processos és un anomenat `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, en matar 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 a l'ordre, per defecte compacta tots els processos amb un mateix nom per 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 dèiem 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 major a menor. Per a llançar un procés amb una prioritat determinada, podem utilitzar l'ordre `nice`. Si volem donar una prioritat diferent a 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` disposi sempre de la possibilitat d'executar processos més ràpidament que els usuaris. Per defecte, la prioritat amb què s'executen els programes és la 0. Un aspecte que caldrà tenir en compte 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 a la cua d'execució. En centres de càlcul en els quals 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 l'ordre `time`, la qual, en passar-li el programa que volem mesurar, ens torna el temps real de CPU que ha utilitzat.

2.5. Altres ordres útils

2.5.1. L'ajuda del sistema

Com hem dit al llarg del document, totes les ordres tenen multitud d'opcions i paràmetres diferents que ens permeten manipular-les segons la nostra elecció. Des del principi es va tenir molt en compte que és imprescindible tenir una bona documentació per a totes elles. Igualment, tota aquesta informació és necessària per als fitxers de configuració del sistema, les noves aplicacions que utilitzem, etc. Per això, el mateix sistema incorpora un mecanisme de manuals amb què podem consultar gairebé tots els aspectes dels programes, utilitats, ordres i configuracions existents. L'ordre més utilitzada és la `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 endavant i enrere amb les tecles d'"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 es divideixen en diferents seccions segons la seva naturalesa:

- 1) Programes executables (aplicacions, ordres, etc.).
- 2) Crides al sistema proporcionades pel *shell*.
- 3) Crides a biblioteques del sistema.
- 4) Arxius especials (generalment els de dispositiu).
- 5) Format dels arxius de configuració.
- 6) Jocs.
- 7) Paquets de macro.
- 8) Ordres d'administració del sistema (generalment aquelles que només el *root* pot utilitzar).
- 9) Rutines del nucli.

Si hi ha més d'un manual disponible per a una mateixa paraula, el podem especificar passant-li el número corresponent de la secció que ens interessa abans de la paraula, per exemple `man 3 printf`. Com les altres ordres, `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 sabem exactament el programa que estem buscant, és `-k` (l'ordre `apropos` fa gairebé el mateix). Amb `man -k` seguida d'una paraula que faci referència a l'acció que vulguem dur a terme es buscarà per 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.

Ordre `mandb`

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

Si el manual no ens proporciona tota la informació que necessitem, podem usar l'ordre `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, biblioteca, etc., podem utilitzar l'ordre `whatis`.

2.5.2. Empaquetat 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 ell. La sintaxi que utilitza és la següent: `tar opciones archivoDestino archivosOrigen`, en què 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 nou arxiu, li hem de passar el paràmetre `c`, i si el volem desar en un arxiu, li hem de passar el `f`. D'aquesta manera, `tar cf final.tar o*` empaquetarà tots els arxius del directori actual que comencin per "o". Si a més els volguéssim comprimir, 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 el `x`, de manera que hauríem d'escriure `tar xf` per indicar el fitxer empaquetat. Si estigués comprimit, hauríem de passar `xzf`.

Tot i 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 programa `gzip` utilitza un format de compressió propi i diferent del `zip` tan popularitzat, que també podem utilitzar en instal·lar 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 utilitzar per a identificar quin format utilitza un arxiu comprimit o empaquetat:

Taula 2.4

Extensió	Format
.tar	tar
.gz	gzip
.tgz	tar + gzip

Extensió	Format
.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 aquest subapartat explicarem quines són les ordres necessàries per a veure informació relativa a aquests. Tot disc dur es divideix en **particions**, a les quals podem accedir com si es tractés d'un dispositiu independent, i les denominarem **unitat**. Això és molt útil perquè ens permet separar de manera adequada la informació que tinguem en el sistema, tenir més d'un sistema operatiu instal·lat al mateix disc, etc. L'ordre `df` ens mostra, de cada unitat muntada en el sistema, l'espai que s'ha utilitzat i el que queda lliure. Interpretarem la sortida següent de `df`:

Filesystem	1k-blocks	Used	Available	Use%	Mounted on
/dev/hda1	7787712	421288	6970828	6%	/
/dev/hdb1	19541504	5742384	13799120	29%	/info
/dev/hdc	664432	664432	0	100%	/CD-ROM

Com podem veure, per cada partició o dispositiu muntat en el sistema l'ordre 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, aquesta ordre ens ensenya la informació per blocs de 1 k, encara que passant-li el paràmetre `-h` (*human readable*) ho 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 a la qual hauríem d'anar per tal de poder veure el seu contingut.

Una altra ordre 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 de veure amb una mica més de deteniment l'organització interna dels discos i com el sistema operatiu els manipula. Tal com dèiem anteriorment, per raons d'eficiència el sistema operatiu divideix l'espai del disc en petits trossos denominats 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 nou arxiu, el sis-

Configuració d'una partició

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.

tema operatiu hi assigna un bloc. D'aquesta manera, en llegir-lo o operar-hi, l'operatiu pot llegir directament tot un bloc (de la mida configurada) en un únic pas. Quan el fitxer ocupa més d'un bloc, se n'hi assignen més, i s'intenta que quedin tan junts com sigui possible, de manera que es puguin llegir consecutivament i s'incrementi, 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. L'ordre `du`, doncs, ens mostra el nombre de blocs que realment utilitza un arxiu determinat en el disc.

Per a saber els paràmetres que tenim configurats en les nostres unitats de disc formatades amb `ext2` o `ext3`, podem utilitzar l'ordre `dumpe2fs`, i passar-li la partició concreta. Veurem que hi ha multitud d'opcions diferents que ens permeten ajustar molt bé el seu comportament (en el manual trobarem què significa cada opció). De tota manera, una vegada haguem formatat una partició, ja no podrem modificar gairebé cap d'aquestes opcions. Si les volguéssim canviar, hauríem de copiar tota la informació de la partició, formatar de nou i tornar a copiar els arxius originals.

Desfragmentació d'un disc

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 és necessari desfragmentar els discos (encara que hi ha programes amb aquesta finalitat), perquè el sistema s'encarrega automàticament de la seva bona organització.

Les funcions del nucli que s'encarreguen de la gestió de fitxers utilitzen una sèrie de mètodes per a agilitar els processos de lectura i escriptura. Un d'ells és la utilització d'una memòria cau de disc, de manera que no s'hagi d'estar constantment llegint i escrivint en el dispositiu físic, que resulta un procés lent i costós. L'únic que fa el mecanisme cau és mantenir una còpia del fitxer amb què s'està treballant en la memòria RAM (molt més ràpida), de manera que el procés sigui transparent per a l'usuari (la còpia a 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 en el 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 l'ordre `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, perdre'ls, també podem configurar l'operatiu perquè no utilitzi el sistema de memòria cau 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 cau de disc amb el disc físic, podem utilitzar l'ordre `sync`. Finalment, també podem comprovar la integritat física d'una partició utilitzant l'ordre `badblocks`, que duu a terme una revisió sobre el dispositiu indicat per a comprovar que no hi hagi cap zona danyada.

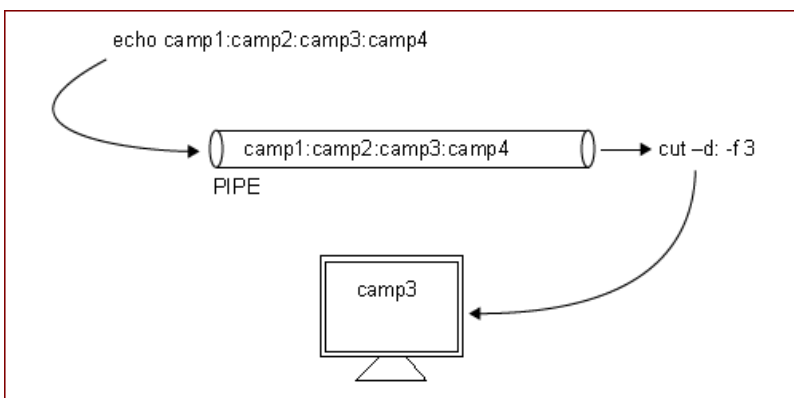
La majoria de les ordres exposades en aquest subapartat necessiten permisos especials per a executar-se, de manera que només l'usuari `root` els podrà utilitzar.

2.6. Operacions amb ordres

2.6.1. Redreçaments

Una vegada hem après a utilitzar algunes de les ordres del sistema, és molt probable que en alguns casos ens interessi utilitzar-les de manera simultània per a agilitar les accions que volem dur a terme. Una operació molt interessant consisteix a poder prendre la sortida d'una ordre perquè serveixi d'entrada a una altra i processar-la adequadament. El sistema operatiu utilitza un mecanisme de *pipes* (canonades) que ens permet redirigir les sortides de qualsevol ordre o programa cap a on vulguem. El seu funcionament és molt simple: es tracta de posar el caràcter `|` entre les ordres, de manera que la sortida de la primer serveix com a entrada per a la segona. Ho veurem amb un exemple: en escriure l'ordre `echo camp1:camp2:camp3:camp4`, l'única cosa que aconseguiríem seria que per pantalla ens aparegués "camp1:camp2:camp3:camp4". Si d'aquesta sortida només volguéssim prendre el "camp3", la podríem redirigir amb un *pipe* cap a l'ordre `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 de manera gràfica:

Figura 2.3



Naturalment, podem connectar tantes canonades com necessitem per a fer accions més pràctiques que la que acabem de veure. Un altre tipus de readreçaments molt pràctics són aquells que estan relacionats amb els fitxers. Aquest tipus de readreçament ens permet prendre tota la sortida d'un ordre 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 anem escrivint fins a prémer "Ctrl+C", podríem utilitzar el següent: `cat > prova.txt`. Amb ">>" podem fer exactament el mateix, però en lloc de crear sempre el nou fitxer, si aquest ja existís, s'hi afegiria la informació al final. Amb "<" el readreçament es fa en sentit contrari, de manera que el contingut del fitxer que li indiquem es dirigirà cap a l'ordre o programa assenyalats.

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 en la qual s'ha executat el programa, les podem manipular perquè es dirigeixin cap a on vulguem. Per a veure això de manera pràctica, intentem esborrar un fitxer que no existeix amb la instrucció següent: `rm fitxer > resultats`. Encara que estem readreçant la sortida de l'ordre cap al fitxer de resultats, per pantalla ens apareixerà un missatge d'error que ens indicarà que no s'ha trobat el fitxer. Això es deu al fet que, per defecte, els readreçaments només accepten la sortida estàndard del programa i no la d'error, que per defecte també es mostra per pantalla. Per a readreçar la sortida d'error, hauríem d'indicar, abans del caràcter ">" el número "2" que és la sortida d'error (l'"1" és la normal). D'aquesta manera, en executar `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 dirigessin cap a un mateix arxiu, podríem utilitzar ">&". A més, amb el caràcter "&" podem encaminar sortides d'un tipus cap a unes 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 readreçaments és significatiu: sempre s'executen d'esquerra a dreta.

2.6.2. Ordres específiques del *bash*

Tot i que algunes de les ordres que hem vist ja són específiques del *bash*, aquest intèrpret d'ordres 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 es denomina mode *background*. Aquest mode indica senzillament que el procés s'està executant, però que el *shell* ens torna la línia d'ordres per a poder continuar executant altres programes. Per tal d'indicar això al *bash*, hem d'escriure el caràcter "&" després de l'ordre o programa que executarem. Una vegada s'ha llançat el procés en mode *background*, es mostra una línia en la qual se'ns indica el número de treball i PID del procés llançat.

Amb l'ordre `jobs` podem veure quins processos estan llançats en mode *background* (en passar-li el paràmetre `-l` també podem veure el seu PID). Si volguéssim passar un d'aquests processos a mode *foreground* (com si l'haguéssim llançat des de la línia d'ordres sense el caràcter "&"), podem utilitzar l'ordre `fg` i indicar el PID del procés. També existeix `bg`, que ens envia un procés determinat a mode *background*. Aquest últim és útil quan, per exemple, executem un programa en mode *foreground* i l'interrompem amb "Ctrl+Z". Si després executem `bg` i li indiquem el seu PID, el procés continuarà la seva execució en mode *background*. Tal com vèiem en els subapartats anteriors, els processos també tenen una jerarquia de pares a fills. Quan executem algun programa en mode *background* no estem interferint en aquesta jerarquia, de manera que si sortim de la sessió, tots aquests processos s'acabaran perquè el pare (l'interpret d'ordres des d'on els hem llançat) ja no estaria en execució. Si volem desvincular un procés del seu pare, podem utilitzar `disown`.

Un altre mecanisme molt útil del *bash* és la història d'ordres. És normal que utilitzant el sistema haguem de repetir moltes instruccions escrites anteriorment. Amb les tecles del cursor a dalt i a baix podem anar veient totes les ordres que hem anat utilitzant i repetir-ne alguna prement "Return". També podem utilitzar `history`, amb la qual cosa es mostraran per pantalla totes les ordres executades, enumerades segons la seva aparició. En escriure `!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 a executar-lo.

L'interpret d'ordres *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 sencer d'un fitxer, directori o ordre. 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 l'ordre `clear`).
- "Shift+RePág": ensenya mitja pantalla anterior.
- "Shift+AvPág": 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.

Bash

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

- "Ctrl+D": surt de l'interpret d'ordres (equival a fer un `logout`).
- `ulimit` és una ordre que ens permet configurar alguns dels aspectes interns relacionats amb el `bash`. Per exemple, permet indicar la quantitat de memòria que pot utilitzar l'interpret d'ordres, el nombre màxim d'arxius que es poden obrir, etc. Aquesta ordre ens pot servir per a restringir una mica les accions que poden fer els usuaris del nostre sistema (en el cas d'administrar servidors amb molts usuaris).

2.6.3. Shell scripts amb `bash`

Els *shell scripts* són fitxers en els quals escrivim una sèrie d'ordres (qualsevol de les que hem vist en aquest apartat) perquè siguin executades. Encara que la seva sintaxi pot arribar a ser molt complexa i hauríem d'entrar en aspectes de programació per a entendre-la clarament, en aquest subapartat explicarem de manera resumida algunes de les seves característiques essencials perquè les puguem entendre i utilitzar mínimament (si volem aprofundir-hi més, podem recórrer al manual del `bash`). La primera línia del *shell script* ha d'especificar l'interpret d'ordres que s'utilitza:

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

Després d'aquesta línia ja podem començar a escriure les ordres que volem executar, una en 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, l'hem de posar entre cometes, si és un número, no és necessari posar-hi res i si volem desar en la variable la sortida d'una ordre, l'hauríem de posar entre caràcters. Per tal de 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
```

en què `condicio` pot fer referència a un arxiu, fer alguna operació de comparació aritmètica (entre caràcters "`(())`"), etc. D'especial utilitat és l'ordre `test`, que ens permet efectuar comprovacions d'arxius, directoris, etc. i ens torna

Ordre `fc`

L'ordre `fc` ens permet, igual que els *shell scripts*, escriure una sèrie d'ordres perquè s'executin però sense haver de desar l'arxiu.

un booleà. D'aquesta manera, per exemple, si volguéssim efectuar 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 s'executaran les instruccions del cas. 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 *script* determinat 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 consecutivament. Per als bucles podem utilitzar alguna de les estructures següents:

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

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

Comentaris en els *shell scripts*

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.

3. Taller de Knoppix

3.1. Introducció

Aquest taller pretén ser la vostra primera experiència amb un entorn UNIX. Per aquesta raó, el seu desenvolupament és guiat pas a pas, però deixa, sens dubte, la porta oberta als més curiosos perquè investiguin per compte propi.

L'objectiu principal d'aquest taller és familiaritzar-nos amb el sistema operatiu GNU/Linux i veure que tot allò que estem acostumats a fer amb altres sistemes operatius, es pot dur a terme exactament igual amb el sistema operatiu GNU/Linux. En aquest taller també es pretén començar a treballar amb la línia d'ordres, familiaritzar-nos-hi i perdre la por a tot allò que no és un entorn gràfic. Així doncs, és hora de posar en pràctica tot el que s'ha exposat fins ara de manera teòrica.

Aquest taller es pot desenvolupar sobre qualsevol ordinador, ja que el risc de danyar la informació que podem tenir és mínim. S'ha triat aquesta distribució perquè per a arrencar-la no es requereixen coneixements previs del sistema operatiu, i perquè, una vegada aturat el sistema, no deixa rastre.

KNOPPIX per defecte munta en el sistema totes les particions del disc dur, però les munta només amb permisos de lectura; per tant, no hi podem escriure res ni executar res, tret que nosaltres ho forcem canviant els permisos. Òbviament, si es disposa d'un altre sistema operatiu GNU/Linux, es pot usar per a fer el seguiment del taller.

El fet de ser una distribució arrencable (*bootable*) des d'un CD-ROM o DVD-ROM i no deixar rastre a l'ordinador on s'ha executat –una vegada ha acabat el procés de parada– provoca que, tot i basar-se en Debian, el sistema de fitxers no segueixi el que marca la Debian Policy sobre això. No obstant això, aquestes diferències no afectaran el desenvolupament del taller, i tot el que hi aprenguem serà vàlid per als 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 pròpies iniciatives, intentem respondre nosaltres mateixos a les nostres inquietuds, consultem l'ajuda que ens ofereix el *man*, fem proves, fallem i analitzem el perquè

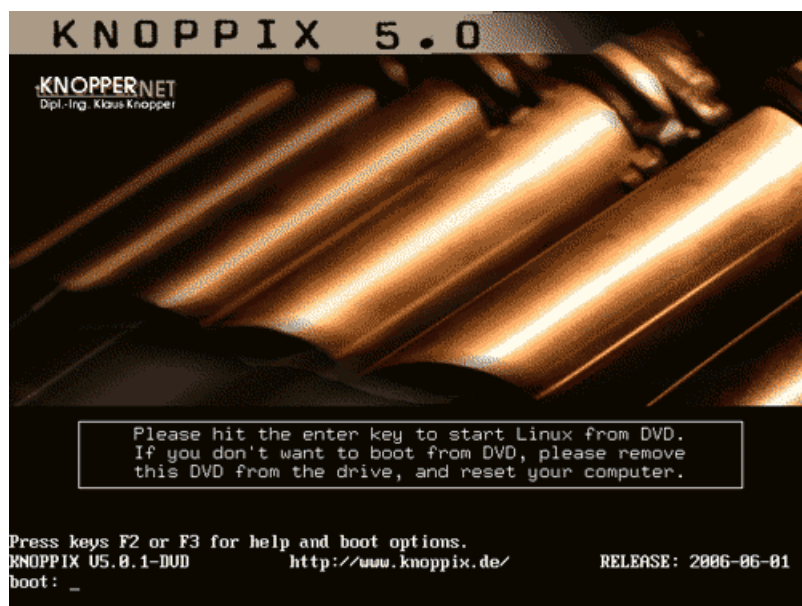
ho hem fet, ho intentem de nou, una i una altra vegada, fins a aconseguir els resultats volguts; així és com s'aprèn UNIX, sense por, traient partit de les errades pròpies.

3.2. Arrencada del sistema

En primer lloc, ens hem d'assegurar que el nostre ordinador arrencarà des del lector de CD-ROM o DVD-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 lector de CD-ROM o de DVD-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 o DVD-ROM de KNOPPIX al lector. Si no fos el cas, faríem els canvis pertinents i els desariem abans de sortir de la BIOS.

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

Figura 3.1



Podem prémer la tecla "F2" o "F3" per entrar a la pantalla en què es mostren les opcions que accepta KNOPPIX per a arrencar, a continuació podem veure la informació que se'ns presenta en prémer la tecla "F2":

Figura 3.2

```

>>>          KNOPPIX 05.0.1 BOOT OPTIONS (Back to main with F1)          <<<

This Edition of Knoppix boots with standard Kernel 2.6.
The following options can be used at the boot:-prompt.

knoppix [Options, see F3]      Knoppix standard Kernel 2.6
debug   [Options, see F3]      Knoppix with verbose kernel messages
expert  [Options, see F3]      Interactive setup (expert mode)
knoppix testdvd                Check DVD for burning errors

memtest                          Memtest86, Memory checking program
m23                               Start m23 software distribution system

fb1280x1024, fb1024x768 or. fb800x600 Framebuffer-Mode (for Notebooks)

knoppix acpi=off noapic pci=bios  Workarounds for broken BIOS

Hit F3 for more options. A complete list of bootoptions can be found in the
file knoppix-cheatcodes.txt on your DVD, inside the "KNOPPIX" directory.

boot:

```

I aquesta és la informació que se'ns presenta en prémer la tecla "F3":

Figura 3.3

```

>>>          KNOPPIX 05.0.1 CHEATCODES (F1 for Main Page)          <<<

The KNOPPIX autoconfiguration scripts accept the following
boot options (see knoppix-cheatcodes.txt for the full list):

knoppix lang=us:es:ida:de:es:fr:it:nl:pl:ru:sk!...      Set keyboard/language
knoppix desktop=icwm:kde:gnome:fluxbox:wmaker:twm!...  Use a different Desktop
knoppix screen=1280x1024 depth=24                      Set XFree resolution and color depth
knoppix toram                                          Load entire CD to ramdisk (1GB)
fb1280x1024 : fb1024x768 : fb800x600                  Use framebuffer mode (for notebooks)
knoppix dma                                           Turn on dma acceleration
knoppix vsync=85 hsync=78                             85Hz vert. / 78kHz horiz. mon. freq.
knoppix 2                                             Runlevel 2, textmode only
knoppix myconfig=scan home=/dev/sda1                  load/mount configuration and homedir
knoppix no(scsi:pcmcia:usb:agp:swp:apm:apic:mce:ddc)  turn off hw-detection
knoppix blind brltty=typ,port,tbl                    braille terminal(type), blind mode
failsafe                                             turn off (almost) ALL hw-detection
expert                                              interactive configuration

More options can be found inside the "KNOPPIX" directory on DVD.

boot:

```

Per exemple, podríem arrencar amb el teclat en espanyol i llengua castellana per a la interacció, amb GNOME com a *windows manager* i activar l'*scroll wheel* del ratolí; per a fer-ho, n'hi hauria prou que teclegéssim en la línia d'ordres (boot:) el següent: `knoppix lang=es gnome wheelmouse`. Però no és el cas, l'exemple anterior era només per a mostrar la potència de KNOPPIX. Nosaltres, després d'inserir a la disquetera un disquet nou formatat –sobre el qual farem els nostres primers passos en GNU/Linux, de manera que garantirem la resta d'informació que puguem tenir en els nostres discos durs–, només escriurem `knoppix 2` i premerem "Intro" per tal d'arrencar el sistema en mode text.

Figura 3.4

```

>>>          KNOPPIX 05.0.1 BOOT OPTIONS (Back to main with F1)          <<<
This Edition of Knoppix boots with standard Kernel 2.6.
The following options can be used at the boot:-prompt.

knoppix      [Options, see F3]      Knoppix standard Kernel 2.6
debug        [Options, see F3]      Knoppix with verbose kernel messages
expert       Interactive setup (expert mode)
knoppix testdvd                      Check DVD for burning errors

memtest      Memtest86, Memory checking program
m23          Start m23 software distribution system

fb1280x1024, fb1024x768 or. fb800x600 Framebuffer-Mode (for Notebooks)

knoppix acpi=off noapic pci=bios      Workarounds for broken BIOS

Hit F3 for more options. A complete list of bootoptions can be found in the
file knoppix-cheatcodes.txt on your DVD, inside the "KNOPPIX" directory.

boot: knoppix 2_

```

Inicialment, el teclat està configurat per als Estats Units (US), així que alguns caràcters no es corresponen amb el teclat espanyol; això ho arreglarem tot seguit, tot i així, pot ser d'interès saber on es troben alguns caràcters (US) en el teclat espanyol:

- "=" és a la tecla ¿.
- "/" és a la tecla – o al teclat numèric.
- "-." és a la tecla ? o al teclat numèric.

Una vegada que hem premut la tecla "Intro", KNOPPIX començarà a carregar el sistema operatiu, i tornarà per pantalla alguns dels resultats dels tests que va executant per a l'autoconfiguració.

Una vegada acabat aquest procés, s'obtindrà la línia d'ordres:

Figura 3.5

```

Welcome to the KNOPPIX live Linux-on-DVD!

Scanning for USB/Firewire devices... Done.
Enabling DMA acceleration for: hda      [Umare Virtual IDE Hard Drive]
Enabling DMA acceleration for: hdc      [Umare Virtual IDE CDROM Drive]
Accessing KNOPPIX DVD at /dev/hdc...
Found primary KNOPPIX compressed image at /cdrom/KNOPPIX/KNOPPIX.
Found additional KNOPPIX compressed image at /cdrom/KNOPPIX/KNOPPIX2.
Total memory found: 401024 kB
Creating /ramdisk (dynamic size=311720k) on shared memory... Done.
Creating unionfs and symlinks on ramdisk...
>> Read-only DVD system successfully merged with read-write /ramdisk.
Done.
Starting init process.
INIT: version 2.86 booting
Running Linux Kernel 2.6.17.
Processor 0 is Intel(R) Pentium(R) M processor 1.70GHz 1687MHz, 2048 KB Cache
ACPI Bios found, activating modules: ac battery button container fan processor thermal video
USB found, managed by udev
Firewire found, managed by udev
Starting udev hot-plug hardware detection... Started.
Autoconfiguring devices... Done.
Mouse is ImPS/2 Generic Wheel Mouse at /dev/input/mice
Soundcard: Ensoniq ES1371 [AudioPCI-97] driver=snd-ens1371
AGP bridge detected.
Video is Umare Inc [Umare SUGA II] PCI Display Adapter, using Xorg(umare) Server
Monitor is Generic Monitor, H:28.0-96.0kHz, U:58.0-75.0Hz
Using Modes "1024x768" "800x600" "640x480"
Scanning for Harddisk partitions and creating /etc/fstab... Done.
Using swap partition /dev/hda2.
Network device eth0 detected, DHCP broadcasting for IP. (Backgrounding)
INIT: Entering runlevel: 2
root@tty1/~#

```

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

3.3. Aturada del sistema

Una vegada dins del sistema, el primer que hem de saber, tal com ja s'ha remarcat, és com parar-lo. Recordem, una vegada més, que no podem parar l'ordinador sense haver aturat abans el sistema operatiu (GNU/Linux). Hi ha múltiples maneres de fer-ho. Les més comunes són: prémer la combinació de tecles "Ctrl+Alt+Supr" o mitjançant l'ordre `halt` (n'hi ha moltes més, mitjançant l'ordre `reboot`, canviant de *runlevel* a 0 o 6, etc.). Una vegada haguem donat l'ordre al sistema perquè s'aturi, aquest començarà a executar les instruccions pertinents d'aturada (desmuntatge de dispositius, aturada de processos, etc.), i al final de tot aquest procés, KNOPPIX expulsarà el CD-ROM o DVD-ROM i ens demanarà que premem "Intro" per a aturar l'ordinador.

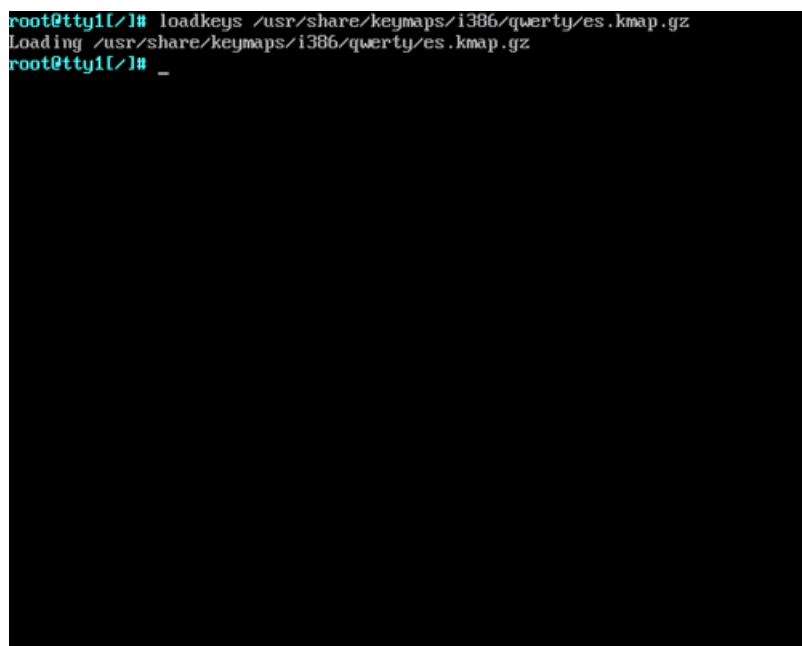
3.4. Configuració del teclat

Una vegada sabem com parar i arrencar el sistema, el tornem a arrencar seguint els mateixos passos d'arrencada anteriors. Una vegada siguem en el sistema, el primer que hem de fer és configurar el mapatge del teclat correctament. Hi ha diverses formes de fer-ho:

- manualment i si arrenquem en mode consola (`knoppix 2`):

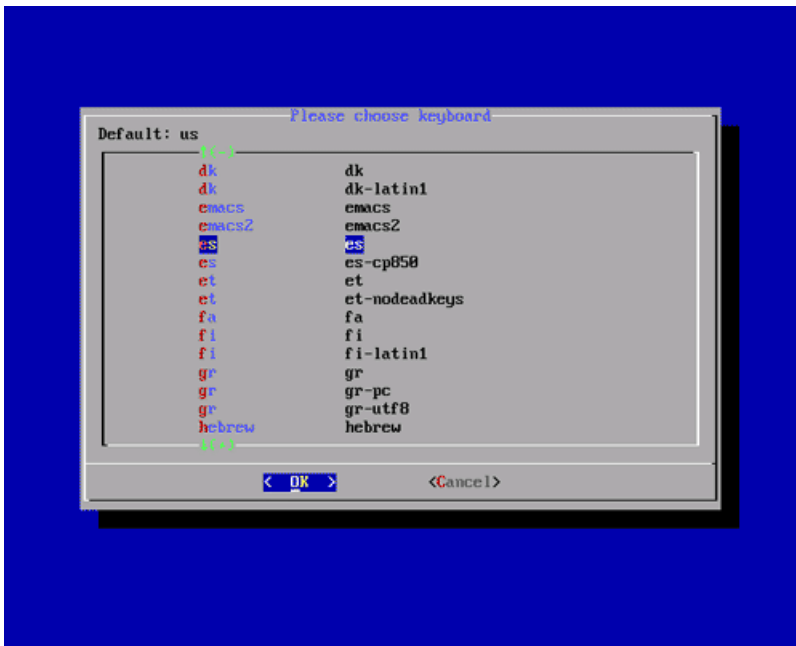
Figura 3.6

```
root@tty1l/1# loadkeys /usr/share/keymaps/i386/qwerty/es.kmap.gz
Loading /usr/share/keymaps/i386/qwerty/es.kmap.gz
root@tty1l/1# _
```



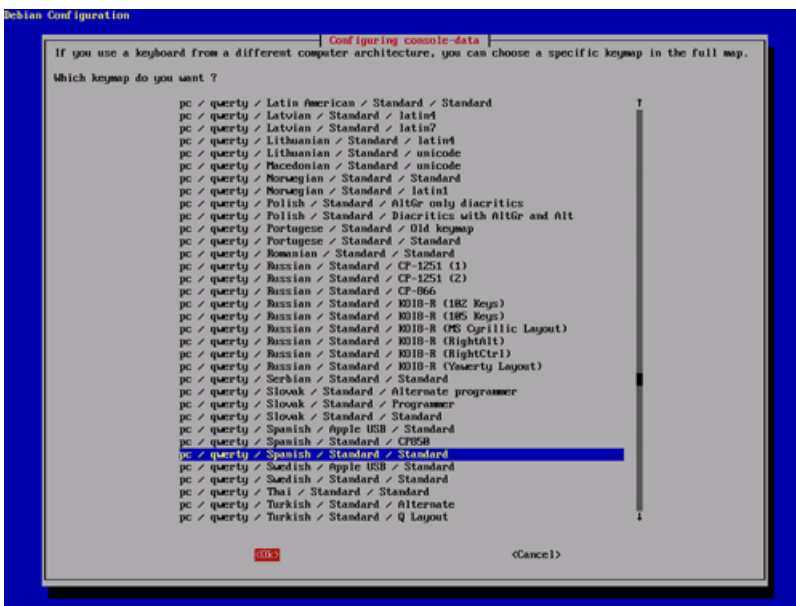
- gràficament amb l'ordre `kbdconfig` i escollint l'opció `es es`.

Figura 3.7



- amb l'ordre `kbd-config` i triant les opcions següents:

Figura 3.8



Tenim un avantatge a l'hora d'escriure en els terminals: és l'opció d'autocompletar prement "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 la mateixa ordre `loadkeys`. Teclegem només una "l" i premem la tecla "Tab" una vegada, el sistema emet un xiulet, i una segona per a obtenir el següent:

Figura 3.9

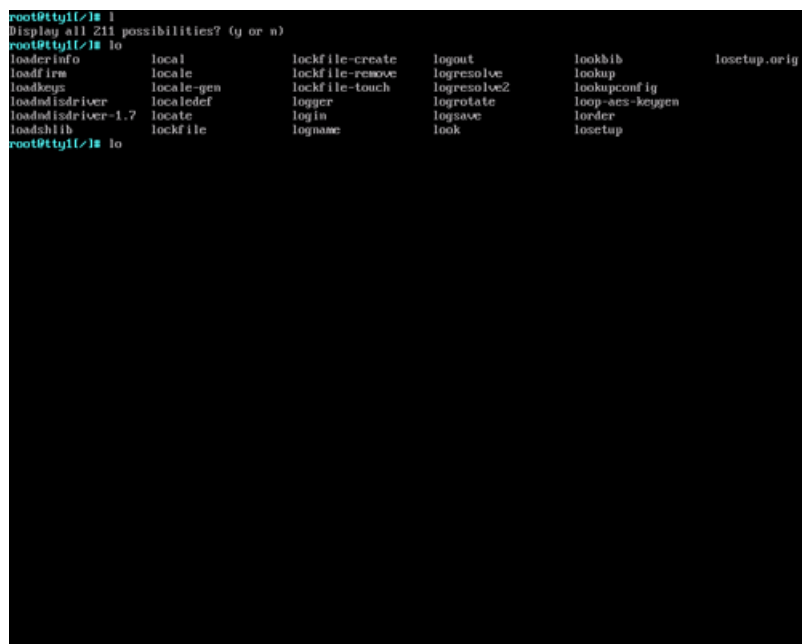
```
root@tty1[/]# l
Display all 211 possibilities? (y or n)
root@tty1[/]# l_
```



En aquest cas, no volem que se'ns mostrin les 211 possibilitats. Llavors, premem la "n" i hi afegim la lletra "o", i repetim l'operació anterior de prémer dues vegades la tecla "Tab". Ara obtenim un resultat diferent:

Figura 3.10

```
root@tty1[/]# l
Display all 211 possibilities? (y or n)
root@tty1[/]# lo
loaders info      local      lockfile-create  logout      lookbib      losetup.orig
loadfirm         locale    lockfile-remove  logresolve  lookup
loadkeys        locale-gen lockfile-touch   logresolve2 lookupconfig
loadndisdriver  localedef logger        logrotate  loop-aes-keygen
loadndisdriver-1.7 locate    login         logsave   lorder
loadshlib        lockfile  logname       look       losetup
root@tty1[/]# lo
```



Veiem que només falta afegir "adk" i tornar a prémer la tecla "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 que teclegem "loadkeys /u" i prémer "Tab" per a obtenir:

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

i tornar a prémer "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 les ordres, com els directoris i fitxers. És una eina l'absència de la qual es fa notar quan ens acostumem a utilitzar-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 l'ordre `pwd`:

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

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

"root": en aquest camp hi ha el nom d'usuari, *root* en aquest cas, informació una mica redundant, ja que indica el mateix el caràcter "#". Però per a la resta d'usuaris és interessant, ja que tots ells tenen el símbol "\$", i així podem saber si és la Isabel, el Daniel o un altre usuari.

Després del caràcter "@", que simplement serveix per a separar camps, hi ha "tty1"; aquest camp ens indica a quina terminal ens trobem, i és que, tal com ja s'ha dit, GNU/Linux és multiusuari i multiprocés; així doncs, no és estrany que puguem accedir el 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, per la qual cosa podem estar provant, per exemple, una ordre en una, en una altra tenir el *man* de l'ordre en qüestió per a anar llegint les seves opcions, mentre que en una altra podem tenir un editor obert per a anar prenent nota del que anem fent (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 al cinquè terminal). Així doncs, provem la combinació "Alt+F2", observem ara que en el camp que

estem estudiant apareix "tty2", i executem una ordre, per exemple `man man`, per a llegir el *man* o l'ajuda de l'aplicació `man`. De moment, la deixem allà i tornem a la `tty1`, i comprovem que tot està com ho havíem deixat.

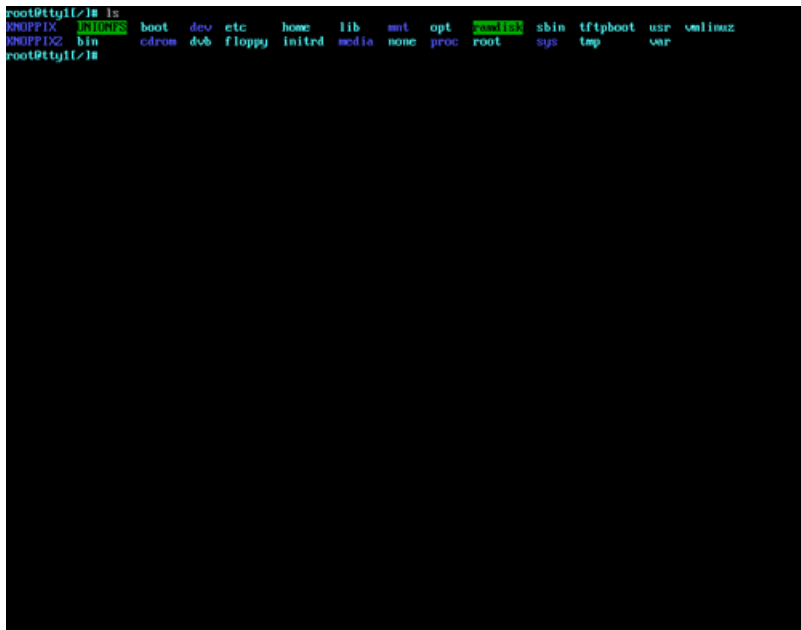
Finalment, i continuant amb la interpretació del *prompt*, abans del camp ja conegut "#" trobem "[/]": aquí se'ns indica el directori actual, l'arrel en aquest cas, tal com ens havia informat l'ordre `pwd`. La diferència rau en el fet que el *prompt* només ens informa del directori actual, mentre que `pwd` ens torna la ruta completa.

Per exemple, si fóssim a `/home/sofia`, tindríem:

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

Ara podem fer una llista dels continguts del directori on som. Per a fer-ho, utilitzarem l'ordre `ls`, primer sense passar-li cap opció:

Figura 3.11



```
root@tty1/1# ls
BINPP1X [111123] boot dev etc home lib mnt opt root sbin tftpboot usr vmlinuz
BINPP1X2 bin cdrom db floppy initrd media none proc root sys tmp var
root@tty1/1#
```

I ara li passem dues opcions (normalment les opcions van precedides de "-", encara que hi ha ordres que no ho exigeixen), per a indicar-li que ens torni informació més detallada. Així que una vegada escrit el "-" hi escriurem les opcions `la` per a obtenir aquesta informació:

Figura 3.12

```

root@tty1:/# ls
KNOPPIX  boot  dev  etc  base  lib  snt  opt  ramdisk /sbin  ttyboot  usr  vmlinuz
KNOPPIX2 bin  cdrom  deb  floppy  initrd  media  none  proc  root  sys  tmp  var
root@tty1:/# ls -la
total 14
drwxr-xr-x 12 root root 1824 Nov  5 11:11 .
drwxr-xr-x 12 root root 1824 Nov  5 11:11 ..
drwxr-xr-x 23 root root 4896 May 28 09:30 KNOPPIX
dr-xr-xr-x  5 root root 2048 May 28 11:28 KNOPPIX2
drwxr-sr-x 29 root root 198  Nov  5 11:11 KNOPPIX
lrwxrwxrwx  1 root root  12  Nov  5 11:11 bin -> /BINONFS/bin
lrwxrwxrwx  1 root root  13  Nov  5 11:11 boot -> /BINONFS/boot
dr-xr-xr-x  9 root root 4896 May 23 18:23 cdrom
drwxr-xr-x 14 root root 13788 Nov  5 16:12 dev
lrwxrwxrwx  1 root root  12  Nov  5 11:11 deb -> /BINONFS/deb
lrwxrwxrwx  1 root root  12  Nov  5 11:11 etc -> /BINONFS/etc
lrwxrwxrwx  1 root root  15  Nov  5 11:11 floppy -> /BINONFS/floppy
lrwxrwxrwx  1 root root  13  Nov  5 11:11 base -> /ramdisk/base
lrwxrwxrwx  1 root root  15  Nov  5 11:11 initrd -> /BINONFS/initrd
lrwxrwxrwx  1 root root  12  Nov  5 11:11 lib -> /BINONFS/lib
drwxr-xr-x 12 root root 1824 Nov  5 16:12 media
drwxr-xr-x  2 root root 1824 Nov  5 16:12 snt
lrwxrwxrwx  1 root root  13  Nov  5 11:11 none -> /BINONFS/none
lrwxrwxrwx  1 root root  12  Nov  5 11:11 opt -> /BINONFS/opt
dr-xr-xr-x 58 root root  8  Nov  5 11:18 proc
drwxr-sr-x  7 root root 198  Nov  5 16:11 ramdisk
lrwxrwxrwx  1 root root  13  Nov  5 11:11 root -> /BINONFS/root
lrwxrwxrwx  1 root root  13  Nov  5 11:11 sbin -> /BINONFS/sbin
drwxr-xr-x 11 root root  8  Nov  5 11:18 sys
lrwxrwxrwx  1 root root  17  Nov  5 11:11 ttyboot -> /BINONFS/ttyboot
lrwxrwxrwx  1 root root  12  Nov  5 11:11 tmp -> ramdisk/tmp
lrwxrwxrwx  1 root root  12  Nov  5 11:11 usr -> /BINONFS/usr
lrwxrwxrwx  1 root root  12  Nov  5 11:11 var -> /BINONFS/var
lrwxrwxrwx  1 root root  16  Nov  5 11:11 vmlinuz -> /BINONFS/vmlinuz
root@tty1:/#

```

Fixem-nos en els diferents colors amb què se'ns mostren els resultats (podem observar els colors esmentats a la nostra pantalla de l'ordinador a mesura que anem progressant en el taller de Knoppix): blau marí per als directoris, 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 en què segur que trobarem executables, per exemple `/usr/bin/` (per a fer-ho, executem `cd /usr/bin/`). El directori `/usr/` és en realitat un enllaç simbòlic sobre `/KNOPPIX/usr/`; podríem accedir, doncs, als seus continguts entrant-hi directament mitjançant l'ordre `cd`, o bé seguir-ne, mitjançant la mateixa ordre, la destinació real. Tanmateix, triem la primera opció, ja que quan ens trobem en un sistema UNIX instal·lat en un disc dur aquest directori serà un directori real:

Figura 3.13

```

riscv-xr-x 1 root root 6253 Mar 3 2001 xviddetect
riscv-xr-x 1 root root 27400 May 2 2006 xvidthune
riscv-xr-x 1 root root 10 Dec 18 2005 xview -> xloadimage
riscv-xr-x 1 root root 8552 May 2 2006 xvinfo
riscv-xr-x 1 root root 5036 Feb 6 2006 xvminitppm
riscv-xr-x 1 root root 28 Jun 1 2005 xmcviewer -> /etc/alternatives/xmcviewer
riscv-sr-x 1 root utmp 50636 Nov 1 2005 xmt
riscv-xr-x 1 root root 3331 Jul 26 2002 xaavr
riscv-xr-x 1 root root 21092 May 2 2006 xad
riscv-xr-x 1 root root 13040 Feb 6 2006 xadtoppm
riscv-xr-x 1 root root 2 Feb 17 2006 xae -> ue
riscv-xr-x 1 root root 24492 May 2 2006 xainfo
riscv-xr-x 1 root root 20912 Apr 24 2006 xaintoppm
riscv-xr-x 1 root root 2 Feb 17 2006 xaue -> ue
riscv-xr-x 1 root root 22176 May 2 2006 xaud
riscv-xr-x 1 root root 11624 Feb 12 2006 xod
riscv-xr-x 1 root root 14772 Apr 21 2006 xzoom
riscv-xr-x 1 root root 45000 Nov 9 2005 yacas
riscv-xr-x 1 root root 3560 Nov 9 2005 yacas_client
riscv-xr-x 1 root root 22 Jun 1 2005 yacc -> /etc/alternatives/yacc
riscv-xr-x 1 root root 237696 Jan 4 2006 yafrau
riscv-xr-x 1 root root 4516 Feb 6 2006 ybstoppm
riscv-xr-x 1 root root 206400 Apr 27 2006 yelp
riscv-xr-x 1 root root 12640 Mar 2 2006 yes
riscv-xr-x 1 root root 795916 Jan 9 2006 yorick
riscv-xr-x 1 root root 10344 May 1 2006 ypcat
riscv-xr-x 1 root root 17 May 4 2006 ypchfn -> /usr/bin/yppasswd
riscv-xr-x 1 root root 17 May 4 2006 ypchsh -> /usr/bin/yppasswd
riscv-xr-x 1 root root 8200 May 1 2006 ypmatch
riscv-xr-x 1 root root 10600 May 1 2006 yppasswd
riscv-xr-x 1 root root 11104 May 1 2006 ypswich
riscv-xr-x 1 root root 6012 Feb 6 2006 yuvsplitppm
riscv-xr-x 1 root root 4968 Feb 6 2006 yuvtoppm
riscv-xr-x 1 root root 9500 Apr 14 2006 zdump
riscv-xr-x 1 root root 5216 Feb 6 2006 zeisstoppm
riscv-xr-x 1 root root 63100 Apr 21 2006 zenity
riscv-xr-x 1 root root 99036 Mar 16 2006 zile
riscv-xr-x 1 root root 66200 May 25 2005 zip
riscv-xr-x 1 root root 26616 May 25 2005 zipcloak
riscv-xr-x 1 root root 1188 Apr 25 2006 zipgrep
riscv-xr-x 1 root root 110100 Apr 25 2006 zipinfo
riscv-xr-x 1 root root 23544 May 25 2005 zipnote
riscv-xr-x 1 root root 26504 May 25 2005 zipsplit
riscv-xr-x 1 root root 1411 Jan 16 2004 zipwapper
riscv-xr-x 1 root root 607 Mar 30 2006 zonetabzpot.py
riscv-xr-x 1 root root 30400 Sep 21 2005 zsoelim
riscv-xr-x 1 root root 1037 Feb 15 2006 zxpdt
root@tty1(bin)

```

Probablement no haguem pogut veure més que les últimes línies de les que s'han mostrat a la pantalla. Podem visualitzar alguns resultats més prement "Shift+RePág" per a retrocedir en llista i "Shift+AvPág" per a avançar-hi. Tot i així, el *buffer* del terminal té un límit i probablement tampoc no podem visualitzar, mitjançant aquesta tècnica, tota la informació tornada per l'ordre `ls`. Així doncs, hem de recórrer a altres tècniques, el readreçament de la sortida (en aquest cas sobre un fitxer) o l'ús de canonades o *pipes* "|" i paginadors (l'ess en aquest cas, que usa les tecles "RePág" i "AvPág" per a desplaçar el contingut mostrat per pantalla, i la tecla "q" per a sortir-hi). Utilitzarem aquesta última, ja que no ens interessa gens desfer una llista del contingut d'un directori; aprofitarem, així mateix, per a posar en pràctica una altra utilitat que ens estalviarà teclejar: en prémer els cursors de dalt a baix ens podem desplaçar per totes les línies d'ordres que hem passat al sistema; així doncs, per a obtenir la línia volguda, `ls -la | less`, n'hi haurà prou que premem una vegada el cursor de dalt i afegir-hi `| less`.

`less` és un paginador molt potent que KNOPPIX –i la majoria de les distribucions– usa per a mostrar els continguts dels *man* (encara que, com la majoria dels serveis a Linux, el podem configurar al nostre gust). Així doncs, podem fer un `man less` per a informar-nos una mica sobre aquesta ordre i, de passada, anar-nos acostumant a la morfologia d'aquesta ajuda. Òbviament, ja que som sota `less`, per a sortir de *man* n'hi ha prou que premem 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 sub-directoris del directori creat, i contenen informació molt especial. El directori ".", que es coneix com el directori actual, fa referència al mateix directori (de manera que si fem un `ls -la .`, obtindrem la mateixa informació que fent un `ls -la`). I el directori "..", que es coneix com el directori pare, fa referència al directori pare del directori on som (en fer un `ls -la ..` obtindrem la llista de continguts del directori pare o el directori immediatament superior).

Activitat

3.1. Observeu el *man* de l'ordre `ls` i enteneu quines funcions tenen els paràmetres `-a` i `-l`.

Recordem on som dins del sistema de fitxers? No? Doncs l'ordre `pwd` ens ho recordarà. Tornem al directori arrel. Hi ha bàsicament dues maneres de fer-ho, i una altra d'especial per a aquest cas; de qualsevol manera, totes es basen en l'ús de l'ordre `cd`, i es diferencien entre elles per l'argument que li passem en cadascun dels casos.

La primera manera d'anar-hi és pujant en els directoris pare mitjançant `cd ..` (atenció, el sistema no entén `cd..`, ja que interpreta que estem intentant executar una ordre anomenada `cd..` que no existeix) iterativament fins a arribar al directori arrel (ho veurem en el *prompt*).

La segona és més eficaç, ja que amb una sola línia d'ordres aconseguirem el nostre propòsit, anar al directori arrel. Per a això, hem de passar el directori arrel com a argument a l'ordre `cd` (`cd /`); aquesta segona opció és molt més potent i com que permet passar directament a directoris que pertanyen a diferents branques, podríem haver fet, per exemple, `cd /etc/rcS.d/` (l'última barra és opcional) per a anar a aquest directori, si fóssim a `/usr/bin/`.

Com hem dit, per a aquest cas (anar al directori arrel) hi ha una tercera opció per a aconseguir la mateixa finalitat i amb variacions: es tracta d'executar l'ordre `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 les distribucions, i segons el Filesystem Hierarchy Standard, *root* ha de tenir un *home* propi `/root`), el directori home de home; una variació d'aquest mètode és "`cd`", ja que "`"` és equivalent al *home* de l'usuari, el directori arrel en aquest cas. Encara hi ha una altra opció per a tornar al directori arrel, o millor dit, per a 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

Una vegada examinades diferents opcions per a fer el mateix, ens trobem encara al 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 arrencar KNOPPIX, i que el mateix sistema, en arrencar, l'ha automuntat (més endavant aprendrem com muntar i desmuntar dispositius).

Recordem que en UNIX els dispositius abans de poder-se usar s'han de muntar, i que abans de retirar el suport s'han de desmuntar. Aquest últim punt és extremadament important, ja que si no, la integritat de les dades no està garantida en absolut. Podem, per exemple, abans de procedir a treballar sobre el disquet, intentar retirar el CD-ROM prement el botó d'expulsió. Sorpresa! no es pot; no hem de pensar que s'ha espatllat el dispositiu i molt menys GNU/Linux. Simplement succeeix que aquest dispositiu també ha estat muntat automàticament pel sistema durant l'arrencada, i que, per tant, n'ha passat a formar part; no tindria cap sentit que poguéssim retirar el CD-ROM sense abans informar-ne el sistema; així doncs, aquest ha pres el control d'aquest dispositiu i, entre altres coses, ha deshabilitat el botó d'expulsió del CD-ROM, precisament a fi que accidentalment aquest sigui retirat sense abans ser desmuntat. No succeeix el mateix amb la disquetera: en aquest dispositiu l'expulsió es fa de manera purament mecànica, així que és impossible que el sistema pugui impedir que nosaltres expulsem el disquet sense informar-lo abans, però això no és recomanable, tal com ja s'ha dit, perquè pot suposar la pèrdua de totes les dades que poguéssis contenir. Així doncs, mentre no aprenguem a muntar i desmuntar dispositius de suport de dades, el disquet ha de romandre a la disquetera des que el sistema arrenca fins que s'atura, ja que durant el procés d'aturada, entre altres operacions, s'efectuaran les de desmuntatge d'aquests dispositius. Passem al disquet: però on és? On l'ha muntat el sistema? Per a respondre aquestes preguntes executem l'ordre `mount` sense arguments ni opcions addicionals (precisament aquesta és l'ordre que s'utilitza per a muntar dispositius, i `umount` per a desmuntar-los):

Figura 3.14

```

root@tty1[bin]# mount
/dev/root on / type ext2 (rw)
/ramdisk on /ramdisk type tmpfs (rw,size=311728k)
/UNIONFS on /UNIONFS type unionfs (rw,dirs=/ramdisk=rw:/KNOPPIX=ro:/KNOPPIX2=ro,delete=whiteout)
/dev/hdc on /cdrom type iso9660 (ro)
/dev/loop on /KNOPPIX type iso9660 (ro)
/dev/loop2 on /KNOPPIX2 type iso9660 (ro)
/proc on /proc type proc (rw)
/proc/bus/usb on /proc/bus/usb type usbfs (rw,devmode=8666)
/devpts on /devpts type devpts (rw)
root@tty1[bin]# _

```

Aquesta ordre executada sense arguments ens mostra els dispositius muntats en el sistema en el moment d'executar-lo i alguna informació addicional sobre ells (aquesta mateixa informació es pot trobar en el fitxer `/etc/mtab`; en conseqüència, en fer `cat /etc/mtab` [per a mostrar el contingut de `mtab`] aconseguiríem els mateixos resultats). En la segona línia retornada per l'ordre ja podem veure que, efectivament, el CD-ROM, `/dev/CDROM`, està muntat en el sistema, i a més podem veure que està muntat sobre el directori `/CD-ROM/`. L'última línia retornada respon a 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 està 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 hi 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 –tot i que ho fa si el fitxer no existeix–, sinó que serveix per a canviar les informacions relatives a dates i hores dels arxius. Podem posar algun contingut en el nostre primer fitxer i comprovar que efectivament ha quedat enregistrat:

```
root@tty1[subdir00]# echo "my first hello world in Linux" > file00
root@tty1[subdir00]# cat file00
my first hello world in Linux
```

Hi afegim una mica més de text:

```
root@tty1[subdir00]# echo "Nice to meet you, we're gonna be good friends" >> file00
root@tty1[subdir00]# cat file00
my first hello world in Linux
Nice to meet you, we're gonna be good friends
```

No ens oblidem d'usar els cursors per a estalviar-nos teclejar, ni de l'autocompletat. Ara podem utilitzar un editor de text per a crear el nostre segon fitxer. Per a això usarem l'editor `vi`. Recordem que aquest editor té dos modes: el d'ordres, al qual s'entra quan s'arrenca, i el mode d'edició. Per a entrar al mode d'edició n'hi ha prou que premem la tecla `"i"`, i per a passar al mode d'ordres cal polsar-hi `"Esc"`. Si volem desar, ens hem de posar en mode ordres i escriure-hi `:w` i `:q` per a sortir (vi disposa de moltes més ordres, però per ara amb aquestes dues n'hi ha prou). És molt interessant conèixer aquestes ordres bàsiques de `vi`, ja que aquest editor hi és a gairebé tots els paquets bàsics d'instal·lació de qualsevol distribució, i si aquesta fallés en algun moment, ens pot ser útil per a modificar algun fitxer i prosseguir aquesta instal·lació.

Executem l'ordre `vi` seguida del nom que volem donar al nostre segon fitxer, premem la tecla `"i"`, escrivim el que ens sembli oportú, premem `"Esc"` i `:wq` per a desar i sortir. Mitjançant `more`, un altre paginador, comprovem que tot ha sortit com esperàvem:

```
root@tty1[subdir00]# vi file01
it seems we're on the right way, mate!!!
I agree.
~
~
~
```

```
:wq
root@tty1[subdir00]# more file01
it seems we're on the right way, mate!!!
I agree.
```

Dispositius de memòria USB. L'USB pot connectar perifèrics com ratolins, teclats, escàners, càmeres digitals, PDA, etc. Però nosaltres ens centrarem en les memòries USB, ja que cada dia és més comú utilitzar aquest tipus de dispositius en comptes dels antics disquets. Aquests ens permeten emmagatzemar diversos centenars de megabytes d'informació en un mitjà molt petit i fàcil de transportar: els dispositius de memòria USB. Hi ha diferents sistemes, com el sistema *flash* o el sistema *memory stick*.

Aquests sistemes funcionen perfectament amb el nucli o *kernel* Linux. N'hi ha prou que els muntem com si fossin discos SCSI, normalment es trobaran a `/dev/sda1`, o si hi ha altres discos SCSI, a `/dev/sdb1` o `/dev/sdc1`.

Com ja hem vist, l'ordre `mount` ens permet muntar qualsevol sistema de fitxers, i amb `umount` els podem desmuntar. Per a saber quants ha muntat i com els ha muntat, hem d'executar l'ordre `mount` sense cap opció. La informació l'obté del fitxer `/etc/mstab`, ja que en aquest fitxer tenim la llista dels sistemes de fitxers muntats en aquell moment. De fet, són els programes `mount` i `umount` els que mantenen aquesta llista en el fitxer `/etc/mstab`.

Com l'ordre següent:

```
knoppix@0[knoppix]$ mount
/dev/root on / type ext2 (rw)
/ramdisk on /ramdisk type tmpfs (rw,size=401312k)
/UNIONFS on /UNIONFS type unionfs (rw,dirs=/ramdisk=rw:/KNOPPIX=ro:/KNOPPIX2=ro,delete=whiteout)
/dev/hdc on /cdrom type iso9660 (ro)
/dev/cloop on /KNOPPIX type iso9660 (ro)
/dev/cloop2 on /KNOPPIX2 type iso9660 (ro)
/UNIONFS/dev/pts on /UNIONFS/dev/pts type devpts (rw)
/proc/bus/usb on /proc/bus/usb type usbfs (rw,devmode=0666)
automount(pid2320) on /mnt/auto type autofs (rw,fd=4,pgrp=2320,minproto=2,maxproto=4)
/UNIONFS/dev/sda1 on /mnt/sda1 type vfat (rw,nosuid,nodev,umask=000,user=knoppix)
```

El fitxer `/etc/fstab` és el que conté quins dispositius es munten usualment, on i com (quines són les seves opcions). Però quan muntem altres dispositius o desmuntem el dispositiu que sigui, això es reflecteix en el fitxer `/etc/mstab`, que és el que ens descriu quins sistemes de fitxers tenim muntats en aquell moment.

Activitat

3.2. Com muntarem el llapis USB o memòria USB perquè qualsevol usuari hi pugui escriure? En cas que vulguem que aquest tipus de memòries USB es muntin per defecte (en

arrencar el sistema) i que tinguin els permisos de lectura i escriptura per a tots els usuaris, quina línia hi hauríem d'afegir?

Activitat

3.3. Repetiu totes les accions de l'activitat 3.2 de maneig de directoris i fitxers però en comptes d'utilitzar el disquet ho farem en una memòria USB.

Ara bé, des del disquet o des del llapis USB intentem esborrar el nostre primer fitxer. Ho farem mitjançant l'ordre `rm`:

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

La veritat és que aquesta pregunta per part del sistema no ens l'esperàvem. Premem "Ctrl+C" per cancel·lar, "n" per no esborrar o "i" per fer-ho. L'important ara és que entenguem per què el sistema ens ha formulat aquesta pregunta. Si llegim el *man* de l'ordre `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 el vam invocar, i prosseguir així la nostra feina en la *tty* on érem). Però ens fixem en l'explicació de l'opció `-i` d'aquest mateix *man*. Què està passant? Sembla com si aquesta opció estigués activada per defecte. En realitat, no és exactament així, el que succeeix és que el sistema per defecte ha establert uns àlies; per tal de veure'ls tots, executem l'ordre *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 de l'ordre `ls` és en colors, o que per a fer un `ls -la` n'hi ha prou que teclegem `la`. I també entenem per què, quan hem executat l'ordre `rm` anterior, el sistema ens ha preguntat si realment ho volíem fer. Mitjançant àlies podem establir i modificar el comportament per defecte d'ordres o fins i tot crear-ne altres de nous:

```
root@tty1[subdir00]# alias hi='echo "I say hello"'
root@tty1[subdir00]# hi
```

```
I say hello
```

Continuant amb el *man* de l'ordre `rm`, ens disposem a llegir per a què serveixen les opcions `-f` i `-r`. La primera serveix perquè l'ordre s'executi sense mostrar *prompt*, o el que és el mateix, desobeint l'opció `-i`, així que, si no haviem 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 les ordres bàsiques destinades a la manipulació de directoris i fitxers; així doncs, podem crear un segon directori a l'arrel del disquet amb tots els continguts del primer que hem creat mitjançant l'ordre `cp`; per a fer-ho, hem de recórrer a la recursivitat:

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

Activitat

3.4. Repetiu aquesta còpia recursiva del `/dir01` del disquet al llapis o memòria USB.

En aquest cas, hem usat l'adreçament absolut (tant per a especificar-ne l'origen com la destinació) per a fer l'operació de còpia. 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 l'adreçament relatiu per a especificar l'origen de l'operació, la seva destinació o ambdues coses. Quan usem l'adreçament relatiu, l'origen de l'adreçament és la posició actual dins del filesystem. Com en la majoria dels casos, el sistema ens ofereix la possibilitat d'obtenir els mateixos resultats emprant diferents mètodes. Es tracta de conèixer-ne com més millor, i saber-ne triar 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/`, el qual també volem situar dues branques per sota de la nostra posició. Així mateix, hauria estat perfectament vàlida la línia `cp -r ../../../../dir02/` per a assolir les mateixes finalitats.

Activitat

3.5. Expliqueu el perquè d'aquesta última asseveració.

Ara podem descendir un subdirectori, amb la qual cosa ens situarem a `/mnt/auto/floppy/dir00/` i podem 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 vàlida, entre moltes altres opcions, `./subdir00/file00`) i, en segon lloc, cal especificar-ne obligatòriament la destinació, que és la posició actual,

és a dir ".". Mitjançant un `ls` podem comprovar si hem obtingut el resultat que volíem. Ara ens podem situar en el directori `pare`, en el qual 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-nos teclejar:

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

Activitat

3.6. Llegiu el *man* de `rm` i enteneu 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 `home`, atès que ja tenim molt clar que aquest usuari només s'ha de fer servir en cas 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 les ordres bàsiques i amb el seu sistema de fitxers. Ha arribat el moment de començar a treballar com cal fer-ho sempre a partir d'ara, és a dir, utilitzant el compte de `home` només quan és estrictament necessari, com en el primer pas que farem a continuació, crear un nou usuari del sistema i assignar-li un compte, operació que només la pot fer el `home`. 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 aquesta ordre que acabem d'executar. No sabem, doncs, ni a quin grup o grups pertany aquest usuari, on té el directori `home`, quin *shell* se li ha assignat per defecte, etc. Molta d'aquesta informació la trobarem en el fitxer de contrasenyes (`/etc/passwd`), així que analitzarem, per camps, el seu 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 desitjada la trobem en l'última línia del fitxer, ja que el nostre usuari ha estat l'últim a afegir-s'hi. Analitzem el contingut d'aquesta línia:

- `user00` és el nom de l'usuari, que ja sabíem perquè l'hem creat nosaltres mateixos.
- `x` la seva contrasenya es troba en el fitxer de shadow, `/etc/shadow`.
- `1.001` és el seu número d'identificació com a usuari (UID).
- `100` és el número d'identificació del seu grup (GID).
- No té cap comentari associat.
- `/home/user00` és el seu directori home, tal com era d'esperar, ja que tots els usuaris tenen el seu *home* a `/home/usuario`.
- `/bin/bash` el *shell* amb què es trobarà l'usuari en iniciar la sessió és el *bash*, el mateix que hem utilitzat fins ara.

La pregunta que ens formulem immediatament davant d'aquestes informacions és relativa a la contrasenya de l'usuari. Hem de consultar, doncs, el fitxer `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 l'ordre `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 una canonada o *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 l'hagués tingut, no el podríem conèixer, ja que tal com s'ha dit, les contrasenyes es guarden encriptades i l'encriptació és unidireccional. Però això tampoc no ens hauria de preocupar, ja que `home`, mitjançant l'ordre `passwd`, pot canviar la contrasenya de qualsevol altre usuari. Així doncs, l'únic que realment hem d'intentar és no oblidar la contrasenya de `home`, que tampoc no té contrasenya en aquest cas, tal com `shadow` ens indica, encara que si això succeís, també té solució. Continuem amb la nostra recerca sobre les propietats del nostre nou usuari. `/etc/group` ens ajudarà a saber a quins grups pertany exactament:

```
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 d'ordres ens preguntem, amb resposta negativa, si `/etc/group` té alguna referència explícita a `user00` (també es podria haver fet mitjançant l'ordre `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 seus usuaris al primer. Però aquesta pràctica no és massa recomanable, ja que hi ha ordres específiques 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. Per tant, l'hem de crear i assignar-hi les propietats i els 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-x 2 user00 users 60 Mar 4 08:12 user00
```

Nous usuaris a Debian

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

Activitat

3.7. Mitjançant `man`, compreneu el funcionament de les ordres `chown` i `chgrp`.

Ha arribat el moment d'entrar en el sistema com a nou usuari. Ho farem mitjançant l'ordre `su`, la qual obre un procés fill amb un `login` amb l'usuari que li haguem 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` aquest no és necessari; a més, es dóna el cas que

l'usuari que hem creat no té contrasenya i, per tant, no se'n demanarà mai. 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]$
```

Primerament, hem de notar el canvi d'aspecte del *prompt*. Hem deixat de ser usuari *home*, per a 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 en el nostre directori *home*, hi creem un fitxer i en canviem els atributs perquè tan sols *user00* el pugui llegir, executar i escriure-hi. De moment, el permís d'execució no té molt de sentit per a nosaltres, ja que no sabem encara crear fitxers executables, però és bo conèixer-lo per endavant per a quan el necessitem:

```
user00@tty1[home]$ cd
user00@tty1[user00]$ echo "only user00 can read, write and execute this file." > user00file
user00@tty1[user00]$ chmod 700 user00file
```

Activitat

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

3.8. Gestió de processos

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

```
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 en el qual s'executa; naturalment, el primer procés sempre correspondrà al *shell*.

Podríem fer 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 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
```

Ara ens podem preguntar per tots els processos que estan corrent 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 l'ordre `top`, que ens mostra l'activitat de la CPU, l'ús de memòria, etc. de manera interactiva.

Ara arrencarem un procés, l'aturarem i l'enviarem a executar en *background*. Per a fer-ho, farem un `sleep 20` i premerem la combinació de tecles "Ctrl+Z" abans de 20 segons, per a aturar-lo, `jobs` per assegurar-nos que efectivament està aturat i conèixer el seu número d'identificació, i `bg` al costat del seu número d'identificació per a fer-lo executar en mode *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[/]#
```

Ens trobem que, un cop transcorreguts els 20 segons, el procés s'ha acabat trobant encara en *background*, i se'ns n'ha informat per pantalla. Podem comprovar mitjançant `ps` que efectivament no hi ha cap procés corrent. En aquest punt, podem aturar un procés i tornar-lo a *foreground* mitjançant `fg`:

```
root@tty1[/]# man ls
<Ctrl+z>
[1]+ Stopped man ls
root@tty1[/]# jobs
[1]+ Stopped man ls
root@tty1[/]# fg 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 perquè `gpm` és un *daemon*, terme que s'analitza amb detall en seccions posteriors).

Hi ha molts tipus de ratolins (mecànics, òptics, làser i *trackball*) i molts tipus de connexió (port sèrie, port PS/2, USB, làser, sense fil: radiofreqüència, infraroig o *bluetooth*), encara que actualment els més utilitzats són els que tenen el tipus de connector USB o el tipus PS/2 (normalment es distribueixen amb els dos tipus de connectors). Si tenim un ratolí del primer tipus, executarem la línia següent:

```
#gpm -m /dev/input/mice -t autops2
```

Si el que tenim és un ratolí connectat al port PS/2, per a activar-lo s'usarà el mateix procediment anterior i es variarà únicament el *device* i el tipus de ratolí:

```
#gpm -m /dev/psaux -t ps2
```

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

```
sofia@tty1[/]$ ps aux | grep gpm
```

n'hauríem d'obtenir una resposta del tipus

```
root 3594 0.0 0.1 1632 532 ? Ss 00:28 0:00 gpm -m
/dev/input/mice -t autops2
```

conforme `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 a tractar d'identificar el tipus de ratolí que s'adapti al que tenim.

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

També podem configurar el `gpm` mitjançant l'ordre `gpmconfig` o la utilitat `dpkg-reconfigure gpm`.

Activitat

3.9. Com configurariem el *touch pad* perquè emuli un ratolí de dos botons?

Cal destacar que el contingut del *buffer* del ratolí es conserva en passar d'una *tty* a una altra.

Activitat

3.10. Aprofitant que el contingut del *buffer* del ratolí es manté entre terminals, obriu en una sessió de `vi` i en una altra situeu-vos a l'arrel del sistema de fitxers, feu-hi una llista dels seus continguts amb detall i porteu aquestes dades a l'editor de text. Deseu els continguts del fitxer de text i mitjançant `cat` comproveu que efectivament hem obtingut els resultats que volíem.

3.10. Altres operacions

Per acabar aquest primer taller, executarem una sèrie d'ordres per a seguir el procés de familiarització amb el sistema i anar adquirint 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 tenim instal·lat, etc.

Així que aquí teniu unes quantes ordres; no oblideu que amb `man` i el nom de l'ordre teniu una ajuda que explica per a què serveixen i totes les opcions que poden tenir.

```
knoppix@0[knoppix]$ uname -a
Linux Knoppix 2.6.12 #2 SMP Tue Aug 9 23:20:52 CEST 2005 i686 GNU/Linux
```

```
knoppix@0[knoppix]$ cat /proc/cpuinfo
processor      : 0
vendor_id    : GenuineIntel
cpu family   : 15
model        : 2
model name   : Intel(R) Pentium(R) 4 CPU 2.80GHz

stepping     : 7
cpu MHz      : 2801.835
cache size   : 512 KB
fdiv_bug     : no
hlt_bug      : no
f00f_bug     : no
coma_bug     : no
fpu          : yes
fpu_exception : yes

cpuid level  : 2
wp           : yes
flags        : fpu vme de pse tsc msr pae mce cx8 sep mtrr pge mca cmov pat pse36 clflush dts
acpi mmx fxsr sse sse2 ss ht tm pbe cid
bogomips    : 5455.87

knoppix@0[knoppix]$ lspci
0000:00:00.0 Host bridge: Silicon Integrated Systems [SiS] SiS645DXHost & Memory & AGP
Controller (rev 01)
0000:00:01.0 PCI bridge: Silicon Integrated Systems [SiS] Virtual PCI-to-PCI bridge (AGP)
0000:00:02.0 ISA bridge: Silicon Integrated Systems [SiS] SiS962 [MuTIOL Media IO] (rev 14)
0000:00:02.1 SMBus: Silicon Integrated Systems [SiS]: Unknown device 0016
0000:00:02.3 FireWire (IEEE 1394): Silicon Integrated Systems [SiS] FireWire Controller
0000:00:02.5 IDE interface: Silicon Integrated Systems [SiS] 5513 [IDE]
0000:00:02.6 Modem: Silicon Integrated Systems [SiS] AC'97 Modem Controller (rev a0)
0000:00:02.7 Multimedia audio controller: Silicon Integrated Systems [SiS] Sound Controller (rev a0)
0000:00:03.0 USB Controller: Silicon Integrated Systems [SiS] USB 1.0 Controller (rev 0f)
0000:00:03.1 USB Controller: Silicon Integrated Systems [SiS] USB 1.0 Controller (rev 0f)
0000:00:03.2 USB Controller: Silicon Integrated Systems [SiS] USB 1.0 Controller (rev 0f)
0000:00:03.3 USB Controller: Silicon Integrated Systems [SiS] USB 2.0 Controller
0000:00:04.0 Ethernet controller: Silicon Integrated Systems [SiS] SiS900 PCI Fast Ethernet (rev 91)
0000:00:08.0 CardBus bridge: O2 Micro, Inc. OZ6912 Cardbus Controller
0000:01:00.0 VGA compatible controller: ATI Technologies Inc Radeon R250 Lf
[Radeon Mobility 9000 M9] (rev 01)

knoppix@0[knoppix]$ df
```

```

S.ficheros      Bloques de 1K  Usado  Dispon  Uso%  Montado en
/dev/root       2471          21     2450   1%   /

/ramdisk        401312        8388   392924   3%   /ramdisk
/UNIONFS        7764900      7371976  392924   95%  /UNIONFS
/dev/hdc        4067388      4067388    0  100% /cdrom
/dev/cloop      4963828      4963828    0  100% /KNOPPIX
/dev/cloop2     2399762      2399762    0  100% /KNOPPIX2
/UNIONFS/dev/sda1 255452      14552   240900    6%  /mnt/sda1

knoppix@0[sda1]$ free

      total        used         free   shared    buffers   cached
Mem:      514720     495232     19488         0       15476    350084
-/+ buffers/cache: 129672    385048
Swap:      497972         0     497972

```

Activitat

3.11. 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 per a trencar el mite que treballar en aquest tipus de sistemes operatius és complicat i difícil. És més, ha servit perquè comencem a intuir la seva potència i versatilitat. Hem après a moure'ns pel seu sistema de fitxers i a fer operacions amb ells, 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 provoca que aquest ens pugui ser molt útil a l'hora d'instal·lar Debian en el proper taller si sorgissin problemes de maquinari. Sempre podrem tornar a arrencar KNOPPIX i estudiar com ha solucionat el problema (bàsicament estudiant els arxius de configuració).

Com a última activitat del taller, podem tornar a arrencar KNOPPIX en mode gràfic, és a dir, sense passar 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 apartat veurem els passos essencials que se segueixen en la majoria dels processos d'instal·lació de GNU/Linux. Encara que cada distribució té el seu propi entorn d'instal·lació, en totes elles hi ha d'haver uns passos bàsics. En aquest apartat, 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 sistemes operatius o més totalment independents. Encara que el procés d'instal·lació d'un altre sistema operatiu en el mateix disc (o en un altre d'instal·lat a l'ordinador) no hauria d'interferir amb les particions dels altres, és aconsellable fer còpies de seguretat de tots els nostres documents. Malgrat que si seguim adequadament els passos que detallarem a continuació és pràcticament impossible perdre informació, sempre és recomanable la prudència i fer còpies dels arxius que realment ens importen.

Preparació de la instal·lació de Linux

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

4.2. Arrencada

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 hi vulguem fer una instal·lació remota, es proporciona un disquet d'arrencada.

El primer pas del procés sol ser triar quin tipus d'instal·lació volem efectuar. Generalment, l'elecció sol ser per a identificar quin tipus d'usuari està instal·lant el sistema per a, segons això, proporcionar-li més o menys informació i deixar-li configurar amb més precisió o menys precisió el sistema. Si tenim prou coneixements, 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 volem donar al sistema.

Seguidament, la majoria de les distribucions ens deixen triar 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 la qual cosa hauríem de seleccionar *qwerty/es* (Spain).

Mètodes d'instal·lació

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.

4.3. Particionar 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 cadascuna d'elles 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.

Elecció del sistema de fitxers

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

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'haguem d'eliminar i crear de nou implica perdre tota la informació que hi tenim. Per aquest i altres motius, hi ha programes que ens permeten modificar la mida de les particions sense haver-les d'eliminar. El *fdisk* és un programa amb llicència GPL que ens permet redimensionar les nostres particions formatades amb FAT (per a sistemes WindowsTM de la branca no NT) sense perdre'n la informació. També hi ha 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'hauréu d'utilitzar algun abans de començar amb tot el procés.

Particions del disc

Si bé amb una o dues particions és suficient per a poder instal·lar GNU/Linux, és interessant dividir el disc en més fragments i situar certs directoris en diferents unitats per a poder efectuar 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.

És recomanable que GNU/Linux utilitzi dues particions en el disc dur. Una servirà per a desar-hi els fitxers del sistema i l'altra, per a la *swap*. La *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. En aquest moment és quan es comença a utilitzar la *swap* per a desar-hi zones de RAM que no s'estan utilitzant, i les intercanvia 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 de la *swap* pot ser tan gran com vulguem, es recomana que sigui el doble que 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 seu rendiment decreix molt, la qual cosa provoca que el sistema quedi pràcticament saturat (per a veure quina quantitat de memòria RAM i *swap* s'està utilitzant, el sistema ens proporciona l'ordre `free`).

Hi ha diverses aplicacions per a fragmentar el disc. Una de les primeres que va aparèixer va ser `fdisk`, encara que actualment n'hi ha d'altres, com `cdisk`, `diskDruid`, etc. En alguns processos d'instal·lació, es pot triar quina volem utilitzar, encara que totes permeten 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 i els Serial ATA, en els quals, en ambdós casos, la *X* és una lletra, corresponent al disc al qual ens vulguem referir de la manera següent:

Taula 4.1

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
<code>/dev/sda</code>	Primer disc de la controladora SCSI o SATA
<code>/dev/sdb</code>	Segon disc de la controladora SCSI o SATA

Si tenim més d'un disc al nostre ordinador, abans d'entrar en el programa de fraccionament podrem triar sobre quin d'ells 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 64 de lògiques. Si no necessitem més de 4 particions, podem triar 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 16 per a cadascuna), de manera que no podem tenir 4 particions

Classes de discos durs

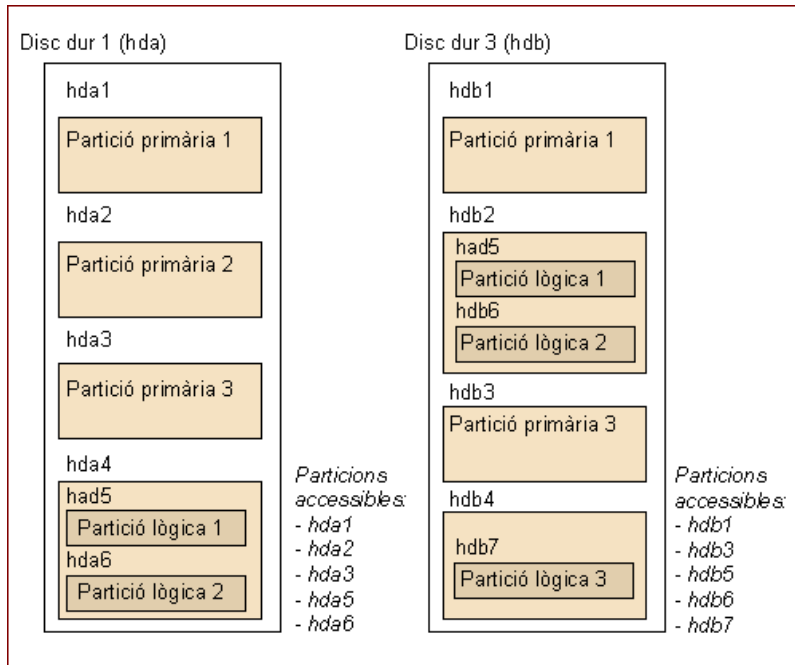
Per als ordinadors personals hi ha tres classes de discos durs: els IDE, els Serial ATA i els SCSI. En l'actualitat, la majoria de les plaques base porta controladores Serial ATA. Les controladores SATA permeten fins a quatre o sis discos durs. Les controladores SCSI (*small computer system interface*) permeten fins a 8 dispositius i tenen taxes de transferència més altes, encara que el seu preu també és bastant més elevat que els IDE.

Master boot record

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*).

primàries creades i després afegir-ne altres de lògiques. En aquest cas, hauríem de crear 3 de primàries i fins a 16 lògiques en la quarta partició primària. En la figura següent, podem veure un exemple de manera gràfica:

Figura 4.1



Quan creem una partició, hi hem d'indicar quin sistema de fitxers utilitzarem per a aquesta (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 la *swap* del sistema. Una vegada fets aquests passos, ja podrem 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-hi de manera estàndard tots aquests *drivers* i funcions. En les primeres versions de Linux, quan el nucli que teníem no incloïa alguna funció, l'havíem de recompilar 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 s'hi facilita ja sabrem 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 podrem triar amb més precisió

Partició activa

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.

els mòduls que necessitem. En alguns d'ells és obligatori passar algun paràmetre especial (com l'adreça 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 comentàvem.

També és veritat 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 de manera automàtica. Tanmateix, és possible que no es detecti adequadament algun dels dispositius que tinguem instal·lats, en aquest cas sí que hi hem d'incloure manualment el mòdul corresponent.

Si en el moment d'instal·lar el sistema ens oblidem d'incloure-hi algun mòdul (o hi instal·lem algun nou dispositiu), sempre podem recórrer a les ordres `insmod` (per a afegir-hi un nou mòdul), `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 solem trobar en el directori `/libmodules/` 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*. Amb aquesta, podrem saber exactament quins components tenim i quins mòduls¹ els corresponen.

⁽¹⁾ El fitxer de configuració dels mòduls se sol trobar a `/etc/modules/`.

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 detall sobre xarxes de computadors, aportarem 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 sempre la seva 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 arrenquen. Si utilitzem aquest mecanisme, ho hem d'indicar, i si no, se'ns preguntarà per la IP i 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 del *gateway* de la nostra xarxa. El *gateway* 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 els 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.

Adreça IP

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.

Si som en una xarxa local, hem de consultar el seu administrador perquè ens en proporcioni tota la informació necessària. Si tenim un altre sistema operatiu instal·lat a l'ordinador, també podem accedir a la seva configuració per 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 hi tenim instal·lats volem arrencar. Encara que només hi haguem instal·lat GNU/Linux, també hi haurem d'instal·lar aquest programa, ja que el seu sistema d'arrencada el necessita.

Hi ha 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 en els 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 inclou 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 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 podrem escollir, de manera molt general i sense entrar en detalls, quin tipus de programes utilitzem.

Paquets

Un paquet és format per un o diversos programes/biblioteques relacionats entre ells, que s'agrupen per a formar un únic bloc. La majoria de les distribucions inclou 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. ho podem instal·lar tot alhora.

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, amb la qual cosa ens estalviarem espai al disc i evitarem 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 triar 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 o DVD, des de servidors a Internet, etc. Debian GNU/Linux va ser la primera a tenir un sistema d'aquest tipus, denominat `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 fer si hem tingut algun problema amb la instal·lació o si 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 el sistema operatiu i accedir al sistema de fitxers per a arreglar el que calgui.

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 (si premem "Ctrl+F2", hi podem accedir) amb les ordres bàsiques perquè puguem fer operacions essencials de recuperació. De tota manera, si ens trobem amb algun inconvenient greu que no ens permet arrencar el sistema correctament i volem totes les eines usuales, també podem arrencar amb un *live-CD* de Knoppix o qualsevol altra distribució i muntar la unitat en què 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 dels processos d'instal·lació utilitza una aplicació denominada *bootstrap*, que també podem instal·lar en el sistema amb el paquet corresponent. Amb ella ens podríem crear el nostre propi procés d'instal·lació o veure com el sistema duu a terme, realment, aquestes operacions d'instal·lació i configuració. La majoria d'elles les podem reproduir amb el programa `base-config` o algun dels altres que ell mateix crida (podeu trobar més informació en el seu manual), encara que sempre podem recórrer als mateixos fitxers de configuració del sistema operatiu per a fer-ho manualment.

Actualització de RedHat

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

5. Taller d'instal·lació de Debian Etch

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 en el nostre ordinador, ja que s'arrencava i s'executava des de CD-ROM o DVD-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 GNU/Linux 4.0, coneguda com a Etch (la més recent en el moment de l'edició d'aquests materials), que es va publicar el 8 d'abril del 2007 després de 21 mesos de desenvolupament constant. Debian GNU/Linux és un sistema operatiu lliure que suporta un total d'onze arquitectures de processador.

Quant al programari, entre les novetats es troben els entorns d'escriptori GNOME 2.14, KDE 3.5.5a i Xfce 4.4; Ofimàtica OpenOffice.org 2.0.4a, KOffice 1.6, GNUcash 2.0.5, GNUMERIC 1.6.3 i Abiword 2.4.6; Coreo Evolution 2.6.3 i missatgeria Gaim 2.0; iceweasel 2.0.0.2 és el navegador Firefox, mentre que icedove 1.5 és el client de correu Thunderbird; ambdós programes són versions que no utilitzen les marques registrades.

També inclou programes criptogràfics, és compatible amb la versió 2.3 de l'FHS i amb aquells programes desenvolupats per a la versió 3.1 de l'LSB.

Debian GNU/Linux 4.0 introdueix una nova interfície gràfica del sistema d'instal·lació que suporta tant grafies que utilitzen caràcters compostos com llengües complexes. El sistema d'instal·lació de Debian GNU/Linux ara està traduït a 58 idiomes. A partir de Debian GNU/Linux 4.0, s'ha millorat la seguretat i l'eficiència del sistema de gestió de paquets.

Debian GNU/Linux s'executa en ordinadors que van des de PDA i sistemes de butxaca a supercomputadors, passant per qualsevol sistema pràcticament intermedi. Hi ha manteniment per a un total d'onze arquitectures: Sun SPARC (sparc), HP Alpha (alpha), Motorola/IBM/IBM PowerPC (powerpc), Intel IA-32 (i386) i IA-64 (ia64), HP PA-RISC (hppa), MIPS (mips, mipsel), ARM (arm), IBM S/390 (s390) i (per primera vegada introduït a Debian GNU/Linux 4.0) AMD64 i Intel EM64T (amd64).

La decisió de triar Debian enfront de 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 té aquestes causes principalment: en primer lloc, aquesta distribució és la que segueix més fidelment la filosofia GNU/Linux i tot es fa gràcies a la feina voluntària de milers de persones; en segon lloc, 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 i que disposa de 18.733 paquets. 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 com que som conscients del fort arrelament que té RedHat, s'ha dedicat un apèndix a aquesta distribució, perquè el lector la pugui conèixer també i se'n 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 d'elles, és tasca de cada usuari anar provant diferents distribucions i anar-se forjant una opinió pròpia de quin és el que s'adapta millor a les seves necessitats o exigències. Per aquest motiu, una vegada més animem a dur a terme tot tipus d'experiments i proves, i que cadascú dirigeixi els seus esforços cap a on cregui que és 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 aquesta llibertat sense cap tipus d'impediment, amb el benefici de tots els seus avantatges i assumint totes les seves conseqüències.

Cal remarcar que a Espanya, i en concret en el sistema educatiu, són moltes les comunitats autònomes que estan apostant pel programari lliure. Va començar Extremadura, amb GnuLinEx, una distribució GNU/Linux basada en Debian. I la van seguir Guadalinux a Andalusia, LliureX a la Comunitat Valenciana, Max a la Comunitat de Madrid, Molinux a Castella - la Manxa, etc. Encara que al principi totes es van basar en Debian, algunes d'elles han canviat i ara es basen en Ubuntu, una distribució GNU/Linux que en l'actualitat s'està fent molt popular.

Així que com que som conscients de la força que està assolint Ubuntu, s'ha dedicat un apèndix a aquesta distribució perquè el lector la pugui conèixer també i se'n pugui formar la seva pròpia opinió.

Activitat

5.1. Aquesta activitat pretén que visiteu les pàgines web que s'indiquen a dalt perquè conegueu les actuacions que s'estan duent a terme, a diferents comunitats autònomes, per a la introducció i el foment de l'ús del programari lliure en el sistema educatiu.

Activitat

5.2. Com que s'instal·larà Debian, és recomanable visitar la seva pàgina web i familiaritzar-se una mica amb els seus continguts. Així doncs, es proposa visitar-ne la web i els subapartats.

Una vegada més s'insta el lector a deixar-se guiar per la seva curiositat i, aquesta vegada, a seguir els enllaços que li semblin interessants.

5.1.1. Sistemes d'instal·lació

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

Debian ofereix diversos tipus d'instal·lació:

- Mitjançant un joc de CD-ROM o DVD-ROM que utilitza el `debian-installer`. Aquest conjunt de CD-ROM o DVD-ROM conté tots els paquets de Debian, per la qual cosa ens permet una instal·lació completa sense disposar d'una xarxa. Es pot baixar Debian GNU/Linux usant bittorrent (ara és el mètode recomanat), jigdo o HTTP/FTP.
- Mitjançant un CD-ROM. Hi ha dues imatges diferents d'instal·lació de xarxa per a CD-ROM ("netinst") que es poden utilitzar per a instal·lar Etch amb el `debian-installer`. Aquestes imatges estan dissenyades de manera que pugueu arrencar des del CD i instal·leu els paquets addicionals que vulgueu mitjançant la Xarxa, per això es diu "netinst". La diferència entre les dues imatges disponibles és que la imatge completa "netinst" conté l'instal·lador i el sistema base, mentre que la imatge de "targeta de visita" és més petita i no conté el sistema base, per la qual cosa l'ha de baixar des de la Xarxa. Podeu obtenir una imatge de CD completa que no necessiti disposar d'una xarxa per a poder dur a terme la instal·lació. Per a això només necessitareu utilitzar el primer CD del conjunt de CD de Debian.
- Mitjançant un disquet d'arrencada (`boot.img`), i quan arrenqui aquest disquet us demanarà que hi inseriu un segon disquet (`root.img`). Si feu la instal·lació mitjançant la Xarxa, necessitareu la imatge `floppy/net-drivers.img`, ja que aquesta conté controladors addicionals per a moltes targetes de xarxa Ethernet, i també inclou el suport per a PCMCIA. En el cas de dur a terme la instal·lació des de CD-ROM, i no poder arrencar des d'ell, llavors arrenqueu primer des d'un disquet i useu la imatge `floppy/cd-drivers.img` com a disc de controladors, per tal d'acabar la instal·lació usant el CD-ROM.
- Mitjançant un dispositiu de memòria USB. La manera més fàcil de preparar el vostre dispositiu de memòria USB és baixar `hd-media/boot.img.gz` i usar `gunzip` per a extreure'n la imatge. A continuació, munteu el dispositiu

tiu de memòria (tindrà un sistema de fitxers FAT) i, finalment, baixeu una imatge de CD "netinst" de Debian i copieu aquest fitxer al dispositiu de memòria (ha de ser `.iso`).

- Mitjançant la Xarxa. Els diferents mètodes d'arrencada de xarxa depenen de la seva arquitectura i configuració d'arrencada des de Xarxa (per exemple, mitjançant DHCP). La manera més fàcil de configurar l'arrencada des de xarxa probablement sigués amb PXE.
- Mitjançant un disc dur. És possible arrencar l'instal·lador sense usar mitjans extraïbles, però només si es disposa d'un disc dur existent, el qual pot tenir un sistema operatiu diferent. Baixeu `hd-media/initrd.gz`, `hd-media/vmlinuz`, i una imatge de CD de Debian en el directori de nivell més alt en el disc dur (la imatge de CD ha de tenir un nom de fitxer que acabi en `.iso`). Ara només és qüestió d'arrencar linux amb `initrd`.

Aquest resum només pretén enumerar-vos els diferents tipus d'instal·lació que té Debian. Però per a poder abordar la instal·lació, us aconsellem que visiteu la seva pàgina web, com ja hem indicat anteriorment, i llegiu detingudament els "COM d'instal·lació".

El grau d'interacció que requereix una instal·lació també depèn del sistema i del tipus d'instal·lació triats. Tal com era previsible, cadascun d'aquests sistemes té els seus avantatges i els seus inconvenients: mentre que una instal·lació estàndard ens permet anar configurant 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 que muntarem és molt gran (per exemple, es podria plantejar la implementació d'aquest tipus d'instal·lació en un departament en el qual convisquessin ordinadors destinats a ús personal, amb d'altres de destinats a la paral·lelització, i en el qual el seu nombre augmenta sovint).

A l'hora de triar un sistema i tipus d'instal·lació, hem de considerar diversos factors, com: quantes instal·lacions farem? Quantes instal·lacions diferents farem? Quin grau d'experiència tenim? Etc. El fet que aquesta sigui 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 queda 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 la velocitat d'accés que hi tenim és molt baixa, no tenim més opció que triar una instal·lació basada en un joc de CD-ROM o de DVD-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 via *gateway*), la millor opció serà decantar-se per una instal·lació per xarxa.

Una instal·lació efectuada per xarxa suposa molts avantatges sobre una efectuada mitjançant CD-ROM o DVD-ROM, i especialment en el cas de Debian, ja que això ens permetrà instal·lar les últimes versions disponibles dels paquets; a més, 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 o DVD-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 o DVD-ROM, ja que Debian es caracteritza per ser una distribució en la qual només s'inclouen paquets que han estat provats exhaustivament). I una vegada finalitzada la instal·lació, sempre podem canviar el contingut del fitxer `/etc/apt/sources.list` i afegir-hi les línies pertinents perquè també pugui accedir a Internet, sobretot per a les actualitzacions de seguretat.

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`. Tots els paquets inclosos en la distribució oficial de Debian són lliures d'acord amb les directrius de programari lliure de Debian. Això assegura l'ús i la redistribució lliure dels paquets i del seu codi font complet. 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* les proveeix l'organització Debian per al benefici dels seus usuaris:

- **main.** Paquets que compleixen amb les directrius de programari lliure de Debian, és a dir, es garanteix l'ús i la 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 part dels binaris també tenen disponible el seu codi font, depenen d'altres paquets que no ho són.
- **non-free.** Paquets que, tot i que potser no costen diners, es troben sota condicions oneroses que restringeixen d'alguna manera el seu ús o redistribució.
- **non-US/main.** Els paquets d'aquesta àrea són lliures en si mateixos però no poden ser exportats des d'un servidor als Estats Units.
- **non-US/non-free.** Paquets que no poden ser exportats dels Estats Units per contenir programari de xifratge o programari que pot afectar assumptes relacionats amb patents.

La distribució oficial de Debian es constitueix del contingut de la secció *main* de l'arxiu de Debian.

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 (estable).** Aquesta és la versió oficial més recent de la distribució Debian GNU/Linux. Consta de programari estable i ben provat, i canvia només en incorporar correccions importants de seguretat o d'usabilitat.
- **Testing (proves).** 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 deixar de ser *unstable* per passar a ser *testing*. *Testing* no té les actualitzacions de l'equip de seguretat en el mateix moment en què surten.
- **Unstable (inestable).** En aquesta distribució es troben 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.

Encara que tots els paquets tenen les seves pròpies dependències, no hi ha cap problema a barrejar paquets de diferents distribucions. `apt-pinning` (consulteu `man apt-pinning`) facilita molt aquesta tasca. Tot i així, en sistemes crítics és recomanable utilitzar només paquets de la distribució estable, els més fiables.

5.2. Instal·lació de Debian Etch

Com s'ha comentat, Debian presenta diversos tipus d'instal·lació, des de DVD-ROM, CD-ROM, memòries USB i disquets, o utilitzant una xarxa. Els tipus d'instal·lació més utilitzats són: CD-ROM o DVD-ROM, CD-ROM mínim o 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 DVD-ROM i, una vegada acabat, analitzarem les diferències entre aquest procés i el d'instal·lació per xarxa.

Per primera vegada a Debian GNU/Linux es proporcionen CD i DVD multiarquitectura que permeten la instal·lació de múltiples arquitectures des d'un únic disc.

5.2.1. Sabors de Debian Etch

Debian GNU/Linux 4.0 o Debian Etch suporta dotze arquitectures principals i algunes variacions de cada arquitectura conegudes com a "sabors" (diferents *kernels* precompilats destinats a donar suport als diferents tipus de maquinari).

Per exemple, Debian GNU/Linux 4.0 per a l'arquitectura Intel x86 ve de sèrie amb la versió 2.6.18 del nucli. Per a més informació, podeu consultar el document que conté les instruccions d'instal·lació del sistema Debian GNU/Linux 4.0 (nom en clau "etch"), per a l'arquitectura Intel x86. A més, conté enllaços a altres fonts d'informació, així com informació de com obtenir el millor del seu nou sistema Debian.

5.2.2. *Installing Debian GNU/Linux 4.0 For Intel x86*

El document bàsic que ens proporciona Debian per a la seva instal·lació és *Installing Debian GNU/Linux 4.0 For i386*, inclòs en el DVD1 (`/doc/install/manual/en/index.html`) disponible en diferents idiomes, entre ells el castellà (`/doc/install/manual/es/index.html`). És recomanable llegir aquest document i tenir-ne una còpia a mà per si sorgís algun problema, com podria ser no disposar d'una unitat de DVD-ROM *bootable*.

5.3. Instal·lació de Debian Etch mitjançant DVD-ROM

Aquesta instal·lació es portarà a terme en un ordinador estàndard sobre un disc dur ATA connectat com a màster sobre el port IDE primari estàndard. Pel que es refereix a controladores i discos SCSI, si aquests són mitjanament estàndards, seran detectats sense cap problema durant el procés d'arrencada, igual que els discos Serial ATA. Des de la pàgina web de Debian es poden baixar els tres DVD que pertanyen a aquesta versió, encara que per a un sistema bàsic d'escriptori en tindrem suficient amb el primer.

5.3.1. Abans de començar la instal·lació

Abans de començar la instal·lació pròpiament dita, caldrà cerciorar-se que disposem d'un espai mínim en el 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 pensem instal·lar-hi també un altre sistema operatiu (n'hi haurà prou que reservem l'espai que considerem per a aquest amb el tipus de partició que necessiti).

Si disposem d'un espai que prèviament havíem reservat, perquè ja teníem en ment instal·lar-hi GNU/Linux, o tenim una partició de qualsevol altre sistema operatiu en què el vulguem instal·lar, també podem prosseguir amb la instal·lació, és a dir, arrencar des del DVD-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 d'aquells amb sistemes de fitxers poc consistents), hem d'alliberar espai per a poder-hi instal·lar Debian GNU/Linux 4.0. La dificultat de fer aquesta operació dependrà estrictament de quin sistema de fitxers sigui el que conté aquesta partició.

Probablement, el sistema operatiu en qüestió siguiés de la família de productes de MicrosoftTM; si el sistema de fitxers és de tipus FAT o FAT32 (utilitzats per MSDOSTM, Windows95TM i Windows98TM) el problema és relativament senzill de resoldre, ja que amb la mateixa distribució se'ns facilita una aplicació (`fips20.exe` en el DVD1 `-/tools/fips20.zip-`) que ens assistirà en la repartició del disc dur i en la creació d'espai per a instal·lar-hi GNU/Linux.

Si ja hi tinguéssim instal·lat un sistema operatiu GNU/Linux, podem utilitzar una de les moltes aplicacions que hi ha per a redimensionar particions de discos durs i crear-ne de noves. Com per exemple `partman` (eina original de GNU/Linux), `fdisk`, `gparted` (editor gràfic de partició de disc dur sota GNU/Linux i Gnome) o `qtparted` (editor gràfic de partició de disc dur sota GNU/Linux i KDE). Aquestes aplicacions gestionen tant sistemes de fitxers de tipus FAT o FAT32 com NTFS.

Si no hi tinguéssim instal·lat cap sistema operatiu GNU/Linux, podem utilitzar el `Gparted Live-CD`, és un *live-CD* que conté un sistema operatiu GNU/Linux petit i bàsic que s'executa directament des del CD-ROM i que conté l'aplicació `Gparted`.

I com a última opció, sempre podem recórrer a aplicacions comercials.

Independentment de l'aplicació que utilitzem, abans sempre cal **desfragmentar el disc**. El desfragmentador de disc és una utilitat que permet analitzar discos locals, i trobar i consolidar carpetes i arxius fragmentats. També pot desfragmentar discos des d'una línia d'ordres mitjançant l'ordre `defrag`. Amb això evitem problemes, ja que podem tenir arxius fragmentats i una part d'ells tenir-los al final de la partició. Per tant, en redimensionar i prendre espai en aquesta part final, ens carregariem aquests fitxers i, depenent del tipus, en el millor dels casos, perdríem la informació, però si són del sistema, podríem inutilitzar el sistema operatiu.

Així que és molt important desfragmentar el disc abans de reparticionar, igual que fer còpies de seguretat de tota la informació del disc. Més val prevenir...!

5.3.2. Arrencada del sistema d'instal·lació

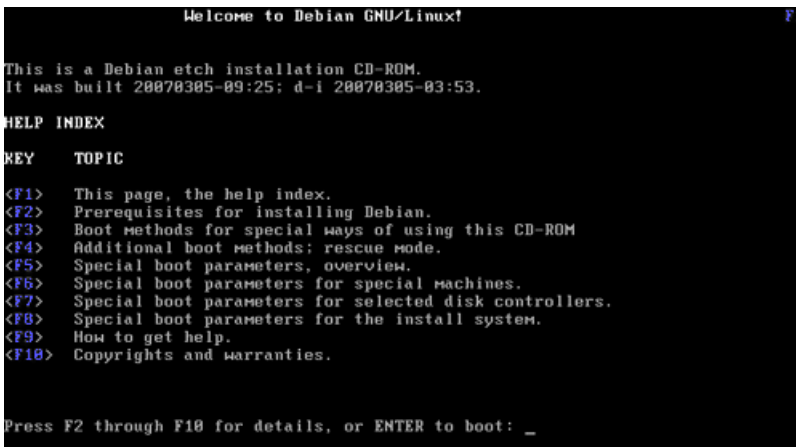
Arribats a aquest punt, podem començar la instal·lació pròpiament dita. Per a això, arrencarem l'ordinador, ens assegurarem que el primer dispositiu a l'hora d'arrencar (*boot*) sigui la unitat de DVD-ROM (entrant a la BIOS) i hi posarem el DVD1. Al cap d'uns moments ens apareixerà una pantalla de benvinguda com la següent:

Figura 5.1



Si premem la tecla de funció "F1", ens apareixerà una pantalla com la següent:

Figura 5.2



Podem prémer qualsevol de les opcions que ens ofereix el menú; si per exemple premem "F6" ens apareixerà el següent:

Figura 5.3

```

SPECIAL BOOT PARAMETERS - VARIOUS HARDWARE F6

You can use the following boot parameters at the boot: prompt,
in combination with the boot method (see <F3>).
If you use hex numbers you have to use the 0x prefix (e.g., 0x300).

HARDWARE                                PARAMETER TO SPECIFY
IBM PS/1 or ValuePoint (IDE disk)       hd=cylinders,heads,sectors
Some IBM ThinkPads                       floppy,floppy=thinkpad
IBM Pentium Microchannel                 mca-pentium no-hlt
Protect I/O port regions                 reserve=iobase,extent[...]
Workaround faulty FPU (old machines)     no387
Laptops with screen display problems     vga=771
Use first serial port at 9600 baud        console=ttyS0,9600n8

If you experience lockups or other hardware failures,
disable buggy APIC interrupt routing     noapic nolapic

For example:
boot: install vga=771 noapic nolapic

Press F1 for the help index, or ENTER to boot: _

```

Però la més destacable és la "F3", en la qual trobarem els diferents mètodes que s'admeten per a la instal·lació. Podem instal·lar amb el mètode per defecte, amb l'instal·lador gràfic, en mode expert, per a un control més gran, o en mode expert amb l'instal·lador gràfic. Teclegem, doncs, `expertgui` per a una instal·lació en mode expert amb l'instal·lador gràfic:

Figura 5.4

```

BOOT METHODS F3

Available boot methods:

install
  Start the installation -- this is the default CD-ROM install.
installgui
  Start the installation using the graphical installer.
expert
  Start the installation in expert mode, for maximum control.
expertgui
  Start the installation in expert mode using the graphical installer.

To use one of these boot methods, type it at the prompt, optionally
followed by any boot parameters. For example:

boot: install acpi=off

If unsure, you should use the default boot method, with no special
parameters, by simply pressing enter at the boot prompt.

Press F1 for the help index, or ENTER to boot: expertgui_

```

I comencem amb la instal·lació:

Figura 5.5

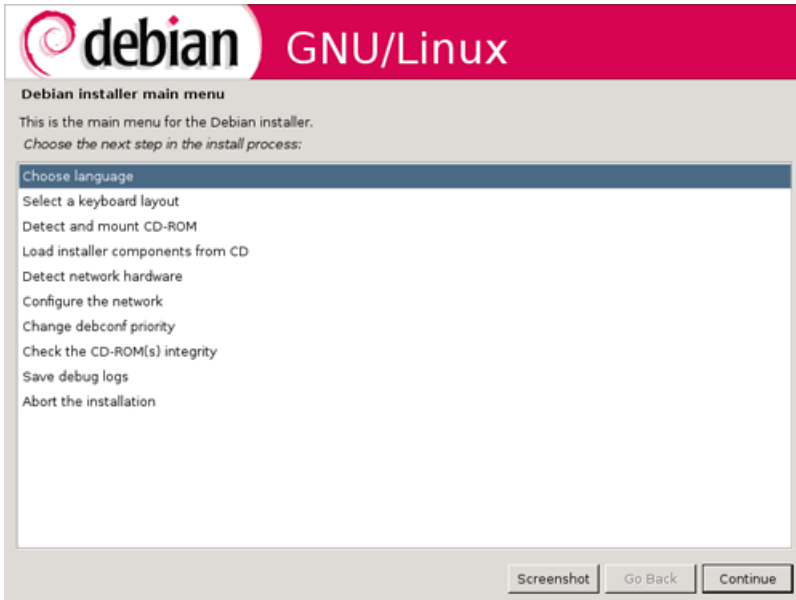
```

PnPBIOS: Scanning system for PnP BIOS support...
PnPBIOS: Found PnP BIOS installation structure at 0xc00f6ca0
PnPBIOS: PnP BIOS version 1.0, entry 0xf0000:0x9406, dseg 0x400
pnp: 00:0b: ioport range 0x4d0-0x4d1 has been reserved
pnp: 00:0b: ioport range 0x1000-0x103f has been reserved
pnp: 00:0b: ioport range 0x1040-0x104f has been reserved
PnPBIOS: 20 nodes reported by PnP BIOS; 20 recorded by driver
PCI: Using ACPI for IRQ routing
ACPI: PCI interrupt 0000:00:07.2[0] -> GSI 19 (level, low) -> IRQ 169
ACPI: PCI interrupt 0000:00:10.0[A] -> GSI 17 (level, low) -> IRQ 177
ACPI: PCI interrupt 0000:00:11.0[A] -> GSI 18 (level, low) -> IRQ 185
ACPI: PCI interrupt 0000:00:12.0[A] -> GSI 19 (level, low) -> IRQ 169
testing the IO APIC.....
Using vector-based indexing
..... done.
PCI: Cannot allocate resource region 4 of device 0000:00:07.1
Simple Boot Flag at 0x36 set to 0x1
UFS: Disk quotas dquot_6.5.1
Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
devfs: 2004-01-31 Richard Gooch (rgooch@atnf.csiro.au)
devfs: boot_options: 0x1
Initializing Cryptographic API
Limiting direct PCI/PCI transfers.
isapnp: Scanning for PnP cards...

```

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

Figura 5.6



A partir d'ara ja disposem, prement la combinació de tecles "Alt+F2" i "Intro", d'un *shell* que, encara que molt bàsic, es pot usar en qualsevol moment de la instal·lació.

En la tty3 ("Alt+F3") el *kernel* va deixant 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 "English" en "Choose language" per a evitar possibles errors de traducció. En general, i si és possible, sempre és millor treballar amb l'idioma original; si bé, una vegada més, es deixa llibertat al lector perquè decideixi ell mateix el que li sembla més adequat; en aquest cas, que sigui ell qui triï l'idioma que utilitzarà durant la instal·lació.

Atès que aquest manual està documentat en espanyol, s'ha seguit la mateixa filosofia i totes les captures de la instal·lació s'han fet en aquest idioma.

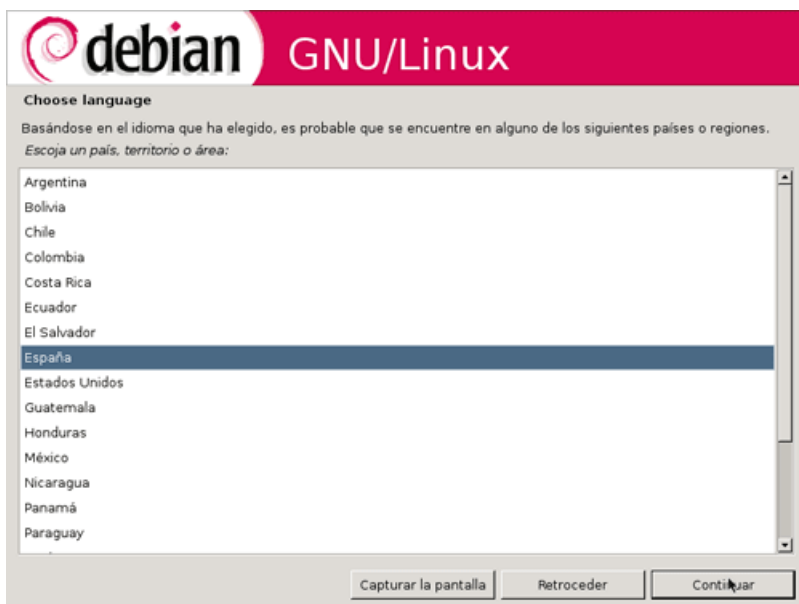
Arribats a aquest punt, triem l'idioma i premem l'opció "Continue":

Figura 5.7



En la pantalla "Escoja un país, territorio o área", igual que en l'idioma, es deixa llibertat al lector perquè decideixi ell mateix. En aquest cas, s'ha seleccionat "Espanya":

Figura 5.8

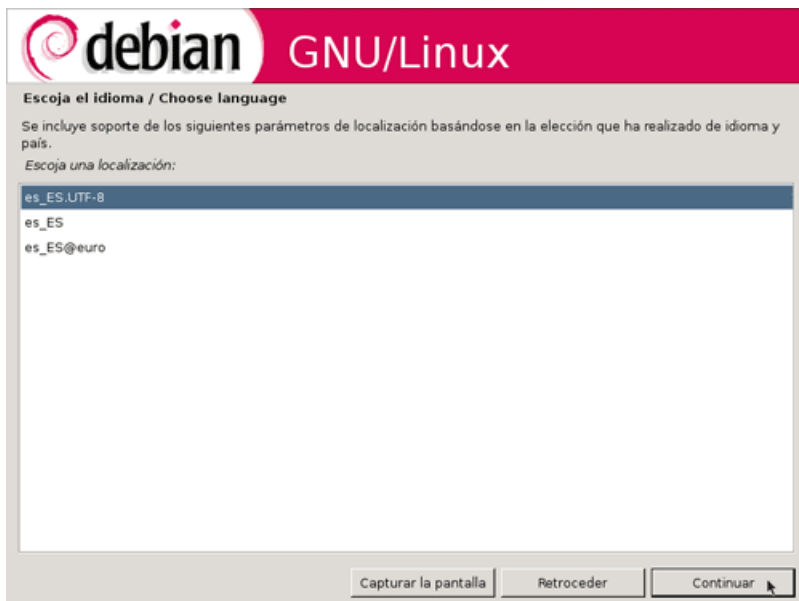


Configuració de l'idioma

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

A continuació, en funció de l'idioma i del país seleccionat, ens proposa diversos paràmetres de localització i triem el que dóna suport a UTF-8:

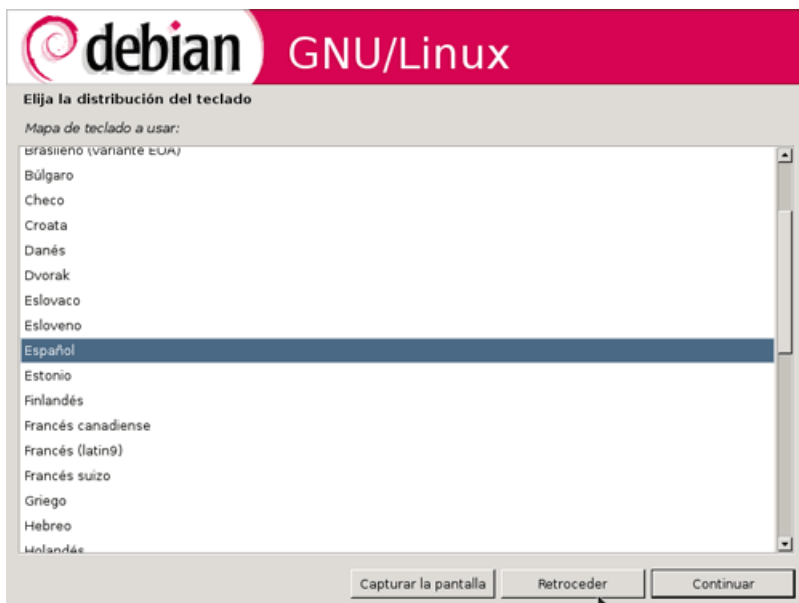
Figura 5.9



5.3.4. Configuració del teclat

Com a primer suggeriment, la interfície d'instal·lació ens proposa que configurem el teclat, "Elija la distribución del teclado" i seleccionem el mapa de teclat que volem usar:

Figura 5.10

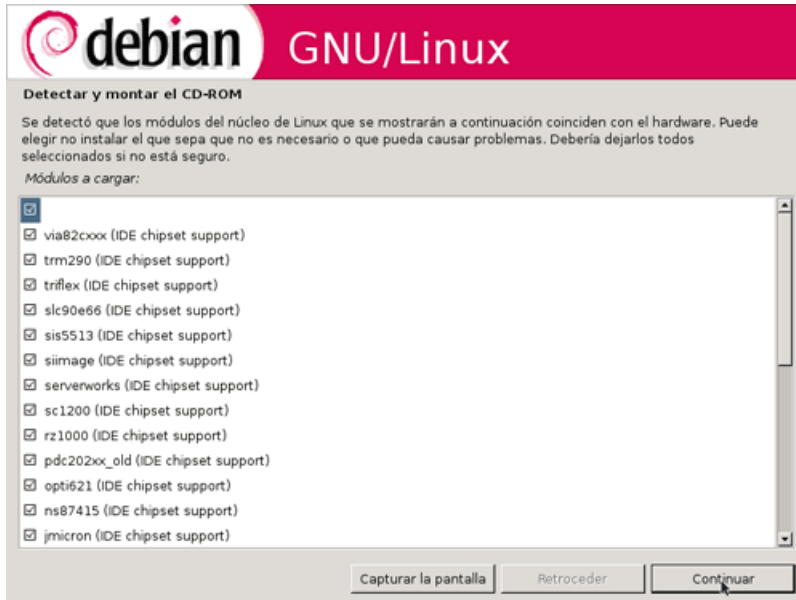


Premem "Continuar" perquè el dispositiu d'entrada respongui al mapatge de caràcters del teclat espanyol. En la tty2 podem comprovar que, efectivament, ja disposem d'accents, per exemple.

5.3.5. Detectar i muntar el CD-ROM

El pas següent és detectar i muntar el CD-ROM (DVD-ROM). Ens informa que l'ha detectat i ens aconsella que seleccionem tots els mòduls que carregarem:

Figura 5.11



I, a continuació, si la instal·lació l'estem duent a terme des d'un ordinador portàtil, ens pregunta si volem iniciar els serveis de targeta PC per a utilitzar targetes PCMCIA, a la qual cosa responem "Sí":

Figura 5.12



5.3.6. Configuració de la xarxa

El pas següent que farem en el procés d'instal·lació és la "Detección del hardware de red":

Figura 5.13



Podem configurar la xarxa mitjançant DHCP o manualment. En aquest cas, escollim l'opció de configurar la xarxa mitjançant DHCP.

Figura 5.14



Una vegada configurada la xarxa, és moment de donar un nom al sistema que estem instal·lant, ens demana el "Nombre de la máquina", que per defecte se suggereix el nom de "debian". Podem esborrar aquest suggeriment i donar-hi un nom que ens agradi més, compost d'una sola paraula.

Figura 5.15



En aquest punt només cal especificar el nom local del sistema. Si aquest forma part d'una xarxa, i per tant necessita domini, aquest s'especifica a la pantalla que ve a continuació.

5.3.7. Partició del disc dur

Una vegada especificat el nom de la màquina i el domini (en cas que sigui necessari) el procés d'instal·lació ens proposa que "Detectemos los discos" i, a continuació, que fem el "Particionado de discos".

Figura 5.16



A l'hora de particionar els discos, la mida, les 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. En tractar-se d'una instal·lació amb finalitats educatives, es

facilita seguidament la informació sobre les particions que es crearan suposant que es treballa 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 l'altra, de *swap* (és veritat que es poden fer instal·lacions sense *swap*, però, tal com ja s'ha dit, no és recomanable en absolut). Per tal d'augmentar l'eficiència del sistema, nosaltres crearem sis particions.

L'eina que ens proporciona Debian per particionar discos és *cfdisk* i 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 aniran variant 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 la mida de la partició, i, finalment, la seva ubicació física en el disc (és recomanable que abans de començar a fraccionar el disc fem un petit esquema de com ho volem fer i que creem les particions a partir d'aquest, per a poder respondre així sempre "[Beginning]" a aquesta pregunta).

La primera partició és la destinada a allotjar l'arrel (/); aquesta no ha de ser gaire gran i per això s'hi destinarà menys d'un deu per cent del disc dur, preferentment en una partició primària, però si no disposem d'ella, la podem crear com a partició lògica sense donar-hi més importància. Li indiquem que la partició serà al principi de l'espai lliure i que el tipus de sistema d'arxius triat per a la partició és *ext3*.

La segona es destinarà a la partició de *swap* (*swap*). Es recomana que aquesta tingui, com a mínim, una mida igual a la de la memòria RAM, 512 Mb, 1.024 Mb, etc.; amb aquestes dimensions ens assegurarem, llevat de casos excepcionals, que no arribarà mai a saturar-se. Aquesta partició també és preferible que sigui primària, però si ha de ser lògica, tampoc no repercutirà en el rendiment del sistema. Si tenim més d'una instal·lació de GNU/Linux al mateix ordinador, es pot utilitzar la mateixa partició *swap* per a totes elles, ja que la informació que s'hi pugui emmagatzemar durant el funcionament del sistema és totalment volàtil. El tipus de sistema d'arxius per a la partició *swap* serà d'intercanvi (*swap area*).

La tercera partició serà per al directori *usr* (*/usr*); cal tenir present que aquesta partició inclourà gran part del programari que s'hi instal·li, per la qual cosa haurà de tenir una mida significativa, entorn d'un quaranta per cent del disc. El seu sistema d'arxius serà *ext3*.

La quarta partició es destinarà al directori `var (/var)`, on s'allotgen biblioteques, fitxers de `log`, etc. Igual que les anteriors, també serà `ext3`.

La cinquena partició estarà destinada a allotjar els directoris `home` dels usuaris (`/home`), la finalitat dels quals és emmagatzemar les dades dels usuaris, i, depenent de la mida del disc dur, se li pot assignar entre un deu i un vint per cent del disc dur, en funció del nombre d'usuaris i de l'ús que es va fer del sistema. Aquesta partició també serà `ext3`.

L'espai restant, la sisena partició, es destinarà al directori `var (/tmp)` i el seu sistema d'arxius també serà `ext3`.

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 una o dues particions i, d'una altra banda, dóna més robustesa al sistema. Entre altres avantatges, tenir les dades repartides entre diferents particions provoca que la corrupció d'una d'elles no impliqui automàticament la pèrdua de tota la informació del sistema. Òbviament, es poden crear altres particions o ometre algunes de les proposades (l'ordre de les particions no afecta el comportament del sistema).

Així doncs, una vegada detectats els discos escollim el mètode de particionat, que pot ser guiat o manual. En el cas de manual, podem crear les particions que vulguem, amb la mida i el tipus de sistema d'arxius, com per exemple tal com s'explica anteriorment.

I si triem l'opció guiada, ens anirà proposant el sistema de particionat. Per ser la nostra primera instal·lació, triarem l'opció guiada.

Figura 5.17



A continuació, ens mostra la selecció de dispositius que cal fraccionar. Si disposem de més d'un disc dur, mitjançant els cursors podrem seleccionar sobre quin d'ells volem operar. Una vegada feta la selecció, premem l'opció "Continuar".

Debian proposa diversos esquemes de particionat: tots els fitxers en una mateixa partició, afegir una partició per a les dades dels usuaris o separar les principals particions. Tal com hem vist anteriorment, no s'aconsella efectuar tota la instal·lació en una mateixa partició.

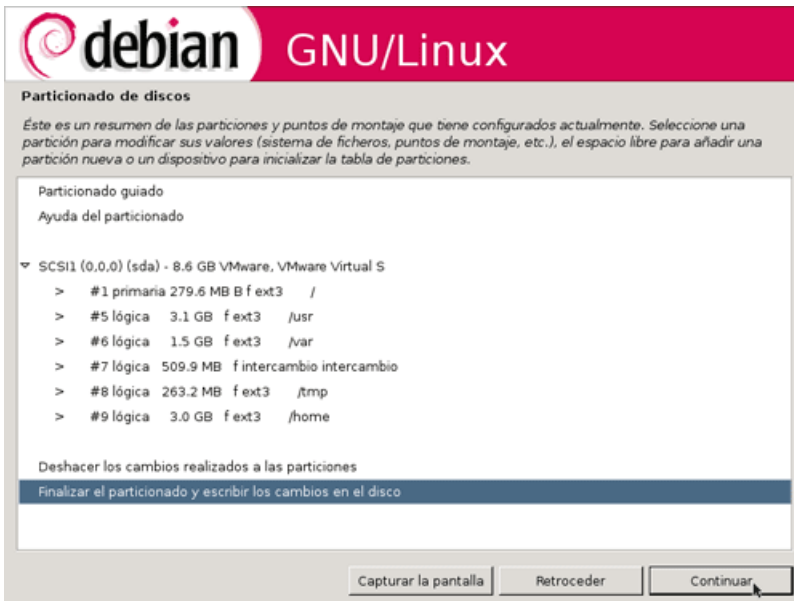
Com que tenim una opció que ens permet separar les particions `/home`, `/usr/`, `/var` i `/tmp` de manera automàtica, aquesta serà l'opció que triarem per a la nostra primera instal·lació d'un sistema Debian GNU/Linux.

Figura 5.18



Així que si seguim l'esquema de particionat anterior, obtenim la taula de particions següent:

Figura 5.19



Tal com podem veure, aquesta és la proposta de particions que ens indica de manera automàtica l'aplicació de particionat. En el cas de voler modificar alguna partició (sia la seva mida, posició o sistema d'arxius), afegir o fins i tot esborrar alguna de les particions proposades, podem seleccionar aquesta partició i fer els canvis oportuns, o anar a l'opció de "Deshacer los cambios realizados a las particiones" i tornar a particionar.

Si per contra volem aplicar aquesta proposta de particionat, només hem de seleccionar l'opció "Finalizar el particionado y escribir los cambios en el disco" i "Continuar". Ara aquesta informació s'escriu en l'MBR i es fan efectius els canvis. Abandonem l'aplicació `cfdisk` i prosseguim amb la instal·lació.

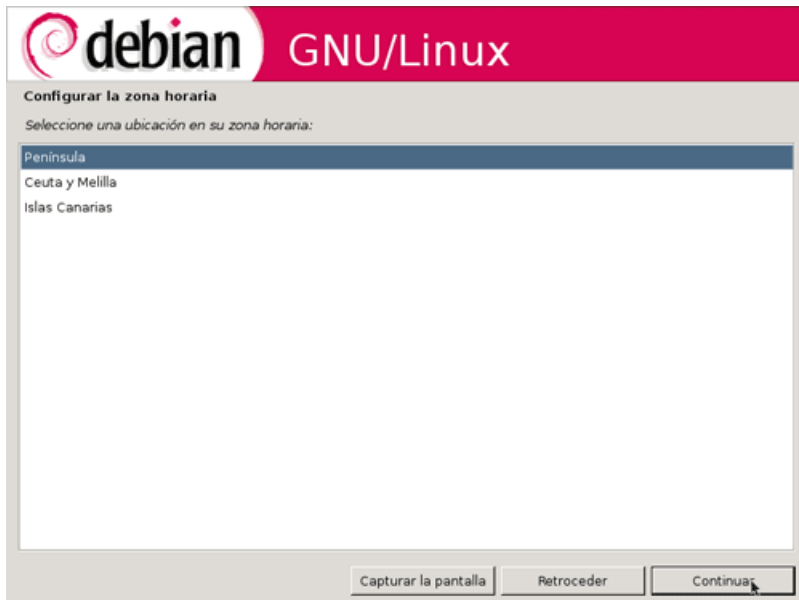
Figura 5.20



5.3.8. Configuració horària

Immediatament després de particionar el disc dur, entrem en la configuració horària del sistema.

Figura 5.21



A continuació, seleccionem tenir el rellotge en hora UTC, hora universal coordinada.

Figura 5.22



Bibliografía

Per a més informació sobre el temps universal coordinat, visiteu l'entrada "Tiempo universal coordinado" de la Wikipedia.

5.3.9. Configurar usuaris i contrasenyes

Una vegada acabats els procediments per a la configuració horària del sistema, ha arribat el torn de configurar els usuaris i les contrasenyes.

En primer lloc, el sistema ens pregunta si volem habilitar l'opció de tenir les contrasenyes encriptades, d'aquesta manera només l'usuari *root* té accés al fitxer (*/etc/shadow*) que conté les contrasenyes. Això suposa una mesura més de protecció per al nostre sistema.

La segona pregunta és per a decidir si permetem que l'usuari *root* accedeixi al sistema o creem un usuari amb permisos per a ser administrador mitjançant l'ordre *sudo*. Aquí deixem que el lector sigui qui decideixi com vol treballar quan hagi de fer les tasques administratives; nosaltres en aquesta primera instal·lació permetrem que l'usuari *root* accedeixi al sistema.

Figura 5.23



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

Figura 5.24



Tal com ja s'ha remarcat, treballar sempre com a *home* és una mala política per diferents motius. A conseqüència d'això, el sistema ens recomana crear un compte d'usuari normal.

Figura 5.25



Ens demanarà el nom complet del nou usuari:

Figura 5.26



I a continuació, el nom d'aquest compte, és a dir, l'usuari per a accedir al sistema:

Figura 5.27



I, finalment, la seva contrasenya, que, per la mateixa raó d'abans, l'hem d'escriure dues vegades:

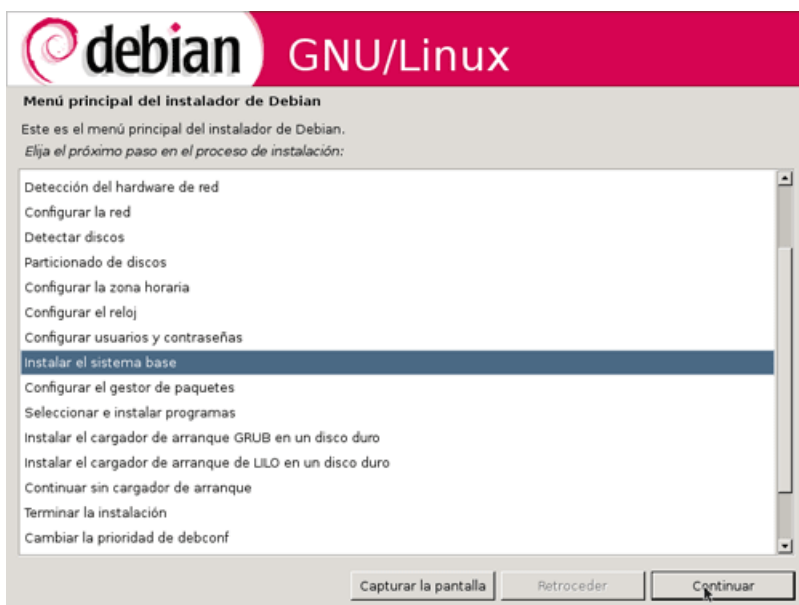
Figura 5.28



5.3.10. Instal·lació del sistema base

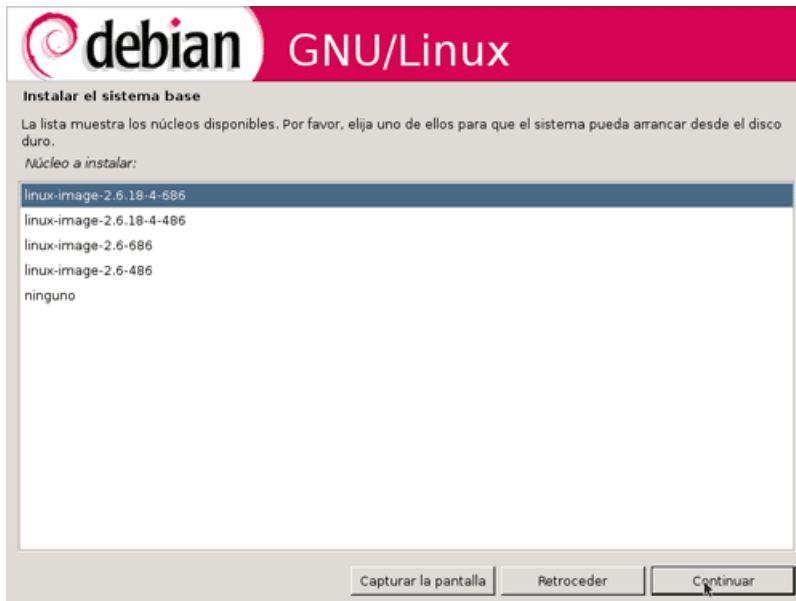
Amb tota aquesta informació, el sistema comença a fer la instal·lació del sistema base:

Figura 5.29



El primer que ens pregunta és el *kernel* que hi volem instal·lar. Com podem veure en la imatge següent, podem triar entre quatre *kernels*. Escollim la primera opció:

Figura 5.30



5.3.11. Configurar el gestor de paquetes

Figura 5.31



5.3.12. Seleccionar i instal·lar programes

Figura 5.32



5.3.13. Instal·lació de GRUB

A continuació, es comença a instal·lar el paquet GRUB boot loader. Aquest analitza el nostre disc dur i busca tots els sistemes operatius que hi ha instal·lats perquè una vegada finalitzada la instal·lació de Debian i reiniciat el nostre ordinador, ens deixi triar qualsevol dels sistemes operatius instal·lats. Ens demana confirmació per a instal·lar-lo en l'MBR, li diem que sí.

5.3.14. Reinicialització del sistema

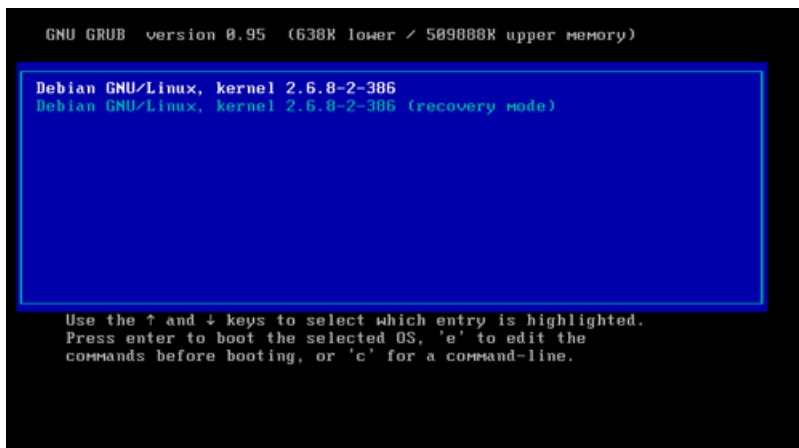
Ha arribat el final de la instal·lació del sistema base, aquesta és la pantalla que apareix:

És el moment de reiniciar el sistema per a arrencar el sistema base que hem instal·lat en el nostre disc dur, i a partir d'ell començar a personalitzar la nostra primera instal·lació de GNU/Linux. Així doncs, retirem el DVD-ROM i li donem a l'opció "Continue" de la imatge anterior.

5.3.15. Arrencada del sistema base

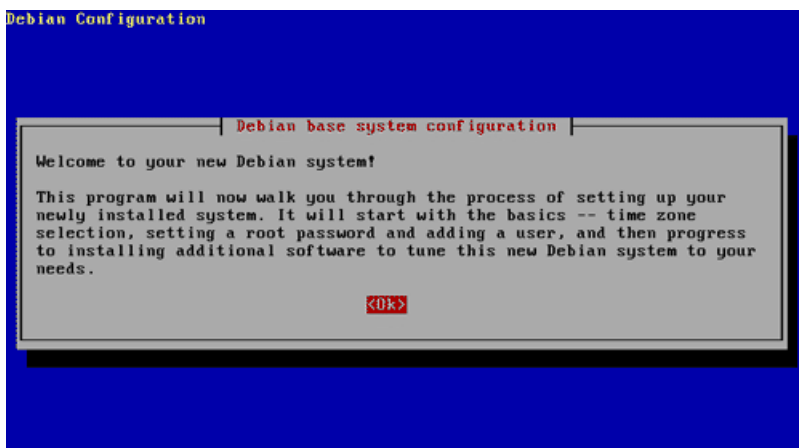
Si tot ha anat com cal, després de reiniciar el sistema observarem que podem seleccionar el sistema operatiu Debian GNU/Linux:

Figura 5.34



Transcorreguts uns moments, en els quals aniran apareixent per pantalla els resultats dels diferents processos que s'executen durant l'arrencada, el sistema entrarà en una pantalla de benvinguda, i ens convidarà a prosseguir la instal·lació, i ens recordarà que podem repetir aquest procés en qualsevol moment executant l'ordre `base-config` com a *root*. Premem "Ok" i prosseguim:

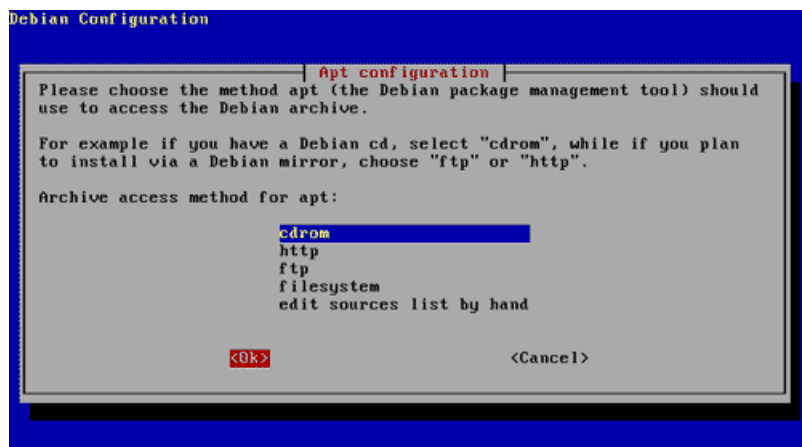
Figura 5.35



5.3.16. Configuració d'apt

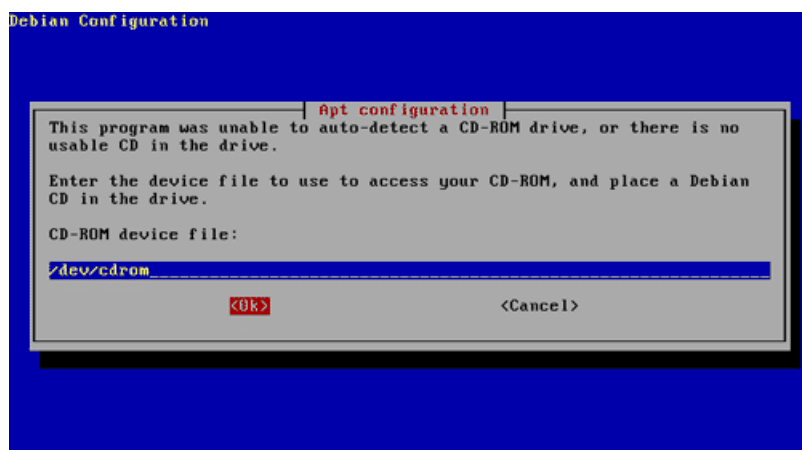
En aquesta secció se'ns pregunta sobre les fonts on `apt` haurà d'anar a buscar la informació per a construir la seva base de dades sobre els paquets (dependències, estat d'instal·lació, etc.). Com que estem fent una instal·lació íntegrament basada en DVD-ROM, triem la primera opció "cdrom" i premem "Ok":

Figura 5.36



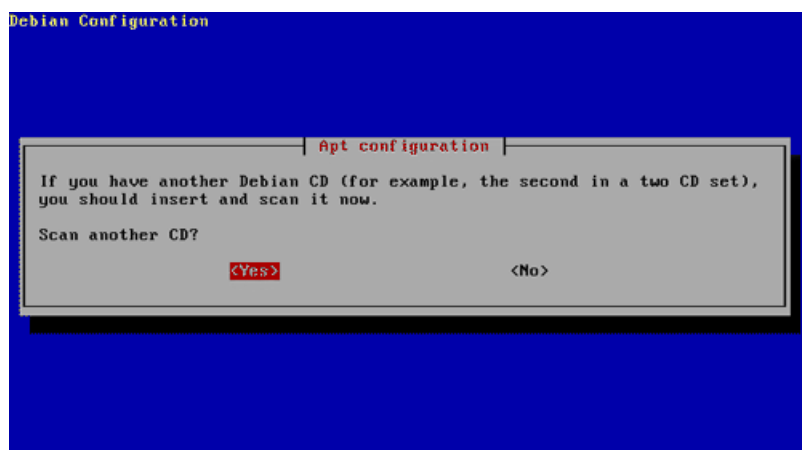
En la pantalla següent, tornem a seleccionar "Ok":

Figura 5.37



I com que hi tenim inserit el DVD-ROM 1, transcorreguts uns moments en els quals apt extreu la informació relativa als paquets que conté aquest DVD, se'ns preguntarà si volem afegir el contingut d'un altre DVD a la base de dades. Una vegada haguem introduït el DVD-ROM 2 al lector, contestem "yes" a aquesta pregunta:

Figura 5.38

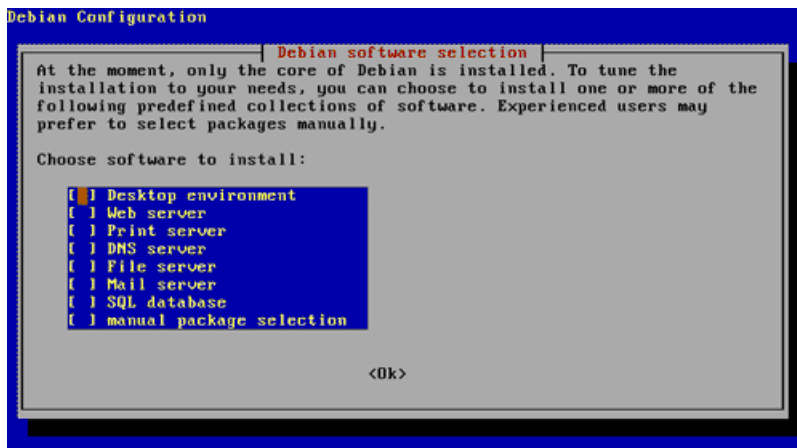


Ho fem amb el tercer DVD-ROM. Una vegada finalitzat l'escaneig dels continguts dels tres DVD-ROM, ens preguntarà si volem afegir alguna altra font d'on apt pugui obtenir paquets, de moment contestarem que no.

5.3.17. Tasksel

A continuació, s'executa el programa de selecció de paquets `tasksel`:

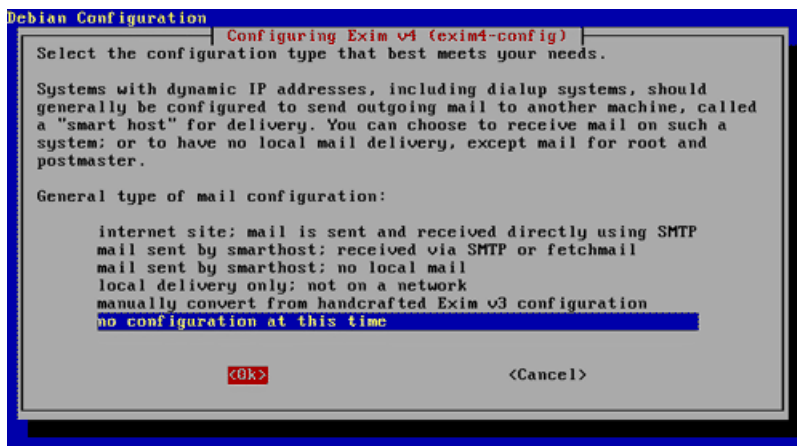
Figura 5.39



No marcarem cap opció, ja que aquest programa també el podrem executar des de la línia d'ordres una vegada haguem acabat la instal·lació. Simplement, haurèm d'executar com a `root` l'ordre `tasksel`. També hi ha un altre programa de selecció de paquets, el `dselect`.

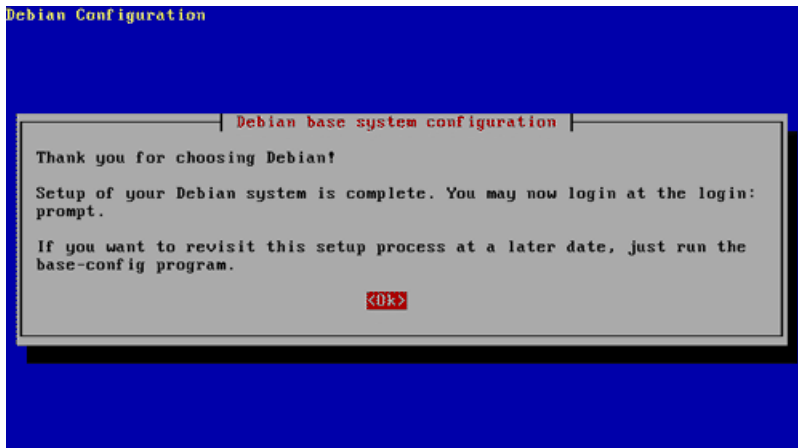
A continuació, ens pregunta per la configuració de l'Exim; com que no és objecte d'aquest taller la configuració d'aquest programa de gestió de correu, triem l'opció 6 per tal de sortir del programa:

Figura 5.40



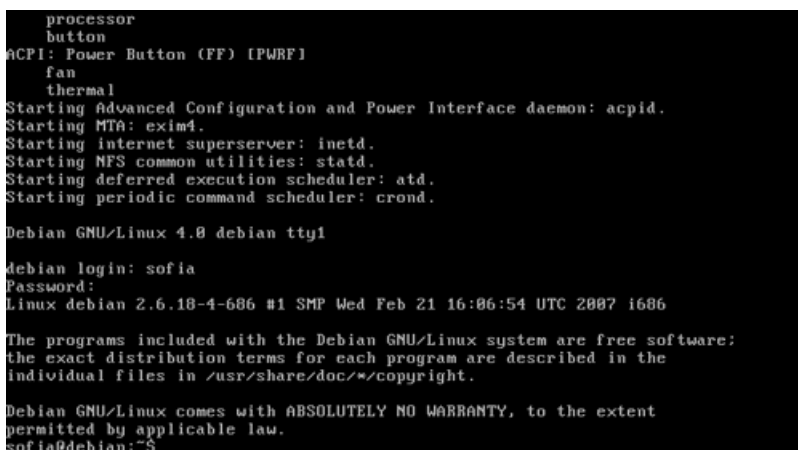
I aquí acaba la configuració del sistema base de Debian:

Figura 5.41



Finalment, després d'uns moments ens apareixerà per pantalla `debian login` i, a continuació, teclejarem l'usuari que hem creat durant el procés d'instal·lació, per exemple *sofia*, i després de prémer l'"Intro", escriurem la contrasenya per poder iniciar sessió en el sistema Debian GNU/Linux que hi acabem d'instal·lar:

Figura 5.42



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 a 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 experimentació, en què es duen a terme proves i experiments, i en el qual no prevalen ni l'estabilitat ni l'eficiència. És, en definitiva, un sistema destinat a l'adquisició de coneixements sobre el mateix sistema. Al contrari, en un sistema destinat a 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 ser-

vidor d'aplicacions, etc.) passa sempre per treballar primer sobre un sistema de desenvolupament, en què s'assajaran i provaran diferents tàctiques per a poder treure conclusions amb vista a muntar el sistema de producció.

El nostre primer sistema serà, evidentment, un sistema de desenvolupament, tant perquè no tenim en ment que 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, l'obtenció de coneixements. Hi 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 l'estudiant que una vegada finalitzat l'apartat i s'hagi fet una idea global de què 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 Etch per xarxa

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

5.4.1. Particularitats d'una instal·lació per xarxa

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

D'altra banda, per a dur a terme una instal·lació per xarxa, generalment n'hi ha prou amb un únic CD-ROM o un únic DVD-ROM que contingui la informació necessària per a poder arrencar un sistema operatiu bàsic sobre el qual es farà córrer el programa d'instal·lació (que ja és inclòs en el CD-ROM o en el DVD-ROM, juntament 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 indicat, molts dels passos que hem de seguir per fer una instal·lació per xarxa són comuns a les dades per a efectuar una instal·lació mitjançant el joc de DVD-ROM de la distribució. Així doncs, entrem en el 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] i [5.3.7]).

Arribats a aquest punt, el normal és que el *kernel* hagi reconegut la nostra targeta de xarxa i, si no és el cas, procedim a instal·lar-hi i configurar-hi el mòdul pertinent per fer-la operativa. En molts casos, la configuració, el 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 (el més còmode és mitjançant DHCP), especificar de quin lloc obtindrem les fonts i, a partir d'aquí, 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 *driver* de la nostra targeta de xarxa no ha estat compilat dins del *kernel*, hem de seleccionar el mòdul necessari per tenir-hi accés. En primer lloc, hem d'esbrinar si la nostra targeta de xarxa ja ha estat detectada durant el procés d'arrencada i si s'ha carregat el seu *driver* corresponent. Per a fer-ho, accedim al segon terminal ("Alt+F2") i executem l'ordre *dmesg*. Ara hem de buscar, entre les moltes línies que ens ha tornat aquesta ordre, si n'hi ha algunes que facin referència a la nostra targeta de xarxa. A tall d'exemple, per a una targeta RTL-8029 (Realtek Semiconductors) s'obté:

```
sofia@debian:~$ 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 recórrer a la documentació que té inclosa o a la seva inspecció visual. Si això no és possible, hi ha altres estratègies per a determinar quina és la nostra targeta, com pot ser, prémer "Alt+F2" per a accedir a la consola i investigar el contingut del fitxer `/proc/pci` (mitjançant `cat`, per exemple), o podem recórrer a la informació que ens pot 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 aquesta finalitat és recórrer a qualsevol cercador, per exemple Google, entrar-hi 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 posar la referència de la targeta en els seus cercadors. Com a últim recurs, es pot recórrer a la documentació de mòduls de xarxa del *kernel*, en què 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 aquella targeta o que calgui recórrer a mètodes avançats per a poder-lo configurar; per aquest motiu, es recomana utilitzar sempre targetes tan estàndards com sigui possible.

Una vegada haguem esbrinat el mòdul que necessitem, després d'instal·lar el *kernel*, hem de seleccionar la proposta que ens suggereix el menú principal "Configure Device Driver Modules". Després d'una pantalla d'avertiment, en què se'ns recorda que molts *drivers* ja són inclosos en el *kernel*, entrarem a la pantalla de selecció de mòduls "Select Category" (podem accedir a aquesta interfície en qualsevol moment, executant l'ordre `modconf`. Aquesta ordre serveix com a *font-end* per a l'administració de *drivers* que han estat compilats de manera modular juntament amb el *kernel*) 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-hi aquest mòdul, podem deixar que l'autoprobe configure el mòdul per nosaltres, si no s'ha de passar cap paràmetre en concret al mòdul en qüestió. Passats uns instants, rebrem el missatge que ens indicarà 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, en el 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 del *host* (abans s'ha suggerit *debian*, per exemple), sense posar-hi el domini. A continuació, hi haurem d'introduir la IP, la màscara de xarxa, el *gateway*, 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 següents en la instal·lació per CD-ROM, fins a arribar a la configuració d'*apt* "Apt Configuration". Arribats a aquest punt, 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, i llevat de problemes legals, respondrem que sí.

Respecte a la resposta sobre la pregunta següent, referent a l'ús de paquets *non-free*, ja es deixa que el mateix estudiant prengui la decisió segons els seus principis ètics. A continuació, se'ns pregunta de quin estat ha de ser el *mirror* del qual `apt` haurà d'obtenir els paquets; hem de triar sempre el que ens sigui accessible de manera més ràpida, que sol 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 mitjançant proxy; si no hem d'usar aquest servei, deixarem en blanc la línia.

És important que els paquets crítics, quant a seguretat es refereix, s'obtinguin de servidors segurs. Per aquesta raó, es recomana que aquests s'obtinguin en concret de *security.debian.org*. Acabats tots aquests passos, `apt` connectarà amb el *mirror* 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 a la instal·lació efectuada 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 tal de poder començar a treure partit de la flexibilitat i potència d'aquest sistema operatiu. En el proper taller aprendrem com configurar el sistema i instal·lar-hi noves aplicacions per a anar-lo adaptant i dotant de totes les eines que estimem necessàries per a cobrir les nostres necessitats.

6. Configuracions bàsiques

6.1. El sistema de *login*

Si encara no tenim configurat l'entorn gràfic, quan arrenquem un sistema GNU/Linux ens apareix una pantalla de *login* en què es demana que l'usuari s'identifiqui abans de començar a utilitzar el sistema. De fet, la majoria de les distribucions llancen diverses consoles a les quals podem accedir a partir d'"Alt+F1", "Alt+F2", etc.

Això ens permet treballar simultàniament amb diferents comptes alhora, tenir diverses sessions obertes per a executar diferents programes, etc. El programa que s'encarrega de gestionar cadascuna 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`, en què la `X` és el número de consola) i llançar l'aplicació de *login*. Aquest mecanisme ens permet molta flexibilitat, ja que el mateix programa `getty` permet comunicar-se 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 *login*, 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 com a:

Taula 6.1

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

Una vegada entrem en el sistema, el programa de `login` 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 `home`. Després de mostrar aquest missatge, el procés de `login` llança el `shell` configurat per defecte per a l'usuari. El primer que fa l'interpret d'ordres és executar el contingut del fitxer `.profile` (que ha de ser en el directori `home` de l'usuari). Aquest fitxer serveix perquè s'executin les instruccions configurades sempre que l'usuari entri en el 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 de manera genèrica a tots els usuaris les opcions que vulguem sense haver de posar les instruccions dins de cadascun dels `.profile` dels usuaris.

6.2. Explorar el `bash`

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

```
# CONFIGURACIONS BASIQUES

mesg 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\$ '
#ALIAS 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 en els quals tenim les ordres, els programes, les 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 *shell* abans del caràcter "#" (per a *root*) o "\$" (per als altres usuaris). Ens podem configurar aquest *prompt* de la manera que vulguem utilitzant les variables del sistema següents:

Taula 6.2

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

Variable PATH

Si volem executar els programes del directori des del qual estem situats sense necessitat de posar `./` al principi, podríem 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 d'ordres. De tota manera, no és recomanable afegir `./` al `PATH` perquè pot representar un forat de seguretat.

Finalment, tenim els *àlies* d'usuari. Els *àlies*² són sinònims, generalment per a les ordres que més utilitzem (per a no haver-les d'escriure completament). Per exemple, en un dels *àlies* que teníem en l'exemple definíem que en escriure `l` s'executés `ls ##color=auto`. D'aquesta manera, podem utilitzar ordres llargues sense haver-ho d'estar escrivint tot cada vegada que els utilitzem.

⁽²⁾Podem veure tots els *àlies* definits a partir de la mateixa ordre `alias`.

Tant en la definició del `PATH` com en la del *prompt* hem utilitzat l'ordre `export`. Aquesta ordre ens permet definir el que anomenem una **variable d'entorn**. Aquestes variables són utilitzades pel *shell* per a fer certes operacions, desar algun tipus d'informació, etc. Podem veure totes les que hi ha declarades amb la mateixa ordre `export`. Amb `set` i `unset` també podem manipular altres atributs que té l'interpret d'ordres.

Ordre echo

Amb `echo $NomVariable` podem veure el contingut d'aquestes variables i atributs.

Alguns d'aquests atributs i variables que té per defecte el *bash* són:

- `PWD`: directori actual.
- `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 *shell*.
- `HOSTNAME`: nom del sistema.
- `OSTYPE`: tipus de sistema operatiu que estem utilitzant.

- MACHTYPE: arquitectura de l'ordinador.
- HOME: directori home de l'usuari.
- HISTFILESIZE: mida del fitxer d'història (nombre d'ordres que es desen).
- HISTCMD: número d'ordre actual en la història.
- HISTFILE: fitxer en el qual es desa la història d'ordres (generalment `.bash history` del directori home de l'usuari).

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

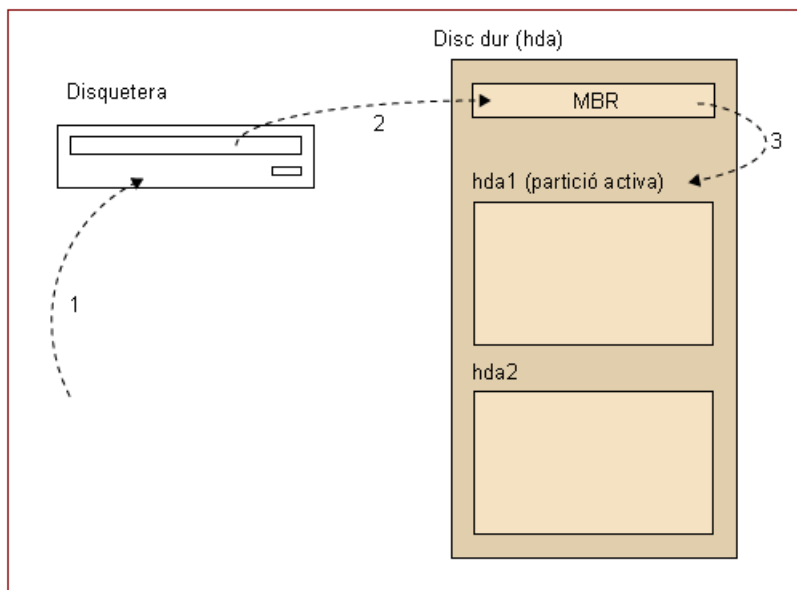
6.3. El sistema d'arrencada

Encara que amb la instal·lació del sistema operatiu ja es porta a terme la configuració i instal·lació d'un sistema d'arrencada, en aquest subapartat 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, Lilo i Grub són els més utilitzats en els entorns GNU/Linux, amb la qual cosa només ens centrarem en aquells.

Abans d'entrar en detall 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 podem configurar la seva seqüència d'arrencada. En general, aquesta seqüència sol començar buscant a la disquette i segueix amb el CD/DVD i el disc dur. Encara que podem instal·lar el Lilo o el Grub en un disquet o en el sector d'arrencada d'un CD, és més usual instal·lar-lo en el disc dur per tal de no haver d'introduir el disc cada vegada que arrenquem el nostre ordinador.

Quan el sistema d'arrencada de l'ordinador busca en el 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 guarda 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 volem 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 disquette i després el disc:

Figura 6.1



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 inspeccionen primer aquesta zona. Si no tenim cap 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 Lilo o Grub en alguna d'aquestes zones. En la configuració del programa ja li indicarem on estan situats els sistemes operatius que volem carregar.

Normalment, als ordinadors moderns la seqüència d'arrencada és: CD/DVD, disc dur, USB, Xarxa, etc.

6.3.1. Grub

Grub ens permet múltiples configuracions diferents per a fer gairebé qualsevol acció amb el sistema d'arrencada de l'ordinador. Grub també ens serveix per a instal·lar un programa a la zona d'arrencada que vulguem de l'ordinador.

Té moltes més possibilitats que el fan molt versàtil: permet tenir un petit interpret d'ordres en arrencar l'ordinador, ens permet accedir als arxius de les particions del disc sense carregar-hi cap sistema operatiu, etc. Com en el cas anterior, en aquest subapartat només veurem la seva configuració bàsica. Si volguéssim aprofundir més en el seu ús, podem recórrer al seu manual o al HOW-TO corresponent.

El sistema d'arrencada del Grub³ 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 el sistema operatiu que volem carregar, només hem d'executar el programa Grub i executar-lo en l'interpret d'ordres 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. A `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-ne 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 en com el primer paràmetre serveix per a designar on és l'arxiu de la primera fase del Grub (hi 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 cadascuna de les fases de càrrega (solen ser 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 se situa el fitxer per a la segona fase de càrrega, que és executada per la primera.

Pas de paràmetres al nucli de Linux

Amb 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.

Amb aquesta configuració, en reiniciar l'ordinador apareixerà, per defecte, l'interpret d'ordres del Grub. Amb ell podem manipular molts aspectes del disc, arrencar el sistema operatiu que vulguem, etc. Si volem engegar un sistema GNU/Linux escriurem 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 del sistema operatiu. Si optem per un menú de selecció per no haver d'escriure aquestes ordres cada vegada que arrenquem l'ordinador, podem generar un fitxer de menú com el següent (els comentaris comencen per "#"):

```
#Especificació del sistema operatiu que es carregarà per
#defecte. Aquest número està en correspondència amb l'ordre dels
#sistemes que hi ha a les seccions locals als sistemes
#operatius.
default 0
```

⁽³⁾El Grub (Grand Unified Bootloader) és el programa d'arrencada del projecte GNU.

Sistema d'arrencada

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 diferents nuclis. Per a configurar-lo només hauríem d'especificar dues seccions locals i posar, en cadascuna, quin nucli utilitzarem.

Ajuda per a Grub

Per tal de veure totes les ordres disponibles en el *shell* del Grub podem prémer "Tab". També s'hi inclouen ajuts per a tenir una referència completa de totes les ordres.

```
#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 XP
title Windows XP
root (hd0,2)
makeactive
```

Per a instal·lar⁴ 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 podem posar l'ordre de `password CONTRASENYA` a la secció global del fitxer de configuració. D'aquesta manera, quan des del menú es vulgui entrar en el *shell* del Grub es demanarà la contrasenya. 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).

També hi ha un altre tipus d'arrencada que utilitza una imatge anomenada 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 *live CD*, 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 de *initrd* i en el del programa *mkinitrd*.

⁽⁴⁾Una altra manera d'instal·lar el Grub és utilitzant el programa `grub-install`.

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 les funcions que utilitzàvem amb els fitxers i aplicar-la 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, l'ordre `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 fer malbé 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 les ordres `mount` i `umount`, que situen (munten) o desmunten tot el sistema de fitxers d'un determinat dispositiu/unitat en un directori existent del sistema. La forma bàsica d'utilitzar l'ordre és `mount dispositiu directori`, en què *dispositiu* pot referenciar qualsevol dispositiu del canal IDE o SCSI (`/dev/hdXX`, `/dev/sdXX`), la disquetera (`/dev/fdX`), cintes de *backup*, memòries USB, etc. i *directori* és la ubicació en què muntarem l'estructura de fitxers del dispositiu. És recomanable que el directori en el qual 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`, en què *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 cau del sistema de fitxers del dispositiu. Igualment, tampoc no podem desmuntar el dispositiu si algun usuari o aplicació està utilitzant algun dels seus arxius o directoris (en intentar-ho, el sistema donaria un missatge d'error).

Fixem-nos que amb l'ordre de muntatge no estem especificant en cap moment el tipus de sistema de fitxers utilitzat en la unitat que volem muntar, de manera que s'haurà de determinar de manera automàtica. Si el volem especificar manualment, podem passar a l'ordre `mount` el paràmetre `-t tipus` en què *tipus* podria ser algun dels de la taula següent (consulteu el manual de `mount` per a veure'n la llista completa):

Taula 6.3

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 <i>swap</i> de GNU/Linux

Directoris de disquetera i CD/DVD

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

Tipus	Sistema
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™ (<i>samba file system</i>)
ntfs	Branca de WindowsNT™
msdos	MS-DOS™
vfat	Branca de Windows95™

Muntatge i desmuntatge de sistemes de fitxer

Per defecte, per a poder muntar/desmuntar sistemes de fitxer, es necessiten privilegis de superusuari. El que se sol fer és definir algun usuari "administrador" en el fitxer `sudoers` de manera que amb l'ordre `sudo` davant de l'ordre pugui fer les tasques d'administració que se li permet des del fitxer `sudoers`.

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

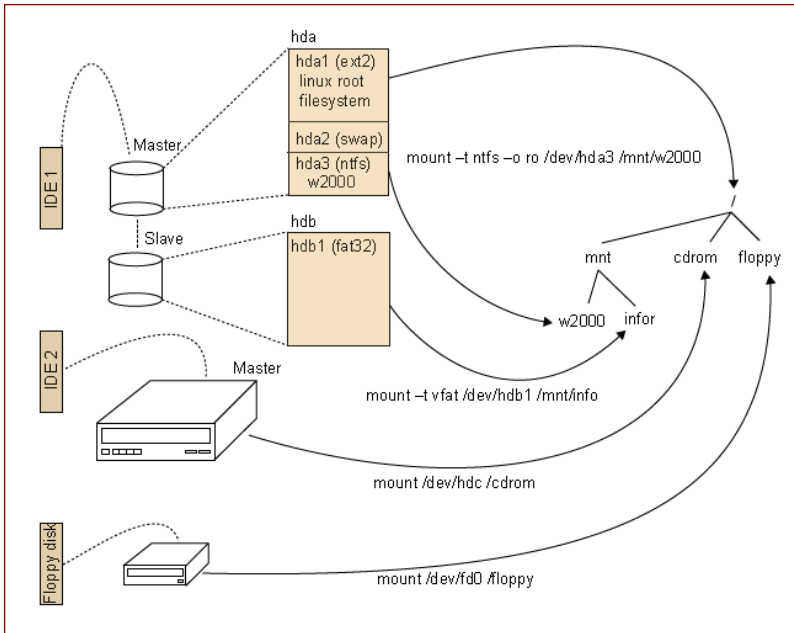
Taula 6.4

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

(Si passéssim `defaults`, s'utilitzarien les opcions `rw`, `suid`, `dev`, `exec`, `auto`, `nouser` i `async`.)

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

Figura 6.2



A més d'aquestes ordres per a muntar i desmuntar unitats, el sistema operatiu ens proporciona una altra manera de fer el mateix i tenir sempre una determinada configuració 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 cadascuna d'aquestes línies serà:

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

En el primer camp hem d'especificar el dispositiu tal com fem amb l'ordre de `mount` i el segon serà el directori en el qual volem muntar la unitat indicada. En el camp de `tipusSistema` 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 l'ordre de `mount`, separades per comes si en posem més d'una. Una opció útil d'aquest camp és la configuració de `auto` o `noauto`, amb la qual cosa indiquem al sistema que munti automàticament (o no) la unitat que s'ha d'arrencar. El camp de `dump` indica si volem fer còpies de seguretat (més informació en el manual de `dump`). Si no utilitzem aquest sistema, hi podem posar un 0. L'últim camp serveix per a indicar l'ordre de muntatge de les unitats. Si hi posem un 0, indiquem que l'ordre no és important. L'arrel del sistema de fitxers és el primer que s'ha de muntar, amb 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 el 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 uti-

litza el nucli del sistema. Seguint la mateixa política del sistema 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, hi ha algunes altres aplicacions que ens faciliten tot aquest procés per estalviar-nos alguns passos. Una d'elles són les `mtools`, que és un paquet amb diverses aplicacions que ens permeten copiar directament arxius a un disquet i des d'un disquet, i algunes altres eines interessants. També hi ha un paquet anomenat `autoofs`, que detecta automàticament la inserció d'algun dispositiu en el sistema i els munta sense necessitat d'escriure-hi cap ordre.

Ordre `mount a`

Amb l'ordre `mount a` es muntarien tots els dispositius que tinguessin l'opció d'auto activada (que hi és per defecte) d'aquest fitxer de configuració.

6.5. Configuració de dispositius

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

Encara que en aquest subapartat només veurem com configurar alguns dels dispositius més freqüentment utilitzats als ordinadors personals, segons la distribució que utilitzem, el procés pot variar significativament. En tot el subapartat ens hem basat en el sistema que s'utilitza en Debian GNU/Linux, tot i 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 estan comprimits en format `gzip`, els podem descomprimir i canviar alguna de les seves entrades per adaptar-les a les nostres necessitats. En cada línia trobem la directiva `keycode`, que indica que estem definint una tecla, indicada en el número que segueix la directiva (podem saber quin número es correspon amb cadascuna de les tecles a partir de l'ordre `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 a prémer la tecla sola (`minus`) o amb el "Shift" (`underscore`). La segona ens mostra la funció que es durà a terme en cas de prémer la tecla alhora que la de "Ctrl" (s'eliminarà el caràcter següent). Tota la informació necessària per a configurar correctament un arxiu d'aquest tipus la podem trobar a 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 l'ordre `loadkeys` (`dumpkeys` ens mostra les configurades).

Font del terminal

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 només és la representació gràfica que hi donem. Tota la configuració de les fonts de caràcters se sol trobar en `/etc/console-tools/config`.

Un altre aspecte relacionat amb el teclat és el tema de les dièresis, els 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 en desactivar-la (`set convert-meta off`) ens permet utilitzar els accents i les dièresis. Finalment, una altra configuració important (indirectament relacionada amb el teclat) és la de **locales**. Amb `locales` podem configurar la zona o zones geogràfiques on som per poder utilitzar tecles especials del teclat, veure les dates en el format al qual estem acostumats, etc. Aquesta configuració és utilitzada per moltes de les biblioteques del sistema, de manera que en moltes ordres 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 les ordres `locale-gen` i `locale` per a veure-la o actualitzar-la.

Reconfigurar *keymap* i locals

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

6.5.2. Targeta de xarxa (tipus Ethernet)

Per a configurar una nova targeta 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 haguem 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 targeta) que existeix el *driver* o mòdul necessari per a ella.

Mòdul de la targeta de xarxa

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

Una vegada el sistema reconeix la targeta, 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 sistema. 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 s'especifiquen les directives necessàries per al seu funcionament correcte.

Vegem-ho 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
```

NIC

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

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 del sistema operatiu que li permet utilitzar els protocols de comunicació de manera interna. D'aquesta manera, si provem funcions de la xarxa sense comunicar-nos amb cap altre ordinador no és necessari 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 dis-

positiu. Aquesta directiva indica que la targeta es pot muntar automàticament quan el sistema arrenca. La directiva `iface` especifica el tipus de targeta i protocol que s'utilitzarà amb ella per mitjà de la sintaxi següent: `iface dispositiu familiaProtocol metodeConfiguracio`. Amb les targetes Ethernet el dispositiu serà `ethX`, en què la `X` serà un número començant 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 dels següents:

- `inet`: IPv4, utilitzat a Internet i a la majoria de les xarxes locals.
- `inet6`: IPv6, la nova versió d'IPv4, que a poc a poc es va instaurant.
- `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, el *gateway* que cal utilitzar, etc.). En la taula següent podem veure quines són aquestes opcions per a la família de protocols `inet`:

Taula 6.5

Config	Opcions	Descripció
loopback		Mètode per a definir la interfície de <code>loopback</code> (s'ha d'utilitzar amb la interfície <code>lo</code>).
static		Mètode per a configurar una NIC amb una adreça IP estàtica.
	address	Adreça IP de la interfície. Camp obligatori.
	netmask	Màscara de l'adreça IP. Camp obligatori.
	broadcast	Adreça de <i>broadcast</i> . Si no s'especifica, es calcula automàticament.
	network	Adreça d'identificació de xarxa.
dhcp	gateway	Adreça IP del <i>gateway</i> que utilitzem per a aquesta interfície.
		Mètode per a configurar de manera remota la IP de tots els ordinadors d'una xarxa local (<i>dynamic host configuration protocol</i>).
	hostname	IP del servidor de DHCP.
	leasehours	Temps, en hores, de lloguer de la IP (passat aquest temps, es renova).
	leasetime	Temps, en segons, de lloguer de la IP.
bootp	vendor	Identificador de tipus de servidor (en general, <code>dhcpcd</code>).
	client	Identificador del tipus de client (en general, <code>dhcpcd</code>).
		Mètode per a configurar de manera remota la IP de tots els ordinadors d'una xarxa local (<i>BOOT Protocol</i>). Actualment, s'utilitza més DHCP.
	bootfile	Fitxer que s'utilitzarà en el moment de l'arrencada.

iproute2

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

Config	Opcions	Descripció
	server	Adreça IP del servidor de BOOTP.
	hwaddr	Adreça MAC del servidor BOOTP.
ppp		Mètode utilitzat amb el protocol <i>point to point protocol</i> , usat als mòdems.
	provider	Proveïdor del servei.

Tot i que en aquest subapartat no entrarem en xarxes de computadors, hem de saber que disposem de moltes ordres per a manejar la configuració de xarxa del sistema operatiu. Les més importants són `ifconfig`, amb la qual podem veure els dispositius configurats, `ifdown` i `ifup`, que ens permeten apagar o engegar la interfície que volem, i `route`, que ens mostra la taula d'encaminament del sistema.

6.5.3. Targeta Wi-Fi

La xarxa sense fil (*wireless LAN*), també denominada Wi-Fi, és una tecnologia basada en la norma IEEE 802.11. Les definicions més comunes són les esmenes *b*, *a* i *g* del protocol original. Les normes 802.11b i 802.11g usen la banda de 2.4 GHz, per la qual cosa poden interferir amb altres aparells que usen la mateixa banda, com telèfons sense fil. La norma 802.11a usa la banda de 5 GHz.

L'ús de l'espectre de freqüències de ràdio són fortament regulades pels països. La majoria dels dispositius de targetes WLAN no es pot distribuir com a codi obert. La part del dispositiu que defineix les freqüències de les ones de ràdio es distribueix normalment com a *firmware*, la qual cosa el fa, en principi, immodificable.

Les xarxes de comunicació sense cables són cada vegada més freqüents, tant en instal·lacions domèstiques com en institucions, escoles o empreses. Per a la seva instal·lació, s'ha de disposar del que es denomina *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 Wi-Fi (sia PCI, PCMCIA, integrada a la placa base o a un adaptador USB), qualsevol ordinador del seu 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 *wireless* hem d'afegir al nucli del sistema els mòduls necessaris, que en molts casos ja són compilats en el mateix nucli. Si no hi tenim cap altra targeta Ethernet instal·lada, es referenciarà com a `eth0`, o `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-hi la configuració necessària perquè s'hi assigni una IP. Naturalment, aquesta interfície del sistema

la podem tractar com qualsevol altra, utilitzant els mecanismes de *firewall* del sistema, amb les aplicacions *iproute2*, etc. A continuació, veurem un exemple de configuració de la targeta *eth1* que és Wi-Fi:

```
# wireless network
auto eth1
iface eth1 inet dhcp
    wireless_essid miwifi
    wireless_channel 6
    wireless_mode managed

    wireless_keymode open
    wireless_key1 millavehexadecimal
    wireless_key2 s:millaveascii
    wireless_defaultkey 1
```

Si busquem en el manual de l'ordre `iwconfig` podem trobar el significat de cadascuna de les ordres anteriors.

6.5.4. Mòdems

Per a la configuració d'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 fer uns determinats passos, 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 la/les IP dels servidors de noms que s'han d'utilitzar. Si tenim aquestes IP, hem d'indicar que utilitzem una configuració estàtica, amb la qual cosa seguidament se'ns demanaran aquestes IP. Només en cas que el nostre proveïdor ens indiqui que la configuració de DNS és dinàmica, hem de triar 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 n'hauríem d'informar adequadament.

Paquets ppp i pppconfig

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

Observació

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.

4) Nom d'usuari i contrasenya: aquesta és la informació que ens proporciona el proveïdor per tal de poder-nos connectar i accedir als seus serveis.

5) Velocitat del mòdem: segons quin mòdem tinguem, podrem accedir a Internet a una velocitat més alta o més baixa. Actualment, tots van a 115.200 bps, amb la qual cosa el més recomanable és deixar el valor 115.200. Si tinguéssim un mòdem més lent, ja se sol detectar i reconfigurar automàticament en el moment de la connexió.

6) Trucada amb polsos o tons: la majoria de les centraletes telefòniques ja funcionen amb tons, encara que a determinades zones rurals encara s'utilitza l'antic sistema de polsos.

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 el qual tenim connectat el mòdem. Si li indiquem que el detecti automàticament, es farà una revisió de tots els ports i es configurarà automàticament. Si no, el podem indicar amb `/dev/ttySX`, en què la `x` és un 0 per al COM1, un 1 per al COM2, etc.

Tota aquesta configuració se sol emmagatzemar en els arxius situats en el directori `/etc/ppp/`. Encara que també podem 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 aturar-lo, podem utilitzar `/etc/init.d/ppp stop`.

6.5.5. ADSL i PPPoE

Asymmetric digital subscriber line (ADSL) és una tecnologia de comunicació que permet la transmissió de dades mitjançant línies telefòniques i cable televisiu. La comunicació asimètrica implica que les taxes de pujada i baixada són diferents, generalment la de pujada és més baixa.

ADSL usa un cable mòdem o un concentrador dependent del mètode que implementa el proveïdor d'Internet. Ambdós connecten els respectius aparells a la interfície de xarxa Ethernet del computador (generalment la primera, `eth0`) i també a un port USB.

La configuració ideal per a la interfície Ethernet seria en mode `dhcp`, i que fos el cable mòdem el que proporcionés l'adreça IP, DNS, etc. ja que d'aquesta manera es configura tot automàticament.

PPPoE (*point-to-point protocol over Ethernet*) és un protocol molt similar a PPP que usa una interfície Ethernet en comptes d'un mòdem. Aquest tipus de protocol ha estat adoptat per alguns proveïdors ADSL d'Internet.

La connexió ADSL és generalment part del servei telefònic. Els senyals (de telèfon i ADSL) es divideixen en un aparell anomenat *splitter* que els separa en respectius rangs de freqüència. El senyal ADSL després entra en un aparell denominat concentrador que es connecta a la interfície Ethernet de l'ordinador amb un cable *twisted pair* amb connectors RJ45 o a un dels ports USB de l'ordinador.

6.5.6. Targeta de so

La targeta de so necessita la inserció d'un mòdul del nucli del sistema per a poder funcionar correctament. Si hi tenim instal·lada l'aplicació `discover`, podem descobrir quin mòdul és el que es correspon amb la nostra targeta per mitjà de l'ordre `discover ##module sound`. També podem conèixer el fabricant i *chipset* mitjançant l'ordre `lspci | grep audio`.

Per a instal·lar-hi el mòdul, podem utilitzar les ordres `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 podem utilitzar la targeta de so adequadament, generalment també se sol instal·lar la infraestructura de so ALSA (*advanced linux sound architecture*). Generalment, la majoria de les distribucions el solen incloure per defecte, encara que si no és així, es pot instal·lar amb el paquet corresponent.

Perquè un usuari normal pugui usar els dispositius del so és necessari que sigui membre del grup `audio`,

```
# addgroup usuari audio
```

Això farà que `usuari` sigui afegit a la llista d'usuaris membres del grup `audio` en l'arxiu de configuració `/etc/group`.

6.5.7. Impressora

En GNU/Linux, la configuració d'impressores es pot fer amb moltes aplicacions diferents. Encara que el `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 altres més fàcils de configurar i gestionar. A continuació, en 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 del `lpd`. És molt recomanable utilitzar algun tipus de filtre automàtic com `magicfilter` o `apsfilter` per a configurar les

ALSA

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

impressores. Aquest tipus de filtre detecta automàticament el tipus de fitxer que s'imprimirà i prepara la impressió adequadament (utilitza un filtre anomenat IFHP).

- `lprng`: aplicacions basades en `lpr` amb l'avantatge que incorporen 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 les ordres de `lpr` i també serveix per a xarxes WindowsTM. Utilitza un conjunt de filtres propis i suporta la gran majoria de les impressores del mercat.

Encara que totes aquestes aplicacions tenen 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 en el qual hi ha 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 estalviarem 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 la IP del servidor d'impressió i la cua).

Si volem configurar un servidor d'impressió per a xarxes WindowsTM o configurar una impressora d'un servidor WindowsTM des d'un client GNU/Linux, hem d'utilitzar un altre tipus de programes. Samba és un conjunt d'aplicacions de GNU/Linux que utilitzen els protocols de les xarxes WindowsTM. Encara que les seves funcionalitats van molt més enllà de la configuració d'un servidor o client d'impressió, per a poder utilitzar impressores en WindowsTM haurem d'utilitzar aquest conjunt d'aplicacions o bé les que ens proporciona `CUPS`.

Configuració segura del servidor d'impressió

En configurar un servidor d'impressió, és important que configurem adequadament des de quines màquines/usuaris permetem la impressió. Altrament, un atacant podria aprofitar la vulnerabilitat i aprofitar-se dels nostres recursos, deixar la impressora sense paper, etc.

Swat

Swat (Samba Web Administration Tool) és una eina molt útil per a la configuració d'un servidor de samba.

7. *Daemons* i *runlevels*

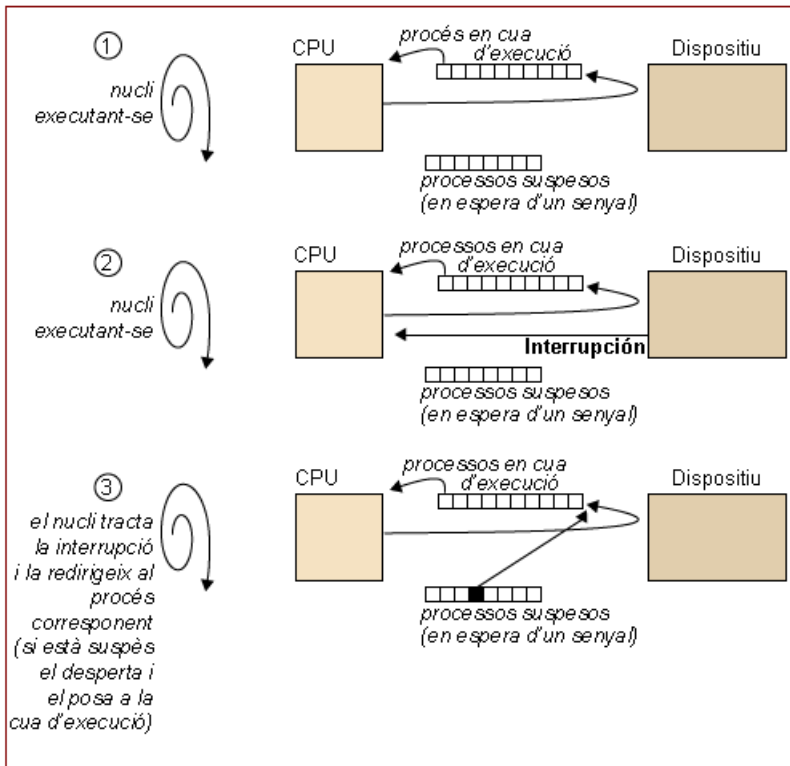
7.1. Els *daemons*

Com ja sabem, GNU/Linux ens permet executar simultàniament tants processos com vulguem i repartir 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 es troba en execució) i redirigir-hi 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 tractat de manera adequada.

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 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 estan esperant.

Un *daemon* (*disk and execution monitor*) és un procés que, generalment, tenim carregat en memòria, que espera algun senyal (que provingui d'una interrupció de dispositiu o del mateix nucli) per a despertar-se i executar les funcions necessàries per a tractar-lo. Encara que aquesta definició també pot encaixar amb altres processos del sistema (llançats pels usuaris o pel mateix sistema), un *daemon* també se sol ajustar a aquest mode d'execució (encara que en alguns casos especials, no). D'aquesta manera, els *daemons* que tinguem carregats no ocupen la CPU mentre no sigui estrictament necessari i per molts que en tinguem en memòria sempre podrem treballar amb l'ordinador sense problemes.

Figura 7.1



Shell scripts dels daemons

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

Encara que un *daemon* sigui un procés com qualsevol altre que s'executa en mode *background*, la manera com els organitzem i tractem sí que és diferent de la resta d'ordres i programes del sistema. Generalment, tots els *daemons* tenen un *shell script* situat en el directori `/etc/init.d/` que ens permet iniciar-lo, aturar-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 i passar-li algun dels paràmetres següents:

- `start`: per a iniciar el *daemon*. Si aquest ja s'estigués executant, es mostra un missatge d'error.
- `stop`: per a aturar el *daemon*. Si no s'estigués executant, es mostra un missatge d'error.
- `restart`: reinicia el *daemon*. Serveix perquè es tornin a llegir els seus arxius de configuració.
- `reload`: encara que no tots els *daemons* ho permeten, aquest paràmetre serveix per a poder recarregar els arxius de configuració sense haver-lo d'aturar.

La majoria d'aquests *scripts* utilitza un programa anomenat `startstop-daemon` que ens proporciona el sistema operatiu i que serveix per al tractament d'aquests processos. És habitual que en administrar un servidor ens haguem de dissenyar els nostres propis *daemons* per fer alguna tasca concreta. En el directori en el qual se situen tots els *shell scripts* dels *daemons* també se'n sol trobar un d'exemple (`/etc/init.d/skeleton`) perquè el puguem utilitzar quan en necessitem configurar 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 $DAEMON
    sleep 1
    start-stop-daemon --start --quiet --pidfile \
      /var/run/$NAME.pid --exec $DAEMON echo "."
    ;;
  *)
    N=/etc/init.d/$NAME echo " Usage: $N {start|stop| \
      restart|force-reload}" >&2
    exit 1
    ;;
esac
exit 0
```

Execució d'un *daemon*

Per tal d'executar un *daemon*, l'hem de cridar amb la seva ruta completa (`/etc/init.d/nomDaemon`) i passar-li el paràmetre que ens interressi. Algunes distribucions incorporen l'ordre `service`, que permet fer el mateix sense haver d'especificar la ruta completa.

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 executarem (`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 aturar-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 tractarem, encara que, com a mínim, tots han de tenir aquesta estructura.

Programació dels *daemons*

Tot i que els *daemons* són programes com qualsevol altre, la seva programació difereix una mica de les aplicacions d'usuari perquè han d'incloure funcions per a quedar aturats temporalment i esperar senyals per a ser despertats, etc.

7.2. Els *runlevels*

Els *daemons* que tinguem en execució en un moment determinat ens marquen els serveis que el sistema operatiu està oferint i/o rebent. El fet que puguem tenir tants *daemons* diferents provoca que haguem de plantejar la seva organització de manera adequada. 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 determinats es puguin oferir uns serveis concrets 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 cadascun d'ells, generalment els sistemes like UNIX ens en proporcionen 6 de diferents amb les propietats següents:

Nivell	Funcionalitat
0	El nivell d'execució 0 està configurat per a aturar el sistema.
1	Aquest nivell es denomina <i>single user</i> , ja que només permet l'entrada al sistema al seu <i>root</i> . S'arrenquen els <i>daemons</i> 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. Aquests nivells també s'anomenen multiusuari, 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.

L'ordre necessària per a canviar de nivell d'execució és `init` (li passem com a paràmetre el nivell d'execució que vulguem) i per a veure on som, `runlevel`.

Les ordres `halt`, `reboot`, `shutdown` o `poweroff` l'únic que fan és cridar el nivell d'execució 0 o 6 i abans fan alguna operació concreta (podeu veure el seu manual per a més informació). Només l'usuari *home* del sistema pot utilitzar totes aquestes ordres.

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/` en què 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 aturar el *daemon* al qual apunten. Amb el mateix nom de l'enllaç s'identifica l'acció que farem: si l'enllaç comença per "S" (*start*) indiquem que volem iniciar el *daemon*, mentre que si comença per "K" (*kill*) indica que el volem aturar. Si el nom no comença per cap d'aquestes lletres, el sistema no fa res amb ell. Després d'aquesta lletra es posa un nombre de dues xifres entre 00 i 99, que indica el seu ordre d'inici o aturada. Aquest ordre és important, ja que alguns *daemons* necessiten que 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* i passar-li com a paràmetre `start` o `stop`, de manera que si n'aturem algun que no s'estigui executant en el moment d'aturada, no passaria res perquè el mateix *shell script* ho tindria en compte. Això ens serveix per a poder canviar de nivell d'execució sense tenir en compte el nivell anterior on érem. En la taula següent podem veure un exemple de configuració per a tres nivells d'execució:

Taula 7.2

Nivell d'execució 2		Daemons en execució
K50sshd	S10syslogd	syslogd
K51apache-ssl	S12kernel	kernel
K52ftpd	S20dhcpd	dhcpd
K53telnet	S50proftpd	proftpd
	S90apache	apache
Nivell d'execució 3		
K50dhcpd	S10syslogd	syslog
K51proftpd	S12kernel	kernel
K52apache	S20sshd	sshd
K53ftpd	S50apache-ssl	apache-ssl
K53telnet		
Nivell d'execució 4		
K50dhcpd	S10syslogd	syslogd
K51proftpd	S12kernel	kernel
K52apache	S20tftpd	tftpd
K53ftpd	S50telnet	telnet
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 cadascun d'ells, etc. Cada línia del fitxer és una directiva amb la sintaxi: `<id> :<run-levels> : <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 farem i el procés que es llançarà. 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 cal 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 en el 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
0:0:wait:/etc/init.d/rc 0
1:1:wait:/etc/init.d/rc 1
2:2:wait:/etc/init.d/rc 2
3:3:wait:/etc/init.d/rc 3
4:4:wait:/etc/init.d/rc 4
5:5:wait:/etc/init.d/rc 5
6:6:wait:/etc/init.d/rc 6

# Ordre que cal executar en prémer CTRL+ALT+DEL
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now

# Definició de les consoles obertes en 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
```

Observació

En aquest fitxer també podríem configurar un terminal que es comuniqués 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 en comunicar-nos amb una línia telefònica.

Com veiem, en aquest fitxer es configura tot el que fa referència als nivells d'execució de manera molt flexible, per la qual cosa podem canviar el que ens interressi per adaptar-ho millor a les nostres necessitats. Fixem-nos que, encara

que aquí definim el nivell d'execució per defecte, també el podríem especificar en engegar el sistema amb 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 `single` a Grub), només s'iniciaran les funcions més necessàries i podrem entrar per tal d'arreglar el que sigui necessari.

7.3. L'arrencada del sistema

El procés pare de tots els altres és `l'init`. Aquest procés s'encarrega d'arrencar 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 *run-levels* 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 estem fent, ja que generalment són imprescindibles per al bon funcionament del sistema operatiu. Una vegada s'han arrencat aquests *daemons* (o *shell scripts*), s'entra en el nivell d'execució configurat per defecte, i s'aturen i s'inicien els *daemons* que hi hagi especificats. Una vegada apresada tota aquesta organització, ja podrem adaptar l'arrencada del sistema a les nostres necessitats i crear i situar els *daemons* que vulguem a qualsevol dels llocs que haguem vist.

7.4. Daemons bàsics

Segons la distribució de GNU/Linux que utilitzem, el mateix procés d'instal·lació ja configura uns *daemons* o uns 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 aquest subapartat veurem com funcionen aquests tres *daemons* bàsics i com els podem configurar. És important saber manejar aquests *daemons* bàsics 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* del sistema operatiu s'encarrega de generar perquè quedi constància de qualsevol acció que s'hi faci. El *daemon* encarregat de dur a terme aquestes tasques és `sysklogd`, 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 a on volem redirigir els *logs* (a un fitxer, a una consola, etc.). En les taules següents 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* són de text i els podem 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 les ordres `last` i `lastb`. Si tinguéssim configurats aquests *logs* en algun altre fitxer, també els podríem veure passant el paràmetre `-f` *fitxero* en l'ordre `last`.

Taula 7.3

Selector			
Servei	significat	Prioritat	significat
authpriv	Missatges d'autoritzacions o d'aspectes de seguretat.	emerg	El sistema és inutilitzable.
cron	<i>Daemon</i> <code>crond</code> i atd.	alert	L'acció s'ha de fer d'immediat.
daemon	<i>Daemons</i> del sistema sense opcions de <i>logs</i> .	crit	Condicions crítiques.
ftp	<i>Daemon</i> 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 <i>daemonsyslogd</i> .		
user	<i>Logs</i> d'aplicacions de nivell d'usuari.		
uucp	Missatges generats pel sistema d'UUCP (Unix-To-Unix Copy Protocol).		
local0~7	Reservats per al seu ús local.		

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

Taula 7.4

Acció	
Destinació	Explicació
Fitxer regular	S'especifica la ruta completa del fitxer. En posar un "-" davant no es requereix que el fitxer sigui sincronitzat cada vegada que s'hi escriu (encara que es perdrà el <i>log</i> en el cas de fallar l'alimentació).

klogd

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

lastlog

Per a veure els últims registres d'entrada dels usuaris, també podem utilitzar l'ordre `lastlog`.

Acció	
Destinació	Explicació
Pipe anomenat	Aquest sistema permet que els missatges es redirigeixin cap a una canonada creada abans d'iniciar el <i>daemon</i> de <code>syslogd</code> amb l'ordre <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	En especificar <code>/dev/ttyX</code> , en què <code>X</code> és un número de consola o <code>/dev/console</code> , els <i>logs</i> es redirigeixen cap a la pantalla especificada.
Màquina remota	Per tal d'especificar que els <i>logs</i> es redirigeixin a una màquina remota, hem de precedir el nom del <i>host</i> remot amb "@".
Usuaris	En especificar el nom d'usuari o usuaris (separats per comes), els <i>logs</i> corresponents s'hi redirigeixen.
Usuaris en línia	Amb "*" especificarem que els <i>logs</i> es redirigeixin a tots els usuaris que en el moment d'ocórrer el <i>log</i> siguin 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 a tots els selectors que es vulgui.)

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 anar comprimint, 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 de manera explícita 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 de manera més amena tot aquest sistema de *logs*. Amb *logger* podem escriure en el sistema de *logs* del sistema. `save` 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). Aquestes ordres també es poden utilitzar per a crear els nostres propis fitxers de *logs* o, si escau, manipular manualment els del sistema (amb el seu manual obtindrem més informació sobre el seu tractament i manipulació).

Observació

Si volguéssim configurar una consola del sistema per a veure tots els *logs* que es van generant, podríem afegir la línia `*.* /dev/ttySX` (en què `X` és la consola en què volem veure els *logs*) en el fitxer `/etc/syslog.conf` i reiniciar el `daemonsyslogd`.

Aspectes legals de l'administració

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

7.4.2. Execucions periòdiques (cron)

Moltes de les tasques d'administració d'un servidor s'han de portar a terme de manera periòdica. També hi ha moltes accions, com l'actualització dels *logs* o les bases de dades internes que utilitzen certes ordres, que s'han d'executar regularment per al seu bon funcionament. Per aquest motiu, és molt important que el mateix sistema operatiu ens proporcionï alguna eina per a 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 cada dia, setmana o mes, respectivament. També existeix el `/etc/cron.d/`, en el qual podem situar arxius amb un format especial per tal de configurar l'execució de determinats programes de manera més flexible.

Configuració del cron

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

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 d'ordres utilitzar i quin serà el seu `PATH`. Les tres línies següents són formades pels camps: `<minut>` `<hora>` `<diaMes>` `<mes>` `<diaSetmana>` `<usuari>` `<ordre>`. Els cinc primers indiquen quan executar l'ordre corresponent (han de ser coherents) i en el sisè trobem l'usuari que s'utilitzarà per a executar l'ordre especificada en l'últim. Ens fixem en com en el fitxer de configuració les ordres que s'executen una vegada al dia, una vegada a la setmana o una vegada al mes són les encarregades de llançar els *shell scripts* que es trobin en els directoris especificats. Si es té el programa *anacron*, s'executen amb ell, si no, s'utilitza el `run-parts`, que encara que no té 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 comentàvem anteriorment. Si volguéssim, podríem canviar les directives d'aquest arxiu per a adaptar-les més a les nostres necessitats.

Programa anacron

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

Configuració d'execucions periòdiques

Si utilitzem el fitxer `/etc/crontab` per a configurar les nostres pròpies execucions periòdiques, cada vegada que el modifiquem hem de reiniciar el `daemoncron`. Si utilitzem la seva estructura de directoris, no és necessari.

Tot i que també podríem utilitzar aquest mateix fitxer de configuració per a posar-hi les nostres pròpies ordres, 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 el 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 utilitzar fitxers particulars per a cadascun d'ells. Amb l'ordre `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`, en els quals podem posar, respectivament, la llista d'usuaris a qui permetem utilitzar el `cron` i a qui no.

7.4.3. Execucions retardades (at i batch)

Si bé el `cron` ens permet fer operacions cada cert període de temps, el `daemonatd` permet executar una ordre 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 l'ordre `at` amb què podem especificar en quin moment volem executar certa operació amb la sintaxi: `at f fichero TIEMPO`. Generalment el fitxer sol ser un programa o *shell script* creat pel mateix usuari en el qual s'escriuen totes les instruccions que es vulguin executar. L'especificació de `TIEMPO` pot arribar a ser molt complexa, ja que 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 formats possibles). Amb `atq` podem veure quines feines tenim retardades i amb `atrm` podem esborrar alguna de les feines

Observació

En utilitzar l'ordre `crontab`, en gravar el fitxer es comprova que la sintaxi sigui correcta.

Càrrega del sistema

La càrrega del sistema és un paràmetre que ens indica el grau d'activitat de l'ordinador. Amb l'ordre `top` podem veure aquesta càrrega de manera interactiva.

que siguin a la cua. Finalment, si volem executar tots els treballs en cua de l'`at`, podem utilitzar l'ordre `atrun`. Aquesta ens permet passar-li el paràmetre `-l LOADAVERAGE`, en què `LOADAVERAGE` ha de ser un número que indiqui a partir de quin moment de càrrega del sistema es podran executar les ordres retardades. Això enllaça directament amb l'ordre `batch`, que serveix exactament per al mateix que `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.

Mecanismes de cua d'execució

Tot el sistema d'`at` i `batch` funciona amb uns mecanismes de cua d'execució (una per a cadascun dels dos). Encara que en podem configurar més, generalment només amb aquestes ja tenim suficient per a fer qualsevol tipus d'execució retardada en el sistema.

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

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, biblioteques o components de programari empaquetats en un sol arxiu preparat per a ser instal·lat i integrat en el 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 crear-ne 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 noves aplicacions, ja que algunes biblioteques 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 les distribucions utilitza un dels dos sistemes de paquets més estesos al 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 dels 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 algun format estàndard, com el *tar*). A partir d'aquest codi font, també podem instal·lar el programa en el nostre sistema operatiu, compilar-lo i situar els executables al lloc on els correspongui.

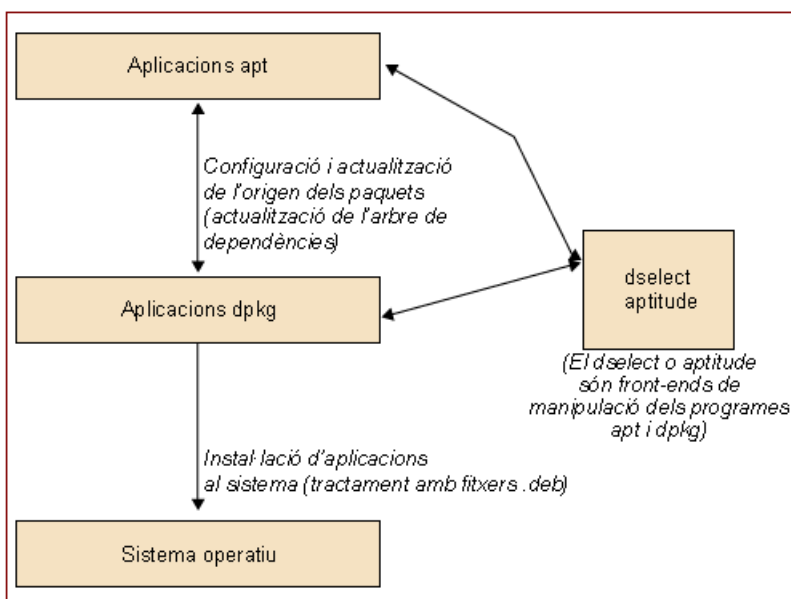
En aquest apartat veurem com s'organitza el sistema de paquets de la distribució Debian⁵ per la gran quantitat d'eines que proporciona i la flexibilitat de la seva configuració. En l'últim subapartat, aprendrem com instal·lar un programa a partir del seu codi font, ja que en alguns casos ens podem trobar que el programa que necessitem no estigui empaquetat. Aquest mode d'instal·lació era el que s'utilitzava sempre abans que apareguessin els primers sistemes de paquets, que van sorgir per a facilitar tot aquest procés.

⁽⁵⁾Debian GNU/Linux va ser la primera distribució que va crear un sistema de paquets.

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 d'on aconseguim els paquets, quins són els que volem i resolen 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 `dselect` o `aptitude`, que serveixen per a manipular els programes `apt` i `dpkg` i proporcionen, en un sol entorn, eines interactives per a la seva manipulació. En la figura següent podem veure aquest esquema:

Figura 8.1



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 de les quals les hagin extret (FTP, CD-ROM, Internet, 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 en el qual 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 cau de paquets (directori en què es desen els arxius `.deb` abans de ser instal·lats).

- `apt-cdrom`: aplicació per a gestionar CD-ROM que continguin paquets.
- `apt-get`: actualització, instal·lació o baixada dels paquets.

Tota la configuració d'`apt` és en el directori `/etc/apt/`. En el fitxer `/etc/apt/sources.list` és on es desa la configuració de les fonts (orígens) dels paquets. Amb totes aquestes fonts es genera una llista de paquets disponibles que podem consultar i instal·lar sempre que ens interessi. 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
```

Programes `dpkg`

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

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 estem utilitzant. Aquest camp és important perquè cada versió de la distribució té els seus propis paquets. En els últims camps podem especificar quin tipus de paquets volem utilitzar.

Paquets des del codi font

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

Si canviéssim aquest fitxer de manera manual, podríem utilitzar l'ordre `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-cdrom add`, amb el qual 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-hi la mateixa aplicació `apt` detectaria quin és el més recent o aquell la baixada del qual implica menys temps 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 les distribucions és la de l'actualització de paquets en els quals s'ha 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 van detectant paquets crítics, es van posant en aquesta font, de manera que només en executar `apt-get update` s'avisava de les noves actualitzacions que hem de fer en el sistema i s'hi 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, al nivell d'usuari, perquè puguem dur a terme les operacions bàsiques amb ells. Els principals programes `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`: en un mateix paquet `deb` moltes vegades s'inclou algun mecanisme per a configurar algunes de les opcions de l'aplicació de manera interactiva. Amb aquesta aplicació podem tornar a configurar el paquet que li indiquem amb els mateixos mecanismes utilitzats a la seva instal·lació.
- `dpkg-scanpackages`: aquest programa serveix per a escanejar un determinat directori del sistema que contingui arxius `.deb` perquè es generi un arxiu d'índex. En aquest arxiu d'índex podem incloure el directori com una font més d'`apt`. Molt útil quan baixem programes no oficials de la distribució.
- `dpkg-scansources`: 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 en 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 li podem passar el paràmetre `-l`, amb el qual 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-hi nous paquets podem utilitzar el paràmetre `-i` seguit del nom de l'arxiu. Si ens dona problemes de dependències, les podem ignorar amb `##ignoredepends=x`, en què la `x` indica la dependència, encara que hem 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 a veure què conté, també podríem

Utilitat de `dpkg`

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

utilitzar `-x`. Per a eliminar els paquets, hem de passar `-r` seguit del nom del paquet, que l'elimina del sistema, però desant els seus arxius de configuració (amb `-P` s'elimina tot).

Un altre paràmetre molt interessant és el de `##force-things x` (en què 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 nou paquet que triem.
- `downgrade`: instal·la el paquet encara que n'hi hagi versions més noves.
- `remove-essential`: encara que el paquet estigui considerat com a essencial en el sistema, l'elimina.
- `depends`: no té en compte les dependències, les considera com a 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.
- etc.

Encara que tots els programes que hem anat comentant al llarg d'aquest subapartat tenen moltes opcions i hi ha 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 `dselect` o `aptitude` ja podrem dur a terme les tasques bàsiques d'instal·lació i eliminació de paquets, és important conèixer adequadament aquestes altres ordres perquè per a fer operacions específiques o per a automatitzar els processos de selecció i instal·lació poden ser molt útils.

8.3. Compilació de nous programes

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 a aconseguir el seu executable i baixar-ne directament el binari. Quan compilem un programa, aquest utilitza les biblioteques disponibles en el sistema, mentre que si el baixem directament, el més probable és que no funcioni adequadament perquè intentarà utilitzar alguna biblioteca que no serà exactament igual que la que

tinguem instal·lada en el sistema. Per això, el més recomanable, quan necessitem instal·lar-hi 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 un fitxer anomenat `README` en el qual s'expliquen pas a pas totes les accions necessàries per a compilar correctament el programa. Encara que és recomanable llegir-lo, en la majoria dels 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 biblioteques s'utilitzen, com s'han de compilar els arxius de codi, etc. Si tenim aquest `Makefile`, ja podem compilar el programa executant `make`. No és necessari 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, en executar `make install` el mateix programa s'encarrega de situar adequadament els binaris i, si existissin, els arxius de documentació. Finalment, si no ens interessés desfer el codi font del programa, ja podem eliminar tot el contingut dels directoris creats.

Compilació d'un programa

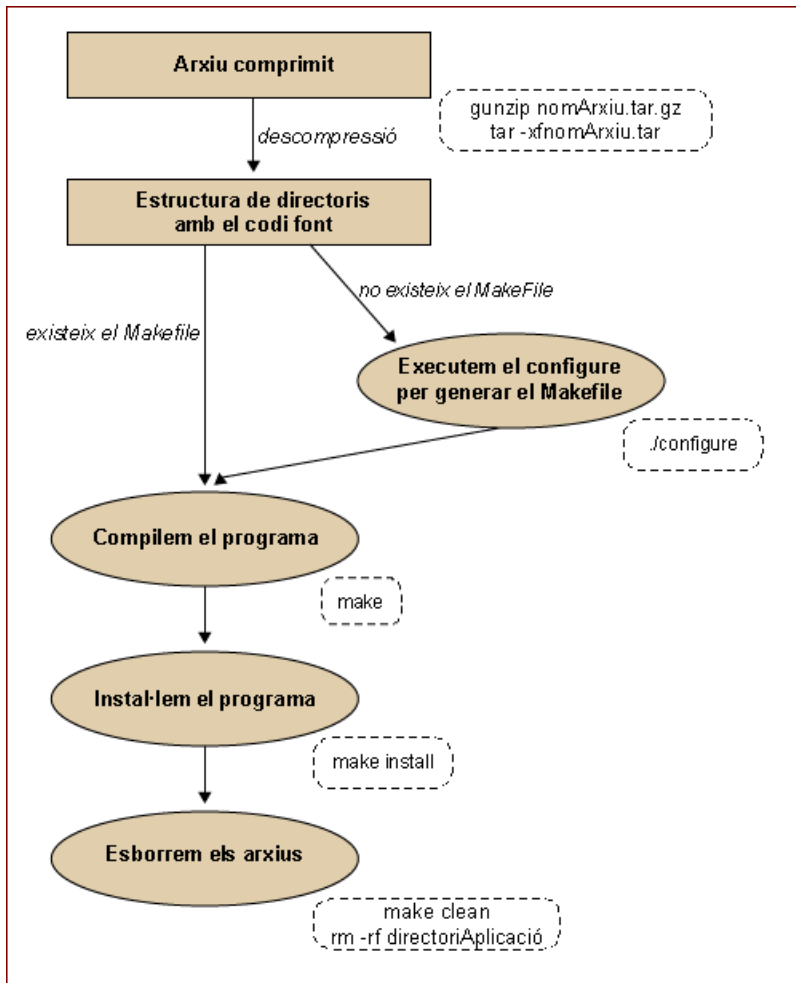
En compilar un programa és molt probable que necessitem tenir instal·lades en el sistema les fonts o les capçaleres de les biblioteques que utilitza. Generalment, aquests paquets solen tenir el mateix nom que la biblioteca, però afegint "-dev" (de desenvolupament) al final.

Temps de compilació

El procés de compilació d'un programa pot durar des de segons a hores, segons l'aplicació. Per exemple, la compilació del nucli Linux va des de 5 o 10 minuts a les 2 hores (segons la versió del nucli i la potència de l'ordinador).

Si el programa no incorpora l'arxiu de `Makefile`, generalment s'hi 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 biblioteques 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 de manera gràfica:

Figura 8.2



Tot i que la majoria del codi font dels programes s'organitza tal 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'objectiu 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 establertes les bases per a poder-hi 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 instal·larem i en configurarem algunes, que són comunes i necessàries en la majoria dels 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 usen aquest sistema per a poder funcionar.

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 hi tenim instal·lat el gestor d'arrencada i hem comprovat el seu funcionament correcte, podem passar al subapartat següent; però si no és així, és a dir, si per a engegar el nostre sistema operatiu necessitem el disquet de reparació que vam crear durant el procés d'instal·lació, ha arribat el moment d'instal·lar un sistema de gestió d'arrencada en el disc dur per tal d'evitar la tediosa tasca d'utilitzar cada vegada el disquet. De qualsevol manera, sempre és interessant tenir-hi instal·lat un sistema gestor d'arrencada, ja que ens permetrà, entre altres coses, poder arrencar diferents *kernels*.

Tal com s'ha exposat, hi ha diferents gestors d'arrencada, entre ells Lilo i Grub. Grub és un poderós sistema gestor d'arrencada, del projecte GNU, que es caracteritza per poder gestionar correctament l'arrencada de qualsevol sistema operatiu que tinguem instal·lat al nostre ordinador; no obstant això, el seu ús i la seva configuració són una mica complexos.

De qualsevol manera, si no disposem de gestor d'arrencada i en volem instal·lar un, el primer que hem de fer és arrencar el sistema operatiu que tenim i comprovar si l'aplicació està instal·lada, per a la qual cosa farem, per exemple, un man de l'aplicació.

9.2.1. Instal·lació de Grub

Actualment, Grub és el gestor d'arrencada que s'instal·la per defecte. De tota manera, si el volguéssim tornar a instal·lar, podem procedir a llegir els subpartats posteriors per informar-nos sobre el sistema gestor de paquets, la seva configuració i el seu ús; o simplement en cas que durant la instal·lació haguem introduït en la base de dades els continguts de tots els CD/DVD o que haguem optat per una instal·lació per xarxa, executarem la línia següent:

```
brau:~# apt-get install grub grub-doc
```

Si no hi 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 en l'MRB, les proves que s'estimin oportunes sobre un disquet i provar el seu correcte funcionament arrencant des d'aquest. Per a això, el primer que farem és copiar els fitxers necessaris a /boot/:

```
brau:~# cp /usr/lib/grub/i386-pc/stage* /boot/  
brau:~# 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 respecte als discos durs; així doncs, per a identificar a quin dispositiu es troba un fitxer en concret, ens podem ajudar de l'ordre `find` una vegada haguem entrat en el mode d'ordres 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 pot ser /vmlinuz, procedirem de la manera següent:

```
brau:~# 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 disposa de moltes més ordres; per a veure una llista d'algunes d'elles n'hi ha prou que premem la tecla "Tab" en la línia d'ordres 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'alguna ordre en concret, n'hi ha prou que escrivim `help` seguit del nom d'aquesta ordre. Una bona manera de treballar sobre el nostre fitxer de configuració de Grub, `/boot/menu.lst`, és obrir una nova sessió en una tty diferent i anar modificant aquest fitxer a mesura que anem treballant sobre la interfície d'ordres de Grub.

Una vegada haguem acabat d'editar el fitxer de configuració, i després d'introduir un disquet verge a la disquetera, escriurem –hi substituïrem el caràcter `X` pel número de disc dur, i `Y` per la partició corresponent– la línia d'ordres 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 d'ordres (`stage2`), i el menú d'arrencada que haguem configurat en el fitxer `/boot/menu.lst`. Ens trobem, 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 d'ordres, podem intentar arrencar directament els diferents sistemes operatius que tinguem instal·lats per tal d'anar fent 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, les ordres necessàries 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 en el 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 d'ordres del disquet escriurem:

```
grub> install (hdX,Y)/boot/stage1 d (hd0,0) \ (hdX,Y)/boot/stage2 p
(hdX,Y)/boot/menu.lst
```

Com s'ha vist, utilitzar Grub és una mica complex, per la qual cosa es recomana que abans d'embarcar-se a la seva instal·lació i posterior configuració, es llegeixin detingudament els *man* i la documentació que s'ha instal·lat amb el paquet `Grub-doc`, també 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 existeixen 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, que el lector pugui optar per la que més li interessi, i, el segon, que conegui sempre més d'una solució a un mateix problema per si alguna d'elles falla.

Activitat

9.1. Per a la plena comprensió del 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* de: `apt`, `apt-cache`, `apt-get`, `sources.list`, `dpkg` i `dselect`.

Per a fer aquestes lectures més còmodes, podem esperar a fer-les quan tinguem configurada la impressora i haguem après a imprimir *man*.

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. En tractar-se d'un fitxer de text, com la majoria dels fitxers de configuració en els sistemes UNIX, el podem editar manualment o mitjançant algunes eines de les quals disposa el sistema per a aquesta finalitat. El contingut d'aquest fitxer dependrà en gran manera 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 haguem modificat el fitxer:

```
brau:/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 efectuats no tindria efecte. En una instal·lació en què els paquets s'obtinguin de manera remota, aquesta ordre s'ha d'executar periòdicament per a anar actualitzant 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 forma d'accedir-hi: *CD-ROM*, *http*, *ftp*, etc. seguit de l'adreça d'accés. Els camps restants fan referència al tipus de paquets al qual volem accedir per aquesta línia.

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 la 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 ni tan sols no ens hem de preocupar d'inserir el CD corresponent per a dur a terme una instal·lació.

Del que ens hem d'assegurar, abans de res, és 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://non-us.debian.org/debian-non-US stable/non-US main \ contrib non-free
deb-src http://non-us.debian.org/debian-non-US stable/non-US main \ contrib non-free
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 per no ser oficials no significa que no tinguin qualitat suficient per a ser inclosos en el nostre fitxer de configuració. Una bona adreça per a obtenir informació sobre la localització d'aquests paquets és <http://www.aptget.org>.

Una vegada editat el fitxer i desat, executarem l'ordre *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 s'efectuen, ja tindrem accés als nous paquets.

En el cas de no disposar de connexió a Internet o que sigui lenta, hem d'optar, sense dubte, per utilitzar el joc de CD/DVD de la distribució per anar instal·lant els diferents paquets. Si durant el procés d'instal·lació no hem inserit tots els CD/DVD, haurà arribat el moment de fer-ho. Inserim el primer CD-ROM/DVD al lector i usem l'ordre `apt-cdrom add` per a incloure els seus continguts en la base de dades:

```
brau:/etc/apt# apt-cdrom add
Using CD-ROM mount point /CD-ROM/
.
.
.
Repeat this process for the rest of the CD in your set.
```

Arribats a aquest punt, repetirem el mateix procés per a tots i cadascun dels nostres CD/DVD de la distribució. Així mateix, es pot utilitzar el mateix procediment per a incorporar dades procedents de CD/DVD no oficials. Una vegada tinguem configurat el nostre accés a Internet, si ho considerem 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 nous paquets.

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`. La primera ordre la 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 la segona, en ser una ordre orientada a la recerca d'informació dins de la base de dades, tant si són paquets instal·lats com sense instal·lar, pot ser utilitzada per qualsevol usuari.

Amb l'objecte 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, hem de conèixer alguns conceptes sobre aquests i sobre la seva relació amb el sistema i el sistema de gestió.

Tipus de paquets segons la seva prioritat

Dins del sistema de paquets es distingeixen cinc tipus diferents segons el seu grau de dependència amb el mateix sistema. Per ordre decreixent de prioritat es classifiquen com a:

- **Required.** Es tracta de paquets indispensables per al correcte funcionament del propi sistema.
- **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 o no 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 el seu 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 ser integrats 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. Pot ser, per exemple, que una determinada aplicació que volem instal·lar depengui d'una biblioteca i, 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 biblioteca. 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 instal·laran aquest paquet també usaran els que recomana.
- **suggests.** Els paquets que se suggereixen permeten obtenir un rendiment del paquet que instal·larem més gran.

- `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 dels paquets esmentats.

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 mantenir els seus 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 es mantingui fins a nou avís la seva versió i la seva configuració).

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.
- `half-installed`. La instal·lació del paquet s'ha començat, però, per alguna raó, no ha acabat.
- `not-installed`. El paquet no és instal·lat en el sistema.
- `unpacked`. El paquet ha estat desempaquetat, però no ha estat configurat.
- `config-files`. En el sistema només hi ha els arxius de configuració d'aquest paquet.

apt-cache

Com ja s'ha dit, `apt-cache` és una ordre orientada a l'anàlisi del sistema de paquets i, per tant, en no ser una arma potencialment perillosa per al sistema, és accessible a tots els seus usuaris. Els paràmetres més utilitzats per a aquesta ordre són els següents:

- `search pattern`. Busca en la base de dades els paquets el nom dels quals contingui *pattern* o en la descripció dels quals 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, així com sobre la seva procedència.
- `depends package`. Explicita les dependències del paquet.
- `show package`. Mostra les dependències directes i les reserves del paquet.

apt-get

`apt-get` és l'ordre que s'utilitza per a gestionar els paquets del sistema. Per aquest motiu, el seu ús està restringit al *root* del sistema. Els paràmetres més utilitzats per a aquesta ordre 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`. Aquesta ordre s'ha d'executar cada vegada que es modifica l'arxiu `/etc/apt/sources.list`.
- `upgrade`. Força l'actualització de tots els paquets instal·lats en el sistema per l'última versió disponible.
- `remove package`. Elimina el paquet, sense eliminar els fitxers de configuració, de cara a possibles reinstal·lacions.
- `remove -purge package`. Elimina per complet el paquet, inclosos els seus arxius de configuració.
- `clean`. Elimina les còpies caducades dels paquets que s'han anat instal·lant, 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. Ordre molt útil de cara a alliberar espai del disc dur, ocupat per fitxers que, probablement, mai més no seran utilitzats.

- `autoclean`. Elimina totes les còpies no desempaquetades dels paquets, independentment de la seva vigència.

9.3.3. `dpkg`

`dpkg` és l'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 en el 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 que escrivim l'ordre `dselect` i, mitjançant els menús d'aquesta interfície, seleccionem els diferents paquets sobre els quals volem operar i especifiquem quin tipus d'operació volem fer sobre ells.

9.3.5. `aptitude`

`aptitude` és una altra GUI que corre sobre `apt`. Per defecte no és instal·lada, per la qual cosa cal fer-ho abans de procedir al seu ús:

```
brau:/etc/apt# apt-get install aptitude
```

Una vegada instal·lada, la llancem mitjançant l'ordre `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. *locales*: configuració regional

Tot i que aparentment el nostre teclat funcioni correctament, ja que podem utilitzar accents, dièresis i altres caràcters no anglesos, a mesura que anem adaptant el sistema a les nostres necessitats, i especialment quan hi instal·lem el sistema gràfic i fem córrer aplicacions sobre ell en el proper taller, ens adonarem que això no és així. Podem, doncs, en aquest punt procedir a configurar correctament aquests aspectes per no haver-ho de fer més endavant.

En primer lloc, comprovarem si el paquet `locales` és instal·lat:

```
brau:/# 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-hi el paquet i configurar-lo:

```
brau:/# apt-get install locales
```

I si ja el tenim, escriurem la línia següent per reconfigurar-lo:

```
brau:/# dpkg-reconfigure locales
```

De les moltes opcions que se'ns ofereixen, escollim [*] `es ES UTF8`, és a dir, ens situem sobre aquesta opció i premem la barra espaiadora. Mitjançant "Tab", ens situem sobre `OK` i premem "Intro". En la propera pantalla seleccionem "C". De tornada a la línia d'ordres, editem el fitxer `/etc/environment` per deixar-lo de la manera següent:

```
LC_ALL=es_ES
LANGUAGE=en_US
LC_TYPE=es_ES
LC_MESSAGES=UTF8

LANG=C
```

Per a fer efectiu el canvi, n'hi ha prou que escrivim l'ordre `locale-gen` i sortirem de totes les sessions que tinguem obertes per a carregar la nova configuració.

9.5. L'arxiu principal d'arrencada, `/etc/inittab`

Encara que el procés d'arrencada d'un sistema GNU/Linux és complex, en aquest apartat només es pretén treballar sobre un dels fitxers principals d'aquest procés: `/etc/inittab`. Aquest arxiu indica el procés d'arrencada, entre d'altres, a quin *runlevel* s'entrarà, i definirà quins processos s'arrencaran

Lectura complementària

Per a saber més sobre `locales`, es recomana visitar la pàgina:

<http://www.uniulm.de/ss-masch/locale/>.

de manera automàtica durant el procés d'arrencada. Per a saber a quin *runlevel* ens trobem, n'hi ha prou que escrivim l'ordre `runlevel`. Per a canviar de *runlevel*, com a *home*, usarem la instrucció `init <runlevel de destí>`.

És interessant obrir aquest fitxer i anar-se familiaritzant amb el seu contingut, ja que això ens permetrà comprendre millor el procés d'arrencada d'un sistema GNU/Linux.

9.6. Muntatge de dispositius, /etc/fstab

`/etc/fstab` és el fitxer que conté la informació sobre les particions i dispositius que es muntaran de manera automàtica durant el procés d'arrencada, i les que es poden muntar posteriorment, així com també estableix qui ho pot fer. 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
```

Les primeres línies les ha generat automàticament el procés d'instal·lació i hi podem veure com es distribueixen 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.

Més interessants són 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 es pot llegir el seu contingut; això queda 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 només de lectura; la segona, que no munti 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'utilitzarà.

Les últimes línies del fitxer anterior són 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 WindowsTM. Si bé és cert que és possible muntar un sistema de fitxers des de la línia d'ordres, com per exemple es faria per a muntar el CD-ROM,

```
brau:/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 escrivint tan sols:

```
brau:/etc/apt# mount /CD-ROM
```

9.7. Configuració de dispositius

Una vegada establertes 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 en funció del 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 sobre quin tipus de dispositiu és el que tenim, llegir documentació sobre com aquest 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 faciliten la informació necessària perquè els desenvolupadors de la comunitat puguin escriure el codi necessari per a poder integrar aquests dispositius en el 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

vedades, ni disposen), si aquest està plenament suportat, de quina informació disposem per a integrar el nou producte en el nostre sistema, etc. A grans trets, els aspectes generals que cal considerar, a l'hora de fer una nova adquisició, són: el grau d'estandardització i la qualitat del producte. Pel que fa a l'estandardització, com més estàndard sigui el producte, segur que hi ha més usuaris que el tenen i, per tant, és molt més probable que estigui plenament suportat. Quant a 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 *winmodems*). 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 subapartat 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 l'ordre `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/`, en el qual 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 a veure com evoluciona temporalment el contingut d'aquests fitxers és utilitzar l'ordre `tail` amb el paràmetre `-f` i readreçar la seva sortida a una `tty` que no estiguem usant; a tall d'exemple `tail -f /var/log/messages > /dev/tty10`).

9.7.1. Configuració del ratolí

Igual que s'ha fet en el taller de KNOPPIX, el *daemon* que s'encarregarà de gestionar el ratolí serà `gpm`. Procedim, doncs, a instal·lar el paquet:

```
brau:/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 l'ordre `gpmconfig` (en general la configuració `-m /dev/psaux -t ps2` 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 se solen 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 aturada 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*. Aquest subapartat servirà d'exemple per a mostrar com s'arrenquen i es paren els *daemons*.

Tant l'arrencada com l'aturada 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 aturar el `daemon``gpm`:

```
brau:/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 procedirem a engegar-lo:

```
brau:/etc/init.d# ./gpm start
Starting mouse interface server: gpm
```

Movem el ratolí i observem com en pantalla apareix el seu punter. Ara procedim a analitzar si en arrencar l'ordinador aquest *daemon* s'arrencarà automàticament. El fitxer `/etc/inittab` ens indica a quin *runlevel* s'arrencarà el sistema operatiu: per defecte, el 2; per tant, en aquest arxiu hauríem de trobar una línia com la següent:

```
# 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`:

```
brau:/etc/init.d# ls -l ../rc2.d/ | grep gpm
lrwxrwxrwx 1 root root 13 feb 21 13:03 S20gpm -> ../init.d/gpm
```

Si aquest enllaç simbòlic no hi fos, i volguéssim que `gpm` s'arrencés automàticament durant el procés d'arrencada, l'hauríem de crear manualment mitjançant `ln -s`. Si, al contrari, l'enllaç existís i no volguéssim que `gpm`

s'engegués durant el procés d'arrencada, n'hi hauria prou que hi esborréssim aquest enllaç simbòlic; no obstant això, no és recomanable esborrar els *scripts* de `/etc/init.d`, ja que són molt útils de cara a arrencar i aturar *daemons*.

9.7.2. 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 anat desenvolupant eines prou potents i fiables que ens poden ajudar a estalviar-nos la tediosa tasca de configurar un mòdem manualment. Una d'aquestes eines és `pppconfig`, que és la que es proposa en aquest text per a configurar el nostre mòdem.

Però abans de començar amb la configuració del nostre mòdem, cal posar de manifest que no són realment mòdems tots els dispositius que s'anuncien o es venen com a tals. Com ja s'ha dit, molts fabricants, amb l'objectiu de reduir costos, han anat substituint components físics per programari, i probablement els mòdems van ser els primers dispositius històricament víctimes d'aquestes pràctiques. Cal prestar 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 mòdems falsos. Per aquesta raó, molts d'aquests dispositius s'han vist reduïts a simples esclaus del seu programari.

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>. Com 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, escrivint `pppconfig`, o mitjançant `dpkg -l | grep pppconfig`), el primer que hem de fer és instal·lar-hi l'aplicació:

```
brau:~# 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 en 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 triar l'opció per defecte, assignació estàtica de DNS, o per assignació dinàmica de DNS, si sabem precisament que el nostre ISP 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/nomdeconfiguracio`.

Bibliografia

Per a més informació sobre aquests dispositius i 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>

En la pantalla següent, hem de triar el mètode d'autenticació per a establir-ne la configuració. En general, i llevat de casos excepcionals, triarem 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 les 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), introduïrem el número que s'ha de marcar i que ens ha d'haver facilitat el nostre ISP.

Arribats a aquest punt, `pppconfig` ens proposa l'execució d'autodetecció 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 haguem 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 triar l'opció `Finished Write files and return to main menu`, i després de confirmar l'operació, de tornada 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 d'ordres. Una vegada en la línia d'ordres, 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 monitorar el trànsit de la connexió, entre d'altres.

Establiment i acabament de connexió: pon, poff

Per a establir la connexió, n'hi haurà prou que escrivim la instrucció `pon` seguida del nom de connexió que vulguem utilitzar; si només hem configurat una connexió, no serà necessari especificar el seu nom. En principi, si no se'n restringeix l'ús, `pon` pot ser executat per qualsevol usuari.

Per a acabar la connexió n'hi haurà prou que executem l'ordre `poff`.

9.7.3. Configuració de mòdems DSL

De la mateixa manera que s'ha fet en l'apartat 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-BandwidthManagement-HOWTO>.

PPPoE (*point-to-point protocol over Ethernet*) és un protocol molt similar a PPP que usa una interfície Ethernet en comptes d'un mòdem. Aquest tipus de protocol ha estat adoptat per alguns proveïdors ADSL d'Internet.

La connexió ADSL generalment forma part del servei telefònic. Els senyals (de telèfon i ADSL) es divideixen en un aparell anomenat *splitter* que els separa en respectius rangs de freqüència. El senyal ADSL després entra en un aparell anomenat concentrador que es connecta a la interfície Ethernet del computador amb un cable *Twisted pair* amb connectors RJ45.

Per a configurar la connexió:

```
# apt-get install pppoeconf
```

En executar `pppoeconf`, l'eina escanejarà totes les interfícies de xarxa fins a trobar-ne una que accedeixi a un concentrador. Triem configurar PPPoE en aquesta interfície.

A ENTER USERNAME donem el nom d'usuari que ens assigna el proveïdor i després la clau. A USE PEER DNS escollim modificar automàticament `/etc/resolv.conf`. Optem per executar PPPoE en l'arrencada i, després, triem establir la connexió. Després d'uns segons veureu a la pantalla una cosa semblant a:

```
Every 2s: /sbin/ifconfig ppp0      Sun May 13 12:11:56 2007

ppp0  Link encap:Point-to-Point Protocol
      inet addr:200.89.50.138 P-t-P:10.52.0.3 Mask:255.255.255.255
      UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1492 Metric:1
      RX packets:5 errors:0 dropped:0 overruns:0 frame:0
      TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:3

      RX bytes:190 (190.0 b) TX bytes:199 (199.0 b)
```

9.7.4. 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 se suporten 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-la. Pot ser que la nostra targeta vingui suportada de dues maneres diferents: la primera és que el seu *driver* hagi estat directament compilat dins del propi *kernel*, i la segona és que el *driver* hagi estat compilat en forma modular perquè es carregui posteriorment. La manera

més senzilla de saber si el *driver* de la nostra targeta de xarxa ha estat compilat dins del propi *kernel* és analitzant el missatge de retorn de `dmesg` tal com s'ha fet a 5.4.3.

Si després d'analitzar-se detalladament la sortida de l'ordre `dmesg` arribem a la conclusió que el *driver* per a la nostra targeta no ha estat carregat, podem executar l'ordre `modconf`, que serveix per a carregar mòduls al *kernel* que han estat compilats juntament amb ell, i comprovar si el seu *driver* apareix en la subsecció `kernel/drivers/net`. Si és així, n'hi haurà prou que el seleccionem per a carregar-lo en el *kernel*.

Si la nostra targeta de xarxa no està suportada per defecte, haurem de recórrer a la recompilació del *kernel*.

Una vegada fet això, editarem el fitxer `/etc/network/interfaces` per a passar a la targeta els paràmetres corresponents a la nostra xarxa. 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 de <http://www.tldp.org/HOWTO/NetworkingOverview-HOWTO.html>, també <http://www.fokus.gmd.de/linux/HOWTO/Net-HOWTO> i el *man* d'interfícies. Si no disposem de targeta de xarxa i volguéssim fer proves, sempre podem recórrer al mòdul `dummy`; en aquest cas, el dispositiu en comptes de dir-se `eth0` es denominaria `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 *kernel* durant el procés d'arrencada per a evitar conflictes entre dispositius.

Arrencada i aturada de serveis de xarxa: ifup, ifdown

La reinicialització de tots els serveis de xarxa (els de `/etc/network/interfaces`) es pot efectuar 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 aturar-los. Així doncs, per a la configuració anterior, si volguéssim aturar i tornar a engegar els serveis d'`eth0`, el que fariem és el següent (s'utilitza l'ordre `ifconfig` per a comprovar-ne els resultats):

```
brau:~# 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)

brau:~# ifdown eth0
brau:~# 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)

brau:~# ifup eth0
brau:~# 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)
```

Targeta de xarxa sense fil

En el cas de tenir una targeta de xarxa Wi-Fi, podeu mirar l'enllaç següent:
<http://ftp.cl.debian.org/man-es/wlan.html>.

9.7.5. 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 poder-los estudiar més detingudament sobre format paper.

La impressió en Linux funciona mitjançant un sistema de cues que són manejades per un *daemon* que corre en el moment d'arrencada. Les impressores poden estar connectades directament al port paral·lel o USB de l'ordinador (impressora local) o connectades per mitjà de la xarxa (impressora de xarxa).

Pel que fa 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 estan apareixent al mercat, especialment les de port USB, són impressores generalment dissenyades per a un sistema operatiu determinat. Aquestes impressores són conegudes popularment com a *winprinters* (i la seva gènesi és semblant a la dels denominats *winmodems*). 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>, en la qual trobarem una llista exhaustiva de les impressores existents al mercat i el seu grau de suport.

CUPS (*common unix printing system*) és un sistema d'impressió modular que usa IPP (*Internet printing protocol*) per a manejar les cues. Aquest és un sistema modern i sofisticat, apte per a l'ús amb un entorn d'escriptori com GNOME o KDE. A més, manté opativament el sistema tradicional de línies d'ordre de BSD `lpr`.

La topologia general del sistema d'impressió sota GNU/Linux és la de client-servidor. El servidor, CUPS, és tipus *daemon*, i en conseqüència el manipularem com a tal. L'arxiu de configuració del servidor és `/etc/cups/cupsd.conf`. Hi podem configurar tantes impressores com vulguem.

Per a instal·lar el sistema CUPS hem d'executar l'ordre:

```
# apt-get install cupsys cupsys-client cupsys-bsd printconf foomatic-filters-ppds
```

Aquest últim paquet conté els PPD (*postscript printer definition*) d'Adobe, que és usat per CUPS per a definir les impressores.

CUPS-PDF és una impressora virtual per a imprimir documents en un arxiu PDF. Per a instal·lar-lo en el nostre sistema, executariem l'ordre:

```
# apt-get install cups-pdf
```

En CUPS hi ha dues maneres de configurar les impressores. Una d'elles és des d'una utilitat que proporciona l'escriptori GNOME o el KDE. L'altra és des de la pàgina web. En ambdós casos es requereixen privilegis d'administrador per a configurar les impressores.

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 històricament ha tingut més implementació en el camp de la impressió). Aquests programes ens poden ser d'utilitat, ja que ens permeten passar a format paper l'ajuda de les ordres, fitxers de configuració, etc. per a poder-los estudiar d'una manera més còmoda. Entre aquests trobem `mpager` (dins del paquet amb el mateix nom), o `enscrip` (també empaquetat amb aquest mateix nom i més potent que l'anterior). A continuació, es detallen un parell de línies per a exemplificar el seu ús:

```
man man | mpage -2 -o | lpr
```

Mitjançant aquesta línia readrecem la sortida de `man` de `man` a `mpage`, que li dona format a dues columnes per full, sense marges, i readrecem 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 apaïsada.

9.7.6. Configuració de targetes de so

Atesa la gran quantitat de targetes de so existents al mercat, es fa gairebé impossible donar una descripció de com configurar-les totes. Es recomana la lectura de <http://www.tldp.org/HOWTO/Sound-HOWTO/> i la visita a les pàgines dels dos projectes més destacats quant a so sota GNU/Linux: <http://www.opensound.com/> i <http://www.alsa-project.org/>.

Advanced linux sound architecture (ALSA) és un projecte sota llicència GNU per a proveir Linux de dispositius d'àudio i MIDI.

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, l'ordre `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 l'ordre `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` en el nou grup creat.

Amb això ja tenim configurada la targeta de so. Ara podem comprovar-ho adreçant un fitxer d'àudio directament a `/dev/dsp`, com suggereix <http://www.tldp.org/HOWTO/Sound-HOWTO/> o aplicacions d'àudio que corren sobre ell. Cal esperar a tenir l'entorn gràfic configurat per a poder-la instal·lar.

9.8. Conclusió

En aquest taller hem après a treballar amb el sistema de paquets de Debian, una cosa fonamental, ja que això ens ha permès aprendre a instal·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 propi dispositiu com del sistema en si mateix. Però, en compensació, podem assegurar que, una vegada configurat un dispositiu, aquest funcionarà perfectament i no ens n'haurèm de preocupar més. I tot això tenint en ment que encara no hem configurat l'entorn gràfic. No obstant això, i per tal d'animar a qui hores d'ara pugui 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[...]."
```

Aaron Van CouWenberghe

10. Arquitectura X-Window

10.1. Què és X-Window?

X-Window és una arquitectura de finestres dissenyada a mitjan dècada de 1980 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 correu un sistema tipus UNIX. Encara que l'arquitectura de finestres X-Window ha 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 X.Org. X.Org és una implementació de codi obert del sistema X Window System que sorgeix com a bifurcació de projecte XFree86.

X-Window està dissenyat amb una arquitectura client/servidor. Aquest tipus d'arquitectura significa que el programari està estructurat en dues parts totalment independents (client i servidor) que es comuniquen a partir d'un enllaç de comunicació. Encara que això implica que el disseny i la codificació són una mica més complexos, aquesta arquitectura proporciona una flexibilitat total en el sentit que client i servidor poden estar ubicats en diferents llocs i utilitzar diferents plataformes i/o sistemes operatius. 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 de forma local a la màquina on està situat el client, encara que no és indispensable.

Els components que integren 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.

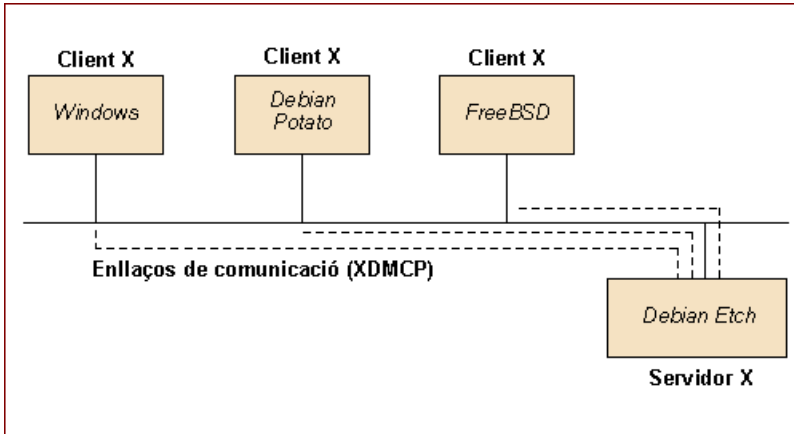
D'aquesta manera, podem utilitzar X-Window en qualsevol tipus d'escenari; per exemple, podríem tenir el servidor instal·lat en un ordinador amb WindowsTM, connectant-se a un client amb GNU/Linux, i utilitzar com a canal de comunicació Internet (protocol IPv4). Encara que la configuració de cadascun d'aquests components (sobretot el client) sí que depèn, en certa manera, de la plataforma en la qual és instal·lat, l'enllaç de comunicació ens permet aïllar els components i donar-los un llenguatge propi per a la seva comprensió.

Arquitectura de finestres

Una arquitectura de finestres (o sistema de finestres) és un entorn gràfic que ens proporciona la possibilitat de tenir diferents aplicacions localitzades en diferents 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 el treball pugui ser més interactiu i amè.

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).

Figura 10.1



En aquesta arquitectura, el servidor està programat 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 client, que mostra els resultats en els dispositius de sortida (generalment el monitor). Encara que la primera impressió que ens pot suggerir 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. Els únics escenaris en els quals podem notar aquest inconvenient és en connexions remotes utilitzant xarxes de comunicacions lentes.

En resum, doncs, les principals característiques i funcions de cadascun dels components d'X-Window són les següents:

Taula 10.1

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	Disseny per a poder treballar interactivament
	Pensat per a minimitzar el trànsit en la xarxa
	Transparent (independent de xarxa)

A mesura que les targetes gràfiques han anat evolucionant, cada vegada hi ha més aplicacions i jocs que 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 de 1998 va aparèixer una tecnologia denominada 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 i aprofitem els elements específics de les targetes gràfiques.

A diferència d'altres sistemes operatius en què l'entorn gràfic està íntimament integrat amb la resta de les funcions, l'arquitectura X-Window és totalment independent del sistema 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 s'inclouen a la biblioteca **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 suposar una feina addicional per als primers desenvolupadors d'aplicacions en XWindow, ja que a més de programar l'aplicació havien de desenvolupar les seves pròpies biblioteques per a la creació de menús, icones, etc. A mesura que X-Window va anar creixent, van anar apareixent el que anomenem **toolkits**, que són biblioteques generalment implementades amb Xlib i que proporcionen un GUI particular.

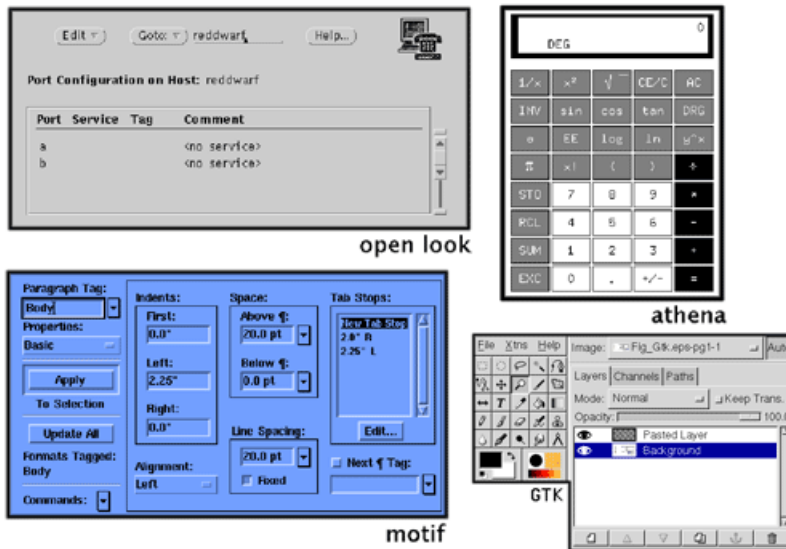
D'aquesta manera, en dissenyar una aplicació podem utilitzar algun d'aquests *toolkits* que ja proporcionen les eines estàndard per a crear menús, botons, gestionar els *cut and paste*, etc. i centrar-nos a programar l'aplicació per si mateixa. El fet de no marcar cap *look and feel* ha estat una altra de les claus de l'èxit de l'arquitectura X-Window, ja que cada fabricant o desenvolupador de programari ha pogut dissenyar-se'n un de propi, amb la qual cosa ha marcat la diferència amb els altres.

Encara que hi ha molts *toolkits* diferents, en la figura següent en podem veure alguns dels més populars que s'han utilitzat al llarg de la història d'X-Window:

Look and feel

El *look and feel* és el disseny utilitzat per als botons, les barres de desplaçament, els menús, etc. d'un entorn gràfic o una aplicació.

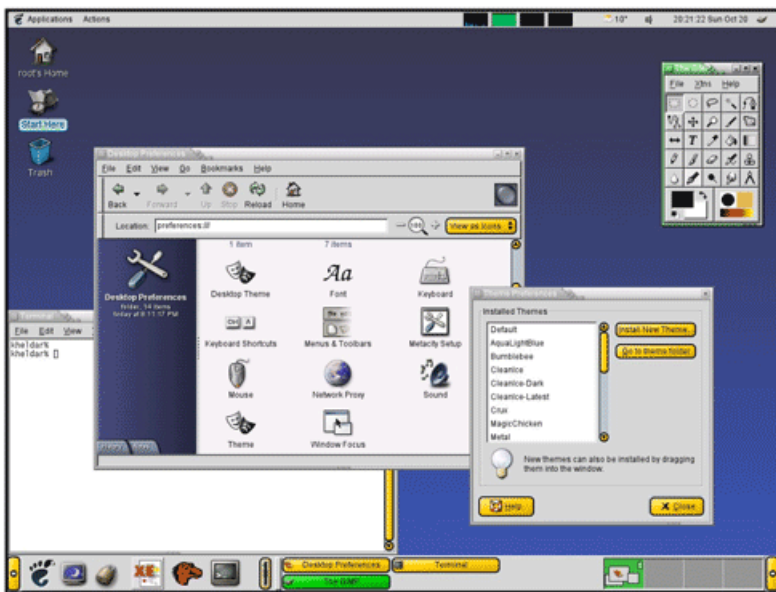
Figura 10.2



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 és ubicat el programa. Encara que la programació d'un *window manager* és molt diferent de la d'una aplicació, també se solen utilitzar *toolkits* particulars que proporcionen un *look and feel* determinat. Actualment hi ha desenes de *window managers* diferents (wmaker, sawmill, olvwm, etc.), i és el mateix usuari qui pot triar el que més li agradi.

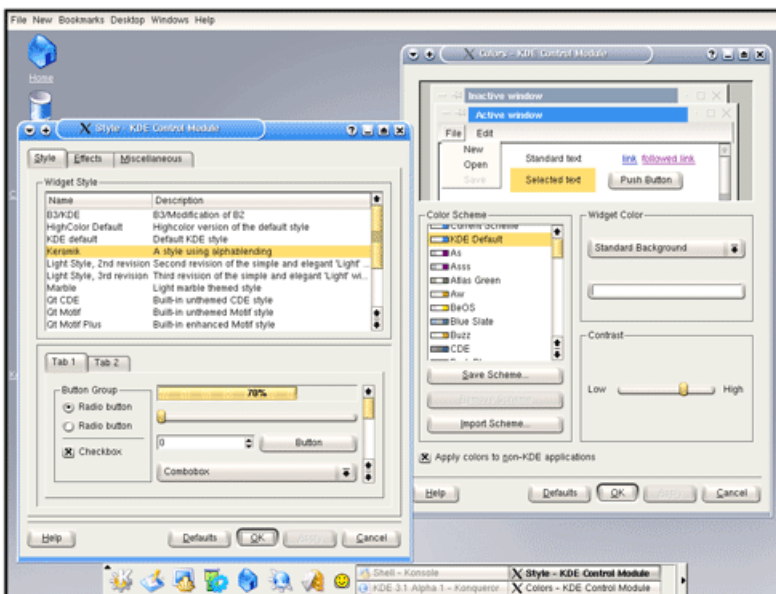
Un altre tipus de programari molt relacionat amb X-Window és el que s'encarrega de proporcionar un entorn integrat per a les aplicacions, l'escriptori, les eines d'administració del sistema, etc. Els més populars que hi ha actualment són KDE (the K Desktop Environment) i GNOME (GNU Object Model Environment). Tots dos proporcionen un *toolkit* particular, un entorn d'escriptori amb moltes funcionalitats i configuracions diferents i una llista d'aplicacions integrades que cada vegada va creixent més. La majoria de les distribucions de GNU/Linux i UNIX proporcionen algun d'aquests dos entorns d'escriptori per ser molt amigables i aportar eines i programari propi de gran qualitat que ajuden en gran manera l'usuari a configurar el sistema i el propi escriptori. Tots dos poden funcionar amb qualsevol *window manager* que compleixi una sèrie de característiques bàsiques. En les figures següents podem veure l'aspecte de tots dos:

Figura 10.3



GNOME

Figura 10.4



KDE

Finalment, un altre tipus d'aplicació que s'utilitza en X-Window és el *session manager*, que són una sèrie de programes que permeten desar la configuració d'una determinada sessió d'usuari perquè en arrencar de nou X-Window es carreguin les aplicacions que tingui configurades. Generalment, en els entorns integrats ja s'incorporen aquestes eines de manera automàtica; si no és així, podem recórrer al que la mateixa infraestructura d'X-Window proporciona: `xsm`.

Activitat

10.1. Llegiu la història d'X-Window en l'article: http://www.linux-mag.com/2001-12/xfree86_01.html.

Activitat

10.2. Vegeu alguns dels *window manager* i entorns d'escriptori existents a: <http://www.xwinman.org>.

10.2. Configuració

Actualment, la majoria de les targetes gràfiques del mercat ja estan suportades, però és possible que des del moment d'aparició al mercat d'una nova targeta fins que es dona suport en X-Window passin unes setmanes o uns quants mesos. De tota manera, cada vegada més els mateixos fabricants estan donant suport a GNU/Linux i, en alguns casos, ja estan proporcionant els seus propis *drivers* per a aquest sistema operatiu. Tot i així, abans de comprar una nova targeta gràfica, sempre és recomanable comprovar si hi ha disponible algun tipus de *driver* per a la distribució que estiguem utilitzant. Per a instal·lar XOrg 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 *xorg*, *xserver-xorg*, etc. i porten implícites diverses dependències de fonts i algunes utilitats bàsiques per al maneig d'X-Window. Una vegada instal·lats aquests paquets, hem de configurar adequadament els dispositius dels quals disposem per a poder arrencar correctament el client i servidor X-Window. Segons la distribució que utilitzem, es fa ús d'un o d'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 detallarem a continuació classificats segons el dispositiu que cal configurar:

- Targeta gràfica
 - *Driver*: les diferents famílies de targetes gràfiques porten uns microprocessadors específics i utilitzen unes funcions determinades per a realitzar les seves operacions. Per aquesta raó, hem d'indicar el *driver* 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 *driver* necessita la nostra targeta amb l'ordre `discover ##xdriver video`.
 - *Identificador*: l'identificador de la targeta pot ser qualsevol nom amb què ens vulguem referir a la nostra targeta. Aquest identificador és utilitzat 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 és imprescin-

dible indicar aquesta quantitat (el sistema la detecta automàticament), sí que és recomanable especificar-la en la configuració.

- *Utilització del framebuffer del nucli*: el *frambuffer* del nucli és un *driver* especial de Linux que permet fer algunes operacions sobre X-Window. Encara que la seva utilització no és obligatòria, generalment s'utilitza perquè el servidor d'X-Window es pugui comunicar directament amb el nucli del sistema. De tota manera, si ens donés algun problema, sempre la podem desactivar.
- Teclat
 - *Regla XKB*: perquè el servidor d'X-Window pugui manejar correctament el teclat, necessita saber quines regles aplicar-hi. Per a la majoria dels 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 104 tecles (els identifiquem amb el nom "pc104"). Els teclats que no porten aquestes tecles s'identifiquen com de 101 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 personalitzar algunes de les tecles del teclat.
- 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, USB, 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 de triar el que més s'ajusti al nostre model i fabricant. Generalment, sabent el model del ratolí ja podrem escollir adequadament l'opció que li correspon.
 - *Emulació de 3 botons*: en cas que el nostre ratolí només tingui 2 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 3 botons.

- 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 dels 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 de monitor o d'un altre varia considerablement.
 - *Característiques*: en la configuració de característiques es preguntarà les resolucions màximes que pot obtenir el nostre monitor, la freqüència d'actualització, 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-les adequadament per a poder aprofitar al màxim les seves 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 en poden ser més o menys segons el programa que utilitzem, tota la configuració es desa en el fitxer `/etc/X11/XF86Config-4`.

Aquest fitxer està organitzat a les diferents seccions que hem anat veient i, recorrent al seu manual, veurem que tenim moltes més possibilitats que ens donen una flexibilitat total per a configurar com vulguem les nostres X-Window. Per tal de provar si realment funcionen, podem executar `x`, amb la qual cosa 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").

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, podem canviar la seva 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" tornariem a la gràfica).

xsession

Quan utilitzem `startx` o `xinit` s'executen les instruccions del fitxer `/etc/X11/xsession`. Si en el directori *home* de l'usuari que iniciés X-Window hi hagués un fitxer `.xsession`, s'executarien les instruccions d'aquest en lloc de les de l'altre.

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 arrencar X-Window amb els valors:

- `console`: qualsevol usuari que sigui 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 el qual generalment no es permet que els usuaris treballin amb l'entorn gràfic per la despesa de recursos que això suposa.

10.3. X display manager

En el subapartat anterior hem vist com configurar X-Window de manera local. Tal com hem anat comentant al llarg de l'apartat, 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 efectui les funcions d'un client X-Window, hem d'instal·lar algun tipus de *X display manager*. Aquests programes obren un port de comunicacions amb què els clients es poden comunicar amb el servidor i treballar amb X-Window de manera remota. Encara que hi ha molts programes d'aquest tipus, un dels primers que va aparèixer, i en el qual es basen molts d'altres, és l'`xdm`.

Els *X display manager* poden actuar tant de manera local com remota. Entre altres funcions, el que fan és mostrar una pantalla (en l'entorn gràfic) perquè l'usuari s'identifiqui amb el seu *login* i contrasenya. Funcionen com qualsevol altre *daemon* del sistema, de manera que el seu inici i aturada es pot configurar com vulguem (utilitzant els nivells d'execució que el sistema proporciona). Hem de tenir en compte que si el configurem perquè funcioni de manera local, en engegar el sistema ens trobarem amb la pantalla d'identificació gràfica i no amb 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 per defecte s'estaran executant.

Quan instal·lem `xdm`, tots els seus fitxers de configuració es deixaran en el directori `/etc/X11/xdm`. Repassarem què conté cadascun d'aquests fitxers:

Taula 10.2

<code>xdm-config</code>	Localització dels arxius de configuració d' <code>xdm</code> .
-------------------------	--

xdm.options	Opcions globals de configuració.
Xaccess	Definició dels equips remots als quals deixem accedir.
Xservers	Servidors locals d'xdm.
Xresources	Configuració de la pantalla de <i>login</i> : colors, fonts, mida, etc.
Xsetup	<i>Script</i> que s'executarà quan s'arrenqui xdm.
Xstartup	<i>Script</i> que s'executarà quan un usuari entri a X-Window. Se solen posar accions relacionades amb l'xdm.
Xsession	<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 l'execució del fitxer <code>/etc/X11/Xsession</code> .
Xreset	<i>Script</i> que s'executarà en acabar una sessió d'usuari.

La configuració dels servidors locals la trobem en el 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-hi 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 en com cada directiva inclou un `:0` o `:1`, que fa referència a la instància d'X-Window que utilitzem. Per defecte, sempre s'utilitza la 0, però en voler més d'un servidor local l'hem de referenciar d'aquesta manera. Al final de cadascuna d'aquestes línies podríem afegir paràmetres especials per a cada servidor d'X-Window (en `man x` trobem tots els possibles), com la profunditat de color que volem per a cadascun, la resolució de la pantalla, etc. D'aquesta manera, podríem treballar amb diferents sessions d'X-Window obertes tal com fèiem amb les consoles.

Seguretat d'X-Window

Si en el fitxer `d'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 de manera remota sense cap tipus d'encriptació pot suposar un forat de seguretat, amb la qual cosa és molt recomanable informar-se'n adequadament abans d'utilitzar-lo.

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 defec-

te inutilitza totes les connexions que es reben. Una vegada realitzats 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 en les quals només volem instal·lar el servidor d'XWindow, hauríem d'instal·lar X-Window tal com vèiem en l'apartat anterior i utilitzar l'ordre `X -query IP`, en què la 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 així successivament.

Xnest

Xnest és un servidor d'X-Window que ens permet obrir en una finestra una altra instància d'X-Window.

11. Taller d'X-Window

11.1. Introducció

En el segon taller es van establir les bases per a la correcta manipulació i gestió de paquets, i vam aprendre a configurar alguns dispositius. Tot i això, i a causa de la seva complexitat, no es va abordar el tema de la configuració de la targeta gràfica o, més ben 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 dur a terme un repàs exhaustiu de tot el sistema, ja que aquesta tasca seria probablement inabordable per raons molt diferents. Es pretén establir les bases perquè cadascú sigui capaç de configurar el seu propi sistema en funció de la targeta gràfica de la qual disposi, dels seus gustos i de les seves preferències. En finalitzar el taller, hauríem de ser capaços d'instal·lar un entorn X, així com de saber treure partit de la seva increïble potència.

Activitat

11.1. A causa de la complexitat del sistema X, es recomana la lectura dels documents següents per a establir conceptes abans de començar a treballar sobre el sistema. A més, aquests ens aportaran coneixements suplementaris, que poden ser posats en pràctica al llarg del taller.

a) XWindow-Overview-HOWTO. Es tracta d'un document senzill que serveix per a assimilar els conceptes bàsics que concerneixen el sistema.

b) XWindow-User-Howto. És un document amb continguts més avançats que l'anterior, però també recomanable.

11.2. Instal·lació i configuració del servidor X

Com ja s'ha dit, l'X Window System és un sistema molt complex que integra moltes biblioteques i aplicacions, algunes de les quals són fonamentals per al seu funcionament, encara que la majoria només s'ha d'instal·lar si, per les nostres necessitats, ens calen. Aquesta és una de les raons per les quals, en Debian, el sistema és distribuït en molts paquets diferents, dels quals només instal·larem aquells que siguin necessaris.

11.2.1. Diferents estratègies per a la instal·lació dels paquets

Atès 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 correcte funcionament 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 estem instal·lant i, a més, pot implicar l'omissió d'alguns paquets essencials, als quals, pel motiu que sigui, no s'hagi fet referència durant el càlcul de dependències. Per aquest motiu, es recomana la construcció del sistema pas a pas, per a anar compronent quins paquets s'estan instal·lant en cada moment, i el perquè.

11.2.2. Instal·lació de paquets bàsics

Joc de xips i dispositiu

El primer pas és determinar el joc de xips de la targeta gràfica i el dispositiu que la suporta. Feu córrer en un terminal l'ordre `lspci` i busqueu les paraules `VGA compatible controller`:. La informació que segueix normalment identifica almenys la marca de la targeta gràfica i possiblement el dispositiu que necessiteu. Per exemple,

```
$lspci
01:03.0 VGA compatible controller: ATI Technologies Inc ES1000 (rev 02)
```

La primera columna llança els números del bus PCI a què està connectada la targeta amb el format `<bus>:<slot>:<func>`.

En executar

```
$lspci -n -s bb:ss.f
```

identifiquem amb els números del bus PCI el `vendor` i `device` ID de la targeta. Per exemple,

```
$lspci -n -s 01:03.0
01:03.0 0300: 1002:515e (rev 02)
```

El `vendor` i `device` ID tenen el format `vvvv:dddd` en nombres hexadecimal. En aquest cas, el `vendor` i `device` ID de la targeta és `1002:515e`. Amb aquesta informació busquem en la llista de targetes gràfiques (<http://www.calel.org/pci-devices/xorg-device-list.html>) el dispositiu que necessitem usar. En aquest cas, el dispositiu és `radeon`.

També necessitem saber les freqüències d'escombratge horitzontal i vertical del monitor. Els escombratges se solen trobar a la secció d'especificacions del manual del monitor. Aquestes dades són molt importants i uns valors erronis poden causar un mal funcionament i fins i tot danys al monitor.

Instal·lació d'X.org

Des de l'interpret d'ordres, com a `root`, escrivim:

```
#apt-get install xorg
```

Aquest és un metapaquet que, usant el sistema de dependències, instal·la un conjunt de paquets X11, com el servidor X.Org, un conjunt de tipus de lletres i eines bàsiques.

A continuació, veurem pas a pas la configuració del paquet `xserver-xorg` feta per `debconf`. Si ens equivoquem en alguna cosa, cancel·lem amb "Ctrl+C" i tornem a reconfigurar amb,

```
#dpkg-reconfigure xserver-xorg
```

Dispositiu X

Triem de la llista el dispositiu que millor vagi amb la targeta gràfica, segons la identificació del joc de xips que hem fet a l'inici. Per exemple, `ati`.

Hi donem un nom, per exemple, per omisió, `Generic Video Card`.

Hi donem la identificació de la targeta al bus PCI, per exemple, `PCI:1:0:0`.

Gairebé sempre ho autodetecta. Deixem en blanc la quantitat de memòria perquè el servidor l'autodetecti. Triem no usar el dispositiu *framebuffer* del nucli.

Teclat

Triem no autodetectar el teclat. Escollim el llenguatge del teclat (`us` per omisió, `es` per a castellà). Triem les regles del teclat `xorg` (per omisió). Escollim el tipus de teclat (`pc104` per omisió). Altres opcions són `pc101`, `pc102`, `pc105`. Aquests dos últims són per a teclats europeus. Deixem la variant i opcions del teclat en blanc.

Ratolí

Triem el dispositiu del ratolí, per exemple, `/dev/input/mice`.

Si el ratolí està connectat a un port de comunicació sèrie, `/dev/ttyS0` correspon al port COM1, `/dev/ttyS1` al port COM2, etc. La unitat `/dev/psaux` és per a ratolins PS/2 i `/dev/input/mice` per a ratolins USB.

Triem el protocol del ratolí, per exemple, `ImPS/2`.

Escollim emular un ratolí amb tres botons.

Mòduls

Triem els mòduls per omisió.

Ruta d'arxius

Triem la configuració per omisió de la secció "Files".

Monitor

Triem autodetectar el monitor. Hi donem un nom, per exemple, per omisió, Generic Monitor.

Escollim amb la tecla de l'espai les resolucions que suporta la targeta gràfica i el monitor, típicament 1.280 x 1.024 o 1.024 x 768. Escollim la manera avançada de configurar el monitor. Introduïm el rang de freqüències de l'escombratge horitzontal, després el vertical. Aquests valors o rangs se solen trobar a la secció d'especificacions del manual del monitor. Si no coneixem aquesta informació, és preferible introduir un rang gran, per exemple, 28-100 per a l'escombratge horitzontal i 40-100 per al vertical. Escollim la profunditat de color, típicament 24 bits.

Execució de *scripts*

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 l'ordre anterior.

11.3. Configuració d'X

L'arxiu `/etc/X11/xorg.conf` conté la configuració d'X.Org i està dividit en seccions:

```
Files      # Ruta dels arxius
Module     # Mòduls dinàmics
InputDevice # Descripció dels dispositius perifèrics
Device     # Descripció dels dispositius gràfics

Monitor   # Descripció del monitor
Screen    # Configuració de la pantalla
ServerLayout # Esquema global
DRI       # Configuració específica a DRI
```

Cada secció comença amb la instrucció `Section` seguida del nom de la secció entre cometes i acaba amb `EndSection`. Hi ha ordres específiques a cada secció. Anirem pas a pas, no necessàriament en l'ordre de l'arxiu, i descriurem les seccions i explicarem el significat de cada ordre, per poder modificar la configuració amb algun editor de text (per exemple `nano`) amb els paràmetres pertinents al maquinari.

Per a fer efectius canvis a `etc/X11/xorg.conf` / hem de reiniciar el *display manager* amb

```
#!/etc/init.d/gdm restart
```

11.3.1. Secció "Device"

Aquesta secció és la que defineix i configura el dispositiu de la targeta gràfica.

L'entrada `Identifier` és simplement un nom per a identificar la secció.

```
Section "Device"
    Identifier "Generic Video Card"
    Driver     "sis"
    BusID      "PCI:1:0:0"

EndSection
```

L'entrada `Driver` n'especifica el dispositiu. En aquest exemple, el dispositiu `sis` suporta moltes varietats de targetes gràfiques amb el joc de xips SiS. Altres dispositius comuns són: `cirrus`, `ati`, `r128` (ATI Rage 128), `radeon` (ATI Radeon), `s3virge`, `savage`, `trident`, `tseng`. Podem usar el mètode descrit a l'inici per a determinar el dispositiu.

L'entrada `BusID` identifica la targeta gràfica al bus PCI, que podem determinar amb `lspci`.

11.3.2. Secció "Monitor"

Aquesta secció defineix el monitor.

```
Section "Monitor"
    Identifier "Generic Monitor"

    Option      "DPMS"
    HorizSync   28-100
    VertRefresh 40-100

EndSection
```

L'entrada `Identifier` és similar a la de la secció *Device*. Hi pot haver diverses seccions *Monitor* en l'arxiu, cadascuna amb diferent identificació. Això és convenient si, per exemple, tenim diversos monitors, per exemple, un a casa i un altre al nostre lloc de treball. Així podem definir fàcilment una configuració per a quan l'ordinador és a casa o a la feina.

L'entrada `HorizSync` especifica la freqüència d'escombratge horitzontal del monitor en unitats de kHz. Pot ser una freqüència fixa, [31.5], múltiples freqüències fixes, [31.5, 35.2], un rang, [30-64], o rangs, [15-25, 30-64]. L'entrada `VertRefresh` especifica els intervals verticals d'actualització en unitats de Hz i els valors tenen el mateix format que `HorizSync`.

11.3.3. Secció "Screen"

Aquesta secció és la que va definir la pantalla, i combina la configuració del dispositiu i monitor.

```
Section "Screen"
    Identifier "Default Screen"
    Device      "Generic Video Card"
    Monitor     "Generic Monitor"
    DefaultDepth 24

    Subsection "Display"
        Depth    1
        Modes    "1024x768" "800x600" "640x480"
    EndSubsection
    Subsection "Display"
        Depth    4
        Modes    "1024x768" "800x600" "640x480"
    EndSubsection
    Subsection "Display"
        Depth    8
        Modes    "1024x768" "800x600" "640x480"
    EndSubsection

    Subsection "Display"
        Depth    15
        Modes    "1024x768" "800x600" "640x480"
    EndSubsection
    Subsection "Display"
        Depth    16
        Modes    "1024x768" "800x600"
    EndSubsection

    Subsection "Display"
        Depth    24
        Modes    "1024x768"
    EndSubsection

EndSection
```

Observem que les entrades `Device` i `Monitor` tenen el mateix valor que l'entrada `Identifier` de les seccions `Device` i `Monitor`. D'aquesta manera, s'entrellacen les configuracions de les diferents seccions.

L'entrada `DefaultDepth` especifica la profunditat de color per omissió, en aquest exemple, 24 bits. Hi pot haver diverses subseccions `Display`. L'entrada `Depth` especifica la profunditat de color de la subsecció. Valors possibles són 8 (256 colors), 16 (64 K colors) i 24 bits. Així, l'entrada `DefaultDepth` especificarà la subsecció `Display` per omissió.

L'entrada `Modes` n'especifica les resolucions. Es pot especificar una resolució ["640x480"] o una llista d'elles ["1600x1200 1280 x 960 1152 x 864 1024 x 768 800 x 600 640x480"]. Normalment s'usa la primera en la llista, si està suportada, si no la següent, i així successivament.

En general, com més gran sigui la profunditat de color, menor serà la resolució màxima possible. Disminuirem la profunditat de color si per a certa resolució suportada la imatge de la pantalla és dolenta, o mantindrem la profunditat disminuint-ne la resolució.

11.3.4. Secció "InputDevice"

Aquesta és la secció que configura els perifèrics com el teclat, ratolí, *mousepad*, *touchscreen*, etc. Els més comuns són el teclat i el ratolí, òbviament, cadascun separatament.

Configuració del teclat:

```
Section "InputDevice"
    Identifier "Generic Keyboard"
    Driver      "kbd"
    Option      "CoreKeyboard"
    Option      "XkbRules"      "xorg"
    Option      "XkbModel"      "pc105"

    Option      "XkbLayout"      "es"
EndSection
```

Tots els teclats tenen un dispositiu en comú anomenat `kbd`. L'entrada `Option "CoreKeyboard"` indica que el teclat definit per la secció és el teclat principal. L'entrada `Option "XkbModel"` especifica el tipus de teclat. Els valors més comuns són "pc101", "pc102", "pc104", "pc105" o "microsoft" per al teclat Microsoft Natural. L'entrada `Option "XkbLayout"` defineix el llenguatge del teclat; "es" per a un teclat en castellà, "us" per a un teclat en anglès.

Configuració del ratolí:

```
Section "InputDevice"
    Identifier "Configured Mouse"
    Driver      "mouse"
    Option      "CorePointer"
    Option      "Device"          "/dev/input/mice"

    Option      "Protocol"        "ImPS/2"
    Option      "Emulate3Buttons" "true"
    Option      "Buttons"         "5"
    Option      "ZAxisMapping"    "4 5"
EndSection
```

El dispositiu comú *mouse* suporta quatre tipus de ratolins: sèrie, Bus, PS/2 i USB. En el primer exemple, es configura un ratolí USB IntelliMouse (amb roda).

L'entrada `Option "CorePointer"` indica que el ratolí definit per la secció és el ratolí principal.

L'entrada `Option "Device"` especifica el dispositiu de la unitat del ratolí. En són exemples `/dev/ttyS0` (serial), `/dev/psaux` (PS/2) i `/dev/input/mice` (USB). En general, existeix l'enllaç simbòlic `/dev/mouse` que apunta a aquest dispositiu.

L'entrada `Option "Protocol"` defineix el tipus de ratolí. Altres protocols són:

```
"MouseMan"
"MouseSystems"
"IntelliMouse"
"ExplorerPS/2"
"ThinkingMouse"

"ThinkingMousePS/2"
"NetScrollPS/2"
"NetMousePS/2"
"GlidePoint"
"GlidePointPS/2"
"MouseManPlusPS/2"
```

Els ratolins sèrie antics de dos o tres botons són normalment suportats pel protocol "Microsoft" o "MouseMan". Els ratolins sèrie amb roda els suporta el protocol "IntelliMouse" i els PS/2 "ImPS/2". El protocol "auto" de vegades ajuda si el maquinari és capaç de detectar automàticament el ratolí.

L'entrada `Option "Emulate3Buttons"` emula el botó del mig prement simultàniament els botons esquerre i dret. Es pot usar també en ratolins amb tres botons.

11.3.5. Secció "ServerLayout"

Aquesta secció és la que relaciona la pantalla amb els perifèrics.

```
Section "ServerLayout"
    Identifier "Default Layout"
    Screen      "Default Screen"
    InputDevice "Generic Keyboard"
    InputDevice "Configured Mouse"

EndSection
```

Observem, en particular, com aquesta secció ho combina tot mitjançant les identifikacions de cada secció definides per `Identifier`.

11.3.6. Secció "DRI"

Algunes targetes modernes incorporen *direct rendering infrastructure* (DRI). Per a usar-lo heu de carregar els mòduls "glx" i "dri" a la secció "Modules".

La secció DRI per omissió és

```
Section "DRI"

    Mode 0666

EndSection
```

Algunes targetes que no suporten DRI poden penjar X.

11.3.7. Secció "Files"

Aquesta secció defineix la ruta d'arxius necessaris per a X. En particular, defineix la ruta dels tipus de lletres en les entrades "FontPath".

11.3.8. Inicialització del servidor

Ha arribat el moment de comprovar si el fitxer de configuració del servidor és correcte i, en conseqüència, el servidor s'arrenca com cal. Per a això n'hi ha prou que executem l'ordre `startx`.

Si tot funciona correctament, al cap d'uns moments la nostra pantalla adquirirà un fons mallat de colors grisosos, 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 acabar atorgant 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-se sobre `Exit` opció `Yes, reallyquit`, 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/Xorg.0.log`) i intentar detectar les possibles fonts d'errors. Les més comunes solen ser: mala elecció del *driver* 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 *driver* que l'*script* ha escollit és el correcte), posar la directiva `UseFBDev` a `true`, quan ha d'estar a `false`, usar resolucions o freqüències d'actualització més altes de les que la targeta pot suportar, etc.

Arribats a aquest punt, mitjançant l'ordre següent evitarem que cada vegada que arrenquem l'ordinador entrem en el mode gràfic:

```
brau:~# rm /etc/rc2.d/S99xdm
```

Això ens serà útil mentre no tinguem configurat totalment aquest entorn. Una vegada acabat, ja serà cada usuari qui decidirà si vol o no vol 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 l'ordre anterior:

```
brau:~# cd /etc/rc2.d
brau:/etc/rc2.d# ln -s ../init.d/xdm S99xdm
```

11.3.9. El fitxer de *log*

Tot i que l'arrencada del servidor hagi resultat reeixida, no per això hem de deixar d'analitzar el contingut del fitxer principal de *log* del sistema X, `/var/log/Xorg.0.log`, ja que això ens ajudarà a depurar errades i errors menors que no han interromput l'arrencada del servidor, però sí que fan baixar el seu rendiment. Alguns exemples típics d'això poden ser: eliminar la línia del fitxer de configuració que fa referència a les fonts ciríl·liques, si no les utilitzem mai, ja que ni tan sols les hem instal·lat; eliminar d'aquest mateix fitxer tot el que fa referència al Generic Mouse, ja que el fa incompatible amb el que hem configurat nosaltres, etc.

11.4. *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 de la seva presentació, així com 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, per la qual cosa potser ens interessa instal·lar-ne alguns de més complexos, com pot ser `WindowMaker`, `BlackBox`, `qwm`, etc. Així doncs, 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 mateix estudiant qui decideixi quin utilitzarà. 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 managers* corrent en un mateix ordinador, cadascun 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 que executem la instrucció:

```
brau:~# apt-get install qwm
```

En aquest moment, ja tenim dos *window managers* instal·lats: el `twm` i el `qwm`. Per a fer-ne córrer l'un o l'altre n'hi haurà prou que indiquem la ruta completa d'ubicació del *window manager* que vulguem utilitzar després de l'ordre `startx` (recordem que l'ordre `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 que executem:

```
brau:~# startx /usr/bin/X11/qwm
```

Per a utilitzar el `twm` s'hauria d'haver especificat la seva ruta completa de la manera següent:

```
brau:~# startx /usr/bin/X11/twm
```

Haver d'especificar cada vegada quin *window manager* volem utilitzar, una vegada ens haguem decidit per un en concret, pot ser una mica pesat. Per a especificar quin *window manager* s'ha d'usar en cas que després de l'ordre `startx` no se n'especifiqui cap en concret, crearem l'arxiu `.xsession` en el directori arrel de l'usuari amb el contingut següent, en cas que volguéssim que el *window manager* per defecte fos el `twm`, per exemple:

```
# ~/.xsession exec twm
```

Si volguéssim que fos `qwm` el *window manager* per defecte, n'hi hauria prou que canviéssim `twm` per `qwm`. L'execució dels diferents processos que impliquen l'arrencada de l'entorn gràfic, així com els fitxers de configuració que es van llegint durant aquest procés, estan fortament determinats. Així doncs, en crear el fitxer anterior, el que hem fet és editar un dels últims fitxers (els que es troben 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

dels que s'han determinat per defecte dins del sistema i que són definits en els fitxers residents a `/etc/X11` i els seus subdirectoris. Per a acabar aquest subapartat 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:

```
brau:~# 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 de l'ordre `update-alternatives` per a establir-lo:

```
brau:~# update-alternatives x-window-manager
```

Lectura complementària

Animem els estudiants a familiaritzar-se una mica amb aquest *window manager* i que visitin la seva pàgina per a ampliar els seus coneixements: <http://www.windowmaker.org>.

11.5. X Session manager

Els *session managers* són programes que poden fer-se córrer sobre una sessió gràfica i que ens permetran establir-ne i modificar-ne paràmetres. `xms` és el *session manager* que hi ha 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 l'ordre `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 estan corrent 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 solen ser una part més dels *desktop managers*, i estan tan integrats que de vegades resulta difícil reconèixer les seves accions. Un exemple típic d'això és la pregunta que se'ns formula sobre si volem desar la sessió en tancar KDE.

11.6. X Display manager

En finalitzar la instal·lació de l'Xserver, vam suggerir que s'eliminés l'enllaç simbòlic `/etc/rc2.d/S99xdm` per tal d'evitar que en tornar a arrencar el sistema en entrar al `runlevel2` s'executés `xdm`, acrònim de *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*:

```
brau:~# /etc/init.d/xdm start
```

Per a aturar-lo, també utilitzarem el mateix procediment que seguiríem per a detenir qualsevol altre *daemon*, amb l'excepció que hem de prémer la combinació de tecles "Ctrl+Alt+F1", per 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:

```
brau:~# /etc/init.d/xdm stop
```

Com hem comprovat, el *display manager* ens demana un *login* i una contrasenya, els mateixos que utilitzem per a accedir al sistema per les `ttys`, si no és que hem imposat alguna restricció. Després de validar-nos, entrem en el mode gràfic de la mateixa manera com ho fèiem mitjançant l'ordre `startx`. La diferència rau en el fet que, quan acabem la sessió gràfica, el servidor no s'aturarà, sinó que continua corrent l'`xdm`.

Un dels inconvenients d'`xdm` és que no ens permet seleccionar amb quin *window manager* volem treballar. Però hi ha altres *display managers*, com poden ser `wdm` (de WindowMaker), `gdm` (del projecte GNOME) o `kdm` (del projecte KDE), que sí que ho permeten.

Podem instal·lar el `wdm`, per a veure el seu aspecte i per a conèixer un altre *display manager*:

```
brau:~# apt-get install wdm
```

En executar-se l'*script* de postinstal·lació, se'ns preguntarà quin *display manager* volem usar, `xdm`, que ja teníem instal·lat, o `wdm`. Seleccionarem aquest últim perquè es creï l'enllaç necessari perquè es llanci `wdm` 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'arrenqui automàticament cap *display manager* en arrencar el sistema, n'hi haurà prou que eliminem els enllaços necessaris, és a dir, el fitxer `/etc/rc2.d/wdm`.

Una vegada arrencada un sessió X des del *display manager*, és a dir, una vegada haguem llançat el *window maker* pertinent, pot ser interessant executar l'ordre `ps tree` per a veure les relacions de dependència entre els diferents processos que estan corrent en aquest moment, juntament amb la informació que ens aportarà la línia `ps aux`.

11.7. Desktop managers

L'aparició de diferents *toolkits*, així com el desenvolupament de diversos projectes que desenvolupaven o usaven biblioteques 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 que pretenen establir les bases per a

la unificació i estandardització, tant de presentació com de polítiques de programació i de desenvolupament d'aplicacions. Un dels primers a 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, atès el seu alt grau d'implementació i de desenvolupament, dedicarem un subapartat respectivament. Però abans podem esmentar altres *desktop managers*, com poden ser: GNUStep, ROX, GTK+Xfce o UDE.

11.7.1. GNOME

GNOME és un projecte que forma part de GNU, que es caracteritza per no necessitar estrictament un *window manager* en concret, encara que es recomana que se n'usi algun que garanteixi el seu correcte funcionament (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 tauler de control disposa sempre d'un *window manager selector* que ens permet escollir quin *window manager* volem usar. GNOME es basa en Gtk toolkit, les pròpies biblioteques 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 tauler de control: GNOME Control Panel. Per a dur a terme una instal·lació bàsica de GNOME, instal·larem el paquet següent juntament amb totes les seves dependències:

```
brau:~# 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, doncs, Sawfish, que va ser desenvolupat estrictament per a complir aquest requisit. Instal·lem el paquet i totes les seves dependències:

```
brau:~# apt-get install sawfish-gnome
```

Tenim, doncs, un altre *window maker* instal·lat. Aturarem el *display manager* i tornarem a llançar-lo perquè aquest nou *window maker* 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. La segona consisteix a 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 propi *session manager* de GNOME l'encarregat de fer i registrar els canvis necessaris perquè això ocorri).

Una vegada familiaritzats una mica amb GNOME, el que podem fer és instal·lar alguns paquets que ens poden ser útils, en concret `gnome-help` i `gnome-terminal`; el primer ens ofereix una interfície en la qual podrem llegir manuals (*mans*), fitxers de text en un entorn gràfic, i el segon instal·la l'`xterm` propi de GNOME.

11.7.2. 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, `Konqueror` i de la seva utilitat de configuració: `Control Panel`. Òbviament, KDE pot estar instal·lat en el mateix sistema en què haguem 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 una visita a la seva pàgina web per a conèixer les seves possibilitats: <http://www.kde.org>. Així mateix, podem executar la línia següent per veure la integració de KDE en Debian:

```
brau:~# apt-cache search kde
```

Els paquets bàsics de KDE són al paquet `kdebase`. Aquest serà, doncs, el primer que hi instal·larem:

```
brau:~# apt-get install kdebase
```

Novament, haurem de reinicialitzar el nostre *window manager* per tal de tenir accés al *desktop manager* de nou instal·lat. Una vegada fet això, podem procedir a instal·lar el gestor d'arxius, paquet `konqueror`. Mitjançant el paquet `kde-i18n-es` podem instal·lar els fitxers necessaris perquè KDE treballi en castellà.

A partir d'aquest punt, ja serà cada usuari qui instal·larà els diferents paquets del projecte que li siguin d'interès. Tal com ja fèiem, per a preestablir el *window maker* per defecte, utilitzarem el menú de l'ordre `update-alternatives` per a seleccionar el *session manager*:

```
brau:~# update-alternatives x-session-manager
```

11.8. Personalització d'alguns aspectes

El disseny de l'entorn gràfic X respon a l'ambiciós objectiu de treure el màxim rendiment del maquinari disponible, usant el mínim dels seus recursos, i oferir la màxima flexibilitat possible. L'estructura de servidor client en què es basa aquest sistema possibilita que aquests objectius siguin una realitat, i les aparents dificultats amb què es troba l'usuari novell desapareixen ràpidament amb una mica de pràctica per a permetre que aflorin els múltiples avantatges que ofereix aquest disseny. En aquest subapartat només es pretén donar una

idea succinta de la potència d'aquest sistema, la qual es posa plenament de manifest quan es treballa en xarxa, encara que queda una mica entenebrida quan es treballa en un sistema *stand alone* com sobre el que es desenvolupa tot aquest curs. De tota manera, s'introdueixen alguns conceptes que poden ser d'utilitat quan s'entra a treballar en xarxa.

11.8.1. Personalització d'aspectes locals

En general, els arxius de configuració es troben en el directori `/etc/X11/` o en alguns dels seus subdirectoris. De manera personalitzada, cada usuari pot redefinir els paràmetres de configuració i afegir-ne de nous creant o editant en el seu directori de *home* 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 de *home* 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 en `$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'arrenqui automàticament `bbkeys` en *background* en iniciar la sessió, aquest contindrà les línies següents:

```
bbkeys
blackbox
```

Xresources

En 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 en el fitxer:

```
Xterm*reverseVideo: true
```


L'ordre `xrdb` és l'encarregada 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 l'ordre. Si 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 que provenen del teclat (*server-independent*) en símbols del sistema (*server-dependent*). La taula de conversió que cal fer servir ha estat seleccionada durant el procés de configuració del teclat, però l'ordre `xmodmap` ens permet modificar el seu contingut. Un exemple del seu ús pot ser el següent:

```
brau:~# xmosmap -e "keycode 127 = Delete"
brau:~# xmosmap -e "keycode 22 = BackSpace"
```

Assignació de símbols

Aquestes ordres van ser utilitzades durant molt temps a causa de la inversió d'assignació de símbols a les taules de conversió; actualment, tanmateix, aquest problema està resolt.

Mitjançant els paràmetres `-pk`, `xmodmap` ens tornarà tot el contingut de la taula de conversió que s'està utilitzant.

11.8.2. Personalització d'aspectes de xarxa

Els aspectes aquí presentats 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 en el 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 tot això, si volguéssim llançar `xeyes` des del primer terminal gràfic, via `xterm`, i "displayar-lo" en el segon, procediríem de la manera següent:

```
brau:~$ set DISPLAY :0.1; export DISPLAY
brau:~$ xeyes
```

Si entrem en una sessió gràfica, obrim un `xterm`, canviem d'usuari mitjançant l'ordre `su` i provem de llançar una aplicació gràfica, se'ns tornarà un missatge d'error que ens indicarà 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 necessiten permisos de `root`, ja que ens evitarà haver d'entrar en l'entorn gràfic com a `root` (pràctica no molt recomanable, i que, encara que per defecte es permeti, moltes vegades es restringeix manualment).

xhost i xauth

L'ordre `xhost` permet establir quins equips poden accedir al servidor gràfic de manera remota, és a dir, quines màquines client poden llançar una aplicació per a 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. L'ordre `xauth` serveix per a determinar quins usuaris poden llançar aplicacions sobre el servidor gràfic. Així doncs, mitjançant la combinació d'aquestes dues ordres podem establir una política de seguretat d'accés al servidor X bastant raonable. `xhost +` per als *stand alone*.

11.9. Configuració d'impressores

La tasca de configurar impressores pot ser facilitada a partir de l'entorn gràfic. Hi ha multitud d'aplicacions per a configurar el sistema d'impressió natiu, i altres que substitueixen aquest sistema per un de propi, comunament també basat en l'estructura client servidor. Per a instal·lar CUPS (Common Linux Printing System), caldrà instal·lar el paquet del servidor d'impressió, `cupsys`; 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 de les ordres habituals en el sistema d'impressió de BSD. Per exemple, podríem instal·lar el sistema d'impressió mitjançant l'ordre:

```
# apt-get install cupsys cupsys-client cupsys-bsd printconf \ foomatic-filters-ppds
```

Aquest últim paquet conté els PPD (*postscript printer definition*) d'Adobe, que utilitza CUPS per a definir les impressores.

En GNOME podem anar a "Sistema/Administración/Impresoras". Fem doble clic a "Impresora nueva", hi introduïm la clau de `root` i seguim els passos per a definir la impressora, depenent de si és local o de xarxa. En el segon pas busquem el fabricant i model de la impressora. Una vegada definida, marquem la impressora amb el botó dret del ratolí i marquem "Propiedades". Marquem "Convertirse en administrador" i modifiquem la configuració si és necessari.

En la seqüència d'imatges següent es pot observar el procés d'instal·lació d'una impressora mitjançant l'assistent `gnome-cups`:

Figura 11.1

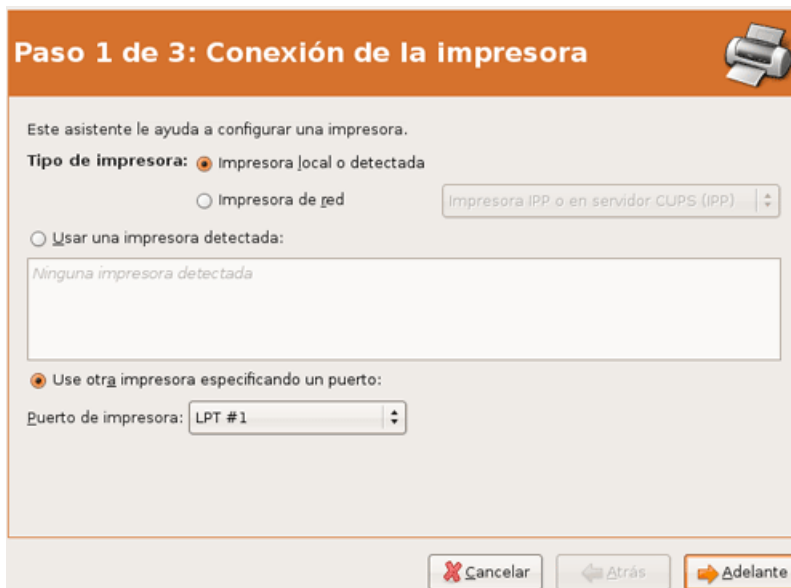


Figura 11.2



Figura 11.3



Paso 3 de 3: Información de la impresora

Nombre: Deskjet-720C

Descripción: Inyección color

Ubicación: Oficina

Cancelar Atrás Aplicar

Una vegada instal·lada la impressora, ens apareix a la finestra de l'eina `gnome-cups`:

Figura 11.4



Si la impressora usa el sistema HP JetDirect és millor instal·lar-hi el paquet `hplip`,

```
# apt-get install hplip
```

i usem l'eina `hp-setup` per a definir la impressora.

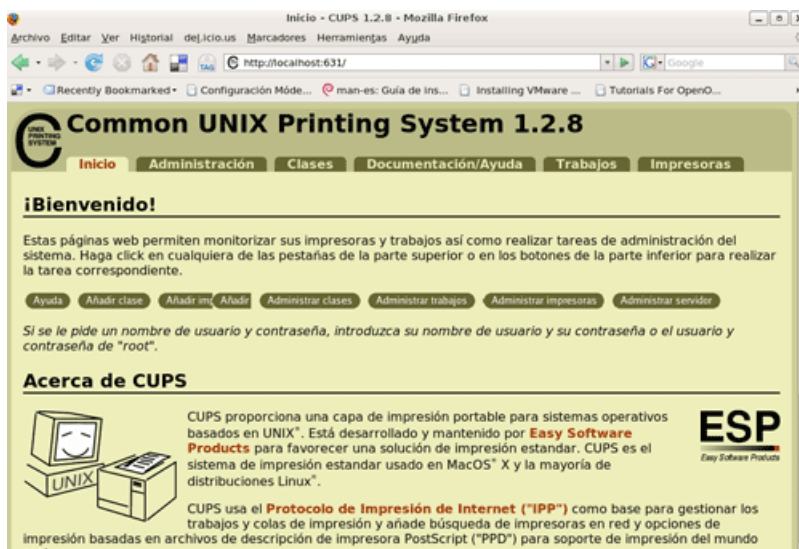
Una altra manera de configurar una impressora mitjançant CUPS és des de la pàgina web del servei CUPS. Únicament cal obrir un navegador i escriure l'URL: `http://localhost:630`, que és el port en el qual escolta el servidor.

En les captures de pantalla següents, es pot observar l'aspecte d'aquesta utilitat d'administració:

Figura 11.5



Figura 11.6



11.10. OpenOffice

En aquest subapartat, encara que s'allunyi una mica de l'objecte principal del curs, presentarem una *suite* ofimàtica que pot ser d'extraordinària utilitat per a aquells que estiguin acostumats a utilitzar programari d'aquest tipus. Es tracta d'OpenOffice.org, projecte derivat de StarOffice, de Sun Microsystems. Es pot destacar que aquest projecte és multiplataforma i que, per tant, pot ser implementat en altres sistemes operatius no del tipus UNIX.

Per a instal·lar la *suite*, ho podem fer mitjançant l'ordre següent (tot en una línia):

```
brau:~# apt-get install openoffice.org openoffice.org-base openoffice.org-calc
```

```
openoffice.org-common openoffice.org-core openoffice.org-draw openoffice.org-  
evolution openoffice.org-gnome openoffice.org-gtk openoffice.org-help-en  
openoffice.org-help-es openoffice.org-impress openoffice.org-java-common  
openoffice.org-l10n-es openoffice.org-math openoffice.org-writer
```

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 *home* de cada usuari hi hagi un petit directori en el qual es desaran les seves configuracions personals.

Una vegada s'hagi instal·lat el programa la primera vegada, cada usuari haurà d'executar el programa seguint perquè es creï el seu directori:

```
/usr/lib/openoffice/program/setup
```

Una vegada fet això i s'hagin respost algunes preguntes, mitjançant l'ordre `openoffice` obrirem la *suite*.

Actualment, la versió d'OpenOffice en Debian Etch és la 2.0.4.

11.11. Conclusió

Amb aquest taller s'acaba el material didàctic del segon mòdul del màster. Hi hem après a instal·lar l'entorn gràfic en el sistema, que, com ja vam veure en el seu moment, no és una part fonamental dins del sistema operatiu. Però és evident la seva utilitat en molts casos, i molts afirmen que el sistema operatiu que ha estat objecte d'estudi en aquest mòdul pot ser una alternativa seriosa a altres sistemes. Per tot això, els autors voldríem manifestar el nostre absolut convenciment 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 podem destacar la seva filosofia, robustesa, adaptabilitat, potència, nivells potencials de seguretat, etc. Estem convençuts que aquest sistema operatiu és una aposta de futur per a la qual, si bé ja ha demostrat que és capaç d'obrir-se un espai al món dels sistemes operatius, només es poden esperar sorpreses positives. Esperem haver proporcionat els coneixements suficients i transmès l'entusiasme necessari perquè el lector iniciï aquí el seu propi camí en el món del GNU/LINUX, així com haver sabut obrir les portes a una comunitat en què tothom és benvingut i respectat.